TECHNISCHE UNIVERSITÄT MÜNCHEN

FAKULTÄT FÜR INFORMATIK

LEHRSTUHL FÜR COMPUTER GRAPHIK UND VISUALISIERUNG

Interactive 3D visualization of ensemble weather forecasts

Marc Rautenhaus

Vollständiger Abdruck der von der Fakultät für Informatik
der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. J. Schlichter
Prüfer der Dissertation: 1. Univ.-Prof. Dr. R. Westermann
2. Univ.-Prof. Dr. G. C. Craig,
Ludwig-Maximilians-Universität München

Die Dissertation wurde am 18.05.2015 bei der Technischen Universität München
eingereicht und durch die Fakultät für Informatik am 22.07.2015 angenommen.

to my family

Abstract

This thesis investigates the feasibility of interactive 3D visualization of ensemble weather predictions. Weather forecasting requires meteorologists to explore large amounts of numerical weather prediction data, and to assess the uncertainty of the predictions. Visualization methods that facilitate fast and intuitive exploration of the data hence are of particular importance.

I present *Met.3D*, a new open-source tool for the interactive 3D visualization of numerical ensemble weather predictions. The tool has been developed with the primary motivation to support weather forecasting during aircraft-based atmospheric field campaigns, however, is applicable to further forecasting, research and teaching activities. My work approaches challenging topics related to the visual analysis of numerical atmospheric model output – 3D visualisation, ensemble visualization, and how both can be used in a meaningful way suited to weather forecasting. Met.3D builds a bridge from proven 2D visualization methods commonly used in meteorology to 3D visualization by combining both visualization types in a 3D context. I address the issue of spatial perception in the 3D view and present approaches to using the ensemble to allow the user to assess forecast uncertainty. Interactivity is key to the approach. Met.3D uses modern graphics technology to achieve interactive visualization on standard consumer hardware. The tool supports forecast data from the European Centre for Medium Range Weather Forecasts (ECMWF) and can operate directly on ECMWF hybrid sigma-pressure level grids. I describe the system architecture and the employed visualization algorithms, and analyse the impact of the ECMWF grid topology on computing 3D ensemble statistical quantities.

Focussing on a particular use case, I present the application of Met.3D to forecasting warm conveyor belt (WCB; large scale air streams associated with mid-latitude low pressure systems) situations. Motivated by forecast requirements of the T-NAWDEX-Falcon 2012 campaign, a method to predict 3D probabilities of the spatial occurrence of WCBs has been developed. Probabilities are derived from Lagrangian particle trajectories computed on the forecast wind fields of the ECMWF ensemble prediction system. Integration of the method into Met.3D facilitates interactive visualization of WCB features and derived probabilities in the context of the ECMWF ensemble forecast. I investigate the sensitivity of the method with respect to trajectory seeding and grid spacing of the forecast wind field. Furthermore, I propose a visual analysis method to quantitatively analyse the contribution of ensemble members to a probability region and, thus, to assist the forecaster in interpreting the obtained probabilities. A case study, revisiting a forecast case from T-NAWDEX-Falcon, illustrates the practical application of Met.3D and demonstrates the use of 3D and uncertainty visualization for weather forecasting and for planning flight routes in the medium forecast range.

Zusammenfassung

Die vorliegende Arbeit untersucht die interaktive 3D-Visualisierung von Ensemble-Wettervorhersagedaten. Die Wettervorhersage erfordert die Auswertung großer Mengen numerischer Vorhersagedaten sowie die Abschätzung der Prognosesicherheit. Visualisierungsmethoden, die eine schnelle und intuitive Analyse der Daten ermöglichen, sind daher von besonderer Bedeutung.

Ich stelle “Met.3D”, ein neues Open-Source-Tool für die interaktive 3D-Visualisierung numerischer Ensemble-Wettervorhersagen, vor. Die Entwicklung des Tools wurde in erster Linie durch Anforderungen an die Wettervorhersage während Flugmesskampagnen in der Atmosphärenforschung geleitet. Das System ist jedoch auch in anderen Bereichen der Vorhersage sowie in der Forschung und Lehre einsetzbar. Meine Arbeit behandelt Themen, die in Bezug auf die visuelle Analyse von Daten numerischer Atmosphärenmodelle eine Herausforderung darstellen: 3D-Visualisierung und Ensemble-Visualisierung sowie deren nutzbringende Verwendung in der Wettervorhersage.

Met.3D stellt eine Verbindung zwischen bewährten und allgemein in der Meteorologie eingesetzten 2D-Visualisierungsmethoden und 3D-Visualisierungsmethoden her. Dabei werden beide Visualisierungsarten in einem 3D-Kontext kombiniert. Ich befasse mich mit der räumlichen Wahrnehmung in der 3D-Ansicht sowie mit Darstellungsmethoden, die es dem Nutzer ermöglichen, Unsicherheiten in der Vorhersage zu erschließen. Interaktivität ist entscheidend für meinen Ansatz. Met.3D nutzt moderne Grafiktechnologie, um interaktive Darstellungen auf Standardhardware zu ermöglichen. Die Software unterstützt Vorhersagedaten des Europäischen Zentrums für mittelfristige Wettervohersage (EZMW). Daten, die auf dem hybriden Sigma-Druck-Gitter des EZMW vorliegen, können ohne Umwandlung visualisiert werden. Ich beschreibe die Systemarchitektur sowie die in Met.3D verwendeten Visualisierungsverfahren und analysiere den Einfluss von Gittereigenschaften der EZMW-Daten auf die Berechnung von 3D-Ensemble-Statistiken.

Des Weiteren behandele ich den konkreten Anwendungsfall der Vorhersage von “Warm Conveyor Belts” (WCBs; großskalige Luftströmungen in extratropischen Tiefdruckgebieten). Ausgehend von den Anforderungen der Wettervorhersage während der T-NAWDEX-Falcon Kampagne 2012 wurde eine Methode entwickelt, um 3D-Auftretenswahrscheinlichkeiten von WCBs vorherzusagen. Die Wahrscheinlichkeiten werden von Lagrangeschen Partikeltrajektorien abgeleitet, die mit Hilfe vorhergesagter Windfelder des EZMW-Ensemble-Vorhersagesystems berechnet werden. Die Integration der Methode in Met.3D ermöglicht die interaktive Visualisierung von WCB-Merkmalen sowie deren Wahrscheinlichkeiten im Kontext der EZMW-Ensemble-Vorhersage. Ich untersuche die Abhängigkeit der Wahrscheinlichkeiten von der verwendeten Trajektorienkonfiguration sowie von der Auflösung der Windfelder.

Darüber hinaus schlage ich eine visuelle Analysemethode vor, die den Beitrag einzelner Ensemble-Member zur ermittelten Wahrscheinlichkeit analysiert und damit die Interpretation der erhaltenen Wahrscheinlichkeiten unterstützt. Eine Fallstudie betrachtet die praktische Anwendung von Met.3D für einen Vorhersagefall von T-NAWDEX-Falcon. Sie zeigt den Nutzen von 3D- und Unsicherheitsvisualisierungstechniken für die Wettervorhersage sowie für die Flugplanung im Mittelfristbereich.

Acknowledgements

I gratefully acknowledge the support of everyone who made this thesis possible. First, I would like to thank my advisor Prof. Dr. Rüdiger Westermann for giving me the opportunity to realise this thesis. I came from an atmospheric science background and was keen to learn about visualization and computer graphics in order to apply techniques from these fields to meteorology. Prof. Dr. Rüdiger Westermann was open to my ideas and gave me freedom and confidence to pursue my goals. I am very grateful for his support, advice and time commitment.

Furthermore, I would like to thank my second referee Prof. Dr. George C. Craig for his interest in my work, his time, and his constructive feedback on this thesis.

In particular, I would like to thank my colleagues from DLR, Dr. Andreas Schäfler and Dr. Andreas Dörnbrack, for many long and fruitful discussions. I am very grateful to Dr. Christian Grams from ETH Zurich for great cooperation and to my student assistant Michael Kern for extensive help with the implementation of Met.3D. Also, Dr. Patrick Jöckel from DLR provided support and helpful advice.

I would like to thank Dr. Stephan Siemen and the visualization team at ECMWF for their interest in my work, their support, and for accommodating me in their group several times during the past years.

I also thank my former and current colleagues Dr. Stefan Auer, Dr. Kai Bürger, Dr. Matthäus G. Chajdas, Shunting Cao, Rachel Chu, Ismail Demir, Dr. Christian Dick, Sebastian Eberhardt, Marie-Lena Eckert, Florian Ferstl, Dr. Roland Fraedrich, Mihaela Jarema, Mathias Kanzler, Dr. Johannes Kehrer, Dr. Tobias Pfaffelmoser, Dr. Florian Reichl, Prof. Dr. Nils Thuerey, Dr. Marc Treib, Dr. Mika Vaaraniemi, Susanne Weitz, Sebastian Wohner, Dr. Jun Wu for constructive discussions, help with implementation issues, and help with organisational issues.

I am very thankful to the Evangelisches Studienwerk Villigst for funding my work by means of a doctoral scholarship, and to the people in Villigst for providing guidance and supportive activities.

Finally, I am enormously thankful to Judith and to my parents. Your support has been tremendous.

Contents

CHAPTER 1

Introduction

> "It was recognized early in the development of what is now ensemble forecasting that **graphical display would be an important means** of conveying the resulting complex information to forecasters (Epstein and Fleming, 1971; Gleeson, 1967), and **operational experience is still accumulating** regarding the most effective means of doing so."
>
> Wilks (2011, *Statistical Methods in the Atmospheric Sciences*, p. 275)

Weather forecasting requires meteorologists to explore large amounts of numerical weather prediction (NWP) data, and to assess the uncertainty of the predictions. Visualization methods that facilitate fast and intuitive exploration of the data hence are of particular importance. In practice, the forecasting process for the most part relies on two-dimensional (2D) visualization methods. Meteorologists use weather maps, vertical cross-sections and a multitude of meteorological diagrams to depict the data. From these image sources, they build "mental models" of the three-dimensional (3D), time-varying forecast atmosphere inside their heads (Hoffman and Coffey, 2004; Trafton and Hoffman, 2007).

Despite the 3D nature of the atmosphere, 3D visualization methods have not found widespread usage, even though there have been promising attempts in the 1990s and early 2000s that suggested added value (Treinish and Rothfusz, 1997; Koppert et al., 1998; McCaslin et al., 2000). Various hindering factors are discussed in the literature, including resistance of forecasters to adapt to new 3D visualization methods that are decoupled from their "familiar" 2D products (Koppert et al., 1998; Szoke et al., 2003), problems with spatial perception in 3D renderings (Szoke et al., 2003), as well as issues due to limited performance (Treinish and Rothfusz, 1997) and the need for dedicated graphics workstation hardware (Koppert et al., 1998).

In addition to 3D space and time, forecast visualization has in recent years become more challenging through the increased use of ensemble weather predictions (e.g. Gneiting and Raftery, 2005; Leutbecher and Palmer, 2008). These sets, or *ensembles*, of forecast runs provide information on forecast uncertainty by representing uncertainty in the real-world observations on which a forecast run is based, and uncertainty in the numerical model that is used to compute a forecast. The European Centre for Medium Range Weather Forecasts (ECMWF), for example, operates the Ensemble Prediction System (ENS), comprising 50 perturbed forecast runs and an unperturbed control run (Buizza et al., 2006; Miller et al., 2010). These 51 forecast members approximate the distribution of possible future weather scenarios (Leutbecher and Palmer, 2008). Ensemble products have become a major tool to assess forecast uncertainty. The development of visualization methods that depict the uncertainty derived from ensemble data is an active topic of research not only for weather forecast ensembles (Obermaier and Joy, 2014). Yet again, ensemble visualization techniques related to weather forecasting published so far mainly focus on two dimensions as well (e.g. Potter et al., 2009; Sanyal et al., 2010).

This thesis is concerned with 3D visualization of ensemble prediction data for use in weather forecasting. I introduce a new open-source visualization tool, *Met.3D*, that provides interactive 3D visualization techniques for ensemble prediction data. There has been an immense progress in mainstream graphics hardware capabilities in recent years. Making use of these developments, Met.3D facilitates interactive visualization of present-day NWP datasets on consumer hardware. The tool has been developed as a new effort to demonstrate the feasibility of using 3D visualization for forecasting, this time also considering uncertainty information from ensemble datasets. It is intended to be used for actual forecasting tasks, as well as a platform to implement and evaluate new 3D and ensemble visualization techniques.

My work has been inspired by a particular application, forecasting the weather situation to plan research flight routes during aircraft-based field campaigns. Specifically, initial work was driven by the desire to have a forecast visualization tool available for campaigns with involvement of the DLR (Deutsches Zentrum für Luft- und Raumfahrt) Institute of Atmospheric Physics (DLR-IPA) that would allow for more interactive and more intuitive exploration of 3D ensemble prediction data. While the methods presented in this thesis are applicable to a broader range of forecasting and visual data analysis tasks, I focus on the application of research flight planning throughout the thesis. Particular focus is put on the requirements of the international T-NAWDEX-Falcon campaign (THORPEX - North Atlantic Waveguide and Downstream Impact Experiment - Falcon, hereafter TNF) and its specific application case of forecasting warm conveyor belts (WCBs; strong, ascending, and often rain producing airstreams associated with mid-latitude weather systems; e.g. Browning and Roberts, 1994; Eckhardt et al., 2004; Pfahl et al., 2014). Serving as a representative example of a research campaign, TNF is used as the "real-world context" for the development of Met.3D and for examples given throughout the thesis.

The thesis ties together elements from the disciplines of visualization, weather forecasting, and flight planning. It is the purpose of this introductory chapter to present a brief overview of the context that motivated the presented work and to specify the objectives that are addressed. The chapter is concluded with an outline of the following chapters.

1.1 Motivation: Weather forecasting during aircraft based research campaigns

An essential element of atmospheric research is in-situ and remote sensing observations from aircraft. For this purpose, a fleet of various aircraft targeting atmospheric regions from the boundary layer to the stratosphere is in use (e.g. Schumann, 2007). To carry out successful research flights for atmospheric measurements, thorough weather forecasting and subsequent flight planning is as essential as the aircraft and its instruments. The focus of forecasting during aircraft-based research campaigns is to find specific regions of interest (in which targeted atmospheric features or processes can be observed) in the NWP data. Subsequently, a flight route is designed considering the scientific objectives of the campaign, the predicted atmospheric situation, and instrument, aircraft and legal constraints. Both fast forecast exploration and forecast uncertainty assessment play a major role in campaign forecasting:

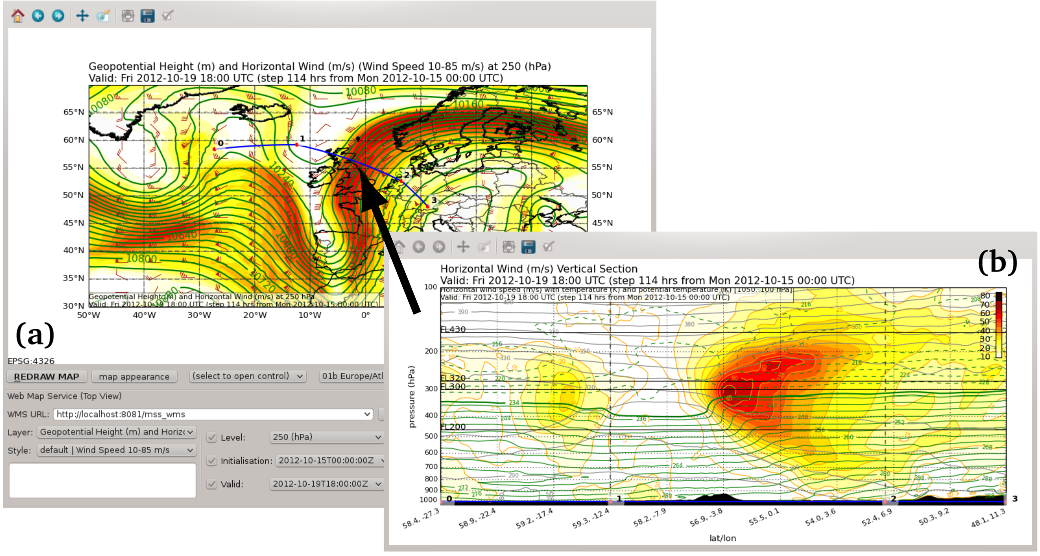
1. When investigating suitable meteorological conditions to specify the route of a research flight (that is, waypoints in 3D space and time), the forecaster is required to examine the NWP data in a short period of time. Atmospheric features relevant to the flight have to be identified quickly, and findings have to be communicated to colleagues. Upper-level features typically important to research flights with high-flying aircraft are of an inherently three-dimensional nature (for example, clouds, jet streams, or the tropopause). However, from the available literature, as well as from my experience in campaigns with DLR involvement, visualization used during campaigns has been solely based on 2D methods, typically with limited interactivity (e.g. Flatøy et al., 2000; Blakeslee et al., 2007; He et al., 2010; Rautenhaus et al., 2012). I am hence interested in investigating how 3D visualization methods and interactivity (to quickly navigate the data space) can be used to aid the forecast exploration.

2. Assessing the forecast's uncertainty has become indispensable as flights frequently have to be planned multiple days before take-off (typically three to seven days; the medium forecast range) to obtain the required approval from air traffic authorities. While the use of ensemble predictions has been reported for recent field campaigns (e.g. Wulfmeyer et al., 2008; Elsberry and Harr, 2008; Ducrocq et al., 2014; Vaughan et al., 2015), they have, to the best of my knowledge, not been used to create specific interactive forecast products for flight planning. However, ensembles provide valuable information; for example, 3D probability fields for the occurrence of a targeted atmospheric process or feature can be derived. Potential flight routes can be planned in regions in which the probability is high. Ensemble forecasts are readily available, for example, from ECMWF. An open question, however, is how the ensemble data can be visualized to improve flight planning in the medium forecast range.

In Rautenhaus et al. (2012), I have introduced a web-service-based tool called the *Mission Support System* (MSS). The development of the MSS was motivated by the demand at the German Aerospace Centre's Institute of Atmospheric Physics (DLR-IPA) for having a tool available for Germany's new High Altitude and Long Range (HALO) research aircraft[1] that is focused on the primary needs of mission scientists responsible for planning a research flight. The MSS provides online access to horizontal maps and vertical cross-sections of numerical weather prediction data and in particular allows the interactive design of a flight route in direct relation to the predictions.

Figure 1.1 shows example horizontal and vertical sections through a wind field forecast by ECMWF. Both types of charts are generated on-demand by the MSS. The user can change the vertical level, geographic region and map projection of the horizontal maps, as well as "draw" a flight track on a map and request a corresponding vertical section. However, the response times for image generation are on the order of seconds (Rautenhaus et al., 2012), thus limiting the real-time interactivity provided by the tool. Also, neither 3D views nor ensemble forecasts are supported.

[1] http://www.halo.dlr.de


Geopotential Height (m) and Horizontal Wind (m/s) (Wind Speed 10-85 m/s) at 250 (hPa)
Valid: Fri 2012-10-19 18:00 UTC (step 114 hrs from Mon 2012-10-15 00:00 UTC)
(a)
EPSG:4326
REDRAW MAP
map appearance
(select to open control)
01b Europe/Atl
Web Map Service (Top View)
Horizontal Wind (m/s) Vertical Section
Valid: Fri 2012-10-19 18:00 UTC (step 114 hrs from Mon 2012-10-15 00:00 UTC)
(b)
pressure (hPa)
lat/lon

WMS URL: http://localhost:8081/mss_wms

Layer: Geopotential Height (m) and Horiz

Style: default | Wind Speed 10-85 m/s

☑ Level: 250 (hPa)

☑ Initialisation: 2012-10-15T00:00:00Z

☑ Valid: 2012-10-19T18:00:00Z

Figure 1.1: State of the art at the beginning of the thesis work. Screenshots of the 2D *Mission Support System* (Rautenhaus et al., 2012), in the displayed example used to analyse the forecast jet stream. **(a)** Upper level map (horizontal section) of horizontal wind (colour and wind barbs) and geopotential height (line contours) at 250 hPa. **(b)** Vertical cross-section of wind speed (colour and orange line contours) and temperature (green line contours) along the flight route drafted on the horizontal map (black arrow). Shown is the ECMWF deterministic forecast from 00:00 UTC 15 October 2012, valid at 18:00 UTC 19 October 2012.

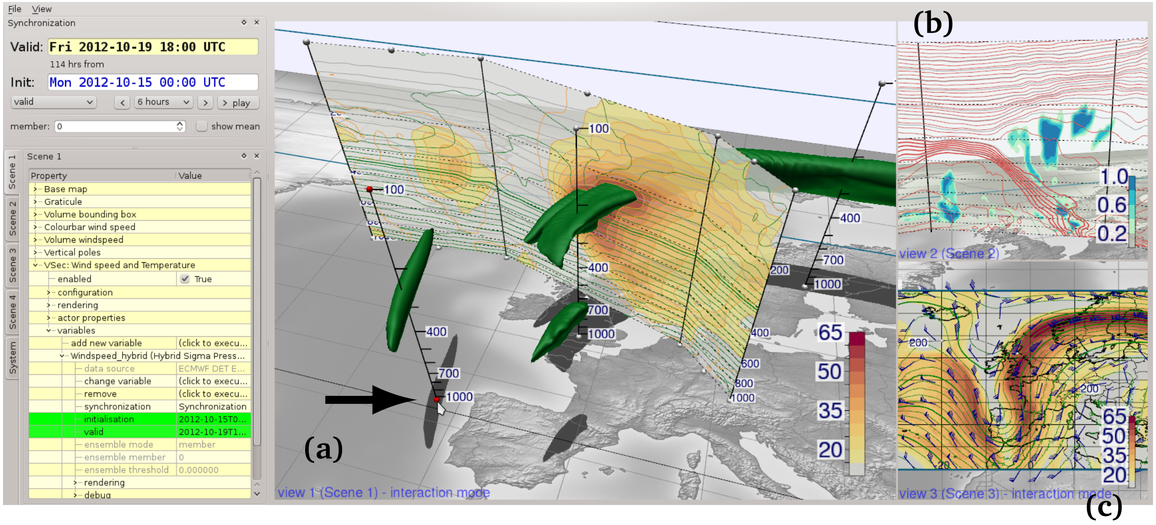
The major objective for the work presented in this thesis is to use interactive 3D visualization of ECMWF ensemble predictions to improve the forecast process for field campaigns. An example of the visualization tool Met.3D that resulted from the work is shown in Fig. 1.2. The same data as in Fig. 1.1 are displayed. In Met.3D, 2D visualizations very similar to those in the MSS are available. They are displayed in a 3D context and can be augmented by 3D visualization elements, for example, an isosurface. Met.3D is designed to support ensemble forecast data. Also, interactivity in Met.3D is instantaneous, that is, when the user changes, for instance, a waypoint of a vertical section, the visualization is updated immediately in real-time.

1.2 The application case: T-NAWDEX-Falcon

The T-NAWDEX-Falcon campaign took place in October 2012 at DLR's Oberpfaffenhofen base in southern Germany. The campaign's major objective was to take in-situ measurements in WCBs. Schäfler et al. (2014) provide details on the campaign and its flight planning.

In order to make the forecast data used in this thesis understandable, I provide a short introduction to WCBs. WCBs are Lagrangian airstreams in extratropical cyclones. They transport warm and moist air from the atmospheric boundary layer (ABL) in a cyclone's warm sector upward and poleward towards the tropopause. The term "WCB" was introduced by Harrold (1973) and Browning (1971) and consolidated in a conceptual airstream model for extratropical cyclones (also including the cold conveyor belt and the dry airstream) by Carlson (1980). WCBs are an atmospheric feature that has been in the focus of several aircraft-based campaigns (e.g. Pomroy and Thorpe, 2000; Vaughan et al., 2003; Schäfler et al., 2014; Vaughan et al., 2015).

WCB airmasses commonly ascend by about 500 to 600 hPa in 48 hours, thereby covering horizontal



Synchronization

Valid: Fri 2012-10-19 18:00 UTC

114 hrs from

Init: Mon 2012-10-15 00:00 UTC

valid ‹ 6 hours › › play

member: 0 □ show mean

Scene 1

Property	Value
Base map	
Graticule	
Volume bounding box	
Colourbar wind speed	
Volume windspeed	
Vertical poles	
VSec: Wind speed and Temperature	
enabled	☑ True
configuration	
rendering	
actor properties	
variables	
add new variable	(click to execu...
Windspeed_hybrid (Hybrid Sigma Press...	
data source	ECMWF DET E...
change variable	(click to execu...
remove	(click to execu...
synchronization	Synchronization
initialisation	2012-10-15T0...
valid	2012-10-19T1...
ensemble mode	member
ensemble member	0
ensemble threshold	0.000000
rendering	
debug	

Figure 1.2: The *Met.3D* visualization tool developed in the course of this thesis. Screenshot of the tool used to analyse the same forecast data as in Fig. 1.1. **(a)** A vertical section of the same data and along the same route shown in Fig. 1.1b, rendered by Met.3D in a 3D context and augmented by a 3D isosurface (55 ms^{-1}) of horizontal wind speed. **(b)** A synchronised vertical section of cloud cover, viewed from a different viewpoint. **(c)** The horizontal section from Fig. 1.1a, rendered by Met.3D. All elements can be interactively moved by the user by picking and dragging the spherical handles (illustrated by the red handle of the vertical axis marked by the black arrow).

distances of up to 2000 km (e.g. Wernli and Davies, 1997; Eckhardt et al., 2004). Due to the strong ascent, condensation leads to strong latent heat release and the formation of clouds and precipitation (e.g. Browning, 1986). Therefore, WCBs are highly relevant for precipitation extremes in the extratropics (e.g. Pfahl et al., 2014). Once the airmasses reach jet level, an outflow region forms near the tropopause. This region is characterised by cirrus clouds that extend over several thousand kilometres along the jet stream. Readers interested in further detail are referred to, for example, Browning (1990) for an overview, Eckhardt et al. (2004) and Madonna et al. (2014) for a climatology, and Browning (1986) and Pfahl et al. (2014) for relevance in large-scale precipitation.

A total of nine scientific flights were conducted in the scope of TNF, divided into three intensive observation periods (IOPs) with each period representing a WCB event over Europe (Fig. 1.3a). During the campaign, Met.3D development had not yet been progressed to the current state. However, in this thesis I frequently use archived forecast data from the campaign to illustrate the developed visualization techniques. The central application case for the presented visualization examples (including Figs. 1.1 and 1.2) is the TNF forecast case of October 19, 2012 (IOP 3 in Fig. 1.3a). The satellite image in Fig. 1.3b provides a real-world observation of major features that appear in the visualizations: A distinct narrow trough was located to the west of the British Isles. Upstream of the trough the former Hurricane *Rafael* transformed into a strong mid-latitude cyclone. East of the trough, ascending WCB airmasses formed a cloud band extending from Spain to the British Isles. The clouds further stretch along a jet stream over southern Scandinavia and the Baltic Sea.

1.3 Objectives

In this thesis, I follow up on the above cited works on 3D visualization in forecasting by Treinish and Rothfusz (1997), Koppert et al. (1998), and McCaslin et al. (2000). For the specific application case of

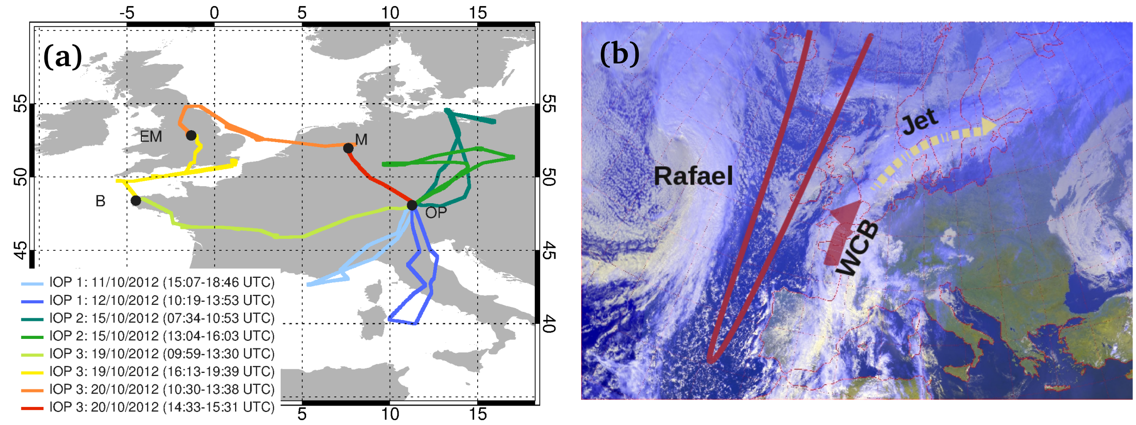

(a)
-5
0
5
10
15
55
50
45
40
EM
M
B
OP
IOP 1: 11/10/2012 (15:07-18:46 UTC)
IOP 1: 12/10/2012 (10:19-13:53 UTC)
IOP 2: 15/10/2012 (07:34-10:53 UTC)
IOP 2: 15/10/2012 (13:04-16:03 UTC)
IOP 3: 19/10/2012 (09:59-13:30 UTC)
IOP 3: 19/10/2012 (16:13-19:39 UTC)
IOP 3: 20/10/2012 (10:30-13:38 UTC)
IOP 3: 20/10/2012 (14:33-15:31 UTC)
(b)
Rafael
Jet
WCB

Figure 1.3: **(a)** Flights conducted during the three intensive observation periods (IOPs) of T-NAWDEX-Falcon (image courtesy of Andreas Schäfler). **(b)** Real-world context for IOP 3, used as an example throughout this thesis: Visible Meteosat satellite image of Europe and the North Atlantic of 12:00 UTC 19 October 2012, (Meteosat operated by EUMETSAT, image processing by DLR-IPA). Important features are the narrow trough to the west of the British Isles (dark red line), the former Hurricane *Rafael* and the WCB manifest in the cloud band east of the trough.

forecasting during aircraft-based field campaigns, I investigate the questions of

- how reported shortcomings can be overcome, and
- how reported methodologies can be extended to support the visualization of uncertainty information from ensemble forecasts.

As noted above, I focus on the requirements of TNF. The major forecasting challenge during the campaign was to predict the likelihood of WCB occurrence within aircraft range. This was expressed by five forecast questions to which I refer throughout the thesis:

- Forecast question A (FQ-A): How will the large scale weather situation develop over the next week, and will conditions occur that favour WCB formation?
- Forecast question B (FQ-B): How uncertain are the weather predictions?
- Forecast question C (FQ-C): Where and when, in the medium forecast range and within the spatial range of the aircraft, is a WCB most likely to occur?
- Forecast question D (FQ-D): How meaningful is the forecast of WCB occurrence?
- Forecast question E (FQ-E): Where will the WCB be located relative to cyclonic and dynamic features?

Given these questions, I formulate the major objective for my work as follows:

> *Ensemble weather prediction data shall be made accessible in an interactive 3D visualization environment that is suited to be used for forecasting during aircraft-based field campaigns. The visualization environment shall facilitate the intuitive exploration of atmospheric features relevant to a research flight, as well as the features' prediction uncertainty, in minimal time. In particular, the user shall be able to visually explore a forecast such that the five forecast questions relevant to TNF can be answered.*

I approach the objective in the following way:

- Review of recent approaches to 2D and 3D visualization of numerical simulations of the atmosphere. While no further studies on the usage of 3D visualization for forecasting have been published in the past decade, 3D visualization has been used to some extend in research environments.
- Analysis of requirements that are posed to a tool to fulfil the objective, and discussion of how well existing approaches are suited.
- Investigation of how recent developments in graphics technology can be utilised to achieve the objective.
- Development of the visualization tool *Met.3D*, making use of state-of-the-art graphics technology to facilitate interactive 3D ensemble visualization.
- Development of methods to predict and to visually analyse a probability of WCB occurrence from the ensemble forecasts. Integration of the methods into Met.3D.
- Demonstration of the tool's usefulness with a case study.

To limit the scope of the thesis, and following the application case "TNF", I focus on support for data from the ECMWF Ensemble Prediction System. Also, to facilitate a straightforward and intuitive user transition to the new 3D tool, I assume the 2D tools used at DLR (that is, in particular the MSS) to be the "state of the art".

1.4 Structure and content of the thesis

The thesis is structured as follows. Chapters 2 and 3 provide fundamental background information. To make the thesis accessible from both an atmospheric science and a visualization science perspective, relevant basics from both disciplines are introduced. In order to facilitate an understanding of the origin and the characteristics of the data that is to be visualized, basic principles of numerical ensemble weather prediction are reviewed in Chapter 2. The chapter introduces operational weather prediction methods (Sects. 2.1 and 2.2), discusses sources of uncertainty in forecasts and how they are approached with ensemble methods (Sect. 2.3), presents details of the ECMWF Ensemble Prediction System (Sect. 2.4), and surveys current methods used in operational forecasting to visualize ensemble products (Sect. 2.5). Basic principles of scientific data visualization are reviewed in Chapter 3, including the concept of the visualization pipeline and common visualization techniques (Sect. 3.1), recent developments in dedicated graphics hardware (Sect. 3.2), and approaches to visualization system design (Sect. 3.3). Readers familiar with either of the disciplines may skip the corresponding chapter.

To put the methods developed in the remainder of the thesis into the context of the literature, I review works in meteorological and ensemble visualization in Chapter 4. The chapter includes a brief history of visualization in meteorology (Sect. 4.1), and an overview of recent visualization software (Sect. 4.2) and literature (Sect. 4.3).

Chapter 5 introduces Met.3D. First, a rationale for the development of Met.3D is discussed in the context of the literature introduced in the previous chapters (Sect. 5.1). Section 5.2 presents Met.3D's visualization capabilities. When introducing 3D visualization to forecasting, we need to consider that the 2D visualization methods commonly used in meteorology provide many advantages (for example, spatial perception) and that meteorologists are used to working with them. In a 3D forecast tool to be used in practice, we hence have to be careful not to replace proven 2D methods, but to put them into a 3D context and to use 3D visualization to add value. I address the challenges of creating such a "bridge" from 2D to 3D visualizations, of improving spatial perception of 3D renderings and of designing interactive methods that provide fast and easy visual access to ensemble information. For 3D depictions, I propose *normal curves* to visualize the structure inside a transparent 3D isosurface. The method provides an intermediate means between a 2D section and a 3D isosurface.

Sections 5.3 and 5.4 address technical aspects. I propose a visualization pipeline architecture that is targeted at supporting ensemble forecast data by utilising recent graphics hardware and the multi-core capabilities of current consumer computers. To avoid time consuming preprocessing of the forecast data prior to visualization, Met.3D operates directly on the ECMWF hybrid sigma-pressure model grid. The characteristics of the data and resulting challenges for visualization are discussed along with Met3D's visualization algorithms and system architecture in Sect. 5.3. Section 5.4 discusses a challenge that arises from aiming at interactive ensemble visualization: the efficient yet accurate computation of statistical quantities from the ensemble predictions. For the considered application, the ECMWF model grid has an unfavourable property. When computing statistical quantities on a per-grid-point-basis an error is introduced, since the vertical positions of the grid points vary between members. Regridding to a common grid is a solution, albeit time consuming and hence undesirable for real-time visualization. I analyse the error introduced when ignoring such a regridding and provide advice on how to handle the issue.

In Chapter 6, I propose a technique to compute 3D probabilities of WCB occurrence. My approach is put into relation to previous work in the field (Sect. 6.1), and its integration into the Met.3D architecture is described. During TNF, we followed the approach of Wernli and Davies (1997) and used Lagrangian particle trajectories computed on the forecast wind field to objectively detect WCB airstreams. Using wind forecasts from the ECMWF ENS, trajectories were started from the ABL for each ensemble member. Those trajectories fulfilling a WCB criterion were gridded into 2D grids and displayed as probability maps showing the occurrence of either or all of WCB inflow, ascent, and outflow. However, generalising this approach to three dimensions poses challenges, as discussed in Sect. 6.2. I present an adapted approach using domain-filling trajectories, which is more accurate, albeit computationally more expensive. In order to find the best method that is still computationally tractable in a forecast setting, both approaches are compared in Sect. 6.3. I analyse their sensitivity to the grid spacing of the forecast wind fields and to the number and locations of the trajectory seeding points.

To facilitate quantitative interpretation of the obtained probabilities, I further propose a visual analysis method for cases in which only low probabilities of the occurrence of WCBs are encountered (Sect. 6.4). In such cases a flight often might not be planned due to the interpreted high uncertainty. However, low probability can have two causes. Either indeed only a small percentage of the ensemble members predict a WCB feature, or large spatial variation of the features in the individual ensemble members causes only marginal overlap and thus low probabilities. In the latter case the probability that a WCB will occur is actually much larger than suggested by the visualized probabilities. However, there is a large uncertainty in where it will occur. To help the user distinguish between these causes, I propose a method that identifies the contribution of individual members to a probability region.

After the introduction of all methods that are required to explore a forecast to answer questions FQ-A to FQ-E, Chapter 7 revisits the TNF forecast case of October 19, 2012. The case study shows how the proposed 3D ensemble visualization workflow is applied to campaign forecasting, and illustrates the use and added value of the presented methods.

The thesis is concluded with a summary and discussion in Chapter 8.

1.5 Publications, supplementary video and code availability

Publications

Major parts of this thesis (in particular Chapters 5, 6, 7, as well as parts of Chapters 1, 4 and 8) are based on the following journal articles (published under a Creative Commons Attribution 3.0 License):

- Rautenhaus, M., Kern, M., Schäfler, A., and Westermann, R.: Three-dimensional visualization of ensemble weather forecasts – Part 1: The visualization tool Met.3D (version 1.0), Geosci. Model Dev., 8, 2329–2353, 2015b

- Rautenhaus, M., Grams, C. M., Schäfler, A., and Westermann, R.: Three-dimensional visualization of ensemble weather forecasts – Part 2: Forecasting warm conveyor belt situations for aircraft-based field campaigns, Geosci. Model Dev., 8, 2355–2377, 2015a

Also, parts of this introduction and of Sect. 4.2 are based on the following journal article (published under a Creative Commons Attribution 3.0 License):

- Rautenhaus, M., Bauer, G., and Dörnbrack, A.: A web service based tool to plan atmospheric research flights, Geosci. Model Dev., 5, 55–71, 2012

A high-level description of the work was published in:

- Rautenhaus, M., Grams, C. M., Schäfler, A., and Westermann, R.: GPU based interactive 3D visualization of ECMWF ensemble forecasts, ECMWF Newsletter, 138, 34–38, 2014

A description of the T-NAWDEX-Falcon flight planning process was published in:

- Schäfler, A., Boettcher, M., Grams, C. M., Rautenhaus, M., Sodemann, H., and Wernli, H.: Planning aircraft measurements within a warm conveyor belt, Weather, 69, 161–166, 2014

Supplementary video

This thesis, in particular Chapters 5 and 7, contains static images of examples of the proposed visualization techniques. The images are complemented by video clips contained in the Supplements to the publications Rautenhaus et al. (2015b) and Rautenhaus et al. (2015a). The videos help to illustrate the interactive capabilities of Met.3D and are referred to throughout the thesis:

- Video 1: The supplementary video to Rautenhaus et al. (2015b) illustrates the visualization capabilities of Met.3D, focussing on the bridge from 2D to 3D, ensemble products and normal curves. It is available from the Geosci. Model Dev. homepage at `http://www.geosci-model-dev.net/8/2329/2015/gmd-8-2329-2015-supplement.zip`

- Video 2: The supplementary video to Rautenhaus et al. (2015a) illustrates the case study presented in Chapter 7. It is available from the Geosci. Model Dev. homepage at `http://www.geosci-model-dev.net/8/2355/2015/gmd-8-2355-2015-supplement.zip`

The videos are screen recordings realised on hardware consisting of a consumer-class six-core Intel Xeon running at 2.67 GHz, equipped with 24 GB of RAM, a 512 GB solid state drive and an Nvidia GeForce GTX 560Ti graphics card with 2 GB of video memory.

Code availability

To facilitate ease of deployment and of future research and developments, I have released the source code of Met.3D as open-source under the GNU General Public License, version 3. Please point your web browser to the software's repository at

`https://bitbucket.org/wxmetvis/met.3d`

to obtain an up-to-date version of the software. The repository also includes a short user guide to the software. I welcome user feedback as well as contributions that help with the further development of the code.

CHAPTER 2

Foundations of ensemble weather prediction

The purpose of this chapter is to survey the foundations of ensemble numerical weather prediction in order to describe the origin and characteristics of the data that is visualized in this thesis. Also, an overview is provided of visualizations of ensemble prediction output that are commonly used in operational forecasting.

Modern weather forecasting is based on real-world observations of the current state of the atmosphere and –the data source for the visualizations in this thesis– on numerical simulations that predict future atmospheric states. Observations are conducted using a global network of measurement platforms. Starting from an observed state of the atmosphere at a given time, numerical weather prediction models simulate the physical processes that govern atmospheric motion and constitution in order to predict a future state. This chapter briefly introduces the fundamental equations used for numerical simulations of the atmosphere. The weather forecasting process and sources of uncertainty in the process are surveyed, and it is discussed how the quantification of uncertainty can be approached by ensemble methods. For visualization, the structure of the discrete data output by numerical atmospheric models is important: Is the data stored on a regular or irregular grid, at which spatial and temporal resolution, including which parameters? Examples of numerical weather prediction models and ensemble prediction systems are presented and their characteristics are discussed. The ECMWF forecast system –as noted in Chapter 1, the source of data for this thesis– is presented in more detail. The chapter is completed with an overview of ensemble products commonly in use at operational meteorological centres.

2.1 Numerical weather prediction

A number of textbooks discuss the foundations of numerical weather prediction, including the book by Haltiner and Williams (1980) and recent texts by Jacobson (2005), Warner (2011), Coiffier (2012) and Inness and Dorling (2013). Readers interested in more detail than presented in this section are referred to these references. The history of modern weather prediction is described in detail by Nebeker (1995).

2.1.1 Dynamical equations

In its core, numerical weather prediction consists of numerically solving the equations of motion for the fluid "air" that constitutes the earth's atmosphere. The state of the air is described by its fundamental state variables temperature, density, pressure, as well as the three spatial components of wind speed. Additionally, water vapour and cloud ice and liquid water are important constituents of the atmosphere, as phase changes release or require energy. In the following, I outline the most important equations that are useful for understanding this thesis. The equations of motion are derived in detail in many textbooks on dynamic meteorology, including Dutton (1976), Etling (2002) and Holton (2004). In the following, the notation of Etling (2002) is used.

Three fundamental physical principles govern atmospheric motion: the conservation of mass, the conservation of energy, and the conservation of momentum. The equations take the form

$$\text{rate of change} = \text{local tendency} + \text{advection} = \text{forces/sources/sinks}. \tag{2.1}$$

When the forces, or alternatively, the sources and sinks that influence a parcel of air are known, the rate of change over time ds/dt of a parcel's scalar state variable s can be computed. Here, "rate of change" denotes the Lagrangian view; that is, the change that an air parcel experiences while it travels with the flow. To determine the temporal change at a fixed position (that is, the local tendency), the rate of change expands into local tendency and advection (Etling, 2002, Ch. 7): $ds/dt = \partial s/\partial t + \vec{v} \cdot \vec{\nabla} s$, where $\vec{v}$ denotes the wind vector.

The **conservation of momentum** originates from Newton's second law (acceleration equals force per unit mass), applied to a small air parcel in the earth's rotating coordinate system. In vector form, the conservation of momentum is written as (Etling, 2002, Ch. 9):

$$\underbrace{\frac{\partial \vec{v}}{\partial t}}_{\substack{\text{local wind speed tendency} \\ \text{(i.e. acceleration)}}} + \underbrace{(\vec{v} \cdot \vec{\nabla})\vec{v}}_{\text{advection}} = - \underbrace{2\vec{\Omega} \times \vec{v}}_{\text{Coriolis force}} - \underbrace{\vec{\nabla}\Phi}_{\text{gravity force}} - \underbrace{\frac{1}{\rho}\vec{\nabla}p}_{\text{pressure gradient force}} + \underbrace{\vec{F_r}}_{\text{friction force}} . \tag{2.2}$$

Here, $\vec{v} = (u, v, w)$ is the wind vector relative to the rotating earth's surface (its components given in ms^{-1}), $\vec{\Omega}$ is the angular velocity of the earth (the vector pointing in the direction of the earth's rotation axis), Φ is the geopotential (defined as $\Phi = gz$ with $g \approx 9.81\text{ms}^{-1}$ being the gravitational acceleration of the earth and z denoting the geometric height (m) above mean sea level), ρ is the density of the air (kg m^{-3}) and p its pressure (Pa). Equation 2.2 states that the change of wind speed with time at a given fixed position in space (Eulerian point of view) is due to advected higher or lower wind speeds, plus acceleration due to the Coriolis force (causing air to be deflected to the right from its original direction on the northern hemisphere and to the left on the southern hemisphere), to gravity, to inhomogeneous spatial pressure distributions (which accelerate the air towards lower pressure), and to friction[1]. If, as stated above, the partial derivatives on the left hand side of Eq. 2.2 (local tendency and advection) are combined into the total (or material) derivative $d\vec{v}/dt$, the equation represents the Lagrangian point of

[1] Note that centrifugal force is often neglected or accounted for as correction term in g; cf. Etling (2002).

view: the forces on the right hand side act on a small infinitesimal parcel of air as it moves through the atmosphere.

The continuity equation, expressing the **conservation of mass**, states that the change of mass within a finite volume (that is, the density) equals the mass flux over the volume boundaries, the divergence. In other words, the local change of air density with time equals the mass divergence at the considered location (Etling, 2002, Ch. 8). That is, air density increases if air converges, and decreases if the air diverges:

$$\underbrace{\frac{\partial \rho}{\partial t}}_{\text{local density tendency}} + \underbrace{\vec{v} \cdot \vec{\nabla} \rho}_{\text{advection}} = - \underbrace{\rho \vec{\nabla} \cdot \vec{v}}_{\text{divergence}} \, . \tag{2.3}$$

Conservation of energy is accounted for by the first law of thermodynamics, stating that temperature change can be due to adiabatic processes (the volume or pressure of an air parcel is changed without external heat exchange) and to external heat exchange (Etling, 2002, Chs. 3 and 5):

$$\underbrace{\frac{\partial T}{\partial t}}_{\text{local temperature tendency}} + \underbrace{\vec{v} \cdot \vec{\nabla} T}_{\text{advection}} = \underbrace{\frac{1}{\rho c_p} \frac{dp}{dt}}_{\text{adiabatic processes}} + \underbrace{\frac{\dot{Q}}{c_p}}_{\text{external heat exchange}} \, , \tag{2.4}$$

where T denotes the temperature of the air (K), c_p is the specific heat of dry air at constant pressure ($c_p = 1004$ J kg^{-1} K^{-1}), and $\dot{Q}$ (J s^{-1} kg^{-1}) is the rate of heating per unit mass (due to, for example, radiation, conduction, or latent heat release; cf. Holton, 2004).

In addition to the *prognostic* (that is, including time derivatives) relationships in Eqs. 2.2 to 2.4, the *diagnostic* (no time derivative) **ideal gas law** connects the state of pressure, density and temperature:

$$p = \rho R T. \tag{2.5}$$

Here, $R = 287$ J kg^{-1} K^{-1} is the gas constant for dry air. Furthermore, for a realistic model of the atmosphere the **conservation of water** needs to be accounted for. For example, using specific humidity q (given as the fraction of the mass of water vapour per total mass of humid air) as variable for water vapour, the equation takes the form

$$\underbrace{\frac{\partial q}{\partial t}}_{\text{local humidity tendency}} + \underbrace{\vec{v} \cdot \vec{\nabla} q}_{\text{advection}} = \underbrace{\text{evaporation}}_{\text{source of water vapour}} - \underbrace{\text{condensation}}_{\text{sink of water vapour}} \, . \tag{2.6}$$

Similar equations can be specified for the different phases of water, specifically for cloud liquid and ice water. Also, mass balance equations for additional atmospheric constituents can be integrated into a model. These can include equations for, for instance, aerosols and trace gases (von Storch et al., 1999).

Table 2.1 summarises the equations. To compute a future state of the atmosphere, at minimum Eqs. 2.2 to 2.6 must be solved for the seven state variables u, v, w, T, p, ρ, q (or substitutes thereof; Inness and Dorling, 2013).

For practical purposes in numerical weather prediction, the presented equations are typically transformed into different coordinate systems and/or simplified by approximations. For example, common practice is to use pressure p instead of geometric height z as vertical coordinate (*isobaric* coordinates). Pressure decreases monotonically with height and can hence be used as a substitute for z. For mete-

Table 2.1: Overview of fundamental equations solved in NWP models, following Etling (2002). For the equation of motion (1), the forces on the right hand side include Coriolis force, gravity force, pressure gradient force and frictional forces (cf. Eq. 2.2). The sources and sinks for the continuity equation (2) are the wind field's divergence (cf. Eq. 2.3). For the first law of thermodynamics (3), sources and sinks include adiabatic compression, long and short wave radiation, and phase changes of water (latent heat). The humidity variables q_i (4) include water vapour, liquid water and ice water. Source and sinks are phase changes (condensation, evaporation, freezing). Additional constituents in (5) can include trace gases (e.g. carbon dioxide and ozone), which are generated or depleted by chemical processes.

	local tendency		advection		forces/sources/sinks
1 (motion)	$\partial\vec{v}/\partial t$	$+$	$(\vec{v}\cdot\vec{\nabla})\vec{v}$	$=$	$\vec{F}_i$
2 (continuity)	$\partial\rho/\partial t$	$+$	$\vec{v}\cdot\vec{\nabla}\rho$	$=$	Q_ρ
3 (thermodynamics)	$\partial T/\partial t$	$+$	$\vec{v}\cdot\vec{\nabla}T$	$=$	Q_T
4 (humidity)	$\partial q_i/\partial t$	$+$	$\vec{v}\cdot\vec{\nabla}q_i$	$=$	Q_{q_i}
5 (constituents)	$\partial c_n/\partial t$	$+$	$\vec{v}\cdot\vec{\nabla}c_n$	$=$	Q_{c_n}

orological upper-air observations (for example, from radiosondes or aircraft), pressure is the natural vertical coordinate. Also, usage of isobaric vertical coordinates simplifies a number of terms in the equations of motion, making them useful for numerical weather prediction as well (Etling, 2002, Ch. 9). A common approximation that is applied to transform the equations of motion into isobaric vertical coordinates (Dutton, 1976) is to apply the *hydrostatic approximation*. If the vertical velocity component w of the wind vector $\vec{v}$ vanishes, pressure and height are related through the *hydrostatic equation* (Etling, 2002, Ch. 6):

$$\frac{\partial p}{\partial z} = -g\rho. \tag{2.7}$$

The *hydrostatic approximation* assumes that Eq. 2.7 also holds for small vertical accelerations. This is mostly true for large scale weather phenomena[2]. Models using the hydrostatic approximation hence cannot be used to simulate atmospheric processes for which the approximation breaks down (for instance, small-scale convective processes).

In isobaric coordinates, the pressure gradient force term in Eq. 2.2 vanishes, and the gradient force is instead expressed by the slope of the geopotential Φ in the $\vec{\nabla}\Phi$ term (where the vertical component of the gradient is now defined with respect to pressure). Also, the vertical component in the momentum equation (Eq. 2.2) becomes a diagnostic equation, as the vertical velocity (now in Pa s^{-1}) can be obtained from the continuity equation (which reduces to $\vec{\nabla}_p\cdot\vec{v} = 0$; Etling, 2002). Models that apply the hydrostatic approximation but no further approximation are called *primitive equation models*[3] (Inness and Dorling, 2013).

2.1.2 Numerical solution of the primitive equations

The primitive equations are a coupled system of non-linear partial differential equations (PDEs) that can only be solved numerically. A common approach is to use *finite differences* (e.g. Press et al., 2007) to approximate the derivatives. The dependent variables are specified on a discrete set of grid points in space and time, arranged in a 3D grid that spans the earth either completely (global model) or only

[2] For example, typical vertical velocities in cyclones and anticyclones are on the order of 0.1 ms^{-1}, whereas vertical motion in cumulus and thunderstorm clouds can reach up to 4 and 30 ms^{-1}, respectively (Etling, 2002).

[3] A further advantage of the hydrostatic approximation is that vertical sound waves are eliminated as solutions to the equations of motion; such waves may otherwise trigger numerical instabilities.

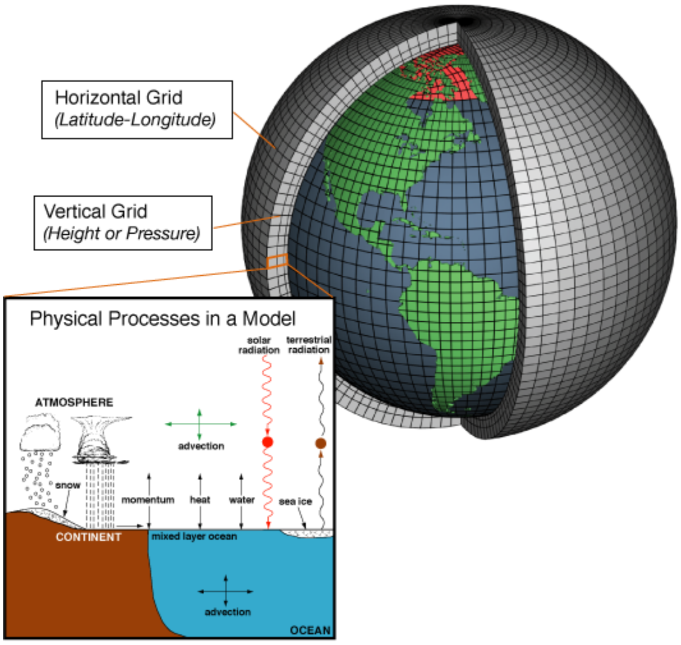


Figure 2.1: Schematic of a global numerical weather prediction model. The equations listed in Table 2.1 are discretised and solved numerically on a three-dimensional grid spanning the earth. Depending on the size of the grid boxes, physical processes on scales smaller than the grid size cannot be simulated explicitly and need to be parametrised. This includes, for example, radiation, turbulence and convection. (Reproduced from National Oceanic and Atmospheric Administration at `http://celebrating200years.noaa.gov`. Figure is in the public domain.)

partly (limited-area model). Figure 2.1 illustrates the approach for a global model. In the example, the equations are expressed in spherical coordinates. The spherical coordinates longitude and latitude are discretised regularly in the horizontal. The vertical coordinate is discretised into several vertical levels, which resemble “shells” surrounding the earth.

At the time of writing, typical grid spacings for global atmospheric models are on the order several tens of kilometres. For example, the UK Met Office Unified Model, a global grid point model (Walters et al., 2014), is run at a grid spacing of 17 km[4]. The physical processes that occur on scales smaller than the grid box size hence cannot be simulated explicitly. They have to be *parametrised*, that is, represented in a simplified way. Stensrud (2007), Warner (2011, Ch. 4) and Coiffier (2012, Ch. 9) list common parametrisations. They include in particular the sources and sinks of radiation (for detailed information on radiation see, for instance, Liou, 2002; Petty, 2006) and water (surface evaporation, cloud processes, precipitation) that appear in Eqs. 2.4 and 2.6, but also small-scale dynamical processes including turbulence and convection (e.g. Stull, 2000; Inness and Dorling, 2013) and boundary layer processes (e.g. Stull, 1988). Improvement of such parametrisations is an active area of research in the atmospheric sciences (e.g. Stensrud, 2007; Jakob, 2010).

An alternative approach to finite differences is the *spectral method* (e.g. Holton, 2004; Press et al., 2007; Coiffier, 2012). In contrast to finite differences, which approximate the *equation* that shall be solved, spectral methods approximate the *solution* (Press et al., 2007, Ch. 20). The spectral method

[4] Current grid spacings obtained from `http://www.metoffice.gov.uk/research/modelling-systems/unified-model/weather-forecasting` (last access April 2015).

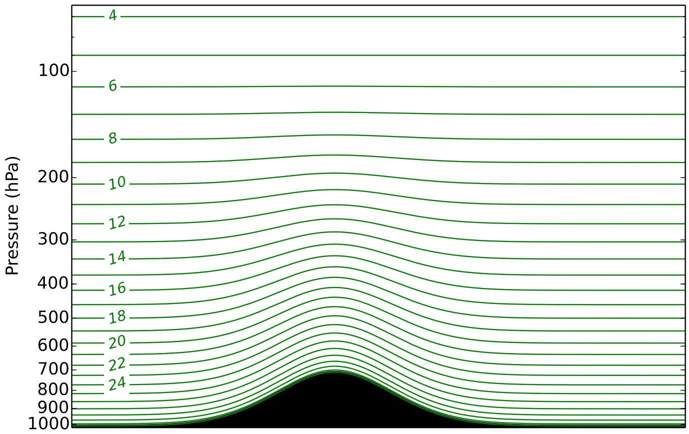


Figure 2.2: The vertical hybrid sigma-pressure coordinate is used in a number of numerical weather prediction models, including the ECMWF model. The coordinate prevents intersections of lower model levels with the surface orography by "following" the terrain. At upper atmospheric levels, the coordinate transitions into constant pressure levels.

represents spatial variations of the variables in terms of finite series of orthogonal basis functions (for example, a Fourier series). For the spherical earth, appropriate basis functions are *spherical harmonics* (which are defined on the sphere and can hence only be applied to global models). The spectral method cannot handle discontinuities in the flow (for example, shock waves), however, works well for smooth solutions (Press et al., 2007, Ch. 20). It is hence in particular suited for atmospheric variables that exhibit wave-like structures, for example, the pressure field with troughs and ridges. One of the advantages of the spectral method is that the derivatives in the equations of motion can be evaluated analytically and do not have to be approximated. They hence are very accurate. For further information (and a list of further advantages), the reader is referred to Riddaway and Hortal (2001).

The length, or *truncation*, of the finite series of spherical harmonics determines the smallest features that can be represented and is thus in some way equivalent to the grid box size. Each retained spherical harmonics mode increases the maximum represented wave number around the globe. For example, when 799 spherical harmonics are retained in the series (commonly written as $T_L 799$ for a truncated series with 799 modes), a wave number of 799 can be represented around the equator. With the earth's radius of approximately 6384 km this corresponds to a minimum wavelength of about 50 km. However, despite advantages, the spectral method requires the model variables to be transformed into grid point space for the application of physical parametrisations. They cannot be formulated in spectral space (Riddaway and Hortal, 2001). Hence, for practical weather forecasting, the spectral method always needs to be combined with a grid point representation. Further information can be found in Riddaway and Hortal (2001).

A problem that occurs both when using pressure and geometric height as vertical coordinate is that some of the lower model levels may intersect with orography. A solution to this problem is to use a "terrain-following" vertical coordinate. A simple form of such a coordinate is the sigma-coordinate proposed by Phillips (1957), which scales the vertical dimension to the interval $(0,1)$: $\sigma = p/p_{sfc}$. Here, p_{sfc} denotes surface pressure below a given grid point. A disadvantage of the sigma-coordinate is that the effects of orography are "felt" by the model up into the highest levels. Current models hence commonly use a hybrid vertical coordinate, for example, a *hybrid sigma-pressure coordinate*. This vertical

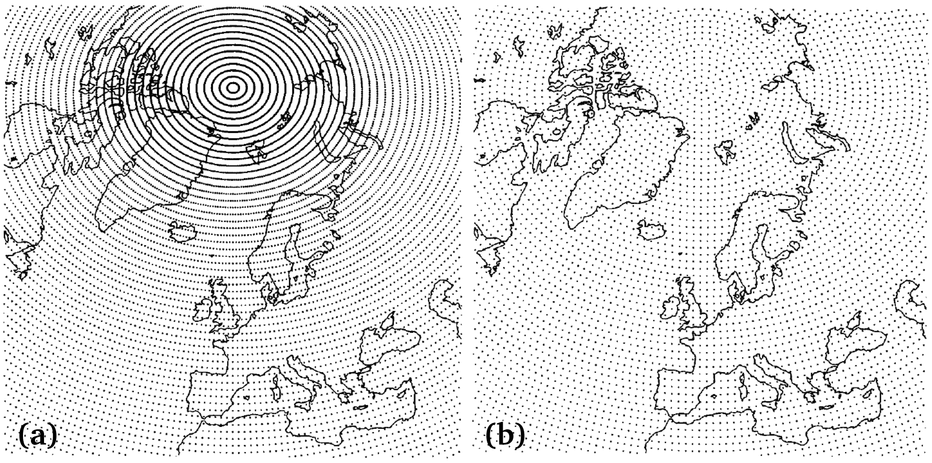


Figure 2.3: The reduced Gaussian grid is used in the ECMWF model for variables that are not computed in spectral space. **(a)** Regular Gaussian grid with a uniform grid spacing in latitude and longitude (as also shown in Fig. 2.1). Note how the grid points converge towards the pole. **(b)** The reduced Gaussian grid keeps the distance between grid points approximately constant by reducing the number of points per parallel towards the pole. (Reprinted from Hortal and Simmons (1991), © American Meteorological Society. Used with permission.)

coordinate transitions into isobaric coordinates at upper levels, as illustrated in Fig. 2.2.

A number of further issues regarding the numerics of a model have to be considered. They include, for example, boundary conditions (surface and top of atmosphere for global models; horizontal boundaries for limited-area models), correct treatment of the earth's poles, and the time integration scheme. Details on these and further issues can be found in, for example, Riddaway and Hortal (2001), Warner (2011), and Coiffier (2012).

2.1.3 Examples of operational NWP models

A number of NWP models, both global and limited-area, are in use at the individual countries' weather agencies. They differ with respect to the domain they cover, to the numerical schemes that are employed (grid point or spectral, grid point topology), to the spatial and temporal resolution, to the implemented physical parametrisations, and to other details. In this thesis I focus on data from the ECMWF Integrated Forecast System (IFS), which is used for all operational atmospheric forecasts made at ECMWF. The characteristics of the numerical model are briefly described here, and put into the context of a few selected other models that are in use:

- The ECMWF IFS[5] is a global hydrostatic model that uses a dual spectral and grid point representation of the atmospheric variables. While horizontal gradients (and a number of further terms) are evaluated in spectral space, the atmospheric variables are transformed to grid point space in order to compute physical parametrisations and to compute the semi-Lagrangian advection terms for time integration. In horizontal grid point space, a reduced Gaussian grid is used (Hortal and Simmons, 1991; Courtier and Naughton, 1994). In contrast to a regular Gaussian grid (as shown in Fig. 2.1) the reduced Gaussian grid approximately keeps the geometric distance between two grid points constant, that is, the number of grid points along a parallel decrease towards the poles (Fig. 2.3). In the vertical, terrain-following hybrid sigma-pressure coordinates are used (Fig. 2.2).

[5] http://old.ecmwf.int/research/ifsdocs/

For time integration, a semi-Lagrangian integration scheme is used (Staniforth and Côté, 1991; ECMWF, 2013b). Further detailed information on the ECMWF model equations can be found in ECMWF (2013b). At the time of writing, the operational ECMWF IFS model is run at T1279L137 resolution. Hence, the minimum resolved wavelength is about 31 km. 137 hybrid model levels are used in the vertical.

- The Global Forecast System[6] (GFS) is the global model of the United States National Weather Service (NWS). The GFS model uses the spectral method as well, at the time of writing it is operationally run at T1534 resolution, corresponding to a grid spacing of approximately 13 km[7]. In the vertical, a hybrid sigma-pressure coordinate system with 64 levels is used. GFS data is publicly available from the NWS.

- The Consortium for Small-scale Modeling model[8] (COSMO; Baldauf et al., 2009, 2011) is a limited-area grid point model. It uses a non-hydrostatic core that is able to resolve convective processes (that is, vertical wind accelerations larger than allowed by the hydrostatic assumption are possible). COSMO can be run at higher spatial resolutions than hydrostatic models and can be used to simulate convective processes. In the horizontal, the model uses a rotated coordinate system. COSMO is used operationally by several European weather agencies. For example, at the time of writing, DWD runs COSMO at 0.0625° (7 km) horizontal grid spacing for an area covering Europe, and at 0.025° (2.8 km) for an area covering Germany[9].

- The ICOsahedral Non-hydrostatic general circulation model[10] (ICON; Zängl et al., 2014) is a very recent global model jointly developed by DWD and the Max Planck Institute for Meteorology in Germany. As stated by its name, the model has a non-hydrostatic core and is thus prepared to be applied to higher horizontal resolutions than hydrostatic models permit. To avoid difficulties at the earth's poles, the model uses dynamical equations that are formulated on an icosahedral triangular grid. In the vertical, a terrain-following variable proposed by Leuenberger et al. (2010) is used.

- The HIgh Resolution Limited Area Model [11] (HIRLAM; Unden et al., 2002) is a limited-area hydrostatic grid point model, also using a terrain-following hybrid sigma-pressure vertical variable. As for COSMO, a rotated horizontal coordinate system is adopted in practice (although other configurations are possible).

- The UK Met Office Unified Model[12] (MetUM; Walters et al., 2014) is a grid point model that can be run at different spatial resolutions in both global and limited-area configurations. The model contains a non-hydrostatic dynamical core whose equations are solved on a regular latitude-longitude grid. In the vertical, a hybrid sigma-height coordinate is used, that is, geometric height is used instead of pressure.

[6] http://www.emc.ncep.noaa.gov/GFS

[7] http://www.emc.ncep.noaa.gov/GFS/impl.php

[8] http://www.cosmo-model.org

[9] http://www.cosmo-model.org/content/tasks/operational

[10] http://www.mpimet.mpg.de/en/science/models/icon.html

[11] http://www.hirlam.org

[12] http://www.metoffice.gov.uk/research/modelling-systems/unified-model

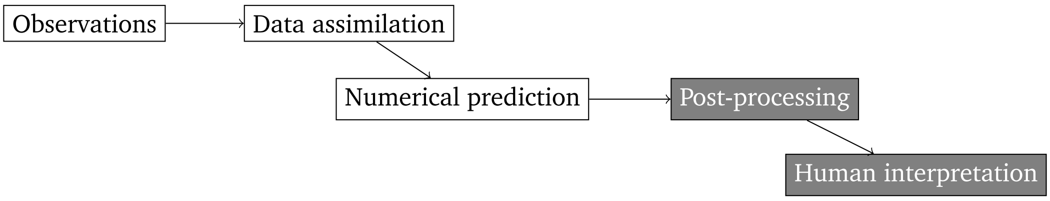


Figure 2.4: Schematic of the weather forecasting process. Atmospheric measurements are collected by a global observation network. They are assimilated into an NWP model, which is integrated in time to compute the forecast. The forecast data are output for further post-processing, including automated statistical evaluation of the forecast data, and visualization to enable meteorologists to explore and interpret the data. This thesis is concerned with visualization as a part of post-processing and human interpretation (grey boxes).

- The Weather Research & Forecasting Model[13] (WRF; Skamarock et al., 2008) is a limited-area community model developed in the United States that targets both operational forecasting and research. Its main distinction is its modular flexibility, allowing a variety of physical parametrisations, non-hydrostatic and hydrostatic configurations and variable resolution[14]. WRF has been adopted by meteorological agencies and research institutions around the world. A brief description of WRF can be found in Coiffier (2012).

There are other atmospheric models than NWP models. *Climate models* are global general circulation models that are very similar to global NWP models but run at coarser resolutions to facilitate longer integration times (von Storch et al., 1999). For example, the ECHAM model[15] (Stevens et al., 2013) is derived from the ECMWF IFS and uses the same dynamical core. The above mentioned MetUM model can also be run in a climate configuration (Walters et al., 2014). *Large-eddy simulation models* (LES), on the other hand, simulate small scale atmospheric phenomena. They are usually defined in a Cartesian coordinate frame, using finite differences for all variables. LES models are able to explicitly simulate processes such as convection and are often used in atmospheric research to develop new parametrisations for use in NWP and climate models. Examples include the EULAG model[16] (Prusa et al., 2008) and the PALM model[17] (Maronga et al., 2015).

2.2 The operational forecasting process

In addition to running a numerical model, a number of further steps are required to compute a weather forecast. To use a numerical model to compute a future state of the atmosphere, the model needs to be started from an initial state. These initial conditions are derived from a global network of observations. The observations, however, are not directly available on the model grid points, and hence a process called *data assimilation* is applied to derive representative initial conditions. Figure 2.4 illustrates the major steps of the weather forecast process. They are briefly described below. Detailed information can be found in Inness and Dorling (2013).

Observations Atmospheric observations are collected by a global network of observing platforms in the World Meteorological Society (WMO)'s Global Observation System (GOS) and shared through the

[13] http://www.wrf-model.org
[14] http://www2.mmm.ucar.edu/wrf/users/model.html
[15] http://www.mpimet.mpg.de/en/science/models/echam.html
[16] http://www2.mmm.ucar.edu/eulag
[17] http://palm.muk.uni-hannover.de

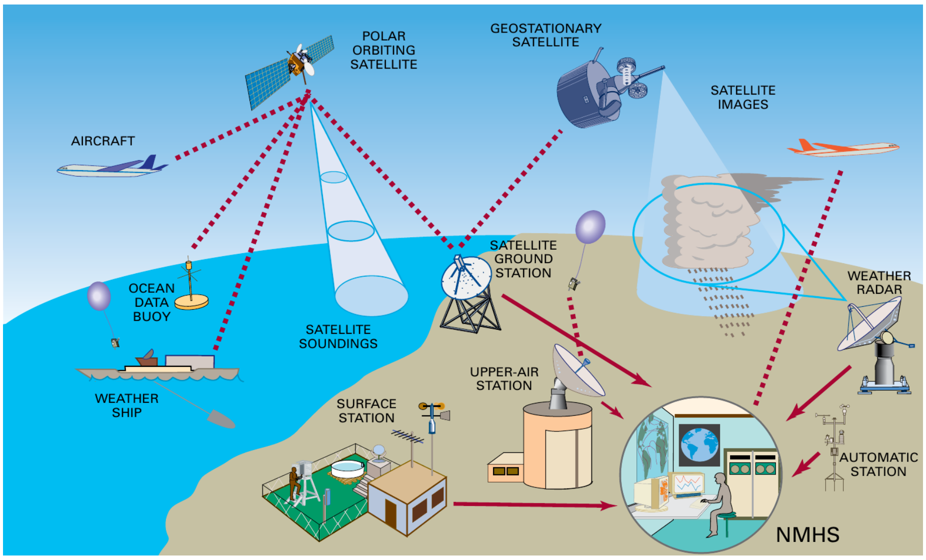


Figure 2.5: Global observations used to compute the initial conditions for an NWP model are made by in-situ and remote sensing platforms. The spatial and temporal coverage of the individual observations is very diverse (cf. Fig. 2.6), as is the measurement uncertainty inherent in the collected data. (Reprinted from Branski (2010), © World Meteorological Organization. Used with permission.)

Global Telecommunication System (GTS) (Branski, 2010). Figure 2.5 illustrates various types of observational data that are used to create the initial conditions for NWP models. The observations come from a variety of measurements, both in-situ (surface stations, radiosonde balloons, ship, buoy and aircraft data) and remote sensing (satellites, radar) (Branski, 2010; Persson and Andersson, 2013; Inness and Dorling, 2013).

It is important to note that observations are not available from all sources at the same spatial and temporal density all over the earth. ECMWF operationally publishes the observation sources that have been considered for its latest forecasts[18]. Figure 2.6 shows examples. Surface stations are abundant in highly populated areas, however, lack in sparsely populated areas and in particular over the oceans. The oceans are partly covered by buoys, which, however, do not exist at the same spatial density as surface stations. In-situ upper-air measurements are taken by radiosonde balloons and also by aircraft. As can be seen, radiosonde soundings are mainly available over North America, Europe and south-east Asia, however, the number of measurements is about an order of magnitude smaller than for surface stations. There is a large void in radiosonde soundings over the remaining parts of the world. The same applies to aircraft measurements, which are mainly available along the world's major flight routes. Also, the latter are restricted to typical flight altitudes, except for the vicinity of airports. Remote sensing data from satellites are also not available globally. Geostationary satellites observe the earth from above the equator and cannot properly see the poles. They do, however, provide observations that are continuous in time. Polar-orbiting satellites orbit around the entire earth; however, at a fixed time they only see a small portion. Satellites also mainly provide 2D observations. More details are available in Inness and Dorling (2013).

Data assimilation The process of creating initial conditions for a numerical forecast model from the

[18] `http://old.ecmwf.int/products/forecasts/d/charts/monitoring/coverage`

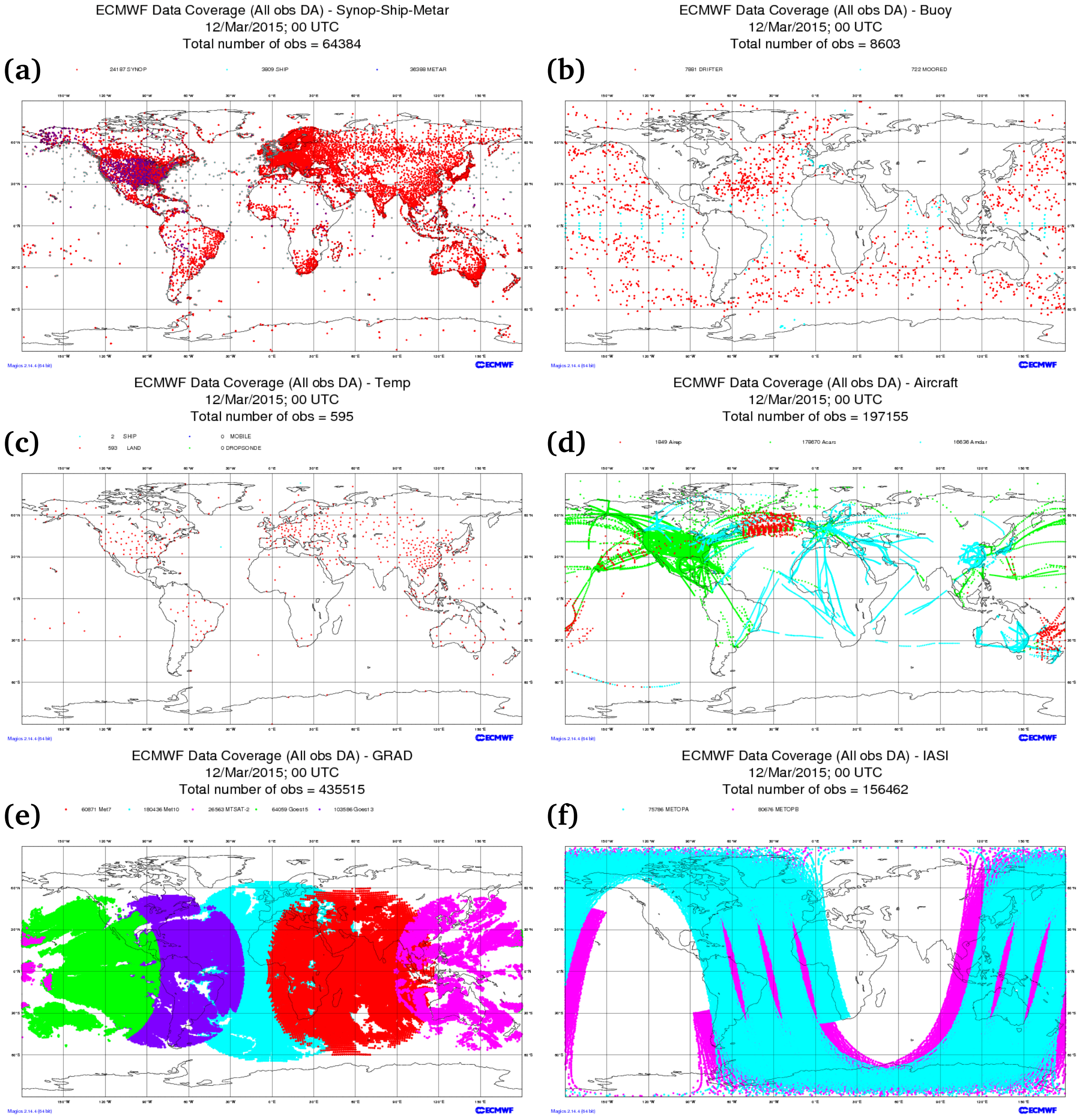

ECMWF Data Coverage (All obs DA) - Synop-Ship-Metar
12/Mar/2015; 00 UTC
Total number of obs = 64384
(a)
24187 SYNOP
3809 SHIP
36388 METAR
ECMWF Data Coverage (All obs DA) - Buoy
12/Mar/2015; 00 UTC
Total number of obs = 8603
(b)
7881 DRIFTER
722 MOORED
ECMWF Data Coverage (All obs DA) - Temp
12/Mar/2015; 00 UTC
Total number of obs = 595
(c)
2 SHIP
593 LAND
0 MOBILE
0 DROPSONDE
ECMWF Data Coverage (All obs DA) - Aircraft
12/Mar/2015; 00 UTC
Total number of obs = 197155
(d)
1849 Airep
179670 Acars
15636 Amdar
ECMWF Data Coverage (All obs DA) - GRAD
12/Mar/2015; 00 UTC
Total number of obs = 435515
(e)
60871 Met7
180436 Met10
26563 MTSAT-2
64059 Goes15
103586 Goes13
ECMWF Data Coverage (All obs DA) - IASI
12/Mar/2015; 00 UTC
Total number of obs = 156462
(f)
75786 METOPA
80676 METOPB
ECMWF

Figure 2.6: Examples of the data coverage of observation platforms used in the data assimilation process at ECMWF. Shown are selected observations that were used for the forecast initialised at 00:00 UTC 12 March 2015. Note that none of the observation methods covers the entire planet. **(a)** Surface stations and ship observations. **(b)** Buoys. **(c)** Vertical radiosonde soundings (*temps*). Note the data void in particular over the oceans. **(d)** Observations taken by aircraft on regular scheduled flights. **(e)** Coverage of geostationary satellites. **(f)** Coverage of the Infrared Atmospheric Sounding Interferometer (IASI) onboard the polar orbiting Metop satellite. (Reproduced by permission of ECMWF.)

available observations is called data assimilation. Ideally, in-situ observations of all simulated variables would be available at each model grid point at an instantaneous time to integrate the model forward in time from an observed state of the atmosphere. Since this is not feasible, the available observations need to be used to construct an initial state of the model. Here, great care needs to be taken to ensure that the initial state is consistent with the model dynamics. For example, small measurement errors in wind speed or pressure can cause the wind speed to be not in balance with the pressure; as a consequence, high frequency gravity waves (to bring the fields back into balance) can develop in the model. Such waves can contaminate a forecast and render it useless (Inness and Dorling, 2013).

To counteract this problem, data assimilation schemes use a short-range forecast field from a previous forecast run and modify this field to best match the observations. In practice, *variational data assimilation* schemes are frequently used (Inness and Dorling, 2013). In short, they are designed to (1) minimise the distance between the original short-range forecast and the modified initial condition field (to minimise the chance that the new field is physically inconsistent), (2) minimise the distance between observations that are received in the time window covered by the assimilation process (note that the observations are not all taken simultaneously), and (3) minimise the presence of high-frequency gravity waves.

Detailed introductions to data assimilation methods are provided by Warner (2011) and Coiffier (2012). As an example, ECMWF uses a four-dimensional data assimilation scheme (4D-Var) (Andersson and Thépaut, 2008; Persson and Andersson, 2013; ECMWF, 2013a) that incorporates observations into model states over a time window of (at the time of writing) 12 hours. The ECMWF method weighs observations according to the assumed accuracy of the observations and of the short-range forecasts. If an observation is used to correct a specific model variable, this also induces changes to other variables (for example, a correction in the modelled moisture field may also have impact on wind and temperature fields; Persson and Andersson, 2013). Details can be found in the ECMWF IFS documentation and in references listed therein (ECMWF, 2013a).

Numerical prediction Running the numerical model to compute future states of the atmosphere requires supercomputers. To provide an order of magnitude: at the time of writing (since fall 2014), ECMWF's forecasts are computed on a Cray XC30 system featuring two compute clusters, each with a total of 84 120 Intel Xeon processing cores, running at 2.7 GHz clock frequency each[19]. The cores are distributed in 3505 compute nodes, each equipped with 64 GiB of memory. Exact timings indicating how long the ECMWF model actually runs are not available, however, are on the order of a few hours[20].

Post-processing Raw simulation data output by an NWP model is usually post-processed before interpreted by a meteorologist. The interpolation of the forecast data fields to a different grid and subsequent visualization can be considered as post-processing, however, there are different types commonly in use. For example, statistical post-processing with *model output statistics* (MOS) techniques tries to correct predicted variables (for instance, the 2m temperature in a particular city) by applying a statistical model trained on the basis of past observations and forecasts (e.g. Wilks, 2011). Such techniques can increase the accuracy of the forecast, however, require a suitable training dataset. As another example, further numerical models including air dispersion models (e.g. FLEXPART; Stohl et al., 2005) or Lagrangian particle trajectory models (e.g. LAGRANTO; Sprenger and Wernli, 2015) can be run on the forecast wind fields. Inness and Dorling (2013), Warner (2011, Ch. 13) and Coiffier (2012, Ch. 10) list further examples of post-processing.

Human interpretation Despite the wealth of information generated by modern NWP systems, human forecasters still play a crucial role in generating the final form of a forecast that is communicated to

[19] http://www.ecmwf.int/en/computing/our-facilities/supercomputer

[20] Compare to the ECMWF dissemination schedule available at http://old.ecmwf.int/services/dissemination/3.1/Meteorological_Bulletin_M3_1_22.html (last access April 2015).

customers. Inness and Dorling (2013) describe typical tasks handled by forecasters and challenges they are confronted with.

A human forecaster explores and interprets the data output by the NWP models. This includes obtaining an understanding for the future development of the synoptic situation and the identification of weather phenomena that are of particular interest to specific customer applications or the general public. While a time series of point forecasts (for example, for temperature and precipitation at a particular location) could be automatically generated from the model output, the forecaster's role is to estimate the reliability and uncertainty of the model output. For instance, in an operational environment, forecasters compare the first few hours of the forecast with the actual weather conditions. In situations in which rapid atmospheric changes occur the model may deviate early from reality, and forecasters can consider this information when further interpreting the simulation data. Forecasters also compare the output of different models and apply knowledge about model characteristics (for example, systematic errors and biases). This can also include knowledge about local characteristics. For instance, in complex terrain models often are not capable of representing small scale effects.

As Inness and Dorling (2013) put forward, one of the challenges that today's meteorologists face is "simply the volume of NWP output available to forecasters" that can be explored and interpreted. Contribution to this challenge is part of this thesis.

The visualization methods that this thesis is concerned with can be classified into the last two items, post-processing and human interpretation.

2.3 Ensemble prediction – forecasting the uncertainty

Forecasting as described in the previous section is referred to as *deterministic* forecasting. The same initial conditions, integrated in time with the same numerical model, result deterministically in the same forecast. Common practice of the world's weather agencies is, as put by Gneiting and Raftery (2005): "For a given set of 'best' input data, one 'best' weather prediction is generated." However, there are obvious sources of uncertainty in the forecast methodology described in the previous sections. As outlined in Sect. 2.2, the atmosphere can never be observed at full spatial coverage. A data void occurs in particular over the oceans and at upper atmospheric levels. Also, the available observations are always associated with measurement uncertainty. Hence, the initial conditions necessarily carry some uncertainty and NWP models will always start the forecast integration from a state at least slightly different from the real atmosphere. Also, the model formulation itself, including its discretisation and physical parametrisations, is only an approximation to the real world. Since the model is also used in the data assimilation process, the model uncertainty also contributes to the uncertainty in the initial conditions. Palmer (2006) notes parameters external to the atmosphere as a third source, including, for instance, volcano eruptions that emit large amounts of aerosols into the atmosphere. These, however, are neglected here.

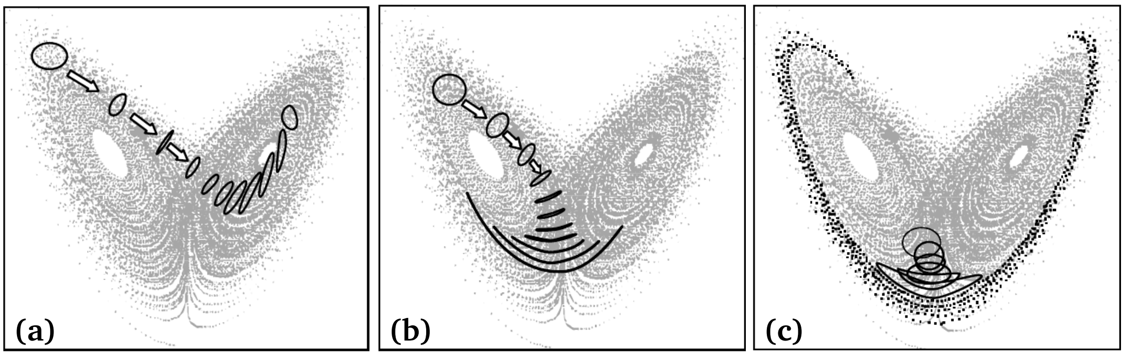


Figure 2.7: Illustration of the effect that small changes in the initial conditions can have on a forecast. Assume that each point in the figures corresponds to a valid state of the atmosphere. The initial ellipses each represent a distribution of possible states given a particular set of observations and their uncertainty. Depending on the atmospheric conditions at the time of observation (corresponding to where the initial ellipse is located), small changes in the observations (corresponding to the points inside the ellipse) may **(a)** lead to forecasts being similar to each other, but may **(b and c)** also lead to heavily diverging forecasts. Compare to Fig. 2.8. (Reprinted from Palmer (2006), © Cambridge University Press. Used with permission.)

2.3.1 Predictability

Due to the *chaotic* nature of the equations of motion, uncertainty in initial conditions and model formulation has severe impact on the computed forecast. In the 1960s, Edward Lorenz discovered that the forecasts made by simple numerical atmospheric models he was using were very different if the initial conditions were slightly modified (Lorenz, 1963, 1965)[21]. In subsequent decades, chaos theory was developed (Lorenz, 1993). In essence, it states that small changes in the initial conditions of a dynamic, non-linear system such as the atmosphere can lead to fundamentally different solutions. Figure 2.7 illustrates the effect. The figure, produced by Palmer (2006) using the Lorenz (1963) model, shows that for slightly different initial conditions within an initial ellipse the forecast values after a finite time integration can still be similar (a), but can also heavily diverge (b and c). For numerical weather prediction, this means that since it will never be possible to observe the state of the entire atmosphere at an instantaneous time down to the smallest scales with no measurement uncertainty attached, even with a perfect numerical model of the atmosphere it will never be possible to endlessly predict future atmospheric evolution. Similarly, model imperfections, even just small numerical round-off errors, may also cause the model forecast to diverge from the "real" atmospheric evolution.

Hence, a single deterministic forecast of an atmospheric state computed from the 'best' initial conditions is, as Wilks (2011) puts it, "at best only one member of an essentially infinite collection of [atmospheric states] that could plausibly occur". It would correspond to the trajectory of a single point in Fig. 2.7. Depending on the current atmospheric situation (corresponding to where the initial ellipse is located in the figure), the deterministic forecast may be good (*skilful*, as in Fig. 2.7a), or just one random representation of many that are possible (Fig. 2.7b and c).

Uncertainty is hence an inherent component of numerical weather prediction. Limits of predictability are estimated to be on the order of about 10 days lead time (Lorenz, 1982; Froude et al., 2013). Thus, if the time interval for skilful deterministic forecasts is limited and furthermore this limit depends on the atmospheric state, it is at least necessary to estimate a forecast's uncertainty. As stated by Tennekes et al.

[21] An effect that due to a 1972 talk by Lorenz entitled "Does the flap of a butterfly's wings in Brazil set off a tornado in Texas?" (Lorenz, 1993, Appendix 1) became known as the *butterfly effect*.

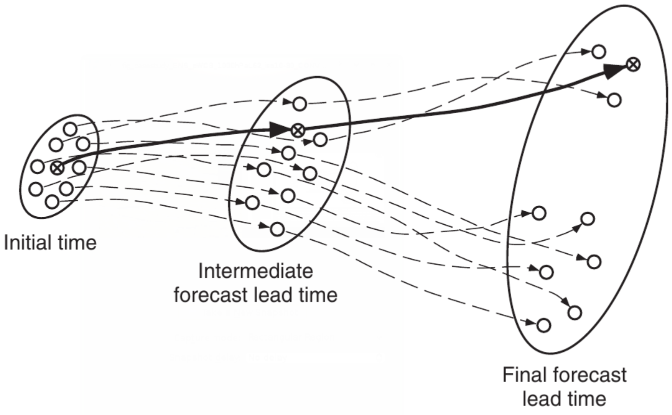


Figure 2.8: Ensemble forecast methods estimate the distribution of possible future atmospheric states by computing forecasts for a finite sample of initial states. Shown is a set (the ensemble) of forecast trajectories in an idealised 2D phase space of atmospheric states (similar to Fig. 2.7). The solid line represents the forecast computed from the "best" initial conditions obtained from the available observations. The dashed lines represent forecasts computed from alternative initial states that, given the uncertainty in the observations, are also possible. Note how in the depicted example, the forecasts at final lead time cluster in two groups (compare to Fig. 2.11). In addition to the illustrated initial condition uncertainty, current ensemble prediction systems also account for uncertainty in model formulation. (Reprinted from Wilks (2011), © 2011, with permission from Elsevier.)

(1986): "no forecast is complete without a forecast of forecast skill". Having uncertainty "predictions" available, forecasters can judge how uncertain the predictions are, and, if multiple forecast scenarios may have a chance to occur.

A way to achieve such estimates of uncertainty is to move from the deterministic forecast approach to a probabilistic approach. If the governing equations of the NWP model operate not on a single set of initial conditions but on a probability distribution, in theory the model should forecast a probability distribution of future atmospheric states. Edward Epstein called this approach *stochastic dynamic prediction* (Epstein, 1969; Lewis, 2014). In terms of Fig. 2.7, a probabilistic NWP model would predict the time evolution of the entire ellipse (corresponding to the probability distribution of initial model states).

For simple, low-dimensional models, this can be achieved by means of the Liouville equation (e.g. Ehrendorfer, 1997, 2006), a "continuity equation" for probability (Wilks, 2011). However, for practical weather forecasting the dimensionality of the phase space of the NWP model is too large to allow the direct solution. The numerical solution of Liouville's equation is impractical even for dimensions N as low as 100 (Leutbecher and Palmer, 2008). For atmospheric models, N is (number of grid points) × (number of variables), and for current operational models on the order of 10^6 to 10^9.

For detailed treatise of predictability, the reader is referred to, for example, Ehrendorfer (1997) and Palmer and Hagedorn (2006).

2.3.2 Ensemble methods

Ensemble methods have evolved as a computationally tractable approach to estimate forecast uncertainty. They approximate stochastic dynamic prediction by a Monte Carlo approach (Leith, 1974) and have become feasible in operational weather forecasting since the early 1990s. Lewis (2005) provides a review of the historical development of ensemble forecasting, overview articles are provided by Gneiting and Raftery (2005) and Leutbecher and Palmer (2008). With respect to initial condition uncertainty, the basic idea of ensemble forecasting is illustrated in Fig. 2.8. A finite sample of initial atmospheric states, the *ensemble*, is drawn from the initial condition distribution. These states are integrated in time by the NWP model to yield a full deterministic forecast each. The collective trajectories of all selected ensemble members in phase space are interpreted to approximate the probability distribution of future atmospheric states. Similarly, members can be chosen to represent the model error, which affects the integration of the forecast trajectories.

Ideally, one would randomly sample a large number of initial states and model realisations and produce the corresponding forecasts (Wilks, 2011). However, Palmer (2006) and Wilks (2011) point out reasons why this approach is not adopted in practice: (1) Since every random sample requires a full run of the NWP model, computation of a large number of forecasts is not feasible even on current supercomputers. (2) The PDF of the initial conditions depends on several factors including instrument uncertainty and the data assimilation process and can in practice only be approximated itself.

Hence, strategies have been adopted that aim at selecting a small, computationally tractable, number of ensemble members (on the order of 20 to 50) that are likely to best represent the forecast PDF and that thus provide the best estimate of the uncertainty. As Palmer (2006) points out, it is important to select these members conservatively in the sense that those deviations from the 'best' initial state and/or model should be selected that are likely to have significant impact on the forecast: if severe weather occurs and none of the ensemble members has forecast the severe weather event, the usefulness of the ensemble is limited and its credibility compromised.

In this respect, different operational centres have adopted different strategies to select initial members. Care needs to be taken to ensure that the selected initial conditions are internally consistent with the model equations (for example, a perturbation of the wind field needs to be balanced by corresponding perturbations in pressure and temperature). Common approaches to choosing initial ensemble members include (Wilks, 2011):

- The *breeding method* (Toth and Kalnay, 1993, 1997) is used in the United States by the National Centers for Environmental Prediction (NCEP). The method uses short runs of the forecast model to compare the unperturbed forecast with randomly perturbed forecasts. Differences are rescaled to match the magnitude of the initial perturbations, added to the forecast state and the procedure repeated. After a number of iterations, those structures that lead to fastest error growth are retained. Inness and Dorling (2013) point out that an advantage of the method is that it is computationally cheap, since subsequent forecast runs of the ensemble can be used to generate the perturbations.

- The generation of initial perturbations at ECMWF includes the usage of *singular vectors* (SVs; Buizza, 1994; Buizza and Palmer, 1995; Buizza et al., 2010). Here, a linearised version of the forecast model is used to identify those atmospheric patterns that lead to fastest error growth. These patterns are called SVs. Inness and Dorling (2013) note that in the mid-latitudes, SVs are often perturbations on upper level troughs and jet streams. In the tropics, shortwave features including tropical storms are frequently captured.

- The Meteorological Service of Canada uses an *Ensemble Kalman filter* (EnKF; Houtekamer and Mitchell, 2005) approach. It follows a Bayesian approach and uses ensemble members from the previous forecast run to define a Gaussian prior distribution. A Gaussian posterior distribution is generated by incorporating observation data for which a known variance is assumed (Wilks, 2011).

- Recently, ECMWF included the usage of an *ensemble of data assimilation* (EDA) into their system (Buizza et al., 2008; Isaksen et al., 2010; Buizza et al., 2010). The EDA represents uncertainties that have been growing during data assimilation by (a) perturbing observations and (b) perturbing parametrisations to simulate model error in the assimilation process.

Through the data assimilation process, model error already becomes a part of initial condition error. Also, as noted above, model error affects the integration of the forecast in time. For instance, it may cause loss of ensemble spread. Palmer (2006) points out that there is no theory to estimate the uncertainty emanating from the numerical approximations. Also, the uncertainty of subgrid parametrisations is often unknown. A number of ensemble approaches exist to account for model error (Palmer, 2006) :

- A *multimodel* ensemble consists of different NWP models.
- A *multiparameterisation* ensemble uses a single model framework with the same dynamical core but different parametrisation schemes.
- A *multiparameter* ensemble uses the same model for all ensemble members, but with stochastic perturbations applied to the parametrisation parameters.

2.3.3 Examples of ensemble prediction systems

Similar to the deterministic forecast systems described in Sect. 2.1, both global and limited-area ensemble predictions systems are in operational use. I provide a few selected examples:

- The ECMWF ensemble prediction system ENS is a 51-member ensemble consisting of one control run and 50 perturbed members of the IFS model, run at half the horizontal resolution of the deterministic forecast (that is, at T639L62). The ensemble is run twice daily at 00:00 and 12:00 UTC. The system is described in more detail in Sect. 2.4, as its data are used in this thesis.
- NCEP operates the Global Ensemble Forecast System (GEFS)[22], including one control forecast and 20 perturbed members based on the GFS model. At the time of writing, the ensemble configuration is updated to operate at T574L64 resolution for the first 192 hours, and at T384L64 up to 384 hours (16 days) lead time. The generation of initial conditions is planned to be changed from the breeding vector technique to an ensemble Kalman filter[23]. The ensemble is run four times daily at 00:00, 06:00, 12:00 and 18:00 UTC. Model uncertainty is accounted for with stochastic parametrisation perturbations.
- The Global Ensemble Prediction System (GEPS; Gagnon et al., 2014) is a 20 member ensemble of Environment Canada's meteorological service. It uses the Global Environmental Multiscale model (GEM; Côté et al., 1998), a global grid point model run at 50 km grid spacing with 40 vertical levels[24]. Model uncertainty is accounted for with different parametrisations and/or parametrisation

[22] `http://www.emc.ncep.noaa.gov/index.php?branch=GEFS`

[23] Description of model configuration (`http://www.emc.ncep.noaa.gov/gmb/yzhu/imp/i201412/NPSR-2014-Zhu.pdf`) available at `http://www.emc.ncep.noaa.gov/gmb/yzhu/html/imp/201412_imp.html` (last access April 2015).

[24] `http://collaboration.cmc.ec.gc.ca/cmc/cmoi/product_guide/docs/changes_e.html`

parameters in each member (Gagnon et al., 2014). Data assimilation is performed via an ensemble Kalman filter; each ensemble member is initialised with a different analysis (Houtekamer et al., 2014)[25]. The system is run twice daily at 00:00 and 12:00 UTC and integrated for 384 hours (16 days) lead time.

- The COSMO Limited-Area Ensemble Prediction System (COSMO-LEPS; Tibaldi et al., 2006; Montani et al., 2011) has been developed in corporation by a number of European states, including Greece, Italy, Poland, Romania, Switzerland and Germany (Schumann, 2009). The initial and boundary conditions for the 16 members of the ensemble are obtained from the ECMWF ENS with a clustering approach (Montani et al., 2011). Schumann (2009) notes that the strengths of the system include the prediction of severe precipitation in particular in the Alps and lower European mountain ranges, as well as of severe storms. Weaknesses include deep convection and extreme temperatures. COSMO-LEPS operates for a domain covering central Europe with a horizontal grid spacing of 10 km and 40 levels in the vertical, and is integrated once daily at 12:00 UTC for 132 hours lead time[26].
- COSMO-DE-EPS (Theis and Gebhardt, 2011) is the first operational ensemble prediction system on a convection-permitting resolution in Germany. The system has been run operationally by DWD since 2012 and uses 20 members at a horizontal grid spacing of 2.8 km. The members are driven by boundary conditions from COSMO integrations run at 7 km horizontal grid spacing, which in turn are driven by boundary conditions from four global models (Theis and Gebhardt, 2011). For each of the four boundary conditions, five different model configurations are run with different parametrisation parameters (that is, a kind of multiparameterisation ensemble). The system is designed for short-range forecasts and run for a domain covering Germany every three hours (00:00, 03:00, 06:00, 09:00, ... UTC) for 21 hours lead time. Details are provided by Theis et al. (2012).

Other meteorological agencies, including the UK MetOffice and the Japan Meteorological Agency, also operate ensemble systems.

2.4 The ECMWF ensemble prediction system

This section describes the ECMWF forecast system in more detail, as the visualization methods presented in this thesis are based upon its data. As described in Sects. 2.1.3 and 2.3.3, ECMWF operates the global ensemble weather prediction system ENS and the high-resolution deterministic integrated forecast system IFS. Both deterministic and ensemble forecasts are computed with the hydrostatic, spectral ECMWF model briefly described in Sect. 2.1.3. The ENS is described in the literature by, for example, Molteni et al. (1996) and Buizza (2006). It has been run operationally since 1992. The history of the ECMWF models is described in detail by Woods (2005). Buizza (2006) list major updates until 2004. Buizza et al. (2005) compare the ECMWF ENS to the global ensemble systems of NCEP and MSC.

Key characteristics of the ENS include[27]:

[25] http://weather.gc.ca/ensemble/verifs/model_e.html

[26] Compare to http://www.cosmo-model.org/content/tasks/operational/leps/default.htm (last access April 2015).

[27] Compare to the ECMWF website at http://www.ecmwf.int/en/forecasts/documentation-and-support/key-characteristics-forecasting-system (last access April 2015).

Members ECMWF operates 50 perturbed members and one unperturbed control forecast in the ENS. The control forecast uses the "best" initial conditions that are also used for the deterministic forecast, however, the ENS is run at a lower resolution than the deterministic forecast.

Initial conditions uncertainty The usage of singular vectors (cf. Sect. 2.3.2) to account for initial condition uncertainty already distinguished previous versions of the ENS from the ensemble systems operated by, for instance, NCEP and MSC (Buizza, 2006). Since 2010, a combined system of SVs and EDA has been used (Isaksen et al., 2010; Buizza et al., 2010). The EDA-based perturbations represent uncertainty originating from the data assimilation cycles, the SVs are used to find perturbations that contribute most to forecast uncertainty during the first 48 hours of the forecast (Buizza et al., 2010).

Model uncertainty Model error is accounted for with the *Stochastic Perturbed Parametrization Tendency* (SPPT) and the *Stochastic Kinetic Energy Backscatter* (SKEB) schemes (Shutts et al., 2011). SPPT perturbs tendency terms in the model's physical parametrisations by a random global field with prescribed statistical properties. SKEB accounts for deficiencies in the dynamical component of the model. Here, random vorticity perturbations are added to the flow field in order to account for local energy dissipation in the simulation. Details and further references for both methods can be found in Shutts et al. (2011).

Horizontal resolution During TNF and at the time of writing, the ENS uses a spectral truncation of T639 in the horizontal, corresponding to a regular latitude-longitude grid of approximately 0.28° by 0.28° (approx. 32km; Miller et al., 2010). The deterministic forecast is run at T1279 resolution (0.15° by 0.15° or 16 km).

Vertical resolution In the vertical, the ECMWF model uses the terrain-following hybrid sigma-pressure coordinate discussed in Sect. 2.1.2 and illustrated in Fig. 2.2. The vertical pressure coordinate p_k of a grid point at level k is defined by a set of fixed coefficients a_k and b_k and the surface pressure p_{sfc} below the grid point (Untch and Hortal, 2004):

$$p_k = a_k + b_k * p_{sfc}. \tag{2.8}$$

With increasing altitude the influence of p_{sfc} decreases. During TNF, the operational ensemble forecast was available with 62 levels (increased by the time of writing to 91 levels; compare to 91 levels for the deterministic forecast during TNF, increased by the time of writing to 137 levels). At this resolution, levels are constant in pressure above approximately 64 hPa (70 hPa)[28].

Temporal range and resolution Forecasts for both the ensemble and deterministic forecasts are available twice daily (starting at 00:00 and 12:00 UTC) at a time step of three hours up to 144 h (6 days) forecast lead time and six hours up to 240 h (10 days) forecast lead time (Persson and Andersson, 2013). The ensemble forecast is further integrated up to 15 days at a coarser T319 resolution, twice weekly up to 32 days (Miller et al., 2010; Persson and Andersson, 2013).

Forecast performance Persson and Andersson (2013) note some characteristics of forecast performance, including dependence on scale and possible forecast "jumpiness" between subsequent

[28] Compare to `http://www.ecmwf.int/products/data/technical/model_levels/` (last access April 2015).

runs. For instance, the forecast performance, or *skill*, of the predictions is dependent on the scale of the atmospheric system that is considered. As a rule of thumb, the larger the scale, the longer its timescale and the higher its predictability (Persson and Andersson, 2013). The skill also depends on the current weather regime, as Ferranti et al. (2014) describe.

Froude et al. (2013) state the predictability of the ensemble mean 500 hPa geopotential height field to be on the order of 10 days lead time. After that time, the ensemble mean converges towards climatology (that is, the forecast carries no further information than simply climatological values).

Every year, ECMWF publishes a technical report on the performance of the forecast system[29]. At the time of writing, the latest performance report dates to December 2014 (Haiden et al., 2014).

Model output ECMWF model output is stored in GRIB[30] (GRIdded Binary) format and archived in the ECMWF Meteorological Archive and Retrieval System (MARS). The model version current during TNF produced a data volume on the order of 1.3 TB per forecast run (about 25 GB per time step for all ensemble members globally stored as spherical harmonics coefficients).

Data availability and retrieval Data is available from MARS, either as global fields in spherical harmonics or reduced Gaussian grid representation, or interpolated in the horizontal to a regular latitude-longitude grid. In the vertical, the data is available on either a set of pre-defined pressure levels (PL), or, higher resolved and thus better suited for 3D visualization, on the native model grid levels (ML).

2.5 Ensemble products in operational forecasting

A number of maps and diagrams are in use at operational meteorological centres to visualize ensemble NWP output. The books by Wilks (2011) and Inness and Dorling (2013) contain overviews of general techniques that are in use. Legg et al. (2002) and Mylne (2006) describe ensemble products that are in use at the UK Met Office. Tennant et al. (2007) describe usage of NCEP ensemble products at the South African weather service, and Novak et al. (2008) survey availability, use and demands concerning ensemble products at the US National Weather Service. Schumann (2009) and Denhard (2011) review the usage of ensemble products at DWD, Schumann (2009) discusses which ensemble prediction system (cf. Sect. 2.3.3) is suitable for which atmospheric conditions. Persson and Andersson (2013) provide guidance for the ECMWF ensemble products. In the following, I survey commonly used techniques. Note that all techniques described in the above references (and presented in this section) solely rely on 2D visualization methods.

A direct way to visualize ensemble output is to plot data from the ensemble members simultaneously side-by-side in individual displays. Due to the large display space requirement, such plots are usually scaled to the size of a postage stamp each and are referred to as *stamp maps*. As an example, Fig. 2.9 shows the ECMWF ENS forecast for storm "Niklas" on 31 March 2015, as forecast by the ensemble members initialised 168 hours prior to the event. The plots show contour lines of mean sea level pressure and 850 hPa temperature. In stamp maps, individual details are not discernible. However, differences

[29] `http://www.ecmwf.int/en/forecasts/tools-and-guidance/quality-our-forecasts`

[30] `https://software.ecmwf.int/wiki/display/GRIB/Home`

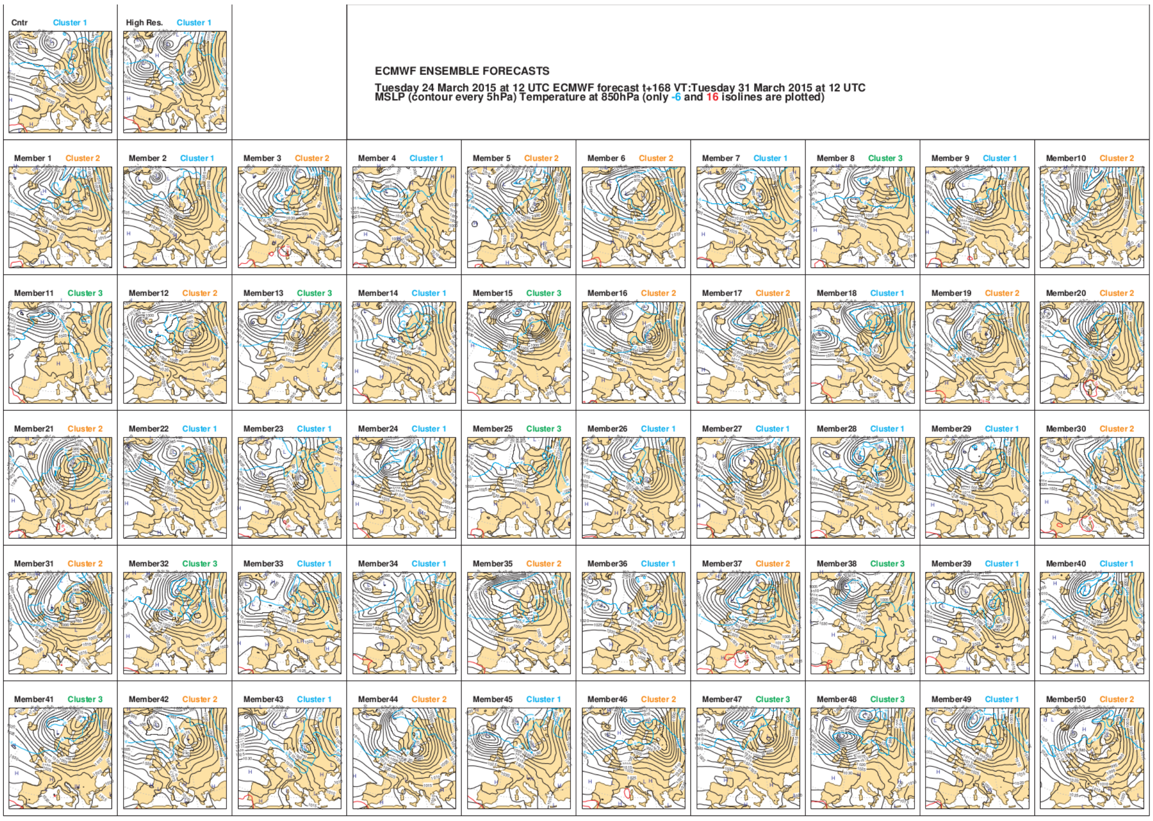


Figure 2.9: Example of a stamp map, as generated operationally at ECMWF. Shown is the 168 hour forecast of storm "Niklas" of the forecast initialised at 12:00 UTC 24 March 2015, valid at 12:00 UTC 31 March 2015. The maps show the individual ensemble members' forecasts of mean sea level pressure (black contour lines, hPa) and 850 hPa temperature (blue contour lines). For each member, the cluster into which it has been classified is listed (cf. Fig. 2.11). (Reproduced by permission of ECMWF.)

in large-scale features, in the example the location and extent of the strong low pressure system over northern Europe, can be recognized by the forecaster.

An alternative plot type that is frequently used is the *spaghetti plot*. In contrast to a stamp map, the spaghetti plot displays all ensemble members in a single image. To avoid cluttering, only few selected contour values are plotted that are chosen to be representative for the visualized field. The more the contours of the individual members diverge, the higher the uncertainty in the forecast. Figure 2.10 shows an example of the 500 hPa geopotential height field as forecast by the NCEP ensemble. At 24 hours lead time, the ensemble members still are very similar to each other except for a region of uncertainty north-west of Greenland. At 144 hours lead time, the members have diverged from each other, although the major features are mostly consistent. Wilks (2011) notes that spaghetti plots have proven to be useful in visualizing the time evolution of the forecast flow, with the contour lines diverging as lead time increases. However, a distinct disadvantage of spaghetti plots is that they become illegible when the members diverge too much.

To identify similarities within ensemble members, they are commonly objectively clustered according to some similarity criterion (Ferranti and Corti, 2011; Atger, 1999). Figure 2.11 shows an example from ECMWF for the time period covering storm Niklas. Ferranti and Corti (2011) describe the current ECMWF clustering algorithm. The 51 ensemble members, as well as the high-resolution deterministic

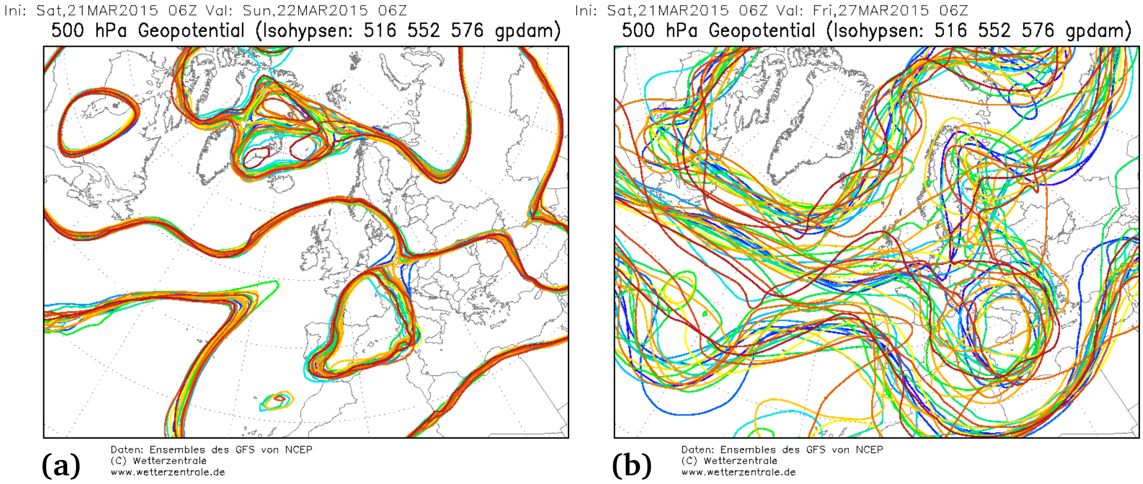

Ini: Sat,21MAR2015 06Z Val: Sun,22MAR2015 06Z
500 hPa Geopotential (Isohypsen: 516 552 576 gpdam)
(a)
Daten: Ensembles des GFS von NCEP
(C) Wetterzentrale
www.wetterzentrale.de
Ini: Sat,21MAR2015 06Z Val: Fri,27MAR2015 06Z
500 hPa Geopotential (Isohypsen: 516 552 576 gpdam)
(b)
Daten: Ensembles des GFS von NCEP
(C) Wetterzentrale
www.wetterzentrale.de

Figure 2.10: Example of a "spaghetti plot". Shown are contour lines of three isovalues (5160 m, 5520 m, and 5760 m) of the geopotential height field at 500 hPa as forecast by all members of the NCEP GEFS (forecast initialised at 06:00 UTC 21 March 2015, valid at **(a)** 06:00 UTC 22 March 2015 (24 hours lead time) and at **(b)** 06:00 UTC 27 March 2015 (144 hours lead time)). While the ensemble members are very similar to each other at 24 hours lead time, they have diverged at 144 hours lead time. (Reproduced from `http://www.wetterzentrale.de`, © www.wetterzentrale.de. Used with permission.)

forecast, are grouped into a small number (a maximum of six) of clusters. For two ensemble members to fall into the same cluster, their development of the 500 hPa geopotential height field over Europe needs to be similar over a given time window. Taking time into account ensures synoptic consistency. The clusters are represented by the members closest to their centre, and assigned to one of four large-scale weather regimes (positive North Atlantic Oscillation (NAO), negative NAO, European Blocking, Atlantic Ridge). The colour of the cluster frame in Fig. 2.11 indicates the regime to which the cluster is most similar (blue for positive NAO, pink for Atlantic Ridge; for details see Ferranti and Corti, 2011). Ferranti et al. (2014) note that forecast skill of the ensemble depends on the weather regime.

As an alternative to plotting direct ensemble output, summary statistics computed per model grid-point are commonly displayed. Inness and Dorling (2013) note three particular advantages of ensembles: (1) the ensemble mean is on average more skilful than any single deterministic forecast, (2) the ensemble standard deviation, or *spread* provides an indication of the forecast uncertainty, and (3) the generation of probability forecasts is possible. These advantages are commonly manifested in maps of ensemble mean and standard deviation, as well as in maps of probabilities for the occurrence of an event. Figure 2.12 shows an example obtained from ECMWF. The plots display the ensemble mean of geopotential height at 500 hPa together with its normalized standard deviation, as well as the standard deviation together with the geopotential field predicted by the deterministic, high-resolution forecast. This depiction indicates which areas of the geopotential height forecast are most affected by uncertainty (large standard deviation). Figure 2.13 shows a typical probability map generated by DWD, displaying the probability of severe precipitation. The probability is computed by determining for each grid box the fraction of ensemble members that exceed a critical threshold. Probability maps are generated mainly for surface parameters relevant for weather warnings, including wind speed, temperature, and precipitation. They can be instantaneous (that is, for a given time step), or computed over a time interval. For example, Persson and Andersson (2013) note that probabilities for extreme wind gusts are computed over a 24 hour period at ECMWF, because it is considered more important to know *that* an

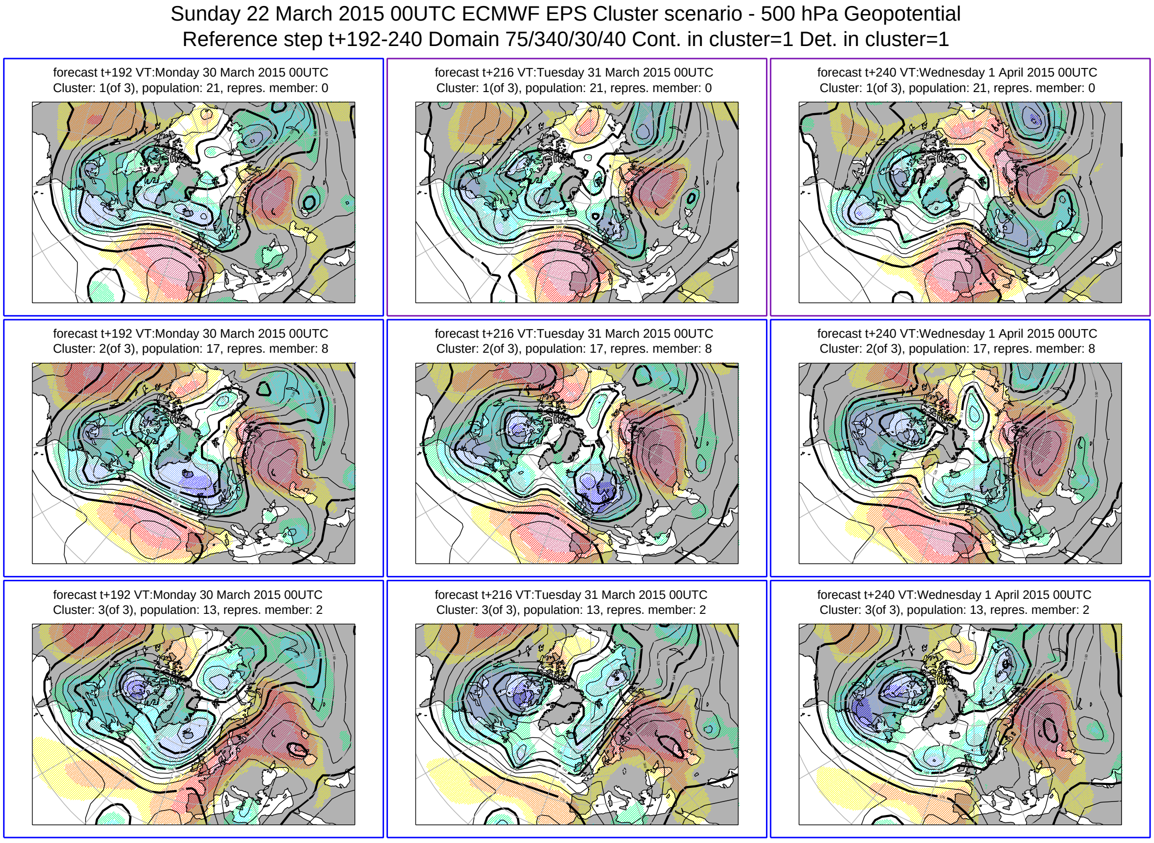

Sunday 22 March 2015 00UTC ECMWF EPS Cluster scenario - 500 hPa Geopotential
Reference step t+192-240 Domain 75/340/30/40 Cont. in cluster=1 Det. in cluster=1
forecast t+192 VT:Monday 30 March 2015 00UTC
Cluster: 1(of 3), population: 21, repres. member: 0
forecast t+216 VT:Tuesday 31 March 2015 00UTC
Cluster: 1(of 3), population: 21, repres. member: 0
forecast t+240 VT:Wednesday 1 April 2015 00UTC
Cluster: 1(of 3), population: 21, repres. member: 0
forecast t+192 VT:Monday 30 March 2015 00UTC
Cluster: 2(of 3), population: 17, repres. member: 8
forecast t+216 VT:Tuesday 31 March 2015 00UTC
Cluster: 2(of 3), population: 17, repres. member: 8
forecast t+240 VT:Wednesday 1 April 2015 00UTC
Cluster: 2(of 3), population: 17, repres. member: 8
forecast t+192 VT:Monday 30 March 2015 00UTC
Cluster: 3(of 3), population: 13, repres. member: 2
forecast t+216 VT:Tuesday 31 March 2015 00UTC
Cluster: 3(of 3), population: 13, repres. member: 2
forecast t+240 VT:Wednesday 1 April 2015 00UTC
Cluster: 3(of 3), population: 13, repres. member: 2

Figure 2.11: Example of a cluster product, based on the 500 hPa geopotential height over Europe as forecast by ECMWF. Shown are three clusters (rows) of the ensemble forecast initialised at 00:00 UTC 22 March 2015, valid at 00:00 UTC **(left)** 30 and **(middle)** 31 March 2015 (192 and 216 hours lead time, respectively), and **(right)** at 00:00 UTC 01 April 2015 (240 hours lead time). The clusters are represented by the member closest to the cluster centre. Note the different extent of the trough over northern Europe. The colour of the figure frames corresponds to the large-scale weather regime to which the cluster is most similar. See text for details. (Reproduced by permission of ECMWF.)

extreme event will occur than *when exactly* it will occur. Due to variation in the ensemble members, the absolute values of probabilities computed per grid point can be low. Hence, probabilities are also commonly computed for areas encompassing multiple grid boxes to determine, for example, whether severe precipitation can occur *somewhere in a given region*. As shown in Fig. 2.13b, the probabilities of severe precipitation occurring in regions of 10× 10 grid boxes are much larger than the probabilities computed per grid point.

As another statistical means to summarise regions in which severe weather events may occur, ECMWF has introduced the *extreme forecast index* (EFI; Lalaurette, 2003), a measure that relates forecast probabilities to the model climate to detect forecast conditions that largely depart from “normal conditions”. The EFI is used, for instance, to generate warning of extreme winds (Petroliagis and Pinson, 2014). Figure 2.14 shows an example of the ECMWF forecast initialised five days prior to the occurrence of storm Niklas. The EFI indicates extreme winds over large parts of Germany.

For point forecasts (that is, for a specific location), *meteograms* are frequently used. Figures 2.15 and 2.16 show operational examples from ECMWF. Time series of box plots (e.g. Wilks, 2011) are used to display statistical ensemble information for the forecast variables that are most relevant for a surface

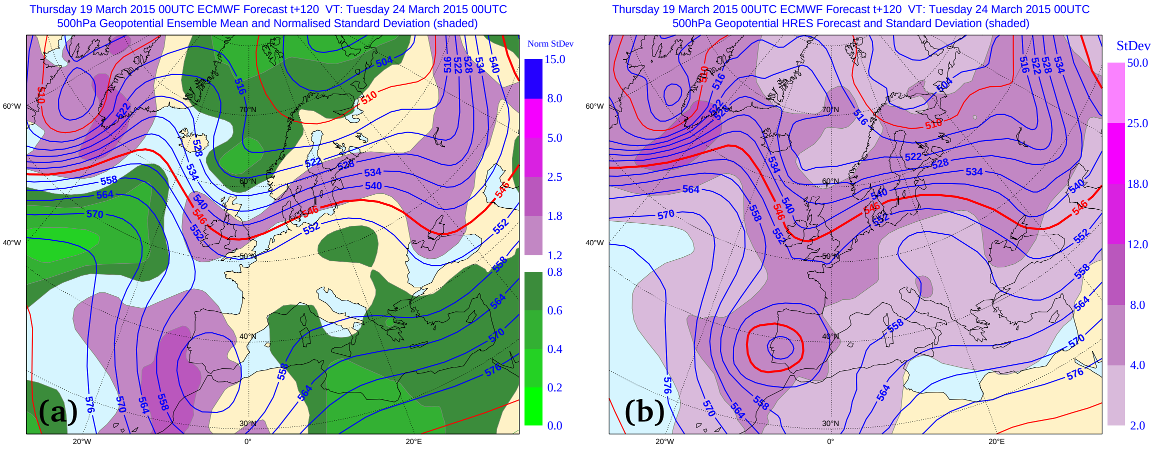

Thursday 19 March 2015 00UTC ECMWF Forecast t+120 VT: Tuesday 24 March 2015 00UTC
500hPa Geopotential Ensemble Mean and Normalised Standard Deviation (shaded)
Norm StDev
15.0
8.0
5.0
2.5
1.8
1.2
0.8
0.6
0.4
0.2
0.0
(a)
Thursday 19 March 2015 00UTC ECMWF Forecast t+120 VT: Tuesday 24 March 2015 00UTC
500hPa Geopotential HRES Forecast and Standard Deviation (shaded)
StDev
50.0
25.0
18.0
12.0
8.0
4.0
2.0
(b)
60°W
40°W
20°W
0°
20°E
70°N
60°N
50°N
40°N
30°N

Figure 2.12: Example of a forecast product showing ensemble mean and standard deviation, as generated operationally by ECMWF. **(a)** Ensemble mean of geopotential height (dam) at 500 hPa (contour lines) and normalized standard deviation of the same field (filled contours). **(b)** Geopotential height (dam) at 500 hPa of the deterministic high-resolution forecast (contour lines) and ensemble standard deviation of the same field (filled contours). Shown is the forecast from 00:00 UTC 19 March 2015, valid at 120 hours lead time at 00:00 UTC 24 March 2015. (Reproduced by permission of ECMWF.)

forecast. In Fig. 2.15, the 10-day ensemble forecast for cloud cover, precipitation, wind speed and temperature is shown. The meteogram shows how the ensemble members are close to each other in the first days of the forecast, then start to diverge. Figure 2.16 shows a meteogram generated from the lower resolution 15-day forecast. Here, forecast information are accumulated into daily mean and displayed together with model climate information. The model climate shows how the current forecast weather deviates from the "norm" and also shows if the ensemble forecast contains more information than climatology (in the example, cloud cover and temperature forecasts in the last few days hardly differ from climatology). Figure 2.16 additionally contains wind rose diagrams to display the distribution of wind direction. Wind roses are traditionally used to show distributions of wind direction time series (for example, the distribution of wind direction at a particular location over a given time period; e.g. Brooks et al., 1946; Crutcher, 1957). Here, they are used to show both temporal and ensemble information (distribution of all members over one day)[31], with wind directions clustered into octa nts. Note the forecast wind speed distribution for 31 March 2015 (9-day forecast in Fig. 2.15 and 12-day forecast in Fig. 2.16). The 12-day forecast already indicates above-average wind speeds, with single members indicating the chance for the extreme storm event that occurred (the average daily wind speed is predicted to be on the order of 10ms^{-1}). Also, *plume plots*, a combination of spaghetti plots and probability maps, are used to display the temporal evolution of other meteorological quantities at the location of interest. Figure 2.17 shows an example from ECMWF showing the evolution of upper level temperature and geopotential height above Munich.

ECMWF also routinely operates a feature detection and tracking method to visualize the evolution of cyclonic features in the ensemble forecasts (Hewson, 1997, 1998, 2009; Hewson and Titley, 2010). An example is provided in Fig. 2.18. The display shows surface cold and warm fronts detected in the individual ensemble members as line features in a spaghetti plot. With increasing forecast lead time, the spaghetti depiction quickly becomes illegible, however, the ECMWF website[32] provides alternative

[31] Compare to `http://www.ecmwf.int/en/forecasts/charts/medium/ens-meteograms` (last access April 2015).

[32] ECMWF "Cyclone Database" available at `http://old.ecmwf.int/products/forecasts/cdb` (last access April 2015).

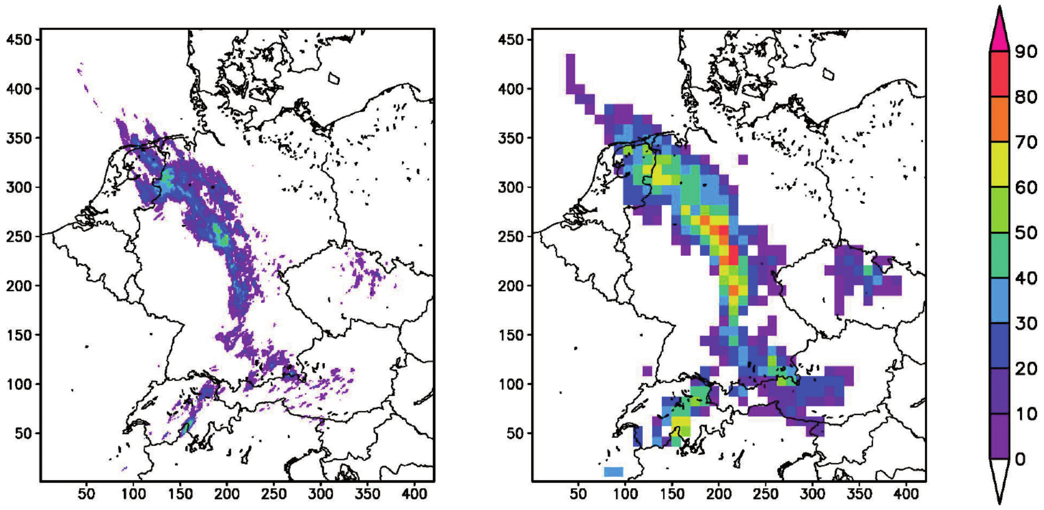
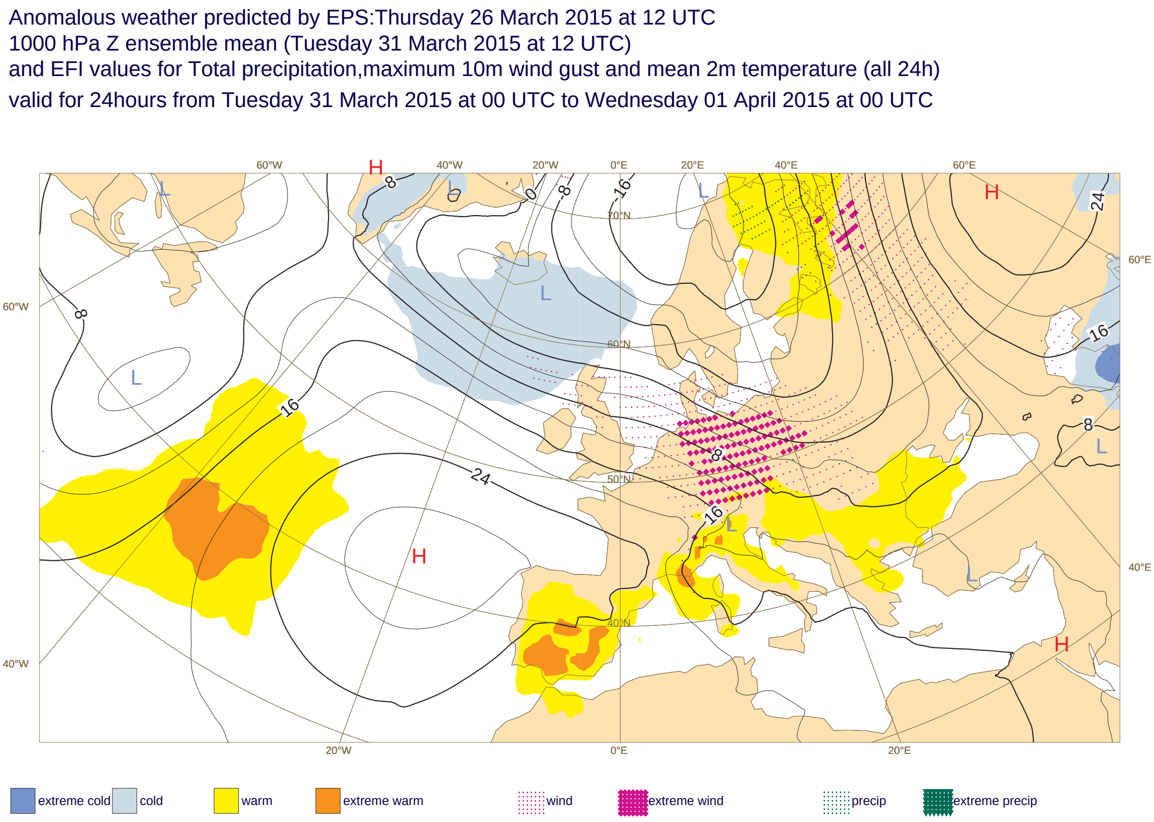


Figure 2.13: Example of a probability field derived from an ensemble forecast. Shown is the probability (%) of severe precipitation, forecast by the COSMO-DE ensemble at DWD. **(a)** Probability of severe precipitation occurring in a 2.8 km grid box. **(b)** Probability of severe precipitation occurring in a region of 10 × 10 grid boxes. Note how the absolute values of probability are much larger in (b). (Reprinted from Theis and Gebhardt (2009), © Deutscher Wetterdienst 2010. Used with permission from DWD.)

means to view the individual ensemble member's features (including, for example, animation). Further cyclonic features (for example, centre of low pressure system, developing waves) are available as well.

Besides the presented general products, specialised ensemble products have been reported in the literature. Novak et al. (2008) point out that uncertainty information is primarily requested by "sophisticated users" such as emergency managers, and hence the "forecasters' operational uncertainty needs are intimately tied to the end products and services they produce". Products include, for example, forecasts of turbulence regions for aviation (Gill and Buchanan, 2014) and extratropical storm and hurricane forecasts (Froude et al., 2007; Hamill et al., 2012). Stephenson and Doblas-Reyes (2000) discuss further statistical approaches to summarising and interpreting output from ensemble predictions. Gneiting and Raftery (2005) emphasise the potential of statistical post-processing to maximise forecast value.



Anomalous weather predicted by EPS:Thursday 26 March 2015 at 12 UTC
1000 hPa Z ensemble mean (Tuesday 31 March 2015 at 12 UTC)
and EFI values for Total precipitation,maximum 10m wind gust and mean 2m temperature (all 24h)
valid for 24hours from Tuesday 31 March 2015 at 00 UTC to Wednesday 01 April 2015 at 00 UTC

Figure 2.14: The ECMWF extreme forecast index relates ensemble predictions to model climatology to detect anomalous weather conditions in the forecast. Shown is the EFI computed from the ensemble forecast initialised at 12:00 UTC 26 March 2015, valid for the 24 hour period covering 31 March 2015. On 31 March 2015, storm “Niklas” hit central Europe with gale-force winds. Note how five days prior to the event the EFI predicted extreme winds over large parts of Germany. (Reproduced by permission of ECMWF.)

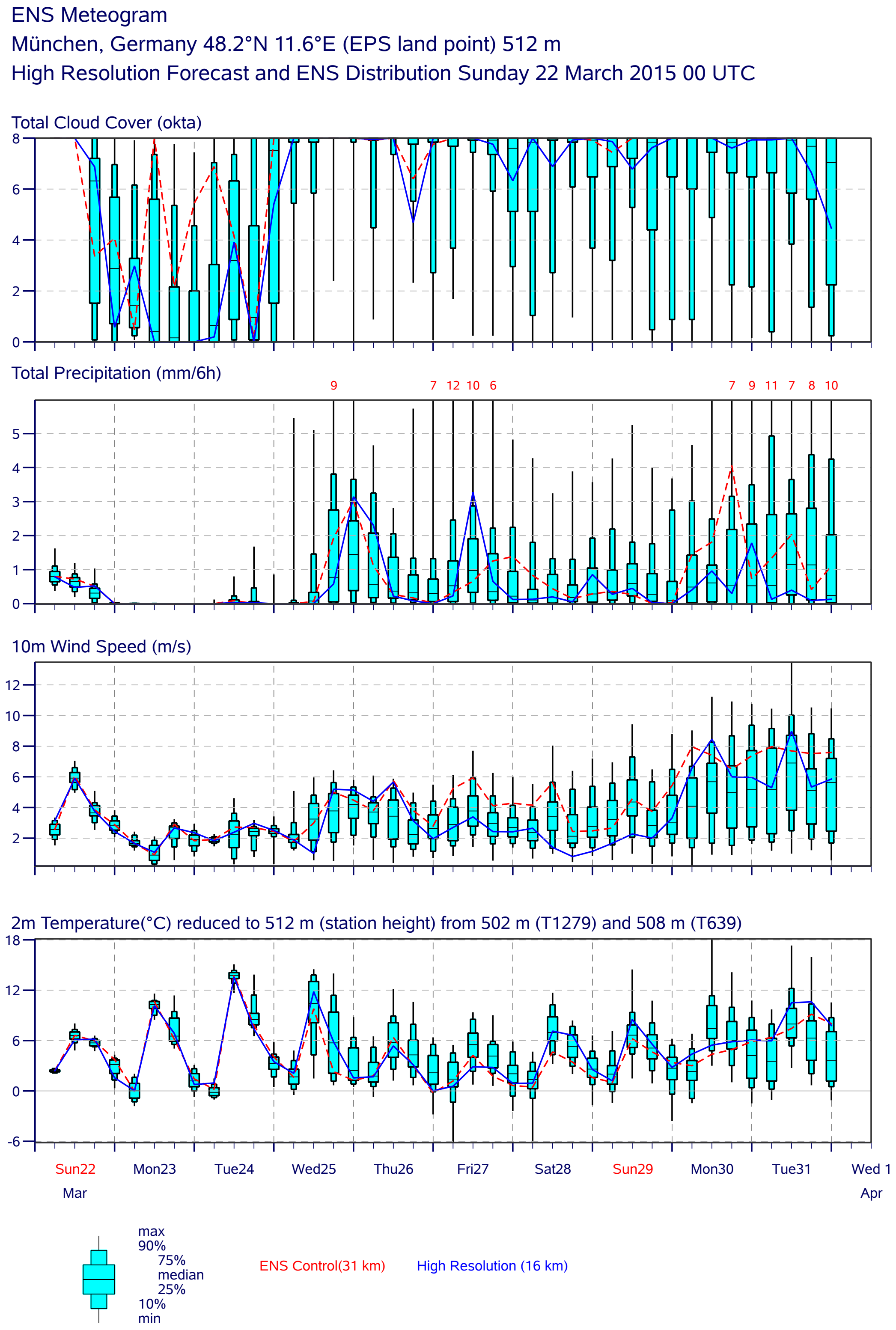
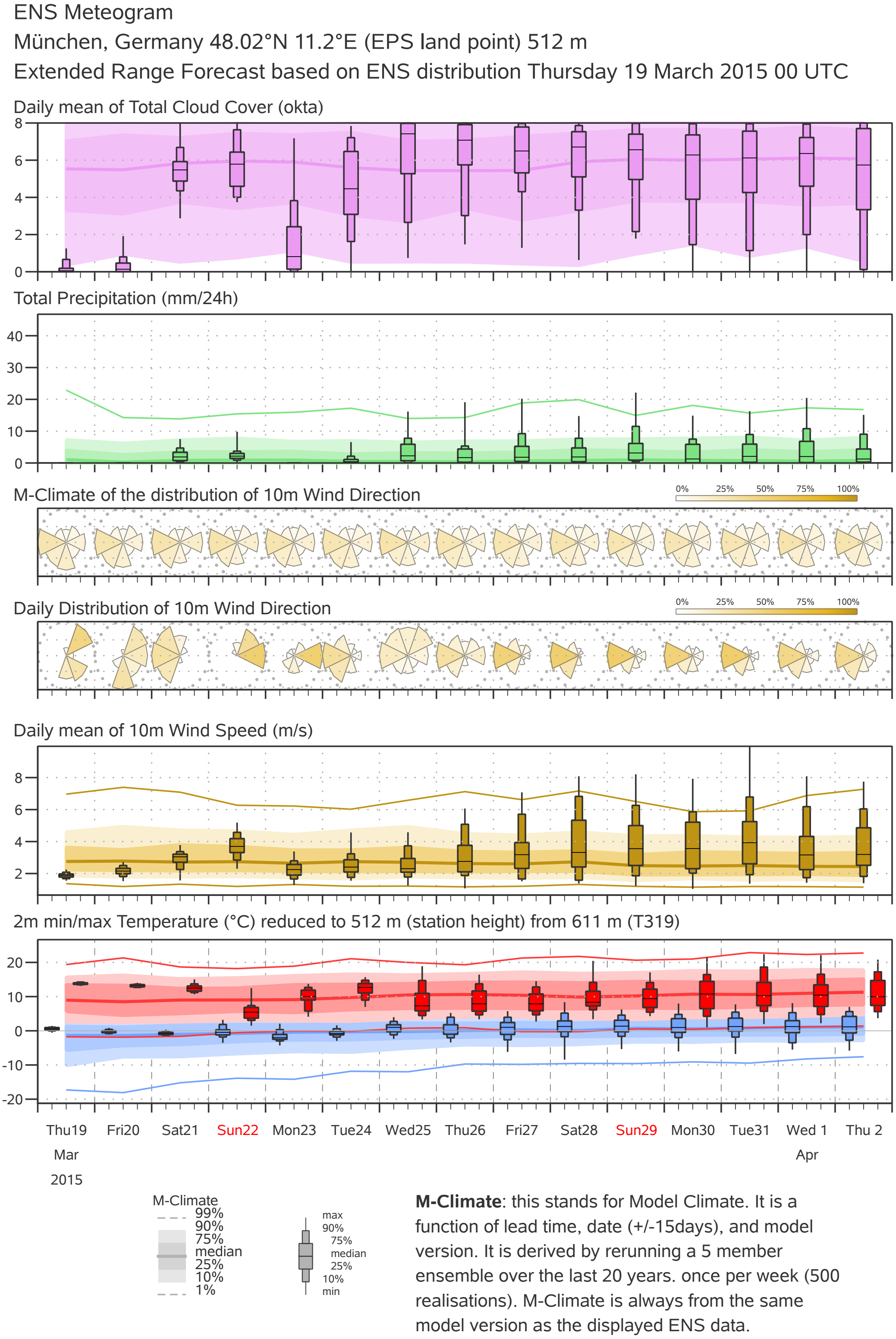


Figure 2.15: Example of an ECMWF 10-day meteogram, depicting time series of selected surface parameters for a particular location. Shown is the prediction of the deterministic (blue lines) and ensemble control (red lines) forecasts initialised at 00:00 UTC 22 March 2015 for Munich, Germany. The ensemble distribution is displayed by means of boxplots at each time step. Note how some members forecast high wind speeds (averages over 10 ms^{-1}) for 31 March 2015, in the example nine days into the future. (Reproduced by permission of ECMWF.)



M-Climate: this stands for Model Climate. It is a function of lead time, date (+/-15days), and model version. It is derived by rerunning a 5 member ensemble over the last 20 years. once per week (500 realisations). M-Climate is always from the same model version as the displayed ENS data.

Figure 2.16: ECMWF 15-day meteogram for Munich, Germany, for the forecast initialised at 00:00 UTC 19 March 2015. This diagram relates forecast daily averages to the model climate (coloured background bars). Note how the ensemble already predicts the possibility of high wind speeds on 31 March 2015 (12-day forecast). (Reproduced by permission of ECMWF.)

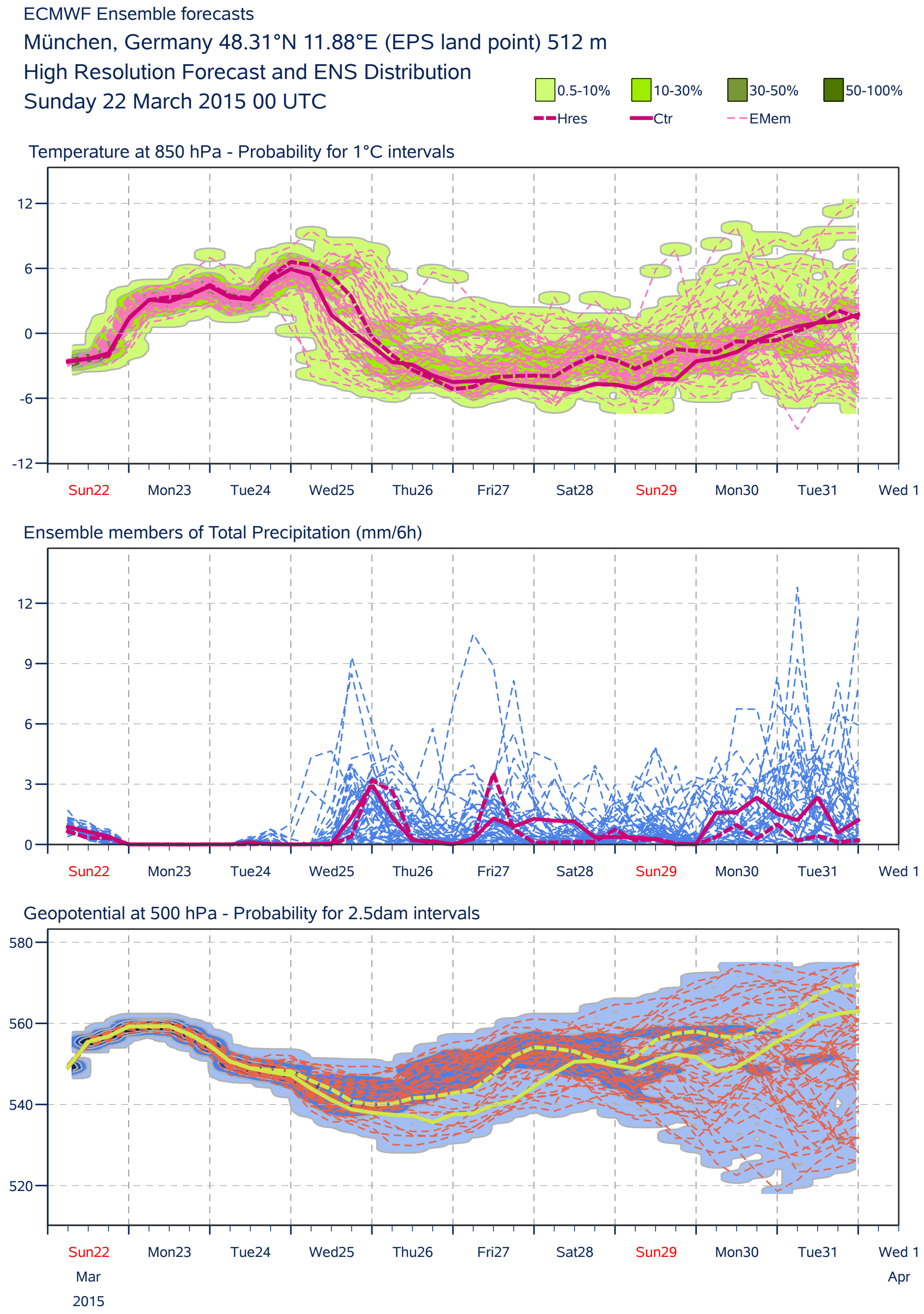


Figure 2.17: Example of an ECMWF plume diagram, depicting time series of selected forecast parameters at a specific location as a spaghetti plot and probability plumes derived thereof. Shown is the ensemble forecast initialised at 00:00 UTC 22 March 2015 for Munich, Germany. (Reproduced by permission of ECMWF.)

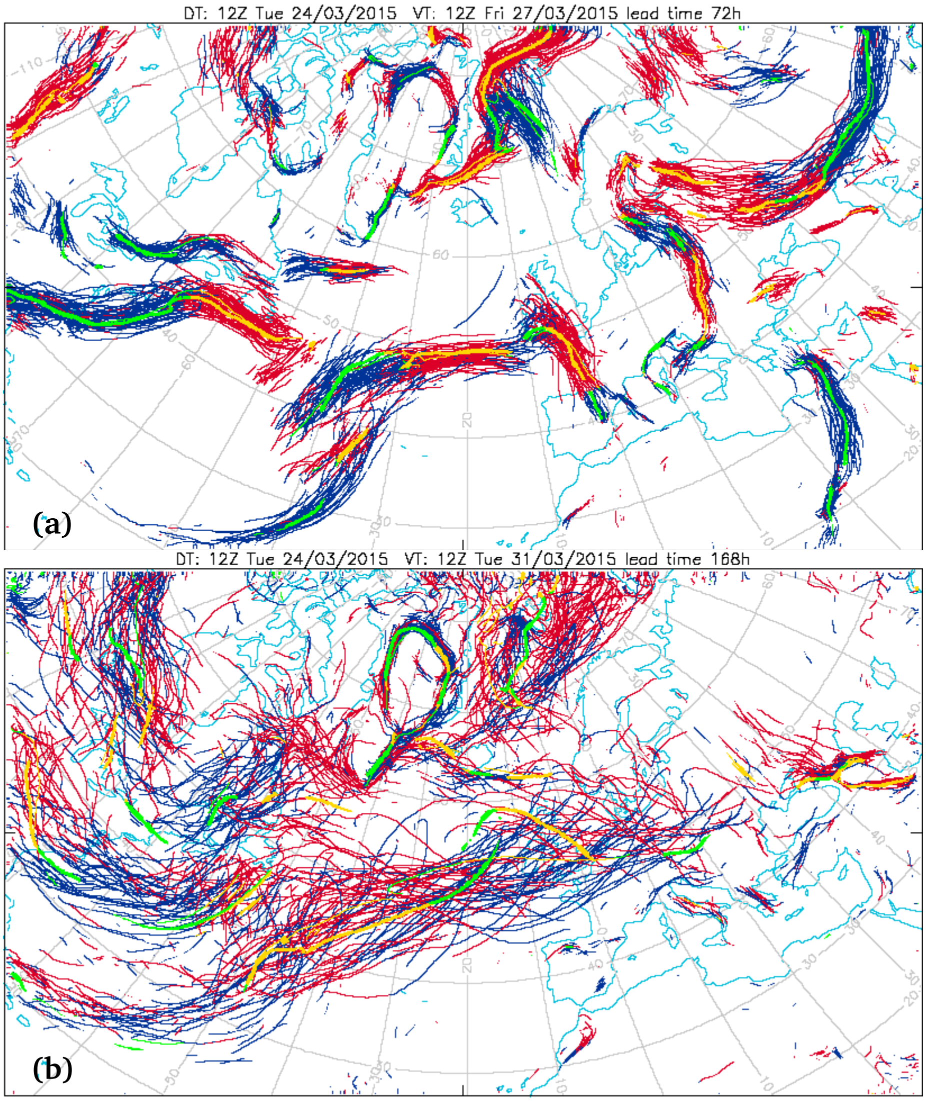

DT: 12Z Tue 24/03/2015 VT: 12Z Fri 27/03/2015 lead time 72h
(a)
DT: 12Z Tue 24/03/2015 VT: 12Z Tue 31/03/2015 lead time 168h
(b)

Figure 2.18: Example of spaghetti plots depicting surface fronts detected with the Hewson and Titley (2010) algorithm. Shown is the ensemble forecast initialised at 12:00 UTC 24 March 2015, valid at **(a)** 12:00 UTC 27 March 2015 (72 hours lead time) and **(b)** 12:00 UTC 31 March 2015 (168 hours lead time). (Reproduced by permission of ECMWF.)

CHAPTER 3

Foundations of visualization

The visualization of scientific data is the second basis for this thesis. This chapter reviews fundamentals of visualization and computer graphics, surveys GPU-based real-time rendering, and describes approaches to visualization system design that are relevant for my work.

The purpose of visualization is to create human-readable images from "raw" data generated by observation or simulation of a given real-world phenomenon. The visualization process includes data processing, mapping of numerical data to graphical representations, and generation of a digital image from these representations (a process called *rendering*). The latter step uses methods from the field of *computer graphics*. Computer graphics is primarily concerned with techniques for rendering (often complex) geometric models by, for example, simulating the transport of light in a scene in order to achieve photorealistic depictions.

Visualization research is commonly distinguished with respect to *scientific visualization* and *information visualization*, although both fields overlap (Ward et al., 2010) and alternative classifications and terminology have been proposed (e.g. Tory and Möller, 2004). Roughly, scientific visualization focusses on the visualization of spatio-temporal, physically based data by depicting the spatial and temporal distribution and evolution of scalar, vector, and tensor data. Information visualization, on the other hand, focusses on abstract (often not physically based) data that may include dimensions other than space and time. This thesis is primarily concerned with methods from scientific visualization, although, following Ward et al. (2010), I do not make "hard" distinctions in the following discussion.

3.1 Basics of visualization and computer graphics

Originally considered a subfield of computer graphics (Ward et al., 2010), the field of *scientific* visualization emerged as its own area of research in the late 1980s (McCormick et al., 1987; DeFanti et al., 1989). A number of textbooks discuss its foundations, each focussing on different key aspects. Recent

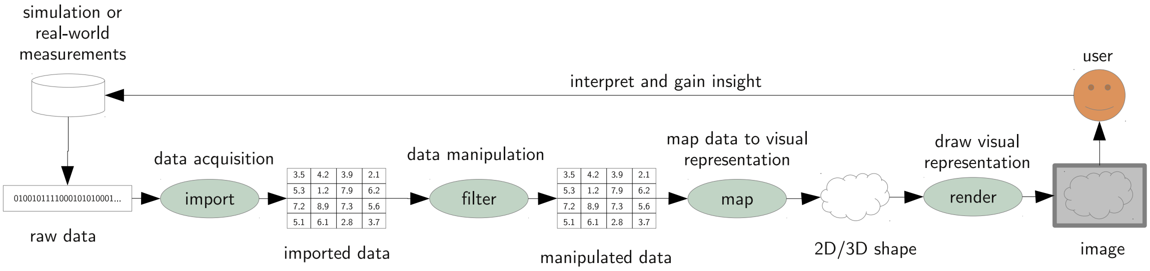


Figure 3.1: Conceptual depiction of a visualization pipeline, following Telea (2014). Data from a numerical simulation or from real-world measurements is imported into the visualization system's internal data model and "flows" through a series of modules that can be considered as a pipeline. The conceptual stages depicted here include data filtering (or manipulation), mapping the data to visual representations (for example, 2D or 3D shapes), and rendering a final image from the visual representations. The user interprets the obtained image and gains insight into the modelled or observed phenomenon.

texts include (in temporal order) Telea (2014), Munzner (2014), Ware (2012), Bethel et al. (2012), Cairo (2012), Ward et al. (2010), Schroeder et al. (2006), and Hansen and Johnson (2005). The book by Telea (2014) focusses on the algorithm level of scientific visualization. Similarly, Ward et al. (2010) explore visualization techniques starting from the algorithmic base, while Ware (2012) explores visualization techniques from a perceptual point of view. The book by Hansen and Johnson (2005) contains an extensive number of essays on all topics of visualization, from algorithmic details to visualization systems. While focussing on the *Visualization Toolkit* (VTK) software library, the book by Schroeder et al. (2006) also serves as a detailed introduction to scientific visualization. Cairo (2012) focusses on information visualization, and Bethel et al. (2012) discuss aspects of high-performance visualization. I also note the introduction to information visualization in the atmospheric domain contained in the PhD thesis by Nocke (2007).

Similarly, a range of books exist that introduce the basics of computer graphics, some of which also contain chapters that explain basic principles of visualization. Here, recent examples include Akenine-Möller et al. (2008), Theoharis et al. (2008), Shirley et al. (2009), Angel and Shreiner (2011), June (2011), Hughes et al. (2013), Boreskov and Shikin (2013), and Guha (2015).

3.1.1 The visualization pipeline

The visualization pipeline is discussed in a number of textbooks, including Schroeder et al. (2006) and Bethel et al. (2012). Here, I follow the discussion by Telea (2014).

The process of visualizing data obtained from a numerical simulation or from real-world measurements can conceptually be decomposed into a sequence of steps. Figure 3.1 illustrates the decomposition. The visualization process can be considered as a "pipeline" through which the data flows. The literature discusses four pipeline stages (which, however, may consist of multiple steps each):

Import The import stage imports raw data into the pipeline. This typically includes reading data from a hard-drive, possibly also resampling data into the internal data model used by the application.

Filtering Once imported, the data may be manipulated in multiple ways, a process known as *filtering*. Filtering can include, for instance, computations on the data, derivation of additional data fields (for example, the computation of potential temperature from pressure and temperature), and masking of invalid data. The result of the filtering stage is a data field that can be visualized.

Mapping The mapping stage transforms the data into an abstract visual representation. A simple example would be the computation of a contour line from a 2D data field. The output of the mapping stage typically includes geometric primitives (that is, points, lines, or triangles).

Rendering Finally, the rendering stage computes an output image by applying computer graphics techniques. The output generated by the mapping stage is, together with user defined parameters including camera viewpoint and lighting, converted to an image. Rendering includes coordinate transforms, lighting, texture mapping, and rasterisation algorithms (details will be given below).

The mapping stage is the part most characteristic to visualization, as Telea (2014) points out. This stage produces the graphics that the user "mentally inverts" to draw conclusions from the original data. Note that the user can only interpret output generated by this stage: If, for example, only a single isovalue is transformed into a contour line, it is impossible to make any inference about the rest of the data. Nevertheless, the applied rendering techniques are also important to facilitate inference. If, for instance, a 3D scene is rendered from a "wrong" viewpoint or with "bad" lighting, the resulting image is not of value either.

3.1.2 Basic (scientific) visualization techniques

Following the conceptual pipeline model, this section introduces basic techniques that can be categorised primarily into the mapping stage, but also into the rendering stage. Focus is put on techniques that would commonly be considered as part of *scientific* visualization. I follow the algorithmic approach used, for example, by Telea (2014). From an algorithmic viewpoint, the literature commonly classifies visualization algorithms according to the *type of data* they operate on. Major classes include (e.g. Schroeder and Martin, 2005; Schroeder et al., 2006; Telea, 2014):

- *Scalar algorithms* operate on datasets that are characterised by possessing exactly one data value per data sample (a *scalar attribute*). An example is the generation of contour lines from a temperature field.

- *Vector algorithms* operate on datasets that are characterised by possessing a *vector attribute* per data sample. Here, an example is to draw vector arrows (*glyphs*) for a wind field.

- *Tensor algorithms* operate on datasets that are characterised by possessing a *tensor attribute* per data sample. An example is the depiction of the curvature of a 2D surface by glyphs.

- *Modelling algorithms* include algorithms that do not fit into the first three classes. Examples include *slicing* through a dataset to reduce its dimensionality, and the modification of generated geometry (for instance, simplification).

In the following, I briefly discuss options to represent discrete, sampled data (as output by numerical simulations), and provide an overview of visualization algorithms most important to this thesis.

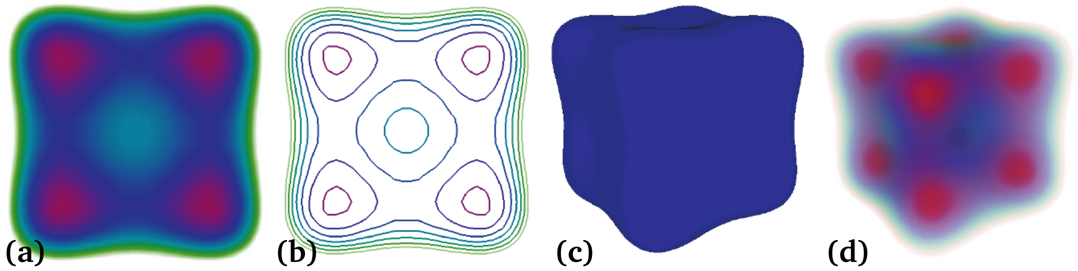


Figure 3.2: The four basic ways to visualize a scalar field in 2D and 3D. **(a)** 2D colour mapping. **(b)** 2D contour lines. **(c)** 3D isosurface. **(d)** 3D direct volume rendering. (Reprinted from Callahan et al. (2008), © 2008 IEEE. Used with permission.)

Data representation

Many visualization algorithms for spatial data require access to samples of a data field at arbitrary locations in space. Hence, for gridded data, some sort of spatial interpolation is required. In the simplest case, the data is available on a uniform grid with regular grid spacing in all three space dimensions. Such grids are straightforward to store in computer memory, it is easy to determine the grid box in which a particular sample point is located, and basic linear interpolation techniques can be applied.

As described in Chapter 2, several grid types are in use for NWP models. Most of them are not regular in at least one dimension (often in the vertical dimension). Depending on the degree of irregularity, sampling a data value at an arbitrary position becomes computationally expensive. The grid type is hence important for the performance of a visualization algorithm. The following types are commonly discussed in the literature (described here for 2D grids, the terminology generalises to 3D; further details are provided by, for example Hansen and Johnson, 2005; Schroeder et al., 2006):

- *Structured grids* have a matrix-like ordering of the discrete sample points. The points can be freely positioned in space, however, neighbour relations are always given by the matrix ordering. Hence, every row and column of the grid holds the same number of elements. Grid cells all have four corner points, but they do not have to have rectangular geometry.
- *Rectilinear grids* are structured grids whose cells are rectangles. The cells, however, can have different sizes.
- *Regular grids* are rectilinear grids whose cells all have the same size.
- *Unstructured grids* include all grids that are not structured.

This terminology can also be applied to individual dimensions. For example, an NWP grid can be regular in longitude and latitude in the horizontal, but structured (using terrain-following coordinates) in the vertical. A model working with a reduced Gaussian grid (cf. Sect. 2.1.2) has an unstructured grid, as the number of grid points decreases towards the poles.

2D scalar field visualization

A simple method to visualize gridded 2D scalar data (and frequently used in the atmospheric sciences) is *colour mapping* (Fig. 3.2a). A *colourmap* function, often implemented as a lookup table (LUT) with a discrete number N of colours, maps a given range of scalar values to colours. Colour mapping can be

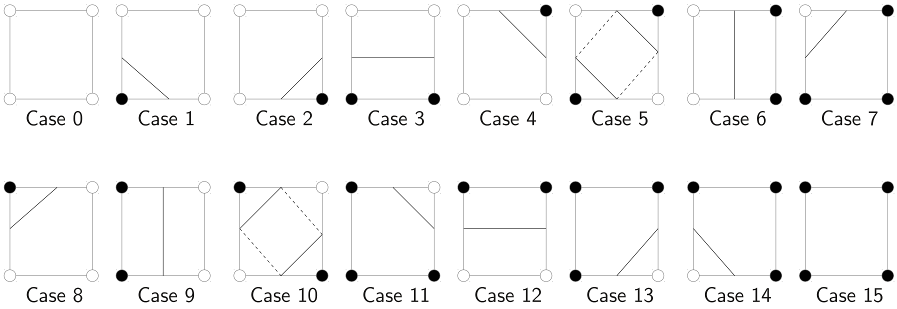


Figure 3.3: The sixteen cases for potential line segments in the marching squares contouring algorithm. Black vertices represent scalar values above the isovalue, white vertices represent scalar values below the isovalue. Note that cases 5 and 10 are ambiguous, with both options equally valid.

implemented to depict individual grid boxes (pseudo-colour plots, cf. Fig. 2.13) or, using interpolation, to depict spatially smooth regions of the N colours (cf. Fig. 2.12).

Another standard technique commonly encountered in the atmospheric sciences is *contouring*. A contour line, also called isoline, is defined as the set of all points that in the spatial domain D covered by a scalar grid have the same scalar value. Contour lines have the property of being perpendicular to the scalar field's gradient. A contour line never stops inside D; it either loops or stops at the boundary of D. Also, a contour line never intersects with itself or with any other contour line in D (Telea, 2014). Two basic classes of algorithms exist to compute contour lines (e.g. Schroeder and Martin, 2005):

- *Cell-based techniques* operate independently on each grid cell. Based on the scalar values of the four grids point surrounding a cell, a contour line segment is generated. The most well-known algorithm of this type is the *marching squares* algorithm (e.g. Wenger, 2013). Marching squares classifies the values of the four scalar values relative to the isovalue into one of 16 topological cases (Fig. 3.3). Based on the case and the actual scalar values, a linear line segment is generated that connects the intersections of the contour line with the cell boundaries. An advantage of cell-based techniques is that parallelisation is straightforward, as the operations are independent for each grid cell. However, unconnected, unsorted line segments are generated – a disadvantage if the continuous geometry of a particular contour line is required for further processing.

- *Line-tracing techniques* (also called *edge-tracking techniques*), in contrast, trace each individual contour line through the entire domain D, yielding the continuous line geometry. However, line-tracing techniques are more difficult to implement, and parallelisation is more involved.

An important aspect for quantitatively interpreting a contour plot is *labelling* of the contour lines. Here, a common requirement is the availability of continuous line geometry (for instance, to find line segments with low curvature). Thus, cell-based methods pose disadvantages if labelling is required. Also, if the grid spacing is coarse, simple linear interpolation as used by marching squares can lead to "jagged" contour lines. Here, higher order interpolation methods can be used. For example, the ECMWF Magics++ library (cf. Sect. 4.2.2) implements interpolation methods proposed by Akima (1974, 1996). For further information on contouring methods, I refer the reader to, for example, Watson (1992) and Wenger (2013).

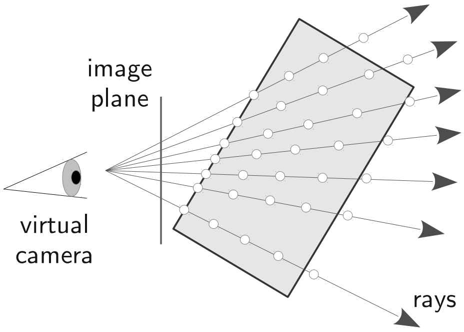


Figure 3.4: Illustration of the *raycasting* algorithm. For each pixel in the output image (the image plane), a ray is cast from the virtual camera through the pixel. The ray is sampled at discrete points in the data volume. For direct volume rendering, the sampled values are accumulated and the pixel colour is computed. For isosurface rendering, the sample positions that enclose the isosurface are determined.

3D scalar field visualization

Analogous to 2D colour mapping and contouring, 3D visualization methods for scalar fields can be *direct* (mapping a part of the volume to a colour) and *indirect* (generating geometry that represents the scalar field). With respect to indirect methods, 3D isosurfaces (Fig. 3.2c) are the 3D analogues to 2D contour lines, displaying the surface corresponding to a single data value. Compared to a 2D isoline, a 3D isosurface can display the spatial structure of the field. However, while 2D plots can (due to the lines not intersecting) accommodate a range of contour lines, only few isosurfaces can be rendered in a single image without cluttering. The *marching cubes* algorithm (Lorensen and Cline, 1987) is the generalisation of marching squares to three dimensions, yielding geometrical primitives that resemble the surface. Instead of line segments, the marching cubes algorithm outputs triangles. With eight vertices per 3D grid cell, marching cubes needs to treat $2^8 = 256$ topological cases (which, however, can be reduced using symmetry).

Direct volume rendering (DVR; e.g. Kaufman and Mueller, 2005; Callahan et al., 2008), in contrast, does not extract geometry. In broadest generality, DVR can be regarded as the simulation of radiative transfer through a medium whose optical properties are defined by the scalar field. In this respect, the technique bears resemblance to radiative transfer modelling in the atmospheric sciences (e.g. Liou, 2002; Petty, 2006). However, in most applications of DVR, the scalar data to be visualized is not associated with physical properties that describe absorption, emission, and scattering. Hence, such properties need to be artificially prescribed. Also, since physically accurate radiative transfer is computationally too expensive for typical visualization applications, much simplified models are used.

A common approach to DVR is *raycasting*, although alternative methods exist. Engel et al. (2006) provide a detailed treatment of the topic. Figure 3.4 illustrates the raycasting principle. For every pixel in the output image (the image plane in Fig. 3.4), a ray is cast from the virtual camera (the observer's eye) through the data volume. In the simplest form, the data volume is sampled along each ray at uniform steps. The sampled scalar values are mapped to optical properties using a *transfer function*. An example of a simple transfer function is a colourmap that maps the scalar value to a colour. For volume rendering, opacity becomes important (otherwise only the first sampled value would contribute to the final pixel colour), hence, a scalar value also needs to be mapped to transparency. More complex transfer functions can also be multidimensional in order to map additional information to colour and transparency. An example is to use both the sampled scalar and the scalar field's gradient at the sample position. With *front-to-back raycasting* and using a transfer function, values sampled along a ray can simply be accumulated using iterative *alpha compositing* (Engel et al., 2006):

$$C_i = C_{i-1} + (1 - \alpha_{i-1}) C_{src} \tag{3.1}$$

$$\alpha_i = \alpha_{i-1} + (1 - \alpha_{i-1}) \alpha_{src}. \tag{3.2}$$

Here, C_i denotes the colour of the pixel after evaluation of sample i, C_{i-1} is the colour of the previous sample, C_{src} is the colour obtained from the transfer function for the current sample. The α values are correspondingly for opacity.

Note that the raycasting approach also can be (and is in this thesis) used to visualize isosurfaces. For this purpose, the compositing process is reduced to finding the position along a ray at which the isosurface is crossed. If, during traversal, two subsequent samples enclose the isovalue, a binary search can be applied to find the crossing position between the samples. If multiple, transparent isosurfaces are rendered, the individual surfaces' colour and transparency values enter the compositing equations.

Vector field visualization

Similar to scalar field visualization, a multitude of techniques for vector fields have been presented in the literature. Basic techniques are presented in the books by, for example, Telea (2014), Hansen and Johnson (2005), and Theoharis et al. (2008). Detailed reviews are provided by Post et al. (2003), Laramee et al. (2004, 2007), McLoughlin et al. (2010), Pobitzer et al. (2011), and Edmunds et al. (2012). Basic techniques include:

- *Glyph-based techniques* depict the direction (and optionally the magnitude) of the vector field at a set of positions by drawing glyphs (icons) that depict characteristics of the flow field. Basic glyphs include arrows (which can be scaled or coloured by magnitude), and –common in meteorology– wind barbs. Glyphs placement strategies include simple placements at regular intervals, as well as methods that, for example, account for characteristics of the flow field.

- *Integral curves* (e.g. McLoughlin et al., 2010) are curves tracing the displacement of a sample point by the flow field. *Streamlines* trace the path of a mass-less particle in a stationary (that is, not time-dependent) vector field; *pathlines* (better known as *trajectories* in the atmospheric sciences) do the same in time-dependent vector fields. *Streaklines* are formed by connecting particles that are constantly released at a specific point. For example, in the real world the constant release of dye at a fixed position in a fluid would form a streakline.

Both glyphs and integral curves generalise to 3D. To add spatial extent, integral curves in 3D are commonly rendered as 3D tubes by extruding line segments to cylinders (Bloomenthal, 1990; Ueng et al., 1996; Fuhrmann and Gröller, 1998; Stoll et al., 2005). In 3D, *stream (streak/path) ribbons* and *stream (streak/path) surfaces* are also used. Stream ribbons visualize a small ribbon of fixed width along a streamline, visualizing the twisting of the vector field around the direction of advection. Stream surfaces generalise the concept to surfaces by advecting an entire seed curve.

Further methods for the visualization of vector fields include texture based methods (Laramee et al., 2004). For example, *line integral convolution* (LIC) "smears" the pixels of a texture (often a simple noise texture) along the vector field. Texture based methods, as well, can be generalised to 3D. Finally, *feature based methods* extract flow features and track, for example, their temporal evolution (e.g. Post et al., 2003). Specific features commonly discussed in the literature are vortices and critical points (points where the vector magnitude is zero). For an overview of further flow visualization techniques, I also refer the reader to the PhD thesis by Bürger (2010).

Modelling techniques

A further category, referred to by Schroeder and Martin (2005) and Telea (2014) as *(domain) modelling techniques*, includes operations on datasets that modify the sampling domain representation of the data (that is, the representation from which data samples are obtained). Examples include *cutting* (the extraction of data along prescribed geometry) and *selection* (extraction of data based on data properties).

Interaction concepts

Nocke (2007) points out that a particular visualization challenge posed by large multivariate (that is, containing multiple scalar or vector attributes), multidimensional datasets is the limited capacity of graphical output devices to represent multiple data dimensions and attributes simultaneously. A number of presentation and interaction concepts are referred to in the literature to approach this problem, including (e.g. Kehrer and Hauser, 2013):

- *Navigation* techniques enable the user to interactively zoom, rotate, or pan the displayed data to obtain both overview and more detailed views.
- *Focus+context* techniques show a part of the data in detail (focus), while also displaying a context for overview and navigation purposes. An example is the display of a magnifier glass. Focus+context techniques are surveyed by Cockburn et al. (2009).
- *Coordinated multiple views* provide simultaneous visualizations of a dataset from different perspectives. For example, attributes of the dataset may be visualized as a scatterplot in one view, while a volume rendering is shown in another. *Brushing and linking* techniques allow the user to interactively select (brushing) subsets of the data in one view (for instance, in the scatterplot) that are subsequently highlighted in the coordinated views (linking).

3.1.3 Perceptual issues

Human visual perception is an intrinsic part of visualization and discussed, for example, by Ware (2012), Hansen and Johnson (2005), Ward et al. (2010), Shirley et al. (2009), and others. This section provides a brief overview of issues related to perception that are important for this thesis. In particular, this includes spatial perception and the perception of colour.

Colour

Colour is inextricably linked to perception as humans use colours to refer to the perception of different wavelengths of light. To systematically classify colours, a *colour model* is required that allows to express

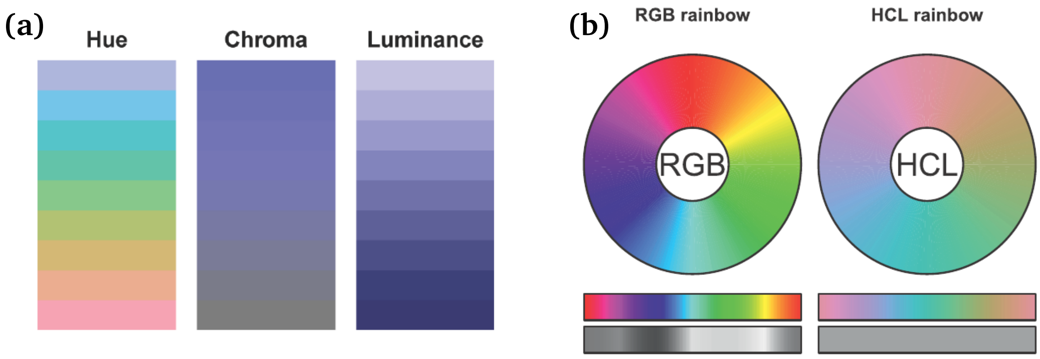


Figure 3.5: The HCL (hue-chroma-luminance) colour model. **(a)** *Hue* describes the actual colour, *chroma* the perceived intensity of the colour, and *luminance* its brightness. **(b)** An RGB-based rainbow colourmap exhibits distinct variations in luminance (shown by the grey bar at the bottom of the figure) and very inhomogeneous colour gradients. In contrast, an HCL-based rainbow provides colours with constant luminance. (Reprinted from Stauffer et al. (2015), © American Meteorological Society. Used with permission.)

a given colour as a combination of basis colours (Theoharis et al., 2008). *Device-dependent* and *device-independent* models are in use, reflecting whether the model's coefficients used to describe a colour will represent the same perceived colour on different output devices. Also, colour models are classified into *additive* and *subtractive* models. An additive colour model represents a colour by adding primary colour components (as done for computer screens), a subtractive model subtracts components (used in printing or painting).

Theoharis et al. (2008) note that an important property of a colour model is *perceptual linearity*. Perceptual linearity refers to the perceived difference between two colours being the same across the entire range of colours. Another important property is *intuitiveness* in usage. Important colour models include:

- In 1931, the Commission Internationale d'Eclairage (CIE) released the *CIE XYZ* colour model, the "mother of all colour models", as Theoharis et al. (2008) note. Three basic colours **X**, **Y** and **Z** form a device-independent basis. The model, however, is neither perceptually linear nor intuitive.

- The *RGB* colour model, widely used in computer graphics, is an additive colour model using red, green and blue as its basis. The RGB model is device-dependent and will potentially produce perceptually different colours on different devices. It is not perceptually linear.

- The *HSV* (hue-saturation-value) colour model aims at providing an intuitive way to control a colour's hue (the actual colour), saturation (grey to full colour), and intensity (or value; black to full colour). HSV can easily be converted to RGB, however, it is not perceptually linear, either.

- Transformations of CIE XYZ that aim at perceptual linearity are the *CIE L*a*b** (luminance-a-b) and *HCL* (hue-chroma-luminance) colour models. Figure 3.5a illustrates the three HCL bases. In the HCL model, the a and b coefficients of CIE L*a*b* are expressed in spherical coordinates, representing hue and chroma (the perceived intensity of the colour, similar to saturation in HSV). Usage of HCL is similarly intuitive as HSV, however, HCL has the distinct advantage of being perceptually linear.

Stauffer et al. (2015) compare colourmaps based on the RGB and HCL models for application in meteorology. They in particular discuss the perceptional non-linearity of the often-encountered RGB

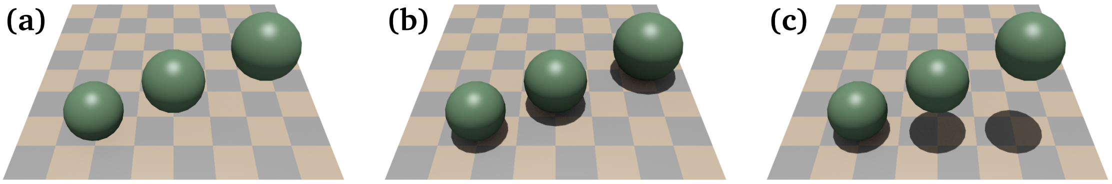


Figure 3.6: Impact of shadows on spatial perception. The positions and sizes of the balls in 2D image space are exactly the same in all three images. However, the different shadows suggest different positions in the perceived 3D space.

"rainbow" colourmap. Figure 3.5b illustrates the issue. While the RGB-based colourmap appears more saturated, its colours exhibit distinct changes in luminance and very inhomogeneous colour gradients. This can lead to unwanted perceived boundaries in visualized data fields, leading to misinterpretations. An HCL-based rainbow, on the other hand, provides colours with constant luminance.

A detailed discussion of colour can be found in Ware (2012). Further important aspects include, for example, colourmaps suited for different types of colour blindness, and the suitability of colourmaps to be printed. Also, Hoffman (1991) discusses the influence of colour on meteorological visualizations, a point elaborated by Hoffman et al. (1993).

Space perception

Particularly important for 3D visualization is *spatial perception*. While for data plotted in two dimensions, space perception in terms of where in horizontal space an object is located is not an issue, judging the position of an object in 3D space is difficult, if not impossible, without sufficient *depth cues* (sources of information about 3D space). Consider the example in Fig. 3.6. The projections to 2D image space of the three spheres are exactly the same in all three images. Without shadows plotted as a depth cue, it is impossible to determine the 3D location of the objects (Fig. 3.6a). In fact, different shadows reveal –despite the spheres being depicted at the same position in image space– different positions in 3D space (Fig. 3.6b and c).

Ware (2012) provides a list of important depth cues, classified into *monocular static cues* (a static 3D image projected onto a 2D medium), *monocular dynamic cues* (the same for animated images), and *binocular cues* (stereo views):

- Monocular static depth cues include *perspective cues* (the convergence of parallel lines to a point, distant objects appearing smaller than close objects of the same size, texture elements becoming smaller with distance), *occlusion* (closer objects occlude objects in the distance), and *depth of focus* (the eye focusses on objects of interest, blurring objects close or far away). Also, *shape-from-shading* (representing diffuse and specular reflection on an object due to lighting) and, as illustrated in Fig. 3.6, *cast shadows* are important depth cues. Artificial spatial cues including specific textures or lines dropped from an object to the ground plane can also improve spatial perception.

- Monocular dynamic depth cues include *motion parallax* (objects close to the virtual camera appear to move faster than distant objects when the camera is moved) and *kinetic depth* (inference of an

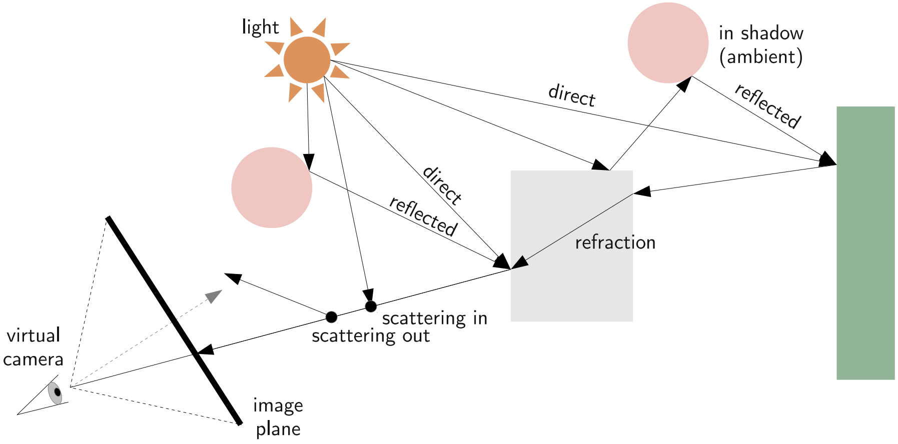


Figure 3.7: Illustration of physical processes of light transport (that is, radiative transfer of the visible wavelengths) that may be simulated by computer graphics algorithms (following Theoharis et al., 2008). Radiation emitted by a light source interacts with objects and fluids in a modelled scene. For example, it can be reflected or refracted at solid objects, or be scattered by molecules and particles in a simulated atmosphere. The simulation of how much light reaches the image plane and can thus be observed by the virtual camera is approximated to different degrees by different computer graphics algorithms.

object's shape by, for example, rotating the object).

- Binocular cues include *eye convergence* (the degree to which both eyes converge when focussing on an object) and *stereoscopic depth* (information about depth provided by slight differences in the images perceived by both eyes).

3.1.4 Basic computer graphics techniques

As noted in the previous section, realistic shading and shadows are important 3D depth cues and contribute to the perceived realism of a depicted scene. In fact, a large part of computer graphics is concerned with the photorealistic depiction of a 3D scene. For such depictions, the physics of interaction of electromagnetic radiation with the scene's objects have to be simulated for the visible wavelengths.

Figure 3.7 illustrates the physical processes that need to be simulated. As noted above for DVR, rendering computations as well bear resemblance to radiative transfer modelling in the atmospheric sciences (e.g. Liou, 2002; Petty, 2006). Rendering in particular includes the numerical solution of the *rendering equation* (RE; similar to the *radiative transfer equation* in the atmospheric literature) to simulate interactions of light with both the scene's objects (reflection, refraction, absorption, emission) and atmosphere (scattering, absorption, emission) (e.g. Akenine-Möller et al., 2008; Theoharis et al., 2008).

Depending on the available computing resources, computer graphics techniques solve the RE to different levels of accuracy. Offline photorealistic rendering attempts to simulate the radiative processes as accurately as possible, many real-time techniques provide only crude approximations. For visualization (in particular interactive real-time visualization), most radiation processes that occur in the real world are not modelled (the visualized objects do not resemble models of real-world objects anyway), and highly simplified illumination models are used. These can be computed quickly while still providing sufficient realism to provide depth cues.

The computer graphics literature distinguishes between *local* and *global illumination* models. Local models consider illumination only locally at a single surface point of an object, thereby neglecting the interaction of light with other objects or with the atmosphere. Given one or more light sources, the surface properties of the point, and the position of a virtual camera observing the scene, the point's illumination can be computed. The most common standard model is the *Phong model*.

Local illumination

The Phong illumination model, proposed by Phong (1975), neglects interaction of light between objects and is not fully based on physics. It estimates the intensity I at a considered surface point as a sum the surface's emission I_e, ambient lighting I_a, and diffuse and specular reflectance I_d and I_s (Theoharis et al., 2008):

$$I = I_e + I_a + I_d + I_s. \tag{3.3}$$

The ambient term compensates for neglecting light interaction between objects and scattering. Without this term, object surfaces not directly illuminated would be black. The diffuse and specular terms depend on the surface properties of the object, as well as on the positions of light source and virtual camera:

$$I_d = I_i k_d cos\Theta = I_i k_d(\vec{n} \cdot \vec{l}) \tag{3.4}$$

and

$$I_s = I_i k_s cos^n \alpha = I_i k_s(\vec{r} \cdot \vec{v})^n, \tag{3.5}$$

where I_i is the incident radiation at the considered point, k_d and k_s are diffuse and specular reflection coefficients, $\vec{n}$ is the normal vector of the object's surface at the considered point, $\vec{l}$ points in the direction of the light source from the point, Θ denotes the angle between $\vec{n}$ and $\vec{l}$, $\vec{r}$ points in the direction of the reflection from the points, $\vec{v}$ points from the point towards the virtual camera, and α denotes the angle between $\vec{r}$ and $\vec{v}$. n denotes an empirical surface shininess.

Extended versions of the Phong model were proposed by, for example, Blinn (1977) and Cook and Torrance (1982), the latter accounting for improved representation of surface properties. Also, further local illuminated models exist that aim at estimating specific terms in the RE more accurately. For further information, the reader is referred to, for example, the books by Theoharis et al. (2008), Akenine-Möller et al. (2008) and references therein.

Shadows

As illustrated in Fig. 3.6, shadows help the eyes to register objects relative to their surroundings. They define the direction of incident light and provide clues for the shape and depth of 3D objects valuable, in particular, in monoscopic depictions. Shadows are formed by the blocking of direct illumination by objects in the scene. The sharpness and size of shadows depends on the size and distance of the light source(s) and on ambient lighting. Shadows consist of an *umbra* (the region in which the light source is fully occluded) and a *penumbra* (partially lit regions). Shadows that have penumbras are called *soft shadows*, shadows with only an umbra are called *hard shadows*. An infinitely distant directional light source (as the sun) casts hard shadows (as in Fig. 3.6).

Offline photorealistic rendering techniques aiming at accurately solving the RE naturally simulate shadows. For real-time rendering, several approaches exist to approximate shadows:

- *Shadow volume* techniques construct a geometrical volume for each object potentially occluding a light source. The volume represents the area in which light is blocked. During evaluation of illumination for a given point on an object's surface, the point drawn needs to be tested for containment in such a volume.

- *Shadow maps* require an additional rendering pass. The scene is first rendered from the point of view of the light source. For each pixel in the resulting image, the distances from light source to object are stored (depth map). During rendering, the point drawn to a particular pixel needs to be transformed to "light source" coordinates and its distance to the light source computed. If the distance is larger than the stored value, the pixel is occluded.

Further information on shadow algorithms can be found in Akenine-Möller et al. (2008), an extensive overview is provided in the book by Eisemann et al. (2012).

Texturing

The evaluation of illumination models (for instance, the Phong model) requires information about surface material properties. If such properties were defined per vertex of the geometrical model of an object, surfaces with high frequency variation would require a large number of vertices. Theoharis et al. (2008) define texturing as "spatially varying one or more of the material attributes of a surface in a predefined manner without affecting the underlying [...] geometry". Attribute patterns are stored in 1D, 2D, or 3D arrays (called *textures*). Texture entries can be scalar values (for instance, to store the diffuse of specular coefficients used in the Phong model) or vectors (for instance, the RGBA components of an image). The represented material attribute is stored in *texture space*, the association between a given surface point and the attribute is called *texture mapping*. Theoharis et al. (2008) provide a thorough introduction to texturing. The basic steps include the computation of coordinates in texture space (*texture coordinates*) for a given surface point, and the usage of a suitable interpolation method to sample the material attribute at the computed texture coordinates.

Global illumination

Global illumination methods aim at simulating all effects of illumination realistically. The "best" approximation to the RE is *stochastic Monte-Carlo raytracing* (e.g. Theoharis et al., 2008), where a large number of random rays approximates the radiative transfer. Since the method is computationally expensive, several approximative techniques have been proposed to speed up the computations. An example is *photon mapping*, where in a first rendering pass "photons" emitted by a light source are "deposited" on surface elements and reused during the rendering pass. A discussion of further real-time methods is provided by Akenine-Möller et al. (2008).

In visualization, isosurfaces, streamtubes and other 3D objects are usually shaded using local illumination models. However, global illumination techniques have been discussed, in particular, for DVR. For example, Lindemann and Ropinski (2011) evaluate the impact of illumination on spatial perception when rendering medical volume data.

Table 3.1: Characteristics of selected consumer-class GPUs current at the time of writing[1]. Recent models possess between a few dozen to up to over 2,000 processing cores and feature a very high memory bandwidth. The models marked with a (*) are used for performance measurements in Chapter 5 (cf. Table 5.4).

setting	model	year	cores	clock	memory	bandwidth	energy
Desktop	Nvidia GeForce GTX 980	2014	2048	1.1 GHz	4 GB	224 GB/s	165 W
Desktop	Nvidia GeForce Titan*	2013	2688	0.8 GHz	6 GB	288 GB/s	250 W
Desktop	AMD R9 290X	2013	2816	1 GHz	max. 8 GB	352 GB/s	290 W
Desktop	Nvidia GeForce GTX 560 Ti*	2011	384	0.8 GHz	2 GB	128 GB/s	170 W
Mobile	Nvidia GeForce GTX 980M	2014	1536	1 GHz	max. 8 GB	160 GB/s	100 W
Mobile	Nvidia Quadro NVS 5400M	2011	96	0.6 GHz	max. 2 GB	28 GB/s	35 W

3.2 Graphics hardware

Primarily driven by applications from the entertainment industry, in particular the commercial market for 3D, high-resolution computer games, graphics processing units (GPUs) have become increasingly powerful in recent years (Nickolls and Dally, 2010). A GPU is a highly parallel processor, with current models possessing between a few dozen to up to over 2,000 processing cores (Table 3.1). Originally designed as special-purpose graphics engines with fixed functionality, GPUs have evolved into fully-programmable processors featuring extremely fast memory (e.g. Owens et al., 2007, 2008). In recent years, the term GPGPU (general-purpose computing on the GPU) has been coined to describe the mapping of arbitrary computations to the GPU. GPUs differ from traditional CPUs by being optimised for high throughput processing of many small parallel operations, compared to low-latency processing of single tasks on CPUs (Fatahalian and Houston, 2008). GPUs are programmed following the single program multiple data (SPMD) paradigm, allowing the processing of many elements (for instance, grid points for fluid simulation or rays casted through a data volume for 3D visualization) in parallel using the same program. This ability to simultaneously use a large number of processing units, paired with the fast memory interface, leads to very high performance in applications where parallelism is abundant.

For interactive visualization systems, both the graphics and the general purpose capabilities of GPUs are of natural interest. The latter, for example, can be used to speed up computations in the data filtering and mapping stages of the visualization pipeline. Current GPUs allow the execution of increasingly complex visualization algorithms at interactive frame rates, that is, a single image can be rendered in a fraction of a second. This enables smooth animations of a scene.

This section introduces fundamental characteristics of GPUs, explains how they can be programmed, and how they can be used for visualization purposes.

3.2.1 The graphics rendering pipeline

GPUs implement a pipeline of fixed function and user programmable operations that compute a rasterised image (usually for screen display) from geometric primitives (typically triangles, lines and points in a 3D coordinate system) and textures. The primitives are shaded (that is, assigned a colour value) and mapped to the screen. Not to be confused with the visualization pipeline discussed in Sect. 3.1, the *graphics rendering pipeline* implements the rendering stage in the visualization pipeline.

[1] Figures obtained from `http://www.geforce.com/hardware`, `http://www.nvidia.com/object/nvs_techspecs.html` and `http://www.amd.com/en-us/products/graphics` (last access April 2015).

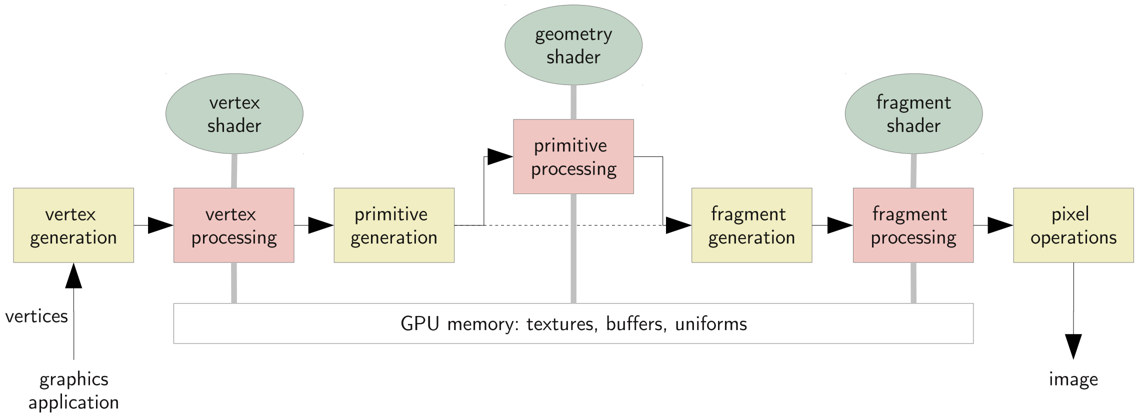


Figure 3.8: Schematic of a simplified GPU graphics pipeline, following Fatahalian and Houston (2008). The graphics application feeds a stream of vertices and associated attributes (for example, colour or normals) into the vertex generation stage. The vertices proceed through the pipeline, which includes fixed function stages (yellow) and user programmable stages (red). Vertices are converted into graphics primitives (points, lines, triangles) and rasterised into fragments that are assembled to form the pixels of the final output image. The user programmable stages can be programmed using shader programs that have access to global textures, buffers and uniform variables in GPU memory. Note that all of vertex, primitive and fragment processing is carried out in parallel. Primitive processing is optional.

The current section provides an overview of the pipeline stages and how they can be programmed. For further information, the reader is referred to the surveys by Fatahalian and Houston (2008), Owens et al. (2008), Nickolls and Dally (2010), Ament et al. (2012), and to the textbooks by Akenine-Möller et al. (2008) and Shirley et al. (2009).

Figure 3.8 schematically illustrates the stages of a simplified graphics rendering pipeline (following Fatahalian and Houston, 2008). The pipeline of current GPUs consists of both programmable stages and of stages with fixed functionality implemented in hardware. The programmable stages, available at different positions in the pipeline, can be programmed with *shader programs*. Historically, the programmable stages were also implemented as fixed functionality, being configurable but not programmable (e.g. Owens et al., 2008). The graphics application controls the execution of the pipeline using a real-time graphics application programming interface (API). Two major APIs exist to program the GPU, *OpenGL*[2] and Microsoft *DirectX*. While OpenGL is platform independent, DirectX is tied to the Windows platform.

The first stage in the pipeline is *vertex generation*. The graphics application inputs a list of vertices and associated information (for example, colour or normals) that describe the geometric primitives of objects to be rendered. Complex 3D graphics scenes can consist of millions of vertices that can be processed independently. Hence, vertex processing is well suited for parallel hardware (e.g. Owens et al., 2008). Vertex generation constructs a stream of records that each contain the data of a single vertex and associated data. In the *vertex processing* stage, a user defined vertex shader program operates independently on each vertex of the input geometry, commonly performing operations including coordinate transforms and the computation of texture coordinates. A vertex shader produces exactly one output vertex from each input record.

Following the vertex stage, the vertices are assembled into a stream of geometric primitives (*primitive generation*). The following programmable stage, *primitive processing*, is optional. OpenGL supports two

[2] `https://www.opengl.org`

Table 3.2: Overview of the evolution of Nvidia GPUs, following Nickolls and Dally (2010). The first GPUs to enter the market provided fixed-function rendering only. Programmable vertex and fragment shader stages appeared in the early 2000s. The first GPUs featuring fully programmable cores that were able to execute any shader stage appeared around 2006.

date	product	transistors	CUDA cores	technology
1997	RIVA 128	3 million	–	3D graphics accelerator
1999	GeForce 256	25 million	–	first GPU (fixed function)
2001	GeForce 3	60 million	–	first programmable vertex shader GPU
2002	GeForce FX	125 million	–	32-bit floating-point programmable GPU (vertex and fragment)
2004	GeForce 6800	222 million	–	
2006	GeForce 8800	681 million	128	first unified graphics and computing GPU
2007	Tesla T8, C870	681 million	128	
2008	GeForce GTX 280	1.4 billion	240	
2008	Tesla T10, S1070	1.4 billion	240	GPU computing clusters, 64-bit floating-point, 4-Gbyte memory
2009	Fermi	3.0 billion	512	

types of shaders for this stage, *geometry shaders* and *tessellation shaders* (of which only the geometry shader is illustrated in Fig. 3.8). Both can generate and discard geometry. The tessellation shader can be used, for example, to change the complexity of the polygonal model of an object for different zoom levels. The geometry shader receives entire graphics primitives and can also output primitives. It can be used, for example, to construct a glyph (for example, a vector arrow, or a wind barb) from a single input point primitive (that is, a single vertex).

After the final geometry has been constructed, fixed hardware functionality is used to clip and cull parts of the geometry that are not visible (for example, primitives that are outside of the viewing region, or that are oriented away from the camera). The remaining geometry is *rasterised* to a set of fragments of the output image (that is, each primitive is sampled in screen space). Each generated fragment contains information about its screen position and its distance from the virtual camera (the fragment depth). Also, attributes associated with the primitives' vertices (for example, colour) are linearly interpolated to the fragments. All these data are accessible in the *fragment processing* stage, the last programmable stage of the pipeline. The fragment shader operates on an individual fragment and has no access to other fragments. This shader can modify the final colour, opacity, and depth value of the fragment. Typical graphics operations implemented in the fragment shader are texturing and lighting.

Finally, the fragments generated by different primitives are assembled into the final pixels of the output image. For instance, multiple semi-transparent fragments may contribute to the same pixel.

3.2.2 Evolution of graphics processing units

Nickolls and Dally (2010) and Owens et al. (2008) trace the development of GPUs from the late 1990s[3]. The first consumer-class desktop GPU that implemented the features found in dedicated high-

[3] My overview follows that of Nickolls and Dally (2010) and is, as is their article, focussed on Nvidia products. Products by AMD (previously ATI) showed similar developments (Owens et al., 2008). Tables that show the development of characteristic figures of both AMD/ATI and Nvidia GPUs are available at `https://en.wikipedia.org/wiki/List_of_AMD_graphics_processing_units` and `https://en.wikipedia.org/wiki/List_of_Nvidia_graphics_processing_units`, respectively (last access April 2015).

end graphics workstations in the 1990s was the Nvidia GeForce 256 in 1999 (Table 3.2). It provided a *fixed function pipeline*, implementing common graphics techniques in hardware that could be configured using OpenGL or DirectX. Programmable shader stages were first introduced with the GeForce 3 in 2001. This GPU featured a vertex processor that could be programmed using assembler (Lindholm et al., 2001). A programmable fragment processor was introduced by ATI (now AMD) with its Radeon 9700 card in 2002, the Nvidia GeForce FX (introduced in 2002) and 6800 (2004) architectures featured both programmable, 32-bit floating point, vertex and fragment processors (Montrym and Moreton, 2005). The processors in these hardware architectures were specialised to the specific task they performed; that is, the vertex processor could not be used for fragment operations and vice versa.

Nvidia's Tesla architecture (GeForce 8800), released in 2006, introduced what Nickolls and Dally (2010) call the "GPU computing era". The architecture featured unified graphics and compute processors that were able to execute any graphics shader stage (vertex, geometry, fragment) and could, in addition to OpenGL and DirectX, also be programmed in C using the new Compute Unified Device Architecture (CUDA) programming model. Also, the architecture traded vector processors, used for GPUs until then, for scalar thread processors better suited for standard programming languages including C (Nickolls and Dally, 2010). Hardware multithreading facilitated the concurrent execution of a large number of threads on these GPU cores (for example, on the GeForce 8800, 12,288 threads could be executed on 128 GPU cores). These novelties made the use of GPUs for general purpose computations feasible. A restriction still valid for current GPU models, however, is that the cores cannot be used entirely independently. The cores are organised in logical units (details dependent on the hardware manufacturer and the specific GPU model) that need to execute the same shader program simultaneously. For example, in Nvidia's *Fermi* architecture (introduced in 2009), 512 GPU cores are organised in 16 units (called *streaming multiprocessors*) of 32 processors each. Any unit of such 32 cores hence executes the same shader program at a given time (Nickolls and Dally, 2010). For further information on GPUs, I refer the reader to the surveys by Owens et al. (2007, 2008), Fatahalian and Houston (2008), and Nickolls and Dally (2010). A detailed description of the Tesla architecture is provided by Lindholm et al. (2008).

3.2.3 GPU programming

In their current versions, both graphics APIs, OpenGL and DirectX, provide CPU functions to, for example, upload data to the GPU and to trigger GPU processing, as well as C-like languages to implement shader programs. OpenGL includes the platform independent OpenGL Shading Language (GLSL), DirectX the Windows-only High Level Shading Language (HLSL). Nvidia also provides a platform independent version of HLSL, called Cg (C for graphics). Met.3D is implemented in OpenGL, hence I focus on OpenGL terminology here.

In previous versions, OpenGL mainly targeted the configuration of the GPU fixed function pipeline. Since version 4.0 (released in 2010), OpenGL deprecates fixed function functionality and is focussed on shader-only usage (although shader programming has been available earlier).

As noted above, vertex, geometry and fragment shaders receive exactly one vertex, primitive, or fragment on which they operate. All programmable shader stages have read access to global variables passed from the CPU (called *uniforms* in OpenGL), to textures (1D, 2D, or 3D arrays of up to four component vectors for RGBA values or arbitrary other data), as well as to buffers (unformatted GPU memory that stores arrays of a variety of data types).

Since the recent OpenGL 4 versions, the options to exchange data between CPU and GPU is becoming increasingly flexible. For example, OpenGL version 4.2 (released in 2011) introduced support to write to textures (which were read-only beforehand) from shader programs. OpenGL 4.3 (released in 2012) added support for uniform arrays of C structs. At the time of writing, OpenGL 4.5 is current (released in 2014). In the future, OpenGL will evolve into the *Vulkan* API[4].

For computations that do not target the graphics pipeline in order to create an image, a number of alternative GPGPU APIs have evolved. At the time of writing, common APIs include the platform independent *OpenCL*, Microsoft's Windows-only *DirectCompute*, and Nvidia's proprietary *CUDA*. Brodtkorb et al. (2013) discuss GPGPU programming strategies and also trace the evolution of further GPGPU programming languages. Also, since version 4.3 (released in 2012), the OpenGL API provides support for *compute shaders*[5]. Compute shaders provide a "lightweight" alternative to using a full GPGPU API and are well suited to execute compute tasks in graphics applications that already use OpenGL (so they do not need to include another API) (e.g. Bailey, 2013).

3.2.4 GPU based real-time visualization

The GPU can be used in several ways to accelerate elements of the visualization pipeline. An obvious application is to use the hardware-accelerated graphics pipeline in the rendering stage to render geometric primitives generated by the mapping stage (for example, line segments output by a marching squares or triangles output by a marching cubes algorithm). GPGPU functionality can be used to accelerate filtering and mapping.

In addition, the data to be visualized can be uploaded to textures in GPU memory and the programmable stages of the graphics pipeline can be used to merge mapping and rendering stages of the visualization pipeline. Straightforward examples of such an approach are outlined by Bailey (2009, 2011), who describes the use of shaders for colour-mapped cutting planes, point clouds, LIC, and terrain mapping directly from 3D scalar data in a texture. For the simple example of visualizing a colour-mapped cutting plane from a 3D scalar field in a regular grid, two triangle primitives describing the cutting plane are input into the graphics pipeline. From the vertices' coordinates, the vertex shader can compute texture coordinates, which are subsequently interpolated by the hardware to the resulting fragments. For each fragment, the fragment shader can use hardware interpolation to interpolate the scalar field to the fragment's position, then use a colour mapping to determine the colour of the fragment.

Ament et al. (2012) elaborate on further techniques, including GPU-accelerated volume rendering and particle-based rendering. Engel et al. (2006) provides a thorough introduction to GPU-based DVR. For example for raycasting, most of the computational work is done in the fragment shader. The geometry of a simple cuboid, covering the data volume to be visualized, is input into the graphics pipeline. For each fragment created from that cuboid, the fragment shader computes a ray from the virtual camera through the fragment and implements the raycasting algorithm (that is, stepping through the volume, interpolating the 3D scalar field in the texture to the sample position, evaluating a transfer function and compositing of the final fragment colour).

An important observation is that such GPU-based visualization algorithms perform extremely well if the scalar datafield is available as a regular grid. In these cases, the GPU's trilinear hardware interpola-

[4] `https://www.khronos.org/vulkan`

[5] `https://www.opengl.org/wiki/Compute_Shader`

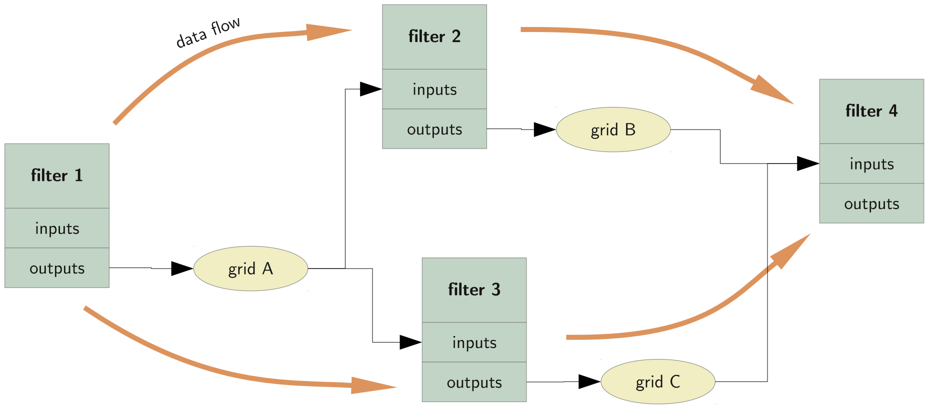


Figure 3.9: Implementation concept for a visualization pipeline, following Telea (2014). Modules from the different stages shown in Fig. 3.1 are implemented as classes in an object-oriented framework. The classes implement interfaces for inputs and outputs and can be connected to form the pipeline structure. In the example shown here, data is generated (or imported) by Filter 1. The resulting data object (grid A) "flows" to Filters 2 and 3, which process the data independently to yield new data objects (grids B and C). These are input into Filter 4, which may yield the final image for user interpretation.

tion can be used to sample the datafield at an arbitrary position in space. If this is not the case (true for most atmospheric models, as outlined in Sect. 2.1.2), application-specific strategies are required.

A large body of literature is available on the usage of GPUs for visualization. For further information, I refer the reader to Bailey (2009, 2011, 2013), Ament et al. (2012), Weiskopf (2006), and Engel et al. (2006). Also, the recent PhD theses by Reichl (2014), Treib (2014) and Fraedrich (2012) provide good starting points for literature on GPU-based visualization.

3.3 Approaches to visualization system design

In scientific visualization, many systems exist whose architecture directly follows the visualization pipeline concept described in Sect. 3.1. In fact, architectures following the concept emerged early in the history of scientific visualization (e.g., Haeberli, 1988; Upson et al., 1989; Dyer, 1990; Lucas et al., 1992). Today, a multitude of visualization systems exist, both commercial and open-source, both general-purpose and highly specialised to a particular purpose. Also, toolkits and libraries exist that provide specific visualization capabilities. In this section, I discuss implementation issues regarding the visualization pipeline and briefly present examples of different types of visualization systems. As an example for an implementation of the visualization pipeline, the *Visualization Toolkit* (VTK; Schroeder et al., 2006) is discussed.

3.3.1 Implementation of the visualization pipeline

From an implementation perspective, a visualization pipeline consists of two main parts: modules that implement algorithms (*sources*, *filters* and *sinks*), and data items (e.g. Telea, 2014). In an object-oriented design, pipeline modules can be implemented as classes derived from a base class that defines an interface with inputs, outputs, and an execution method. If a base class for data items is defined (say `grid`), the pipeline modules can be connected to form a network of modules that share input and output `grid` items. The resulting network resembles a *directed acyclic graph* (DAG; e.g. Sedgewick and Wayne, 2011), that is, all connections between two modules are directed and there are no cycles. However, pipelines can *fan out* (the output of one module is connected to multiple inputs of subsequent modules) and *fan in* (a module's input is connected to multiple outputs). A schematic example is illustrated in Fig. 3.9. When new data is input into the pipeline, the modules need to be executed in *topological order* of the DAG in order to ensure that no operation is executed before its input data items are available. Since this order of execution models a "flow" of data through the pipeline, the design is often called a *dataflow* application model (cf. Telea, 2014).

Despite this conceptual simplicity, Geveci and Schroeder (2011) point out that "actually implementing the pipeline architecture is challenging". As Telea (2014) notes, for practical applications, additions are required for an efficient implementation. Examples include memory management for data items (for example, caching and reference counting), typed inputs and outputs for pipeline modules (to ensure correct behaviour), pipeline traversal methods that minimise the number of executed operations when input data or algorithm parameters change, and parallelisation.

Nevertheless, Moreland (2013) notes that "[...] the simplicity, versatility, and power of visualization pipelines make them the most widely used framework for visualization systems today". He surveys research published on visualization pipelines and discusses main characteristics of different implementations:

Execution A pipeline can be *event driven* and *demand driven*. Event driven pipelines execute as soon as new data is available at the pipeline's sources, *pushing* the data towards the sinks. A demand driven pipeline, in contrast, is executed when a sink demands data from its inputs. A request propagates up the pipeline, and data is *pulled* from module to module. In a demand driven pipeline, caching can be used to avoid repeated computation of results that have not changed since the previous computation. Execution control can be *centralised* or *distributed*. In a centralised implementation, a single unit manages the execution of all modules; in a distributed implementation, each module controls its own execution. Moreland (2013) points out that an advantage of centralised control is that the pipeline can be analysed to optimise execution (for example, with respect to caching and parallel execution). Pipelines may be designed to support *out-of-core streaming* to process datasets that are too large to fit into memory. Here, the dataset is divided into pieces (for instance, by spatial decomposition) which can be processed independently. The pieces are processed sequentially (*streaming*).

Metadata In addition to the actual data flowing through the pipeline, metadata (for instance, information about spatial extent and time) that describe the data are important. Moreland (2013) notes that multiple pipeline passes are required to process datasets with metadata. For example, a source may advertise the spatial extent it can offer. In a first pass, this information flows through the pipeline, with each module potentially modifying the extent. Next, the sink decides which subregion to request. In a second pass, this request propagates back to the source. Finally, in a third pass, the actual data flows through the pipeline to the sink. Moreland (2013) points out difficulties in handling data with a time dimension: "[...] until recently, visualization pipelines operated on data at a single snapshot in time [...]". While animation over time can be implemented by re-executing a pipeline upon time change, algorithms that require multiple time steps (for example, the computation of trajectories) may hit a memory barrier when attempting to load all time steps into memory. Here, Biddiscombe et al. (2007) propose what they call a *continue execution* mode as a kind of temporal streaming. Metadata can enable other types of requests as well. Examples include *prioritised streaming* (stream the "most important" data first) and *query-driven visualization* (the user defines "interesting" characteristics of the data).

Parallelism Different kinds of parallelism can be utilised in a pipeline implementation. *Task parallelism* exploits that modules in a parallel section of the pipelines can be executed independently and hence in parallel. However, Moreland (2013) notes that pipelines in real working environments only rarely contain many independent subpipelines. *Pipeline parallelism* can be exploited when multiple data items sequentially flow through a pipeline. Here, upstream modules can already process new data while downstream modules are still occupied with processing the previous data item. *Data parallelism* partitions a dataset into pieces that can be processed in parallel by multiple instances of the pipeline. Finally, *hybrid parallel* implementations utilise combinations of the three parallelisation types.

Example: VTK

The open-source *Visualization Toolkit*[6] (VTK; Schroeder et al., 2006) is the most widely used visualization library that implements the pipeline concept. Geveci and Schroeder (2011) describe the architecture of VTK. The toolkit provides an extensive array of filters, mapping and rendering modules. VTK is implemented in C++, however, features bindings to a number of interpreted languages including Python, Java, and Tcl/Tk. The library implements a large number of visualization algorithms that operate on different kinds of data grid topologies.

VTK's architecture closely follows the approach outlined above, using three basic types of objects: (1) data objects, (2) objects to process, transform, filter and map data, and (3) objects to execute a pipeline. Algorithms implemented in the VTK pipeline treat input data as immutable, that is, no input data item can be modified. Output is always provided in new data items. The pipeline architecture is *demand driven* by default. When a sink requires data, it "asks" for input. If the input filter already "has" the data, it is returned. Otherwise, the input filter asks its own inputs for the data it requires to perform the computation. This way, a request propagates up the pipeline until a filter either does not require any inputs or has the requested data item available. Computed data items are cached by the generating filter along with their metadata (that is, the filter configuration with which the item was created).

The original VTK architecture allowed only one time step to be processed at once. Biddiscombe et al. (2007) describe time extensions to the implementation, enabling time-dependent algorithms including path line integration and temporal interpolation. This is achieved by adding support for a "temporal dataset" and a corresponding pipeline module that, upon execution, loops the upstream part of the pipeline to process and cache multiple time steps. If a temporal dataset is too large to store all time steps in memory, a "continue execution" mode is introduced to pause algorithms that require only progressive access to time steps.

VTK supports *streaming* by allowing a dataset too large to fit into memory to be decomposed into several pieces that are processed sequentially. Such a decomposition can also be used to build *data parallel* pipelines. VTK supports distributed-memory parallel processing using the message passing interface (MPI), hence, multiple instances of a pipeline can process the parts of a decomposed dataset in parallel.

[6] `http://www.vtk.org`

However, VTK's filters for the most part are *not thread-safe*[7] (although capabilities for multithreading are continuously added[8]). Hence, even on shared-memory systems, multiple instances of a pipeline are required to enable data parallelism.

Recently, Vo et al. (2011) (also see Vo et al., 2010; Vo, 2011; Vo et al., 2012) extended the data parallelism in VTK to task and pipeline parallelism in their Parallel Dataflow Streaming (PDS) framework. To exploit pipeline parallelism, PDS combines demand-driven and event-driven execution models and uses a hybrid push/pull approach to push new data into upstream modules that idle while downstream modules process a pull request (realised by introducing a new pipeline module that pushes data into the pipeline). Task parallelism can be exploited for pipelines that fan in and out and have multiple independent branches. Vo et al. (2011) introduce a scheduler to "supervise" the execution of modules in the pipeline. Modules do not execute immediately but instead put an execution request into a priority queue. The scheduler executes the modules as soon as a thread becomes available.

With respect to rendering, VTK's rendering subsystem for the most part uses the "old" OpenGL fixed-function pipeline. At the time of writing, VTK version 6.1 requires OpenGL version 1.1 (released in 1997), with some functionality requiring OpenGL 2.1[9] (released in 2006). In version 6.2, released in March 2015, the rendering backend is continued to be rewritten to use OpenGL 2.1 features (Hanwell, 2014; Avila et al., 2014).

3.3.2 Alternative architectures

As noted above, Moreland (2013) points out that visualization pipelines are the most widely used framework for visualization systems today. Alternative concepts exist for other types of computer graphics applications. For example, the *scene graph* concept is common in computer aided design and game engines (e.g. Theoharis et al., 2008; Eberly, 2005). A scene graph is focussed on managing the representation of a graphical scene, consisting of nodes that typically represent aggregations of renderable entities (for instance, geometric elements), coordinate transformations, conditional selections, and also computational processes (which can be similar to a module in a visualization pipeline). Similar to VTK for the visualization pipeline concept, several open-source frameworks that implement scene graphs exist. Notable examples include OpenSceneGraph[10] and OpenSG[11]. Only few studies using scene graphs for scientific visualization have been reported, including Fritsch and Kada (2004) and Weiler et al. (2004). The same applies to combined scene-graph-pipeline systems (e.g. Kalkusch and Schmalstieg, 2006; Zehner et al., 2010).

For further information on alternative architectural approaches for the design of graphics applications, I refer the reader to Eberly (2005), Theoharis et al. (2008), Shirley et al. (2009) and Cozzi and Ring (2011). For instance, the latter book discusses issues regarding the design of a graphics engine for virtual globes.

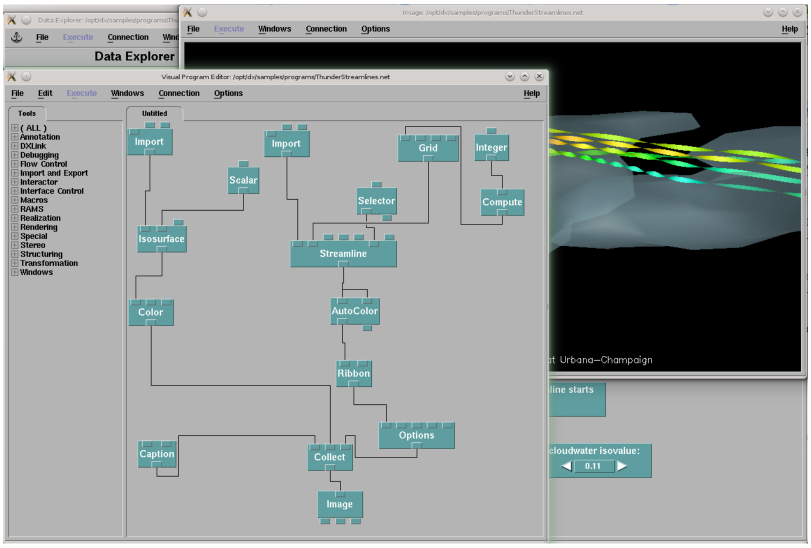


Figure 3.10: Screenshot of the open-source visualization system *OpenDX* (Abram and Treinish, 1995). The software exposes the visualization pipeline to the user in a graphical user interface, so that modules can be connected interactively. The resulting visualizations can be viewed in a separate window.

3.3.3 Examples of visualization systems

Visualization systems are either specific to a given application domain or even task, or general-purpose aiming at being as flexible as possible. I review systems specific to the atmospheric sciences in Chapter 4. With respect to general-purpose software, the visualization pipeline is basis to a number of systems including the *Application Visualization System* (AVS; Upson et al., 1989; Favre and Valle, 2005), *apE* (Dyer, 1990), *IRIS Explorer* (Foulser, 1995; Walton, 2005), *OpenDX* (Abram and Treinish, 1995), *SCIrun* (Parker and Johnson, 1995; Weinstein et al., 2005), and *amira* (Stalling et al., 2005).

Many general-purpose systems expose the pipeline to the user, allowing to choose and connect modules in a graphical user interface. An example is OpenDX, shown in Fig. 3.10. OpenDX is one of the general-purpose tools whose usage in atmospheric science research and also for weather forecasting has been reported in the literature (Treinish and Rothfusz, 1997). In Chapter 4, I provide further details.

A number of visualization applications build on VTK. Here, examples include *ParaView*[12] (Henderson et al., 2004; Ayachit et al., 2012), *VolView*[13], and *VisTrails*[14] (Bavoil et al., 2005; Freire et al., 2011). The three tools follow different usage paradigms. Paraview is an open-source general-purpose visualization tool. It offers all visualization capabilities of VTK, providing not only 2D and 3D scientific visualization techniques for a multitude of grid topologies but also numerous information visualization techniques. Paraview's user interface also exposes the pipeline, the user hence has to be familiar with the pipeline concept. VolView, on the contrary, hides the pipeline from the user. It is an application specialised to provide volume rendering capability, in particular for medical datasets. VisTrails follows a different

[7] VTK FAQ at `http://www.vtk.org/Wiki/VTK/FAQ#Is_VTK_thread-safe_.3F` (last access April 2015).

[8] For example, a new framework for shared memory parallel algorithms is introduced in version 6.1, see `http://www.vtk.org/Wiki/VTK/VTK_SMP` (last access April 2015).

[9] VTK FAQ at `http://www.vtk.org/Wiki/VTK/FAQ#OpenGL_requirements` (last access April 2015).

[10] `http://www.openscenegraph.org`

[11] `http://www.opensg.org`

[12] `http://www.paraview.org`

[13] `http://www.kitware.com/opensource/volview.html`

[14] `http://www.vistrails.org`

approach. It exposes the visualization pipeline as a *workflow* to the user, capturing the steps that the user takes to achieve a data exploration task. This data *provenance* (Freire et al., 2011) approach aims at managing rapidly-evolving workflows in scientific exploration, allowing the user to, for instance, reuse parts of a previous workflow, view differences between workflows and the resulting visualizations.

CHAPTER 4

Visualizing numerical weather predictions – state of the art

After the foundations of numerical ensemble weather prediction and scientific visualization have been introduced in Chapters 2 and 3, this chapter connects both disciplines. I survey previous and recent research and developments in meteorological visualization, and, thus, create the scientific and technological context for the work presented in the remainder of the thesis. Section 4.1 reviews the history of visualization in meteorology from its beginnings in the 1960s until commodity personal computers became powerful enough to run interactive visualization software in the 1990s. Many of the basic techniques presented in Chapter 3 that are relevant for this work have been developed in this time period. Section 4.2 provides an overview of visualization tools that are commonly in use in today's research and operational forecast environments, showing which visualization techniques have found their way into common usage. Finally, recent activities in visualization science that are related to meteorology are discussed in Sect. 4.3. This includes in particular research that is related to ensemble visualization.

4.1 Historical aspects

Traditionally, meteorologists have employed a variety of hand-drawn 2D meteorological charts and diagrams. In his textbook on the principles of meteorological analysis, Saucier (1955) surveys different types of depiction in usage in the 1950s. He classifies the common types into (1) meteorological maps (including different map projections), (2) cross-section charts, (3) sounding charts (including thermodynamic charts and hodograph charts), and (4) time-section charts. These 2D depictions of meteorological observations typically included contour lines, wind vectors, barbs, or streamlines. There was also an early desire to depict atmospheric measurements in three dimensions. Saucier (1955, Ch. 2) notes:

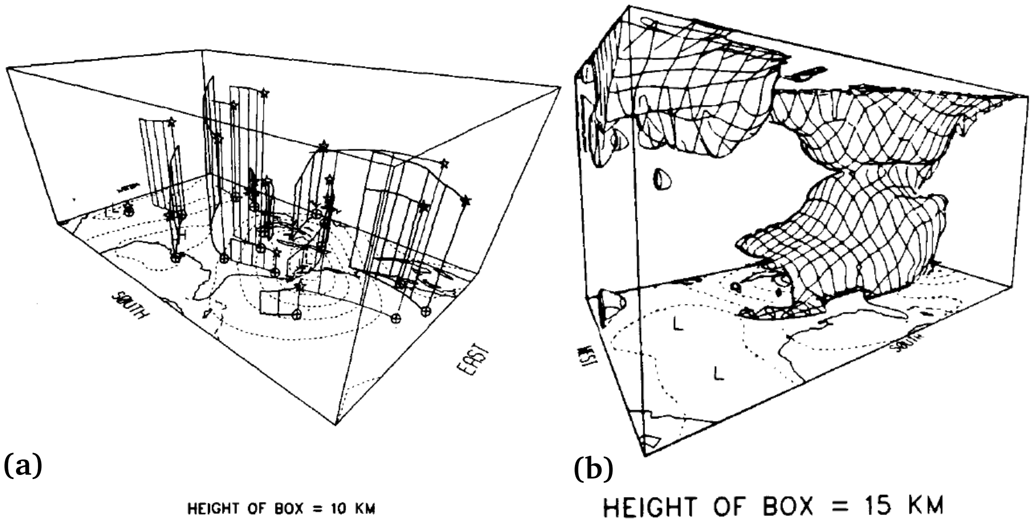


Figure 4.1: Images from a computer generated 3D animation created by Grotjahn and Chervin (1984). **(a)** Simulated particle trajectories. **(b)** Wireframe isosurface of potential temperature gradient, illustrating a frontal zone. The animation was produced on film. (Reprinted from Grotjahn and Chervin (1984), © American Meteorological Society. Used with permission.)

> “Indeed, a large part of [the meteorologist’s] work is aimed at developing ability to deduce three-dimensional structure from analysis of a few level charts.” [...] “A three-dimensional representation [of the state of the atmosphere] drawn to scale would be the most accurate and natural form, but it would present numerous difficulties in plotting and analysis.”

One early approach was proposed by Carlin (1954), who suggested to draw multiple pressure level charts in a way they can be viewed with a 3D effect in a stereoscope.

4.1.1 Early visualization approaches

The development of the first computer-based visualization techniques for meteorology is closely tied to the development of computer hardware. With the advent of the early atmospheric circulation models, the first automated methods to plot simulation data were introduced to the atmospheric sciences. As reported by Papathomas et al. (1988), the earliest tool specific to meteorology was the NCAR Graphics package developed in the late 1960s. As a notable example, Washington et al. (1968) presented 2D contour lines of simulation data from the NCAR general circulation model (Kasahara and Washington, 1967; Kasahara, 1968) displayed on a cathode ray tube (CRT) screen. They used an early variant of the marching squares algorithm (cf. Sect. 3.1.2).

In the 1970s, computer generated *animation* of simulation and observation data generally became a feasible tool for science research (Robinson, 1978), and also the first computer animated movies of the time evolution of atmospheric simulations were created. Grotjahn and Chervin (1984) describe the creation of such movies at NCAR (created with the –still monochrome– NCAR Graphics package). They already used 3D perspective views of black and white contours. Figure 4.1 shows examples of particle trajectories computed by a simulation and of a wireframe isosurface of the potential temperature gradient, indicating a frontal zone. The movies were produced on 35 mm and 16 mm film. Grotjahn and Chervin (1984) note that at the time of their paper, over 600 movies had been produced at NCAR. At the same time, interest in “true” 3D displays grew and methods were developed to generate stereo-

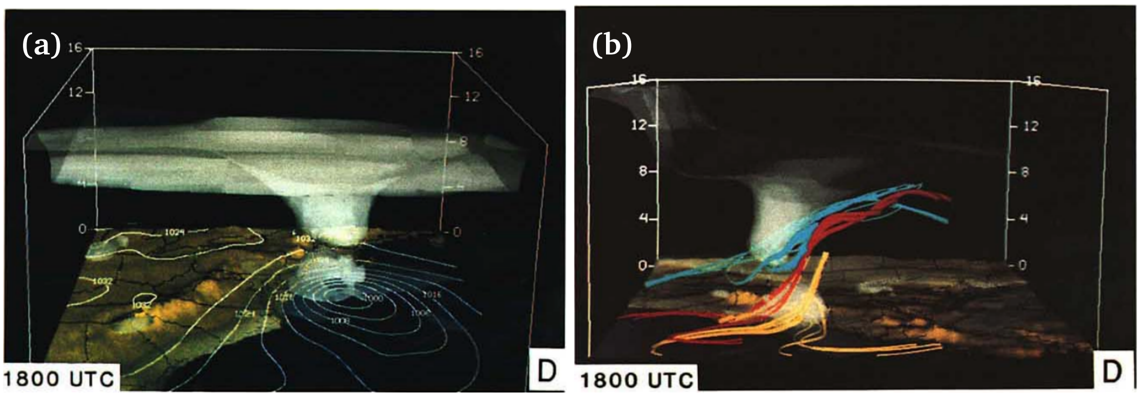


Figure 4.2: Images from an animation generated with the 4D McIDAS system applied to a model simulation of the "Presidents' Day Cyclone" (Hibbard et al., 1989). **(a)** 2 PVU isosurface of potential vorticity (showing the dynamic tropopause), rendered above contour lines of the surface pressure field. The tropopause fold above the low pressure core is clearly visible. **(b)** Trajectories, coloured according to their region of origin. Note how the trajectories capture conveyor belt structures. (Reprinted from Hibbard et al. (1989), © American Meteorological Society. Used with permission.)

scopic projections, first of observational (mainly satellite) data (Hasler, 1981; Hasler et al., 1981, 1985; Hibbard, 1986), but also of simulation data (Papathomas et al., 1987).

At the university of Wisconsin-Madison, the *Man computer Interactive Data Access System* (McIDAS)[1] had been developed since 1973 (Smith, 1975; Suomi et al., 1983; Lazzara et al., 1999). McIDAS was a workstation system originally designed to process and view meteorological satellite and conventional observation data. In the mid-1980s, a stereo terminal was developed for the system to facilitate stereographic views. Hibbard (1986, 1987) reports on extensive experiments with monochrome 3D stereo visualization of observational data obtained from the McIDAS system. In the 1980s, high attention was given to psychophysical aspects, specifically visual perception (cf. Papathomas et al., 1987, 1988; Schiavone and Papathomas, 1990). In this line, Hibbard (1986) discusses a number of challenges of 3D visualization and perception, including the correct usage of visual clues to create an illusion of depth, choosing a good aspect ratio to avoid misleading angles and slopes in the display, system performance and user handling. He presents 3D views of satellite cloud images, of wind trajectories and streamlines, of contour surfaces, and of radar data. Performance measures are given for the response time of the system (that is, the time from a user request to a new image), which, however, are still on the order of minutes. Hibbard (1986) notes that "while the displays presented in [his] paper create a striking four-dimensional illusion when viewed with stereo and animation, they are still a long way from reaching the potential of four-dimensional display for meteorologists". For improvement, he suggests the combined display of multiple variables, the application of the system to model simulation data, the improvement of spatial perception (the "location problem" as he calls it), the use of colour, and improving efficiency for better interactivity. In a similar effort, Von der Haar et al. (1988) present 3D displays for time-varying meteorological observation data. They create 3D models of satellite and radar data, visualized as isosurfaces and wireframe models, and discuss application of their visualizations to pilot briefing, forecasting and research, and teaching. Notably, they propose a "fly-through" option in which an animation is generated in which the user "flies" through reconstructed clouds.

[1] https://www.ssec.wisc.edu/mcidas

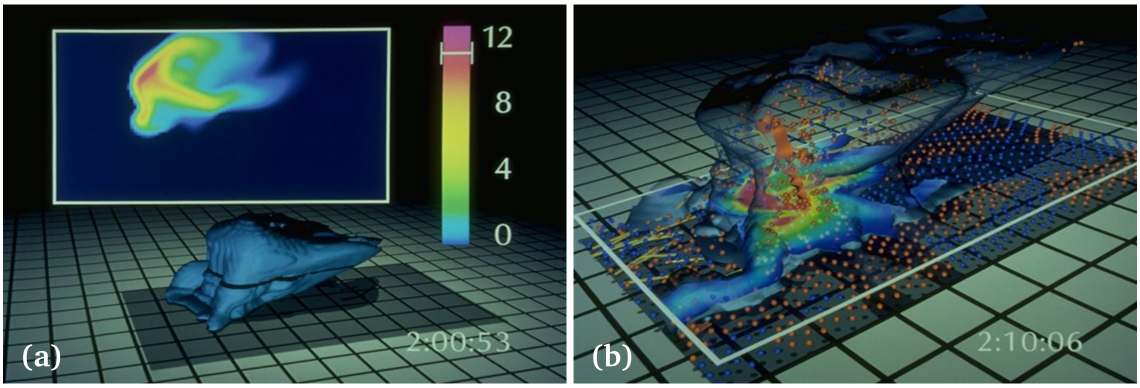


Figure 4.3: Images from an award-winning movie of a developing thunderstorm created by (Wilhelmson et al., 1990). The movie took 11 months to complete (see text for details). (From Wilhelmson et al. (1990), © 1990 SAGE Publications. Colour version of the images obtained from `http://severewx.atmos.uiuc.edu/04/online.4.4.super.html`. Used with permission.)

McIDAS, still running on large mainframe computers, was extended[2] to handle model simulation data and and multiple simultaneous parameters in colour (Hibbard and Santek, 1989a). Hibbard et al. (1989) describe the application of McIDAS to a model study of the 1979 "Presidents' day cyclone". In addition to the techniques for observations presented by Hibbard (1986), they use isosurfaces of potential vorticity to depict the tropopause on top of a topographic map and contour lines of surface pressure. Particle trajectories are rendered as shaded tubes. Figure 4.2 shows examples from the case study, showing a tropopause fold and trajectories depicting the cyclone's conveyor belts. Describing their effort to create the animations, Hibbard et al. (1989) stress the need for an interactive system to create such visualizations, as "[...] satisfying [the] goals required many trial and error adjustments [...]", which were "[...] very time consuming, usually requiring several hours to see the result of even a small change." Hibbard and Santek (1989b) elaborate on this point and describe their intent to construct such a system.

Also in the late 1980s, Robert Wilhelmson and colleagues raised attention[3] with story-boarded 3D animation movies of numerically modelled storms (Wilhelmson et al., 1990). Figure 4.3 shows two images from the video "Study of a Numerically Modeled Severe Storm"[4] created in 1989. Creating the movie was a major undertaking. Wilhelmson et al. (1990) report:

> "Over the 11-month period, from the first meeting to the completion of the final version of the video, four scientific animators worked on the project for approximately 1 man-year. In addition, script writers, artistic consultants, and postproduction personnel were involved. The final version took about 200 hr of computer time on a Silicon Graphics Iris 4D/240 GTX using Wavefront software and special software written by the visualization group."

Further details on visualization activities up to the late 1980s can be found in the surveys by Papathomas et al. (1988) and Schiavone and Papathomas (1990). They review visualization activities in

[2] The work took place in the context of a four-year NASA project, whose final report is available at `http://ntrs.nasa.gov/search.jsp?R=19910008266` (last access April 2015).

[3] As noted by Middleton et al. (2005), the video "Study of a Numerically Modeled Severe Storm" (1989) won a number of awards and was nominated for the National Academy's Short Film–Animation Category Award. It also appeared as cover image on Edward Tufte's book on "Visual Explanations" (Tufte, 1997)

[4] Excerpts from the video are available at `https://www.mediaport.net/CP/CyberScience/BDD/fich_050.en.html` and `http://www.fondation-langlois.org/html/e/media.php?NumObjet=13230` (last access April 2015).

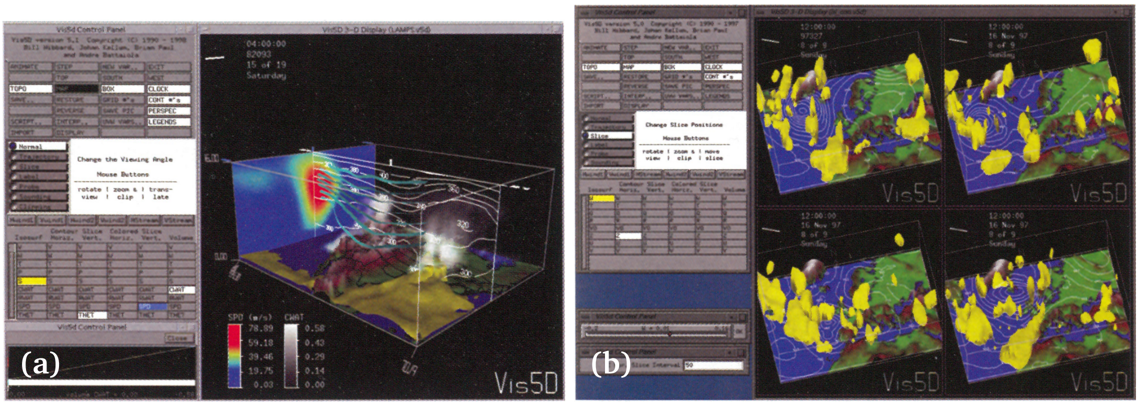


Figure 4.4: Two screenshots of the Vis5D visualization tool. **(a)** A display combining different visualization types available in the software: 2D contour lines, 2D colour mapping, terrain, isosurface, volume rendering. **(b)** Four members of an ECMWF ensemble forecast. (Reprinted from Hibbard (2005), © 2005, with permission from Elsevier.)

meteorology, the application of computer graphics to the atmospheric sciences, and related developments in computer graphics up to the late 1980s, with a focus on activities in the United States. Also, Middleton et al. (2005) provide remarks on the history of meteorological visualization.

Papathomas et al. (1988) note further visualization activities in the US, including the NASA AOIPS workstation and the NOAA PROFS workstation. Notably, the NASA effort was also featuring a stereo display, while NOAA focussed primarily on using 2D graphics to display multiple meteorological variables simultaneously (Papathomas et al., 1988). Schiavone and Papathomas (1990) survey the literature from the perspective of how depth-perception cues, motion-perception cues and value-perception cues have been applied for the 3D visualization of meteorological data using particles, surfaces, volumes, and volumetric animation. They describe challenges and potentials of the methods, noting, for instance, work on the visualization of clouds (Kajiya and Von Herzen, 1984; Gelberg and Stephenson, 1987), on the usage of particle systems for animating the atmospheric flow (Papathomas et al., 1987), and on techniques for motion animation of satellite observations (Warnecke, 1987). Both Papathomas et al. (1988) and Schiavone and Papathomas (1990) make strong statements for interdisciplinary work in meteorology and computer graphics, stating that "the future holds exciting prospects for computer graphics applications in meteorology" (Papathomas et al., 1988).

4.1.2 Interactive workstations

Since around 1990, workstations with increasingly powerful graphics performance appeared (Middleton et al., 2005; Hibbard, 2005), leading to a number of open-source as well as commercial visualization tools. Based on their initiative to create an interactive McIDAS system (Hibbard and Santek, 1989b), Bill Hibbard and coworkers developed the *Vis5D* software (Hibbard and Santek, 1990; Hibbard et al., 1994, 1996; Hibbard, 2005). In its first versions still requiring a dedicated graphics workstation (Hibbard and Santek, 1990), the tool later was able to run on standard Linux computers. It became a major 3D visualization tool in meteorology in subsequent years (Hibbard, 2005; Middleton et al., 2005). For instance, Vis5D was used at DKRZ (Böttinger et al., 1998), coupled to the ECMWF Metview software (Daabeck and Hibbard, 2003), and used as basis for a 3D forecasting workstation at NOAA (McCaslin et al., 2000, cf. next section). Figure 4.4 shows two screenshots of Vis5D. Hibbard (2005) describes the functionality of the last Vis5D release. The tool supports model simulation data on regular, time-varying grids with multiple simulation parameters (in the vertical, irregular pressure levels are supported). Data can be displayed as 2D contour lines or pseudo-colours on horizontal and vertical sections, as 3D isosurfaces, and as volume rendering. Wind data can be depicted as vector glyphs or streamlines (cf. Fig. 4.4a). A topographical map can be displayed as geo-reference. Also, particle trajectories can be computed on the data and vertical soundings can be generated. Vis5D provides support for comparing multiple datasets. This functionality can be used, for example, to compare ensemble members (cf. Fig. 4.4b). Multiple datasets can be attached to the same or multiple displays. If multiple displays are present, they can be "grouped" and synchronised. Hibbard (2005) points out that all rendering techniques are interactive. The complete list of functionality can be found in the user guide[5]. The development of Vis5D ceased in the early 2000s (Hibbard, 2005), with the last release version dating to November 2001[6].

A number of other 3D visualization tools appeared in the 1990s, mostly general-purpose, commercial, and not primarily targeted at meteorology. A number of systems used in the atmospheric sciences are listed by Schröder (1997), Böttinger et al. (1998) and Middleton et al. (2005). They include the commercial systems (cf. Sect. 3.3) *Application Visualization System*[7] (AVS; Upson et al., 1989; Favre and Valle, 2005), *Iris Explorer*[8] (Walton, 2005), the *IBM Data Explorer* (DX; Abram and Treinish, 1995; Watson, 1995) (later renamed to *OpenDX* and made open-source[9], discontinued in 2007), and *amira* (Stalling et al., 2005) (now *Avizo*). Schröder (1997) lists a number of further systems and evaluates their usability for meteorological data.

However, these tools were primarily used by visualization specialists, as Böttinger et al. (1998) and Middleton et al. (2005) point out. Atmospheric scientists in their daily work relied mainly on command-line driven 2D plotting and analysis tools including the *Grid Analysis and Display System*[10] (GrADS), or the commercial *Interactive Data Language* (IDL)[11]. Böttinger et al. (1998) describe the visualization activities at DKRZ in the 1990s. In addition to 2D analysis and visualization software, Vis5D and AVS were used.

4.2 State of the art in meteorological visualization software

This section provides an overview of visualization software tools that are in common use at the time of writing. Papathomas et al. (1988) distinguish visualization tools in meteorology with respect to application in a research setting, application in an operational forecast setting, and application for an end-user (for instance, the general public). In the following, I focus on the first two (most relevant for this thesis) categories. As Koppert et al. (1998) point out, a tool in an operational setting should offer techniques tailored to the specific forecasting task and not confuse the forecaster with large sets of parameters that need to be configured. A research setting, on the other hand, demands a tool that is flexible to adapt to different exploration tasks and data formats. Its visualizations should be highly configurable by the user.

[5] Available online at `http://vis5d.sourceforge.net` (last access April 2015).

[6] Vis5d+ release notes, available at `http://vis5d.sourceforge.net/NEWS.html` (last access April 2015).

[7] `http://www.avs.com`

[8] `http://www.nag.com/Welcome_iec.asp`

[9] `http://www.opendx.org`

[10] `http://grads.iges.org/grads`

[11] `http://www.exelisvis.com/ProductsServices/IDL.aspx`

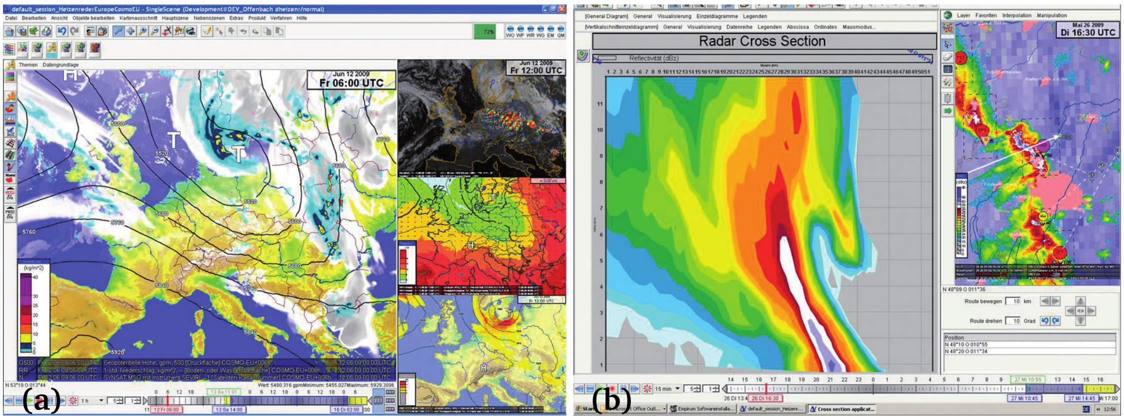


Figure 4.5: Screenshots of the NinJo forecasting workstation used by the meteorological services of Germany, Switzerland, Denmark and Canada. **(a)** Main interface of the software, featuring multiple views that offer a variety of 2D visualization methods to depict observations and numerical prediction data. **(b)** Example of a vertical section generated by the system. (Reprinted from Heizenrieder and Haucke (2009), © Deutscher Wetterdienst 2010. Used with permission from DWD).

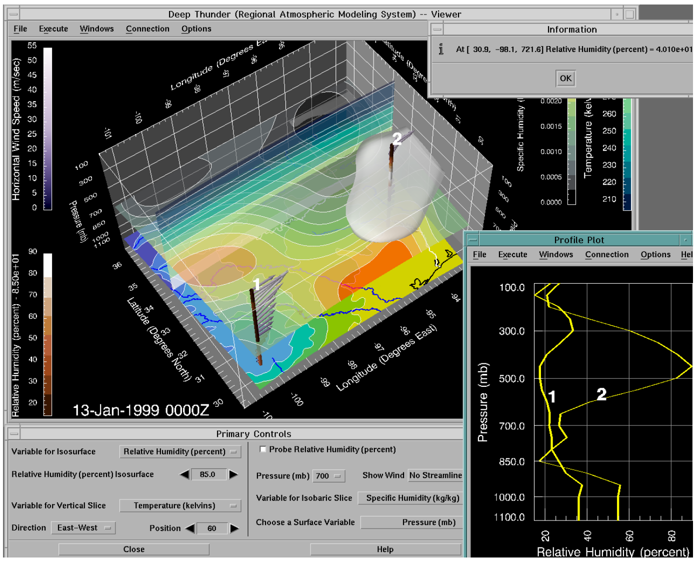
In this section, recent examples from both forecast and research settings are presented. In addition, attention is given to visualization tools that have been employed for research flight planning, the primary application case in this thesis. As stated above, atmospheric scientists working with numerical model data mainly rely on command-line based, scriptable analysis and (for the most part) 2D visualization tools in their practical daily work. This has –from my experience– not changed since the reports by Middleton et al. (2005) and Böttinger et al. (1998). A major reason why 2D methods are often preferred is that they are well suited to convey quantitative information, as noted by Middleton et al. (2005). 2D contour lines and colour mappings can be used to convey a large range of data values. In a 3D depiction, only a small number of isosurfaces can be displayed without cluttering and occlusion. However, a 3D image is able to convey spatial structure in all three dimensions, a distinct advantage compared to 2D methods. On the downside, as discussed in Sect. 3.1.3, spatial perception is more challenging in 3D. Determining the location of a data feature displayed in a 2D image is usually not an issue. In a 3D projection, achieving good spatial perception can be difficult. Major influencing factors are, for example, shadows (Wanger et al., 1992) and illumination models (e.g. Weigle and Banks, 2008; Lindemann and Ropinski, 2011, and references therein).

However, to put the development of Met.3D into the context of existing tools, the following sections focus on approaches that use 3D visualization.

4.2.1 Visualization in forecast environments

In particular in forecasting, 2D visualization systems prevail at the time of writing. Most national weather services employ meteorological workstation systems with extensive functionality to plot maps, cross sections, and a variety of meteorological diagrams. Functionality provided by these systems is very similar. As an example, the *NinJo* workstation[12] is a Java-based forecasting workstation developed and used by the meteorological services of Germany, Switzerland, Denmark and Canada. NinJo development started in 1999 (Koppert, 2001), details of the system are described by Heizenrieder and Haucke (2009).

[12] http://ninjo-workstation.com


Deep Thunder (Regional Atmospheric Modeling System) -- Viewer
File Execute Windows Connection Options
Information
OK
Horizontal Wind Speed (m/sec)
Relative Humidity (percent) - 8.50e+01
Longitude (Degrees East)
Latitude (Degrees North)
Pressure (mb)
Specific Humidity
Temperature (kelvins)
13-Jan-1999 0000Z
Primary Controls
Close
Help
Profile Plot
File Execute Windows Connection Options Help
Pressure (mb)
Relative Humidity (percent)
1
2

At [30.9, -98.1, 721.6] Relative Humidity (percent) = 4.010e+01

Variable for Isosurface: Relative Humidity (percent)

Relative Humidity (percent) Isosurface: 85.0

Variable for Vertical Slice: Temperature (kelvins)

Direction: East-West Position: 60

□ Probe Relative Humidity (percent)

Pressure (mb): 700 Show Wind: No Streamline

Variable for Isobaric Slice: Specific Humidity (kg/kg)

Choose a Surface Variable: Pressure (mb)

Figure 4.6: Screenshot of the 3D forecasting tool presented by Treinish (1996), Treinish and Rothfusz (1997) and Treinish (1998). The tool was based on the then-commerical IBM Data Explorer. (Reprinted from Treinish (1998), © 1998 IEEE. Used with permission.)

The software is based on 2D visualization methods. It provides support for observation data and for NWP models in use at the mentioned weather services. Figure 4.5 shows screenshots of the system. Emphasis is put on multiple views that simultaneously display different forecast parameters. NinJo provides a sophisticated time navigation, and meteorological charts such as vertical soundings or time series can be displayed and analysed interactively.

A number of further workstation systems are similar to NinJo. Daabeck (2005) surveys operational workstations in use in Europe as of 2005, a few of which still exist today. In 2015, systems in use include: AWIPS-II[13] is an also Java-based workstation system developed by the U. S. National Weather Service. In contrast to NinJo, AWIPS-II is available as open-source (at the time of writing, a beta-version is available). Diana[14] (Bergholt, 2008) is the workstation system of the Norwegian Meteorological Institute (met.no), providing similar 2D functionality as NinJo and AWIPS-II. As AWIPS-II, Diana is available as open-source. Synergie[15] is the workstation system of Météo-France. It is also used widely throughout African countries. VisualWeather[16] is a commercial workstation system used throughout the world.

The few reports on the usage of 3D visualization in forecasting date to the 1990s and early 2000s. They are described here in more detail.

- Treinish (1996), Treinish and Rothfusz (1997) and Treinish (1998) reported on experiments with 3D visualization for local forecasting during the 1996 Olympic Games in Atlanta. The project's rationale was that the application of a high-resolution local NWP model by the U.S. NWS to forecasting during the Olympic Games required a new approach to visualizing the forecast data (Treinish, 1996; Snook et al., 1998). In a joint effort between IBM, the NWS and NOAA, a forecast visualization tool based on the IBM Data Explorer (cf. Sects. 3.3 and 4.1, as well as Fig. 3.10;

[13] http://www.unidata.ucar.edu/software/awips2
[14] https://diana.wiki.met.no
[15] http://www.mfi.fr/en/page/meteorological-information-systems/synergie-forecasting.php
[16] http://www.iblsoft.com/products/visualweather

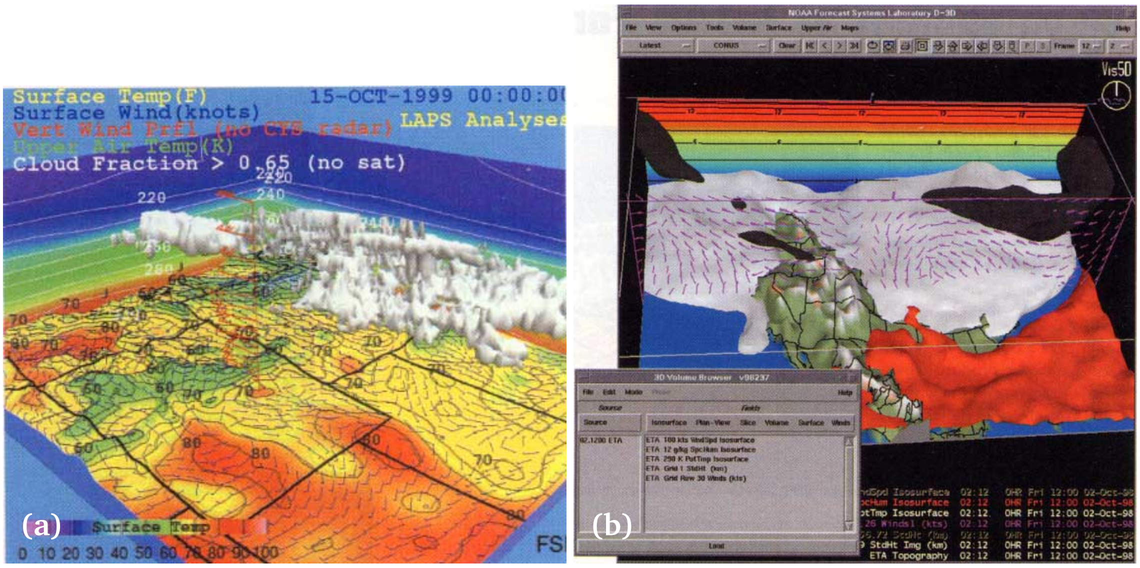

Surface Temp (F)
Surface Wind(knots)
Vert Wind Prfl (no CTS radar)
Upper Air Temp(K)
Cloud Fraction > 0.65 (no sat)
15-OCT-1999 00:00:0
LAPS Analyses
(a)
Surface Temp
0 10 20 30 40 50 60 70 80 90 100
NOAA Forecast Systems Laboratory D-3D
Vis5D
3D Volume Browser v98237
File Edit Mode Help
Source
Fields
Isosurface Plan-View Slice Volume Surface Winds
02.1200 ETA
ETA 100 kts WndSpd Isosurface
ETA 12 g/kg SpcHum Isosurface
ETA 290 K PotTmp Isosurface
ETA Grid 1 StdHt (km)
ETA Grid Rew 30 Winds (kts)
Load
(b)
ETA Topography 02:12 0HR Fri 12:00 02-Oct-98

Figure 4.7: Screenshots of the D3D forecasting tool built in the late 1990s at the U.S. Forecast Systems Laboratory, as presented by McCaslin et al. (2000). The tool was based on Vis5D (cf. Fig. 4.4), however, featured a different user interface that matched the interface of the 2D AWIPS D2D software in use at the National Weather Service Weather Forecast Offices. (Reprinted from McCaslin et al. (2000), `http://dx.doi.org/10.1145/563788.604449`. Figure and article are in the public domain.).

Abram and Treinish, 1995) was designed. Using the DX capabilities, the tool offered similar visualization functions as Vis5D (that is, including 3D isosurfaces and volume rendering, 2D filled and line contours, wind vectors, a probe for vertical profiles). Figure 4.6 shows an example screenshot of the tool. Functionality for different tasks of visual analysis of the forecast data were separated into different "classes" of sub-tools, each featuring specialised methods for either data exploration, analysis, and communication (Treinish, 1998). Treinish (1998) describes a typical workflow of the system, focussed on first interacting with the visualization to select a suitable combination of forecast variables, then creating a time animation of the selected scene. Treinish and Rothfusz (1997) concluded that an advantage of their 3D methods was

> "[...] that they virtually eliminated the need to laboriously evaluate numerous two-dimensional images [..] by presenting all the relevant information to the forecaster in an easy-to-interpret, four-dimensional display. Conceptual models that would normally require inference from a significant amount of two-dimensional data (e.g., the horizontal extent of cloud dissipation in the lee of the Appalachian mountains) were immediately obvious in three-dimensional animations. Originally intended for media displays, these applications quickly gained favor by forecasters as valuable operational forecasting tools."

However, (Treinish, 1998) and Treinish and Rothfusz (1997) also noted a lack of interactivity due to limitations in computational performance. For example, they stated bottlenecks in the creation of animations (requiring up to 30 minutes of wall clock time) and in reading the forecast data into memory.

- Schröder (1997), Lux and Frühauf (1998) and Koppert et al. (1998) presented *RASSIN* (named *VISUAL* within DWD), a 3D forecasting system for usage within the DWD. The system was based on graphics implementations from the Fraunhofer Institut für Graphische Datenverarbeitung. Data access was provided by the GRIB database of DWD (Schröder, 1997). Focus was put on visualizing model data from the DWD "Europamodell" (one of the DWD NWP models in use at that time) directly from the rotated model grid with terrain-following vertical coordinates (cf. Sect. 2.1.2). Similar to the approach by Treinish and Rothfusz (1997), RASSIN provided functionality to display 2D sections and 3D isosurfaces (calculated using a Marching-Cubes implementation that was modified to operate on the model grid; Lux and Frühauf, 1998). Discussing their experience with an operational test of the software, Koppert et al. (1998), too, point out the importance of system performance for user acceptance. They furthermore highlight the need for common concepts of operations (user interface and workflow) when forecasters are asked to transition from a 2D to a 3D environment. On the same software basis, the 3D system *TriVis* for usage in the media was developed (Schröder, 2000; Haase et al., 1999, 2000). This system is still actively developed and commercially marketed[17].

- McCaslin et al. (2000) and Szoke et al. (2002) presented *D3D*, a 3D software built at the U.S. Forecast Systems Laboratory (FSL) on top of the *Vis5D* tool. For D3D, the Vis5D user interface was rewritten to match that of the 2D *AWIPS D2D* software in use at the National Weather Service Weather Forecast Offices (WFOs). Szoke et al. (2002) provide an overview of the tool's functionality. D3D provided the most extensive array of visualization methods of the three approaches discussed in this section. Visualization techniques included 3D isosurfaces and volume rendering, 2D horizontal and vertical sections, vertical soundings and data probes, as well as trajectories (for the latter also see Barjenbruch et al., 2002). Figure 4.7 shows example screenshots of the system. D3D accessed forecast data through the general AWIPS database.

  Notably, "real-time forecast exercises" were conducted to evaluate the value of 3D visualization, and the software was installed at a number of WFOs. A few case studies were presented, including usage of D3D for the examination of tropical cyclones (Watson et al., 2002), the usage of 3D trajectories (Barjenbruch et al., 2002), and the analysis of the synoptic situation during a tornado outbreak (Nietfeld, 2003). Szoke et al. (2003) report on experiences gained with the system. They discuss the reluctance of forecasters to switch from 2D to 3D, but also confidently state that for forecasters trained with D3D it is "hard to deny that examining the atmosphere using a 3D tool is not more effective and complete than using 2D displays". Szoke et al. controversially discuss the value of D3D for forecasting. For example, with respect to 3D isosurfaces, they positively note that

  > "[...] a general consensus was that [3D isosurfaces] should allow operational forecasters to better focus on the important aspects of a forecast more quickly and with less chance of missing some critical feature because they failed to explore the 'correct' 2-D level [...] or the most appropriate cross section [...]",

  but also that

  > "[...] it would take some time for forecasters to determine how best to use isosurfaces. Clearly the atmosphere is three-dimensional, and the isosurface allows one to view it as such, but it was still difficult for the forecasters to overcome the tendency to want to view fields in the traditional 2-D manner."

[17] http://www.askvisual.de

> Szoke et al. (2003) also point out problems with spatial perception in the 3D views. They approached the issue with a switch in D3D that allowed the forecaster to toggle an overhead view, as well as with a vertically movable background map that could be elevated to the height of an isosurface. According to Szoke et al., forecasters involved in the 3D forecast exercises agreed that there needs to be training in how to best use 3D depiction in forecasting. The authors suggest to teach university courses with 3D visualization, in order to make the next generation of meteorologists familiar with the concepts. Szoke et al. also positively report on the interactivity introduced by their system. Interactively moveable vertical soundings and cross sections, for example, were very well perceived by the forecasters.
>
> There was also an approach to ensemble visualization with D3D. Alpert (2003) suggest to interpret the ensemble dimension as the vertical coordinate in Vis5D and to view a 2D map of an ensemble product as a 3D isosurface. Unfortunately, no figures are available from this approach. For further information about the D3D project (including sample animations), the reader is referred to the D3D website[18].

An extensive further source of information on both recent and historical visualization aspects in operational forecasting are the proceedings of the ECMWF workshops on meteorological operational forecasting, taking place biannually since 1987. The proceedings are available from the ECMWF website[19]. For example, developments in recent years have focussed on 2D visualization based on Open Geospatial Consortium (OGC) web services. Notably, the *Web Map Service* (WMS; de la Beaujardiere, 2002, 2006) is explored as a standardised way to transmit only small images instead of the large simulation output. At the time of writing, most major workstations have implemented a WMS interface (e.g. Matula, 2009; Eymann, 2010; Kertész et al., 2010). Also, ECMWF is developing a fully web-based 2D workstation, *ecCharts*, to provide forecast products to the Centre's member states' weather services (Raoult et al., 2010; Lamy-Thépaut et al., 2013). In this respect, further information on web-service-based visualization is available from the website of the OGC MetOcean domain working group[20].

4.2.2 Visualization in research environments

This section reviews tools commonly used by atmospheric scientists at the time of writing in *research settings*. 2D data analysis and visualization tools are reported, however, as in the previous section, more detail is provided for 3D applications. For alternative overviews of visualization systems used in research (both 2D and 3D), the reader is referred to Middleton et al. (2005) and Nocke (2007).

As stated above for forecast environments, 2D web-based visualization has gained increased interest in recent years, including WMS based implementations (e.g. Blower et al., 2013) and web-based forecast exploration tools for teaching (e.g. Limbach et al., 2014). Here, I focus on analysis and visualization tools used by atmospheric scientists to analyse their "custom" datasets. 2D data analysis and visualization tools typically provide functions for data import and remapping, statistical analysis, and plotting. Typical tasks when working with output of numerical weather prediction models in research are summarized by Warner (2011, Ch. 11), including the calculation of derived variables, tracking of

[18] D3D website: `http://laps.noaa.gov/d3d` (last access April 2015).

[19] Articles are available in the ECMWF archive: `http://old.ecmwf.int/publications/library/do/references/list/162` (last access April 2015).

[20] Available at `http://external.opengeospatial.org/twiki_public/MetOceanDWG/WebHome` (last access April 2015).

air parcels, and the analysis of patterns, model spectra, and specific diagnostics. Plotting is usually done with standard 2D techniques, however, most systems in use today also offer some (restricted) 3D visualization support. The functionality offered by the various tools overlaps widely, as also pointed out by Middleton et al. (2005). Nevertheless, every approach has its own strengths and weaknesses. Common tools include:

- *NCAR Graphics*[21], along with the *NCAR Command Language*[22] (NCL), features an extensive array of 2D visualization techniques for meteorological maps and charts. NCL provides its own programming language, can read a multitude of data formats and provides a number of processing and statistical analysis techniques.

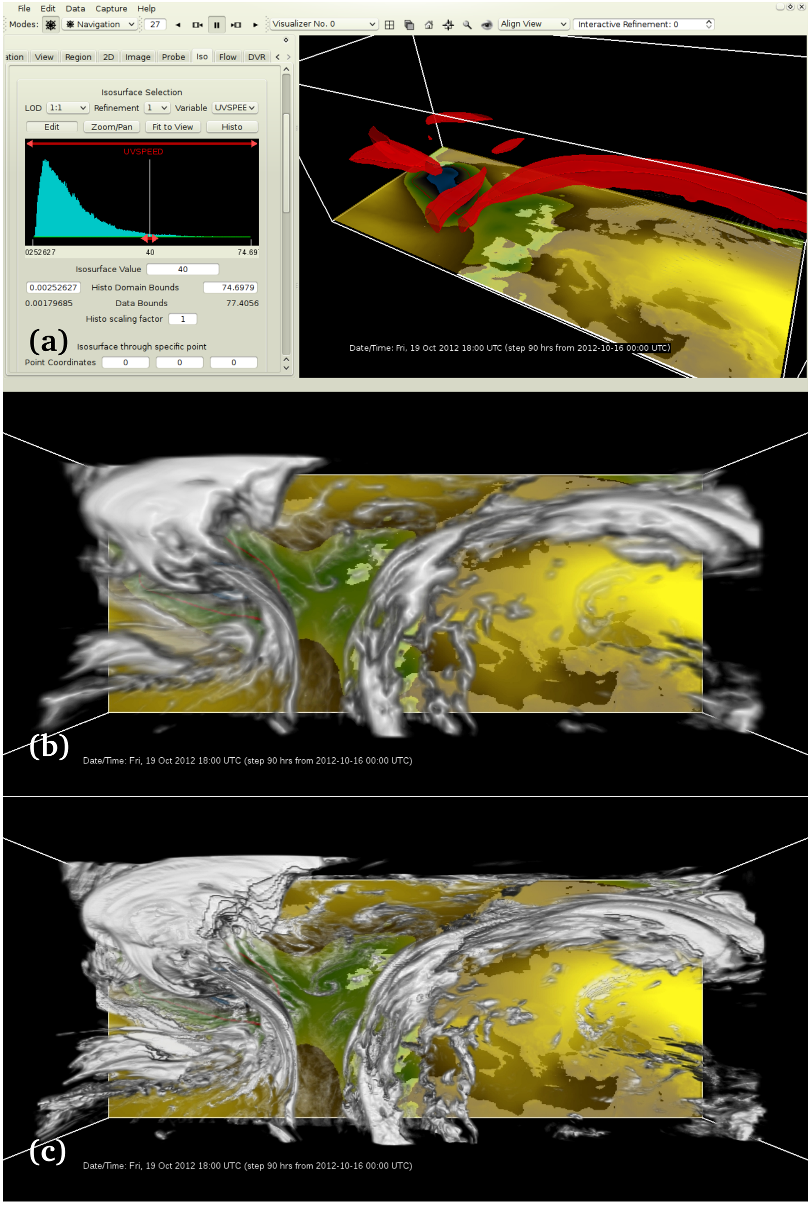
- The *Python* programming language[23] (Pérez et al., 2011) is commonly used, along with specialised libraries for plotting, interactive usage, and data processing and analysis. Frequently used open-source libraries include *matplotlib*[24] (Hunter, 2007) and the UK Met Office *SciTools*[25] for plotting, *IPython*[26] (Pérez and Granger, 2007) for interactive command line usage, and *SciPy*[27] (Oliphant, 2007) for data analysis, but many other libraries exist. Notably, a part of the *Ultrascale Visualization Climate Data Analysis Tools*[28] (UV-CDAT; Williams et al., 2013) is built on Python. Also, via PyNGL[29] and PyNIO, Python bindings to NCAR Graphics exist.

- On the commercial side, *IDL*[30] and *Matlab*[31] are software packages that, similarly to NCL and Python with its libraries, provide a scripting-language interface with a multitude of processing and analysis functions and an extensive number of plotting options, including some 3D support. In particular IDL is heavily used in the atmospheric sciences, as also reported by Middleton et al. (2005).

- *GrADS*[32], *Ferret*[33] and *GMT*[34] are specialised open-source analysis and plotting packages for the atmospheric and oceanographic sciences. GrADS and Ferret provide their own scripting languages, while GMT is a collection of small tools for a variety of mapping tasks.

- The ECMWF *Metview*[35] software (open-source since version 4; Russell et al., 2010) takes a hybrid role between research and operational forecasting. It features a graphical user interface for data processing and visualization, but can also be scripted with its own macro language (Siemen et al., 2010). Metview's graphics are based on the ECMWF *Magics++*[36] library. Metview is heavily used at ECMWF and cooperating institutions. Its history is described by Woods (2005, Ch. 15).

[21] http://www.ncarg.ucar.edu
[22] http://www.ncl.ucar.edu
[23] https://www.python.org
[24] http://matplotlib.org
[25] http://scitools.org.uk
[26] http://ipython.org
[27] http://www.scipy.org
[28] http://uvcdat.llnl.gov
[29] http://www.pyngl.ucar.edu
[30] http://www.exelisvis.com/ProductsServices/IDL.aspx
[31] http://www.mathworks.com
[32] http://grads.iges.org/grads
[33] http://www.ferret.noaa.gov
[34] https://www.soest.hawaii.edu/gmt
[35] https://software.ecmwf.int/wiki/display/METV/Metview
[36] https://software.ecmwf.int/wiki/display/MAGP/Magics



File
Edit
Data
Capture
Help
ation
View
Region
2D
Image
Probe
Iso
Flow
DVR
Isosurface Selection
Edit
Zoom/Pan
Fit to View
Histo
UVSPEED
0252627
40
74.69
(a)
Isosurface through specific point
Date/Time: Fri, 19 Oct 2012 18:00 UTC (step 90 hrs from 2012-10-16 00:00 UTC)
(b)
Date/Time: Fri, 19 Oct 2012 18:00 UTC (step 90 hrs from 2012-10-16 00:00 UTC)
(c)
Date/Time: Fri, 19 Oct 2012 18:00 UTC (step 90 hrs from 2012-10-16 00:00 UTC)

Modes: Navigation 27

Visualizer No. 0

Align View

Interactive Refinement: 0

LOD 1:1 Refinement 1 Variable UVSPEE

Isosurface Value 40

0.00252627 Histo Domain Bounds 74.6979

0.00179685 Data Bounds 77.4056

Histo scaling factor 1

Point Coordinates 0 0 0

Figure 4.8: The Vapor visualization tool. **(a)** Screenshot of the application window, displaying an isosurface (40 ms^{-1}) of horizontal wind speed. Shown is the ECMWF deterministic forecast initialised at 00:00 UTC 16 October 2012, valid at 18:00 UTC 19 October 2012 (cf. Fig. 1.3). **(b)** Volume rendering of predicted cloud cover fraction of the same forecast, loaded at a low wavelet resolution. **(c)** The same as (b) but at higher wavelet resolution.

In research environments, 3D visualization is more frequently used than in operational forecast environments. Vis5D, as mentioned above, had been widely used into the 2000s, however, its development was discontinued. More recently, prominent tools include *Vapor*, the Unidata *Integrated Data Viewer* IDV, and the general-purpose tool *Paraview* (cf. Sect. 3.3).

- Vapor[37] (Norton and Clyne, 2012; Clyne et al., 2007) is an open-source 3D visualization software developed at the United States National Centre for Atmospheric Research. Similar to Vis5D, it features a number of 2D and 3D visualization techniques to view time varying gridded datasets.

[37] http://www.vapor.ucar.edu

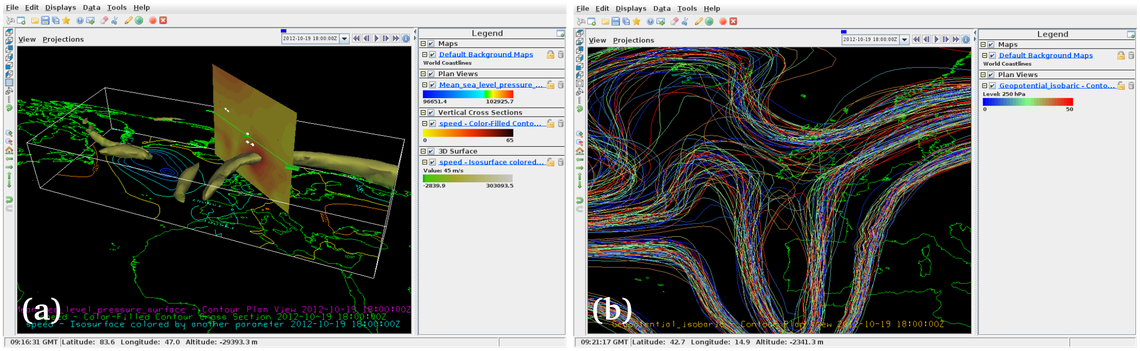

File Edit Displays Data Tools Help
View Projections
2012-10-19 18:00:00Z
Legend
Maps
Default Background Maps
World Coastlines
Plan Views
Mean_sea_level_pressure_...
96651.4
102925.7
Vertical Cross Sections
speed - Color-Filled Conto...
0
65
3D Surface
speed - Isosurface colored...
Value: 45 m/s
-2839.9
303093.5
(a)
Mean_sea_level_pressure_surface - Contour Plan View 2012-10-19 18:00:00Z
speed - Color-Filled Contour Cross Section 2012-10-19 18:00:00Z
speed - Isosurface colored by another parameter 2012-10-19 18:00:00Z
09:16:31 GMT Latitude: 83.6 Longitude: 47.0 Altitude: -29393.3 m
File Edit Displays Data Tools Help
View Projections
2012-10-19 18:00:00Z
Legend
Maps
Default Background Maps
World Coastlines
Plan Views
Geopotential_isobaric - Conto...
Level: 250 hPa
0
50
(b)
Geopotential_isobaric - Contour Plan View 2012-10-19 18:00:00Z
09:21:17 GMT Latitude: 42.7 Longitude: 14.9 Altitude: -2341.3 m

Figure 4.9: Examples of the IDV visualization tool showing the ECMWF ensemble forecast from 00:00 UTC 17 October 2012, valid at 18:00 UTC 19 October 2012. **(a)** Ensemble control forecasts. 3D isosurface (45 ms^{-1}) of horizontal wind speed, augmented by a vertical cross-section of horizontal wind speed (colour in ms^{-1}) and contour lines of mean sea level pressure (hPa). **(b)** Spaghetti plot of all 51 ensemble members at 250 hPa.

One of Vapor's particular features is its wavelet-compressed data format (Clyne et al., 2007; Clyne, 2012; Norton and Clyne, 2012), allowing progressive access to multiple resolution levels of the data. Datasets in the Vapor data format undergo a wavelet transform and are stored as wavelet coefficients. This enables the user at runtime to truncate the number of coefficients loaded, hence to reduce the dataset to a coarser, compressed version. This is particularly useful for high-resolution datasets whose size surpasses the available memory. If the user finds an interesting region, the region can be selected and the full resolution be loaded only for that particular region. Most model data needs to be converted to Vapor's data format. However, at the time of writing, output from a few NWP models (in particular US models including WRF) can be read directly.

Figure 4.8 shows examples of Vapor's visualization capabilities applied to the T-NAWDEX-Falcon dataset. Figure 4.8b and c shows data of the same time step, but loaded at different wavelet resolutions. Vapor offers visualization techniques for 3D isosurface and direct volume rendering, 2D colour mapped planes, steady and unsteady flow lines, surface topography, geometric models and, recently as of March 2015, 2D contour lines. For analysis, Vapor features an integrated Python compute engine. Also, extensions to IDL are available. The tool's user interface is designed around the visualization algorithms. Each type of algorithm provides a tab pane (on the left side of the window in Fig. 4.8a) on which relevant interface elements are arranged. Vapor uses geometric height as its vertical coordinate in visualizations and internal data model. For model data available only on pressure levels (such as the ECMWF forecasts) a height variable (for instance, geopotential height or flight level) needs to be pre-computed and passed to the tool.

Norton and Clyne (2012) list features and limitations of Vapor. A distinct advantage is that due to the progressive data access model, users can trade-off performance for data resolution and thus achieve interactivity even on low-performance commodity hardware. Limitations include a restriction to structured, regular grids (although the grid spacing need not be uniform), and that all data in Vapor are assumed to arise from a single numerical experiment (that is, the comparative visualization of multiple datasets is not possible). At the time of writing, Vapor does not provide techniques for ensemble data or forecasting functionality. However, as of March 2014, data in GRIB format are supported.

- In the early 1990s as well, Bill Hibbard and coworkers started work on the *Visualization for Al-*

gorithm Development (VisAD) library (Hibbard et al., 1992, 1994), with the goal of simplifying the visualization of multiple heterogeneous data types. A few years later, VisAD, originally implemented in C, was rewritten in the then-new Java programming language to facilitate easy portability between computer platforms (Hibbard, 1997, 1998). VisAD is in continued development today and has become the basis for a number of Java-based visualization tools (Murray et al., 2001), in particular, the Unidata *Integrated Data Viewer*[38] (IDV; Murray and McWhirter, 2007; Murray et al., 2009) and for the latest version of McIDAS (Achtor et al., 2008).

IDV is a comprehensive open-source Java application for the analysis and visualization of geosciences data. Example screenshots are displayed in Fig. 4.9. The tool supports a variety of 2D and 3D visualization methods, including 3D techniques similar to Vis5D (in fact, the tool's visualization style bears much resemblance to Vis5D) and basic ensemble support (for example, spaghetti plots). IDV's core strengths are its portability, its support for a multitude of data formats (including NetCDF, GRIB, and numerous other formats), and an extensive array of visualization techniques (in addition to 2D and 3D spatial displays, a large number of meteorological charts, including vertical soundings and observation plots, are available). Notably, IDV can access data sources directly from the internet, from which a large number of sources (in particular data from the U.S.) are publicly available. IDV uses the Java3D API for rendering, an interface using either OpenGL or DirectX. IDV also provides 3D stereo support and a "fly-through" option. For example, Yalda et al. (2012) use IDV's 3D capabilities for interactive immersion learning. Further information about IDV's capabilities can be found in its user guide[39].

- On a broader scope, the general-purpose visualization tool *Paraview*[40] (cf. Sect. 3.3; Henderson et al., 2004; Ayachit et al., 2012) can also be used with meteorological data. For instance, in the context of a graduate university course, Dyer and Amburn (2010) investigated how Paraview can be used in a meteorological setting. As noted in Sect. 3.3, Paraview builds on the VTK visualization library and offers the extensive visualization capabilities of VTK. However, Paraview is not specifically designed for meteorological data and hence it takes some effort to create visualizations. At the time of writing, the visualization group at DKRZ offers an extensive tutorial on how to use Paraview with meteorological datasets[41]. For instance, DKRZ has developed a plugin to directly load ICON data into Paraview.

  Another general-purpose visualization tool that has been applied to meteorological datasets is the initially commercial IBM Data Explorer, later released as open-source as *OpenDX*[42]. As noted in the previous section, OpenDX was employed in the 1990s by Lloyd Treinish and coworkers in one of the attempts to use 3D visualization in weather forecasting.

- Commercial 3D general-purpose visualization codes are rarely used in atmospheric research. Notably, *Avizo*[43] (formerly *amira*) in its "Green" version is used at DKRZ for the visualization of

[38] https://www.ssec.wisc.edu/mcidas/software/v

[39] Available at http://www.unidata.ucar.edu/software/idv/docs/userguide (last access April 2015).

[40] http://www.paraview.org

[41] Available from the DKRZ website at https://www.dkrz.de/Nutzerportal-en/doku/vis/sw/paraview (last access April 2015).

[42] http://www.opendx.org

[43] http://www.fei.com/software/avizo3d

climate visualizations[44], mainly for animations for the media. Previously, AVS[45] (Chen, 1997) has been used at DKRZ, as reported by Böttinger et al. (1998). Also, the (at the time writing discontinued) *SimVis* framework (Doleisch et al., 2004; Doleisch and Hauser, 2012) is used at DKRZ[46]. The software is specialised to provide brushing and linking techniques (cf. Sect. 3.1) for the analysis of datasets.

4.2.3 Visualization for research flight planning

The state of the art in weather forecasting for research flight planning can be regarded as a hybrid between the operational forecast and research settings reviewed in the previous subsections. While similar data exploration requirements exist as in operational forecast settings, the tools that are employed often stem from research environments. For example, in Rautenhaus et al. (2012) we note that in recent years, available forecast data have been mainly provided by means of static or animated plots created by the scientists supporting the aircraft mission. Meteorological predictions have often been presented with the help of simple web interfaces. An early such interface is described in Flatøy et al. (2000). During recent campaigns, including POLARCAT-GRACE (e.g. Roiger et al., 2011) and CONCERT 2008 (e.g. Voigt et al., 2010), DLR-IPA provided access to ECMWF forecasts and Meteosat imagery through an interface allowing the user to view up to two synchronised loops of the products (an example of which is provided in Rautenhaus et al., 2012).

Institutions other than DLR that conduct airborne atmospheric measurements include the American National Center for Atmospheric Research (NCAR) and the National Aeronautics and Space Administration (NASA). NCAR operates so-called field catalogues[47]. On these web sites, forecast and observational data from satellites and radar can, similar to the DLR-IPA web site, be viewed in loops of pre-computed images. The web pages also serve as a collection of links relevant to each campaign.

While providing predefined batch imagery through a website has the advantage of having an easy-to-use interface that can be accessed from any computer with a browser and Internet access, the approach has the disadvantage that the possibilities to interact with the data are very limited. More freedom in the exploration of the predictions, in controlling on-demand computations and, in particular, the possibility to plan the flight route in direct relation to the available data makes the support of research missions much more efficient. In this respect, NASA has developed the Real Time Mission Monitor (RTMM, Blakeslee et al., 2007), and, in collaboration with the University of Alabama, Huntsville, the Waypoint Planning Tool (WPT, He et al., 2010). The RTMM integrates real-time information from the NASA research aircraft, auxiliary observation platforms (satellites, radar), and other Earth science datasets into a GoogleEarth interface to improve the experiment's real-time situational awareness. The Java-based WPT allows the user to interactively define a flight route on maps of provided atmospheric data.

With respect to field campaigns with DLR involvement, the MSS (cf. Fig. 1.1) has frequently been used in recent years. As noted in Chapter 1, the tool generates horizontal and vertical 2D sections of the forecast data (in particular, vertical sections along planned flight routes; Rautenhaus et al., 2012), the

[44] See `https://www.dkrz.de/Nutzerportal-en/doku/vis/sw/avizo-en`; an extensive tutorial is available at `https://www.dkrz.de/pdfs/docs/Tutorial_Avizo_Green_8.pdf` (last access April 2015).

[45] `http://www.avs.com`

[46] Compare to `https://www.dkrz.de/Nutzerportal-en/doku/vis/sw/simvis` (last access April 2015).

[47] Available at `http://catalog.eol.ucar.edu` (last access April 2015).

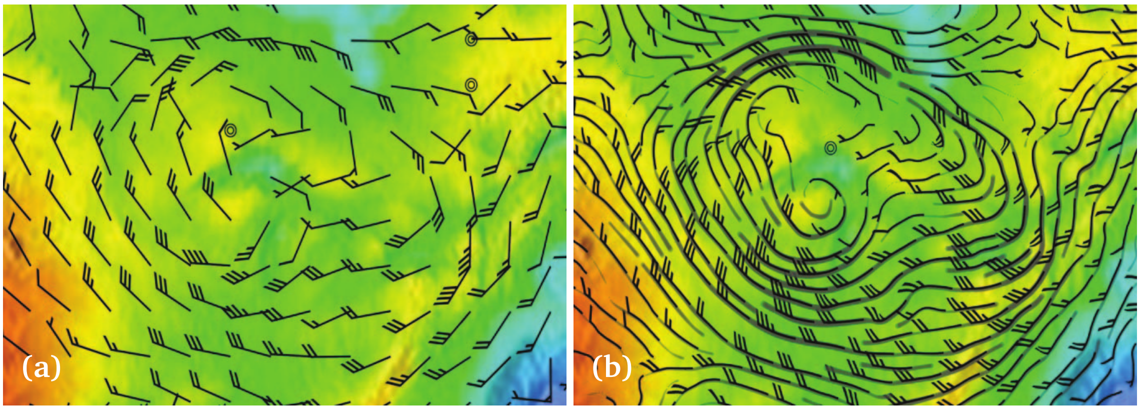


Figure 4.10: The improved wind barbs approach by Pilar and Ware (2013). **(a)** A wind field depicted by traditional wind barbs. **(b)** The same wind field visualized with a combination of stream lines and wind barbs proposed by Pilar and Ware. Small structures in the wind field that are missed by the visualization in (a) are captured in (b). (Reprinted from Pilar and Ware (2013), © 2013 IEEE. Used with permission.)

layout and use of which motivated the design of my proposed bridge from 2D to 3D that I describe in Sect. 5.2.

4.3 Visualization research

Besides the visualization techniques that are available in the software tools surveyed in Sect. 4.2, a number of studies pertaining to meteorological simulation data have been published by the visualization community. This section provides a brief overview of recent research. Particular focus is put on publications dealing with the visualization of ensembles.

4.3.1 Visualization research related to meteorology

A number of studies have been published with respect to improving the visual depiction of atmospheric flow, for example, by means of streamlines and flow texture representations (e.g. Jobard and Lefer, 1997; Jobard et al., 2001). Here, recent examples include the works by Ware and Plumlee (2013), Pilar and Ware (2013), and Ware et al. (2014), who study alternative flow depictions and their combination with scalar fields from a perceptual point of view. Ware and Plumlee (2013) investigate how 2D weather maps displaying three or more variables can be improved. Alternative approaches to depict the wind vector field and multiple scalar variables are explored, using static and animated displays with different colour, texture, and glyph schemes to target distinct perceptual channels. Ware and Plumlee (2013) evaluate their approaches with a user study, noting, for instance, the effectiveness of a wind depiction by animated particle traces. Pilar and Ware (2013) investigate how the 2D display of streamlines and wind barbs can be improved (Fig. 4.10). In their work, wind barbs (and alternatively arrow glyphs) are placed along streamlines to combine advantages of both approaches to visualize the flow field. The streamlines achieve a better spatial sampling of the flow, capturing small scale structures sometimes missed by regularly placed wind barbs. They also have the advantage to everywhere being tangential to the flow (which wind barbs are only at their tip). Yet, the approach by Pilar and Ware (2013) maintains the advantages of a glyph-based depiction of flow velocity and direction. In case of wind barbs along the streamlines, the "traditional" depiction that meteorologists are used to is maintained. Further flow-visualization examples include the usage of multi-touch devices (Butkiewicz and Ware, 2011), and GPU-based flow visualization via animated particle tracing for time dependent climate simulation output (Cuntz et al., 2007).

Research in flow visualization is closely related to feature tracking and visualization. Here, examples include the tracking of flow features based on the geometry of their trajectories (Kendall et al., 2012) and the tracking of jet stream features, proposed by Limbach et al. (2012) to compute climatologies of jet stream events from ECMWF analysis data. Lee et al. (2013) track events of the Madden-Julian Oscillation in a climate simulation, depicting the results in a GoogleEarth based display. Griffith et al. (2005) and Heus et al. (2009) investigate the tracking of cumulus clouds simulated by an LES model. Their studies combine feature tracking with research into virtual reality environments, allowing the user to select the cloud to be tracked in a virtual reality workbench (Koutek, 2003). A number of other, more general, studies have investigated feature-based flow visualization and frequently used examples from the atmospheric domain. Here, the reader is referred to the survey articles by, for example, Post et al. (2003) and McLoughlin et al. (2010). Also, with respect to virtual reality, 3D visualization has been investigated with respect to immersive virtual reality environments (Ziegler et al., 2001; Hibbard, 2005), including the usage of immersive virtual reality visualization for teaching in meteorology classes (Gallus et al., 2003, 2005).

In the context of the IEEE Visualization 2004 contest, Doleisch et al. (2004) demonstrate the application of interactive brushing and linking and focus+context techniques (cf. Sect. 3.1) in the SimVis framework (cf. Sect. 4.2) to the exploration of a simulation of Hurricane Isabel. Similar techniques in the same tool are used by Kehrer et al. (2008) and Ladstädter et al. (2010), who use SimVis with ECMWF re-analyses and ECHAM datasets. Without employing prior knowledge of the data, the interactive visual exploration techniques in the tool are used to generate hypotheses about possible indicator parameters and regions for climate change. The work by Doleisch et al. (2004), Kehrer et al. (2008) and Ladstädter et al. (2010) applies some of a large body of visualization techniques from the field of interactive visual analysis of multifaceted data to atmospheric data. For further information about such techniques, the reader is referred to the overview by Doleisch and Hauser (2012) and the survey article by Kehrer and Hauser (2013). Also, the PhD thesis by Nocke (2007) extensively discusses the application of visual data mining techniques to climate sciences.

A number of studies explore psychological aspects of the design of weather-related displays. For example, Hegarty et al. (2010) investigate the effects of salience of the depiction of specific forecast variables on a weather map on typical inference tasks. They note that weather maps should be designed to make task-relevant information salient in a display. Trafton and Hoffman (2007) suggest improvements to meteorological visualizations and tools, based on notions of human-centric computing, and Nadav-Greenberg et al. (2008) explore psychophysical issues of the effect of uncertainty visualization on decision making in weather forecasting.

Further studies include works on cloud rendering (e.g. Harris and Lastra, 2001; Harris et al., 2003; Riley et al., 2003; Hergenröther et al., 2002; Trembilski, 2001), the generation of synthetic satellite imagery from model data (e.g. Keil et al., 2006; Lakshmanan et al., 2012; Millington et al., 2012), the coupling of atmospheric simulation and visualization software codes (e.g. Shen et al., 2011; Fjukstad et al., 2010), approaches to volume rendering (e.g. Song et al., 2006; Kniss et al., 2002), and the usage of 3D visualization to depict observation data (e.g. Arthur et al., 2010). Recently, the IEEE Visualization

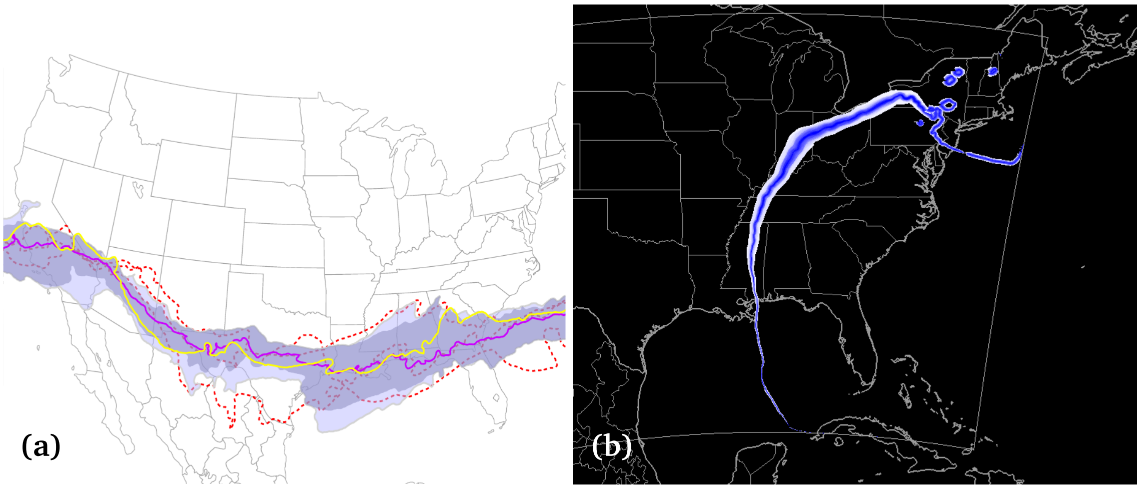


Figure 4.11: Alternatives to spaghetti plots recently suggested in the literature. **(a)** Whitaker et al. (2013) propose a generalisation of box plots (cf. Fig. 2.15) to contour lines. **(b)** Sanyal et al. (2010) propose uncertainty ribbons to depict the uncertainty along a contour line. (Reprinted from Whitaker et al. (2013), © 2013 IEEE, and from Sanyal et al. (2010), © 2010 IEEE. Used with permission.)

2014 contest considered the visualization of volcanic ash clouds (Günther et al., 2015).

4.3.2 Ensemble visualization

Ensemble visualization aims at identifying variability, similarities, and differences among ensemble members. It is closely related to uncertainty visualization, of which Pang et al. (1997), Johnson and Sanderson (2003), and recently Bonneau et al. (2014) provide general overviews. MacEachren et al. (2005) provide an overview focussed on geospatial data.

In Sect. 2.5, I have surveyed common techniques used to visualize ensemble forecasts in operational forecasting. 2D visualizations of statistical quantities that summarize the ensemble distribution or that represent relative frequencies for events are frequently used. For example, current products provided in ECMWF's *ecCharts* system (Lamy-Thépaut et al., 2013) include maps of mean and standard deviation, maps of threshold probabilities (for example, the probability of precipitation exceeding a critical threshold) and of derived statistical measures (for example, the extreme forecast index, Lalaurette, 2003). Many of the presented techniques are also used in the research literature. For instance, spaghetti plots are frequently encountered: The *American Meteorological Society* journal homepage finds on the order of 100 publications using this type of plot[48], including, for instance, the studies by Toth et al. (1997), Doyle and Reynolds (2008), Schumacher and Davis (2010), and Matsueda et al. (2011). Recently, Böttinger et al. (2015) explored how techniques implemented in the visualization tools NCL, Avizo Green and Paraview (cf. Sect. 4.2.2) can be used to visualize uncertainty in 2D fields obtained from ensembles of decadal climate simulations.

In a recent survey–also including applications outside the atmospheric domain–, Obermaier and Joy (2014) classify ensemble visualization methods described in the literature into *location-based methods* and *feature-based methods*. Location-based methods compare ensemble properties at fixed locations in the dataset. In the simplest case, this includes the ensemble mean, standard deviation, or probability as

[48] Search on the keyword “spaghetti”, conducted in March 2015 on `http://journals.ametsoc.org/search/advanced`.

computed at a given grid point. Such statistical quantities have been visualized via colour maps, opacity, texture, and animation (Djurcilov et al., 2002; Rhodes et al., 2003; Lundstrom et al., 2007). Also, glyphs have been used to display, for example, uncertainty in wind fields (Wittenbrink et al., 1996).

Feature-based methods, on the other hand, extract features from each ensemble member and aim at visually comparing the detected features. Examples include spaghetti plots (where the isolines are the features), the joint display of detected cyclonic features (Hewson and Titley, 2010), and visualization techniques for the prediction of hurricane tracks (Cox et al., 2013). In recent studies, Whitaker et al. (2013) and Mirzargar et al. (2014) offer alternatives to spaghetti plots. They generalise boxplots to *contour boxplots* in order to enable an improved quantitative and qualitative analysis of ensembles of 2D isocontours and level-sets (Fig. 4.11a).

In 3D, the effect of uncertainty on the position of 3D isosurfaces has been the topic of a number of studies. It has been approached with, for instance, geometric displacements (Grigoryan and Rheingans, 2004) and surface animation (Brown, 2004). In a study concerning the reconstruction of the earth's subsurface model, Zehner et al. (2010) visualize confidence intervals around an isosurface using additional transparent surfaces as well as lines connecting the surfaces. Recently, techniques have used stochastic modelling of uncertainty in scalar ensembles to quantify and visualize the possible occurrences of isosurfaces (Pöthkow and Hege, 2011; Pöthkow et al., 2011; Pfaffelmoser et al., 2011; Pfaffelmoser and Westermann, 2012). The latter studies all include examples from the atmospheric domain.

A few articles in the visualization literature have presented software tools that put special emphasis on ensembles in earth-science applications. Potter et al. (2009) present the *Ensemble-Vis* tool and investigate the usage of multiple linked views to visualize 2D weather simulation ensembles. They conclude that the combination of standard statistical displays (spaghetti plots, maps of mean and standard deviation) with user interaction facilitates clearer presentation and simpler exploration of the data. In their *Noodles* tool, Sanyal et al. (2010) enhance spaghetti plots by glyphs and confidence ribbons to highlight the Euclidean spread of 2D contour ensembles (Fig. 4.11b). They describe the usage of their methods by atmospheric researches investigating different parametrisations in the Weather Research and Forecasting (WRF) model. Sanyal et al. also highlight the positive effect of interactivity and linked views on the user and note the challenge of potential generalization of their work to three dimensions. Recently, Höllt et al. (2014) have presented *Ovis*, a system for the visualization of 2D ocean heightfield ensemble data. They again use linked views of maps, statistical plots and 3D renderings and demonstrate the use of time-series glyphs for the comparative visualization of the ensembles at two different positions over time. Höllt et al. discuss the application of their tool to off-shore oil operations and the planning of underwater glider paths.

CHAPTER 5

The 3D ensemble visualization tool Met.3D

In Chapters 2, 3, and 4, I have reviewed relevant foundations of ensemble weather prediction and of scientific visualization, as well as the current state of the art in meteorological visualization. As discussed in Chapter 4, first 3D visualization systems for meteorological use have been developed in the 1980s. However, in this period of large mainframe computers, response times of the early systems were too slow to allow for continuous real-time interaction. In the 1990s, the period of dedicated graphics workstations, Vis5D emerged as the first widespread open-source desktop visualization tool specific to the atmospheric domain, alongside with commercial general purpose visualization systems including IBM DX and AVS. In the late 1990s, attempts to use 3D visualization in forecasting have been reported by Treinish and Rothfusz (1997), Koppert et al. (1998) and McCaslin et al. (2000). Besides the reported positive aspects, the hindering factors (including a decoupling of proven 2D methods from new 3D techniques, spatial perception in 3D views, and usability, performance and hardware issues) caused the projects to be discontinued (cf. Sect. 4.2.1). No similar projects have been reported since. Today, 2D visualization methods still prevail in both operational forecast settings and research environments, although 3D visualization tools including Vis5D, Vapor and IDV (cf. Sect. 4.2.2) have found usage in the latter. The availability of ensemble visualization techniques in interactive systems is only starting to emerge (for example, in IDV; cf. Fig. 4.9).

This chapter presents the 3D ensemble visualization tool *Met.3D*, the software that has been developed in this thesis work in order to achieve the objectives stated in Sect. 1.3. In line with the works of Treinish and Rothfusz (1997), Koppert et al. (1998), and McCaslin et al. (2000), Met.3D facilitates the application of 3D visualization techniques to forecasting. However, it differs from the previous approaches with respect to its central support for ensemble forecasts, GPU-based visualization, as well as a number of further features that will be discussed in this chapter. I note that Met.3D is not intended to be a full-featured meteorological workstation; this would be beyond the scope of the presented work. However, besides the primary objective to be used for forecasting, I have designed the software in a way that it can be used as a framework into which new ensemble visualization techniques can be implemented and evaluated with respect to their value for forecasting.

The present chapter is divided into four sections. Based on the literature reviewed in the previous chapters, I motivate the development of Met.3D in Sect. 5.1. In Sect. 5.2, Met.3D is described from a user perspective, illustrating the available visualization methods and the tool's usage concept. Section 5.3 focuses on methodological and technical aspects. The system architecture is described, with emphasis put on Met.3D's parallelised data processing pipeline and the GPU based visualization algorithms. Finally, Sect. 5.4 investigates whether statistical quantities can be computed directly from the individual ensemble members' hybrid model level grids (which differ slightly due to the variation in surface pressure among the members), or whether the members need to be regridded to a common grid.

5.1 Rationale for the development of Met.3D

With respect to the thesis objectives listed in Sect. 1.3, the works by Treinish and Rothfusz (1997), Koppert et al. (1998) and McCaslin et al. (2000) are most closely related to my work. All three have been discontinued; no software is available anymore. Also related are the open-source meteorological visualization tools Vis5D (discontinued in 2001), Vapor, and IDV. However, while all possess extensive functionality and distinct strengths, none of these tools have been designed for or are directly suited to visualize ensemble forecasts in 3D in a way suited to meet the thesis objectives. The same applies to general purpose visualization systems including the open-source ParaView and OpenDX (discontinued in 2007), and the commercial Avizo Green. Issues still include a straightforward transition from 2D to 3D, 3D perception, rendering performance, as well as the visualization of ensemble products, and, specific to this thesis, techniques to forecast WCBs.

In the following, I propose a list of goals for characteristics and functionality of a tool that targets the 3D visualization of ensemble weather predictions in a way suited to weather forecasting. In the context of the reviewed literature, I evaluate existing tools, describe potential design options to achieve these goals, and discuss the rationale for choosing to develop the new visualization tool Met.3D.

Bridge from 2D to 3D. As noted in Chapter 4, 2D methods prevail in the atmospheric sciences. Forecasters and researchers are used to these methods, and they provide many advantages. It is hence desirable in a 3D visualization system to reproduce 2D forecast products as *closely as possible to their "originals"*. My assumption is that in a 3D context, added value can be achieved by allowing the user to move 2D products in 3D space using continuous real-time interaction, and by combining 2D products with 3D visualization elements. The literature provides support for this argument. For example, in a summary of five reports from US research centres involved in meteorological visualization research (Rhyne et al., 1992), E. Pepke and K. W. Johnson (Florida State University) note:

> "Advanced 3D visualization techniques will never supplant more traditional 2D techniques. 3D graphics can give a better overall impression of the data, but 2D graphics often permit more precise measurement of specific features. The combination will always be more effective than either alone."

My goal is hence not to replace proven 2D techniques, but to put them into a 3D context and to use 3D elements to add value. Continuous interaction is discussed shortly.

A second important aspect of a bridge from 2D to 3D visualization is to achieve good *spatial perception* in the 3D view. 2D elements need to show clear geographic reference, and it must be obvious where in space the depicted 3D elements are located. In a campaign settings, I assume forecasters to work with monocular displays (notebooks and desktop computers). Hence, monocular depth cues that are known to be effective (including, for instance, shadows and dropped lines; cf. Sect. 3.1.3) need to be supported.

Ensemble support and extensibility. Ensemble support needs to be an integral part of the tool. Access to uncertainty information obtained from the ensemble should not only be possible but *be in the centre of the exploration process*. The ensemble, or uncertainty, dimension should play an equivalently prominent role as the forecast valid and initialisation time dimensions. This includes straightforward access to existing 2D techniques as surveyed in Sect. 2.5 (for example, maps of ensemble mean and standard deviation), their extension to 3D, and the exploration of individual members of the ensemble. For research in ensemble visualization (for this thesis in particular to develop methods that target the forecast questions listed in Sect. 1.3), it is necessary to be able to integrate and evaluate new ensemble visualization techniques. This may include techniques emerging from the literature, as surveyed in Sect. 4.3.

Suitability for (campaign) forecasting. For usage in a forecast setting, a tool must provide straightforward access to preconfigured forecast products, as well as to time navigation (including forecast valid and initialisation times) and, considering ensemble support, to ensemble navigation. The tool should provide the user with access to all available forecast data without requiring knowledge of how the data is structured on disk (that is, no explicit loading of data files by the user).

Continuous-interactive visualization performance. Response time, the time required to display a new image after the user has interacted with, for example, a visualization parameter, camera or time step, is crucial to the acceptance of an interactive visualization tool, as Szoke et al. (2003) and Hibbard (2004) emphasize. A step further, Faconti and Massink (2001) elaborate on the importance of *continuous* real-time interaction as a requirement for interactive systems to support native human behaviour. Following Piringer et al. (2009), this is what I believe needs to be achieved for thorough exploration of forecast data – as Piringer et al. point out, “continuous interaction [...] reduces the risk of losing interesting scenarios”.

In the past decade, the capabilities and performance of consumer-class graphics hardware have increased at a tremendous speed. Section 3.2 described how GPUs have evolved in the past 15 years. In particular the introduction of fully programmable shader stages and the possibility to store and randomly access 3D volume data in GPU memory have, only a few years ago, enabled visualization algorithms that largely outperform CPU-based approaches by implementing both mapping and rendering stages of the visualization pipeline on the GPU. Many studies from the visualization literature have shown how the power provided by GPUs can be used for interactive real-time visualization (cf. Sect. 3.2). This power should be exploited to the best possible extent to achieve continuous interactivity in a 3D ensemble visualization tool.

A second important aspect of interactivity is an efficient data handling and processing in the filtering stage of the visualization pipeline. Usage of multi-threaded CPU processing and caching strategies substantially contributes to high application performance as well (Piringer et al., 2009).

Support for hybrid sigma-pressure vertical coordinates. As described in Sect. 2.4, data from the ECMWF ensemble (as well as deterministic) prediction system are available horizontally either globally in spectral or reduced Gaussian coordinates, or regionally interpolated to a regular latitude-longitude grid. In the vertical, either hybrid sigma-pressure model levels (at the time of TNF 62 levels) or a set of predefined constant pressure levels (at the time of TNF 12 levels) are available. In a typical campaign setting, only regional data is required. Also, restrictions in internet bandwidth have in recent years prevented the transmission of global data fields from ECMWF (cf. Rautenhaus et al., 2012). Hence, while a horizontally regular grid is the realistic case for the considered campaign forecasts, two arguments can be given to require visualization algorithms to operate vertically on hybrid sigma-pressure levels. First, for 3D visualization, maximum vertical resolution is desired. Regridding the model grid to constant pressure levels results either in loss of accuracy in regions with complex orography, or in an increased data volume (if the number of levels is increased). Also, regridding introduces an additional interpolation step, which may degrade data quality. Second, processing the data prior to visualization may be time consuming. Since time is a valuable resource in weather forecasting, the forecast tool should be able to visualize new forecast data directly upon reception. An argument against the use of hybrid sigma-pressure levels is that the spatial interpolation required by visualization algorithms becomes computationally expensive if the grid is not uniform (cf. Sect. 3.1). However, this may be counteracted by the use of GPU-based algorithms.

Techniques to forecast WCBs. To target forecast questions FQ-C, D, and E (cf. Sect. 1.3), methods need to be developed that detect WCB structures in the forecast and that quantify the uncertainty of the structures. These methods must be accessible in the tool as "regular" forecast products.

Vapor, IDV, ParaView, as well as the discontinued tools Vis5D and OpenDX were, at the beginning of this thesis work, only able to fulfil a very limited part of the listed goals. With respect to a bridge from 2D to 3D, the visualization of "traditional" 2D products (for instance, the one shown in Fig. 1.1) in existing 3D software is until now only possible in a limited way. The discussed 3D visualization systems all provide some functionality to create 2D elements, including cutting planes with colour mappings and contour lines. However, to the best of my knowledge, it is not possible to create complete multi-variable 2D forecast products as they are available in 2D systems (including a combination of contour line and colour mappings of multiple variables, wind barbs or streamlines for flow visualization, graticule and coastlines on constant pressure levels for horizontal sections, along multiple waypoints for vertical sections) in a way such that they can be moved using continuous interaction and are connected to time (and ensemble) navigation. With respect to spatial perception in 3D, depth cues supported by existing systems are limited as well. Although IDV supports stereoscopic views if corresponding hardware is available[1], to the best of my knowledge neither of the discussed tools (if at all) provides straightforward support for real-time shadows or dropped lines that could be used for forecasting.

Support for ensemble datasets in the reviewed systems is either not existing or only very limited. Notably, IDV is able to handle an ensemble dimension to create basic visualizations including spaghetti plots (cf. Fig. 4.9). In terms of forecast functionality, all of the discussed tools have been designed for usage in research settings. None of the tools provides support to easily navigate both forecast initialisation and valid time dimensions and the ensemble dimension. Vapor is restricted to a dataset from a

[1] Compare to `http://www.unidata.ucar.edu/software/idv/docs/userguide/Faq.html` (last access April 2015).

single numerical experiment (cf. Sect. 4.2), restricting visualized data to one forecast initialisation time and one ensemble member. In all tools, the user has to have knowledge of the file structure. Also, the tools require the user to have at least basic understanding of the employed visualization algorithms to use the user interface. ParaView and OpenDX expose the entire visualization pipeline to the user. Due to their age, all tools rely on the old OpenGL fixed-function pipeline and CPU-based mapping stages. Only Vapor makes use of more recent programmable OpenGL functions, however, is aiming at being compatible with old hardware. Vertical hybrid sigma-pressure coordinates pose problems to most systems, only Vapor (as of version 2.2 of February 2013, not at the beginning of this thesis work) is able to process terrain following grids[2].

Techniques to forecast WCB situations with ensemble methods have, to the best of my knowledge, not been reported before. Such techniques are new developments and are missing in any existing tool.

Design alternatives

Given the discussed shortcomings of existing tools with respect to the listed goals, I considered the following options to approach the open issues:

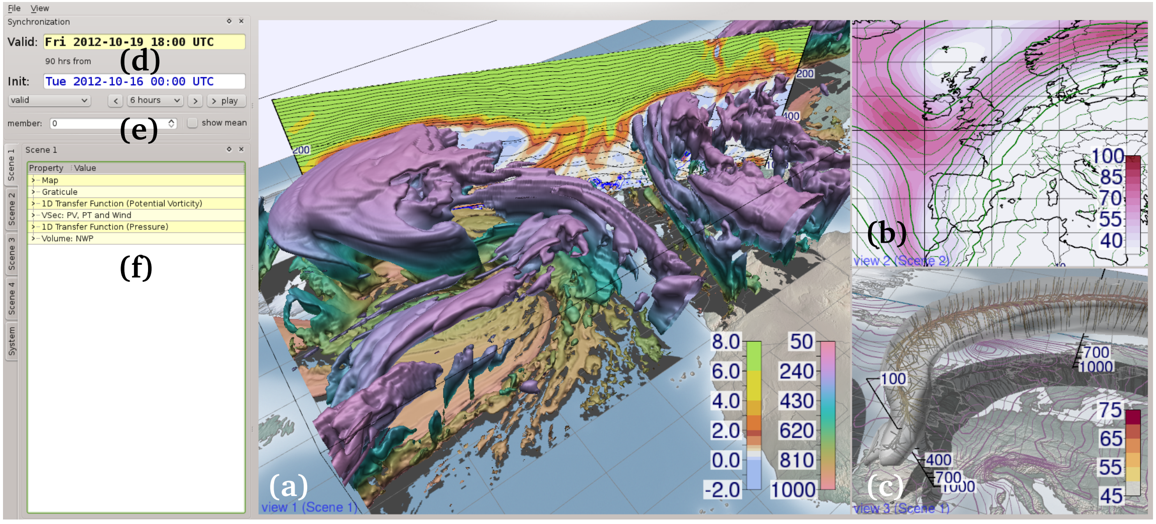
1. Modify and extend an existing visualization tool to meet the requirements.
2. Develop a new tool, based on open-source visualization (or graphics) libraries (with possible choices including VTK and OpenSceneGraph).
3. Develop a new tool with new processing and rendering implementations, with an architecture designed to focus on the goals listed above.

All three approaches have advantages as well as disadvantages. A distinct advantage of the first option would have been to start with a functioning application that already includes a number of visualization techniques. However, all tools in question would have required significant changes to user interface, data management, processing and rendering structure to achieve the goals listed above.

The second option would have given more freedom. However, the choice in potentially suitable available libraries was, to the best of my knowledge, rather limited, including in particular VTK and OpenSceneGraph. VTK's rendering stage (following a CPU-based mapping stage) is based on old fixed-function techniques (cf. Sect. 3.3) and would have needed to be replaced. An option would have been to use the processing stages of the VTK pipeline, together with OpenSceneGraph for rendering, as done by, for instance, Zehner et al. (2010), or with a new shader-based rendering stage. However, VTK's pipeline implementation imposes strong constraints on the architecture and, based on its current functionality (for instance, missing thread-safety; cf. Sect. 3.3), would have needed to be modified and extended as well. Modifications and extensions would have included, in particular, control over parallel execution and memory management, as well as support for the ensemble. The works by Biddiscombe et al. (2007) and Vo et al. (2011) demonstrate the rather high complexity of solutions to such modifications.

Given the considerable modifications and extensions that would have been necessary to any existing tool or library, I decided to follow the last approach. It provides freedom in concept, architecture and implementation to facilitate the investigation of how recent developments in graphics hardware and visualization techniques can be used to achieve the project's goals. In particular, the architecture can

[2] Release notes of version 2.2.0, available at `http://www.vapor.ucar.edu/docs/vapor-installation/release-notes` (last access April 2015).



Synchronization

Valid: Fri 2012-10-19 18:00 UTC

90 hrs from

Init: Tue 2012-10-16 00:00 UTC

valid < 6 hours > > play

member: 0 □ show mean

Scene 1

Property	Value
Map	
Graticule	
1D Transfer Function (Potential Vorticity)	
VSec: PV, PT and Wind	
1D Transfer Function (Pressure)	
Volume: NWP	

Figure 5.1: The main user interface of Met.3D, with examples of 2D and 3D visualization techniques applied to ECMWF ensemble forecasts. **(a)** Isosurfaces of cloud cover fraction of 0.5 coloured by elevation (hPa), and a vertical section of potential vorticity (PVU). **(b)** Horizontal section with contour lines of the mean geopotential height field (m) and filled contours of its standard deviation (m). **(c)** Normal curves applied to the wind field to visualize the jet core. The white isosurface shows 45 ms^{-1}. Colour coding in ms^{-1}. **(d–f)** User interface, see text for details.

be designed with as little overhead as necessary for the given application to best exploit the available hardware resources – a crucial element to achieve continuous real-time interactivity. Despite not using VTK, the concept of the visualization pipeline is, due to its modularity, favourable to the project's goals. I hence propose an alternative implementation, focussing on supporting GPU-based mapping and rendering, ensemble data, memory management, and CPU parallelism.

In the following, I first describe the result of the development by surveying the capabilities of Met.3D in Sect. 5.2. In Sect. 5.3, architecture and implemented algorithms of the tool are described.

5.2 Met.3D overview

The purpose of this section is to provide an overview of the capabilities and characteristics of Met.3D at the time of writing. The tool can be run on any Linux computer that provides a GPU capable of OpenGL 4 (this includes most models released in and after 2010; cf. Sect. 3.2). Met.3D supports forecast data on hybrid sigma-pressure levels (cf. Sect. 2.1.2) from the ECMWF Ensemble Prediction System (cf. Sect. 2.4). Data from ECMWF's high-resolution deterministic forecast are supported as well, as are generic forecast data that are provided on constant pressure levels in the vertical (that need not be uniformly spaced). All forecast data must be available on a regular longitude-latitude grid in the horizontal. It can be distributed over arbitrarily many GRIB or CF-NetCDF files. As noted in the previous section, Met.3D's architecture follows the pipeline concept. However, in contrast to Paraview and OpenDX, the pipeline is not exposed to the user. Instead, a *data processing pipeline* is preconfigured for each forecast dataset. A dataset is exposed to the user as a *data source*. Multiple data sets (that can be provided at differing grid spacings and/or vertical levels) are supported.

5.2.1 User interface

Figure 5.1 shows the graphical user interface (GUI) of Met.3D. The forecast data fields can be displayed in multiple 3D views (Fig. 5.1a, b, and c). In the horizontal, a cylindrical longitude-latitude projection is used. As common in meteorology, the logarithm of pressure serves as the vertical coordinate[3]. Vertical scale, i.e. the proportion of vertical to horizontal units, can be specified for each view individually. Time navigation is provided for the forecast initialisation (or base, or run) time and the forecast valid time (Fig. 5.1d). This way, subsequent forecast runs can be checked for consistency by keeping the valid time fixed and changing the initialisation time. A distinct feature is the ensemble navigation. The user can select a specific forecast member for exploration, animate over members and toggle the ensemble mean for all currently displayed data fields (Fig. 5.1e).

Visual entities such as a horizontal or vertical cross-section, the base map or a 3D isosurface are represented by *actors* and are assigned to a *scene*. A scene, in other words a collection of actors, can be assigned to one (or more) of the views for rendering. An actor can be part of multiple scenes. For example, a cross-section could be viewed as a traditional 2D image in one view, and be combined with a 3D isosurface in another. If the section is relocated, its position is updated in both views. Default instances of the actors can be created by the user at runtime. They can be configured to represent a specific forecast product (for example, a horizontal section displaying geopotential height and wind speed) and saved to file. The saved actor can be restored in subsequent sessions. To keep the user interface simple, properties that the user can modify for a particular actor (for instance, the isovalue of an isosurface, the forecast variable displayed by an actor, or the associated colour palette) are arranged in a tree-like, hierarchical structure on the left of the Met.3D window (Fig. 5.1f). Basic properties (for instance, to load predefined forecast products) are arranged on the uppermost tree nodes. Properties that require expert knowledge from the user can be hidden at deeper levels.

Trafton and Hoffman (2007) point out the importance of visual comparisons in the forecasting process. Met.3D's actors can be synchronized in time and ensemble dimension, its views can be synchronized to the same camera viewpoint. Thus, side-by-side comparison of different data sets is facilitated.

5.2.2 A bridge from 2D to 3D

To help forecasters transition to the 3D visualization environment, I have implemented horizontal and vertical 2D sections. The sections reproduce the look of the corresponding products in the DLR MSS (Rautenhaus et al., 2012), providing filled and line contours, wind barbs, coast lines and graticule. In Met.3D, the sections are embedded into the 3D context and can be interactively moved in space by the user in real time. This provides a very fast means to explore the atmosphere's vertical structure (by sliding a horizontal section up and down), or the change in forecast variables along a flight track when a waypoint is relocated (by moving a vertical section). Also, the camera can be moved interactively to zoom in, pan, or tilt the view – for instance, to view multiple sections stacked on each other from an angled viewpoint. Figure 5.2 illustrates the concept. The forecast wind field is visualized by means of a horizontal and vertical section. The horizontal map –largely resembling the corresponding product from the MSS– is stacked on top of surface level contours displaying the mean sea level pressure (Fig. 5.2b). The vertical section is augmented by a 3D isosurface of wind speed (Fig. 5.2c); the isovalue is

[3] Note that this is a prerequisite for displaying horizontal sections on constant pressure surfaces. Some other tools, including Vapor, use geometric height as the vertical coordinate.

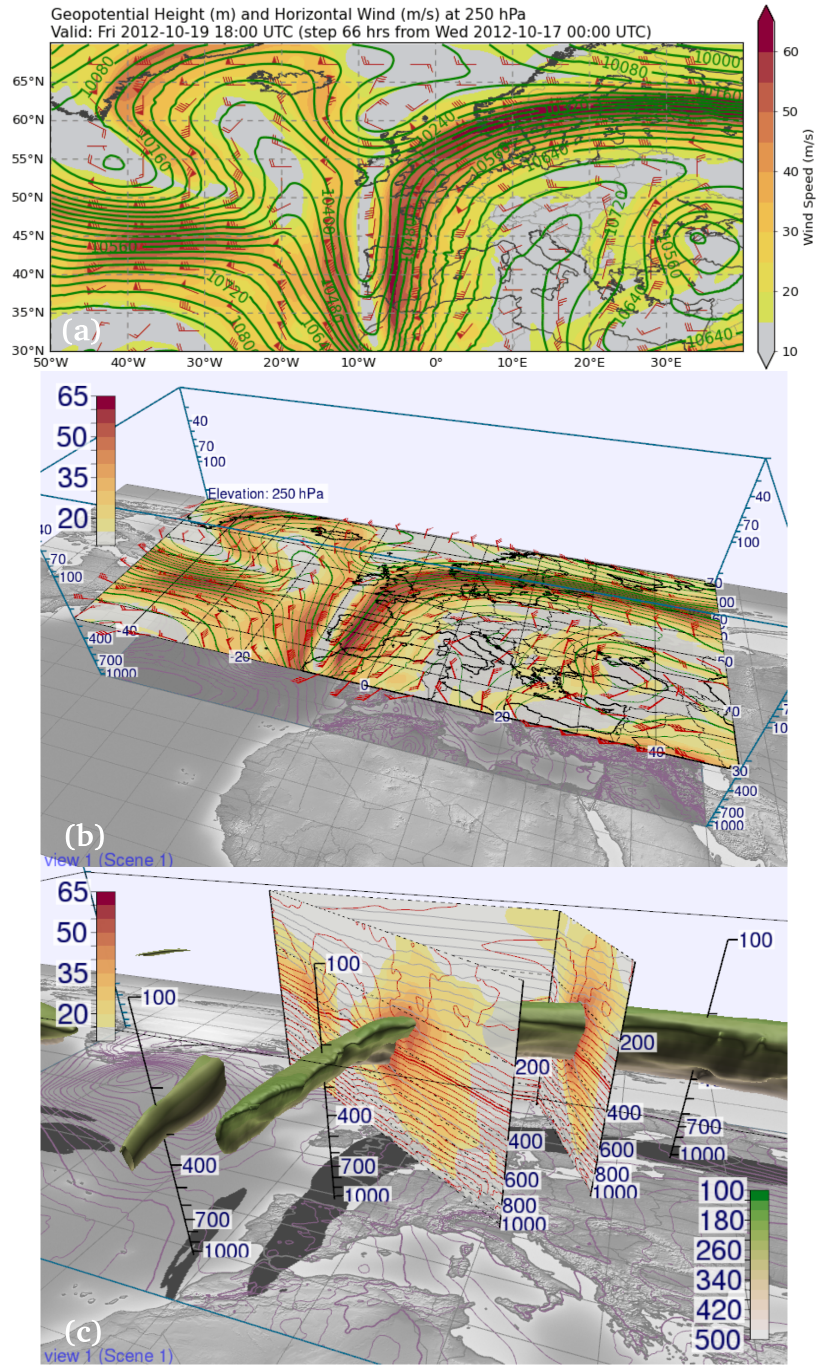


Figure 5.2: Bridge from 2D to 3D visualization. **(a)** Horizontal section of geopotential height (contour lines) and horizontal wind speed (colour) at 250 hPa, as obtained from the DLR Mission Support System. ECMWF deterministic forecast from 00:00 UTC 17 October 2012, valid at 18:00 UTC 19 October 2012. **(b)** The same data, rendered by Met.3D and mapped into the 3D context. The section can be interactively moved by the user. **(c)** Vertical section of horizontal wind speed (colour) and potential temperature (contour lines) in Met.3D, amended by a 50 ms^{-1} isosurface of wind speed, coloured by pressure (hPa). Note how spatial perception of the 3D isosurface is aided by rendering shadows and labelled vertical poles (animated version of this figure in Video 1 (see Sect. 1.5) at 00:05 min).

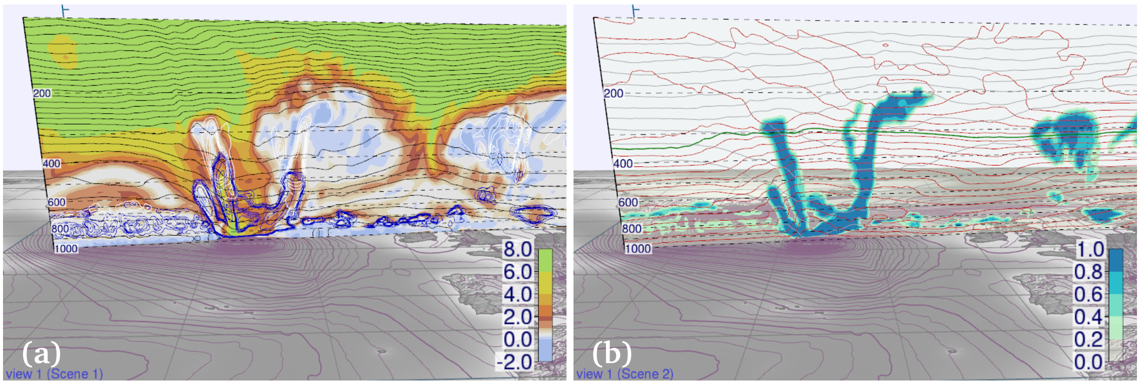


Figure 5.3: Vertical sections can be moved interactively in Met.3D to explore the vertical structure of the atmosphere, for example along potential flight track segments. **(a)** Potential vorticity (colour coding in PVU), **(b)** cloud cover fraction. Red colours in (a) mark the 2-PVU surface and thus the dynamic tropopause. Note the low tropopause along the trough. Same forecast as in Fig. 5.2 (animated version of this figure in Video 1 at 01:24 min).

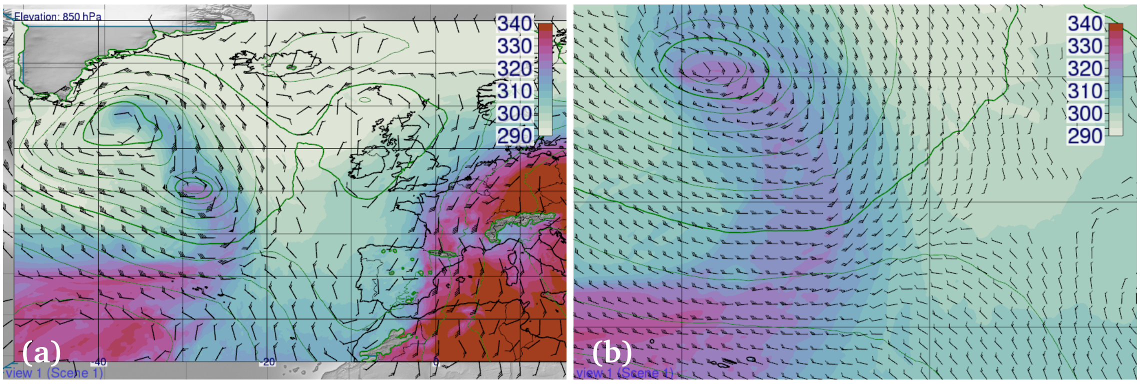


Figure 5.4: Met.3D automatically scales size and density of wind barbs overlain on horizontal sections. **(a)** and **(b)** Equivalent potential temperature (colour coded in K) at 850 hPa, overlain with contour lines of geopotential height. Same forecast as in Fig. 5.2 (animated version of this figure in Video 1 at 01:54 min).

chosen such that the strongest winds of the jet stream, an important indicator for the large scale flow of the upper troposphere, are captured. The 3D display allows us to locate the vertical section in space and additionally provides information on the spatial structure of the jet.

I approach the challenge of spatial perception by drawing projections of all rendered structures to the surface to imitate shadows generated by a directional light source above the scene. As illustrated in Fig. 5.2b and c, the shadows help to qualitatively judge the elevation of a feature, and also show its horizontal location (cf. Sect. 3.1.3 and Fig. 3.6). To improve the quantitative judgement of elevation, the user can colour the isosurface according to pressure elevation, and place vertical poles in the scene that provide labelled pressure axes (Fig. 5.2c). The poles can be interactively moved in the scene (by picking and dragging handles that appear in an *interaction mode*), so that different locations can be probed.

Vertical sections can be drawn along an arbitrary number of waypoints (Fig. 5.2c). Analogous to vertical poles, each waypoint and section segment displays a handle in interaction mode that the user can drag to move the waypoint or segment. They can also be moved synchronously in multiple scenes, as illustrated in Fig. 5.3. Displayed are sections of potential vorticity (Fig. 5.3a, the red colours around values of 2 PVU show the dynamic tropopause) and cloud cover fraction (Fig. 5.3b). Wind barbs

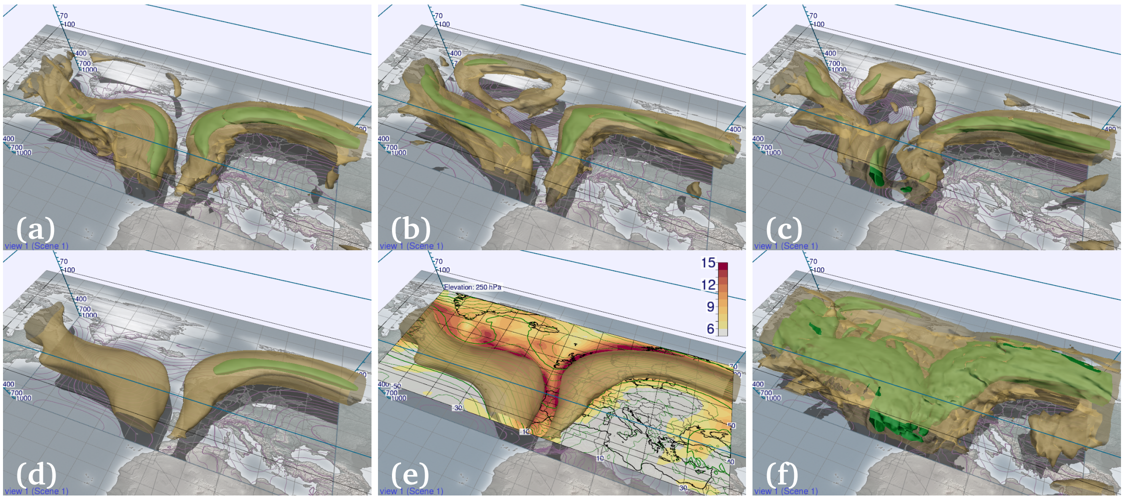


Figure 5.5: Navigation through the ensemble. Visualized are the 50 ms^{-1} (green opaque) and 30 ms^{-1} (yellow transparent) isosurfaces of horizontal wind speed (forecast from 00:00 UTC 15 October 2012 valid at 18:00 UTC 19 October 2012). **(a)** Control run, members **(b)** 27 and **(c)** 33, **(d)** ensemble mean, **(e)** ensemble mean augmented by a horizontal section of standard deviation (ms^{-1}), **(f)** ensemble maximum (animated version of this figure in Video 1 at 02:26 min).

overlain on a horizontal section can be configured to automatically scale in size and density. In Fig. 5.4, the horizontal section of equivalent potential temperature shows the different character of airmasses transported by Rafael. When the user zooms into the view, Met.3D increases the density of the wind barbs (Fig. 5.4b). The frontal zone along which the typical change in wind direction occurs can now be well perceived.

As described in Sect. 3.1.3, it is important to address perceptual issues with respect to the use of colours in the visualizations (also cf. Hoffman et al., 1993). To map scalar value to colour, I have implemented the perceptually-based HCL colour space (cf. Sect. 3.1.3). Following Zeileis et al. (2009) and Stauffer et al. (2015), the user can create colour palettes by specifying ranges in hue, chroma and luminance. Alternatively, colours can be explicitly specified to reproduce colour bars the user is familiar with. An example is the colour palette for potential vorticity shown in Fig. 5.3.

5.2.3 Ensemble support

Ensemble support plays a central role in Met.3D. The tool enables the forecaster to explore variation in the ensemble, to identify regions in which the forecast is uncertain, and to explore possible forecast scenarios. The user can interactively navigate through the ensemble members to judge the variability in the forecast. Judging uncertainty based on such animation bears resemblance to the widely used stamp maps (cf. Fig. 2.9), with side-by-side display traded for animation. With animation, however, regions of high fluctuation naturally attract attention. Each member can also be explored individually. Products that display statistical quantities, as shown in Figs. 2.12 and 2.13, are available. In Met.3D, the statistical quantities can also be visualized in 3D. Furthermore, statistical measures including threshold probabilities, mean, minimum, maximum and standard deviation can be derived on-demand. For threshold probabilities (for example, wind speed exceeding 45 ms^{-1} or cloud cover fraction being below 0.2) the threshold value can be adjusted interactively.

Fig. 5.5 shows an example of exploring the upper level ensemble wind field of the forecast from

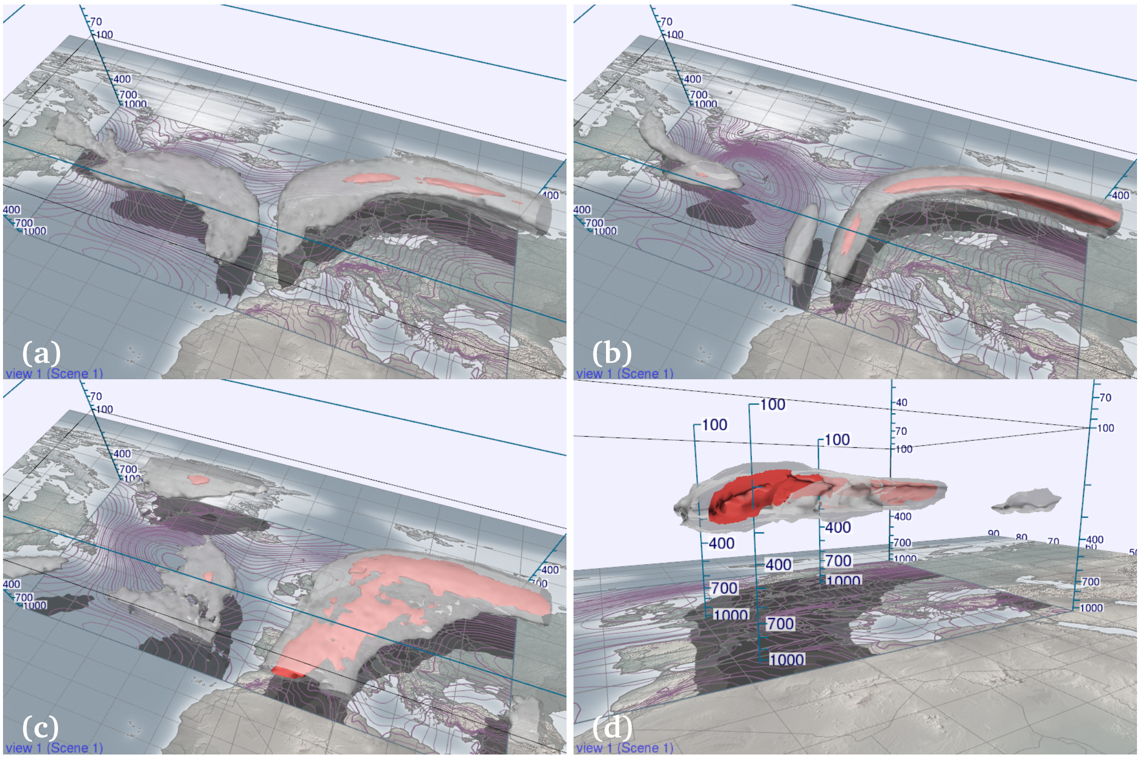


Figure 5.6: Probability fields computed from the ensemble, valid at 18:00 UTC 19 October 2012. **(a)** and **(b)** Probability of horizontal wind speed exceeding 50 ms^{-1}, as computed from the forecast initialised (a) at 00:00 UTC 15 October 2012 and (b) at 00:00 UTC 17 October 2012. Shown are the 70% (red opaque) and 10% (white transparent) isosurfaces. Note how the ensemble converges. **(c)** and **(d)** Probability of contrail occurrence (Schmidt-Appleman criterion fulfilled and relative humidity greater than 80%), as viewed from different camera positions (80% red opaque and 50 % white transparent) (animated version of this figure in Video 1 at 03:23 min).

Monday, 15 October 2012, 00:00 UTC, valid at Friday, 19 October 2012, 18:00 UTC. To visualize the jet stream, two wind speed isosurfaces are rendered. Animation attracts attention to large variation of the ensemble regarding position, structure, and strength of the jet stream over the Atlantic, highlighting high uncertainty in this area. On the other hand, the strong jet extending from Spain to Scandinavia is predicted with higher certainty: While in the mean wind field the 45 ms^{-1} signal over the Atlantic is largely smoothed out, it is present over Europe (Fig. 5.5d). However, adding a horizontal section of wind speed standard deviation (Fig. 5.5e) to the isosurface of mean wind speed reveals that the position of the jet is uncertain in particular on its northern side.

Fig. 5.6 shows the probability of wind speed exceeding 45 ms^{-1}. A high probability of over 70% can again be found over northern Europe (Fig. 5.6a). The large horizontal extent of the area of low (10%) probability above the Atlantic reflects the uncertainty. The actual jet can occur anywhere in this region. Two days later, with decreasing forecast lead time, the ensemble has significantly converged and the uncertainty has decreased (Fig. 5.6b). Fig. 5.6c and d shows the probability of the Schmidt-Appleman criterion (Schumann, 1996), an indicator for the occurrence of contrails (aircraft-induced clouds that also have been the target of research flights; Voigt et al., 2010; Kaufmann et al., 2014). Visualization of the probability of the Schmidt-Appleman criterion being fulfilled shows that contrails, in the example, can only occur between about 400 and 200 hPa. In the given case, a high probability can be observed on the leading downstream edge of the jet.

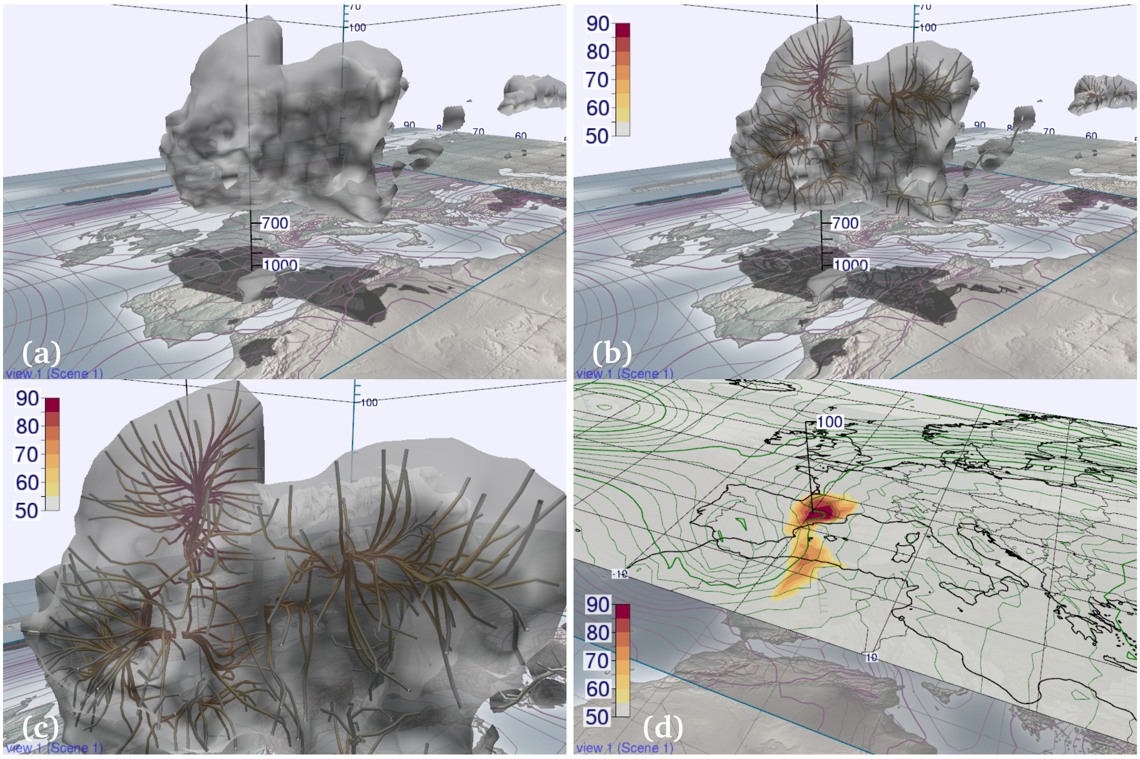


Figure 5.7: Normal curves help to analyse the topology of 3D scalar fields. They reveal the distribution of data values in a subdomain enclosed by a 3D isosurface and enable fast identification and tracking of local extrema. **(a-c)** Probability of cloud ice water content exceeding 0.01 gkg^{-1}. The white transparent isosurface shows 40% probability. Colour coding in %. **(d)** Details of the identified maximum are inspected with a horizontal section at 250 hPa. Forecast from 00:00 UTC 17 October 2012 valid at 12:00 UTC 20 October 2012 (animated version of this figure in Video 1 at 04:28 min).

5.2.4 Normal curves

In the volume visualizations shown in Figs. 5.5 and 5.6, the structure of the scalar fields inside the transparent isosurfaces cannot easily be inferred. As stated in Sect. 4.2, this is a disadvantage of 3D visualization: While an isosurface allows inference on the three-dimensional spatial structure of the displayed data field, it only displays a single data value. Although two or three isosurfaces can be rendered in a single image using transparency, the image quickly becomes illegible when more surfaces are used. *Normal curves* were used by Pfaffelmoser et al. (2011) to estimate the spatial distance between two isosurfaces. In 2D, the technique already appears in Saucier (1955) as a graphical aid to clarify the depiction of features in scalar fields, and to identify maxima, minima and saddle-points (cf. Saucier, 1955, Fig. 4.07d). For application in Met.3D, I propose to use *3D normal curves* as an intermediate means between a 2D section and a 3D isosurface to visualize the structure of scalar fields in the interior of an isosurface.

The curves are started on a transparent isosurface and proceed along the field's gradient direction, i.e. normal to the isosurface. The spacing of the curves can be controlled by the user (cf. Sect. 5.3.4). I colour the curves according to the scalar value. This way, a visual sampling of a subdomain of the volume is achieved. In contrast to a 2D section that samples a planar subdomain, the normal curves sample a 3D subdomain enclosed by an isosurface via a discrete set of lines. Following the gradient, the curves converge at local extrema of the data field. This way, the user can at a glance identify the locations and strengths of present extrema, and judge the strength and direction of the gradient between an extremum and the outer isosurface.

Figure 5.7 illustrates the approach. The goal is to identify regions of maximum probability of cloud ice water content exceeding 0.01 gkg^{-1}, and to track the regions' evolution over time. The normal curves immediately show a maximum in the upper part of the transparent 40% isosurface (Fig. 5.7b,c). The corresponding shadows reveal that the maximum is approximately located above the Pyrenees. Interaction with the vertical axis shows a vertical position between 300 and 200 hPa. Further visual aids can now be added to obtain more quantitative information. In the example, the horizontal section can be immediately placed in the region of interest, without the need to search the entire vertical extent of the model atmosphere (Fig. 5.7d).

While extrema can also be identified with an inner opaque isosurface (cf. Fig. 5.6) or by interacting with 2D sections, the normal curve approach requires less interaction steps. This is advantageous if the absolute values of the extrema are not known beforehand (with isosurfaces the user needs to search over isovalues), and if the extrema shall be visually tracked over ensemble members or time. Concerning time, in particular probability values tend to decrease with increasing forecast lead time, hence a fixed isosurface is not well suited to visualize the temporal evolution of a maximum.

In Fig. 5.1c (also shown in the video at 05:40 min), the method is applied to the upper level wind field shown in Fig. 5.5. Here, the normal curves inside the 45 ms^{-1} isosurface converge to the string-like line of local maxima in the wind field – the curves are used to identify the position of the jet core and its strength.

5.2.5 Trajectories

In Chapter 6, Lagrangian particle trajectories (called path lines in the visualization literature, cf. Sect. 3.1) are used to detect WCB structures. Met.3D provides support to display trajectories output by the LAGRANTO model (Wernli and Davies, 1997; Sprenger and Wernli, 2015). The model has been widely used in the atmospheric science literature to compute trajectories from ECMWF wind fields (cf. Sect. 6.2). Trajectories are supported by a number of the surveyed existing meteorological visualization tools as well (cf. Sect. 4.2.2). New features in Met.3D are the support for ensemble trajectories, for on-demand filtering of trajectories according to some pre-defined criterion, and for shadows cast to the surface.

Trajectories are rendered as 3D tubes. The positions of the trajectory particles at a given time step can be displayed as 3D spheres. Figure 5.8 shows examples of trajectories seeded at regular intervals in the lower atmospheric levels. In Fig. 5.8a, the complete set of trajectories is rendered, in Fig. 5.8b and c, two different settings of a selection according to a given ascent per time interval are applied (the criterion can be changed interactively). Trajectories filtered according to an ascent of more than 500 hPa in 48 hours can be interpreted as representing a WCB (details are provided in Chapter 6). Figure 5.8d, e, and f shows snapshots of an animation through the ensemble.

To avoid pre-computation of the trajectories, an integration of LAGRANTO into Met.3D is presented in the Bachelor thesis by Gehring (2015). The approach integrates LAGRANTO's computational core for ECMWF forecasts and enables the user to define seed regions with point, line, planar or volume geometry. Trajectory computations are triggered from the Met.3D user interface, the results are visualized

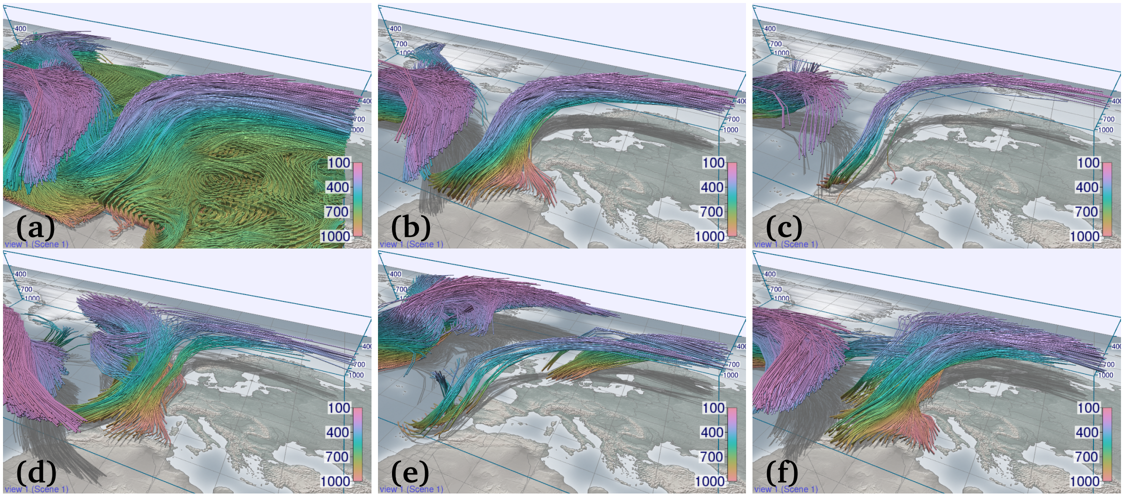


Figure 5.8: Trajectories can be interactively filtered according to a predefined criterion. Shown are trajectories started at regular intervals below 700 hPa. Forecast from 00:00 UTC 15 October 2012, trajectories are started at 06:00 UTC 18 October 2012 and integrated forward in time for 48 hours. **(a)** All trajectories of the control forecast. **(b)** Trajectories of the control forecast selected according to an ascent of 500 hPa in 48 hours. **(c)** The same as (b) but with an ascent of 650 hPa. **(d-e)** The same as (b) but for members (d) 6, (e) 33, and (f) 49.

using the trajectory actor.

5.2.6 Diagrams and interactive analysis

Diagrams of vertical profiles at a given horizontal location are another standard type of 2D display used in weather forecasting (e.g. Stull, 2000, 2014). A particular type frequently used is the *Skew–T diagram*, whose name originates from skewed isotherms plotted as part of the displayed coordinate system. The diagram displays vertical profiles of temperature and dew point temperature (the latter a measure of moisture in the air) on top of a number of isolines of thermodynamically relevant quantities. It is used to analyse the vertical structure and stability of the atmosphere.

The Bachelor thesis by Heidelmann (2015) describes how Skew–T diagrams are rendered to Met.3D's 3D context. The diagrams are attached to a vertical pole and can, similar to the moveable poles, be moved interactively in the scene (Fig. 5.9). As for horizontal and vertical sections, the interactivity provides a very fast means to explore the spatial variation of the plotted profiles. Functionality to plot vertical profiles is also reported in the studies of Treinish and Rothfusz (1997) and McCaslin et al. (2000). However, the ability to interactively move the diagram in the 3D scene and the interactive ensemble support in Met.3D are new. Besides animation over ensemble members, we follow the approach by Ihász and Tajti (2011) and depict ensemble statistical quantities (for instance ensemble mean and standard deviation, see Fig. 5.9b) in the diagram.

The Met.3D framework allows for the implementation of further diagrams (for example, histograms) in a sidebar. To facilitate further analysis of visualized elements, an "analysis mode" is provided in which the user can select specific objects to trigger the execution of analysis techniques. An example will be presented in Sect. 6.4, where isosurfaces in the 3D view can be selected to obtain further information about the displayed data by means of a histogram in the sidebar.

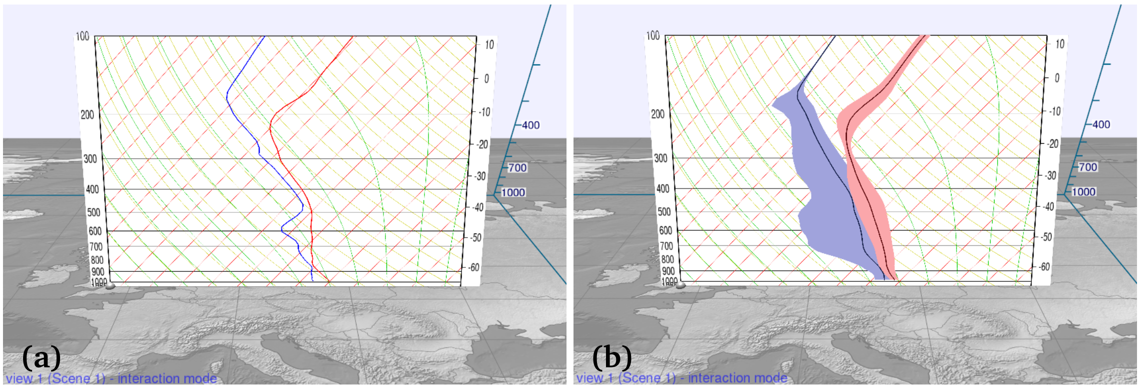


Figure 5.9: Interactively movable Skew–T diagram rendered into the 3D context, showing a vertical profile at a location in southern England. **(a)** Vertical profiles of temperature (red) and dew point temperature (blue). Control forecast from 00:00 UTC 15 October 2012 valid at 18:00 UTC 19 October 2012. **(b)** Ensemble mean of temperature (red) and dew point temperature (blue), plotted in an envelope showing the standard deviation of both quantities at the respective elevations.

5.2.7 Overview of actors

The previous subsections have illustrated the capabilities of Met.3D. In the following, the actors available in Met.3D at the time of writing are summarised. A visual overview is provided in Fig. 5.10.

- A number of actors create the georeferenced and spatial context for those actors that visualize NWP data. The *graticule actor* (Fig. 5.10a) renders a graticule and coast and country borders. Similarly, the *base map actor* (Fig. 5.10b) displays a georeferenced image as a base map. The *volume bounding box actor* (Fig. 5.10c) adds a rectangular bounding box to the scene that helps to confine the region covered by available NWP data. Finally, the *movable poles actor* (Fig. 5.10d) provides labelled vertical axes similar to those of the volume bounding box but that can be interactively moved by the user.

- The *1D transfer function (colour bar) actor* (Fig. 5.10e) provides a mapping from a scalar value to a colour and a transparency value. The actor is used in conjunction with the actors displaying NWP data. Colour maps can be defined using the HCL approach described by Zeileis et al. (2009) and Stauffer et al. (2015) (cf. Sect. 3.1.3). Alternatively, a number of colour maps from the Python matplotlib package (that is used in the MSS; cf. Sect. 4.2.3) are available.

- The *surface topography actor* (Fig. 5.10f) provides basic terrain rendering, displaying the surface pressure field as terrain. Using a transfer function actor, the terrain can be coloured according to some other 2D NWP variable, for example, 2m temperature.

- 2D maps and vertical cross-sections are provided by the *horizontal section actor* (Fig. 5.10g) and *vertical section actor* (Fig. 5.10h). These actors are designed to resemble the look of the corresponding sections in the MSS. Both actors can display an arbitrary number of forecast variables, each of which can be visualized using line and filled contours (the latter with a transfer function) or pseudo colour plots (depicting the model grid boxes). As described above, the horizontal section actor also provides the possibility to render wind barbs. Both horizontal and vertical sections can be interactively moved in real-time. A shadow of the horizontal section can be mapped to

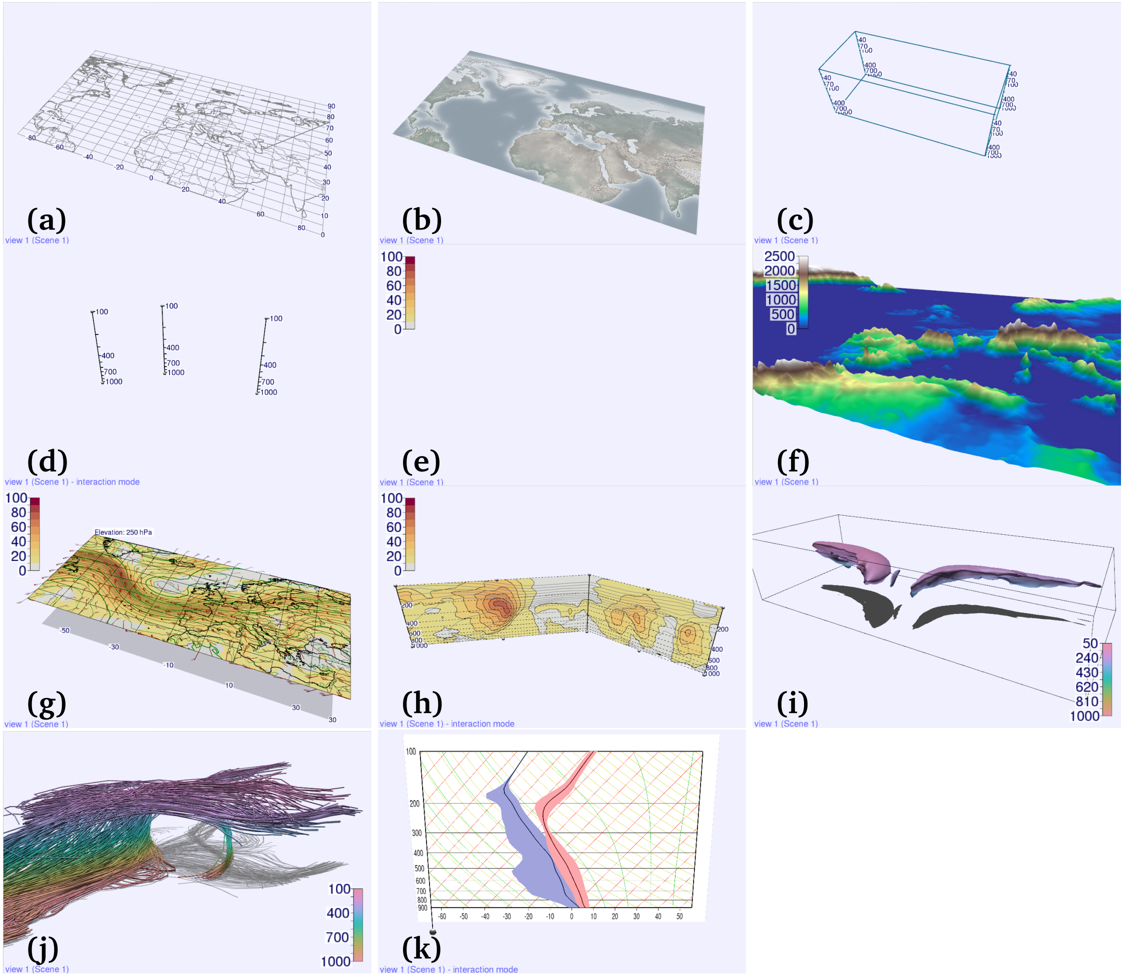


Figure 5.10: Overview of actors (the visual entities) available in Met.3D at the time of writing. **(a)** Graticule. **(b)** Base map. **(c)** Volume bounding box. **(d)** Movable poles. **(e)** Colour bar (1D transfer function). **(f)** Surface topography. **(g)** Horizontal section. **(h)** Vertical section. **(i)** Volume isosurface, including normal curves. **(j)** Trajectories. **(k)** Skew–T diagram.

the bottom of the scene. For the horizontal section actor, labels showing contour values can be displayed at the boundaries of the section.

- The *volume actor* (Fig. 5.10i) provides functionality to render transparent and opaque isosurfaces, as well as normal curves. Isosurfaces can be coloured according to some other NWP variable (using a transfer function). Normal curves can be coloured according to the scalar field from which they are computed. Shadows of both isosurfaces and normal curves can be mapped to the bottom of the scene.

- Lagrangian particle trajectories (path lines) are rendered by the *trajectory actor* (Fig. 5.10j). The actor supports trajectories from the LAGRANTO model (Sprenger and Wernli, 2015). The trajectories are rendered as 3D tubes coloured by pressure elevation. Shadows of the tubes can be mapped to the bottom of the scene. Since a trajectory is a time-integrated line, the positions of the particles at a specified time can be displayed by 3D spheres. By animating the spheres over time, the user can visualize their temporal evolution. Trajectories can be filtered according to a

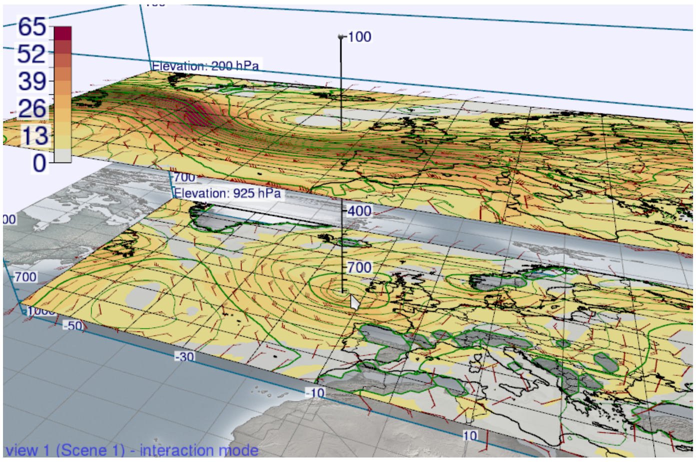


Figure 5.11: Example of two instances of the same actor in one scene. Shown are two instances of a horizontal section of geopotential height and wind speed (same configuration as in Fig. 5.2b), rendered at different altitudes (200 hPa and 925 hPa). A movable vertical axis is used to illustrate the relation between the low pressure core visible in the lower section and the jet-level winds in the upper section. Green contour lines show geopotential height, colour coding in ms^{-1}.

pre-defined criterion.

- The *vertical profile actor* (Fig. 5.10k) displays interactively movable Skew–T diagrams in the 3D context. The diagrams depict vertical profiles of temperature and dew point. Following the approach by Ihász and Tajti (2011), ensemble information are displayed by showing the ensemble mean and/or a selected member in an envelope of an ensemble statistic (for instance, minimum and maximum or standard deviation).

I note that while the visualization examples in Figs. 5.1 to 5.10 display only single instances of the presented actors, multiple instances are possible as well. As an example, Fig. 5.11 shows two horizontal sections at different elevations. A vertical axis is used to illustrate the relation between a low pressure core at low elevations and the jet stream at upper levels.

5.3 System architecture and visualization algorithms

After the visualization capabilities of Met.3D were introduced in the previous section, this section describes their methodological and technical realisation. I present the architectural concept of Met.3D and discuss the methods used to achieve the goal of continuous-interactive visualization performance. Following the discussion in Sect. 5.1, I propose a pipeline architecture that focusses on the determined goals. It accounts for the ensemble dimension and forecast initialisation and valid time dimensions, utilises pipeline and task parallelism, and features GPU-based mapping and rendering. In Sect. 5.3.1, I extent the list of goals from Sect. 5.1 by analysing data requirements as well as types of visualization tasks that need to be supported by the architecture. Based on this analysis, the system design and the ensemble-aware data pipeline are described in Sects. 5.3.2 and 5.3.3. The GPU-based visualisation algorithms are presented in Sect. 5.3.4.

5.3.1 Requirements

Met.3D's architecture builds on a number of assumptions and requirements with respect to data used, tasks performed, and to the operation environment in which the tool is deployed.

Data assumptions

I make a number of assumptions with respect to structure and volume of the forecast data that are used with Met.3D. As noted in Sect. 1.3, the thesis focusses on supporting forecast data from the ECMWF ENS and IFS (described in Sect. 2.4), as these are primarily used during DLR field campaigns. One of the design goals (cf. Sect. 5.1) is to natively support the data in the structure and file format in which they are provided by ECMWF's MARS archive and used in the data processing system of the DLR MSS (that is, on a regular grid in the horizontal and hybrid sigma-pressure coordinates in the vertical). This requirement minimises the time span between data availability at ECMWF and visualization by avoiding preprocessing of the data prior to usage in Met.3D. Forecast data hence needs to be read directly from GRIB files output by MARS or from NetCDF-CF[4] files (as used in the MSS). This is in contrast to many studies on GPU-based visualization that preprocess and reorganise data to be able to read it from disk in the most efficient way (e.g. Fraedrich, 2012; Treib, 2014; Reichl, 2014, and references therein). Also, for example, Vapor preprocesses its datasets into the wavelet representation (cf. Sect. 4.2.2; Clyne, 2012).

The architecture also needs to support the on-demand computation of forecast parameters not output directly by the ECMWF model (for example, relative humidity and potential vorticity). Of course, pre-computation should also be possible. For this purpose, Met.3D can be connected to the data processing system of the MSS, which derives additional quantities from the forecast parameters output by ECMWF.

The examples in this thesis use ENS data interpolated horizontally to both 1°×1° and to 0.25°×0.25° in longitude and latitude. 1°×1° is the grid spacing we were able to operationally retrieve during TNF, as permitted by the available internet bandwidth and interpolation time required by MARS. The higher resolved deterministic data is used at 0.15°×0.15° grid spacing. In the vertical, all 62 (ENS) and 91 (deterministic), respectively, levels available during TNF are used (cf. Sect. 2.4). The forecast domain used in the examples encompasses 100° in longitude by 40° in latitude, resulting in 101×41×62 grid points for ENS data fields at 1°×1° grid spacing, 401×161×62 points at 0.25°×0.25° grid spacing, and 669×268×91 points for the deterministic forecast at 0.15°×0.15° grid spacing. Using floating point precision (4 bytes per value), the data fields require approximately 1 MB, 16 MB and 62 MB per member, time step, and forecast parameter in graphics memory.

While this domain covers only a fraction of the globe, from my experience at DLR, its extent can be considered representative for field campaigns. The important conclusion for the Met.3D architecture is that the data field of a single variable, a single member, and a single time step can be considered small enough to fit into both CPU and GPU memory. Such data fields can thus be considered as the smallest data entity in Met.3D. It can be stored in a single 3D texture, and visualization algorithms can be assumed to have full spatial access to a data field. If a domain four times as large is used, the data sizes will increase to 4 MB (ensemble at 1°), 64 MB (ensemble at 0.25°) and 248 MB (deterministic at 0.15°), respectively. The memory of recent GPUs (cf. Table 3.1) is still large enough to fit multiple variables of this size. Nevertheless, while not in the scope of this thesis, the architecture of Met.3D

[4] `http://cfconventions.org`

Table 5.1: Overview of types of tasks that need to be feasible with a visualization pipeline architecture targeting ensemble weather prediction data, along with requirements for the efficient implementation of such tasks. "Time" refers to both forecast valid and initialisation time dimensions. Requirements marked by a "*" have been outside the scope of this thesis, however, are discussed in the text.

Tasks requiring access to a single variable/time step/ensemble member	
	Pipeline modules that require only a single data item from their input at a time. For instance, regridding of a single data item, or visualization of an isosurface of a single scalar field.
Tasks requiring progressive access to time steps/ensemble members	
	Modules that require multiple time steps, ensemble members, or variables, but only one at a time. Examples include the computation of ensemble mean and standard deviation, and the computation of trajectories (although for the latter two time steps may be required at a time to interpolate in time).
Tasks requiring concurrent (random) access to multiple variables/time steps/ensemble members	
	Modules that require random access to arbitrary fields in the forecast dataset. Examples include the combined visualization of multiple forecast parameters, comparative visualization of different time steps or members, and ensemble statistics that only consider selected ensemble members.
Requirements for the efficient execution of the task groups in a pipeline	
a	Global memory management for both CPU and GPU memory, caching of as many intermediate results as possible (e.g. to account for dynamic user behaviour such as repeatedly navigating back and forth in time or ensemble member).
b	Thread-safe algorithms and memory management to facilitate task and pipeline parallel execution via multi-threading where possible.
c	Possibility to use GPU shaders in algorithms.
d	Item "c" requires all data items to be GPU aware, that is some relation between CPU and GPU representation of the data item.
e	Variable, time and ensemble metadata is available everywhere within the pipeline.
f	No overhead for algorithms that make no use of particular dimensions.
g*	Support for pre-loading and pre-execution of likely upcoming requests and storage of results in cache, if memory is available.
h*	Support for streaming via domain-decomposition.

should be designed to facilitate future extensions to accommodate streaming algorithms (cf. Sect. 3.3) for datasets that do not fit into memory in their entirety.

Pipeline time-ensemble requirements

Besides data size and grid topology, expected visualization and processing tasks pose requirements. In a "classical" visualization pipeline implementation, as depicted in Fig. 3.9, each module processes one request at a time. The pipeline may fan in and out, but with requests propagating upstream from the sink, each module requests at most one (possibly none) data item from each of its inputs. This approach works well for visualization techniques that do not require information about the time or ensemble behaviour of a dataset. Many techniques possess that property, for example, contouring, volume rendering, and slicing (cf. Biddiscombe et al., 2007).

However, a specific characteristic of visualizing weather forecast data is that for many operations, multiple variables, time steps, or ensemble members are required. If pipeline modules need to request multiple data fields from an input, a simple architecture as shown in Fig. 3.9 is not sufficient. Consider, for example, the computation of ensemble statistics and of trajectories. Both operations require the module that implements the algorithm to request multiple members or time steps from its input. Hence, some sort of request looping or random access mechanism is required.

In the context of adding support for time-dependent processing to VTK, Biddiscombe et al. (2007) analyse the requirements that time-dependent visualization tasks pose to a pipeline architecture. They identify different groups of tasks, including tasks that require progressive access to time steps (requiring multiple time steps but only one at a time), tasks that require contiguous chunks of time steps at the same time, and tasks that require random access to arbitrary time steps. To enable time-dependent processing in the VTK architecture, Biddiscombe et al. add support for looped execution of sub-pipelines as well as a time-cache module that is able to hold multiple time steps in memory.

Following Biddiscombe et al. (2007), groups of tasks can also be identified when in addition to a single time dimension (say the forecast valid time), the ensemble dimension, the forecast initialisation time dimension, and multiple forecast variables are considered as well. Consider that access to a forecast dataset with multiple variables (each of which have the considered dimensions) may conveniently be represented by a single reader module in the pipeline. A subsequent module computing, for instance, an ensemble statistic, hence needs the possibility to request multiple ensemble members. Similarly, a module that computes a derived forecast parameter (for example, relative humidity or potential vorticity) requires access to multiple input variables. Also, Met.3D actors, sinks in the pipeline, frequently request multiple variables at the same time (consider, for example, the four variables that are requests if the displayed time step is changed for the horizontal section in Fig. 5.2).

The pipeline architecture hence needs to support not only modules that (1) perform tasks requiring access to a single data field at a time, but also modules requiring (2) progressive access to multiple time steps or ensemble members (but only one at a time), and, most general, (3) random access to multiple forecast fields at the same time.

Table 5.1 summarises these groups and also lists a group that encompasses requirements that enable a pipeline to efficiently execute time and ensemble dependent algorithms. In particular, note the potential task and pipeline parallelism inherent in groups 2 and 3. Moreland (2013), focussing on the topology of the pipeline, notes that pipelines in real working environments will seldom be complex enough to provide many sections that can be processed in parallel. However, a specific characteristic of the tasks in groups 2 and 3 is that, by sending multiple requests into a pipeline or parts thereof, potential parallelism is introduced. While the organisation of modules in a pipeline might not provide parallel sections, spawning requests for multiple variables, time steps, or members do provide parallel tasks. To utilise this parallelism, however, the implementation requires thread-safe pipeline modules and a global execution model that supports the parallel execution of parts of the pipeline.

Operation environment

To be deployed in campaign settings, Met.3D needs to be optimised to standard commodity hardware with monocular displays, that is, regular desktop computers and notebooks. I assume the system to possess a multi-core CPU. Stereoscopic displays are outside the scope of this thesis. To be able to exploit the performance of recent graphics hardware, the computer is assumed to have a GPU capable of OpenGL 4 (equivalently, of DirectX 11). This is true for almost all GPUs released in and after 2010 (cf. Sect.3.2).

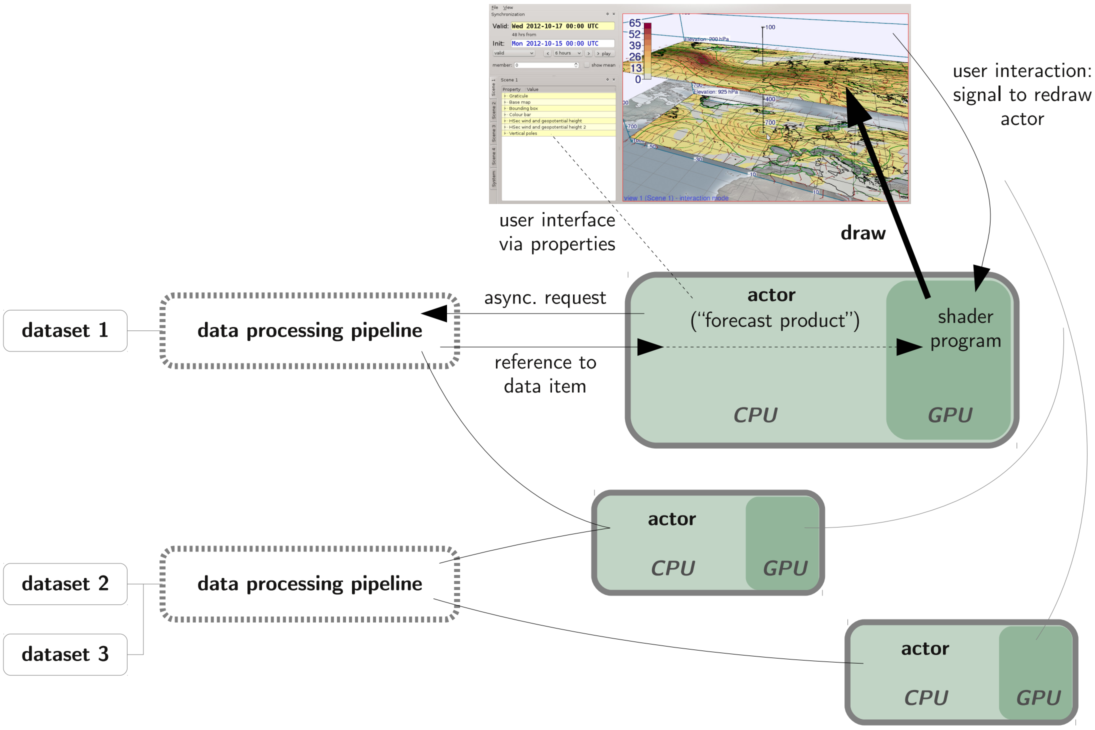


Figure 5.12: Illustration of the Met.3D system design. Actors consist of CPU and GPU parts and are connected to one or multiple data processing pipelines. Requests for data fields are sent asynchronously into the connected pipeline(s). The GPU part of an actor implements its visualization technique and renders the actor into the scene. User interface elements are provided as properties in the property tree on the left of the Met.3D window (cf. Sect. 5.2). See text for further details.

5.3.2 System design

Figure 5.12 illustrates the high-level architecture of Met.3D. An overview of the basic objects in the architecture is provided in Table 5.2. With respect to the conceptual pipeline model in Fig. 3.1, Met.3D's *data processing pipeline* covers the data acquisition and filtering stages, while the *actors* cover the mapping and rendering stages. From a conceptual point of view, Met.3D's pipeline can also accommodate mapping tasks. However, for performance reasons, the GPU-based visualization algorithms for the most part directly operate on the forecast data and are thus a combination of mapping and rendering.

Figure 5.12 shows a few possible scenarios. An actor can be connected to one or multiple pipelines, which in turn provide (processed) data of one or multiple datasets on a hard-drive (with a dataset typically including all forecasts of a particular model at a fixed grid topology). If an actor requires a new data item (for example, after the user has changed a property), a request is sent to the corresponding pipeline(s). The request is sent asynchronously; that is, the actor does not wait for a data item to be delivered. Instead, when the item is available, the actor is notified. Each actor consists of a CPU and a GPU part. Upon notification of available data, the actor receives a reference to the corresponding data item (which is memory managed, so that it can be shared and cached). The data item, in turn, contains references to both CPU *and GPU* memory of the data. The latter reference is passed to the GPU part of the actor, which draws the visualization to the viewport. GPU acceleration is implemented with OpenGL 4 and GLSL, using vertex, geometry, fragment, and compute shaders (cf. Sect. 3.2). On the CPU side, Met.3D is implemented in C++ using the Qt 4 toolkit (e.g. Blanchette and Summerfield,

Table 5.2: Basic objects of the Met.3D architecture, their purpose and major properties. See text for further details.

object	purpose	important properties
data item	smallest data entity in Met.3D, e.g., the 3D grid of a single forecast parameter of a single member at a single time step, or a set of trajectories	provides methods to upload the data to GPU memory
data source	implements an algorithm that generates a data item by reading data from disk or by processing input data items (e.g., computation of the ensemble mean)	provides methods to request, get, and release data, as well as to generate a task graph required to generate a request; implements a thread-safe method for the actual computation
request	unique reference to a data item that contains all information required by a data source to generate the data item	
actor	implements a visual entity (cf. Fig. 5.10), e.g., to visualize a data item obtained from a data source	communicates with data sources by sending asynchronous requests
task	container object that keeps a reference to a data source and a request (i.e. a “task” to be processed); connected tasks constitute a task graph	provides adjacency lists that link to further tasks that together may resemble a directed, acyclic task graph
memory manager	caches data items in CPU and/or GPU memory, using their request as key	provides methods to store, get, and release a data item, and to test whether a data item is stored for a given request
scheduler	schedules a task graph for execution by workers	
worker	executes a task	each worker runs in a separate thread to facilitate parallel execution of tasks

2008).

In the following, I describe the details of the architecture. The data processing pipeline is discussed in Sect. 5.3.3, and the GPU-based techniques implemented in the actors are described in Sect. 5.3.4.

5.3.3 Data processing pipeline

The data processing pipeline is composed of modules (called *data sources*) that create, read or process data and that can be combined in flexible ways. Fig. 5.13 illustrates the concept. Algorithms in the data sources (for example, ensemble statistics or trajectory filtering, cf. Chapter 6) can be implemented to execute on either CPU or GPU (the latter via compute shaders). All data sources are connected to a *memory manager* that caches intermediate results. The actors that implement the visualization methods are placed at the end of a pipeline. As noted above, they send *requests* into the pipeline to obtain a specific *data item*. These requests are composed of multiple key/value pairs similar to the Web

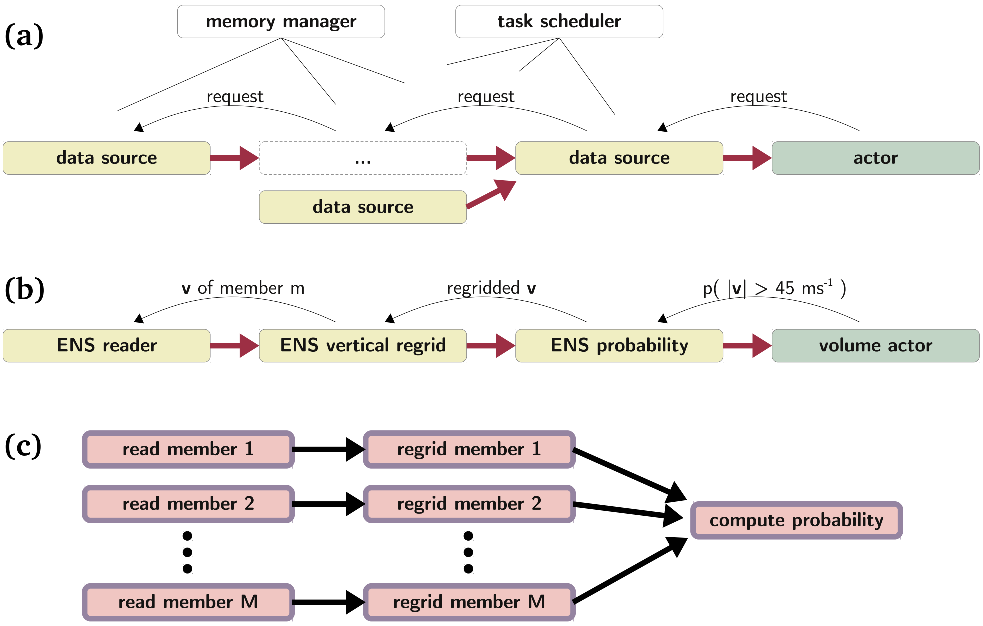


Figure 5.13: Pipeline concept of Met.3D: **(a)** Data sources are connected to form a pipeline, into which a visualization *actor* sends data requests. **(b)** Sample pipeline to visualize the probability of horizontal wind speed exceeding 45 ms^{-1}. A request for the probability triggers further requests up the pipeline. **(c)** Task graph generated by the pipeline in (b). Note that while the pipeline topology in (b) does not have parallel sections, the task graph does.

Map Service requests used in the MSS (see Rautenhaus et al., 2012, for details). A request emitted into a pipeline propagates from data source to data source. Each data source interprets the keys it requires. If the requested operation has been executed before and the result has been cached by the memory manager, no action is taken. Otherwise, the data source defines a processing *task* to perform the requested operation. The task, however, is not executed immediately. If applicable, remaining keys are passed on to the data source's input(s). If a data source requires additional input, it can also append keys to the request.

All processing tasks defined this way are assembled into a *task graph* that is passed to a *scheduler* for execution. Based on the dependencies provided by the graph structure and information carried by the tasks, the scheduler can process the tasks. For example, tasks that have to be performed for multiple variables/members/time steps of the forecast can be executed in parallel.

The task-graph-based, asynchronous approach with global memory management and parallel scheduling distinguishes my architecture from VTK. In VTK, more responsibility for memory management and execution is maintained by the modules, and missing thread-safety of the modules prevents task-parallel execution of simultaneous requests (Geveci and Schroeder, 2011; Biddiscombe et al., 2007). While Vo et al. (2011) enable task parallelism for pipelines that fan out, Met.3D's approach is more flexible with respect to the request dependent, dynamic branching introduced by parallelism in a task graph.

As an example, consider the simple pipeline depicted in Fig. 5.13b. The volume actor at the end of the pipeline emits a request for a scalar field containing the probability of horizontal wind speed exceeding 45 ms^{-1}. The module computing the probability field requires the wind field of each ensemble member, regridded to a common grid. Hence, requests for regridded data fields containing the members' wind speed are emitted and a task is set up to compute the probability from these fields. The regridding module, in turn, requests that the wind speed fields are read from disk by the reader module. For

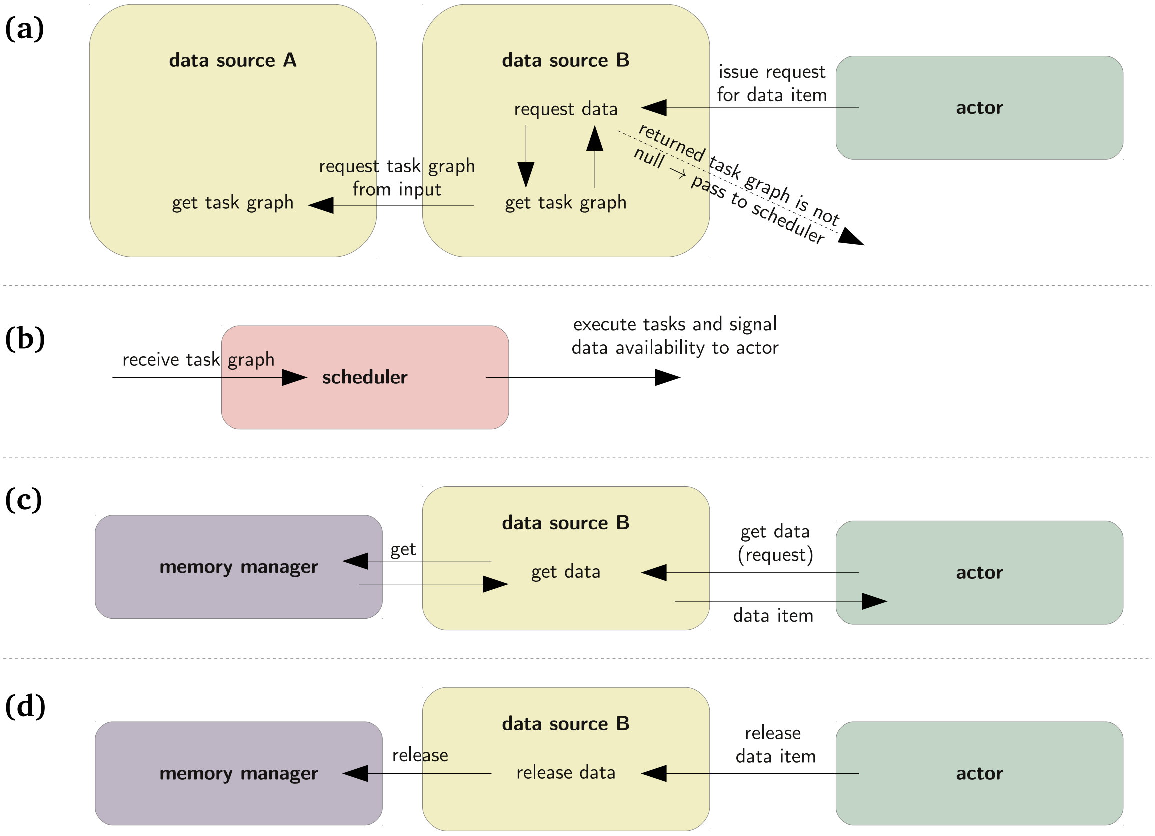


Figure 5.14: Schematic of a Met.3D data request. **(a)** An actor issues a request to a data source by calling the data source's *request data* method. The request is passed to the data source's *get task graph* method. This method generates a new task graph by creating a task for the data source itself and adding task graphs obtained from its input sources as dependencies. The resulting task graph is passed to the scheduler. **(b)** The scheduler schedules the tasks in the graph for execution (cf. Fig. 5.15). Once the data item of data source B has been generated, a signal is sent to the actor. **(c)** Upon reception of the signal, the actor obtains the data item by calling the data source's *get data* method. At that time, the data item is stored in the memory manager cache. The data source acts as a broker between memory manager and actor. **(d)** After the actor has finished using the data item, it notifies the data source. The data item is released in the memory manager (but remains cached until the memory is required by a new data item).

an ensemble of size M, the resulting task graph (Fig. 5.13c) contains M tasks to read the wind field of a single member, M tasks to regrid these fields to a common grid, and one task to compute the probabilities. While the pipeline itself does not possess parallel sections, the generated tasks are well suited to be executed in parallel.

Task graph generation and parallel execution

As noted above, the data requests emitted by an actor into a pipeline are *asynchronous*. That is, Met.3D's main thread, responsible for user interaction, is not blocked while requested data is being processed. Instead, the tasks in the task graph generated in response to a request are executed in parallel *worker threads* (following the master/worker pattern; e.g. Mattson et al., 2004). As soon as the data item requested by the actor has been generated, a signal is sent to the actor (using the Qt 4 library's signal/slot mechanism; cf. Blanchette and Summerfield, 2008), which in turn can retrieve and visualize the item. Figures 5.14 and 5.15 illustrate how asynchronous data requests are handled by the Met.3D

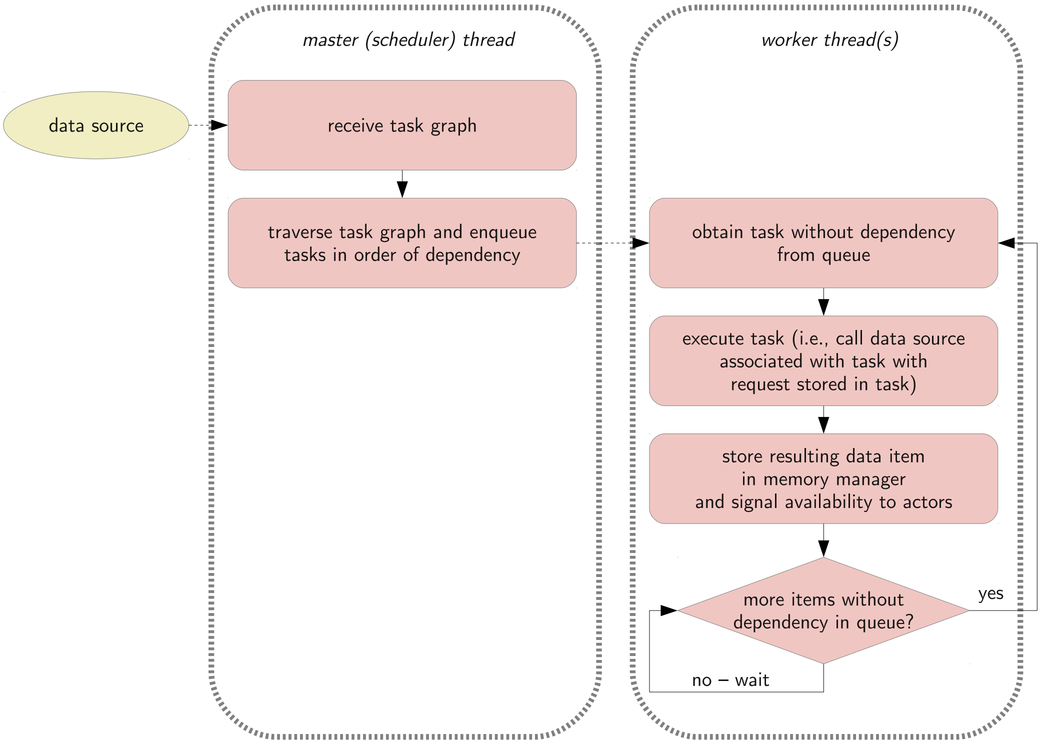

master (scheduler) thread
worker thread(s)
data source
receive task graph
traverse task graph and enqueue tasks in order of dependency
obtain task without dependency from queue
execute task (i.e., call data source associated with task with request stored in task)
store resulting data item in memory manager and signal availability to actors
more items without dependency in queue?
yes
no – wait

Figure 5.15: Master/worker structure of Met.3D's task parallel pipeline execution. Upon reception of a new task graph, the scheduler traverses the task graph in depth-first order and puts the visited tasks into a task queue. This way, tasks without dependencies are enqueued first. As long as tasks are contained in the queue, those without dependency are dequeued by one of the N worker threads. The task is executed, the resulting data item is stored in the memory manager, and the corresponding task graph connections are updated to remove the dependency. If applicable, a signal is emitted to connected actors.

architecture. For the following discussion, I assume that the requested data item has not been processed before and thus is not available in the memory manager cache.

Each data source provides methods to *request* (trigger the generation and execution of the task graph), *get* (obtain a reference to the data after the notification about its availability has been received) and *release* (inform the memory manager that the data is not required anymore) data items (cf. Table 5.2). To request a new data item, the actor in Fig. 5.14a calls the *request* method of data source B. The request is passed to the data source's *get task graph* method, which generates and returns the required task graph. If the data source is not dependent on inputs (for example, if the source reads data from disk), the task graph consists of a single task representing the data source's computation. If, as illustrated in the example, input dependencies exist, the task graph is a directed, acyclic graph (e.g. Sedgewick and Wayne, 2011). It is constructed recursively. As illustrated in Fig. 5.14a, the *get task graph* methods of all input sources are called. The returned task graphs are stored as dependencies for the calling task. In the example, the task graphs returned by data sources A1 and A2 are stored as a dependencies for the task generated by data source B. Note that if a data item required from an input data source is available in the memory manager cache, an empty task is returned. The data item is prevented from being deleted until it is used. Of course, data source B could, in the example, request multiple members, variables, or time steps from data sources A1 and A2.

The task graph thus constructed is passed to the scheduler for execution (Fig. 5.14b). I will describe the execution mechanism shortly; assume for now that is was successful and the generated data item has been stored in the memory manager. Once the actor receives the signal that the requested data item is available, it calls the *get data* method of data source B (Fig. 5.14c) to obtain a reference to the item. Here, the data source acts as a broker between the actor and the memory manager, which contains the data item. After the actor has finished using the data item (for instance, when a new data item is available), it releases the item (Fig. 5.14d). The release is required to remove the lock placed on the item by the memory manager. However, the released data item remains in the memory manager cache as a "passive" item for future use. It is deleted only if memory is required for new data items.

Figure 5.15 illustrates the task parallel execution of a task graph. The scheduler, upon reception of a new task graph (Fig. 5.14b), traverses the graph in depth-first order (e.g. Sedgewick and Wayne, 2011). Tasks are thus visited in order of dependency and can be enqueued in a serial *task queue* (those tasks without dependencies are enqueued first). Traversal is done in a separate thread in order to not impede the main application thread. The tasks placed in the task queue are handled by a number N of *worker threads*, where N depends on the number of CPU cores in the system. As long as tasks without dependency are contained in the queue, a random worker thread will remove the task from the queue and execute the associated data source with the request stored in the task. The resulting data item is stored in the memory manager, and the task is removed from the task graph (thus deleting a dependency for subsequent tasks). If appropriate, a signal is emitted to inform connected actors of the availability of the data item. The process is repeated as long as tasks are contained in the queue.

Care needs to be taken to avoid duplicate tasks. For example, multiple actors displaying the same forecast variable (for instance, a horizontal section actor and a volume actor) may request the same data item. Actors run in the application's main thread, and hence requests are sent sequentially to the pipeline. The first actor to request the item will trigger the creation of the corresponding task graph. Since, due to multi-threading, the other actors' requests will be in the pipeline before the execution of the first task graph has finished, duplicate creation and processing of tasks may occur if no action is taken. A similar situation may occur if a part of a new task graph is identical with a part of a graph that has already been enqueued. To avoid such duplicates, data sources in Met.3D query the scheduler to determine whether a task for the current request is already in queue. Also, the scheduler implements methods to discard duplicate sub-graphs and to combine graphs. Note that care also needs to be taken to ensure cache consistency. If a duplicate task is identified, the system needs to adjust its dependency connections in order to ensure that the corresponding data item is not deleted from cache before all requesting modules have used the item.

Another issue regards tasks whose computation includes the usage of system resources that are available only in limited number. This includes, for example, tasks that read data from disk and that use the GPU. Here, the scheduler supports configuration to allow only a limited number of tasks (for example, two for disk access and one for GPU access) to simultaneously access the resource.

Performance

To indicate an order of magnitude of the response times that Met.3D achieves when the displayed data field is changed, Table 5.3 lists approximate timings for changing the forecast time in the horizontal section in Fig. 5.2b (that is, four forecast variables are required per ensemble member). Additionally, the table lists the response time for applying a new selection criterion to the trajectories of all members.

Table 5.3: Order of magnitude of response times achieved by Met.3D. Timings are measured on hardware consisting of a consumer-class six-core Intel Xeon running at 2.67 GHz, equipped with 24 GB of RAM, a solid state drive and an Nvidia GeForce Titan graphics card. Task parallelism, if enabled, is facilitated by 12 worker threads. **(Upper rows)** Approximate time required by Met.3D to display a new image after the user has advanced the forecast time. Timings are given for the horizontal section in Fig. 5.2b, which uses four forecast variables (reading the ensemble from disk hence involves reading 4 × 51 × 1 MB at 1° grid spacing and 4 × 51 × 16 MB at 0.25° grid spacing). **(Bottom row)** Approximate time (excluding rendering time) required to change the trajectory selection criterion for computing the probability of WCB occurrence in Sect. 6.2. In this example, the trajectories of all members are available in cache, but have not been selected yet.

data	source	task parallelism	1°	0.25°
single member	disk (NetCDF)	no	60 ms	390 ms
		yes	47 ms	365 ms
ensemble mean	disk (NetCDF)	no	1.4 s	27 s
		yes	0.85 s	17 s
single member	member in cache	no	13 ms	25 ms
		yes	13 ms	25 ms
ensemble mean	mean in cache	no	13 ms	25 ms
		yes	13 ms	25 ms
change dp	trajectories in cache	no	151 ms	2.5 s
		yes	44 ms	1 s

This functionality is required in Chapter 6. Timings are provided for displaying a single member of the ensemble and for displaying the ensemble mean (the latter as an example of a statistic that requires all members of all variables when computed on-demand), both when data needs to be read from disk and when it is available in cache. Also, all timings are given for both pipeline task parallelism enabled (in the example using 12 worker threads) and disabled.

Depending on the nature of the request, enabled task parallelism leads to performance gains on the order of 10% to 40% for reading data from disk. As noted above, access to disk needs to be serialised, hence, only those parts of the individual tasks that do not directly perform a read call may be processed in parallel. For applying a new selection criterion to the trajectories of all members, response time is reduced by a factor of two to three. In this example, all data are available in memory and parallelism can be applied more effectively to the individual members.

The timings also show that disk access remains a bottleneck in the system. In particular when all ensemble members need to be read at high resolution, the available bandwidth of the hard drive limits the performance. However, if the CPU memory permits to cache the data items, response times are very low.

5.3.4 GPU based visualization algorithms

Met.3D's visualization algorithms support data fields on both hybrid sigma-pressure levels and on pressure levels. The difference is how the data fields are sampled on the GPU to obtain a value at a particular position in longitude-latitude-pressure space – an operation required by all visualization algorithms. In the horizontal, data fields on a regular longitude-latitude grid are supported.

To use the data on the GPU, a single time step of a single forecast variable of a single member (which

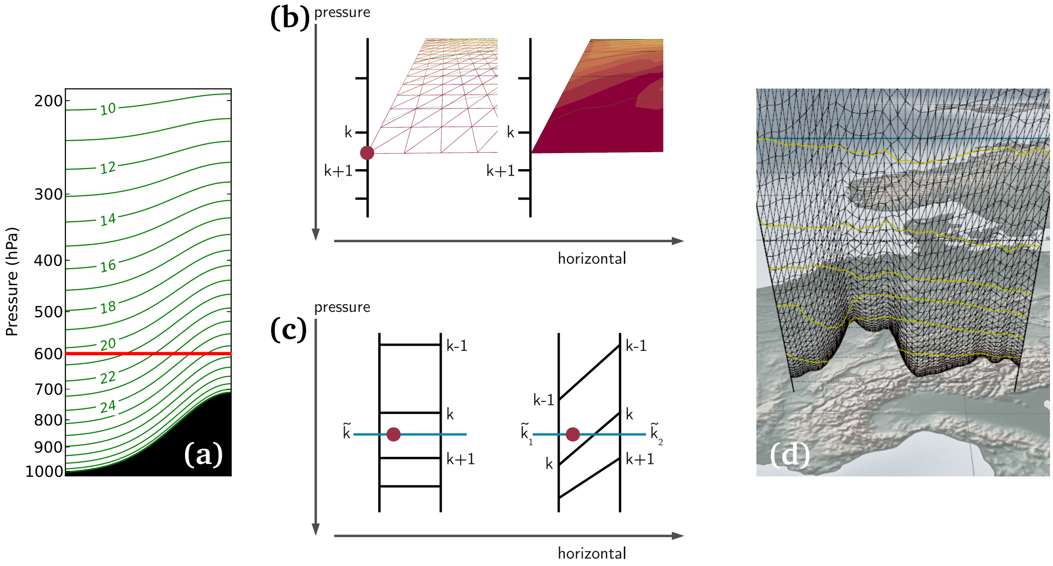


Figure 5.16: Sampling of data fields in GPU shaders. **(a)** Since the elevation of the model levels (green lines, the example shows levels from the 31 level model, level indices k in green) changes with surface pressure (black curve at the bottom; cf. Sect. 2.1.2), the data value for a given pressure value p can be located at different levels in the grid (the red line marks the location of $p = 600$ hPa). **(b)** For each vertex of a horizontal section, model levels k and $k+1$ are found by binary search. The scalar value is linearly interpolated in $ln(p)$ between these two levels. **(c)** PL grids are rectilinear (left), allowing the usage of trilinear hardware interpolation between the grid points surrounding a sample position (red dot). For ML grids (right), the sample position can be located between different model levels k for two adjacent grid columns, thus prohibiting hardware interpolation. **(d)** Example of how the surface orography affects the vertical displacement of the grid points in a vertical section.

I assume to fit into GPU memory, cf. Sect. 5.3.1) is stored in a 3D texture (cf. Sect. 3.2) in GPU memory. Longitude-latitude axes, as well as pressure levels for PL grids, are stored in an additional 1D texture. For ML grids (cf. Sect. 2.4), the corresponding 2D p_{sfc} field and the coefficients a_k and b_k are stored. This allows for computation of the pressure coordinate of a grid point on-the-fly, without the need to use additional graphics memory for a 3D texture with pressure values.

Shadows

To imitate shadows cast by visualized objects to the surface, I assume a directional light source from the top of a scene. This way, hard shadows (cf. Sect. 3.1.3) are cast that appear directly underneath the objects. The shadows thus reveal the horizontal positions of the objects on the basemap. At the time of writing, shadows in most actors are implemented by simple projection of the actor's geometry to the surface.

Horizontal sections

Horizontal 2D sections on a pressure surface p are rendered by placing the vertices of a grid of triangles horizontally at the positions of the data grid points and vertically at p (Fig. 5.16a). Data sampling only needs to be done when p is changed. Executed in parallel for each vertex, a binary search in the vertex shader yields the model levels (or pressure levels) k and $k+1$ enclosing p in the corresponding grid column. Following the ECMWF FULLPOS interpolation routines (Yessad, 2014), interpolation between these two levels is done linearly in $ln(p)$. The results are cached in a 2D texture. Filled contours are rendered by assigning colour to each fragment within a triangle in the fragment shader, using the horizontally hardware-interpolated scalar value. To obtain a colour, a transfer function actor provides a 1D texture that is used as an LUT, mapping a scalar value to a colour (see below). Line contours are generated by a marching squares (cf. Sect. 3.1) implementation in a geometry shader. Each grid cell of the cached 2D cross-section texture is examined in parallel and, if applicable, a line segment is drawn. Graticule, coast and border lines are overlain on each horizontal section to improve spatial perception (cf. Fig. 5.2b). Wind barbs are also generated in a geometry shader. It takes the horizontal wind field's u and v components as input and generates the geometry of the barbs, again exploiting GPU parallelism. To imitate a shadow generated by the section, a grey rectangle is drawn at surface level.

Vertical sections

Vertical sections are rendered with a similar grid of triangles. A triangle vertex is drawn for each vertical (model or pressure) level and each of a number of intermediate horizontal points along a line connecting the waypoints the user has specified (Fig. 5.16d). The distance between the intermediate points can be specified. A vertex shader computes the vertical position of each vertex and places it accordingly. This operation is a simple lookup for PL data and involves interpolation of p_{sfc} and computation of the model level pressure for ML grids. Scalar values are interpolated horizontally, also in the vertex shader, on the level on which the vertex is placed. They are also cached in a 2D texture that is updated if a waypoint is moved. Filled and line contours are generated equivalently to those in the horizontal sections.

3D isosurfaces

3D isosurfaces are rendered with front-to-back raycasting (cf. Sect. 3.1) implemented in the fragment shader. For each fragment (pixel) of the output image, a ray is cast through the data volume, sampling it at regular intervals and thus finding isosurface crossings. For this type of visualization algorithm, sampling the scalar volume is more expensive, as we need to interpolate in all three spatial dimensions to an arbitrary position in longitude-latitude-pressure space. For PL data, the grid is rectilinear (Fig. 5.16b) and can be sampled using texture mapping (cf. Sect. 3.2), thus benefiting from the fast trilinear hardware interpolation provided by modern GPUs. By mapping the longitude-latitude-pressure coordinates of the sampling position to texture coordinates (t_{lon}, t_{lat}, t_p) on the unit cube, the GPU interpolates the 3D texture at an arbitrary position. For regular grids, this mapping is a simple linear scaling. Since, however, PL grids retrieved from MARS are irregularly spaced in the vertical, we need a method to map pressure to t_p. This is realised by means of an LUT stored in an additional 1D texture. The level indices k can be linearly scaled to $t_{p,k} \in (0..1)$. Since we know the pressure values p_k at the levels k, we can compute a continuous $\tilde{k}$ for intermediate p by linearly interpolating in $ln(p)$ (Fig. 5.16b). $\tilde{k}$ can subsequently be scaled to t_p. These mappings from p to t_p are precomputed for a number, say 2048, of pressure values and stored in the LUT that can be accessed in the shader.

ML grids are not rectilinear and thus sampling becomes more complicated. As illustrated in Fig. 5.16b, the continuous level index $\tilde{k}$ in general is not the same for adjacent grid columns. In the worst case, a given p is located between different model levels in its four surrounding grid columns. Trilinear hardware interpolation requires $\tilde{k}$ to be the same in all surrounding grid columns, it hence cannot be used. Consequently, we need to split the interpolation into four vertical interpolations in the grid columns and a subsequent bilinear horizontal interpolation. A naïve approach is to use the binary search used for the horizontal sections for the vertical interpolations, however, our experiments showed that rendering times can be reduced by a factor of about two when again making use of an LUT approach for hardware interpolation. However, the horizontal interpolation needs to be implemented in software. ML sampling is hence over four times more expensive than PL sampling.

To use hardware interpolation for ML in the vertical, we need to extend the LUT approach. First, the horizontal texture coordinates t_{lon} and t_{lat} are set to the horizontal position of the grid columns. Since the model level pressure varies with p_{sfc}, we in principle need to precompute one LUT for every p_{sfc} value that occurs in the forecast field. We instead make use of a 2D LUT, containing LUTs for discrete values of p_{sfc} reflecting the expected range of p_{sfc} in the data. Using bilinear hardware interpolation, this LUT is used to interpolate in both p_{sfc} and $ln(p)$ to obtain a mapping from $ln(p)$ to t_p. The additional memory requirement is reasonable: For an LUT using 2048 entries in the vertical and 600 entries for p_{sfc} between 1050 hPa and 450 hPa, approximately 9 MB of GPU memory are required in float precision (i.e. 4 bytes/value). The table can be shared among variables on the same grid.

The traversal of the data volume is accelerated with an empty-space skipping strategy (Krüger and Westermann, 2003). The longitude-latitude-pressure space covered by a data field is divided uniformly into a regular grid of $N_i \times N_j \times N_k$ cells. For each cell, minimum and maximum data values are computed. In the shader, the information is used to skip cells in which an isosurface cannot possibly be located. Due to the different horizontal and vertical scales, care has to be taken when choosing the step size for traversing non-empty cells. Depending on the factor that is used to scale $ln(p)$ to a z-coordinate in visualization space, the vertical distance between two grid points often is considerably smaller than the horizontal distance. The step size needs to be chosen small enough to ensure that no grid point is skipped during traversal.

Once an isosurface crossing has been identified, the isosurface normal (equivalent to the gradient of the scalar field at the crossing position) is computed via central differences. The pixel colour is subsequently determined using the commonly used Blinn-Phong lighting model (cf. Sect. 3.1; e.g. Engel et al., 2006). Colour can be predefined or obtained from a transfer function. Also, a second scalar field can be mapped to the isosurface to colour, for example, a wind speed isosurface by temperature.

Shadows are realised by means of a shadow map approach (cf. Sect. 3.1). The isosurfaces are rendered once from the top to a binary offscreen texture. This texture is subsequently drawn on top of the basemap. It needs to be updated every time a parameter affecting the isosurface is changed.

Trajectories

Trajectories are rendered as 3D tubes, with tube geometry being generated from a trajectory line by a geometry shader. This way, only the trajectories' line segments need to be stored in GPU memory. The implementation follows the approach by Bürger (2010). For each line segment connecting two particle positions (vertices), the geometry shader creates a triangle mesh that approximates a tube. For each vertex, a normal is required. This normal is rotated around the line segment to create the tube vertices. Care must be taken to avoid holes in the geometry. Except for the first and last vertex of a trajectory, the normal between two segments needs to be perpendicular not to one of the segments but to the tangent at the connection. Also, normals of a tube need to be consistent with each other to prevent the tube

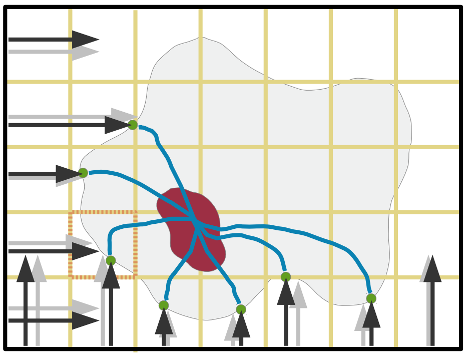


Figure 5.17: Computation of normal curves. Seeding points for the curves (green dots) are placed at the intersections between axis aligned rays (black arrows) and the outer isosurface (only rays from two directions are shown for illustration). Only a single seed is allowed in each grid box of the yellow volume.

geometry from becoming "twisted". Unfortunately, this requires the normals to be computed iteratively (cf. Bürger, 2010, and references therein), which cannot be accomplished on-the-fly by the parallelised geometry shader – it only has access to three line segments at most.

Analogous to my assumption of single forecast data fields being the smallest data entity for grids, I assume the set of all trajectories computed for a single forecast member and starting at a single time step to be the smallest data entity for trajectories. This set of trajectories can be stored in a vertex buffer in GPU memory. To render a subset of trajectories (for example, selected as the result of applying a selection criterion), indices of selected trajectories are stored in an index buffer. This approach has the advantage of only requiring an update of the index buffer when the selection criterion is changed. The trajectory actor hence requires three input data sources. The first data source provides the geometry of all trajectories available for a given member and (start) time step, the second provides a selection of a subset of the trajectories, and the third provides precomputed normals.

Spheres showing particle positions at a given time step are rendered as *billboards* (screen-aligned, textured quads; e.g. Cozzi and Ring, 2011). Finally, similar to horizontal sections, shadows are rendered by projecting the tube or sphere geometry to the surface.

Normal curves

Normal curve computation is implemented in a compute shader. Fig. 5.17 illustrates the proposed normal curve algorithm. To generate a set of seed points, rays aligned with the three world space axes (longitude, latitude, pressure) are cast through the data volume. The rays are started at regularly spaced points (grey arrows; the spacing can be adjusted by the user). To avoid the regular pattern of these initial start points being reflected by the normal curves, we disturb the ray positions by a random factor (black arrows). The intersection points of the rays with the selected outer isosurface are then used as initial seed points for the normal curves (green dots). In particular in regions of high curvature, multiple rays can hit the isosurface at close-by points on the surface. To prevent normal curves from being started close together, a regular volume with a grid size of the average initial ray distance is placed over the scene (yellow grid). Only one seed is allowed per grid cell. Hence, if a seed point falls into a cell already occupied, it is discarded (illustrated in the orange grid cell). The normal curves are integrated in parallel in the direction of the scalar field's gradient, using a 4th-order Runge-Kutta scheme. The gradient is computed with the same method used for isosurface shading. If present, the integration can be stopped at an inner opaque isosurface (illustrated by the red isosurface in Fig. 5.17). The computed curves are visualized with the same techniques as trajectories.

Surface topography

Surface topography is rendered by interpreting the 2D surface pressure field as a *height map* and applying vertex-shader-based displacement mapping (e.g. Cozzi and Ring, 2011). Similar to horizontal sections, vertices are placed at the grid points of the 2D surface field to yield a triangle mesh covering the grid. The surface pressure field (or any other 2D variable containing pressure data, for example, mean sea level pressure) is uploaded to the GPU as a 2D texture. In the vertex shader, the vertical pressure coordinate of each vertex is obtained from the texture, and the vertex is vertically displaced. Similar to the isosurfaces, pixel colour is determined by using the Blinn-Phong model. The terrain can also be coloured by applying a transfer function to a second 2D field.

Transfer functions

Following Zeileis et al. (2009) and Stauffer et al. (2015), I have implemented the HCL colour space due to its perceptual linearity (cf. Sect. 3.1.3). The implementation reuses methods from the *R colorspace* package[5]. Following R. Stauffers “Online HCL creator”[6], the transfer function actor provides properties to specify hue, chroma and luminance values to create qualitative, divergent, and sequential colourmaps. A generated colourmap is stored in a 1D texture in GPU memory, where it is used as a lookup table by other actors. In addition, colourmaps can be defined by specifying a range of RGBA values and corresponding scalar values.

Performance

Table 5.4 lists typical rendering times for selected visualizations shown in this thesis. Measurements have been taken on (at the time of writing) one high-end (GeForce Titan) and one mid-range (GeForce GTX 560Ti) consumer class graphics card (cf. Table 3.1). The figures show that rendering of the 2D horizontal and vertical cross-sections is very fast, even if the sections are moved by the user. Rendering of isosurfaces is more demanding. However, note that the performance of the raycaster strongly depends on the visualized data and on camera viewpoint. In particular the effectiveness of the empty-space skipping strategy for a selected isovalue depends on the spatial distribution of the data values.

During user interaction, the step size used by the raycaster to sample the data fields can be reduced. While this temporarily reduces image quality, rendering time is also reduced. All images shown in this thesis have been produced with a step size of 0.1. In Videos 1 and 2, the step size has been configured to be reduced to 1 during camera movements. The timings in Table 5.4 show that with a step size of 1, Met.3D is able to achieve rendering times on the order of 100 ms or less for all listed examples and with both GPUs. The Titan GPU typically is about 30% to 50% faster than the GTX 560Ti, achieving rendering times on the order of 100 ms even for a step size of 0.1 in most examples. From my experiments, pressure level data are rendered by a factor of two to three times faster than comparable hybrid model level data. For example, in Chapter 6, I compute probability fields on the “standard” ENS hybrid model levels and

[5] Available at `http://cran.r-project.org/web/packages/colorspace/index.html` (last access April 2015).

[6] Available at `http://hclwizard.org` (last access April 2015).

Table 5.4: Order of magnitude of rendering times achieved by Met.3D. Timings are measured on hardware consisting of a consumer-class six-core Intel Xeon running at 2.67 GHz, equipped with 24 GB of RAM, and the two listed graphics cards (cf. Table 3.1). Shown is the time required to render selected visualizations from this thesis. For all measurements, ECMWF ENS data at a grid point spacing of 1° in both longitude and latitude are used and the data fields are available in GPU memory. Timings are average values of continuous rendering over 30 seconds. A Met.3D window of 1600 by 900 pixels is used (the size used for Videos 1 and 2, corresponding to a viewport of 1192 by 864 pixels). "Animated" for cross-sections refers to vertically sliding a horizontal section or moving a waypoint of a vertical section.

figure	setting	GeForce Titan	GeForce GTX 560 Ti
Fig. 5.2b	static	1.7 ms	2.3 ms
Fig. 5.2b	animated	2.6 ms	2.8 ms
Fig. 5.3a	static	2.3 ms	6.2 ms
Fig. 5.3a	animated	2.4 ms	6.8 ms
Fig. 5.5a	step size 0.1	211 ms	417 ms
Fig. 5.5a	step size 1	64 ms	107 ms
Fig. 5.6a	step size 0.1	119 ms	222 ms
Fig. 5.6a	step size 1	43 ms	62 ms
Fig. 5.6c	step size 0.1	133 ms	248 ms
Fig. 5.6c	step size 1	48 ms	72 ms
Fig. 5.1c	step size 0.1	105 ms	273 ms
Fig. 5.1c	step size 1	46 ms	83 ms
Fig. 6.5 S2a	model levels (step size 0.1)	178 ms	361 ms
Fig. 6.5 S3a	pressure levels (step size 0.1)	55 ms	102 ms

on similar pressure levels (that is, the same number of levels, similarly stretched in the vertical). The pressure level grids are rendered about two to three times faster.

5.3.5 Discussion

The proposed architecture facilitates the implementation of all methods required to achieve the thesis goals. The performance measurements in Tables 5.3 and 5.4 show that the combination of global memory management, task parallelism in the pipeline, and GPU-based visualization algorithms in many cases yields response times that facilitate a natural, continuous interaction with the scene. The two major limiting factors are disk access and the requirement for using software-based interpolation in the ML raycaster.

Met.3D demonstrates that current consumer hardware has become powerful enough to enable continuous-interactive visualization of combined 2D/3D forecast products, and is also able to handle ensemble forecast data for practical forecast domains. However, despite these encouraging results, I consider further improvements of the current implementation desirable with respect to both performance and functionality. In the following, I discuss some of the presented methods and offer suggestions for further investigations and improvements.

Data processing pipeline

With respect to the data processing pipeline, I recommend to further investigate the following three issues: First, the question of which techniques can be employed to further accelerate disk access. Second, the question of how parallelism can be better exploited to improve processing performance in the pipeline. Third, the question of how the architecture can be adapted to accommodate datasets in which the data field of a single variable, single member and single time step exceeds the available GPU memory.

- Due to the data sizes used in this thesis, the global memory management implemented in Met.3D is able to cache intermediate results, thereby facilitating very fast response times. However, the timings in Table 5.3 show that disk access is a bottleneck in the system. Assuming the availability of sufficient system memory, the issue can be approached by *pre-loading* and *pre-execution* techniques. Support for such techniques is listed in Table 5.1 as an additional requirement. While outside the scope of the present thesis work, integration into the Met.3D architecture will due to task parallelism and memory management be straightforward. Techniques that estimate upcoming requests to the pipeline can hide disk access by pre-loading data while the user is exploring the current visualization. Consider, for example, a user that animates a scene over time. While the user spends time viewing a particular time step, the system can pre-load the data of upcoming time steps in the background. For the implementation, an additional pipeline module will be required that monitors requests sent to the pipeline. Based, for instance, on the last monitored requests (other strategies may be possible as well), upcoming requests can be estimated and sent to the pipeline. Pre-load requests can be given a lower priority, so that the scheduler can distinguish between user requests that should be processed immediately and pre-load requests that can be processed as soon as resources are available.

- The effectiveness of the implemented task parallelism strongly depends on the type of request sent to the pipeline. It is most efficient when many tasks can independently operate on data in system memory. A good example is a change of the selection criterion for a visualization of probabilities of gridded trajectories listed in Table 5.3 – the trajectories of all ensemble members are cached in memory, and the members can be processed in parallel. However, if the independent tasks need to read data from disk, parallelism is limited, as the code section accessing the hard drive needs to be serialised. I have not conducted systematic measurements of system workload, however, I suggest to do so in order to conduct further studies to improve parallelism in the pipeline. Notably, the use of *data parallelism* in single pipeline modules might further improve performance. Also, studies on alternative scheduling algorithms (e.g. Kwok and Ahmad, 1999) and on how to best schedule usage of the GPU (which, as the hard drive, is a hardware resource to which access needs to be serialised) in the pipeline may help to improve the system.

- To apply Met.3D to datasets of which not all ensemble members simultaneously fit into system memory, the implementation of algorithms with progressive data access (cf. Table 5.1) is required. The Met.3D architecture could accommodate such algorithms on the global level of the task scheduler and memory manager. The algorithm requiring progressive access, say the computation of the ensemble mean, would be executed until it tries to obtain data from an input source. If the data is not available yet, the corresponding call to the memory manager would suspend the ensemble-mean thread until the data item has been read to memory. Once available, the ensemble-mean thread is woken up. It processes and releases the data item. When the thread tries to obtain the next required data item, it is sent to sleep again and the process is repeated.

If a dataset is so large that a single time step of a single member of a single variable does not fit into memory, suitable approaches include streaming (as implemented by VTK) or level of detail (the usage of lower-resolution versions of the dataset that can be progressively refined as the region of interest is made smaller, as implemented by Vapor). Level of detail would be feasible with Met.3D by, for instance, integrating a data reader module able to read the Vapor file format. A streaming approach that subdivides the dataset into smaller blocks would in particular be challenging for the visualization algorithms. The pipeline implementation is able to load subregions of a dataset, challenges would include efficient pre-loading.

Visualization algorithms

Concerning the visualization algorithms, I suggest to continue research and developments with respect to the following issues:

- I note that as for the data pipeline, comprehensive performance optimisations of the visualization algorithms were outside the scope of this thesis. In particular with respect to the raycaster, further optimisations are possible (and desirable), for example, by integrating an adaptive step size strategy that only samples the data volume at grid cell boundaries. However, due to the complex geometry of the ML grid cells, computing the boundary crossings may create overhead. It needs to be investigated whether such an approach leads to further improvements.

- While the GPU-based implementation of the marching squares contouring algorithm is very fast, a distinct shortcoming is the lack of labels displaying the isovalues along the contour lines. Here, suitable methods need to be developed that display labels without compromising the efficiency of GPU-based contour line generation during user interaction.

- The shadows implemented in Met.3D, as well as the movable vertical axes, provide valuable depth cues. I suggest to investigate further improvements to spatial perception. At the time of writing, shadows are computed separately for each actor and are cast to the surface of the scene. I consider it desirable to have shadows being cast globally, including all displayed objects. Such functionality would be possible, for example, by implementing a global shadow map (e.g. Akenine-Möller et al., 2008; Wenger, 2013). Also, further global illumination techniques may yield improved spatial perception (e.g. Jönsson et al., 2014).

- Met.3D is well suited to implement and evaluate further visualization techniques. I suggest to implement some of the techniques reviewed in Sect. 4.3 (for instance, the techniques proposed by Pilar and Ware (2013, cf. Fig. 4.10), Sanyal et al. (2010, cf. Fig. 4.11), and Whitaker et al. (2013, cf. Fig. 4.11)), and to evaluate their suitability for interactive forecast exploration in a 3D context.

- The support of further types of NWP model grids may require some effort. Support for hybrid sigma-pressure coordinates is integrated in the GPU shader programs that implement the visualization algorithms. As described in Sect. 2.1.3, a number of NWP models use vertical coordinates other than hybrid sigma-pressure coordinates. To support further vertical coordinates, the corresponding shader codes will need to be modified. Support for horizontal grid layouts other than regular in longitude and latitude will also require widespread modifications. Here, it will also be challenging to maintain interactive rendering performance.

5.4 Impact of (not) regridding on ensemble statistical quantities

Statistical quantities in Met.3D are computed per grid point. Probabilities, for example, are computed by evaluating for every member and for each grid point a given probability criterion (for instance, wind speed exceeding a given threshold). The evaluation of the criterion yields for every member a binary volume, with the bits set when the criterion is fulfilled. Probabilities are computed by counting the number of members with a set bit for each grid point. Other statistical measures are computed similarly for each grid point over the ensemble dimension.

For 2D grids, this is common procedure (Wilks, 2011) and also for 3D grids not an issue as long as a given grid point is located at the same spatial position in all members. However, due to surface pressure varying between ensemble members, this is not the case for data on ML grids. Hence, depending on the vertical gradient of the forecast variable from which a statistical quantity is computed, an error is introduced. One approach to this issue is to vertically regrid all ensemble members to a common grid, for example, the one defined by the mean surface pressure (as done in the example pipeline in Fig. 5.13). This, however, introduces an additional interpolation step and demands computational resources.

In this section, I investigate the visual and quantitative differences between statistical quantities computed from the original ML grids and those computed from data fields regridded to a common grid. The differences are compared to an additional error that is introduced by linearly interpolating the statistical quantities. At ECMWF, maps of statistical quantities on pressure levels are computed from the individual member's forecast data on these pressure levels. This implies that a forecast meteorological variable is first interpolated to the target vertical position for each member (using linear interpolation in p or $ln(p)$, cf. Yessad, 2014), followed by the computation of the statistical quantity. If, on the contrary, we first compute the statistical quantity on the 3D model grid and then linearly interpolate to the target vertical position, an error is introduced due to the non-linear nature of most statistical measures. The same problem arises in the horizontal dimensions.

In the following, I analyse regridding and interpolation error for the forecast data I had available from TNF. I present results from the forecast initialised at 00:00 UTC 15 October 2012 and valid at 114 hours lead time at 18:00 UTC 19 October 2012. This case is representative for the dataset, results for other time steps of the TNF dataset are similar.

5.4.1 Variation in grid point pressure

First, I estimate typical vertical grid point displacements that can be observed between ensemble members. Figure 5.18a shows the standard deviation of p_{sfc} for the example case. It reaches values of 8 to 10 hPa in the uncertain regions of the forecast. This particularly applies to the low pressure systems over the Atlantic and the northern British Isles. Fig. 5.18b shows a vertical cross-section of the maximum pressure difference between any two members per grid point in these two areas. Close to the surface, the difference reaches 40 hPa, corresponding (at low altitudes) to an elevation offset of about

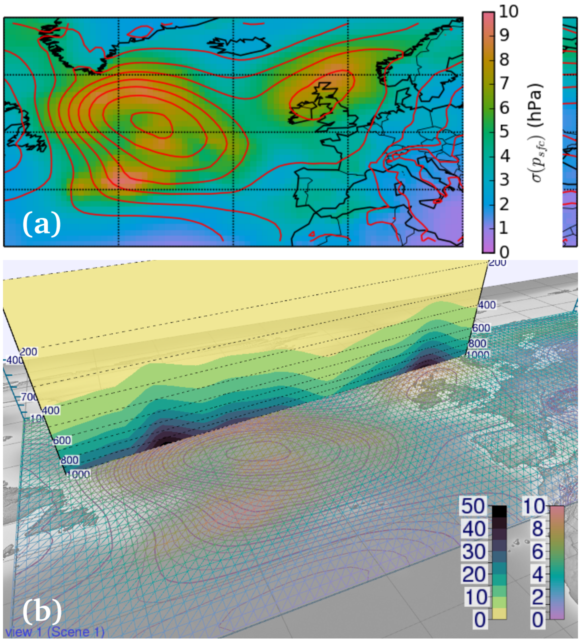


Figure 5.18: **(a)** Standard deviation of surface pressure, $\sigma(p_{sfc})$. Forecast from 00:00 UTC 15 October 2012, valid at 18:00 UTC 19 October 2012. Red contour lines show mean sea level pressure. **(b)** Vertical section of the pressure difference (yellow-blue-black colour bar in hPa) between highest and lowest ensemble member, rendered on top of a wireframe map of $\sigma(p_{sfc})$.

400 m. In most other regions, however, differences are smaller. Also, as expected from the model grid geometry, differences vanish in upper atmospheric levels.

5.4.2 Difference due to vertical regridding

Vertical regridding is implemented as a data source that can be integrated into the Met.3D ensemble processing pipeline (cf. Fig. 5.13). The user can toggle between visualizations from original and from regridded data fields, and, if required, permanently enable regridding. If statistical quantities are computed from the original member grids, the resulting field is interpreted on a grid defined by the mean surface pressure.

On my test hardware (cf. Sect. 5), the cost of single-threaded CPU regridding on average is about 60 ms per member and variable for the TNF ENS forecast at 1° grid spacing (256,742 grid points per 3D field) and about 1 s at 0.25° grid spacing (4,997,262 grid points). Even though multiple ensemble members can be processed in parallel on a multicore machine and the regridding process could be further sped up using the GPU, there is a delay in particular for high-resolution datasets and visualizations using multiple variables.

I have visually inspected a number of 2D and 3D renderings of statistical quantities of several meteorological variables. As expected, the largest visual differences appear close to the surface. They become most manifest in horizontal sections, which are most sensitive to vertical variations in a 3D data field. Fig. 5.19 shows two typical low-altitude examples, the probability of horizontal wind speed exceeding 20 ms^{-1}, $p(|\vec{v}| > 20$ ms$^{-1})$, and the standard deviation of relative humidity, σ(RH). From my inspection I find that differences tend to be larger for variables that depend on moisture and variables derived

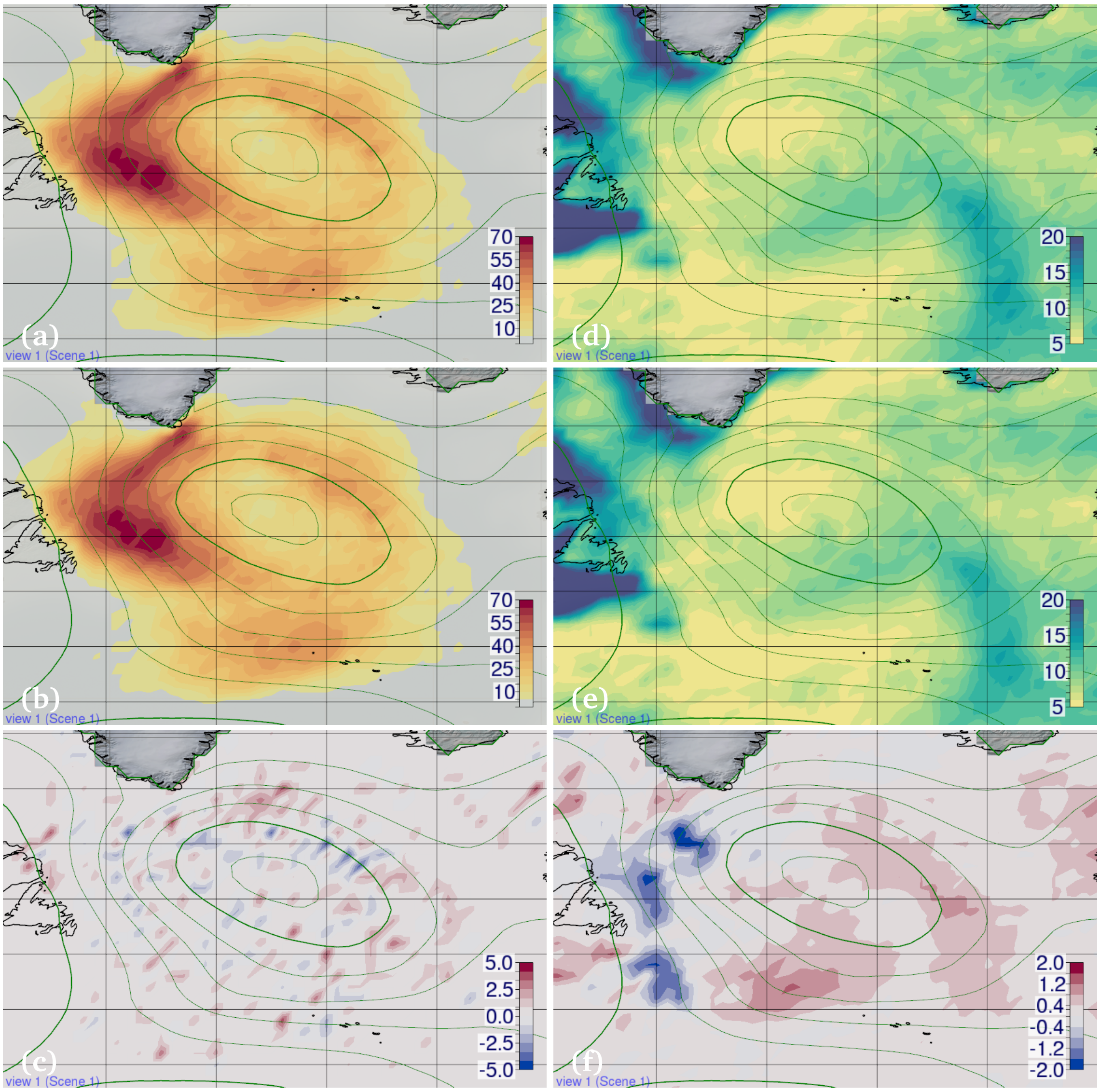


Figure 5.19: Visual differences between statistical quantities computed from a vertically regridded ensemble to those computed from the original ensemble. Horizontal section at 950 hPa (approx. model levels 51-55 in Figs. 5.20 and 5.21) of **(a-c)** $p(|\vec{v}| > 20 \text{ ms}^{-1})$ (%) and **(d-f)** $\sigma(\text{RH})$. Same forecast as in Fig. 5.18. Shown is **(a)** the probability and **(d)** standard deviation computed from the original model grid, **(b)** and **(e)** computed from members regridded to the grid defined by the mean p_{sfc}, and **(c)** and **(f)** the difference between both fields.

thereof, however, I could not find any examples in which visualized structures were significantly altered. For example, while there is some visible difference in $\sigma(\text{RH})$ along Rafael's warm front, the structure itself is not significantly altered.

Visual differences strongly depend on the employed colour palette and visualized data range. Depending on the range of values covered by a single colour, small changes might simply not be reflected in the visualization. To ensure that differences in general are small, I have performed a statistical analysis of the entire TNF dataset. Figure 5.20 shows results for three statistical quantities computed from the wind field of the example case: mean $\mu(|\vec{v}|)$, standard deviation $\sigma(|\vec{v}|)$, and $p(|\vec{v}| > 20 \text{ ms}^{-1})$. The scatter plots show that for all three quantities the largest differences appear at lower altitudes (higher

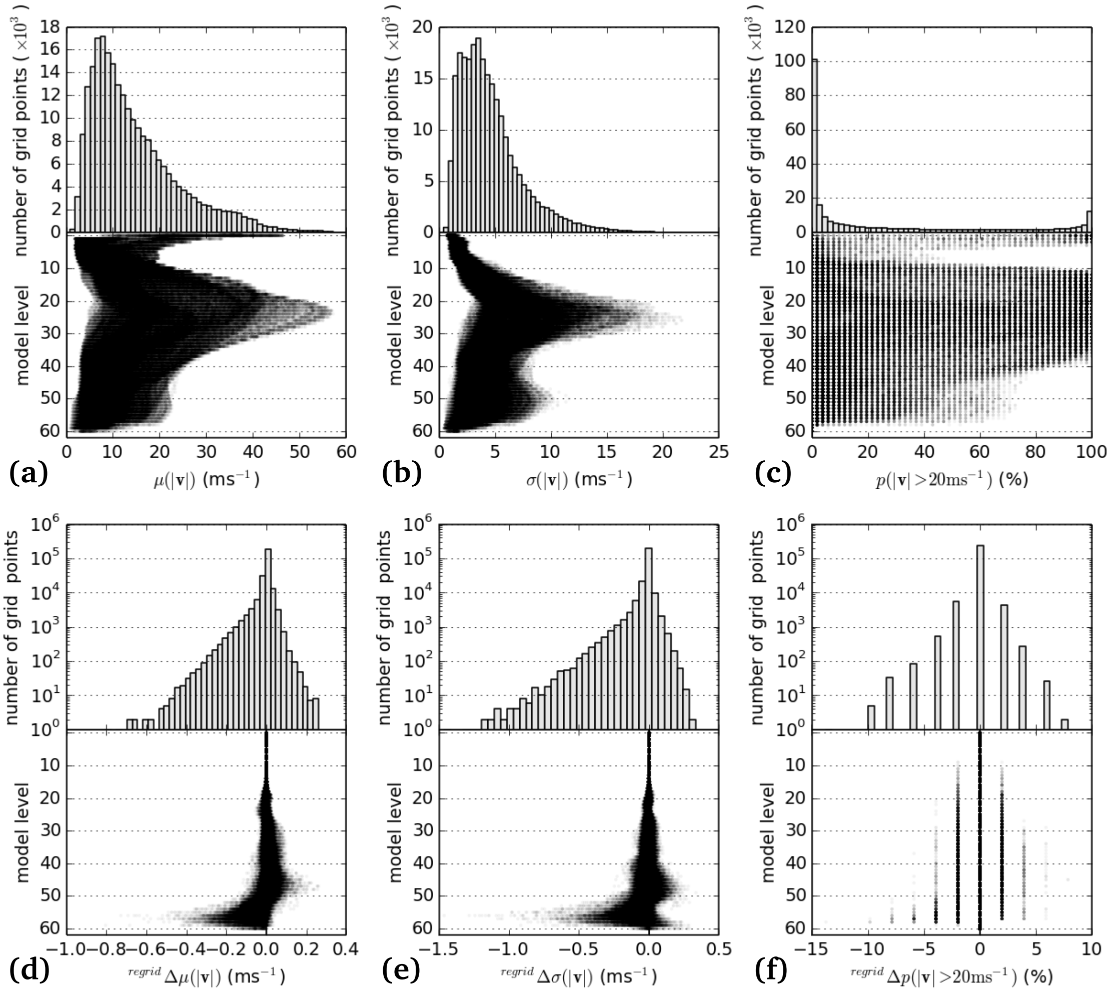


Figure 5.20: Distribution of differences between statistical quantities computed from a vertically regridded ensemble to those computed from the original ensemble. Plots are generated from all 256,742 grid points of the data field. Same forecast as in Fig. 5.18. Shown are **(a)** and **(d)** $\mu(|\vec{v}|)$, **(b)** and **(e)** $\sigma(|\vec{v}|)$, and **(c)** and **(f)** $p(|\vec{v}| > 20\ \mathrm{ms}^{-1})$. **(a-c)** Distribution and vertical occurrence of absolute values of the quantities. **(d-f)** Distribution and vertical occurrence of differences due to regridding (denoted by ${}^{regrid}\Delta$). Note the logarithmic scale of the histograms in (d-f). Probability values are discrete due to the size of the ensemble (51 members).

model level indices). Also, differences mostly are small compared to absolute values of the quantities. For example, at only few grid points the difference in $\sigma(|\vec{v}|)$ and $p(|\vec{v}| > 20\ \mathrm{ms}^{-1})$ exceeds 1 ms^{-1} and 10%, respectively. The range of differences observed in Fig. 5.19 is well reflected in the histogram.

Larger differences appear for statistical quantities computed from moist variables (Fig. 5.21). Again, the histogram for $\sigma(\mathrm{RH})$ confirms the range of differences shown in Fig. 5.19 (Fig. 5.21d). For probabilities of potential vorticity and cloud cover, differences of up to 30% can occur (Fig. 5.21e,f). However, for most grid points, differences are smaller.

Figure 5.22 shows a histogram of $\sigma(p_{sfc})$ of the example case, overlain with the bin-averaged difference in $\sigma(|\vec{v}|)$. As can be expected, larger differences on average occur in regions with high $\sigma(p_{sfc})$. However, even for large $\sigma(p_{sfc})$, most differences are small (not shown). We hence cannot state that large $\sigma(p_{sfc})$ in general accounts for large differences.

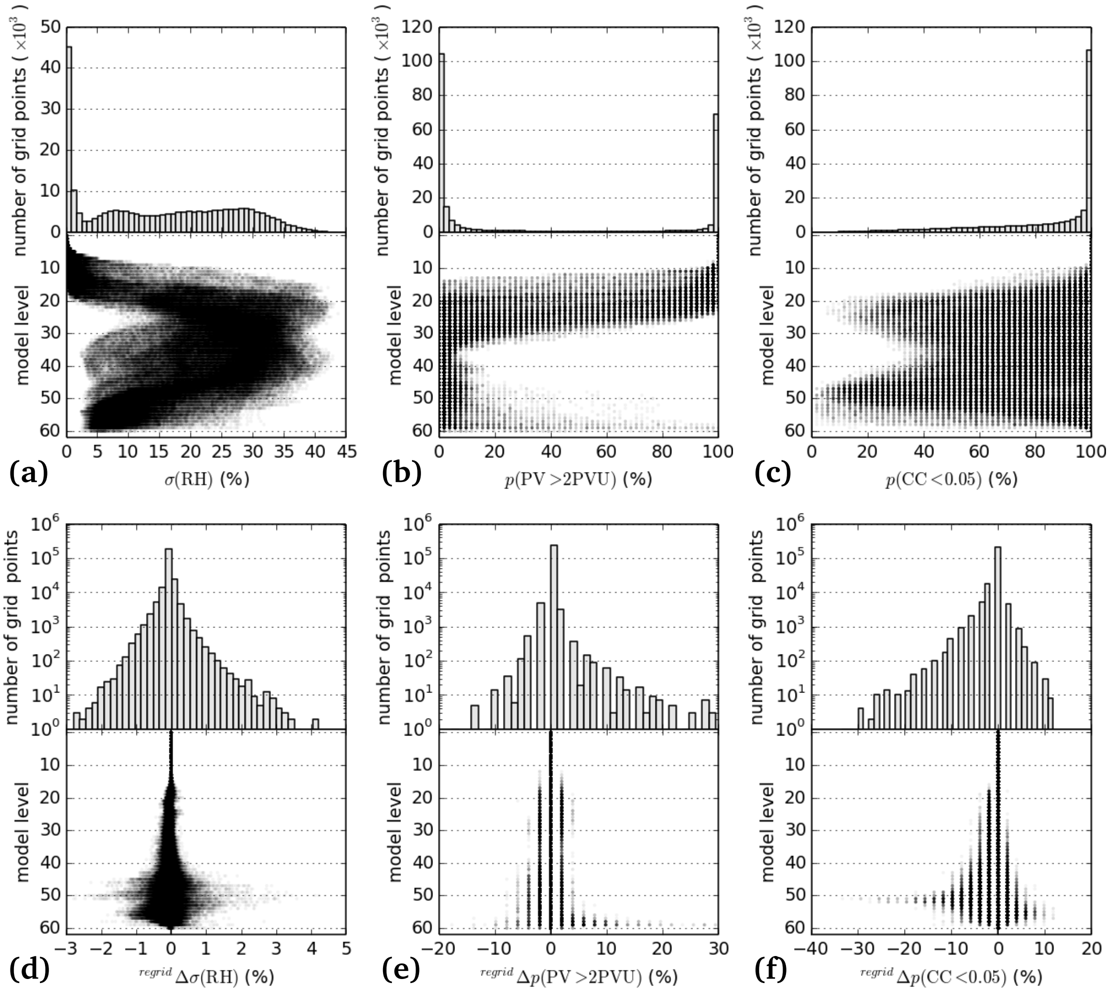


Figure 5.21: The same as Fig. 5.20 but for variables depending on moisture. **(a)** and **(d)** Standard deviation of relative humidity. **(b)** and **(e)** Probability of potential vorticity exceeding 2 PVU. **(c)** and **(f)** Probability of grid box cloud cover fraction falling below 0.05.

5.4.3 Error due to vertical interpolation of statistical quantities

The error introduced by vertical linear interpolation of a statistical quantity depends on the quantity. Consider the example given in Table 5.5. Due to the linear nature of the ensemble mean, there is no difference whether we first compute the mean at the grid points and then interpolate to the sample location or vice versa. For non-linear quantities including standard deviation and probability, the results are different.

Figure 5.23 shows distributions of the interpolation errors for $\sigma(|\vec{v}|)$ and $p(|\vec{v}| > 20\ \mathrm{ms}^{-1})$. Note that in contrast to the differences caused by regridding, the largest errors due to interpolation occur in upper atmospheric levels, where the vertical distance between model levels becomes larger. Between the surface and approximately model level 10 (approximately 100 hPa), the order of magnitude of the interpolation errors is comparable to that of the differences due to regridding. At middle atmospheric levels, both errors are at a minimum, as shown by the vertical profile of horizontally averaged differences. At the upper boundary of the model atmosphere, interpolation errors become significantly larger.

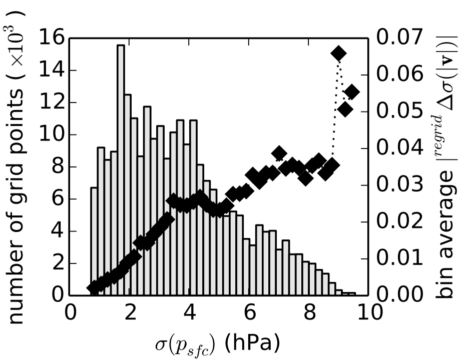


Figure 5.22: Histogram of $\sigma(p_{sfc})$, overlain with the bin-average difference of $\sigma(p_{sfc})$ against the differences between $\sigma(|\vec{v}|)$ computed from a vertically regridded ensemble and computed from the original member grids. Same forecast as in Fig. 5.18.

Table 5.5: Example of vertically interpolating statistical quantities. Consider an ensemble of three members and corresponding scalar quantities s_1 .. s_3 at the two vertical levels k and $k+1$. While the mean value $\mu(s)$, interpolated to the midlevel between k and $k+1$, equals the mean of the interpolated scalar values, this is not true for the standard deviation $\sigma(s)$ and the probability that a scalar value exceeds 1.5, $p(s > 1.5)$. The subscript i refers to "interpolated".

level	s_1	s_2	s_3	$\mu(s_i)$	$\mu_i(s)$	$\sigma(s_i)$	$\sigma_i(s)$	$p(s_i > 1.5)$	$p_i(s > 1.5)$
k	0.8	1.7	1.8	1.433		0.45		0.66	
midlevel	1.4	1.45	1.4	1.4166	1.4166	0.24	0.44	0	0.5
k+1	2.0	1.2	1.0	1.40		0.43		0.33	

These regions, however, are not relevant for the forecast cases that are of interest for this thesis.

5.4.4 Discussion

The examples show that the errors introduced by computing the statistical quantities from the original member grids are of comparable magnitude to the errors introduced by vertically interpolating the computed quantities. For most grid points, both are negligible and result in only little difference in the visualization. However, for some variables and cases (in particular moist variables), differences can be of the same order of magnitude as the statistical quantity itself.

I conclude that for general exploration of the forecast data it is sufficient for the user to use the "fast" option and visualize quantities computed from the original member grids. However, if the result is crucial for an important decision, our advice is to switch to regridded quantities and accept the additional compute time. The "best" results and those most comparable to products obtained from ECMWF can be achieved by first interpolating each member to the desired vertical pressure and then computing the statistical quantities. In this case, neither regridding nor vertical interpolation of the quantity corrupts the result. In Met.3D, this is possible for horizontal sections.

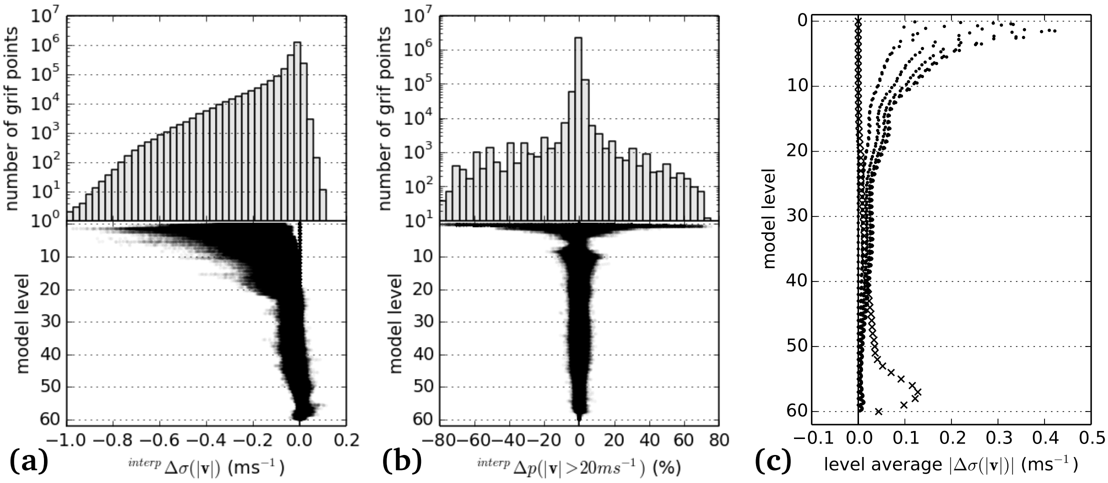


Figure 5.23: Distribution of errors due to vertical linear interpolation (denoted by $^{interp}\Delta$) of statistical quantities. **(a)** Distribution of errors of $\sigma(|\vec{v}|)$ (top), and vertical occurrence of the errors (bottom). **(b)** The same for $p(|\vec{v}| > 20\ \text{ms}^{-1})$. **(c)** Vertical profile of level average differences due to regridding (crosses) and interpolation (dots). Same forecast as in Fig. 5.18.

CHAPTER 6

Forecasting warm conveyor belt occurrence

Met.3D, as described in Chapter 5, provides the methods required to interactively explore an ensemble forecast in order to investigate forecast questions FQ-A and FQ-B (cf. Sect. 1.3). A meteorologist can explore the large scale weather situation by utilising the combination of familiar 2D products and 3D visualizations, and judge the uncertainty of the predictions by animating the ensemble members and interactively visualizing derived statistical quantities including ensemble mean, standard deviation and probabilities. The purpose of this chapter is to add the methods that are required to investigate forecast questions FQ-C, FQ-D, and FQ-E.

To plan a flight that allows aircraft measurements within a WCB, the forecaster is interested in the spatial and temporal distribution of WCB features in the ensemble forecast. As a summary measure of the uncertainty information, the probability of WCB occurrence, p(WCB), is of particular interest. It provides for a given location in 3D space at a given time the probability of encountering a WCB airmass. To compute p(WCB) from an ensemble weather forecast, we first need to detect WCB features in the individual ensemble members. In early studies of, for instance, Harrold (1973), Carlson (1980) and Browning (1986), conveyor belt airstreams have been identified by manual inspection of satellite imagery or by isentropic analysis. Subsequent studies have used Lagrangian particle trajectories computed with wind fields from numerical model output to investigate case studies of extratropical cyclones. For example, Whitaker et al. (1988) and Hibbard et al. (1989) show the existence of three distinct airstreams in a modelling case study of the 1979 “President’s Day storm” and relate the airstreams to the conceptual model by Carlson (1980). Further case studies, including Kuo et al. (1992), Schultz and Mass (1993), Mass and Schultz (1993), and Reed et al. (1994), also interpret computed trajectories in consideration of the Carlson (1980) model, however, note that they are able to identify rather a continuum of flow paths than discrete airstreams.

In more recent studies, Lagrangian particle trajectories are frequently used to *objectively* detect WCB structures in numerical model output. For this thesis, I am interested in the specific ways trajectories are used in the literature to detect WCBs. In particular, this includes the employed objective detection criteria and the spatial and temporal spacing of the trajectories as well as the employed wind fields. In Sect. 6.1, I review literature on trajectory-based WCB detection. My approach to derive 3D fields of $p(\text{WCB})$ from the detected features and its integration into Met.3D is described in Sect. 6.2. In Sect. 6.3, I analyse the sensitivity of the presented method to the grid spacing of the forecast wind fields and to the number and locations of the trajectory seeding points. From the results, I suggest a setup that is suited to be used during field campaigns. Finally, I propose a visual analysis method for cases in which only low probabilities of the occurrence of WCBs are encountered (Sect. 6.4). The method helps to distinguish the case in which only few members predict a WCB but at approximately the same location, from the case in which many members predict a WCB but the spatial variation is high. The method can be applied to probabilities of other features as well.

6.1 WCB detection based on objectively selected Lagrangian particle trajectories

Wernli and Davies (1997) introduce objective criteria to extract what they call “coherent ensembles of trajectories” (CET, a bundle of trajectories started at different locations; not to be confused with the meaning of “ensemble” in “ensemble forecasts”) from a set of trajectories covering the entire domain of interest. They use wind fields from the ECMWF global atmospheric model, interpolated (from a spectral truncation of T213) to a regular latitude-longitude grid of 0.75° × 0.75° with 31 levels in the vertical and a six-hour time interval. Trajectories are started on every model grid point below 800 hPa (approx. 7 levels). Wernli and Davies show that nearly identical CETs are obtained by selecting trajectories that experience either a moisture decrease of 12 g kg^{-1} in 48 hours or an ascent of more than 620 hPa in 48 hours. The approach allows to focus on the dynamically most relevant cores of an extratropical cyclone’s airstreams. In a subsequent article, Wernli (1997) applies the suggested method to the case study of Browning and Roberts (1994) and relates the obtained CETs to the WCB model. Unlike the analysis of a continuum of airstreams in a cyclone this method selects the strongest ascending airmasses within the WCB.

Stohl (2001) and Eckhardt et al. (2004) compute climatologies of WCBs. Stohl (2001) seeds the trajectories on a 1° × 1° grid in the horizontal and on two vertical levels at 500 m and 1500 m above sea level (asl). He notes that the results of his climatology are sensitive to the WCB selection criterion, and settles for the –as he writes– “somewhat arbitrary” criterion of 8000 m in 48 hours (the approximate time scale at which air flows through a single synoptic system). Similarly, Eckhardt et al. (2004) start trajectories on a 1° × 1° grid at 500 m asl. They note that “any criterion used for an automatic classification of WCBs is necessarily subjective”. In their work, trajectories travelling more than 10° eastward and 5° northward and ascending more than 60% of the average tropopause height within 48 hours are classified as WCB trajectories.

A number of studies use the trajectory model LAGRANTO (Sprenger and Wernli, 2015), originally introduced by Wernli and Davies (1997). Spichtinger et al. (2005) analyse ice supersaturation in the vicinity of a WCB’s outflow region, Grams et al. (2011) present a case study of an extratropical transition. Schäfler et al. (2011) analyse aircraft measurements and Madonna et al. (2014) present a climatology of WCBs. All four studies settle for a criterion of an ascent of more than 600 hPa in 48 hours to select

WCB trajectories. In terms of seeding, Schäfler et al. (2011) start their trajectories on every model grid point between the surface and 850 hPa of the deterministic ECMWF T799L91 forecast, interpolated to a regular latitude-longitude grid of 0.25° × 0.25°, and using the approximately 17 lowest levels. Madonna et al. (2014) seed their trajectories at 80 km distance in the horizontal and at 20 hPa vertical distance on levels between 1050 and 790 hPa. Their wind field is available at 1° × 1° grid spacing.

During TNF (Schäfler et al., 2014), LAGRANTO has been used with wind fields from the ECMWF ensemble forecast covering the North Atlantic and Europe. To keep the computational demand tractable for the operational forecast setting, the available ENS spectral resolution of T639 was interpolated to 1° × 1° in latitude and longitude. In the vertical, all available 62 levels were used. A six-hour time step was used. Trajectories were started for each member at 1° horizontal spacing at five levels constant in pressure between 1000 hPa and 800 hPa. The selection criterion was set to an ascent of 500 hPa in 48 hours.

In summary, the reviewed studies have all restricted trajectory seeding to lower atmospheric levels. The horizontal distance between start points mostly corresponds to the grid spacing of the driving wind fields. While the exact selection criterion for WCB trajectories varies, all studies use a criterion that filters trajectories according to a given ascent in a two day period.

6.2 Probability of warm conveyor belt occurrence

6.2.1 Computation of p(WCB)

I follow the approach of Wernli and Davies (1997) and detect WCB features by selecting Lagrangian particle trajectories according to a given ascent Δp in a given time period Δt. Trajectories are computed with LAGRANTO. I use the same ECMWF ENS wind fields as described in Sect. 5.3.1. From the available spectral truncation of T639, the wind forecasts are horizontally interpolated by the ECMWF MARS to a regular latitude-longitude grid of 1° × 1° (the same data used during TNF) and (additionally) 0.25° × 0.25°. In the vertical, all available 62 hybrid sigma-pressure model levels are used.

Once trajectories have been computed and selected, a gridded field of p(WCB) can be derived by relating each ensemble member's trajectories to a binary grid, and by computing for each grid point the relative number of members that predict a WCB feature at that grid point. In a more formal way, the method to compute p(WCB) at time t can be summarised as follows:

1. For every ensemble member m and every available forecast time step $t_0 \in (t - 48h..t)$, integrate 3D Lagrangian particle trajectories, started at a fixed set of seeding points, from t_0 forward in time for $\Delta t = 48$ hours.

2. Select those trajectories that fulfil a specified WCB criterion (for example, an ascent of $\Delta p = 600$ hPa in $\Delta t = 48$ hours).

3. For each member m, create a 3D binary grid B^m that for every grid point with indices k, j, i, B^m_{kji}, contains a set bit ($B^m_{kji} = 1$) if the grid point is located "inside" a WCB airmass at time t, where "inside" needs to be determined from the trajectory positions at t.

4. For each grid point compute the probability of WCB occurrence by counting the number of members with a set bit for the point: $p(\text{WCB})_{kji} = 1/M \sum_m B^m_{kji}$, where M denotes the number of ensemble members.

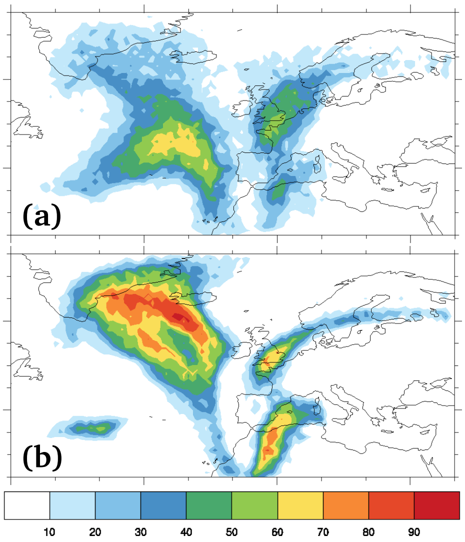


Figure 6.1: Total column probability of WCB occurrence (%), as available during TNF. Probabilities are computed from ABL-started trajectories filtered for an ascent of 500 hPa in 48 h. Forecasts from **(a)** 00:00 UTC 15 October 2012 and from **(b)** 00:00 UTC 17 October 2012, both valid at 18:00 UTC 19 October 2012. Compare to Fig. 3 in Schäfler et al. (2014).

For **t**rajectories seeded approximately in the **a**tmospheric **b**oundary **l**ayer, I call this method an *ABL-T method*. Note that the grid topology of B needs to be identical for each member in order to avoid errors due to variations in grid point positions, as is the case for probabilities derived from ECMWF NWP output (cf. Sect. 5.4).

The method poses several challenges. With respect to step (1.), trajectory seeding needs to be sufficiently dense to spatially sample the WCB features. The literature reviewed in Sect. 6.1 indicates that grid spacings of 1° or less should be sufficient. For step (2.), the WCB criterion must be carefully chosen, as the ascent that a trajectory experiences may depend on factors including seasonal variability or the horizontal and vertical grid spacing of the employed wind forecasts. Also, interactivity must be considered to enable a user to change Δp and Δt during forecasting to judge the sensitivity of $p(\mathrm{WCB})$ on these parameters. Third, we need to find a suitable gridding strategy that determines in step (3.) whether a grid point is located inside a WCB airmass. The simplest approach is to extract, for each member, the particle positions of all WCB trajectories at time t, and to compute for each particle the grid cell B^m_{kji} in which it is contained.

During TNF, this simple approach was applied in 2D to compute $p(\mathrm{WCB})$ for total grid columns, as well as for three vertical intervals to distinguish inflow, ascent, and outflow. In the horizontal, a regular grid with a spacing of 1° × 1° in latitude and longitude was used. However, with this approach no physical assumptions are made about the air volume represented by each particle. The result is sensitive to both trajectory seeding points and grid topology of B, and the positions of the WCB particles are only captured with an accuracy on the order of the grid spacing of B. Also, due to the changing area of the grid cells with latitude, the result is biased towards lower probabilities close to the poles. Examples of the resulting total column $p(\mathrm{WCB})$ field are shown in Fig. 6.1 and can also been found in Schäfler et al. (2014, their Fig. 3). Due to the described issues, the results should only be interpreted in a qualitative manner.

In 3D, more complexity is added as the vertical extent of the grid cells also has to be taken into

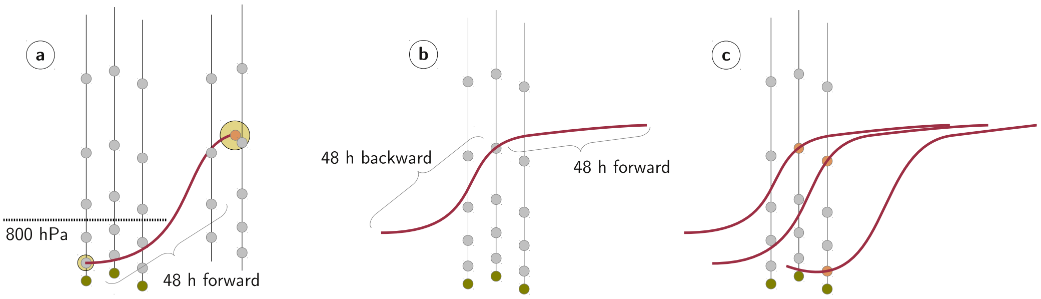


Figure 6.2: Methods to compute p(WCB). **(a)** ABL-T method using trajectories started in the atmospheric boundary layer and integrated 48 hours forward in time. To get 3D gridded information on WCB location, an air parcel volume needs to be assumed for each particle so that grid points overlapping with the volume can be determined. **(b)** DF-T method using domain-filling trajectories started from every grid point of the p(WCB) grid and integrated both 48 hours forward and backward in time. No volume has to be assumed as selected WCB trajectories are located exactly on grid points **(c)**.

account. To eliminate bias and sensitivity, one possibility is to assume an air parcel mass and geometry for the trajectory particles, as illustrated in Figure 6.2a. In the example, the particle is associated with a spherical air parcel. Given the required thermodynamic variables at the particle position at start time t_0 and gridding time t, the volume and thus radius of the parcel at t can be computed and the overlapping grid points found. However, due to the large difference in vertical and horizontal scale of our grids (on the order of 100 km in the horizontal and 100 m in the vertical), the usage of spherical geometry requires the computation of a very large number of trajectories. Yet, geometry that reflects the different scales (for example ellipses, cylinders or simple rectangular boxes) is difficult to motivate physically. Also, usage of large air parcels neglects potential deformation of the parcels by the wind field.

An approach not requiring any such assumptions is to use **d**omain-**f**illing **t**rajectories (in the following referred to as *DF-T method*). Here, I first specify the grid topology for B. Next, as illustrated in Fig. 6.2b and c and Fig. 6.3, for every member and each grid point B^m_{kji}, a trajectory starting on B^m_{kji} is computed. This way, I can be certain that each B^m_{kji} is placed exactly on a trajectory and no assumptions about shape of the particle volume need to be made. After applying a WCB selection criterion to the trajectories, the bits of the grid points from which WCB trajectories were started are set. However, the approach requires increased computational resources. Seeding points are now required on all tropospheric layers and hence a larger number of trajectories is required. Also, trajectories additionally have to be computed backward in time to also capture those situations in which a WCB trajectory passes its seeding point in the ascent or outflow phase. Step (1.) in the method description above is hence extended to also integrate the trajectories backward in time for Δt hours from time t.

As an example, Fig. 6.3 shows results of selecting domain-filling trajectories that ascend more than 500 hPa in 48 h (Fig. 6.3a–c) and more than 600 hPa in 48 h (Fig. 6.3d–f). Note how the 30 % isosurface of p(WCB) over the English Channel almost vanishes with 600 hPa filtering (Fig. 6.3f).

In Sect. 6.3, I compare four DF-T and ABL-T setups with varying grid topology with respect to obtained p(WCB) and to computational demand. The comparison allows to find a setup well suited for usage in campaign forecasting.

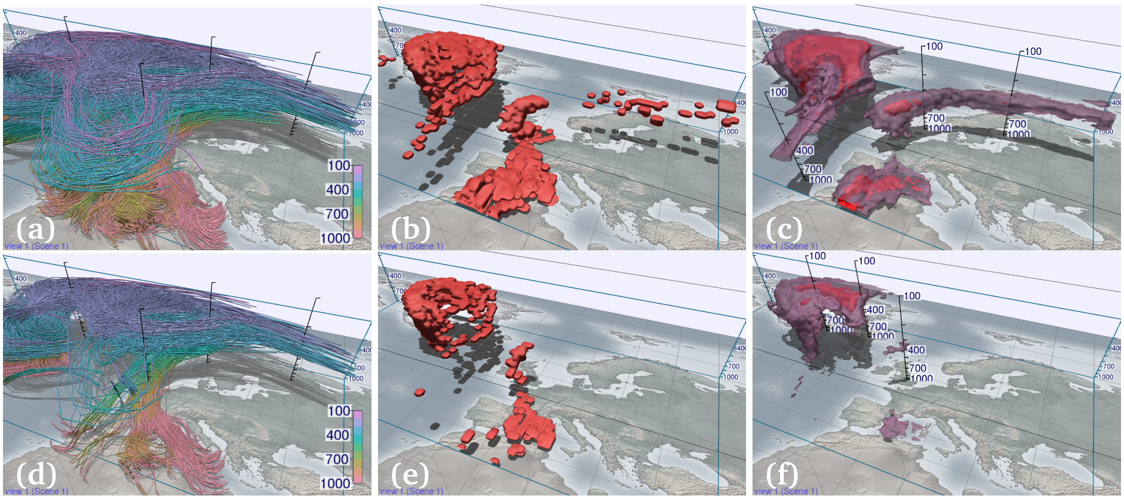


Figure 6.3: Derivation of p(WCB) with DF-T setup (S3.). **(a, d)** Trajectories started at 18:00 UTC 19 October 2012, computed with wind fields from the ensemble control forecast from 00:00 UTC 17 October 2012, integrated forward and backward in time for 48 h each. Trajectories are selected according to an ascent of **(a–c)** 500 and **(d–f)** 600 hPa in 48 h. Colour encodes altitude (hPa). **(b, e)** Volume rendering of the binary grid B, representing the start positions of the selected trajectories. **(c, f)** Probability of WCB occurrence derived from all 51 members of the ensemble. The red opaque isosurface shows 30 % probability, the purple transparent isosurface 10 % probability. Vertical axes are labelled with pressure altitude (hPa).

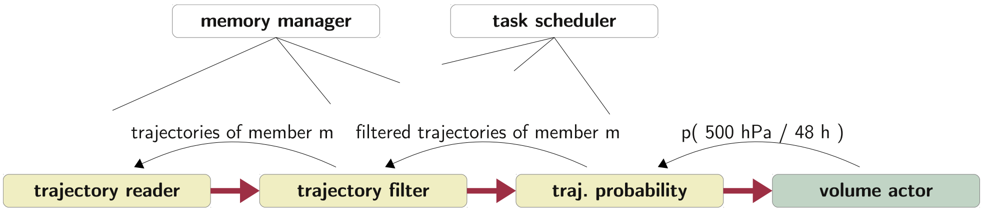


Figure 6.4: Sample Met.3D data processing pipeline to visualize p(WCB), depicted analogous to Fig. 5.13. Different pipeline modules (yellow) are responsible for reading and selecting trajectories, and for computing the p(WCB) field. In the example, a volume actor (green; for details see Sect. 5.2) visualizes the resulting p(WCB) data. A request for the probability of occurrence of trajectories, emitted by the volume actor, triggers further requests up the pipeline. Intermediate results are cached by the memory manger, connected to each pipeline module (indicated by the black lines). Pipeline execution can be parallel and is controlled by a task scheduler, also connected to all pipeline modules (for details see Sect. 5.3.2).

6.2.2 Implementation

Trajectories computed with LAGRANTO are stored in NetCDF files. Trajectory selection and the computation of p(WCB) take place in Met.3D and have been implemented in a number of modules in the Met.3D data processing pipeline (cf. Sect. 5.3.2). Analogous to Fig. 5.13, Fig. 6.4 shows an example setup. Separate pipeline modules are responsible for reading trajectory data from disk, filtering the data according to the selection criterion, gridding and probability computation. This architecture allows modules to be exchanged when, for example, data from a different trajectory model should be read or a different selection criterion should be applied.

Hardware permitting, parts of the pipeline (for example, trajectory selection) can be executed in parallel. Intermediate results in the pipeline are cached by Met.3D's memory manager (cf. Sect. 5.3.2). Both parallel execution and caching increase the interactivity of the system with respect to changing the selection parameters Δp and Δt.

To select trajectories according to the ascent criterion, the maximum pressure change occurring within a trajectory over the time interval Δt is required. For the grid spacings used here, the data volume of the trajectories of all members amounts to multiple GB per time step if stored in binary NetCDF format (approximately 2.4 GB for 1° horizontal grid spacing if a vertical region of interest of 52 levels is selected, and approximately 38 GB if the horizontal grid spacing is decreased to 0.25°). Reading the data from disk and performing the selection can hence be slow. I thus make use of the fact that the only information required to compute the probabilities is whether the trajectory started from a grid point fulfils the selection criterion. The data volume that needs to be loaded can be largely reduced by precomputing the maximum pressure change Δp for a range of time intervals Δt. Now, for a given Δt, only the maximum Δp for each trajectory (= grid point) needs to be read. The selection process is reduced to comparing each trajectory's Δp to the given threshold value. This way, I am able to provide an interactively adjustable selection criterion to the user.

6.3 Choice of p(WCB) method and grid spacing for forecasting

To use a p(WCB) method for forecasting during a campaign, a number of criteria need to be fulfilled:

(a) The trajectories need to be computed in a short period of time (for our application this is preferably less than one hour), so that results are available soon after the forecast wind fields become available from ECMWF (with "available" I mean that the retrieval of the forecast data from MARS, yielding the interpolated wind fields described in Sect. 5.3.1, has finished),

(b) the amount of trajectory data needs to be small enough to be handled interactively in Met.3D,

(c) the grid spacing needs to be fine enough to capture the important features that are present in a "best possible" forecast.

6.3.1 Evaluated setups

I evaluate four different setups with respect to the given criteria:

S1. As the "best possible" p(WCB) forecast, I use a DF-T setup with trajectories computed on the ECMWF ENS wind fields at the highest available grid spacing (T639L62 spectral resolution, horizontally interpolated by MARS to a regular grid of 0.25° × 0.25° in latitude and longitude, with 62 hybrid sigma-pressure levels in the vertical). Care must be taken with respect to the choice of the B^m and p(WCB) grids. A straightforward choice is to use the ECMWF grid on which the wind fields are available. However, as discussed in Sects. 2.1.2 and 2.4, the vertical position of the grid points on all but the uppermost hybrid sigma-pressure levels depends on the surface pressure field, which varies between ensemble members and time steps. Hence, if for a given time step the individual members' wind grids are used for the B^m, the problem described in Sect. 5.4 arises: The grid points are located at different vertical positions across the ensemble, and hence an error is introduced when computing the probability. To avoid this problem while staying as close as possible to the ECMWF grid, I use the grid defined by the time step's ensemble minimum surface pressure for the B^m of all members. The minimum surface pressure is chosen to ensure that all grid points are located above the surface (if the mean surface pressure is used, grid points in the lowest levels can be located below the surface in some members). I focus on a vertical region of interest of up to approximately 100 hPa and for the B^m and p(WCB) grids discard the model levels above this elevation (retaining the lower 52 levels).

S2. The same setup as (S1.), but with horizontal wind field, B, and p(WCB) grid spacing reduced to 1° × 1°. As in (S1.), the lower 52 vertical levels are used for B and p(WCB).

S3. The same setup as (S2.), but with B and p(WCB) grids defined by a constant surface pressure of 1000 hPa, not by the ensemble minimum surface pressure. The wind forecast data remain as in (S2.). The advantage of this setup is that the p(WCB) grid can be interpreted as a structured pressure level grid and thus be visualized much more efficiently (cf. Sect. 5.3.4). This way, the interactivity in Met.3D can be improved. The drawback, however, is that some of the lower-level grid points are now located below the surface and become invalid. This reduces the vertical grid spacing in the lower troposphere above mountainous terrain.

S4. An ABL-T setup using a grid B that is regular in the horizontal with a grid spacing of 1° × 1° as in (S2.) and (S3.). In the vertical, the grid is regular in pressure with a grid spacing of 10 hPa. This spacing is on the order of the average spacing of the model level grids used in (S2.) and (S3.), and results in a comparable number of vertical levels in the region of interest (90 levels between 1000 hPa and 100 hPa). Usage of a regular pressure level grid can be motivated physically: From hydrostatic balance (e.g. Wallace and Hobbs, 2006, Sect. 3.2), it is known that for a column of air with constant mass m the difference in pressure δp between top and bottom boundary of the column stays constant with height: $-\delta p = g\rho\delta z = mgA$, where g is the acceleration due to gravity (assumed to be constant), ρ the density of the air, δz the geometric height of the column and A the cross-sectional area of the column. I start the trajectories on those grid points of B that are located below 700 hPa and classify a grid point as belonging to a WCB if a particle is positioned in the corresponding grid cell. This way, while I implicitly assume a particle geometry that is rectangular in longitude, latitude and pressure, the mass represented by the particle remains constant when rising at constant latitude. The artefact of decreasing grid cell area A towards the poles remains, though. For trajectory integration, the same forecast data as in (S2.) and (S3.) are used.

For all trajectory computations, LAGRANTO is driven with ECMWF ENS forecast data at six-hour time steps. The model internally uses a 30-minute time step for the integration, trajectory positions are output at six-hour intervals.

6.3.2 Setup comparison

In terms of computational resources, setup (S1.) is the most demanding configuration. On my test system (six-core Intel Xeon running at 2.67 GHz; 24 GB RAM; 512 GB solid state drive), the computation of the trajectories of a single time step takes about 50 CPU minutes per member. The data output for a time step of all members, stored in binary NetCDF format, amounts to approximately 38 GB. While such simulations are feasible for research settings, they are not suited for forecasting. For setups (S2.) and (S3.), the number of trajectories decrease by a factor of 16. The time required to compute the trajectories reduces to about three CPU minutes per time step and member, about 2.4 GB of trajectory

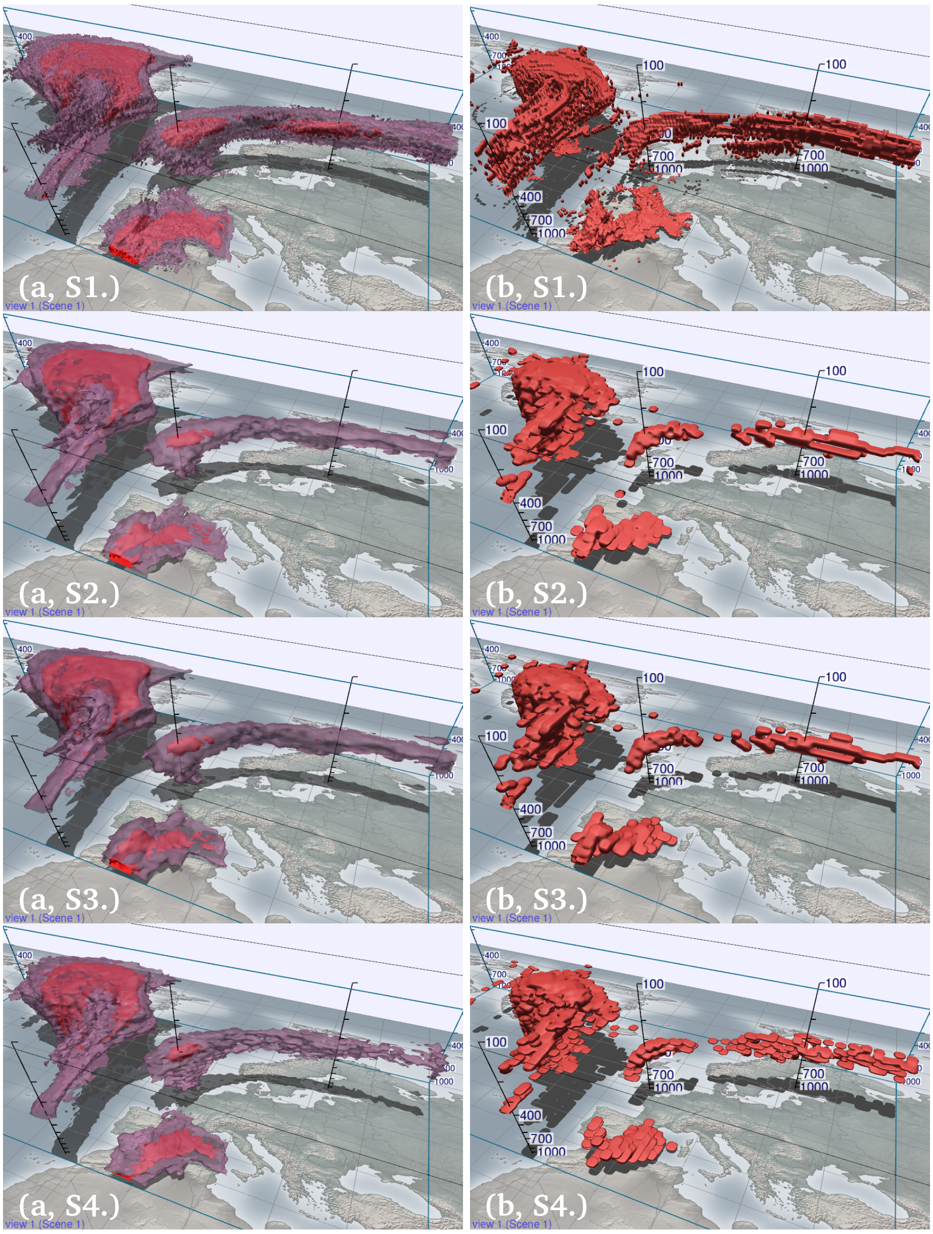


Figure 6.5: Comparison of setups to compute p(WCB). Same forecast as in Fig. 6.3. The selection criterion is set to 500 hPa in 48 h. **(a)** Volume rendering of p(WCB) (red opaque isosurface shows 30 % probability, purple transparent isosurface 10 % probability). **(b)** Volume rendering of the binary grid B of a single member (member 12), representing the WCB features for this member. **(S1.)** DF-T setup with a horizontal grid spacing of 0.25° × 0.25° , in the vertical the lower 52 (of 62) hybrid sigma-pressure levels (up to approximately 100 hPa) defined by the ensemble minimum surface pressure are used. **(S2.)** As (S1.) but with 1° × 1° horizontal grid spacing. **(S3.)** As (S2.) but with vertical levels defined by a constant surface pressure of 1000 hPa. **(S4.)** ABL-T setup with 1° × 1° horizontal grid spacing and a regular vertical grid with a grid spacing of 10 hPa.

data are produced per time step for the entire ensemble. With the current ENS size of 51 members, this setting is feasible for forecasting if a small compute cluster is available. For setup (S4.), the time further reduces to about one CPU minute and data volume reduces to approximately one GB.

In Figs. 6.5 and 6.6, the four setups are compared by means of four typical visualizations of the Met.3D workflow: (a) the volume rendering of $p(\mathrm{WCB})$ isosurfaces already used in Fig. 6.3c, (b) a volume rendering of WCB features in forecast member 12 (as captured by the binary grid B^{12}), (c) a horizontal section at 410 hPa through the ascent region associated with precipitation, and (d) a horizontal section through the inflow region at 950 hPa. The TNF forecast case of 19 October 2012 that already served for the examples in Chapter 5 is used. The main features (cf. Fig. 1.3: inflow over the Mediterranean Sea, ascent over the English Channel and Southern England, outflow over Scandinavia and Russia, as well as a strong ascent associated with former Hurricane Rafael over the North Atlantic) are well represented by all setups. However, in the regions of maximum $p(\mathrm{WCB})$, setups (S2.) and (S3.) predict probabilities that are decreased by about 10% compared to the "reference" setup (S1.). This is visible in the smaller extent of the 30% isosurface in Fig. 6.5a as well as in the horizontal sections (Fig. 6.6c and d). Also, single member WCB structures are more solid in setup (S1.), as illustrated in the 3D view of the binary volume of member 12 (Fig. 6.5b). The decrease is caused by the lower horizontal grid spacing of the driving wind fields, in which fewer trajectories experience strong ascent – potentially due to smoothed vertical velocities. Nevertheless, setups (S2.) and (S3.) capture the shape and location of the $p(\mathrm{WCB})$ features equally well as (S1.).

The differences between setups (S2.) and (S3.) are negligible. While virtually no differences can be found in the visualizations of the WCB ascent at 410 hPa (Fig. 6.6c), the differences become more pronounced in the lower atmospheric layers (Fig. 6.6d). This can be explained with the grid topology: At higher altitudes, the elevation of the model levels becomes increasingly independent of surface pressure and hence the difference in the $p(\mathrm{WCB})$ grids vanishes. However, even at low altitudes the observed differences in $p(\mathrm{WCB})$ remain within a few percent.

The bottom rows of Figs. 6.5 and 6.6 show the results for the ABL-T setup (S4.). Despite the crude assumption with respect to air parcel geometry, the major $p(\mathrm{WCB})$ features are captured well. However, this setup tends to predict slightly higher probabilities compared to (S2.) and (S3.) in the atmospheric boundary layer, and slightly lower probabilities at higher altitudes.

Results for other time steps are similar (not shown). I conclude that from the presented candidates, setups (S3.) and (S4.) are best suited to be used in a forecast setting. While showing small differences with respect to the absolute predicted values, both capture the shape and locations of regions of elevated $p(\mathrm{WCB})$. Also, both are feasible to compute in less than an hour and the results can, due to the structured vertical grid layout, be visualized more efficiently than the results computed by the setups based on hybrid sigma-pressure vertical coordinates (cf. Sect. 5.3.4 and Table 5.4).

6.4 Probability region contribution

The methods introduced so far allow to visualize the computed $p(\mathrm{WCB})$ fields and to find regions in which the occurrence of a WCB is most likely. However, it remains an open question how the magnitudes of the displayed probabilities should be interpreted. A distinct property of the examples presented in Sect. 6.3 are relatively low probabilities. For instance, in Fig. 6.3c maximum values only reach about 30%. As mentioned in the introduction, such low magnitudes can have two causes: Either indeed only 30% of all ensemble members predict the WCB event, or large spatial variation of the features in the individual members causes only marginal overlap and thus low probabilities. Also, noise in the individual binary volumes can cause empty grid cells in the features and decrease probability values. Interpreting the data correctly and being able to distinguish between these causes is very important for

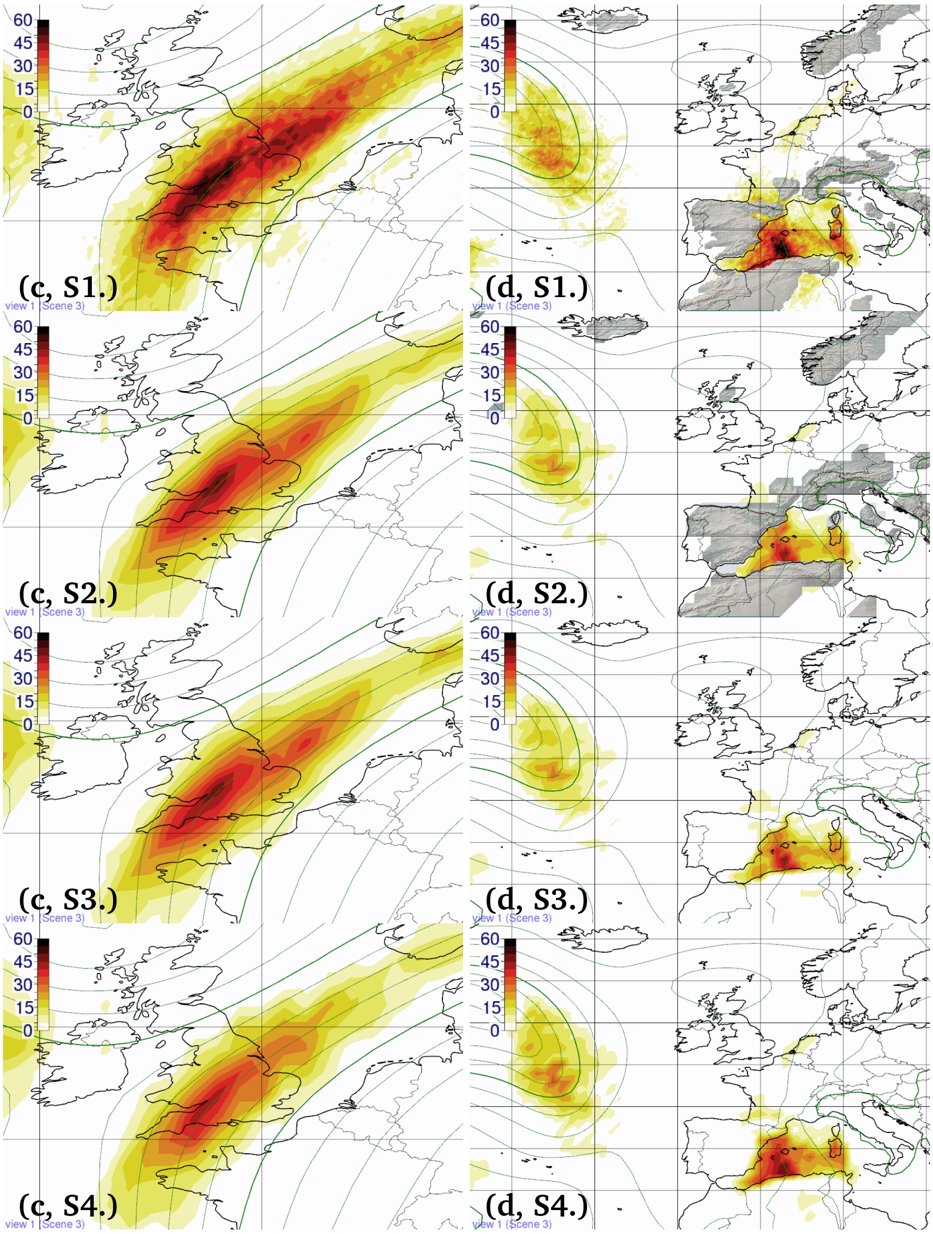


Figure 6.6: Same as Fig. 6.5, but showing: **(c)** Horizontal section of $p(\text{WCB})$ at 410 hPa. **(d)** Horizontal section of $p(\text{WCB})$ at 950 hPa. Colour coding in %. Green contour lines show ensemble mean geopotential height.

making decisions on potential flight routes.

The issue can be approached by looking at the individual ensemble members, as illustrated in Fig. 6.7. While due to limited print space Fig. 6.7 only shows a small selection of members, we indeed find that much more than 30% of the members predict a WCB feature. However, it is difficult for a human user to remember how many of the 51 members showed a WCB feature. Visualizing the WCB features of all members in a single view (Fig. 6.7f) results in massive clutter and, thus, does not reveal insight.

I am interested in providing the following information: Given a region bounded by a probability isosurface, how many individual ensemble members predict a WCB feature that overlaps with this region and that, thus, contributes to the probability value at any of the grid points inside the isosurface? To

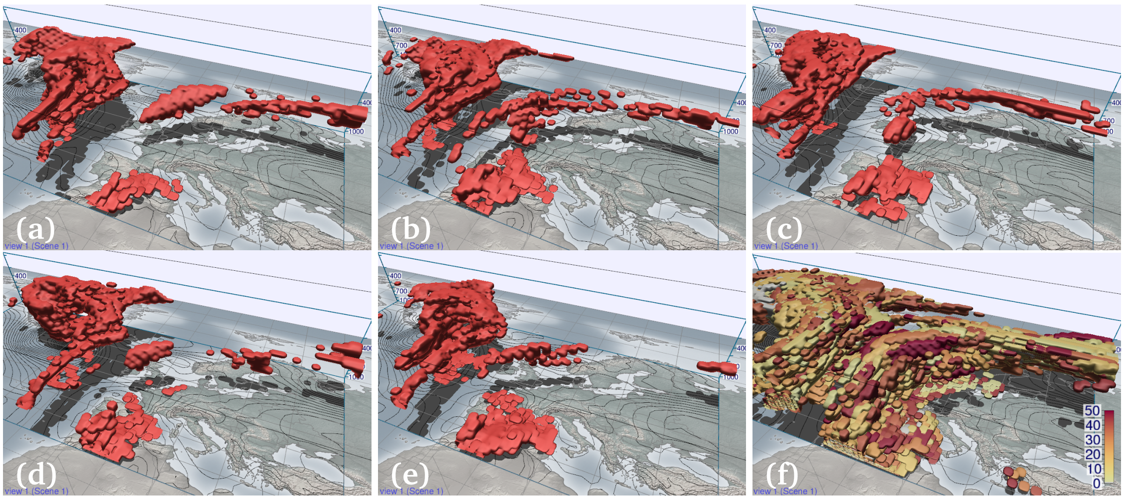


Figure 6.7: WCB features (binary grids B^m) of further members m of the forecast shown in Fig. 6.3b. Members **(a)** 2, **(b)** 4, **(c)** 34, **(d)** 36 and **(e)** 42. Note that location and shape of the WCB features vary strongly. **(f)** WCB features of all 51 members of the ensemble, visualized in a single image and distinguished by colour (colour coding denotes member number). Black contour lines in all images show sea level pressure of the corresponding member (of the ensemble mean in (f)).

determine this number of members, I propose a method that applies region growing to identify the grid points inside the isosurface, then uses the members' binary grids B^m to determine which members have contributed. To efficiently make use of the B^m, I condense the binary grids into bitfields that are stored together with the probability volume. For the current example and for the 51 members of the ECMWF ensemble, each grid point $p(\mathrm{WCB})_{kji}$ is augmented by a bitfield stored in a 64-bit integer variable (one bit for each member). The bitfields are generated during evaluation of the probability criterion (in this case, step (3.) in Sect. 6.2).

Fig. 6.8 illustrates the approach. In a hypothetical ensemble of ten members, nine members predict a WCB feature (coloured bars). However, the maximum probability value that occurs is 30% (red region). To determine the contribution to the region, the algorithm scans the volume for grid points exceeding the 30% value. Starting from the first identified point, a region growing algorithm determines all grid points belonging to the red region. Combining the bitfields of the identified points with a bitwise "or"-operation reveals that in total, members 1, 2, 3, 4, 5, 6, 8 and 9, thus 80% of the ensemble, contribute to the region. We hence know that much more than 30% of all members predict a WCB. The information is stored for each of the identified grid points in a separate data field, the *contribution volume*. It needs to be recomputed every time the probability isovalue changes. For example, applying the algorithm to the white 10% region in Fig. 6.8 yields a contribution of 90%.

The contribution volume can be used in visualizations of $p(\mathrm{WCB})$ to colour a probability isosurface according to the number of members that contribute. Fig. 6.9 shows the application of the method to the WCB forecast from Fig. 6.3c, setup (S3.). Whenever an isosurface point is identified and visualized (cf. Sect. 5.3.4 for the raycasting algorithm), the eight data points that enclose the isosurface position are sampled. Since the isosurface value is interpolated from these eight points, at least the point with the maximum probability value is located inside the isosurface, and the point with the lowest value is located outside the isosurface (otherwise no crossing could be found between the points). Thus, by sampling the contribution volume at the grid point with the maximum value (and exploiting the fact that

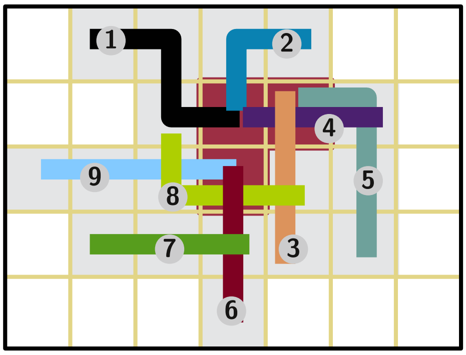


Figure 6.8: Schematic 2D example of a case in which many ensemble members predict a WCB feature but spatial variation causes low probability values. Consider an ensemble of ten members, of which nine members predict a WCB feature (depicted by the different coloured and numbered lines). In the example, only a maximum of three features overlap in any grid cell (resulting in a maximum probability of 30 %; red grid cells). By storing the indices of all members that contribute to a given grid cell, the presented method is able to determine the members that contribute to a probability region. In the example, 8 members (that is, 80 %) contribute to the red region. The grey grid cells illustrate the 10 % region.

all grid points of a contiguous structure in the contribution volume carry the same value) the number (or percentage) of contributing members can be obtained. Indeed, Fig. 6.9c shows that about 85% of the example's ensemble members contributed to the 30% isosurface – an immediate hint to the forecaster to have a closer look at the predicted structure.

In addition, region growing can be applied to yield information on how many disjoint WCB features contribute from a particular member, and how the sizes of these features compare to the size of the region bound by the probability isosurface. The diagram in Fig. 6.9d is displayed by Met.3D when the user selects an isosurface with the mouse pointer. It shows the sizes of the WCB features in the individual members in a stacked box plot. The size of the probability isosurface is displayed by the red line. Single features are divided into solid bars, depicting the fraction of the feature that overlaps with the probability isosurface, and a transparent bar, depicting the full size of the feature. If more than one feature contributes from a given member, each disjoint feature is shown in a different colour. For the example in Fig. 6.9, this information reveals further insight: first, most members contribute exactly one contiguous feature; second, these features are for the most part substantially larger than the isosurface region (also compare the size of the probability isosurface to the WCB features in Fig. 6.7). The user can infer that most members' features indeed represent WCB events. A WCB is hence very likely to occur.

Of course, the method can also be applied to probability fields other that $p(\text{WCB})$; similarly low probabilities can also occur for features derived from other NWP fields.

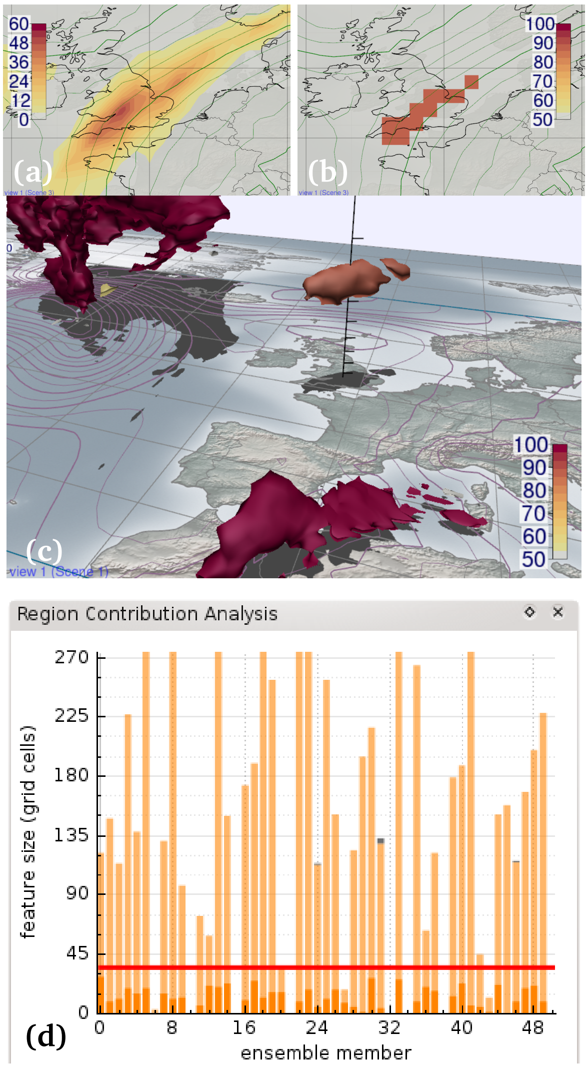


Figure 6.9: Application of the region contribution algorithm to the WCB forecast from Fig. 6.3c. **(a)** Horizontal section of p(WCB) at 415 hPa over Southern England (colour coding in %). **(b)** Grid boxes that intersect with the 415 hPa surface and that exceed the isosurface threshold of 30 % (the red isosurfaces in Fig. 6.3c), coloured by the percentage of contributing members as identified by the region growing algorithm (colour coding in %). Green contour lines in (a) and (b) show ensemble mean geopotential height. **(c)** The 30 % isosurfaces of Fig. 6.3c coloured by the percentage of contributing members. Purple contour lines show ensemble mean sea level pressure. **(d)** Size (in grid cells) of WCB features in the members contributing to the 30 % isosurface above Southern England. If multiple features contribute from a given member, they are stacked using different colours (in the example, small secondary features exist in members 24, 31, and 46). The bar of each feature is divided into total feature size (light colour) and the fraction of the feature that overlaps with the 30 % isosurface (solid colour). The red horizontal line marks the size of the 30 % isosurface.

CHAPTER 7

Case study: T-NAWDEX-Falcon IOP 3

At this point, all visualization and analysis methods are available that are required to use Met.3D to answer the forecast questions listed in the introduction. This chapter demonstrates how Met.3D can be used in practice. The presented case study revisits the TNF forecast case for October 19, 2012 (Fig. 1.3), the case that has already been used in the examples in the previous chapters and that is also discussed in Schäfler et al. (2014). The case study is accompanied by *Video 2* (cf. Sect. 1.5), the Supplement to Rautenhaus et al. (2015a). The video helps to convey the full value of Met.3D's interactive 3D visualizations. It contains this chapter's static figures, as well as additional content, in animated form and is intended to be used side-by-side with the thesis. Start times for the video are provided throughout the following text. To compute p(WCB), setup (S3.) from Sect. 6.3 is used.

Assume the forecast activities to take place on Monday, 15 October 2012. The ensemble and deterministic predictions initialised at 00:00 UTC on that day, as well as the preceding model runs, are available to the forecaster (in the following, I abbreviate forecast valid times as "12Z/19" for 12:00 UTC 19 October 2012, and forecast initialisation (or base, or run) times as "IT00Z/15" for 00:00 UTC 15 October 2012). We are interested in areas that favour WCB development in Central Europe, being reachable with the DLR Falcon aircraft from the campaign base in Oberpfaffenhofen, Southern Germany. Due to requirements from air traffic authorities, potential flight routes need to be announced at least three days in advance of a flight. Hence, our aim is to explore the atmospheric situation in order to evaluate suitable flight conditions towards the end of the week.

7.1 Weather situation

Our first step is to study the large scale weather situation in the deterministic high-resolution forecast to analyse whether a promising synoptic situation will develop (forecast question FQ-A). The upper level flow is of particular interest. WCBs frequently occur on the leading edge (i.e. downstream) of troughs

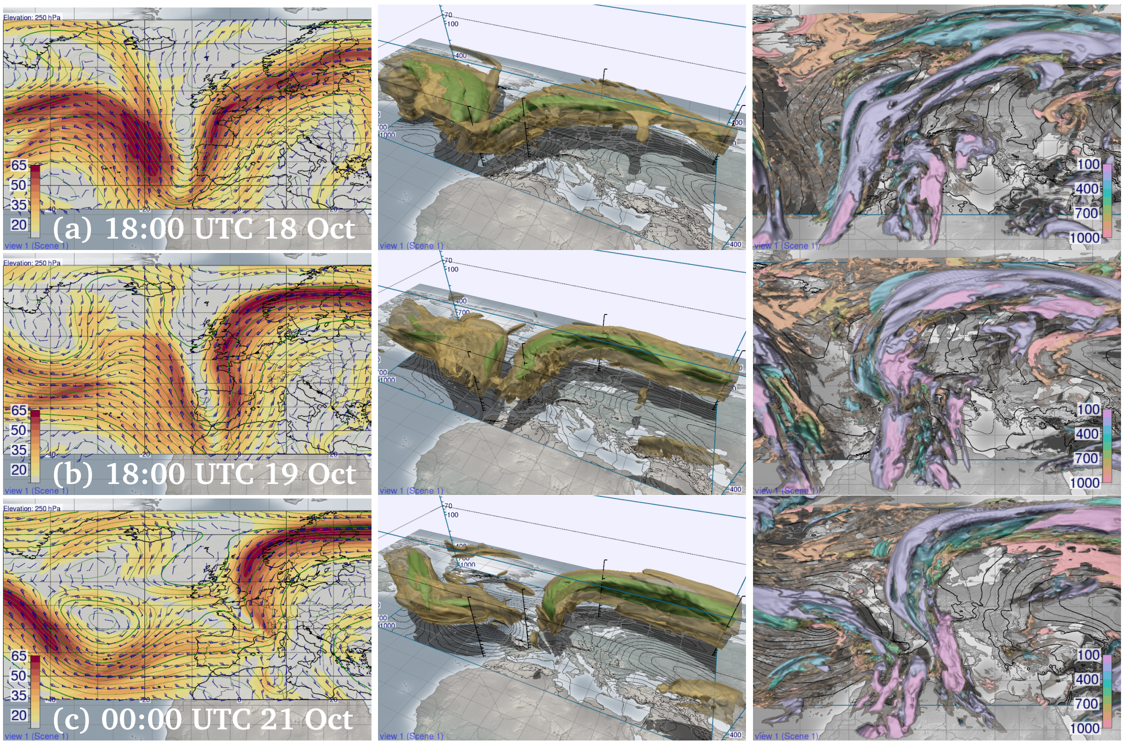


Figure 7.1: Time sequence of (left) horizontal section with contour lines of geopotential height and filled contours of wind speed (ms^{-1}) at 250 hPa, (middle) jet stream (opaque isosurface 50 ms^{-1}, transparent isosurface 30 ms^{-1}, and black contour lines of sea level pressure) and (right) clouds (opaque isosurface cloud cover fraction of 0.7, transparent isosurface cloud cover fraction of 0.2, and black contour lines of sea level pressure). Colour coding in the right panel denotes cloud elevation in hPa. Deterministic forecast from 00:00 UTC 15 October 2012, valid at **(a)** 18:00 UTC 18 October 2012, **(b)** 18:00 UTC 19 October 2012, and **(c)** 00:00 UTC, 21 October 2012.

(where low pressure systems develop), and WCB outflow is often associated with jet streaks. We start with a Met.3D configuration featuring three views: a horizontal section of wind speed and geopotential height (initially placed at jet stream level at 250 hPa), 3D isosurfaces of wind speed, and 3D isosurfaces of cloud cover. We explore the time period from Wednesday, 17 October, to Sunday, 21 October. Figure 7.1 shows screenshots of the individual views at three selected time steps. To capture the 3D spatial structure of the jet, the isosurfaces of wind speed are visualized at 30 ms^{-1} and 50 ms^{-1}. Cloud cover is visualized by isosurfaces at 0.2 and 0.7, the latter coloured by elevation. Both 3D views contain contour lines at surface level showing the mean sea level pressure. Video 2 shows the Met.3D window with the full time animation.

A number of events of interest to our objectives can be observed: A distinct trough over the Atlantic moves eastward and narrows over time. At the same time, high pressure over Central and Eastern Europe intensifies. At upper levels, a pronounced jet stream extends from Spain over Southern England to Scandinavia, causing strong winds over Western Europe blowing from a southerly direction. On the leading edge of the trough, upper level cirrus clouds are embedded in the jet, whereas upstream, i.e. on the rear side of the trough, only scattered low level clouds are present. Further upstream (south of Greenland in Fig. 7.1), the large-scale flow and cloud field are perturbed by the extratropical transition of former Hurricane Rafael (cf. Fig. 1.3; cloud field visible in Video 2). It approaches from the south and transforms into an extratropical cyclone. The leading edge of the trough, covering France and Southern

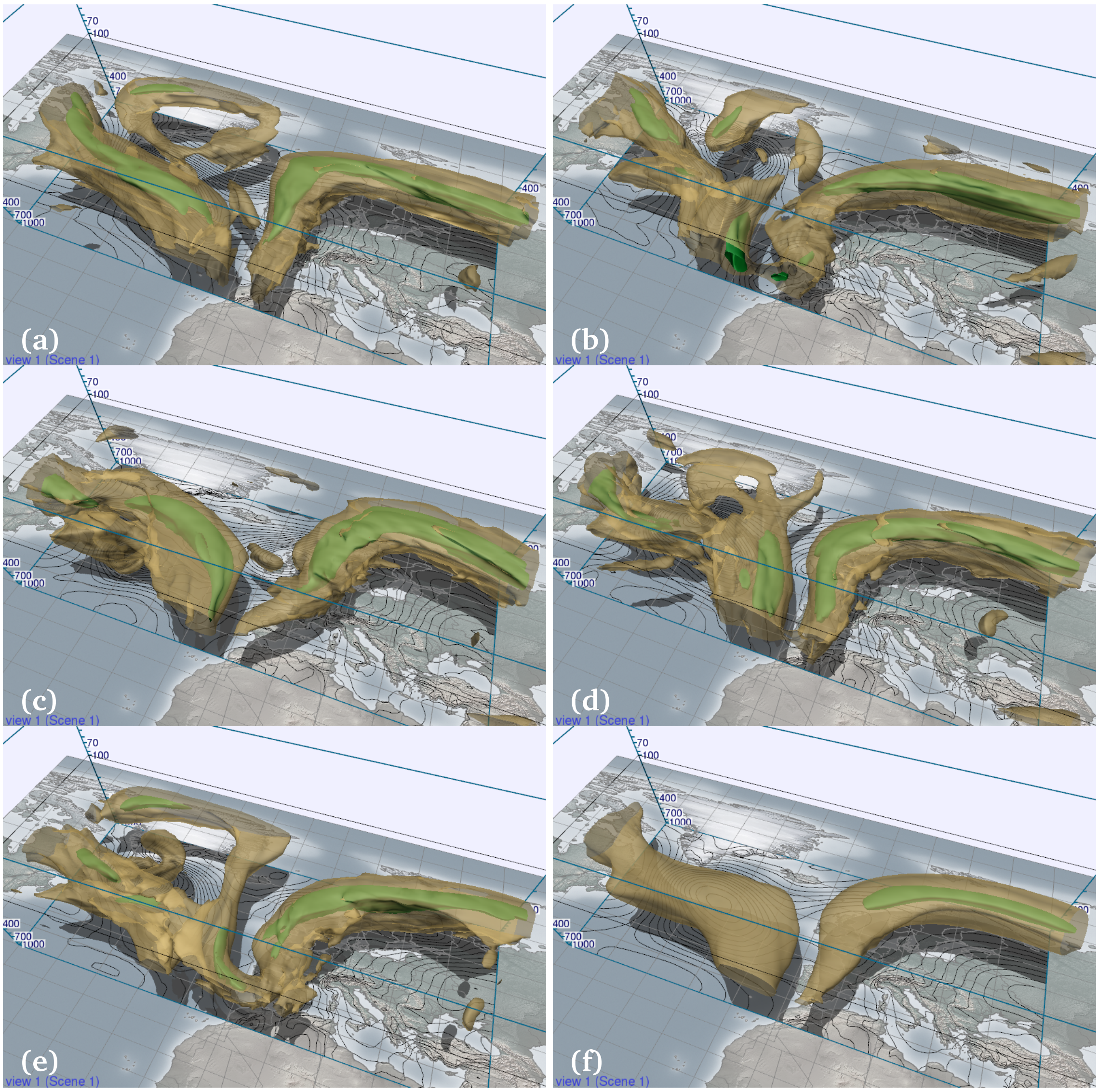


Figure 7.2: Navigation through the ensemble. Members **(a)** 27, **(b)** 33, **(c)** 37, **(d)** 43, **(e)** 45 and **(f)** the ensemble mean of horizontal wind speed (forecast from 00:00 UTC 15 October valid at 18:00 UTC 19 October 2012). Shown are the 50 ms^{-1} (green opaque) and 30 ms^{-1} (yellow transparent) isosurfaces. Black contour lines show sea level pressure.

England, would be well reachable with the Falcon.

Before we explore further forecast data, we obtain information about the uncertainty of the forecast (FQ-B). First, we check the consistency of the deterministic forecast by comparing the currently used forecast (IT00Z/15) to the two previous runs from IT12Z/14 and IT00Z/14. Video 2 (at 00:36 min) shows how the forecast runs are toggled for the forecast valid at 18Z/19. While the IT00Z/15 and IT12Z/14 runs show a fairly consistent situation, the trough is much broader in the IT00Z/14 forecast. Also, the strong jet on its leading edge has a different shape and is located further east and further north. For specifying a flight route, this spatial uncertainty is an important factor.

To get a more comprehensive picture, we explore the ensemble forecast of IT00Z/15 for occurrence, location and intensity of trough and jet in the individual members. Figure 7.2 shows selected ensemble members and the ensemble mean of the jet stream visualization for the forecast valid at 18Z/19 (the animation over the members is contained in Video 2 at 01:14 min). The jet over Europe is present in all members with similar intensity. However, we observe variation in shape and location that is in part stronger than the difference between the IT00Z/14 and IT00Z/15 deterministic forecasts. Nevertheless, the majority of the members predict a comparable jet structure over Europe. This also becomes apparent in the ensemble mean, which despite averaging features a jet core of over 50 ms^{-1}. In contrast, the variation observed in the jet structure further upstream over the central North Atlantic is larger, indicating that the predicted evolution of the extratropical transition of Hurricane Rafael is very uncertain. Here the 50 ms^{-1} signal is smoothed out in the mean.

In summary, we conclude that at least parts of the region approximately covering France, Southern England and the Benelux will be located on the downstream side of the trough.

7.2 Warm conveyor belt occurrence

Next, we examine the $p(\mathrm{WCB})$ data to determine whether a suitable WCB event is likely to occur in our region of interest (FQ-C). Figure 7.3 shows selected time steps from the IT00Z/15 forecast; the corresponding animation is shown in Video 2 at 01:55 min. We choose an initial selection criterion of $\Delta p = 500$ hPa in $\Delta t = 48$ hours and visualize the predicted fields with a 3D isosurface of a low probability (10 %). To track the temporal evolution of the $p(\mathrm{WCB})$ field inside the isosurface (in particular the evolution of the maxima), the 3D normal curves proposed in Chapter 5 are used[1].

Indeed, we find that on both 18 October and 19 October, WCB airmasses are likely to ascend on the leading edge of the trough over France and Southern England. These airmasses are potentially of interest to a research flight. Since the normal curves reveal larger probabilities on 19 October, we focus on this day. At 12Z/19 (Fig. 7.3b) and 18Z/19 (Fig. 7.3c), the ascent signal is most apparent in the prediction. At 00Z/19 and 06Z/19 (Fig. 7.3a), the airmass is still close to the surface and too far south to be reached by a single Falcon flight. At 00Z/20 (Fig. 7.3d) and 06Z/20, the airmass has reached upper levels and WCB activity is dominated by outflow. For the campaign objectives, the time around 12Z/19 and 18Z/19 is most interesting to us: the air is ascending and hence meteorologically active (precipitation is associated with the ascending phase of a WCB), and it is located in an area that can be well reached by the Falcon. The 3D visualization allows to judge the vertical extent, shape, and elevation of the region of high probability. The normal curves, coloured by probability, reveal that the maximum values of $p(\mathrm{WCB})$ on 19 October are on the order of 20% to 30%. By moving the camera and using vertical poles, we see that the region enclosed by the 10% isosurface is tilted westwards (left column of Fig. 7.3, Video 2 at 02:29 min). At 18Z/19, the maximum is located at around 400 hPa.

Due to the low magnitudes of $p(\mathrm{WCB})$, we next intend to clarify (a) whether indeed only a few ensemble members predict the WCB, and (b) how the predicted probability changes with a changing selection criterion (FQ-D). Figure 7.4a shows a screenshot of Met.3D with the region contribution analysis (Sect. 6.4) applied to the forecast valid at 18Z/19 (Video 2 at 02:42 min). A 20% isosurface is used to capture the regions of maximum predicted $p(\mathrm{WCB})$. Indeed, for both 12Z/19 (not shown) and 18Z/19 the analysis confirms that over 85% of the ensemble members have contributed to the 20% probability region

[1] Normal curves are well suited in this case to obtain an overview of the situation, as the magnitudes of maximum $p(\mathrm{WCB})$ values and their variation between time steps are not known beforehand (hence it is difficult to choose a suitable value for an inner opaque isosurface as done for the jet visualization; see Sect. 5.2). Normal curves converge at local extrema and hence at a glance highlight maxima, regardless of their magnitude.

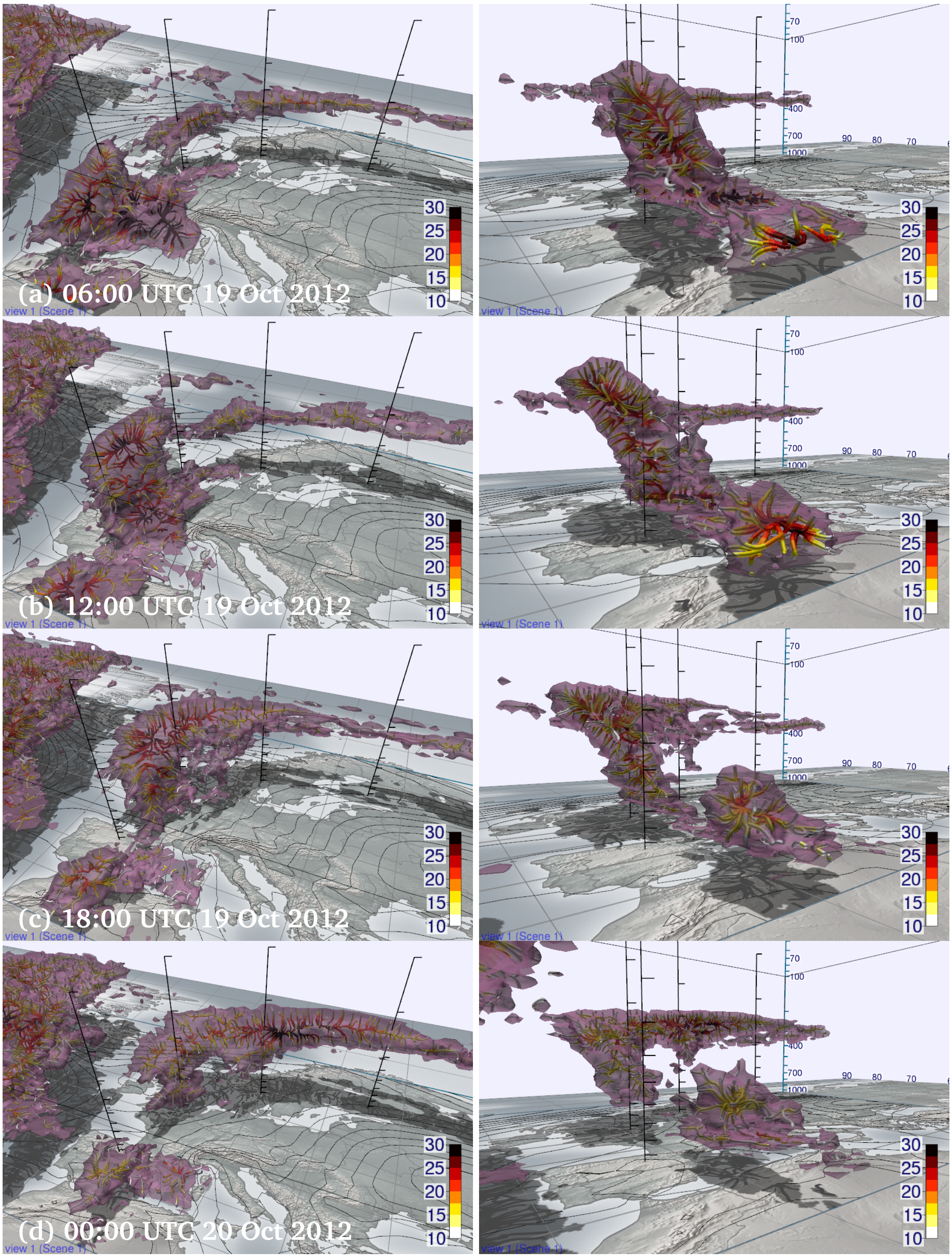
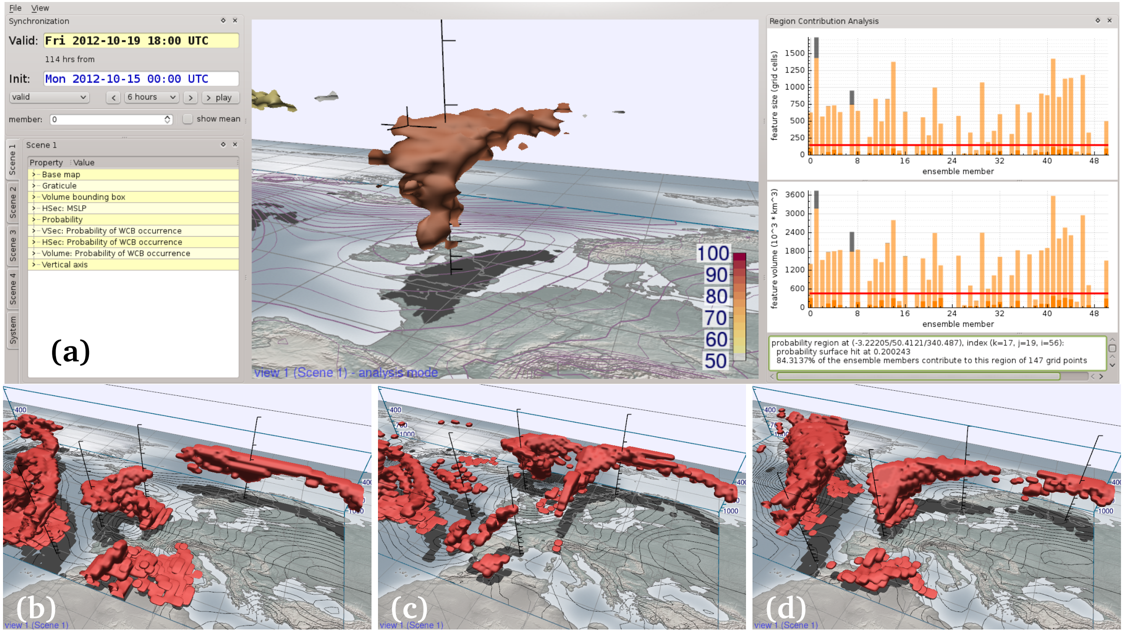

(a) 06:00 UTC 19 Oct 2012
(b) 12:00 UTC 19 Oct 2012
(c) 18:00 UTC 19 Oct 2012
(d) 00:00 UTC 20 Oct 2012
30
25
20
15
10
70
100
400
700
1000
90
80
70
view 1 (Scene 1)

Figure 7.3: Subsequent time steps of p(WCB) (computed with DF-T setup (S3.)), rendered from different viewpoints. Forecast from 00:00 UTC 15 October 2012, valid at **(a)** 06:00 UTC, **(b)** 12:00 UTC, **(c)** 18:00 UTC 19 October and at **(d)** 00:00 UTC 20 October 2012. Trajectory filtering is set to 500 hPa in 48 h. The purple transparent isosurfaces show a probability of 10 %. The interior structure of the isosurfaces is visualized using the 3D normal curves approach proposed in Chapter 5. The normal curves follow the gradient of the p(WCB) field and converge at local maxima. They are coloured by probability (%) to allow to visually track probability maxima (red to black convergence zones). Black surface contour lines show ensemble mean sea level pressure.



Valid: Fri 2012-10-19 18:00 UTC
114 hrs from
Init: Mon 2012-10-15 00:00 UTC
valid · < · 6 hours · > · > play
member: 0 · □ show mean

Scene 1

Property	Value
Base map	
Graticule	
Volume bounding box	
HSec: MSLP	
Probability	
VSec: Probability of WCB occurrence	
HSec: Probability of WCB occurrence	
Volume: Probability of WCB occurrence	
Vertical axis	

probability region at (-3.22205/50.4121/340.487), index (k=17, j=19, i=56):
probability surface hit at 0.200243
84.3137% of the ensemble members contribute to this region of 147 grid points

Figure 7.4: Region contribution analysis applied to the ensemble forecast from 00:00 UTC 15 October 2012, valid at 18:00 UTC 19 October 2012. **(a)** Screenshot of the Met.3D configuration. 20 % isosurfaces of $p(\mathrm{WCB})$ are coloured by the percentage of contributing members. The contribution distribution of the feature over Southern England is shown in the histograms on the right side of the window (feature size in (top) grid cells and (bottom) 10^3 km^3; see Fig. 6.9 for details on the diagram). **(b–d)** WCB features (binary grids B) as predicted by the individual ensemble members **(b)** 2, **(c)** 9 and **(d)** 19. Purple surface contours in (a) and black surface contours in (b–d) show sea level pressure.

over the English Channel. The difference between 20% and 85% indicates large spatial variation in the ensemble. Also, the histogram (on the right side of Fig. 7.4a) shows that the majority of individual WCB features that overlap with the 20% isosurface cover a larger volume than the resulting probability region itself. This implies that the regions that experience ascent in the individual members are larger than the region enclosed by the isosurface. To validate these findings, we animate over the individual members (Fig. 7.4b, e, and f; Video 2 at 03:16 min). Indeed, almost all members predict a WCB feature on the leading edge of the trough. However, as expected, the variability in shape and location of the predicted features is very large. In addition to members in which the WCB air ascends at 18Z/19, members in which the air is already in the outflow stage (elongated features at jet stream level) or still in the inflow stage (close to the surface) are equally present. This indicates additional temporal uncertainty. Hence, while there seems to be a good chance to sample WCB air on October 19 in the region covering Western France to Southern England, the location in space and time of the WCB ascent is still uncertain in the IT00Z/15 forecast.

To judge the strength of the predicted ascent, we modify the trajectory selection criterion. Figure 7.5 (Video 2 at 03:38 min) shows how the predicted $p(\mathrm{WCB})$ changes with Δp. By decreasing Δp (Fig. 7.5a), we can confirm a high likelihood of ascending airmasses in the region of interest[2]; the probability increases with decreasing Δp. Increasing Δp (Fig. 7.5b) reduces the predicted probabilities. However, the location of the maximum remains at the same position. The region in which high probabilities for

[2] Note that the normal curves are again advantageous for this interaction as they allow to visually track the location and magnitude of maximum probabilities despite the changing magnitudes.

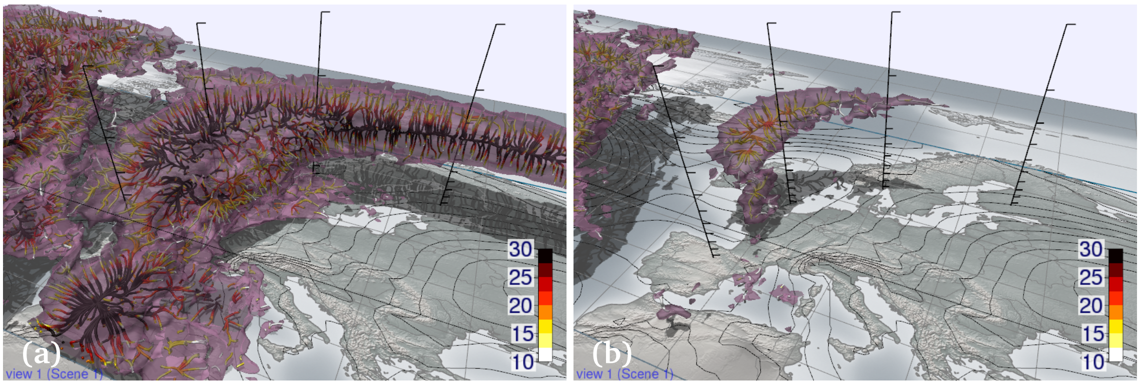


Figure 7.5: Adjusting the filter criterion for the p(WCB) forecast shown in Fig. 7.3c (forecast from 00:00 UTC 15 October 2012, valid at 18:00 UTC 19 October 2012). Filter criterion of **(a)** 400 hPa and **(b)** 550 hPa in 48 h (in Fig. 7.3c a criterion of 500 hPa in 48 h is used). The purple transparent isosurfaces show a probability of 10 %. Normal curves inside the isosurfaces are coloured by probability (%). Black contour lines show ensemble mean sea level pressure.

ascending airmasses are forecast is hence also the region in which the strongest updrafts occur.

7.3 WCB characteristics

The next goal is to characterise the predicted ascent with respect to related atmospheric processes (FQ-E). We take a closer look at the WCB trajectories of the ensemble control run and visualize the trajectory particle positions at single time steps. Animation over the time steps of the trajectories computed forward and backward from 18Z/19 reveals that the air that at 18Z/19 has ascended to the region over the Channel originates from the ABL over the Western Mediterranean Sea and Northwestern Africa around 18Z/18 (Fig. 7.6a, Video 2 at 04:10 min). It is lifted over Spain in the early hours of October 19 and over the course of the day continues its ascent over Western France, the Channel and Southern England (Fig. 7.6b and c). By vertically shifting a horizontal section of geopotential height and equivalent potential temperature of the deterministic forecast at 18Z/18 (similar to the ensemble control but chosen here for its added detail), we discover a cyclone over the Northern British Isles, and a weaker surface low located on the west coast of France (Fig. 7.6d, Video 2 at 04:28 min). South of Spain, warm and moist air (high equivalent potential temperature) is advected northward. This airmass represents the WCB inflow region; it is subsequently lifted by the WCB. In contrast, on the rear side of the trough, colder and drier airmasses over the East Atlantic are transported southward to Spain. Over the following 24 hours, the cyclone over the British Isles remains stationary, the weaker surface low moves towards Norway (Fig. 7.6e and f, Video 2 at 04:52 min). Animation over the ensemble members reveals that most other members predict similar ascents originating from the Western Mediterranean Sea and Northwestern Africa. Figure 7.7 reproduces the visualization of Fig. 7.6c for the members shown in Fig. 7.4b, c, and d. The trajectory particles that represent the WCB airmasses are lifted along similar paths. However, the temporal evolution of the WCBs differs in the members. At 18Z/19, the airmasses are at different stages of their ascent.

Figure 7.8 shows vertical sections of potential vorticity (PV) and cloud cover of the deterministic forecast valid at 18Z/19 (animated in Video 2 at 05:19 min). The dynamic tropopause, as indicated by the 2-PVU-surface, folds along the trough (Fig. 7.8a). On the rear side of the trough, dry stratospheric air

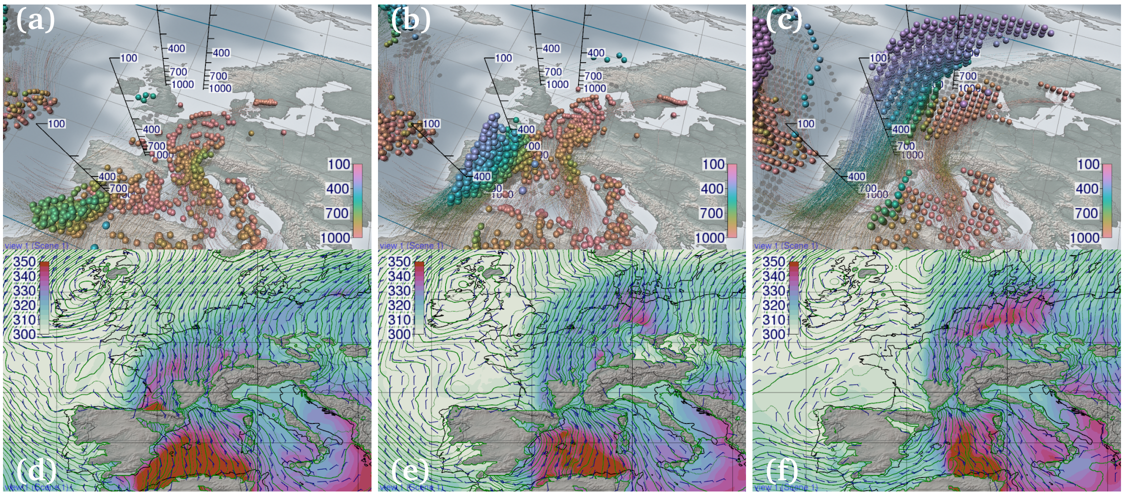


Figure 7.6: **(a–c)** Particle positions of the (backward) WCB trajectories of the ensemble control forecast, started at 18:00 UTC 19 October 2012 and computed on the forecast initialised at 00:00 UTC 15 October 2012. Colour codes pressure elevation in hPa. **(d–f)** Horizontal sections of geopotential height (contour lines), wind barbs and equivalent potential temperature (colour coded in K) of the deterministic forecast from 00:00 UTC 15 October 2012 at 950 hPa. Forecasts are valid at **(a, d)** 18:00 UTC 18 October 2012, **(b, e)** 06:00 UTC 19 October 2012, and **(c, f)** 18:00 UTC 19 October 2012.

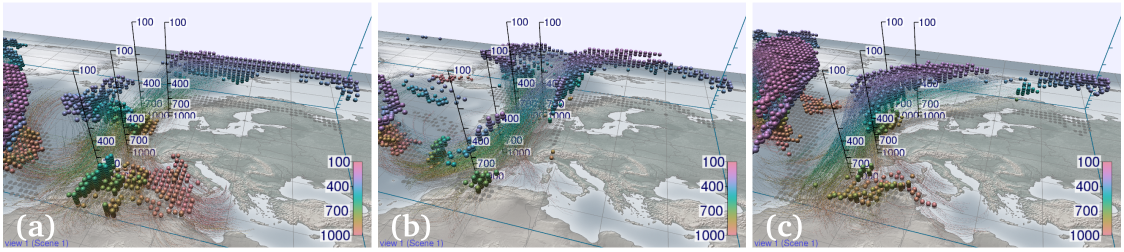


Figure 7.7: The same as Fig. 7.6c, but for the ensemble members **(a)** 2, **(b)** 9 and **(c)** 19. Also compare to the visualizations of the corresponding binary grids *B* shown in Fig. 7.4b–d.

is transported downward. On its leading edge, the tropopause is elevated where it transitions into the anticyclonic region over Central Europe. Between 700 and 500 hPa, increased values of PV indicate regions of diabatic PV production. They coincide with the cold front that can be identified from the strong gradient in equivalent potential temperature (dense contour lines below the clouds in Fig. 7.8b). The cold front tilts westward with height, matching the tilted structure of the p(WCB) isosurface described in the previous section. Ahead (east) of the front, predicted cloud cover largely coincides with the location of the WCB. Overall, the situation resembles the classic conceptual WCB model (Browning, 1986). The WCB outflow predicted over the North Sea at 00Z/20 is related to lower PV values aloft. This is consistent with predicted ice water and cloud cover in this region (not shown).

7.4 Potential flight segments

Given the findings from the previous subsections, we interpret the p(WCB) maximum as the most likely location for the predicted WCB event and draft potential flight segments. Figure 7.9 shows the corresponding Met.3D configuration. For 12Z/19 and 18Z/19, we slide a horizontal section trough

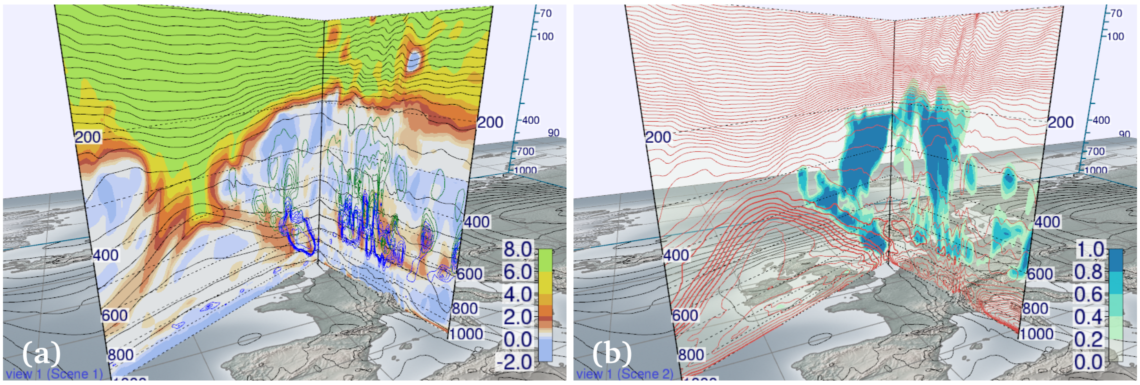
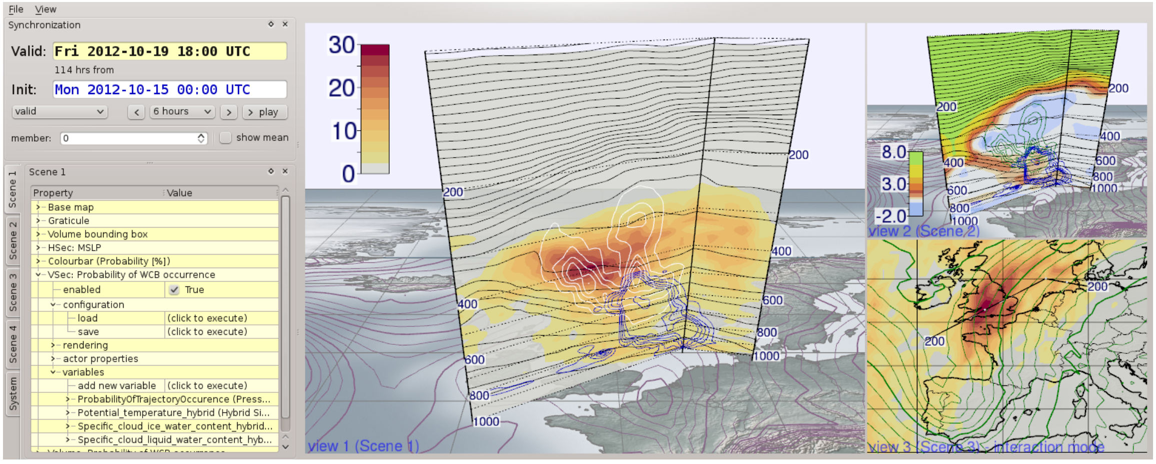


Figure 7.8: **(a)** Vertical section of potential vorticity (colour coding in PVU; red colours in the left plot mark the 2-PVU surface and thus the dynamic tropopause), potential temperature (grey contour lines), liquid and ice water content (blue and green contour lines). **(b)** Vertical section of cloud cover fraction (colour coding) and equivalent potential temperature (red contour lines). Black surface contour lines in both (a) and (b) show sea level pressure. Deterministic forecast from 00:00 UTC 15 October 2012, valid at 18:00 UTC 19 October 2012.


File View

Synchronization

Valid: Fri 2012-10-19 18:00 UTC

114 hrs from

Init: Mon 2012-10-15 00:00 UTC

valid | < | 6 hours | > | > play

member: 0 □ show mean

Scene 1

Property	Value
Base map	
Graticule	
Volume bounding box	
HSec: MSLP	
Colourbar (Probability [%])	
VSec: Probability of WCB occurrence	
enabled	☑ True
configuration	
load	(click to execute)
save	(click to execute)
rendering	
actor properties	
variables	
add new variable	(click to execute)
ProbabilityOfTrajectoryOccurence (Press...	
Potential_temperature_hybrid (Hybrid Si...	
Specific_cloud_ice_water_content_hybrid...	
Specific_cloud_liquid_water_content_hyb...	

Figure 7.9: Planning potential flight legs with Met.3D (ensemble forecast from 00:00 UTC 15 October 2012, valid at 18:00 UTC 19 October 2012). The large view in the middle shows a vertical section of p(WCB) (colour scale in %), potential temperature (black contour lines), liquid and ice water content (blue and white contour lines). Purple surface contours show ensemble mean sea level pressure. The small view on the upper right shows a vertical section of potential vorticity (same colour coding and contour lines as in Fig. 7.8a). The small view on the lower right shows a horizontal section at 390 hPa, showing p(WCB) (same colour scale as in the large view) and contour lines of ensemble mean geopotential height. The maximum p(WCB) along the proposed leg can be found over southern England at around 400 hPa. The vertical section of PV shows how a flight at that altitude, going westward, would penetrate the tropopause shortly after sampling the WCB.

the p(WCB) volume to determine precise locations of the maxima (Video 2 at 05:42 min). At 12Z/19, maximum probabilities are located above the Pyrenees at low levels, in the Bordeaux area between 700 and 600 hPa, and south of Brittany around 400 hPa. Six hours later, the maximum is most prominent above Southern England at altitudes around 400 hPa. A vertical section is used to explore potential flight segments. It allows to estimate at which elevation a flight should take place, and, by moving the section, to quickly assess how spatially relocating the leg will impact the expected measurements. In the given case, the 2D sections suggest flight legs at 12Z/19 over France at elevations between 800 and 600

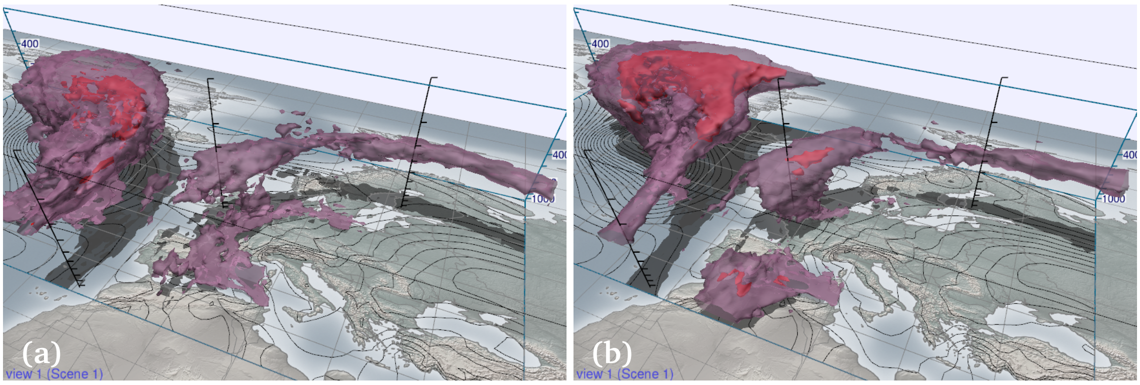


Figure 7.10: Convergence of p(WCB) with decreasing forecast lead time. Forecasts from **(a)** 12:00 UTC 15 October 2012 and **(b)** 12:00 UTC 16 October 2012, valid at 18:00 UTC 19 October 2012. Filter criterion is 500 hPa in 48 h. Isosurfaces show 30 % (red opaque isosurface) and 10 % (purple transparent isosurface). Black surface contours show ensemble mean sea level pressure. The forecast from 00:00 UTC 17 October 2012 is shown in Fig. 6.3c.

hPa (WCB ascent) and at 18Z/19 over Southern England at elevations around 400 hPa (WCB outflow)[3].

However, given the uncertainty in the temporal evolution of the WCB (previous section), we need to carefully monitor developments in subsequent forecast runs. Figure 7.10 (Video 2 at 07:38 min) shows the predictions for 18Z/19 for forecast runs subsequent to the IT00Z/15 run. Over the next two days, the ensemble predictions converge toward higher p(WCB) over the English Channel and Southern England. The elevation of the predicted maximum in p(WCB) remains approximately constant. Indeed, the research flights conducted during TNF showed that the targeted WCB occurred as predicted (Schäfler et al., 2014).

[3] During TNF, we did not have the vertical p(WCB) information available. We placed the flight along a horizontally pre-defined flight leg over France which appeared to fit well with the 2D p(WCB) product. We were only able to guess at which altitudes we should fly. In fact, from the 3D p(WCB) data we find that the flight should rather have been planned south of the pre-defined flight leg.

CHAPTER 8

Summary and conclusions

In this thesis, I investigated the feasibility of designing interactive 3D visualization techniques for numerical ensemble weather prediction data in a way suited to weather forecasting. Particularly motivated by the application case of forecasting during aircraft-based atmospheric field campaigns, I have designed *Met.3D*, a new GPU-based 3D visualization tool targeted at interactive visual exploration of ensemble weather predictions. Focussing on the example of the T-NAWDEX-Falcon 2012 campaign, I proposed and evaluated –as an example of feature-based ensemble visualization in Met.3D– methods to compute and to visually analyse 3D probabilities of the occurrence of warm conveyor belts (large scale airstreams associated with mid-latitude weather systems). By applying the developed techniques to a forecast case that occurred during T-NAWDEX-Falcon, I demonstrated that interactive 3D ensemble visualization is both technically and methodologically feasible to be employed for weather forecasting.

8.1 Summary

The thesis is an interdisciplinary work, combining elements from the areas of visualization and weather prediction. In Chapters 2, 3, and 4 I described the scientific and technological context of my work. Chapter 2 surveyed fundamentals of numerical ensemble weather prediction, explaining the nature of the data and the weather forecasting process. Analogously, Chapter 3 surveyed fundamentals of scientific visualization and recent developments in hardware-accelerated real-time computer graphics. Both fields were combined in Chapter 4. In the light of the introduced foundations, the chapter described the state of the art in meteorological visualization and constituted the basis for the subsequent rationale to develop a new system.

To summarise these chapters: Weather prediction models numerically solve a set of differential equations that describe the temporal evolution of the atmosphere (Sect. 2.1). The equations have chaotic nature, making the simulations prone to small errors in initial conditions and model formulation. The atmosphere cannot be observed without uncertainty and a numerical model cannot be a perfect representation of reality. Hence, there always will be uncertainty in a weather forecast (Sects. 2.2 and 2.3). To characterise this uncertainty, one would ideally forecast a probability distribution of future atmospheric states. This, however, is only possible in approximation. Ensemble forecasts have been established as the primary tool to do so. Based on disturbed initial conditions and model characteristics, a set (at the time of writing on the order of up to 50) of simulations (called ensemble members) are computed that represent possible forecast scenarios. A number of meteorological agencies operate ensemble prediction systems. In this thesis, I focussed on predictions by the European Centre for Medium Range Weather Forecasts. ECMWF operates a global ensemble system consisting of 50 perturbed forecast members as well as one unperturbed control run (Sect. 2.4). In operational weather forecasting, several graphical products are derived from ensemble predictions to aid the meteorologist with the interpretation of the data (Sect. 2.5). Common products include stamp maps (all members plotted side-by-side), the depiction of contour lines in spaghetti plots (all members plotted on top of each other), and maps of ensemble mean, standard deviation, and probabilities. Also, derived products are used, including the clustering of forecasts to identify members with similar flow characteristics.

On the visualization side, the concept of the visualization pipeline abstracts the visualization process by decomposing it into stages (Sect. 3.1). In the conceptual model, data “flows” through stages that acquire data, process data (“filtering”), map the data to geometric representations, and render the geometric representation to an image. The mapping stage is the most characteristic stage of the visualization process and includes algorithms for, for example, isosurface contouring and streamline computation. In the rendering stage, computer graphics techniques are applied. In computer graphics, graphics processing units have become mainstream in the past 15 years (Sect. 3.2). Originally designed as highly specialised co-processors to speed up graphics operations, they have evolved into massively parallel multi-purpose processors. Since the introduction of programmable GPUs about a decade ago, many studies have demonstrated the power of GPU-based visualization algorithms as an alternative to CPU-based mapping of data to geometric primitives. A number of general-purpose visualization systems use the concept of the visualization pipeline as a template for their implementation (Sect. 3.3). In data-flow systems, modules are connected (in some systems by the user in a graphical interface) to form a pipeline that represents the flow of data from the data source to the final image. Such systems emerged in the 1990s and are still in use today.

In meteorology, 3D depictions of atmospheric observation and simulation data have been investigated since the availability of first computer graphics methods in the 1960s and 1970s (Sect. 4.1). The first notable implementation of a 3D visualization tool for the scientific end user appeared around 1990 with the University of Wisconsin-Maddison’s Vis5D software. It was followed by a number of more general-purpose tools that could also be applied to meteorological data. In weather forecasting, 2D visualization methods have been used almost exclusively until today (Sect. 4.2). A few experiments with 3D visualization were conducted by the U.S. and German weather services in the late 1990s and early 2000s, however, suffered from issues including a decoupling of proven 2D methods from new 3D techniques, spatial perception in 3D views, and usability, performance and hardware matters. In meteorological research environments, 2D visualization techniques prevail as well, however, a small number of 3D visualization tools exists. Ensemble visualization methods are mostly limited to research publications from the visualization community, which discuss techniques based on detected features and on statistical quantities computed at a fixed location (Sect. 4.3).

In Chapter 5, I discussed open problems with existing software to achieve the thesis objective of interactively visualizing ensemble predictions in 3D in a way suited to forecasting (Sect. 5.1). Deficiencies still included a straightforward transition from 2D to 3D, 3D perception, rendering performance, as well as the visualization of ensemble products, and, specific to this thesis, techniques to forecast WCBs. I proposed a list of goals for characteristics and functionality of a visualization tool that should be achieved on top of existing approaches to meet the thesis objective. In summary: proven 2D techniques should not be replaced by 3D techniques but be put in a 3D context and augmented by 3D techniques to add value, spatial perception should be improved by adding monocular depth cues, access to ensemble visualizations should be in the centre of the exploration process, common forecasting functionality including time and ensemble navigation must be ensured, continuous-interactive visualization (as opposed to discrete interactivity) is required to facilitate natural exploration and to minimise the risk of missing valuable information, interactive visualization performance should be enabled by usage of modern GPU and multi-thread techniques, the terrain-following vertical coordinate of the ECMWF forecasts should be natively supported for maximum vertical resolution and to avoid preprocessing of the data, and techniques to forecast WCB situations should be available. Three design alternatives were considered, including the extension of existing tools, the design of a tool based on existing visualization libraries, and the development of a new tool. Due to the demand to create an architecture as close as possible to the graphics hardware (to best exploit its performance) and to significant expected modifications and extensions to existing software that would have been required for the first two options, I decided to develop a new tool.

After discussing the rationale for developing a new visualization tool, I presented Met.3D (Sect. 5.2). The tool directly targets the proposed goals. It features a bridge from proven 2D visualization methods commonly used in meteorology to 3D visualization. 2D products (including horizontal and vertical sections and Skew-T-diagrams) are rendered in a 3D context, a product's position can be changed interactively. When 3D elements (including isosurfaces and trajectories) are visualized, spatial perception is improved by displaying shadows on the earth's surface, enabling the user to judge the horizontal position and relative elevation of an element. Quantitative height information can be obtained by means of interactive vertical axes. To visually provide information on forecast uncertainty, Met.3D implements central support for ensemble forecasts. The tool is designed to allow integration of both feature-based and location-based ensemble visualization techniques. In the presented version, forecast products can be animated over the ensemble dimension, and statistical quantities (including ensemble mean, standard deviation, and probabilities) can be derived and visualized on demand. I proposed normal curves, a novel visualization technique to reveal the structure inside a transparent 3D isosurface of a scalar field. With normal curves, the locations and magnitudes of local extrema in the visualized data can be identified at a glance – this is particularly useful to track the temporal evolution of probability fields, in which the absolute maximum values change over time.

Following the architectural approaches introduced in Sect. 3.3, the architecture of Met.3D is built on the pipeline concept. I proposed a pipeline architecture that is highly focussed on support for ensemble and time dimensions, for GPU support, efficient memory management, and for multi-threading task parallelism (Sect. 5.3). The combination of these properties distinguishes the implementation from existing libraries described in the literature. All visualization algorithms are implemented on the GPU and are able to handle the terrain-following hybrid sigma-pressure coordinate of the ECMWF model. Concerning the efficient computation of statistical quantities from forecast data on hybrid sigma-pressure grids, a challenge is that with these coordinates, the vertical positions of the grid points vary between ensemble members. I investigated the difference between regridding all members to a common grid and ignoring this step, and showed that ignoring the variation in grid point pressure between the ensemble members has –in most situations– little impact on the visualization (Sect. 5.4).

In Chapter 6, I proposed methods to compute and to visually analyse 3D probabilities for the occurrence of WCBs, and I integrated both methods into Met.3D. Following the literature, WCBs are detected by means of Lagrangian particle trajectories (Sect. 6.2). By computing trajectories for each member of the ECMWF ensemble forecast, a distribution of WCB features is obtained from which probabilities of occurrence can be derived. I discussed different approaches to trajectory seeding and gridding, and showed that probabilities derived from trajectories computed at a horizontal grid spacing of 1° in latitude and longitude capture the same WCB structures as trajectories computed at a higher grid spacing of 0.25° (Sect. 6.3). A proposed visual analysis method supports the interpretation of the probability fields (Sect. 6.4). The method facilitates fast visual estimation of the number of ensemble members that forecast a WCB feature in a region of interest bounded by a probability isosurface. In particular for situations in which the magnitude of observed probabilities is low, the method helps to distinguish the case in which only few members predict a WCB, but at approximately the same location, from the case in which many members predict a WCB but the spatial variation is high. The method can be applied to probabilities of other features as well.

Finally, in Chapter 7, I presented a case study, revisiting a forecast case that occurred during T-NAWDEX-Falcon. The case study demonstrated how the methods introduced in Chapters 5 and 6 can be used in practice for forecasting.

Summary of contributions

In this thesis, I contributed to the challenging problem of how weather forecast data, in particular ensemble forecast data, can be visualized in 3D in a practical way suited to forecasting. My work extends the studies of Treinish and Rothfusz (1997), Koppert et al. (1998) and McCaslin et al. (2000), and also extends the capabilities of current 3D visualization tools employed in research settings. To summarise my main contributions:

- I developed the new GPU-based 3D visualization tool *Met.3D* that natively supports ensemble forecast data and that puts the interactive visual exploration of the ensemble forecast into the centre of the exploration process. To the best of my knowledge, a tool with such functionality did not exist before.

- To counteract decoupling of proven 2D visualization techniques and 3D techniques, I proposed and implemented a bridge from 2D to 3D visualization, displaying complete 2D forecast products (in the way they appear in the MSS, the "reference" 2D software) in a 3D context, and adding value by allowing them to be interactively moved in real time and by augmenting them with 3D elements. In comparison to existing 3D tools, I improved 3D spatial perception by usage of shadows and vertical axes.

- I proposed a multi-threaded visualization pipeline architecture designed to support the forecast ensemble and initialisation and valid time dimensions, and to provide GPU support, efficient memory management, and CPU-multi-treading. The combination of these properties distinguishes the implementation from existing implementations described in the literature. In Met.3D, the architecture facilitates efficient usage of available hardware resources.

- I proposed GPU-based implementations of several 2D and 3D visualization algorithms in a way such that they directly support vertical hybrid sigma-pressure coordinates.
- I proposed a new visualization technique that displays 3D *normal curves* that, following the gradient of a scalar field, converge at local maxima of the field. The curves reveal information about the internal structure of an isosurface and allow maxima whose absolute values change over time to be tracked in 3D.
- I analysed the impact of ignoring the variation in vertical grid point position when computing ensemble statistical quantities directly from the member's hybrid sigma-pressure model grids, finding that the variation in most cases has little impact on the visualization.
- I proposed a new method, based on Lagrangian ensemble trajectories, to compute 3D probability fields for the occurrence of warm conveyor belts. I analysed the sensitivity of the method with respect to seeding of the trajectories and suggested a setup suited to be used during field campaigns.
- I proposed a new visual analysis method to help the user interpret 3D probability fields. The method determines the contribution of the individual members to a region bound by a probability isosurface. It thereby helps to distinguish the case in which only few members predict a feature but at approximately the same location, from the case in which many members predict a feature but the spatial variation is high.
- I presented a complete forecast case study that demonstrates how interactive 3D ensemble visualization can be used for forecasting.

To facilitate best benefit of my contributions to the scientific community, I made the source code of Met.3D, including all described techniques, available under an open-source license.

8.2 Conclusions

With Met.3D and the proposed WCB methods, we are now able to analyse ensemble prediction data in a way previously impossible. With respect to the case study and to discussions with my co-authors of Rautenhaus et al. (2014, 2015b,a) (of whom Andreas Schäfler, Christian Grams and myself have actively been involved in forecasting during aircraft-based field campaigns), I note a few conclusions from my work, reflecting my opinion.

1. Combination of 2D and 3D visualization methods gives a more complete picture of the forecast atmosphere. 3D elements can depict different aspects of the data than horizontal and vertical 2D sections alone. For example, usage of isosurfaces and normal curves allows for very fast initial judgement of the predicted WCB situation. However, I would not want to abandon the familiar 2D sections; for many tasks (obtaining quantitative information, visualizing multiple forecast parameters in the same plot, analysing the vertical structure of the atmosphere along a flight segment)

they are superior to 3D methods. If 3D visualization is used, achieving good spatial perception is important.

Furthermore, while I think that 3D visualization helps to understand the atmospheric situation in many cases, it does not work equally well for all forecast variables. For the isosurfaces of wind speed and WCB probability used in the case study, 3D visualization is well suited. For variables that highly fluctuate in space (as is often the case for variables depending on moisture, such as relative humidity), isosurfaces are problematic. For these cases, additional methods that help the user focus on the regions and features of interest will need to be developed.

2. One of the primary advantages of Met.3D is the high pace at which a forecast can be explored. Continuous interactivity, the possibility for the user to change a parameter that affects the visualization and to receive immediate visual feedback, is key to this property. It facilitates the very fast analysis of static scenes (moving the camera to explore spatial structure of a feature, moving a vertical axis), of dynamic processes (animation over time), of uncertainty (animation over the ensemble, comparison of different forecast base times), and of sensitivity (changing a parameter that affects a displayed statistical quantity).

   However, I find that while interactivity enables the user to quickly visualize a large amount of data, the user is also confronted with many more images than he would be if he were restricted to, for example, a limited number of horizontal sections. Here, as Trafton and Hoffman (2007) suggest, a virtual “sketchpad” that captures elements discovered by the forecaster and that allows him to represent his “mental model” of the atmosphere would be useful. The sketchpad could also be used to communicate the findings to colleagues, a common challenge during campaigns.

3. The presented methodology to predict probabilities of WCB occurrence illustrates challenges of feature-based approaches to analyse ensemble data. The region contribution approach helps to interpret the derived probabilities, however, further work will be useful. For example, I would like to automatically obtain information about how features in different members correspond to each other: Do other members predict the same situation but shifted in space or in time? Such information would allow to identify different scenarios forecast by the ensemble, and uncertainty could be differentiated with respect to space and time. Detection and visualization of further 3D cyclonic features would also be very useful (following, for instance, the 2D approach by Hewson and Titley, 2010, cf. Fig. 2.18). For a single member the user could see, for example, at a glance which WCB transports which airmass along which route, driven by which cyclone and in relation to which front and jet stream. How to meaningfully visualize such features for an entire ensemble to depict their uncertainty is an open research question.

4. A drawback of the Met.3D visualization approach is that since it uses the complete ensemble dataset, interactive usage requires the forecast data to be available on the local hard drive. For field campaigns based at remote locations, this is not feasible. In these cases, web based approaches such as the DLR Mission Support System (Rautenhaus et al., 2012) might be the better choice. Alternatively, dedicated ensemble compression schemes might enable more efficient remote handling (for example, based on wavelet approaches as suggested by Clyne, 2012; Treib et al., 2012), or remote visualization solutions such as VirtualGL[1] could be used to locate data

[1] http://www.virtualgl.org

and visualization system at the same site while allowing users to explore the data remotely using a modest internet connection.

5. In this thesis I focussed on the application case of weather forecasting during aircraft-based field campaigns. However, I see Met.3D applicable to a wider range of applications, including the analysis of ensemble simulation output in atmospheric research and the usage of the tool to support teaching in meteorology classes. In particular with respect to the latter, I agree with Szoke et al. (2003) that it is important to make new students of meteorology as soon as possible familiar with the power that interactive 3D visualization provides.

6. I also see much potential for further research in meteorological visualization. With respect to 3D visualization, further improvement of spatial perception is very important. In the Met.3D version presented here, shadows are only rendered on the earth's surface. Global illumination techniques (e.g. Jönsson et al., 2014) that, for example, allow 3D elements to mutually cast shadows on each other, may further improve the user's judgement of spatial relationships. Also, the impact of different projections on perceived spatial distance needs to be studied – Met.3D currently is restricted to a cylindrical map projection in the horizontal. Additional challenges include the efficient rendering from further native model grid topologies and real-time placement of text labels to convey quantitative information. The latter applies in particular to 2D and 3D contour lines and surfaces. Due to the employed GPU implementation of the 2D marching squares contouring algorithm, continuous line geometry is not easily available. Hence, it is challenging to compute positions for labels.

7. With respect to ensemble and uncertainty visualisation, open questions are abundant, as reflected by the literature surveyed in Sect. 4.3. I have introduced a feature-based approach for WCBs. This approach on trajectory-based ensemble analysis can be further extended. For example, trajectories can be applied to detect further Lagrangian features. Different selection criteria can reveal airmasses that have undergone specific physical or chemical processes, or that originate in specific geographic regions. In this respect, more complex selection algorithms and the visualization of combined probabilities of multiple features will be challenging. I suggest that further interactive ensemble visualization approaches, both feature-based and location-based, be implemented in Met.3D to study their feasibility and applicability in meteorology.

8. I would have liked to conduct formal evaluations of the presented visualization techniques to study their impact on tasks performed by meteorologists and atmospheric researchers in their daily work. In this respect, I suggest to conduct user studies in the line of Hoffman and Coffey (2004) and Trafton and Hoffman (2007) – with Met.3D, the software necessary to conduct such studies is now available. Furthermore, it is intended that Met.3D will be actively used during upcoming field campaigns, including a future NAWDEX campaign scheduled for 2016 (it will again target WCBs). I suggest to use these campaigns for evaluations.

With the development of Met.3D, I have demonstrated how I envision 3D and ensemble techniques to become a part of standard meteorological visualization. The tool provides a solid software infrastructure that opens the door to investigate the above listed and other research questions, thus enabling the further advancement of meteorological visualization. Considering the ever increasing data volume generated by ensemble weather prediction systems, effective and intuitive visualization methods are and will be important to weather forecasting. The atmosphere is three-dimensional, and while we need to conduct user studies to formally prove the added value through 3D visualization, in my opinion forecast analysis can be made much more intuitive by using interactive 3D methods, thus decreasing the time a meteorologist needs to analyse a forecast dataset.

Bibliography

Abram, G. and Treinish, L.: An Extended Data-Flow Architecture for Data Analysis and Visualization, in: Proceedings of the 6th Conference on Visualization '95, VIS '95, IEEE Computer Society, Washington, DC, USA, 1995. 63, 70, 73

Achtor, T., Rink, T., Whittaker, T., Parker, D., and Santek, D.: McIDAS-V: a powerful data analysis and visualization tool for multi and hyperspectral environmental satellite data, Proceedings SPIE, 7085, 2008. 79

Akenine-Möller, T., Haines, E., and Hoffman, N.: Real-Time Rendering, AK Peters, 3rd edn., 2008. 42, 51, 52, 53, 55, 119

Akima, H.: A method of bivariate interpolation and smooth surface fitting based on local procedures, Commun. ACM, 17, 18–20, 1974. 45

Akima, H.: Algorithm 760: Rectangular-grid-data surface fitting that has the accuracy of a bicubic polynomial, ACM Trans. Math. Softw., 22, 357–361, 1996. 45

Alpert, J. C.: 3-dimensional animated displays for sifting out medium range weather events, in: 19th Conference on IIPS, 9-13 February 2003, Long Beach, California, 2003. 75

Ament, M., Frey, S., Müller, C., Grottel, S., Ertl, T., and Weiskopf, D.: GPU-Accelerated Visualization, in: High Performance Visualization, edited by Bethel, E. W., Childs, H., and Hansen, C., chap. 11, pp. 223–259, CRC Press, 2012. 55, 58, 59

Andersson, E. and Thépaut, J. N.: ECMWF's 4D-Var data assimilation system – the genesis and ten years in operations, ECMWF Newsletter, 115, 8–12, 2008. 22

Angel, E. and Shreiner, D.: Interactive Computer Graphics: A Top-Down Approach with Shader-Based OpenGL, Addison-Wesley, 6th edn., 2011. 42

Arthur, D. K., Lasher-Trapp, S., Abdel-Haleem, A., Klosterman, N., and Ebert, D. S.: A New Three-Dimensional Visualization System for Combining Aircraft and Radar Data and Its Application to RICO Observations, J. Atmos. Oceanic Technol., 27, 811–828, 2010. 82

Atger, F.: Tubing: An Alternative to Clustering for the Classification of Ensemble Forecasts, Wea. Forecasting, 14, 741–757, 1999. 31

Avila, L., Chaudhary, A., and Jhaveri, S.: VTKVolume Rendering Improvements in VTK, Kitware Source, 31, 12–15, 2014. 62

Ayachit, U., Geveci, B., Moreland, K., Patchett, J., and Ahrens, J.: The ParaView Visualization Application, in: High Performance Visualization, edited by Bethel, E. W., Childs, H., and Hansen, C., chap. 18, pp. 383–400, CRC Press, 2012. 63, 79

Bailey, M.: Using GPU Shaders for Visualization, IEEE Comput. Graphics Appl., 29, 96–100, 2009. 58, 59

Bailey, M.: Using GPU Shaders for Visualization, Part 2, IEEE Comput. Graphics Appl., 31, 67–73, 2011. 58, 59

Bailey, M.: Using GPU Shaders for Visualization, Part 3, IEEE Comput. Graphics Appl., 33, 5–11, 2013. 58, 59

Baldauf, M., Seifert, A., Damrath, U., Stephan, K., Pflüger, U., and Majewski, D.: Das Regionalmodell COSMO-DE, promet, 35, 87–97, 2009. 18

Baldauf, M., Seifert, A., Förstner, J., Majewski, D., Raschendorfer, M., and Reinhardt, T.: Operational Convective-Scale Numerical Weather Prediction with the COSMO Model: Description and Sensitivities, Mon. Wea. Rev., 139, 3887–3905, 2011. 18

Barjenbruch, D. B., Thaler, E., and Szoke, E. J.: Operational Applications of Three Dimensional Air Parcel Trajectories Using AWIPS D3D, in: Proceedings of the 18th IIPS Conference, 13-18 January 2002, Orlando, Florida, 2002. 74

Bavoil, L., Callahan, S. P., Crossno, P. J., Freire, J., Scheidegger, C. E., Silva, C. T., and Vo, H. T.: VisTrails: enabling interactive multiple-view visualizations, in: Visualization, 2005. VIS 05. IEEE, pp. 135–142, IEEE, 2005. 63

Bergholt, L.: Diana - an opensource production and visualisation package, in: Proceedings of the 11th ECMWF Workshop on Meteorological Operational Systems, 12-16 November 2007, Reading, England, pp. 150–152, 2008. 72

Bethel, E. W., Childs, H., and Hansen, C., eds.: High Performance Visualization: Enabling Extreme-Scale Scientific Insight, Chapman and Hall/CRC, Boca Raton, FL, 2012. 42

Biddiscombe, J., Geveci, B., Martin, K., Moreland, K., and Thompson, D.: Time Dependent Processing in a Parallel Pipeline Architecture, IEEE Trans. Visual. Comput. Graphics, 13, 1376–1383, 2007. 61, 89, 103, 104, 107

Blakeslee, R., Hall, J., Goodman, M., Parker, P., Freudinger, L., and He, M.: The Real Time Mission Monitor- A Situational Awareness Tool For Managing Experiment Assets, in: NASA Science and Technology Conference 2007, 19-21 June 2007, College Park, MD, 2007. 3, 80

Blanchette, J. and Summerfield, M.: C++ GUI Programming with Qt 4, Prentice Hall, 2nd edn., 2008. 105, 108

Blinn, J. F.: Models of Light Reflection for Computer Synthesized Pictures, in: Proceedings of the 4th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ’77, pp. 192–198, ACM, New York, NY, USA, 1977. 52

Bloomenthal, J.: Calculation of Reference Frames Along a Space Curve, in: Graphics Gems, edited by Glassner, A. S., pp. 567–571, Academic Press Professional, Inc., San Diego, CA, USA, 1990. 47

Blower, J. D., Gemmell, A. L., Griffiths, G. H., Haines, K., Santokhee, A., and Yang, X.: A Web Map Service implementation for the visualization of multidimensional gridded environmental data, 47, 218–224, 2013. 75

Bonneau, G.-P., Hege, H.-C., Johnson, C., Oliveira, M., Potter, K., Rheingans, P., and Schultz, T.: Overview and State-of-the-Art of Uncertainty Visualization, in: Scientific Visualization, edited by Hansen, C. D., Chen, M., Johnson, C. R., Kaufman, A. E., and Hagen, H., Mathematics and Visualization, pp. 3–27, Springer London, 2014. 83

Boreskov, A. and Shikin, E.: Computer Graphics: From Pixels to Programmable Graphics Hardware, CRC Press, 2013. 42

Böttinger, M., Gülzow, V., and Biercamp, J.: Visualisierung als Werkzeug zur Analyse von Klimasimulationsdaten, in: Umweltinformatik '98: Vernetzte Strukturen in Informatik, Umwelt und Wirtschaft, edited by Haasis, H. D. and Ranze, K. C., 1998. 69, 70, 71, 80

Böttinger, M., Pohlmann, H., Röber, N., Meier-Fleischer, K., and Spickermann, D.: Visualization of 2D uncertainty in decadal climate predictions, in: Proc. Workshop on Visualization in Environmental Sciences (EnvirVis 2015), 2015. 83

Branski, F.: Pioneering the collection and exchange of meteorological data, World Meteorological Organization (WMO) Bulletin, 59, 12, 2010. 20

Brodtkorb, A. R., Hagen, T. R., and Sætra, M. L.: Graphics processing unit (GPU) programming strategies and trends in GPU computing, J. Parallel Distr. Com., 73, 4–13, 2013. 58

Brooks, C. E. P., Durst, C. S., and Carruthers, N.: Upper winds over the world: Part I. The frequency distribution of winds at a point in the free air, Q.J.R. Meteorol. Soc., 72, 55–73, 1946. 34

Brown, R.: Animated Visual Vibrations As an Uncertainty Visualisation Technique, in: Proceedings of the 2nd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia, GRAPHITE '04, pp. 84–89, ACM, New York, NY, USA, 2004. 84

Browning, K. A.: Radar measurements of air motion near fronts, Weather, 26, 320–340, 1971. 4

Browning, K. A.: Conceptual Models of Precipitation Systems, Wea. Forecasting, 1, 23–41, 1986. 5, 127, 148

Browning, K. A.: Organization of clouds and precipitation in extratropical cyclones, in: Extratropical Cyclones: The Erik Palmén Memorial Volume, edited by Newton, C. W. and Holopainen, E. O., pp. 129–153, Amer. Meteor. Soc., 1990. 5

Browning, K. A. and Roberts, N. M.: Structure of a frontal cyclone, Q.J.R. Meteorol. Soc., 120, 1535–1557, 1994. 2, 128

Buizza, R.: Localization of optimal perturbations using a projection operator, Q.J.R. Meteorol. Soc., 120, 1647–1681, 1994. 26

Buizza, R.: The ECMWF Ensemble Prediction System, in: Predictability of Weather and Climate, edited by Palmer, T. and Hagedorn, R., chap. 17, pp. 459–488, Cambridge University Press, Cambridge, 2006. 28, 29

Buizza, R. and Palmer, T. N.: The Singular-Vector Structure of the Atmospheric Global Circulation, J. Atmos. Sci., 52, 1434–1456, 1995. 26

Buizza, R., Houtekamer, P. L., Pellerin, G., Toth, Z., Zhu, Y., and Wei, M.: A Comparison of the ECMWF, MSC, and NCEP Global Ensemble Prediction Systems, Mon. Wea. Rev., 133, 1076–1097, 2005. 28

Buizza, R., Bidlot, J.-R., Wedi, N., Fuentes, M., Hamrud, M., Holt, G., Palmer, T., and Vitart, F.: The ECMWF Variable Resolution Ensemble Prediction System VAREPS, ECMWF Newsletter, 108, 14–20, 2006. 2

Buizza, R., Leutbecher, M., and Isaksen, L.: Potential use of an ensemble of analyses in the ECMWF Ensemble Prediction System, Q.J.R. Meteorol. Soc., 134, 2051–2066, 2008. 27

Buizza, R., Leutbecher, M., Isaksen, L., and Haseler, J.: Combined use of EDA– and SV–based pertubations in the EPS, ECMWF Newsletter, 123, 22–28, 2010. 26, 27, 29

Bürger, K.: Particle-based flow visualization, Ph.D. thesis, Technical University of Munich, 2010. 48, 114, 115

Butkiewicz, T. and Ware, C.: Multi-touch 3D exploratory analysis of ocean flow models, in: Oceans 2011, pp. 1–10, IEEE, 2011. 82

Cairo, A.: The Functional Art: An introduction to information graphics and visualization, New Riders, 2012. 42

Callahan, S. P., Callahan, J. H., Scheidegger, C. E., and Silva, C. T.: Direct Volume Rendering: A 3D Plotting Technique for Scientific Data, Computing in Science & Engineering, 10, 88–92, 2008. 44, 46

Carlin, A. V.: Meteorological charts in three dimensions, Mon. Wea. Rev., 82, 97–100, 1954. 66

Carlson, T. N.: Airflow Through Midlatitude Cyclones and the Comma Cloud Pattern, Mon. Wea. Rev., 108, 1498–1509, 1980. 4, 127

Chen, P. C.: Portable and desktop 3-D weather data visualisation system in UNIX and Non-UNIX environments, in: Proceedings of the 6th ECMWF Workshop on Meteorological Operational Systems, 17-21 November 1997, Reading, England, pp. 195–201, 1997. 80

Clyne, J.: Progressive Data Access for Regular Grids, in: High Performance Visualization, edited by Bethel, E. W., Childs, H., and Hansen, C., chap. 8, pp. 145–169, CRC Press, 2012. 78, 102, 156

Clyne, J., Mininni, P., Norton, A., and Rast, M.: Interactive desktop analysis of high resolution simulations: application to turbulent plume dynamics and current sheet formation, New J. Phys., 9, 301+, 2007. 77, 78

Cockburn, A., Karlson, A., and Bederson, B. B.: A Review of Overview+Detail, Zooming, and Focus+Context Interfaces, ACM Comput. Surv., 41, 1–31, 2009. 48

Coiffier, J.: Fundamentals of Numerical Weather Prediction, Cambridge University Press, 2012. 11, 15, 17, 19, 22

Cook, R. L. and Torrance, K. E.: A Reflectance Model for Computer Graphics, ACM Trans. Graph., 1, 7–24, 1982. 52

Côté, J., Gravel, S., Méthot, A., Patoine, A., Roch, M., and Staniforth, A.: The Operational CMC-MRB Global Environmental Multiscale (GEM) Model. Part I: Design Considerations and Formulation, Mon. Wea. Rev., 126, 1373–1395, 1998. 27

Courtier, P. and Naughton, M.: A pole problem in the reduced Gaussian grid, Q.J.R. Meteorol. Soc., 120, 1389–1407, 1994. 17

Cox, J., House, D., and Lindell, M.: Visualizing uncertainty in predicted hurricane tracks, Int. J. Uncertainty Quant., 3, 143–156, 2013. 84

Cozzi, P. and Ring, K.: 3D Engine Design for Virtual Globes, A K Peters/CRC Press, 2011. 62, 115, 116

Crutcher, H. L.: On the standard vector-deviation wind rose, J. Meteor., 14, 28–33, 1957. 34

Cuntz, N., Leidl, M., Kolb, A., Rezk-Salama, C., and Böttinger, M.: GPU-based Dynamic Flow Visualization for Climate Research Applications., in: Proc. Simulation and Visualization, 2007. 82

Daabeck, J.: Overview of meteorological workstation development in Europe, in: 21st International Conference on Interactive Information Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology, 9 - 13 January 2005, 2005. 72

Daabeck, J. and Hibbard, B.: Large Operational User of Visualization, SIGGRAPH Comput. Graph., 37, 5–9, 2003. 69

de la Beaujardiere, J., ed.: Web Map Service Implementation Specification, Version 1.1.1, Open Geospatial Consortium Inc., 2002. 75

de la Beaujardiere, J., ed.: OpenGIS Web Map Service Implementation Specification, Version 1.3.0, Open Geospatial Consortium Inc., 2006. 75

DeFanti, T., Brown, M. D., and McCormick, B. H.: Visualization: expanding scientific and engineering research opportunities, Computer, 22, 12–16, 1989. 41

Denhard, M.: Herausforderungen in der Interpretation und Kommunikation von probabilistischen Vorhersagen, promet, 37, 72–78, 2011. 30

Djurcilov, S., Kim, K., Lermusiaux, P., and Pang, A.: Visualizing scalar volumetric data with uncertainty, Comput. Graph., 26, 239–248, 2002. 84

Doleisch, H. and Hauser, H.: Interactive Visual Exploration and Analysis of Multivariate Simulation Data, Computing in Science & Engineering, 14, 70–77, 2012. 80, 82

Doleisch, H., Muigg, P., and Hauser, H.: Interactive Visual Analysis of Hurricane Isabel with SimVis, in: IEEE Visualization 2004 Contest, Austin, Texas, 2004. 80, 82

Doyle, J. D. and Reynolds, C. A.: Implications of Regime Transitions for Mountain-Wave-Breaking Predictability, Mon. Wea. Rev., 136, 5211–5223, 2008. 83

Ducrocq, V., Braud, I., Davolio, S., Ferretti, R., Flamant, C., Jansa, A., Kalthoff, N., Richard, E., Taupier-Letage, I., Ayral, P.-A., Belamari, S., Berne, A., Borga, M., Boudevillain, B., Bock, O., Boichard, J.-L., Bouin, M.-N., Bousquet, O., Bouvier, C., Chiggiato, J., Cimini, D., Corsmeier, U., Coppola, L., Cocquerez, P., Defer, E., Delanoë, J., Di Girolamo, P., Doerenbecher, A., Drobinski, P., Dufournet, Y., Fourrié, N., Gourley, J. J., Labatut, L., Lambert, D., Le Coz, J., Marzano, F. S., Molinié, G., Montani, A., Nord, G., Nuret, M., Ramage, K., Rison, W., Roussot, O., Said, F., Schwarzenboeck, A., Testor, P., Van Baelen, J., Vincendon, B., Aran, M., and Tamayo, J.: HyMeX-SOP1: The Field Campaign Dedicated to Heavy Precipitation and Flash Flooding in the Northwestern Mediterranean, Bull. Amer. Meteor. Soc., 95, 1083–1100, 2014. 3

Dutton, J. A.: The Ceaseless Wind: An Introduction to the Theory of Atmospheric Motion, McGraw Hill, 1976. 12, 14

Dyer, D. S.: A dataflow toolkit for visualization, IEEE Comput. Graphics Appl., 10, 60–69, 1990. 59, 63

Dyer, J. and Amburn, P.: Desktop Visualization of Meteorological Data Using Paraview, Kitware Source, 14, 7–10, 2010. 79

Eberly, D. H.: 3D Game Engine Architecture: Engineering Real-Time Applications with Wild Magic, CRC Press, 2005. 62

Eckhardt, S., Stohl, A., Wernli, H., James, P., Forster, C., and Spichtinger, N.: A 15-Year Climatology of Warm Conveyor Belts, J. Climate, 17, 218–237, 2004. 2, 5, 128

ECMWF: IFS documentation – Cy40r1, Part II: Data assimilation, Tech. rep., European Centre for Medium-Range Weather Forecasting, Reading, UK, 2013a. 22

ECMWF: IFS documentation – Cy40r1, Part III: Dynamics and numerical procedures, Tech. rep., European Centre for Medium-Range Weather Forecasting, Reading, UK, 2013b. 18

Edmunds, M., Laramee, R. S., Chen, G., Max, N., Zhang, E., and Ware, C.: Surface-based flow visualization, Computers & Graphics, 36, 974–990, 2012. 47

Ehrendorfer, M.: Predicting the uncertainty of numerical weather forecasts: a review, Meteorol. Zeitschrift, 6, 147–183, 1997. 25

Ehrendorfer, M.: The Liouville equation and atmospheric predictability, in: Predictability of Weather and Climate, edited by Palmer, T. and Hagedorn, R., chap. 4, pp. 59–98, Cambridge University Press, Cambridge, 2006. 25

Eisemann, E., Schwarz, M., Assarsson, U., and Wimmer, M.: Real-Time Shadows, A K Peters/CRC Press, 2012. 53

Elsberry, R. L. and Harr, P. A.: Tropical cyclone structure (TCS08) field experiment science basis, observational platforms, and strategy, Asia-Pacific J. Atmos. Sci., 44, 209–231, 2008. 3

Engel, K., Hadwiger, M., Kniss, J., Rezk-Salama, C., and Weiskopf, D.: Real-Time Volume Graphics, A K Peters, Wellesley, Mass., 2006. 46, 47, 58, 59, 114

Epstein, E. S.: Stochastic dynamic prediction, Tellus, 21, 739–759, 1969. 25

Epstein, E. S. and Fleming, R. J.: Depicting Stochastic Dynamic Forecasts, J. Atmos. Sci., 28, 500–511, 1971. 1

Etling, D.: Theoretische Meteorologie: Eine Einführung, Springer, Berlin, 2nd edn., 2002. 12, 13, 14

Eymann, G.: Web Development Activities by Central Development of DWD (FEZE), in: ECMWF Web Developers' Meeting 2010, Reading, UK, 2010. 75

Faconti, G. P. and Massink, M.: Continuous interaction with computers: issues and requirements., in: Proc. of HCI International 2001 (the 9th International Conference on Human-Computer Interaction), 5-10 August 2001, New Orleans, USA, vol. 3, 2001. 87

Fatahalian, K. and Houston, M.: A closer look at GPUs, Commun. ACM, 51, 50–57, 2008. 54, 55, 57

Favre, J. M. and Valle, M.: AVS and AVS/Express, in: The Visualization Handbook, edited by Hansen, C. D. and Johnson, C., chap. 33, pp. 655–672, Academic Press, 2005. 63, 70

Ferranti, L. and Corti, S.: New clustering products, ECMWF Newsletter, 127, 6–11, 2011. 31, 32

Ferranti, L., Corti, S., and Janousek, M.: Flow-dependent verification of the ECMWF ensemble over the Euro-Atlantic sector, Q.J.R. Meteorol. Soc., 2014. 30, 32

Fjukstad, B., Hagen, T.-M. S., Stødle, D., Ha, P. H., Bjørndalen, J. M., and Anshus, O.: Interactive Weather Simulation and Visualization on a Display Wallwith Many-Core Compute Nodes, in: Para 2010: State of the Art in Scientific and Parallel Computing, 6-9 June 2010, Reykjavík,, 2010. 82

Flatøy, F., Hov, Ø., and Schlager, H.: Chemical forecasts used for measurement flight planning during POLINAT 2, Geophys. Res. Lett., 27, 951–958, 2000. 3, 80

Foulser, D.: IRIS Explorer: A Framework for Investigation, ACM SIGGRAPH Comput. Graph., 29, 13–16, 1995. 63

Fraedrich, R.: Interactive VisualizationTechniques for Large-ScaleParticle Simulations, Ph.D. thesis, Technical University of Munich, 2012. 59, 102

Freire, J., Koop, D., Santos, E., Scheidegger, C., Silva, C., and Vo, H. T.: VisTrails, in: The Architecture Of Open Source Applications, edited by Brown, A. and Wilson, G., pp. 367–384, lulu.com, 2011. 63, 64

Fritsch, D. and Kada, M.: Visualisation Using Game Engines, in: Proceedings of the XXth Congress of the ISPRS, vol. 35, pp. 621–625, 2004. 62

Froude, L. S. R., Bengtsson, L., and Hodges, K. I.: The Prediction of Extratropical Storm Tracks by the ECMWF and NCEP Ensemble Prediction Systems, Mon. Wea. Rev., 135, 2545–2567, 2007. 35

Froude, L. S. R., Bengtsson, L., and Hodges, K. I.: Atmospheric predictability revisited, Tellus A, 65, 2013. 24, 30

Fuhrmann, A. and Gröller, E.: Real-time techniques for 3D flow visualization, in: Visualization ’98. Proceedings, pp. 305–312, IEEE Computer Society Press Los Alamitos, CA, USA, 1998. 47

Gagnon, N., Deng, X. X., Houtekamer, P. L., Beauregard, S., Erfani, A., Charron, M., Lahlou, R., and Marcoux, J.: Improvements to the Global Ensemble Prediction System (GEPS) from version 3.1.0 to version 4.0.0, Tech. rep., Canadian Meteorological Center of Environment Canada, Dorval, Québec, Canada, 2014. 27, 28

Gallus, W. A., Yarger, D. N., Cruz-Neira, C., and Heer, R.: An Example of a Virtual Reality Learning Environment, Bull. Amer. Meteor. Soc., 84, 18–20, 2003. 82

Gallus, W. A., Cervato, C., Cruz-Neira, C., Faidley, G., and Heer, R.: Learning Storm Dynamics with a Virtual Thunderstorm, Bull. Amer. Meteor. Soc., 86, 162–163, 2005. 82

Gehring, D.: Interaktive Lagrangesche Partikeltrajektorien für das 3D-Ensemble-Wettervorhersagetool Met.3D, Bachelor’s thesis, Technical University of Munich, 2015. 97

Gelberg, L. M. and Stephenson, T. P.: Supercomputing and Graphics in the Earth and Planetary Sciences, IEEE Comput. Graphics Appl., 7, 26–33, 1987. 69

Geveci, B. and Schroeder, W.: VTK, in: The Architecture Of Open Source Applications, edited by Brown, A. and Wilson, G., pp. 385–400, lulu.com, 2011. 60, 61, 107

Gill, P. G. and Buchanan, P.: An ensemble based turbulence forecasting system, Met. Apps, 21, 12–19, 2014. 35

Gleeson, T. A.: Probability Predictions of Geostrophic Winds, J. Appl. Meteor., 6, 355–359, 1967. 1

Gneiting, T. and Raftery, A. E.: Weather Forecasting with Ensemble Methods, Science, 310, 248–249, 2005. 1, 23, 25, 35

Grams, C. M., Wernli, H., Böttcher, M., Čampa, J., Corsmeier, U., Jones, S. C., Keller, J. H., Lenz, C.-J., and Wiegand, L.: The key role of diabatic processes in modifying the upper-tropospheric wave guide: a North Atlantic case-study, Q.J.R. Meteorol. Soc., 137, 2174–2193, 2011. 128

Griffith, E. J., Post, F. H., Koutek, M., Heus, T., and Jonker, H. J. J.: Feature Tracking in VR for Cumulus Cloud Life-cycle Studies, in: Proceedings of the 11th Eurographics Conference on Virtual Environments, EGVE’05, pp. 121–128, Eurographics Association, Aire-la-Ville, Switzerland, 2005. 82

Grigoryan, G. and Rheingans, P.: Point-based probabilistic surfaces to show surface uncertainty, IEEE Trans. Visual. Comput. Graphics, 10, 564–573, 2004. 84

Grotjahn, R. and Chervin, R. M.: Animated Graphics in Meteorological Research and Presentations, Bull. Amer. Meteor. Soc., 65, 1201–1208, 1984. 66

Guha, S.: Computer Graphics Through OpenGL: From Theory to Experiments, CRC Press, 2nd edn., 2015. 42

Günther, T., Schulze, M., Friederici, A., and Theisel, H.: 2014 IEEE Scientific Visualization Contest Winner: Visualizing Volcanic Clouds in the Atmosphere and their Impact on Air Traffic, IEEE Comput. Graphics Appl., to appear, 2015. 83

Haase, H., Bock, M., Hergenröther, E., Knöpfte, C., Koppert, H. J., Schröder, F., Trembilski, A., and Weidenhausen, J.: Where Weather Meets the Eye – A Case Study on a Wide Range of Meteorological Visualisations for Diverse Audiences, in: Data Visualization '99, edited by Gröller, E., Löffelmann, H., and Ribarsky, W., Eurographics, pp. 261–266, Springer Vienna, 1999. 74

Haase, H., Bock, M., Hergenröther, E., Knöpfle, C., Koppert, H. J., Schröder, F., Trembilski, A., and Weidenhausen, J.: Meteorology meets computer graphics - a look at a wide range of weather visualisations for diverse audiences, Computers & Graphics, 24, 391–397, 2000. 74

Haeberli, P. E.: ConMan: A Visual Programming Language for Interactive Graphics, in: Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '88, pp. 103–111, ACM, New York, NY, USA, 1988. 59

Haiden, T., Janousek, M., Bauer, P., Bidlot, J., Ferranti, L., Hewson, T., Prates, F., Richardson, D. S., and Vitart, F.: Evaluation of ECMWF forecasts, including 2013-2014 upgrades, Tech. Rep. 742, European Centre for Medium-Range Weather Forecasting, Reading, UK, 2014. 30

Haltiner, G. J. and Williams, R. T.: Numerical Prediction and Dynamic Meteorology, Wiley, 2nd edn., 1980. 11

Hamill, T. M., Brennan, M. J., Brown, B., DeMaria, M., Rappaport, E. N., and Toth, Z.: NOAA's Future Ensemble-Based Hurricane Forecast Products, Bull. Amer. Meteor. Soc., 93, 209–220, 2012. 35

Hansen, C. D. and Johnson, C.: The Visualization Handbook, Academic Press, 2005. 42, 44, 47, 48

Hanwell, M.: New OpenGL Rendering in VTK, Kitware Source, 30, 4–5, 2014. 62

Harris, M. J. and Lastra, A.: Real-Time Cloud Rendering, Comput. Graphics Forum, 20, 76–85, 2001. 82

Harris, M. J., Baxter, W. V., Scheuermann, T., and Lastra, A.: Simulation of Cloud Dynamics on Graphics Hardware, in: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware, HWWS '03, pp. 92–101, Eurographics Association, Aire-la-Ville, Switzerland, 2003. 82

Harrold, T. W.: Mechanisms influencing the distribution of precipitation within baroclinic disturbances, Q.J.R. Meteorol. Soc., 99, 232–251, 1973. 4, 127

Hasler, A. F.: Stereographic Observations from Geosynchronous Satellites: An Important New Tool for the Atmospheric Sciences, Bull. Amer. Meteor. Soc., 62, 194–212, 1981. 67

Hasler, A. F., des Jardins, M., and Negri, A. J.: Artificial Stereo Presentation of Meteorological Data Fields, Bull. Amer. Meteor. Soc., 62, 970–973, 1981. 67

Hasler, A. F., Pierce, H., Morris, K. R., and Dodge, J.: Meteorological Data Fields "In Perspective", Bull. Amer. Meteor. Soc., 66, 795–801, 1985. 67

He, M., Goodman, H. M., Blakeslee, R., and Hall, J. M.: The Waypoint Planning Tool: Real Time Flight Planning for Airborne Science, in: American Geophysical Union 2010 Fall Meeting, 13-17 December 2010, San Francisco, CA, 2010. 3, 80

Hegarty, M., Canham, M. S., and Fabrikant, S. I.: Thinking about the weather: How display salience and knowledge affect performance in a graphic inference task, J. Exp. Psychol. Learn., 36, 37–53, 2010. 82

Heidelmann, C.: Interactive vertical profiles for ensemble weather predictions in the 3D weather forecasting tool Met.3D, Bachelor's thesis, Technical University of Munich, 2015. 98

Heizenrieder, D. and Haucke, S.: Das meteorologische Visualisierungs- und Produktionssystem NinJo, promet, 35, 57–69, 2009. 71

Henderson, A., Ahrens, J., and Law, C.: The ParaView Guide, Kitware Clifton Park, NY, 2004. 63, 79

Hergenröther, E., Bleile, A., Middleton, D., and Trembilski, A.: The abalone interpolation: a visual interpolation procedure for the calculation of cloud movement, pp. 381–387, 2002. 82

Heus, T., Jonker, H. J. J., Van den Akker, H. E. A., Griffith, E. J., Koutek, M., and Post, F. H.: A statistical approach to the life cycle analysis of cumulus clouds selected in a virtual reality environment, J. Geophys. Res., 114, D06 208+, 2009. 82

Hewson, T. D.: Objective identification of frontal wave cyclones, Met. Apps, 4, 311–315, 1997. 34

Hewson, T. D.: Objective fronts, Met. Apps, 5, 37–65, 1998. 34

Hewson, T. D.: Tracking fronts and extra-tropical cyclones, ECMWF Newsletter, 121, 9–19, 2009. 34

Hewson, T. D. and Titley, H. A.: Objective identification, typing and tracking of the complete life-cycles of cyclonic features at high spatial resolution, Met. Apps, 17, 355–381, 2010. 34, 40, 84, 156

Hibbard, B.: VisAD: Connecting People to Computations and People to People, SIGGRAPH Comput. Graph., 32, 10–12, 1998. 79

Hibbard, B.: The Top Five Problems That Motivated My Work, IEEE Comput. Graphics Appl., 24, 9–13, 2004. 87

Hibbard, B. and Santek, D.: The VIS-5D System for Easy Interactive Visualization, in: Proceedings of the 1st Conference on Visualization '90, VIS '90, pp. 28–35, IEEE Computer Society Press, Los Alamitos, CA, USA, 1990. 69

Hibbard, W. and Santek, D.: Visualizing large data sets in the earth sciences, Computer, 22, 53–57, 1989a. 68

Hibbard, W. and Santek, D.: Interactivity is the Key, in: Proceedings of the 1989 Chapel Hill Workshop on Volume Visualization, VVS '89, pp. 39–43, ACM, New York, NY, USA, 1989b. 68, 69

Hibbard, W., Dyer, C. R., and Paul, B.: Display of scientific data structures for algorithm visualization, in: Proceedings of the 3rd conference on Visualization '92, pp. 139–146, IEEE Computer Society Press Los Alamitos, CA, USA, 1992. 79

Hibbard, W. L.: Computer-Generated Imagery for 4-D Meteorological Data, Bull. Amer. Meteor. Soc., 67, 1362–1369, 1986. 67, 68

Hibbard, W. L.: 4-D Display of Meteorological Data, in: Proceedings of the 1986 Workshop on Interactive 3D Graphics, I3D '86, pp. 23–36, ACM, New York, NY, USA, 1987. 67

Hibbard, W. L.: Future directions in 3-D Meteorological display, in: Proceedings of the 6th ECMWF Workshop on Meteorological Operational Systems, 17-21 November 1997, Reading, England, pp. 186–190, 1997. 79

Hibbard, W. L.: Vis5D, Cave5D, and VisAD, in: The Visualization Handbook, edited by Hansen, C. D. and Johnson, C., chap. 34, pp. 673–688, Academic Press, 2005. 69, 70, 82

Hibbard, W. L., Santek, D., Uccellini, L., and Brill, K.: Application of the 4-D McIDAS to a Model Diagnostic Study of the Presidents' Day Cyclone, Bull. Amer. Meteor. Soc., 70, 1394–1403, 1989. 67, 68, 127

Hibbard, W. L., Paul, B. E., Santek, D. A., Dyer, C. R., Battaiola, A. L., and Martinez, M. F. V.: Interactive Visualization of Earth and Space Science Computations, Computer, 27, 65–72, 1994. 69, 79

Hibbard, W. L., Anderson, J., Foster, I., Paul, B. E., Jacob, R., Schafer, C., and Tyree, M. K.: Exploring Coupled Atmosphere-Ocean Models Using Vis5D, Int. J. High Perform. C., 10, 211–222, 1996. 69

Hoffman, R. R.: Human Factors Psychology in the Support of Forecasting: The Design of Advanced Meteorological Workstations, Wea. Forecasting, 6, 98–110, 1991. 50

Hoffman, R. R. and Coffey, J. W.: Weather Forecasting and the Principles of Complex Cognitive Systems, in: Proceedings of the Human Factors and Ergonomics Society Annual Meeting, vol. 48, pp. 315–319, SAGE Publications, 2004. 1, 157

Hoffman, R. R., Detweiler, M., Conway, J. A., and Lipton, K.: Some Considerations in Using Color in Meteorological Displays, Wea. Forecasting, 8, 505–518, 1993. 50, 94

Höllt, T., Magdy, A., Zhan, P., Chen, G., Gopalakrishnan, G., Hoteit, I., Hansen, C. D., and Hadwiger, M.: Ovis: A Framework for Visual Analysis of Ocean Forecast Ensembles, IEEE Trans. Visual. Comput. Graphics, 20, 1114–1126, 2014. 84

Holton, J. R.: An Introduction to Dynamic Meteorology (International Geophysics), Academic Press, 4th edn., 2004. 12, 13, 15

Hortal, M. and Simmons, A. J.: Use of Reduced Gaussian Grids in Spectral Models, Mon. Wea. Rev., 119, 1057–1074, 1991. 17

Houtekamer, P. L. and Mitchell, H. L.: Ensemble Kalman filtering, Q.J.R. Meteorol. Soc., 131, 3269–3289, 2005. 26

Houtekamer, P. L., Deng, X., Mitchell, H. L., Baek, S.-J., and Gagnon, N.: Higher Resolution in an Operational Ensemble Kalman Filter, Mon. Wea. Rev., 142, 1143–1162, 2014. 28

Hughes, J. F., van Dam, A., McGuire, M., Sklar, D. F., Foley, J. D., Feiner, S. K., and Akeley, K.: Computer Graphics: Principles and Practice, Addison-Wesley Professional, 3rd edn., 2013. 42

Hunter, J. D.: Matplotlib: A 2D Graphics Environment, Comput. Sci. Eng., 9, 90–95, 2007. 76

Ihász, I. and Tajti, D.: Use of ECMWF's ensemble vertical profiles at the Hungarian Meteorological Service, ECMWF Newsletter, 129, 25–29, 2011. 98, 101

Inness, P. M. and Dorling, S.: Operational Weather Forecasting, Wiley-Blackwell, 2013. 11, 13, 14, 15, 19, 20, 22, 23, 26, 30, 32

Isaksen, L., Haseler, J., Buizza, R., and Leutbecher, M.: The new Ensemble of Data Assimilations, ECMWF Newsletter, 123, 17–21, 2010. 27, 29

Jacobson, M. Z.: Fundamentals of Atmospheric Modeling, Cambridge University Press, 2nd edn., 2005. 11

Jakob, C.: Accelerating Progress in Global Atmospheric Model Development through Improved Parameterizations: Challenges, Opportunities, and Strategies, Bull. Amer. Meteor. Soc., 91, 869–875, 2010. 15

Jobard, B. and Lefer, W.: Creating Evenly-Spaced Streamlines of Arbitrary Density, in: Visualization in Scientific Computing '97, edited by Lefer, W. and Grave, M., Eurographics, pp. 43–55, Springer Vienna, 1997. 81

Jobard, B., Favre, J. M., Mangli, A., Ballabio, M., Consoli, M., Ponti, C., Maric, D., de Morsier, G., Kaufmann, P., Arpagaus, M., Bettems, J. M., Zala, E., and Quiby, J.: Advanced flow representations applied to wind visualization, in: Proceedings of the 8th ECMWF Workshop on Meteorological Operational Systems, 12-16 November 2001, Reading, England, pp. 172–178, 2001. 81

Johnson, C. R. and Sanderson, A. R.: A Next Step: Visualizing Errors and Uncertainty, IEEE Comput. Graphics Appl., 23, 6–10, 2003. 83

Jönsson, D., Sundén, E., Ynnerman, A., and Ropinski, T.: A Survey of Volumetric Illumination Techniques for Interactive Volume Rendering, Comput. Graphics Forum, 33, 27–51, 2014. 119, 157

June, F.: An Introduction to 3D Computer Graphics, Stereoscopic Image, and Animation in OpenGL and C/C++, CreateSpace Independent Publishing Platform, 2nd edn., 2011. 42

Kajiya, J. T. and Von Herzen, B. P.: Ray Tracing Volume Densities, SIGGRAPH Comput. Graph., 18, 165–174, 1984. 69

Kalkusch, M. and Schmalstieg, D.: Extending the scene graph with a dataflow visualization system, in: Proceedings of the ACM symposium on Virtual reality software and technology, VRST '06, pp. 252–260, ACM, New York, NY, USA, 2006. 62

Kasahara, A.: Computer experiments in the global circulation of the earth's atmosphere, in: Proceedings of the December 9-11, 1968, fall joint computer conference, part II, AFIPS '68 (Fall, part II), pp. 1259–1272, ACM, New York, NY, USA, 1968. 66

Kasahara, A. and Washington, W. M.: NCAR global general circulation model of the atmosphere, Mon. Wea. Rev., 95, 389–402, 1967. 66

Kaufman, A. and Mueller, K.: Overview of Volume Rendering, in: The Visualization Handbook, edited by Hansen, C. D. and Johnson, C., chap. 7, pp. 127–174, Academic Press, 2005. 46

Kaufmann, S., Voigt, C., Jeßberger, P., Jurkat, T., Schlager, H., Schwarzenboeck, A., Klingebiel, M., and Thornberry, T.: In situ measurements of ice saturation in young contrails, Geophys. Res. Lett., 41, 702–709, 2014. 95

Kehrer, J. and Hauser, H.: Visualization and Visual Analysis of Multifaceted Scientific Data: A Survey, IEEE Trans. Visual. Comput. Graphics, 19, 495–513, 2013. 48, 82

Kehrer, J., Ladstadter, F., Muigg, P., Doleisch, H., Steiner, A., and Hauser, H.: Hypothesis Generation in Climate Research with Interactive Visual Data Exploration, IEEE Trans. Visual. Comput. Graphics, 14, 1579–1586, 2008. 82

Keil, C., Tafferner, A., and Reinhardt, T.: Synthetic satellite imagery in the Lokal-Modell, Atmospheric Research, 82, 19–25, 2006. 82

Kendall, W., Huang, J., and Peterka, T.: Geometric Quantification of Features in Large Flow Fields, IEEE Comput. Graphics Appl., 32, 46–54, 2012. 82

Kertész, S., Siemen, S., Lamy-Thépaut, S., Ii, F., Russell, I., and Karhila, V.: OGC Clients in Metview 4, in: Third workshop on the use of OGC/GIS standards in meteorology, Exeter, UK, 2010. 75

Kniss, J., Hansen, C., Grenier, M., and Robinson, T.: Volume rendering multivariate data to visualize meteorological simulations: a case study, in: Proceedings of the symposium on Data Visualisation 2002, VISSYM '02, pp. 189–ff, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2002. 82

Koppert, H. J.: A JAVA based-meteorological workstation, in: Proceedings of the 8th ECMWF Workshop on Meteorological Operational Systems, 12-16 November 2001, Reading, England, pp. 87–89, 2001. 71

Koppert, H. J., Schröder, F., Hergenröther, E., Lux, M., and Trembilski, A.: 3D visualisation in daily operation at the DWD, in: Proceedings of the 6th ECMWF Workshop on Meteorological Operational Systems, 17-21 November 1997, Reading, England, pp. 119–142, 1998. 1, 5, 70, 73, 74, 85, 86, 154

Koutek, M.: Scientific Visualization in Virtual Reality: Interaction Techniques and Application Development, Ph.D. thesis, Delft University of Technology, 2003. 82

Krüger, J. and Westermann, R.: Acceleration Techniques for GPU-based Volume Rendering, in: Proceedings of the 14th IEEE Visualization 2003 (VIS'03), 22-24 October 2003, Seattle, Washington, pp. 287–292, IEEE Computer Society, Washington, DC, USA, 2003. 114

Kuo, Y.-H., Reed, R. J., and Low-Nam, S.: Thermal Structure and Airflow in a Model Simulation of an Occluded Marine Cyclone, Mon. Wea. Rev., 120, 2280–2297, 1992. 127

Kwok, Y.-K. and Ahmad, I.: Benchmarking and Comparison of the Task Graph Scheduling Algorithms, J. Parallel Distr. Com., 59, 381–422, 1999. 118

Ladstädter, F., Steiner, A. K., Lackner, B. C., Pirscher, B., Kirchengast, G., Kehrer, J., Hauser, H., Muigg, P., and Doleisch, H.: Exploration of Climate Data Using Interactive Visualization, J. Atmos. Oceanic Technol., 27, 667–679, 2010. 82

Lakshmanan, V., Rabin, R., Otkin, J., Kain, J. S., and Dembek, S.: Visualizing Model Data Using a Fast Approximation of a Radiative Transfer Model, J. Atmos. Oceanic Technol., 29, 745–754, 2012. 82

Lalaurette, F.: Early detection of abnormal weather conditions using a probabilistic extreme forecast index, Q.J.R. Meteorol. Soc., 129, 3037–3057, 2003. 33, 83

Lamy-Thépaut, S., Sahin, C., and Raoult, B.: ecCharts service, ECMWF Newsletter, 134, 7–9, 2013. 75, 83

Laramee, R. S., Hauser, H., Doleisch, H., Vrolijk, B., Post, F. H., and Weiskopf, D.: The State of the Art in Flow Visualization: Dense and Texture-Based Techniques, Comput. Graphics Forum, 23, 203–221, 2004. 47

Laramee, R. S., Hauser, H., Zhao, L., and Post, F. H.: Topology-Based Flow Visualization, The State of the Art, in: Topology-based Methods in Visualization, edited by Hauser, H., Hagen, H., and Theisel, H., Mathematics and Visualization, chap. 1, pp. 1–19, Springer, Berlin, Heidelberg, 2007. 47

Lazzara, M. A., Benson, J. M., Fox, R. J., Laitsch, D. J., Rueden, J. P., Santek, D. A., Wade, D. M., Whittaker, T. M., and Young, J. T.: The Man computer Interactive Data Access System: 25 Years of Interactive Processing, Bull. Amer. Meteor. Soc., 80, 271–284, 1999. 67

Lee, T.-Y., Wong, P. C., Hagos, S., Tong, X., Shen, H.-W., and Leung, L. R.: Feature Tracking and Visualization of the Madden-Julian Oscillation in Climate Simulation, IEEE Comput. Graphics Appl., 33, 29–37, 2013. 82

Legg, T. P., Mylne, K. R., and Woolcock, C.: Use of medium-range ensembles at the Met Office I: PREVIN - a system for the production of probabilistic forecast information from the ECMWF EPS, Met. Apps, 9, 255–271, 2002. 30

Leith, C. E.: Theoretical Skill of Monte Carlo Forecasts, Mon. Wea. Rev., 102, 409–418, 1974. 25

Leuenberger, D., Koller, M., Fuhrer, O., and Schär, C.: A Generalization of the SLEVE Vertical Coordinate, Mon. Wea. Rev., 138, 3683–3689, 2010. 18

Leutbecher, M. and Palmer, T.: Ensemble forecasting, J. Comput. Phys., 227, 3515–3539, 2008. 1, 2, 25

Lewis, J. M.: Roots of Ensemble Forecasting, Mon. Wea. Rev., 133, 1865–1885, 2005. 25

Lewis, J. M.: Edward Epstein's Stochastic-Dynamic Approach to Ensemble Weather Prediction, Bull. Amer. Meteor. Soc., 95, 99–116, 2014. 25

Limbach, S., Schömer, E., and Wernli, H.: Detection, tracking and event localization of jet stream features in 4-D atmospheric data, Geosci. Model Dev., 5, 457–470, 2012. 82

Limbach, S., Sprenger, M., Schömer, E., and Wernli, H.: IWAL - an Interactive Weather Analysis Laboratory, Bull. Amer. Meteor. Soc., 96, 903–909, 2014. 75

Lindemann, F. and Ropinski, T.: About the Influence of Illumination Models on Image Comprehension in Direct Volume Rendering, IEEE Trans. Visual. Comput. Graphics, 17, 1922–1931, 2011. 53, 71

Lindholm, E., Kilgard, M. J., and Moreton, H.: A User-programmable Vertex Engine, in: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ’01, pp. 149–158, ACM, New York, NY, USA, 2001. 57

Lindholm, E., Nickolls, J., Oberman, S., and Montrym, J.: NVIDIA Tesla: A Unified Graphics and Computing Architecture, IEEE Micro, 28, 39–55, 2008. 57

Liou, K. N.: An Introduction to Atmospheric Radiation, Academic Press, 2nd edn., 2002. 15, 46, 51

Lorensen, W. E. and Cline, H. E.: Marching Cubes: A High Resolution 3D Surface Construction Algorithm, in: Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, vol. 21 of *SIGGRAPH ’87*, pp. 163–169, ACM, New York, NY, USA, 1987. 46

Lorenz, E. N.: Deterministic Nonperiodic Flow, J. Atmos. Sci., 20, 130–141, 1963. 24

Lorenz, E. N.: A study of the predictability of a 28-variable atmospheric model, Tellus, 17, 321–333, 1965. 24

Lorenz, E. N.: Atmospheric predictability experiments with a large numerical model, Tellus, 34, 505–513, 1982. 24

Lorenz, E. N.: The Essence Of Chaos, University of Washington Press, 1993. 24

Lucas, B., Abram, G. D., Collins, N. S., Epstein, D., Gresh, D. L., and McAuliffe, K. P.: An architecture for a scientific visualization system, in: Proceedings of the 3rd conference on Visualization ’92, pp. 107–114, IEEE Computer Society Press Los Alamitos, CA, USA, 1992. 59

Lundstrom, C., Ljung, P., Persson, A., and Ynnerman, A.: Uncertainty Visualization in Medical Volume Rendering Using Probabilistic Animation, IEEE Trans. Visual. Comput. Graphics, 13, 1648–1655, 2007. 84

Lux, M. and Frühauf, T.: A visualization system for operational meteorological use, in: Proceedings of the Sixth International Conference in Central Europe on Computer Graphics and Visualization (WSCG’98), 9-13 February 1998, Plzen, Czech Republic, pp. 525–534, 1998. 73, 74

MacEachren, A. M., Robinson, A., Hopper, S., Gardner, S., Murray, R., Gahegan, M., and Hetzler, E.: Visualizing Geospatial Information Uncertainty: What We Know and What We Need to Know, Cartography and Geographic Information Science, 32, 139–160, 2005. 83

Madonna, E., Wernli, H., Joos, H., and Martius, O.: Warm Conveyor Belts in the ERA-Interim Dataset (1979–2010). Part I: Climatology and Potential Vorticity Evolution, J. Climate, 27, 3–26, 2014. 5, 128, 129

Maronga, B., Gryschka, M., Heinze, R., Hoffmann, F., Kanani-Sühring, F., Keck, M., Ketelsen, K., Letzel, M. O., Sühring, M., and Raasch, S.: The Parallelized Large-Eddy Simulation Model (PALM) version 4.0 for atmospheric and oceanic flows: model formulation, recent developments, and future perspectives, Geosci. Model Dev., 8, 2515–2551, 2015. 19

Mass, C. F. and Schultz, D. M.: The Structure and Evolution of a Simulated Midlatitude Cyclone over Land, Mon. Wea. Rev., 121, 889–917, 1993. 127

Matsueda, M., Kyouda, M., Toth, Z., Tanaka, H. L., and Tsuyuki, T.: Predictability of an Atmospheric Blocking Event that Occurred on 15 December 2005, Mon. Wea. Rev., 139, 2455–2470, 2011. 83

Mattson, T. G., Sanders, B. A., and Massingill, B. L.: Patterns for Parallel Programming, Addison-Wesley Professional, 2004. 108

Matula, J.: Exploiting Web Services for meteorological applications, in: Second workshop on the use of OGC/GIS standards in meteorology, Toulouse, France, 2009. 75

McCaslin, P. T., McDonald, P. A., and Szoke, E. J.: 3D visualization development at NOAA forecast systems laboratory, ACM SIGGRAPH Comput. Graph., 34, 41–44, 2000. 1, 5, 69, 73, 74, 85, 86, 98, 154

McCormick, B. H., DeFanti, T. A., and Brown, M. D.: Visualization in Scientific Computing-A Synopsis, IEEE Comput. Graphics Appl., 7, 61–70, 1987. 41

McLoughlin, T., Laramee, R. S., Peikert, R., Post, F. H., and Chen, M.: Over Two Decades of Integration-Based, Geometric Flow Visualization, Comput. Graphics Forum, 29, 1807–1829, 2010. 47, 82

Middleton, D., Scheitlin, T., and Wilhelmson, B.: Visualization in Weather and Climate Research, in: The Visualization Handbook, edited by Hansen, C. D. and Johnson, C., chap. 44, pp. 845–871, Academic Press, 2005. 68, 69, 70, 71, 75, 76

Miller, M., Buizza, R., Haseler, J., Hortal, M., Janssen, P., and Untch, A.: Increased resolution in the ECMWF deterministic and ensemble prediction systems, ECMWF Newsletter, 124, 10–16, 2010. 2, 29

Millington, S. C., Saunders, R. W., Francis, P. N., and Webster, H. N.: Simulated volcanic ash imagery: A method to compare NAME ash concentration forecasts with SEVIRI imagery for the Eyjafjallajökull eruption in 2010, J. Geophys. Res., 117, D00U17+, 2012. 82

Mirzargar, M., Whitaker, R. T., and Kirby, R. M.: Curve Boxplot: Generalization of Boxplot for Ensembles of Curves, IEEE Trans. Visual. Comput. Graphics, 20, 2654–2663, 2014. 84

Molteni, F., Buizza, R., Palmer, T. N., and Petroliagis, T.: The ECMWF Ensemble Prediction System: Methodology and validation, Q.J.R. Meteorol. Soc., 122, 73–119, 1996. 28

Montani, A., Cesari, D., Marsigli, C., and Paccagnella, T.: Seven years of activity in the field of mesoscale ensemble forecasting by the COSMO-LEPS system: main achievements and open challenges, Tellus A, 63, 605–624, 2011. 28

Montrym, J. and Moreton, H.: The GeForce 6800, IEEE Micro, 25, 41–51, 2005. 57

Moreland, K.: A Survey of Visualization Pipelines, IEEE Trans. Visual. Comput. Graphics, 19, 367–378, 2013. 60, 61, 62, 104

Munzner, T.: Visualization Analysis and Design, A K Peters/CRC Press, 1st edn., 2014. 42

Murray, D. and McWhirter, J.: Evolving IDV – Creating Better Tools for the Community, in: 23th Conference on International Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology, American Meteorological Society, 2007. 79

Murray, D., Hibbard, B., Whittaker, T., and Kelly, J.: Using VisAD to build tools for visualizing and analyzing remotely sensed data, in: Geoscience and Remote Sensing Symposium, 2001. IGARSS '01. IEEE 2001 International, vol. 1, pp. 204–206, IEEE, 2001. 79

Murray, D., McWhirter, J., Ho, Y., and Whittaker, T. M.: IDV at 5: New features and future, in: 25th Conference on International Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology, American Meteorological Society, 2009. 79

Mylne, K.: Predictability from a forecast provider's perspective, in: Predictability of Weather and Climate, edited by Palmer, T. and Hagedorn, R., chap. 23, pp. 596–613, Cambridge University Press, Cambridge, 2006. 30

Nadav-Greenberg, L., Joslyn, S. L., and Taing, M. U.: The Effect of Uncertainty Visualizations on Decision Making in Weather Forecasting, Journal of Cognitive Engineering and Decision Making, pp. 24–47, 2008. 82

Nebeker, F.: Calculating the Weather: Meteorology in the 20th Century, Academic Press, 1995. 12

Nickolls, J. and Dally, W. J.: The GPU Computing Era, IEEE Micro, 30, 56–69, 2010. 54, 55, 56, 57

Nietfeld, D. D.: The Synoptic Environment of the 11 April 2001 Central Plains Tornado Outbreak Viewed in Three Dimensions, in: Proceedings of the 19th IIPS Conference, 9-13 February 2003, Long Beach, California, 2003. 74

Nocke, T.: Visuelles Data Mining und Visualisierungsdesign für die Klimaforschung, Ph.D. thesis, Universität Rostock, 2007. 42, 48, 75, 82

Norton, A. and Clyne, J.: The VAPOR Visualization Application, in: High Performance Visualization, edited by Bethel, E. W., Childs, H., and Hansen, C., chap. 20, pp. 415–428, CRC Press, 2012. 77, 78

Novak, D. R., Bright, D. R., and Brennan, M. J.: Operational Forecaster Uncertainty Needs and Future Roles, Wea. Forecasting, 23, 1069–1084, 2008. 30, 35

Obermaier, H. and Joy, K. I.: Future Challenges for Ensemble Visualization, IEEE Comput. Graphics Appl., 34, 8–11, 2014. 2, 83

Oliphant, T. E.: Python for Scientific Computing, Comput. Sci. Eng., 9, 10–20, 2007. 76

Owens, J. D., Luebke, D., Govindaraju, N., Harris, M., Krüger, J., Lefohn, A. E., and Purcell, T. J.: A Survey of General-Purpose Computation on Graphics Hardware, Comput. Graphics Forum, 26, 80–113, 2007. 54, 57

Owens, J. D., Houston, M., Luebke, D., Green, S., Stone, J. E., and Phillips, J. C.: GPU Computing, Proceedings of the IEEE, 96, 879–899, 2008. 54, 55, 56, 57

Palmer, T. and Hagedorn, R., eds.: Predictability of Weather and Climate, Cambridge University Press, 2006. 25

Palmer, T. N.: Predictability of weather and climate: from theory to practice, in: Predictability of Weather and Climate, edited by Palmer, T. and Hagedorn, R., chap. 1, pp. 1–29, Cambridge University Press, Cambridge, 2006. 23, 24, 26, 27

Pang, A. T., Wittenbrink, C. M., and Lodha, S. K.: Approaches to uncertainty visualization, The Visual Computer, 13, 370–390, 1997. 83

Papathomas, T. V., Schiavone, J., and Julesz, B.: Stereo Animation for Very Large Data Bases: Case Study- Meteorology, IEEE Comput. Graphics Appl., 7, 18–27, 1987. 67, 69

Papathomas, T. V., Schiavone, J. A., and Julesz, B.: Applications of Computer Graphics to the Visualization of Meteorological Data, SIGGRAPH Comput. Graph., 22, 327–334, 1988. 66, 67, 68, 69, 70

Parker, S. G. and Johnson, C. R.: SCIRun: A Scientific Programming Environment for Computational Steering, in: Supercomputing '95, Proceedings of the 1995 ACM/IEEE Conference on Supercomputing, ACM, New York, NY, USA, 1995. 63

Pérez, F. and Granger, B. E.: IPython: A System for Interactive Scientific Computing, Comput. Sci. Eng., 9, 21–29, 2007. 76

Pérez, F., Granger, B. E., and Hunter, J. D.: Python: An Ecosystem for Scientific Computing, Comput. Sci. Eng., 13, 13–21, 2011. 76

Persson, A. and Andersson, E.: User guide to ECMWF forecast products, version 1.1, European Centre for Medium-Range Weather Forecasts (ECMWF), Reading, UK, 2013. 20, 22, 29, 30, 32

Petroliagis, T. I. and Pinson, P.: Early warnings of extreme winds using the ECMWF Extreme Forecast Index, Met. Apps, 21, 171–185, 2014. 33

Petty, G. W.: A First Course in Atmospheric Radiation, Sundog Publishing, 2nd edn., 2006. 15, 46, 51

Pfaffelmoser, T. and Westermann, R.: Visualization of Global Correlation Structures in Uncertain 2D Scalar Fields, Comput. Graphics Forum, 31, 1025–1034, 2012. 84

Pfaffelmoser, T., Reitinger, M., and Westermann, R.: Visualizing the Positional and Geometrical Variability of Isosurfaces in Uncertain Scalar Fields, Comput. Graphics Forum, 30, 951–960, 2011. 84, 96

Pfahl, S., Madonna, E., Boettcher, M., Joos, H., and Wernli, H.: Warm Conveyor Belts in the ERA-Interim Dataset (1979–2010). Part II: Moisture Origin and Relevance for Precipitation, J. Climate, 27, 27–40, 2014. 2, 5

Phillips, N. A.: A coordinate system having some special advantages for numerical forecasting, J. Meteor., 14, 184–185, 1957. 16

Phong, B. T.: Illumination for Computer Generated Pictures, Commun. ACM, 18, 311–317, 1975. 52

Pilar, D. H. F. and Ware, C.: Representing Flow Patterns by Using Streamlines with Glyphs, IEEE Trans. Visual. Comput. Graphics, 19, 1331–1341, 2013. 81, 119

Piringer, H., Tominski, C., Muigg, P., and Berger, W.: A Multi-Threading Architecture to Support Interactive Visual Exploration, IEEE Trans. Visual. Comput. Graphics, 15, 1113–1120, 2009. 87

Pobitzer, A., Peikert, R., Fuchs, R., Schindler, B., Kuhn, A., Theisel, H., Matković, K., and Hauser, H.: The State of the Art in Topology-Based Visualization of Unsteady Flow, Comput. Graphics Forum, 30, 1789–1811, 2011. 47

Pomroy, H. R. and Thorpe, A. J.: The Evolution and Dynamical Role of Reduced Upper-Tropospheric Potential Vorticity in Intensive Observing Period One of FASTEX, Mon. Wea. Rev., 128, 1817–1834, 2000. 4

Post, F. H., Vrolijk, B., Hauser, H., Laramee, R. S., and Doleisch, H.: The State of the Art in Flow Visualisation: Feature Extraction and Tracking, Comput. Graphics Forum, 22, 775–792, 2003. 47, 48, 82

Pöthkow, K. and Hege, H. C.: Positional Uncertainty of Isocontours: Condition Analysis and Probabilistic Measures, IEEE Trans. Visual. Comput. Graphics, 17, 1393–1406, 2011. 84

Pöthkow, K., Weber, B., and Hege, H.-C.: Probabilistic Marching Cubes, Comput. Graphics Forum, 30, 931–940, 2011. 84

Potter, K., Wilson, A., Bremer, P. T., Williams, D., Doutriaux, C., Pascucci, V., and Johnson, C. R.: Ensemble-Vis: A Framework for the Statistical Visualization of Ensemble Data, in: Int. Conference on Data Mining Workshops, Miami, Florida, pp. 233–240, IEEE Computer Society, Los Alamitos, CA, USA, 2009. 2, 84

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P.: Numerical Recipes: The Art of Scientific Computing, Cambridge University Press, 3rd edn., 2007. 14, 15, 16

Prusa, J. M., Smolarkiewicz, P. K., and Wyszogrodzki, A. A.: EULAG, a computational model for multiscale flows, Computers & Fluids, 37, 1193–1207, 2008. 19

Raoult, B., Bispham, P., Brady, A., Casado, J. L., Correa, R., Lamy-Thepaut, S., Orford, T., Richardson, D., Sahin, C., Siemen, S., Valiente, C., and Varela, D.: ECMWF Web re-engineering project, in: Third workshop on the use of OGC/GIS standards in meteorology, Exeter, UK, 2010. 75

Rautenhaus, M., Bauer, G., and Dörnbrack, A.: A web service based tool to plan atmospheric research flights, Geosci. Model Dev., 5, 55–71, 2012. 3, 4, 80, 88, 91, 107, 156

Rautenhaus, M., Grams, C. M., Schäfler, A., and Westermann, R.: GPU based interactive 3D visualization of ECMWF ensemble forecasts, ECMWF Newsletter, 138, 34–38, 2014. 155

Rautenhaus, M., Grams, C. M., Schäfler, A., and Westermann, R.: Three-dimensional visualization of ensemble weather forecasts – Part 2: Forecasting warm conveyor belt situations for aircraft-based field campaigns, Geosci. Model Dev., 8, 2355–2377, 2015a. 9, 10, 141, 155

Rautenhaus, M., Kern, M., Schäfler, A., and Westermann, R.: Three-dimensional visualization of ensemble weather forecasts – Part 1: The visualization tool Met.3D (version 1.0), Geosci. Model Dev., 8, 2329–2353, 2015b. 9, 155

Reed, R. J., Kuo, Y.-H., and Low-Nam, S.: An Adiabatic Simulation of the ERICA IOP 4 Storm: An Example of Quasi-Ideal Frontal Cyclone Development, Mon. Wea. Rev., 122, 2688–2708, 1994. 127

Reichl, F.: Interactive Sample-Based Rendering of High-Resolution Data Sets, Ph.D. thesis, Technical University of Munich, 2014. 59, 102

Rhodes, P. J., Laramee, R. S., Bergeron, R. D., and Sparr, T. M.: Uncertainty Visualization Methods in Isosurface Rendering, in: Proceedings Eurographics 2003, 1-5 September 2003, Granada, Spain, pp. 83–88, 2003. 84

Rhyne, T., Bolstad, M., Rheingans, P., Petterson, L., Shackelford, W., Botts, M. E., Pepke, E., Johnson, K. W., Hibbard, W., Dyer, C. R., Paul, B., and Treinish, L. A.: Visualization requirements in the atmospheric and environmental sciences (five case study reports), in: Proc. IEEE Conf. on Visualization 1992, pp. 428–435, IEEE, 1992. 86

Riddaway, R. W. and Hortal, M.: Numerical methods (ECMWF meteorological training course lecture series), Tech. rep., European Centre for Medium-Range Weather Forecasting, Reading, UK, 2001. 16, 17

Riley, K., Ebert, D., Hansen, C., and Levit, J.: Visually accurate multi-field weather visualization, in: Visualization 2003, VIS 2003, IEEE, pp. 279–286, 2003. 82

Robinson, A. L.: Computer Films: Adding an Extra Dimension to Research, Science, 200, 749–752, 1978. 66

Roiger, A., Schlager, H., Schäfler, A., Huntrieser, H., Scheibe, M., Aufmhoff, H., Cooper, O. R., Sodemann, H., Stohl, A., Burkhart, J., Lazzara, M., Schiller, C., Law, K. S., and Arnold, F.: In-situ observation of Asian pollution transported into the Arctic lowermost stratosphere, Atmos. Chem. Phys. Discuss., 11, 16 265–16 310, 2011. 80

Russell, I., Siemen, S., Ii, F., Kertész, S., Lamy-Thépaut, S., and Karhila, V.: Metview 4 – ECMWF's latest generation meteorological workstation, ECMWF Newsletter, 126, 23–27, 2010. 76

Sanyal, J., Zhang, S., Dyer, J., Mercer, A., Amburn, P., and Moorhead, R.: Noodles: A Tool for Visualization of Numerical Weather Model Ensemble Uncertainty, IEEE Trans. Visual. Comput. Graphics, 16, 1421–1430, 2010. 2, 83, 84, 119

Saucier, W. J.: Principles of Meteorological Analysis, Dover Publications, 1955. 65, 96

Schäfler, A., Dörnbrack, A., Wernli, H., Kiemle, C., and Pfahl, S.: Airborne lidar observations in the inflow region of a warm conveyor belt, Q.J.R. Meteorol. Soc., 137, 1257–1272, 2011. 128, 129

Schäfler, A., Boettcher, M., Grams, C. M., Rautenhaus, M., Sodemann, H., and Wernli, H.: Planning aircraft measurements within a warm conveyor belt, Weather, 69, 161–166, 2014. 4, 129, 130, 141, 150

Schiavone, J. A. and Papathomas, T. V.: Visualizing Meteorological Data, Bull. Amer. Meteor. Soc., 71, 1012–1020, 1990. 67, 68, 69

Schröder, F.: Visualisierung meteorologischer Daten, Springer, Berlin, Heidelberg, 1997. 70, 73, 74

Schröder, F.: Animation meteorologischer Daten für Massenmedien, in: Dynamische Visualisierung, edited by Buziek, G., Dransch, D., and Rase, W.-D., pp. 159–174, Springer Berlin Heidelberg, 2000. 74

Schroeder, W., Martin, K., and Lorensen, B.: Visualization Toolkit: An Object-Oriented Approach to 3D Graphics, Kitware, 4th edn., 2006. 42, 43, 44, 59, 61

Schroeder, W. J. and Martin, K. M.: Overview of Visualization, in: The Visualization Handbook, edited by Hansen, C. D. and Johnson, C., chap. 1, pp. 3–35, Academic Press, 2005. 43, 45, 48

Schultz, D. M. and Mass, C. F.: The Occlusion Process in a Midlatitude Cyclone over Land, Mon. Wea. Rev., 121, 918–940, 1993. 127

Schumacher, R. S. and Davis, C. A.: Ensemble-Based Forecast Uncertainty Analysis of Diverse Heavy Rainfall Events, Wea. Forecasting, 25, 1103–1122, 2010. 83

Schumann, T.: Nutzung von Ensemble-Prognosen im Vorhersagedienst, promet, 35, 23–29, 2009. 28, 30

Schumann, U.: On conditions for contrail formation from aircraft exhausts, Meteorol. Z., N.F. 5, 4–23, 1996. 95

Schumann, U.: Probing the Atmosphere with Research Aircraft - European Aircraft Campaings, in: Observing Systems for Atmospheric Composition, edited by Visconti, G., DiCarlo, P., Brune, W. H., Schoeberl, M., and Wahner, A., pp. 85–96, Springer, New York, 2007. 2

Sedgewick, R. and Wayne, K.: Algorithms, Addison-Wesley Professional, 4th edn., 2011. 60, 109, 110

Shen, B. W., Tao, W. K., and Green, B.: Coupling Advanced Modeling and Visualization to Improve High-Impact Tropical Weather Prediction, Comput. Sci. Eng., 13, 56–67, 2011. 82

Shirley, P., Ashikhmin, M., and Marschner, S.: Fundamentals of Computer Graphics, A K Peters, 3rd revised edn., 2009. 42, 48, 55, 62

Shutts, G., Leutbecher, M., Weisheimer, A., Stockdale, T., Isaksen, L., and Bonavita, M.: Representing model uncertainty: stochastic parametrizations at ECMWF, ECMWF Newsletter, 129, 19–24, 2011. 29

Siemen, S., Ii, F., and Russel, I.: Metview Macro – A powerful meteorological batch language, ECMWF Newsletter, 125, 30–32, 2010. 76

Skamarock, W. C., Klemp, J. B., Dudhia, J., Gill, D. O., Barker, D. M., Duda, M. G., Huang, X.-Y., Wang, W., and Powers, J. G.: A Description of the Advanced Research WRF Version 3, Tech. rep., National Center for Atmospheric Research, Boulder, Colorado, USA, 2008. 19

Smith, E.: The McIdas System, IEEE Transactions on Geoscience Electronics, 13, 123–136, 1975. 67

Snook, J. S., Stamus, P. A., Edwards, J., Christidis, Z., and McGinley, J. A.: Local-Domain Mesoscale Analysis and Forecast Model Support for the 1996 Centennial Olympic Games, Wea. Forecasting, 13, 138–150, 1998. 72

Song, Y., Ye, J., Svakhine, N., Lasher-Trapp, S., Baldwin, M., and Ebert, D. S.: An Atmospheric Visual Analysis and Exploration System, IEEE Trans. Visual. Comput. Graphics, 12, 1157–1164, 2006. 82

Spichtinger, P., Gierens, K., and Wernli, H.: A case study on the formation and evolution of ice supersaturation in the vicinity of a warm conveyor belt's outflow region, Atmos. Chem. Phys., 5, 973–987, 2005. 128

Sprenger, M. and Wernli, H.: The LAGRANTO Lagrangian analysis tool – version 2.0, Geosci. Model Dev., 8, 2569–2586, 2015. 22, 97, 100, 128

Stalling, D., Westerhoff, M., and Hege, H.-C.: amira: A Highly Interactive System for Visual Data Analysis, in: The Visualization Handbook, edited by Hansen, C. D. and Johnson, C., chap. 38, pp. 749–767, Academic Press, 2005. 63, 70

Staniforth, A. and Côté, J.: Semi-Lagrangian Integration Schemes for Atmospheric Models - A Review, Mon. Wea. Rev., 119, 2206–2223, 1991. 18

Stauffer, R., Mayr, G. J., Dabernig, M., and Zeileis, A.: Somewhere over the rainbow: How to make effective use of colors in meteorological visualizations, Bull. Amer. Meteor. Soc., 96, 203–216, 2015. 49, 94, 99, 116

Stensrud, D. J.: Parameterization Schemes: Keys to Understanding Numerical Weather Prediction Models, Cambridge University Press, Cambridge, 2007. 15

Stephenson, D. B. and Doblas-Reyes, F. J.: Statistical methods for interpreting Monte Carlo ensemble forecasts, Tellus A, 52, 300–322, 2000. 35

Stevens, B., Giorgetta, M., Esch, M., Mauritsen, T., Crueger, T., Rast, S., Salzmann, M., Schmidt, H., Bader, J., Block, K., Brokopf, R., Fast, I., Kinne, S., Kornblueh, L., Lohmann, U., Pincus, R., Reichler, T., and Roeckner, E.: Atmospheric component of the MPI-M Earth System Model: ECHAM6, J. Adv. Model. Earth Syst., 5, 146–172, 2013. 19

Stohl, A.: A 1-year Lagrangian “climatology” of airstreams in the northern hemisphere troposphere and lowermost stratosphere, J. Geophys. Res., 106, 7263–7279, 2001. 128

Stohl, A., Forster, C., Frank, A., Seibert, P., and Wotawa, G.: Technical note: The Lagrangian particle dispersion model FLEXPART version 6.2, Atmos. Chem. Phys., 5, 2461–2474, 2005. 22

Stoll, C., Gumhold, S., and Seidel, H.-P.: Visualization with stylized line primitives, in: Visualization, 2005. VIS 05. IEEE, pp. 695–702, 2005. 47

Stull, R.: Practical Meteorology: An Algebra-based Survey of Atmospheric Science, Creative Commons License, Copyright 2014 by Roland Stull, 2014. 98

Stull, R. B.: An Introduction to Boundary Layer Meteorology (Atmospheric Sciences Library), Springer, softcover reprint of the original 1st ed. 1988 edn., 1988. 15

Stull, R. B.: Meteorology for Scientists and Engineers, Cengage Learning, 2nd edn., 2000. 15, 98

Suomi, V. E., Fox, R., Limaye, S. S., and Smith, W. L.: McIDAS III: A Modern Interactive Data Access and Analysis System, J. Climate Appl. Meteor., 22, 766–778, 1983. 67

Szoke, E. J., Grote, U. H., McCaslin, P. T., and McDonald, P. A.: D3D: Overview, update, and future plans, in: Proceedings of the 18th IIPS Conference, 13-18 January 2002, Orlando, Florida, 2002. 74

Szoke, E. J., Grote, U. H., McCaslin, P. T., and McDonald, P. A.: D3D Update: Is it being used?, in: Proceedings of the 19th IIPS Conference, 9-13 February 2003, Long Beach, California, 2003. 1, 74, 75, 87, 157

Telea, A. C.: Data Visualization: Principles and Practice, A K Peters/CRC Press, 2nd edn., 2014. 42, 43, 45, 47, 48, 59, 60

Tennant, W. J., Toth, Z., and Rae, K. J.: Application of the NCEP Ensemble Prediction System to Medium-Range Forecasting in South Africa: New Products, Benefits, and Challenges, Wea. Forecasting, 22, 18–35, 2007. 30

Tennekes, H., Baede, A. P. M., and Opsteegh, J. D.: Forecasting forecast skill, in: Workshop on Predictability in the Medium and Extended Range, 17-19 March 1986, ECMWF, Reading, UK, 1986. 24

Theis, S. and Gebhardt, C.: Grundlagen der Ensembletechnik und Wahrscheinlichkeitsaussagen, promet, 35, 104–110, 2009. 35

Theis, S. and Gebhardt, C.: COSMO-DE-EPS: Das konvektionserlaubende Ensemble am DWD, promet, 37, 53–61, 2011. 28

Theis, S., Gebhardt, C., and Bouallègue, Z. B.: Beschreibung des COSMO-DE-EPS und seiner Ausgabe in die Datenbanken des DWD, Tech. rep., Deutscher Wetterdienst, Offenbach, Germany, 2012. 28

Theoharis, T., Papaioannou, G., Platis, N., and Patrikalakis, N. M.: Graphics and Visualization: Principles & Algorithms, A K Peters/CRC Press, 2008. 42, 47, 49, 51, 52, 53, 62

Tibaldi, S., Paccagnella, T., Marsigli, C., Montani, A., and Nerozzi, F.: Limited-area ensemble forecasting: the COSMO-LEPS system, in: Predictability of Weather and Climate, edited by Palmer, T. and Hagedorn, R., chap. 18, pp. 489–513, Cambridge University Press, Cambridge, 2006. 28

Tory, M. and Möller, T.: Rethinking Visualization: A High-Level Taxonomy, in: Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on, vol. 0, pp. 151–158, IEEE, Los Alamitos, CA, USA, 2004. 41

Toth, Z. and Kalnay, E.: Ensemble Forecasting at NMC: The Generation of Perturbations, Bull. Amer. Meteor. Soc., 74, 2317–2330, 1993. 26

Toth, Z. and Kalnay, E.: Ensemble Forecasting at NCEP and the Breeding Method, Mon. Wea. Rev., 125, 3297–3319, 1997. 26

Toth, Z., Kalnay, E., Tracton, S. M., Wobus, R., and Irwin, J.: A Synoptic Evaluation of the NCEP Ensemble, Wea. Forecasting, 12, 140–153, 1997. 83

Trafton, J. G. and Hoffman, R. R.: Computer-Aided Visualization in Meteorology, pp. 337–357, Lawrence Erlbaum Associates, New York, 2007. 1, 82, 91, 156, 157

Treib, M.: GPU-Based Compression for Large-Scale Visualization, Ph.D. thesis, Technical University of Munich, 2014. 59, 102

Treib, M., Bürger, K., Reichl, F., Meneveau, C., Szalay, A., and Westermann, R.: Turbulence Visualization at the Terascale on Desktop PCs, IEEE Trans. Visual. Comput. Graphics, 18, 2169–2177, 2012. 156

Treinish, L. A.: Weather forecasting for the 1996 Olympics, IEEE Comput. Graphics Appl., 16, 10–13, 1996. 72

Treinish, L. A.: Task-specific visualization design: a case study in operational weather forecasting, in: Proceedings Visualization '98, Research Triangle Park, North Carolina, pp. 405–409, 1998. 72, 73

Treinish, L. A. and Rothfusz, L. P.: Three-Dimensional Visualization for Support of Operational Forecasting at the 1996 Centennial Olympic Games, in: Proceedings of the 13th IIPS Conference, 2-7 February 1997, Long Beach, CA, pp. 2–8, 1997. 1, 5, 63, 72, 73, 74, 85, 86, 98, 154

Trembilski, A.: Two methods for cloud visualisation from weather simulation data, 17, 179–184, 2001. 82

Tufte, E. R.: Visual Explanations: Images and Quantities, Evidence and Narrative, Graphics Press, 1st edn., 1997. 68

Ueng, S. K., Sikorski, C., and Ma, K. L.: Efficient streamline, streamribbon, and streamtube constructions on unstructured grids, IEEE Trans. Visual. Comput. Graphics, 2, 100–110, 1996. 47

Unden, P., Rontu, L., Järvinen, H., Lynch, P., Calvo, J., Cats, G., Cuxart, J., Eerola, K., Fortelius, C., Garcia-Moya, J. A., Jones, C., Geert, Lenderlink, G., Mcdonald, A., Mcgrath, R., Navascues, B., Nielsen, N. W., Degaard, V., Rodriguez, E., Rummukainen, M., Sattler, K., Sass, B. H., Savijarvi, H., Schreur, B. W., Sigg, R., and The, H.: HIRLAM-5 Scientific Documentation, 2002. 18

Untch, A. and Hortal, M.: A finite-element scheme for the vertical discretization of the semi-Lagrangian version of the ECMWF forecast model, Q.J.R. Meteorol. Soc., 130, 1505–1530, 2004. 29

Upson, C., Faulhaber, T., Kamins, D., Laidlaw, D., Schlegel, D., Vroom, J., Gurwitz, R., and Van Dam, A.: The application visualization system: a computational environment for scientific visualization, IEEE Comput. Graphics Appl., 9, 30–42, 1989. 59, 63, 70

Vaughan, G., Garland, W. E., Dewey, K. J., and Gerbig, C.: Aircraft Measurements of a Warm Conveyor Belt – A Case Study, J. Atmos. Chem., 46, 117–129, 2003. 4

Vaughan, G., Methven, J., Anderson, D., Antonescu, B., Baker, L., Baker, T. P., Ballard, S. P., Bower, K. N., Brown, P. R. A., Chagnon, J., Choularton, T. W., Chylik, J., Connolly, P. J., Cook, P. A., Cotton, R. J., Crosier, J., Dearden, C., Dorsey, J. R., Frame, T. H. A., Gallagher, M. W., Goodliff, M., Gray, S. L., Harvey, B. J., Knippertz, P., Lean, H. W., Li, D., Lloyd, G., Martínez-Alvarado, O., Nicol, J., Norris, J., Öström, E., Owen, J., Parker, D. J., Plant, R. S., Renfrew, I. A., Roberts, N. M., Rosenberg, P., Rudd, A. C., Schultz, D. M., Taylor, J. P., Trzeciak, T., Tubbs, R., Vance, A. K., van Leeuwen, P. J., Wellpott, A., and Woolley, A.: Cloud Banding and Winds in Intense European Cyclones: Results from the DIAMET Project, Bull. Amer. Meteor. Soc., 96, 249–265, 2015. 3, 4

Vo, H. T.: Designing a Parallel Dataflow Architecture for Streaming Large-scale Visualization on Heterogeneous Platforms, Ph.D. thesis, Salt Lake City, UT, USA, 2011. 62

Vo, H. T., Osmari, D. K., Summa, B., Comba, J. a. L. D., Pascucci, V., and Silva, C. T.: Streaming-Enabled Parallel Dataflow Architecture for Multicore Systems, Computer Graphics Forum, 29, 1073–1082, 2010. 62

Vo, H. T., Comba, J. L. D., Geveci, B., and Silva, C. T.: Streaming-Enabled Parallel Data Flow Framework in the Visualization ToolKit, Comput. Sci. Eng., 13, 72–83, 2011. 62, 89, 107

Vo, H. T., Osmari, D. K., Comba, J., Lindstrom, P., and Silva, C. T.: Hyperflow: A heterogeneous dataflow architecture, in: Eurographics Symposium on Parallel Graphics and Visualization, pp. 1–10, The Eurographics Association, 2012. 62

Voigt, C., Schumann, U., Jurkat, T., Schäuble, D., Schlager, H., Petzold, A., Gayet, J. F., Krämer, M., Schneider, J., Borrmann, S., Schmale, J., Jessberger, P., Hamburger, T., Lichtenstern, M., Scheibe, M., Gourbeyre, C., Meyer, J., Kübbeler, M., Frey, W., Kalesse, H., Butler, T., Lawrence, M. G., Holzäpfel, F., Arnold, F., Wendisch, M., Döpelheuer, A., Gottschaldt, K., Baumann, R., Zöger, M., Sölch, I., Rautenhaus, M., and Dörnbrack, A.: In-situ observations of young contrails - overview and selected results from the CONCERT campaign, Atmos. Chem. Phys., 10, 9039–9056, 2010. 80, 95

Von der Haar, T. H., Meade, A. C., Craig, R. J., and Reinke, D. L.: Four-Dimensional Imaging for Meteorological Applications, J. Atmos. Oceanic Technol., 5, 136–143, 1988. 67

von Storch, H., Güss, S., and Heimann, M.: Das Klimasystem und seine Modellierung: Eine Einführung, Springer, 1999. 13, 19

Wallace, J. M. and Hobbs, P. V.: Atmospheric Science: An Introductory Survey, Academic Press, 2 edn., 2006. 134

Walters, D. N., Williams, K. D., Boutle, I. A., Bushell, A. C., Edwards, J. M., Field, P. R., Lock, A. P., Morcrette, C. J., Stratton, R. A., Wilkinson, J. M., Willett, M. R., Bellouin, N., Bodas-Salcedo, A., Brooks, M. E., Copsey, D., Earnshaw, P. D., Hardiman, S. C., Harris, C. M., Levine, R. C., MacLachlan, C., Manners, J. C., Martin, G. M., Milton, S. F., Palmer, M. D., Roberts, M. J., Rodríguez, J. M., Tennant, W. J., and Vidale, P. L.: The Met Office Unified Model Global Atmosphere 4.0 and JULES Global Land 4.0 configurations, Geosci. Model Dev., 7, 361–386, 2014. 15, 18, 19

Walton, J.: NAG's IRIS Explorer, in: The Visualization Handbook, edited by Hansen, C. D. and Johnson, C., chap. 32, pp. 633–654, Academic Press, 2005. 63, 70

Wanger, L. R., Ferwerda, J. A., and Greenberg, D. P.: Perceiving spatial relationships in computer-generated images, IEEE Comput. Graphics Appl., 12, 44–58, 1992. 71

Ward, M. O., Grinstein, G., and Keim, D.: Interactive Data Visualization: Foundations, Techniques, and Applications, A K Peters/CRC Press, 2010. 41, 42, 48

Ware, C.: Information Visualization: Perception for Design, Morgan Kaufmann, 3rd edn., 2012. 42, 48, 50

Ware, C. and Plumlee, M. D.: Designing a better weather display, Information Visualization, 12, 221–239, 2013. 81

Ware, C., Kelley, J. G. W., and Pilar, D.: Improving the Display of Wind Patterns and Ocean Currents, Bull. Amer. Meteor. Soc., 95, 1573–1581, 2014. 81

Warnecke, G.: The Visualisation of the Ceaseless Atmosphere, in: Remote Sensing Applications in Meteorology and Climatology, edited by Vaughan, R., vol. 201 of *NATO ASI Series*, pp. 245–257, Springer Netherlands, 1987. 69

Warner, T. T.: Numerical Weather and Climate Prediction, Cambridge University Press, 2011. 11, 15, 17, 22, 75

Washington, W. M., Olear, B. T., Takamine, J., and Robertson, D.: The application of CRT contour analysis to general circulation experiments, Bull. Amer. Meteor. Soc., 49, 882–888, 1968. 66

Watson, A. I., Fournier, J. D., Lericos, T. P., and Szoke, E. J.: The use of D3D when examining Tropical Cyclones, in: Proceedings of the 18th IIPS Conference, 13-18 January 2002, Orlando, Florida, 2002. 74

Watson, D.: Contouring: A Guide to the Analysis and Display of Spatial Data, Pergamon, 1992. 45

Watson, D.: Meteorological data visualisation using IBM Visualisation Data Explorer, in: Proceedings of the 5th ECMWF Workshop on Meteorological Operational Systems, 13-17 November 1995, Reading, England, pp. 238–251, 1995. 70

Weigle, C. and Banks, D. C.: A Comparison of the Perceptual Benefits of Linear Perspective and Physically-Based Illumination for Display of Dense 3D Streamtubes, IEEE Trans. Visual. Comput. Graphics, 14, 1723–1730, 2008. 71

Weiler, M., Klein, T., and Ertl, T.: Direct volume rendering in OpenSG, Computers and Graphics, 28, 93–98, 2004. 62

Weinstein, D. M., Parker, S., Simpson, J., Zimmerman, K., and Jones, G. M.: Visualization in the SCIRun Problem-Solving Environment, in: The Visualization Handbook, edited by Hansen, C. D. and Johnson, C., chap. 31, pp. 615–632, Academic Press, 2005. 63

Weiskopf, D.: GPU-Based Interactive Visualization Techniques (Mathematics and Visualization), Springer, 2007 edn., 2006. 59

Wenger, R.: Isosurfaces: Geometry, Topology, and Algorithms, CRC Press, 2013. 45, 119

Wernli, H.: A Lagrangian-based analysis of extratropical cyclones. II: A detailed case-study, Q.J.R. Meteorol. Soc., 123, 1677–1706, 1997. 128

Wernli, H. and Davies, H. C.: A Lagrangian-based analysis of extratropical cyclones. I: The method and some applications, Q.J.R. Meteorol. Soc., 123, 467–489, 1997. 5, 8, 97, 128, 129

Whitaker, J. S., Uccellini, L. W., and Brill, K. F.: A Model-Based Diagnostic Study of the Rapid Development Phase of the Presidents's Day Cyclone, Mon. Wea. Rev., 116, 2337–2365, 1988. 127

Whitaker, R. T., Mirzargar, M., and Kirby, R. M.: Contour Boxplots: A Method for Characterizing Uncertainty in Feature Sets from Simulation Ensembles, IEEE Trans. Visual. Comput. Graphics, 19, 2713–2722, 2013. 83, 84, 119

Wilhelmson, R. B., Jewett, B. F., Shaw, C., Wicker, L. J., Arrott, M., Bushell, C. B., Bajuk, M., Thingvold, J., and Yost, J. B.: A Study of the Evolution of a Numerically Modeled Severe Storm, Int. J. High Perform. C., 4, 20–36, 1990. 68

Wilks, D. S.: Statistical Methods in the Atmospheric Sciences, Academic Press, 3rd edn., 2011. 1, 22, 24, 25, 26, 27, 30, 31, 33, 120

Williams, D. N., Bremer, T., Doutriaux, C., Patchett, J., Williams, S., Shipman, G., Miller, R., Pugmire, D. R., Smith, B., Steed, C., Bethel, E. W., Childs, H., Krishnan, H., Prabhat, P., Wehner, M., Silva, C. T., Santos, E., Koop, D., Ellqvist, T., Poco, J., Geveci, B., Chaudhary, A., Bauer, A., Pletzer, A., Kindig, D., Potter, G. L., and Maxwell, T. P.: Ultrascale Visualization of Climate Data, Computer, 46, 68–76, 2013. 76

Wittenbrink, C. M., Pang, A. T., and Lodha, S. K.: Glyphs for visualizing uncertainty in vector fields, IEEE Trans. Visual. Comput. Graphics, 2, 266–279, 1996. 84

Woods, A.: Medium-Range Weather Prediction: The European Approach, Springer, 2006 edn., 2005. 28, 76

Wulfmeyer, V., Behrendt, A., Bauer, H.-S., Kottmeier, C., Corsmeier, U., Blyth, A., Craig, G., Schumann, U., Hagen, M., Crewell, S., Di Girolamo, P., Flamant, C., Miller, M., Montani, A., Mobbs, S., Richard, E., Rotach, M. W., Arpagaus, M., Russchenberg, H., Schlüssel, P., König, M., Gärtner, V., Steinacker, R., Dorninger, M., Turner, D. D., Weckwerth, T., Hense, A., and Simmer, C.: RESEARCH CAMPAIGN: The Convective and Orographically Induced Precipitation Study, Bull. Amer. Meteor. Soc., 89, 1477–1486, 2008. 3

Yalda, S., Zoppetti, G., Clark, R., and Mackin, K.: Interactive Immersion Learning: Flying through Weather Data onboard the GEOpod, Bull. Amer. Meteor. Soc., 93, 1811–1813, 2012. 79

Yessad, K.: FULL-POS in the cycle 40T1 of ARPEGE/IFS., Tech. rep., Meteo-France, 2014. 112, 120

Zängl, G., Reinert, D., Rípodas, P., and Baldauf, M.: The ICON (ICOsahedral Non-hydrostatic) modelling framework of DWD and MPI-M: Description of the non-hydrostatical dynamical core, Q.J.R. Meteorol. Soc., 2014. 18

Zehner, B., Watanabe, N., and Kolditz, O.: Visualization of gridded scalar data with uncertainty in geosciences, Comput. Geosci., 36, 1268–1275, 2010. 62, 84, 89

Zeileis, A., Hornik, K., and Murrell, P.: Escaping RGBland: Selecting colors for statistical graphics, Comput. Stat. Data Anal., 53, 3259–3270, 2009. 94, 99, 116

Ziegler, S., Moorhead, R. J., Croft, P. J., and Lu, D.: The MetVR case study: meteorological visualization in an immersive virtual environment, in: Proceedings of the 2001 IEEE Visualization Conference, pp. 489–596, 2001. 82