Lehrstuhl für Steuerungs- und Regelungstechnik
Technische Universität München
Univ.-Prof. Dr.-Ing./Univ. Tokio Martin Buss

Aspects of Visual Attention for Autonomous Mobile Robots

Tingting Xu

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:		Univ.-Prof. Paulo Lugli, Ph.D.
Prüfer der Dissertation:		
	1.	TUM Junior Fellow Dr.-Ing. Kolja Kühnlenz
	2.	Univ.-Prof. Gordon Cheng, Ph.D.

Die Dissertation wurde am 16.12.2009 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 13.04.2010 angenommen.

Foreword

This thesis summarizes my three-year research work carried out at the Institute of Automatic Control Engineering (LSR) of the Technische Universität München from 2006 to 2009.

First of all, I would like to express my profound gratitude towards my doctoral advisor, my “Doktorvater”, Dr.-Ing. Kolja Kühnlenz, who always supported me with his immense experience and invaluable advice, took the time for all my questions, and made me believe in myself. I sincerely thank my “Doktorgroßvater” Prof. Dr.-Ing./Univ. Tokio Martin Buss, who gave me the opportunity to conduct research in an inspiring working environment, for his encouragement and trust.

I would like to thank Prof. Gordon Cheng, Prof. Werner Schneider, Prof. Laurent Itti, and Dr. Jan Zwickel for their valuable discussions and encouragement. Great thanks also go to the other ACE team members – Andrea Bauer, Klaas Klasing, Georgios Lidoris, Quirin Mühlbauer, Florian Rohrmüller, Stefan Sosnowski, Tianguang Zhang, and Dirk Wollherr – for their fruitful discussions and immense assistance in all the phases of my work. Special thanks go to my colleagues/friends Andrea Bauer, Hao Ding, Michelle Karg, Micheal Scheint, Zheng Wang, and Haiyan Wu, for their continuous support, encouragement, and friendship as well as their careful proofreading of this thesis. I am very lucky to have them and I have really enjoyed the happy hours we spent together both in the office and in the TU-Mensa. I would also like to thank all the students who contributed to this thesis: Dong Chen, Nikolay Chenkov, Timo Fritzsch, Yuan Gong, Patrik Leyendecker, Thomas Pototschnig, Hao Wu, and Lei Ying, for their extraordinary assistance and efforts. I greatly appreciate the technical supports from my colleagues Jens Hölldampf, Quirin Mühlbauer, Matthias Rugger, Thomas Schauß, Nikolay Stefanov, Ulrich Unterhinninghofen, Herr Jaschik, Herr Gradl, Herr Kubick, Herr Stoeber, and Herr Lowitz, as well as all the help in administrative issues from Frau Schmid, Frau Werner, and Frau Renner.

Finally, I would like to thank my husband Tianguang and my parents for their unconditional love, patience, and encouragement.

Munich, December 2009 Tingting Xu

to Tianguang

...

Abstract

The deployment of technical systems in complex and unstructured everyday environments has become an essential direction of robotic research, where the limited computation capacity and the real-time requirements become the bottleneck of the system development. Cognitive abilities to interpret and select essential information from a large amount of sensory data are important and necessary, especially for a mobile robotic system.

From the extensive works in biology, cognitive psychology, and neuroscience, visual attention is considered to be one of the most powerful cognitive processes dealing with visual information selection. Considering the challenges arising in the aforementioned context, both biologically plausible and technically applicable robot visual attention strategies should be developed to bridge the gap between fundamental studies and specific technical realizations.

This thesis focuses on the investigation of goal-directed visual attention strategies for autonomous mobile robots, explored from three different perspectives: the stimulus-dependent aspect, the task-relevant spatial aspect, and the task-relevant temporal aspect. Two information-based metrics are proposed to enable well-timed perception of temporal and spatial stimuli, which is a critical factor for awareness of unexpected events and for ensuring the working order of robots. Integrated approaches to top-down and bottom-up attention selection are elaborated, where the determination of robot spatial attention allocation for task-relevant information is investigated. A human-inspired temporal attention control strategy is proposed, considering the challenge of a limited field of view in multi-object tasks. Evaluation and demonstration are carried out in simulations and experiments. The main contributions are qualitative improvements of sensitive awareness of environment dynamics, efficient, flexible, and adaptable enhancement of task-relevant information, as well as significant reduction of the overall perception uncertainty through temporal attention planning.

In this work, application-oriented attention control considering characteristics of mobile robots functioning in dynamic environments is studied in a general and integrated manner for the first time. The contributions advance the state of the art in cognitive robot design and provide valuable insights for future research.

Zusammenfassung

Eine wesentliche Forschungsrichtung im Bereich der Robotik ist der Einsatz von technischen Systemen in komplexen, unstrukturierten, alltäglichen Umgebungen, in denen die begrenzte Rechenleistung und die Echtzeitanforderungen die Engpässe der Systementwicklung darstellen. Kognitive Fähigkeiten spielen eine Schlüsselrolle, um die wesentlichen Informationen aus umfangreichen Sensordaten zu extrahieren und zu interpretieren, insbesondere für mobile Robotersysteme.

In zahlreichen Arbeiten in Biologie, kognitiver Psychologie und Neurowissenschaft wird die visuelle Aufmerksamkeit als eine der mächtigsten kognitiven Prozesse für die Auswahl visueller Information angesehen. Unter Berücksichtigung der obengenannten Herausforderungen, sollen biologisch plausible und technisch anwendbare visuelle Aufmerksamkeitsstrategien für Roboter entwickelt werden, um die Lücke zwischen den fundamentalen Forschungen und spezifischen technischen Realisierungen zu schließen.

Der Fokus dieser Arbeit befasst sich mit der Entwicklung zielgerichteter visueller Aufmerksamkeitsstrategien für mobile Roboter, die aus drei verschiedenen Perspektiven untersucht wird: der reizbasierte Aspekt, der aufgabenorientierte räumliche Aspekt und der aufgabenorientierte zeitliche Aspekt. Zwei informationsbasierte Metriken werden vorgeschlagen, um eine rechtzeitige Wahrnehmung der räumlichen und zeitlichen Reize zu ermöglichen, die einen kritischen Faktor für die Erkenntnis unerwarteter Erreignisse und Garantie des Arbeitszustandes eines Roboters darstellt. Integrierte Konzepte für top-down und bottom-up basierte Aufmerksamkeitsselektion sind entwickelt, wobei die räumliche Bestimmung der Roboteraufmerksamkeit für aufgabenrelevante Objekte betrachtet wird. Eine Strategie zur zeitlichen Koordination der Aufmerksamkeitssteuerung inspiriert vom menschlichen Verhalten wird vorgeschlagen, die die Problematik begrenzter visueller Sichtfelder in Multi-Objekt-Aufgaben berücksichtigt. Evaluierung und Demonstration erfolgen in Simulationen und Experimenten. Beiträge sind die qualitative Verbesserung der Empfindlichkeit für Wahrnehmung einer dynamischen Umgebung, effiziente, flexible und anpassungsfähige Performanzsteigerung der aufgabenrelevanten Informationsselektion, sowie eine signifikante Reduktion der gesamten Wahrnehmungsunsicherheit durch die zeitliche Aufmerksamkeitsplanung.

In dieser Arbeit werden erstmalig anwendungsorientierte Aufmerksamkeitssteuerungen, die Eigenschaften von Robotern berücksichtigen, in einer integrierten und allgemeinen Form untersucht. Die Beiträge verbessern den Stand der Technik im Design kognitiver Roboter und liefern wertvolle Einblicke für die zukünftige Forschung.

Contents

1 Introduction 1

1.1 Challenges 2

1.2 Main Contributions and Outline of the Thesis 4

2 Related Work 8

2.1 Biological Terminology 8

2.2 Fundamental Theories in Cognitive Psychology and Neuroscience 9

2.3 Computational Models of Visual Attention 11

2.4 Technical Realizations in the Robotics Domain 13

2.5 Summary 19

3 Information-Based Bottom-Up Perception for Attention Control 20

3.1 Introduction 20

3.2 Local Surprise for Unexpected Stimuli 21

3.2.1 State of the Art 21

3.2.2 The Saliency Map Model 21

3.2.3 Local Surprise Definition 26

3.2.4 Performance Evaluation 28

3.2.5 Discussion 31

3.3 Global Surprise for Representation of Environment Dynamics 32

3.3.1 State of the Art 33

3.3.2 Global Surprise Definition 33

3.3.3 Experimental Investigation 34

3.3.4 Discussion 35

3.4 Surprise-Driven Robot Attention Control 36

3.4.1 Robot Decision-Action Design 36

3.4.2 Performance Demonstration 36

3.4.3 Discussion 41

3.5 Summary 41

4 Integrated Approaches to Top-Down and Bottom-Up Attention Control 43

4.1 Introduction 43

4.2 Top-Down Biased Bottom-Up Attention Strategy (TBB) 44

4.2.1 State of the Art 44

4.2.2 Model of TBB 45

4.2.3 Performance Evaluation 52

4.2.4 Discussion 58

4.3 Autonomous Switching of Attention Mechanisms (TOB) 61
4.3.1 State of the Art 61
4.3.2 Model of TOB 62
4.3.3 Performance Evaluation 66
4.3.4 Discussion 69
4.4 Summary 71

5 Human-Inspired Temporal Attention Control for Multi-Object Tasks 72
5.1 Introduction 72
5.2 Experimental Study of Human Attention during Motion 73
5.2.1 Experiment: Materials and Procedure 74
5.2.2 Results and Inspirations for Robotic Systems 75
5.2.3 Discussion 79
5.3 Temporal Attention Planning for Multiple Target Objects 80
5.3.1 State of the Art 80
5.3.2 System Definition 81
5.3.3 Temporal Attention Planning 83
5.3.4 Simulation Results 87
5.3.5 Discussion 95
5.4 Towards Attention and Motion Coordination in Multi-Robot Systems . . . 96
5.4.1 State of the Art 96
5.4.2 Behavior-Based Attention and Motion Coordination 97
5.4.3 Simulation Results 98
5.4.4 Discussion 101
5.5 Summary 101

6 Conclusions and Future Directions 103
6.1 Concluding Remarks 103
6.2 Outlook 105

A Experimental Platform: The Autonomous City Explorer (ACE) Robot 106
A.1 Hardware Components 106
A.2 The Active Multi-Focal Vision System 107
A.3 Attention Control for Human-Robot Interaction 108

B A High-Speed Multi-GPU Implementation of Bottom-Up Attention 112
B.1 State-of-the-Art Implementations 112
B.2 Graphics Processing Units 113
B.3 Multi-GPU Implementation Details 115
B.4 Results and Discussion 124

Bibliography 127

Notations

Abbreviations

2D	two-dimensional
3D	three-dimensional
AI	artificial intelligence
ACE	Autonomous City Explorer
AKF	adaptive Kalman-filter
ANOVA	ANalysis Of VAriance
AP	attention planning
BY	blue-yellow
C	color
CM	conspicuity map
CUDA	Compute Unified Device Architecture
DoG	difference-of-Gaussian
fps	frame per second
FM	feature map
FOA	focus of attention
FOV	field of view
GPU	Graphics Processing Unit
GS	global surprise
HFOV	horizontal field of view
HSV	color space containing hue, saturation and value channels
I	intensity
IOR	inhibition-of-return
KF	Kalman-filter
LS	local surprise
O	orientation
pdf	probability density function
RG	red-green
RGB	color space containing red, green, and blue channels
RR	Round-Robin algorithm
SIFT	scale-invariant feature transform
SN	switching number
TBB	top-down biased bottom-up
TOB	top-down or bottom-up
WTA	winner-take-all

Conventions

Scalars, Vectors, and Matrices

x	scalar
$\boldsymbol{x}$	vector
X	constant
$\boldsymbol{X}$	matrix
$f(\cdot)$	scalar function
$\boldsymbol{f}(\cdot)$	vector function
$\\|\cdot\\|$	Euclidian norm
$\|\cdot\|$	absolute value
$\|\cdot\|_{\geq 0}$	negative values discarded
$\min(\cdot)$	minimum value
$\max(\cdot)$	maximum value
$\hat{x}$	homogeneous coordinate
$\tilde{x}$	estimation error
$\dot{x}$, $\ddot{x}$	equivalent to $\frac{\mathrm{d}}{\mathrm{d}t}x$ and $\frac{\mathrm{d}^2}{\mathrm{d}t^2}x$

Subscripts and Superscripts

$(\cdot)_{\max}$	maximum value
$(\cdot)_{\min}$	minimum value
$(\cdot)^{-1}$	inverse
$(\cdot)^{+}$	pseudo-inverse
$(\cdot)^{T}$	transposed
$(\cdot)^{*}$	optimal or expected value
$(\cdot)^{-}$	predicted (a priori) value
$(\cdot)_{pos}$	position
$(\cdot)_{pose}$	pose
$(\cdot)_{vel}$	velocity
$(\cdot)_{tran}$	translation
$(\cdot)_{rot}$	rotation
$(\cdot)_{k}$	value at time step k
$(\cdot)_{k+1\|k}$	prediction for time step $k+1$ based on the result of time step k
${}_{0}(\cdot)$	value in the world frame
${}_{r}(\cdot)$	value in the robot frame
${}_{c}(\cdot)$	value in the camera frame

Symbols

Information-Based Bottom-Up Perception for Attention Control

A, B	constants
c	fine scale (center)
c_{ex}	constant self-excitation factor
c_{inh}	constant neighbor-induced inhibition factor
C_{inh}	constant inhibitory term
e	eigenvalue
$E(\cdot)$	expectation value
G	Gabor-filter operator
H	histogram of a grayscale image
i	pixel index
j	pixel value range
$\bar{I}$, $\bar{C}$, $\bar{O}$	intensity, color and orientation conspicuity map
k	time step
$KL(\cdot\|\|\cdot)$	Kullback-Leibler divergence
l	index
M	image or map
$N(\cdot)$	iterative normalization
$N_{1D}(\cdot)$	1D-normalization
p	probability density function
r, g, b	pixel value in red, green, and blue channels in the RGB color space
s	coarse scale (surround)
S	saliency map
S_0	world frame
T	global surprise value
$Var(\cdot)$	variance
x, y	horizontal and vertical pixel coordinate in 2D image
x', y'	transformed pixel coordinate in a Gabor-filter
α	shape parameter in a Gamma pdf
β	inverse scale in a Gamma pdf
χ	wavelength of the cosine factor in a Gabor-filter
δ	scale difference between a fine scale and a coarse scale
ϵ	standard deviation of the Gaussian envelope in a Gabor-filter
γ	spatial aspect ratio in a Gabor-filter
$\gamma(\cdot)$	Gamma pdf
$\Gamma(\cdot)$	Euler Gamma function
λ	saliency value
ν	information measure
Ω	robot view direction
ψ	phase offset in a Gabor-filter

σ	image scale
σ_{ex}	standard deviation of the Gaussian self-excitation distribution
σ_{inh}	standard deviation of the Gaussian neighbor-induced inhibition distribution
τ	local surprise value
θ	orientation angle of the normal to the parallel stripes of a Gabor-filter
$\Psi(\cdot)$	Digamma function
ξ	forgetting factor
$\ominus$	across-scale subtraction
$\oplus$	across-scale combination

Integrated Approaches to Top-Down and Bottom-Up Attention Control

$\boldsymbol{A}$	state transition matrix
$bt, tb, tt, t1, t2, b1, b2$	transition condition
$B0T0K0$	exhaustive search without attentional pre-selection
$B1T0K0$	purely bottom-up attention selection
$B1T1K0$	top-down biased bottom-up attention selection
$B1T1K1$	top-down biased bottom-up attention selection with KF-adaptation
BU_e	bottom-up state in exploring mode
BU_s	bottom-up state in searching mode
c	contribution of a FM or a CM
$\boldsymbol{c}$	contribution vector of FMs or CMs
$\boldsymbol{H}$	observation transition matrix
i	feature index
$\boldsymbol{I}$	unit matrix
j	center-surround scale index
k	time step
$\boldsymbol{K}$	Kalman gain
$KL(\cdot\|\|\cdot)$	Kullback-Leibler divergence
l	frame number counter
L	constant
m	index of the candidate image region
M	total number of target objects
n	dimension of the system state
n_1	number of the detected target objects
n_2	total number of the candidate image regions
n_3	average time cost of SIFT matching
p	pdf
$\boldsymbol{P_x}$	covariance matrix of system state estimation
$\boldsymbol{Q}$	covariance matrix of the process noise
$\boldsymbol{R}$	covariance matrix of the measurement noise
T_0	computation time of initialization

T_1	mechanical time of initialization
T_2	approximate time of image capturing
T_3	mechanical time of camera saccade
T_4	mechanical time of robot rotation
TD_s	top-down state in searching mode
TD_o	top-down state in operating mode
T_{min}, T_{max}	lower or upper boundary of saliency value to build an object map
$\boldsymbol{v}$	measurement noise
V	average grayscale value in an image region
w	weight for a FM or a CM
$\boldsymbol{w}$	weighting vector for FMs and CMs
$\boldsymbol{x}$	system state
$\boldsymbol{y}$	observation or measurement
$\boldsymbol{z}$	process noise
$\boldsymbol{\mu}$	system state estimation

Human-Inspired Temporal Attention Control for Multi-Object Tasks

$\mathbf{0}_n$	matrix of zeros of dimension $n \times n$
${}^0\boldsymbol{T}_r, {}^r\boldsymbol{T}_c$	homogeneous transformation matrix
${}_0\boldsymbol{x}_r$	robot position in the world frame
$\boldsymbol{A}$	system transition matrix
e	eigenvalue of P
F	degree of freedom in ANOVA computation
$\boldsymbol{H}$	measurement matrix
$\boldsymbol{I}_n$	unit matrix of dimension $n \times n$
j	object index
J	cost functional
$\boldsymbol{K}$	Kalman gain
l	index
$\boldsymbol{L}$	confident sensing range along the camera optical axis
M	total number of task-relevant objects
M_{seen}	number of the visible objects
n	dimension of the task space
O	object
p	p-value of one-way ANOVA computation
$pFOA_{\xi,\eta}$	percentage targeting duration of the FOA in system state ξ on object η
pSN_{ξ}	percentage switching number of the FOA in state ξ
P	participant
$\boldsymbol{P}$	covariance matrix of object position estimation
$\boldsymbol{Q}_k$	process noise covariance matrix
$\boldsymbol{R}_k$	measurement noise covariance
$\boldsymbol{S}_0$	world frame

$\boldsymbol{S}_c$	camera frame
$\boldsymbol{S}_i$	image plane
$\boldsymbol{S}_r$	robot frame
$\boldsymbol{v}_k$	measurement noise
$\boldsymbol{w}_k$	process noise
$\boldsymbol{x}$	Cartesian point
X	task-irrelevant distractors
$\boldsymbol{y}$	system measurement
η	index of the attended target object
$\boldsymbol{\Omega}$	camera orientation with respect to the robot frame
Π	FOV
$\boldsymbol{\Psi}$	camera view angles including pan- and tilt-angle
σ_x, σ_z	standard deviation of the perception process error
ς	scalar
Θ	robot orientation with respect to the world frame
ξ	state index

List of Figures

1.1 Perception-action closed loop. 1

1.2 Various aspects of robot attention behaviors: a) initial state; b) attending to a task-irrelevant (implicitly task-relevant) stimulus; c) promoting task-relevant information; d) attending to one of task-relevant objects; e) temporal attention selection of task-relevant objects. Stars: task-relevant information/objects; human-shaped symbols: task-irrelevant stimuli. 3

1.3 Outline of the thesis. 5

1.4 The Autonomous City Explorer robot [213] 6

2.1 Human eye diagram. Source: NEI Catalog number NEA09, National Eye Institute (URL: www.nei.nih.gov). 8

3.1 The saliency map computation model 22

3.2 Example images in a dyadic Gaussian pyramid. From left to right: 640×480 at scale 0; 320×240 at scale 1; 160×120 at scale 2; 80×60 at scale 3. . . 23

3.3 Example images from left to right: I-map, RG-map and BY-map at scale 3 in the Gaussian pyramids. 23

3.4 Example images from left to right: O-maps at scale 3 in the Gaussian pyramids for orientations 0°, 45°, 90°, and 135°. 24

3.5 Example images. Left: a BY-map at scale 3; Middle: a BY-map at scale 6; Right: $BY(3,6)$, a BY feature map at scale 3-6. 25

3.6 Example images from left to right: Conspicuity maps $\bar{I}$, $\bar{C}$, and $\bar{O}$ in intensity, color and orientation channels as well as the resulting saliency map S. 25

3.7 Computation structure of LS maps. 26

3.8 LS induced by salient object onset – a human entering the scene. 29

3.9 Robot gaze shift towards the position of the LS maximum, which is induced by temporal novelty (hand/cup movement in the left two columns) and by spatial saliency (salient red cylinder mounted on the wall in the middle two columns). Upper row: original images; Middle row: the respective saliency maps; Lower row: LS maps. 29

3.10 Image sequence for object detection task. Left sub-columns: the original input images. Right sub-columns: the respective masked result images with the frame number. Masked regions: the robot FOA; Rectangles: image regions containing the maximum LS; Circles: the detected objects. 30

3.11 Comparison of LS map, motion map, and difference map of consecutive saliency maps computed using two consecutive input images. 32

3.12 Experimental investigation of GS using four different scenes with different scene dynamics. From left to right: a static indoor scene; a city-center scene with slight pedestrian movement; an indoor scene with an unexpected event – a human entering the scene; a street scene with a high-velocity vehicle. . 35
3.13 Robot decision-action design. 36
3.14 Left: the ACE robot with an active camera head and four artificial landmarks mounted on the wall. Right: color-based landmark detection in 2D input images. 37
3.15 A top view of the robot locations and view directions during the robot locomotion, considering or not considering the GS. Triangles: the actual robot positions; Solid circles: the artificial landmarks; Solid lines: robot view directions for the self-localization task; Dashed line: robot view direction toward the maximum LS. 37
3.16 Image sequence captured at different locations (x-coordinate on the images) for the self-localization task without consideration of GS and LS. The numbers under the images denote the camera view directions with respect to the locomotion direction. 39
3.17 Robot position estimation errors using constant view directions without planning (red, dash-dot line with point markers), using planned view directions (blue, dashed line with square markers), and using planned view direction and considering LS/GS as well (black solid line). 39
3.18 GS values during robot locomotion. The threshold was set to 0.1 bit. . . . 40
3.19 Image sequence captured at different locations (x-coordinate on the images) for the self-localization task taking visual stimuli (LS and GS) into consideration. The numbers under the images denote the camera view directions with respect to the locomotion direction. 40

4.1 Overview of the TBB model consisting of prediction, verification and adaptation. 46
4.2 Saliency map computation illustrated in the feed-forward connections and model parameter update illustrated in the feedback connections. 47
4.3 Contribution of CMs in building a salient image region, illustrated by the squares. Upper image: the object map; Lower images: the conspicuity maps in I-, C-, and O-channels from left to right. 49
4.4 Column 1: object maps predicted using B1T1K1; Column 2: saliency maps computed using B1T1K1; Column 3: object maps predicted using B1T0K0; Column 4: saliency maps computed using B1T0K0. Numbers on the object maps indicate the fixation sequence along with a descending saliency value of the selected image region candidates. “Yes” on the object maps indicate that an image region candidate contains a target object. 53
4.5 The sample image of the target object used in the experiment (left) and four images of the target objects captured by the high-resolution telephoto camera during the experiment using B1T1K1. Blue circles: the matched SIFT feature points. 54

4.6 Update of thresholds for object map. Left column: object map (upper) and saliency map (lower) computed using the initial thresholds; Right column: object map (upper) and saliency map (lower) using the updated thresholds. Numbers on the object maps indicate the fixation sequence along with a descending saliency value of the selected image region candidates. “No” on the object maps indicate that an image region candidate contains a target object. 55

4.7 The weights variation for $CM\text{-}I$, $CM\text{-}C$, and $CM\text{-}O$ using and not using KF. 55

4.8 Left column: object map (upper) and saliency map (upper) with Kalman filtering; Right column: object map (upper) and saliency map (upper) without Kalman filtering. Numbers on the object maps indicate the fixation sequence along with a descending saliency value of the selected image region candidates. 56

4.9 From left to right: update of the object maps (upper) and the saliency maps (lower) aided by a KF. Rectangles: image regions selected as candidates; Circles in the object maps: image regions with a descending saliency value from a) to c), inhibited by the Kalman filtering; Circles in the saliency maps: image regions with an ascending saliency value from a) to c), enhanced by the Kalman filtering. 57

4.10 Comparison of the approximate necessary fixations (left) and detection rates (right) for 4 target objects using four different strategies. 57

4.11 Saliency map computed directly from a sample image. Left: sample image. Right: the respective saliency map. 59

4.12 Weights update of FMs at different scales in different channels: I-channel (top left), RG-channel (top right), BY-channel (bottom left), and O_{0°-channel (bottom right). 60

4.13 Finite state machine of the autonomous switching mechanism. Three internal robot modes: exploration, searching and operation. Four different attention states: bottom-up state in exploring mode (BU_e), bottom-up state in searching mode (BU_s), top-down state in searching mode (TD_s), and top-down state in operating mode (TD_o). 63

4.14 Left column: bottom-up state; Right column: top-down state; Upper images: original input images; Lower images: the resultant robot attention windows. Rectangles: the salient/surprising image regions (the same as the masked region in bottom-up state); Masked regions: the current robot FOA; Circle: the detected target object. 64

4.15 Experimental setup consisting of three different signs. 66

4.16 Results of experiment 1 comprising original input images (the first and third rows) and their respective resultant images with the robot FOA (the second and the fourth rows). Numbers on the original images indicate the frame number. The markers on the left top corner of the resultant images are defined as follows: Rectangle: bottom-up attention in searching state (BU_s); Diamond: top-down attention in searching state (TD_s); Circle: top-down attention in operating state (TD_o); Star: bottom-up attention in exploring state (BU_e). Rectangles: the salient/surprising image regions; Masked regions: the current robot FOA; Circle: the detected target object. 67

4.17 Results of experiment 2 comprising the resultant images of robot FOA. Numbers indicate the frame numbers. The solid circles: the ACE robot. The arrows on the robot: the view direction of the active camera head. The dashed line: the robot trajectory. Rectangles: the salient/surprising image regions; Masked regions: the current robot FOA; Circle: the detected target objects. 69

4.18 Relative entropy (observation uncertainty) evolution (dashed line) and the respective attention state (solid line). The semi-transparent areas indicate the time intervals in which the robot was in operating state. The blank areas indicate the time intervals in which the robot was in searching state. Some representative time points are shown with their frame numbers. . . . 70

5.1 Example scenarios for robots with limited FOV facing more than one task-relevant object. Left: the robot teacher SAYA looking at her students [208]. Middle: the cooperation of two robots for repairing a broken water pipe [28]. Under restricted communication, each robot should not only observe the water pipe but also monitor the manipulators of the other robot. Right: Nao robots in RoboCup 2009 [158]. 72

5.2 Experiment overview captured by the overlooking camera. Left: Start positions of participant P1, P2, P3, object (O), as well as a reference bar of 0.8 m; Right: final positions of the participants. An equal distribution around the object is built. 74

5.3 Participants' state variation among *actor*, *observer* and *idler*. The solid line: P1; the dash-dot line: P2; The dashed line: P3. 76

5.4 Participants' body movements computed as the relative angle between the reference bar and the line formed by each participant and the target object in [deg]. Solid lines: P1; Dash-dot lines: P2; Dashed lines: P3; Dotted lines: difference between the left relative angle and the right relative angle of each participant (the formation error). 77

5.5 Participants' gaze direction switching among other participants (P1, P2, or P3), the target object (O), and other distractors (X). Solid lines: P1; Dash-dot lines: P2; Dashed lines: P3. 78

5.6 Left: Percentage number of the participants' FOA switchings pSN_a in the *actor* state and pSN_o in the *observer* state, shown in the mean values and their standard deviations; Right: Percentage FOA duration attending to the other participants (P) $pFOA_{P,a/o/i}$, to the object (O) $pFOA_{O,a/o/i}$, and to other distractors (X) $pFOA_{X,a/o/i}$ in the state *actor*, *observer* or *idler*, shown in the mean values and their standard deviations. 78

5.7 Definition of frames of reference. 81

5.8 Standard deviations of the vision-based position measurement errors $\sigma_x({}_c\boldsymbol{x}_j)$ in the x_c-direction (upper row) and $\sigma_z({}_c\boldsymbol{x}_j)$ in z_c-direction (lower row) at different positions ${}_c\boldsymbol{x}_j$. Left column: real measurements; Right column: interpolated surfaces $\boldsymbol{f}$ based on the real measurements. 88

5.9 Overview of simulation 1. A robot with position $(0, 0)$ m and four static objects are located in this scenario. The blue solid line indicates the initial view direction of the robot. 89

5.10 Observation at each time step. Solid circles: the robot and the static objects; Solid lines: limited FOV of the robot; Stars: estimated object positions; Top-right number in each sub-figure: frame number (time step). 90

5.11 Results of simulation 1. Top: predicted optimal view direction for individual object and the actual view direction at each time step; Middle: individual uncertainty of object position estimation at each time step; Bottom: position estimation error of each object at each time step. The solid line with circular markers: actual robot view direction; Dashed lines: object 1; Solid lines: object 2; Dash-dot lines: object 3; Dotted lines: object 4. 91

5.12 Overview of simulation 2. A robot with two objects is located in this scenario. Object 2 (O2) moves at a constant velocity. 92

5.13 Results of simulation 2. The actual robot view directions, position estimation errors, and velocity estimation errors of object 1 and 2 using the AP strategy (row 1 to 3) and the RR algorithm (row 4 to 6). 93

5.14 Overview of simulation 3. Left: initial positions of robot 0, 1, and 2 and the predefined trajectories of robot 1 and 2, as well as a view from the robot perspective (top left); Top right: Sony VID30 camera with a pan/tilt platform; Bottom right: an input image and marker detection result. . . . 94

5.15 Results of simulation 3. Row 1 and 2: the position estimation errors of robot 1 and 2. Solid lines: vision-based object position measurements or predictions (if the respective robot is not in the FOV); Dashed lines: position estimation errors. Row 3: the actual robot view direction at each time step. Positive pan-angle denotes that robot 0 is attending to robot 1. Negative pan-angle denotes that robot 0 is attending to robot 2. 94

5.16 Combination of two human-inspired robot behaviors *observer* and *actor*. . 98

5.17 Simulation results in a leader-referenced formation task. Left: desired formation of robot 0 with robot 1 (magenta) and robot 2 (cyan); Right: pan-angle, x- and y-position (from top to bottom) of robot 0. Solid lines: actual value; Dashed lines: desired value. 99

5.18 Overview of the leader-referenced formation task (until time step 35). Circles on the dotted line: trajectory of robot 0 at each time step; Squares on the dash-dot line: trajectory of robot 1 at each time step; Triangles on the dashed line: trajectory of robot 2 at each time step; Large triangles in solid lines: rectangles formed by robot 0, 1, and 2. 99
5.19 Simulation results in a 3-robot system for a neighbor-/unit-center referenced formation task. Left: initial positions of the robots (upper) and final positions of the robots (lower); Right: pan-angle, x- and y-position (from top to bottom) of one robot. Solid lines: actual value; Dashed lines: desired value. 100
5.20 Simulation results in a 6-robot system for a neighbor-/unit-center referenced formation task. Left: initial positions of the robots (upper) and final positions of the robots (lower); Right: pan-angle, x- and y-position (from top to bottom) of one robot. Solid lines: actual value; Dashed lines: desired value. 100

A.1 New revision of the high-performance active camera platform [97, 225] . . . 107
A.2 Human prediction map model based on color and motion maps. 108
A.3 Motion maps before and after the stabilization algorithm. 109
A.4 Determination of search windows for efficient image processing. Upper images: the search windows on the original images with current robot FOA denoted by circles; Lower images: the search windows on the human prediction maps. 110
A.5 Left: attention selection before (upper) and after (lower) the camera motion control. Right: possible false positive errors on the original images (upper) and their respective human prediction maps (lower). Circles: current robot FOA. 111

B.1 The GPU thread batching model. 114
B.2 Data flow diagram for GPU implementation of saliency map computation . 116
B.3 Image data padding 117
B.4 The convolution in the horizontal direction 118
B.5 The Cuda-image of 256 × 256 pixels (left) and the filter kernel (right) prepared for the FFT 120
B.6 The image lists configuration 121
B.7 The pipeline structure 123
B.8 The multiplexer and demultiplexer structure 123
B.9 The petri-net structure for the multi-threaded mode 124
B.10 Four GPUs installed in a PC. 125
B.11 Comparison of computation times using 1 to 4 GPUs. 126

List of Tables

3.1 LS values computed using different sample saliency values of two pixels λ_1 and λ_2. Group 1 and 2: If the saliency value variations are the same, the pixel with a higher saliency value wins a higher LS value; Group 3 and 4: The pixel with a higher saliency value variation wins a higher LS value. . . 28

4.1 Symbol definition for different strategies. "+": with; "−": without. 52
4.2 Computational and mechanical time cost for object detection. The constants $T_0 + T_1 = 6$ s, $T_2 = 0.033$ s, $T_3 = 1$ s, $T_4 = 20$ s. n_1: total number of saliency maps computed using top-down information; n_2: total number of target object candidates; n_3: the average times for SIFT computation in one candidate region. 58
4.3 Average computation time in the experiment. 70

5.1 Comparison of the AP strategy and the RR algorithm in terms of position estimation uncertainty, position estimation error, and visual coverage. Data are averaged over ten simulations. 91
5.2 Inspirations from human behavior in the previous experimental investigation. 95

B.1 Computational time registration using 1 GPU. 125
B.2 Comparison between iLab's implementation [148] and this implementation. 126

1 Introduction

A wealth of information creates a poverty of attention
and a need to allocate that attention efficiently.
Herbert A. Simon 1971 [173]

In recent years, developing cognitive abilities for technical systems has become a very popular focus of robotics research. Humans, capable of elegant cognitive mechanisms such as perception, attention, memory, action, learning, and planning in everyday concerns, can be regarded as an efficient biological model for technical systems. Fundamental cognitive processes in the human brain have been intensively studied in cognitive psychology and neuroscience, and can be modeled as a perception-action closed loop, illustrated in Fig. 1.1, in which humans perceive the environment via various sensor modalities such as vision, hearing, taste, smell, touch etc, select and process essential information, and make an action decision to interplay with the world. A mapping of the fundamental findings in cognitive psychology and neuroscience about human cognitive information processing on a robot system is envisioned such that a robot system can be more reliable, flexible, adaptive, and robust.

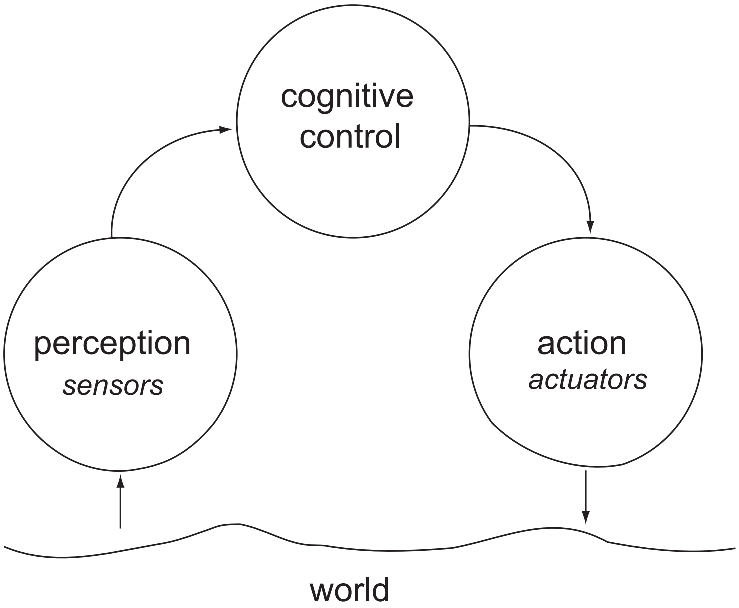

Fig. 1.1: Perception-action closed loop.

Among various sensor modalities, vision is a very strong source of information and can provide a large amount of information about the world. Through continuous development of visual sensor technology, more and more information can be acquired in a fixed time interval. Real-time information acquisition is no longer a major problem.

However, due to the limited processing capacity or the real-time constraints, not all the information can be further processed in detail. Relevant information should be selected and processed either at a higher resolution or earlier, while the others should be inhibited by the cognitive process concurrently. This kind of visual information selection process is called *visual attention*, and it plays an essential role in human perception and cognition. Consistent selection of the environment fraction of interest is called attention selection, which is facilitated by rapid eye movements named *saccades*.

Studies about human visual perception show that visual attention selection is affected by two distinct types of attentional mechanisms: top-down and bottom-up. Top-down signals are derived from the task specification or the previous knowledge and highlight the task-relevant information. It is goal-directed and efficient for task accomplishment. In contrast, bottom-up attention selection is inspired by neuronal architecture of early vision and driven by distinct stimuli based on primary visual features. Interaction and coordination of both proceed gaze fixation point selection and guide the visual behavior.

To deal with the limited processing capability of the most technical systems, especially autonomous mobile robots, a biologically plausible and technically applicable visual attention system is to be developed, in order to bridge the gap between the fundamental cognitive studies and the robotics research.

The main challenges faced by developing robot attention control in the perception-action closed loop are summarized below.

1.1 Challenges

Vision and attention have been intensively studied in cognitive psychology and neuroscience work. Various computational models of attention selection have also been proposed to achieve a human-like visual attention behavior in a natural environment. However, attention modeling and implementation are a relatively new research topic in the robotics domain. Up to now, technical realizations of visual attention control have only been accomplished for very limited scenarios. Most technical attentional mechanisms for camera control in mobile robotics are based on principles of task-relevant information maximization, neither considering bottom-up influences nor capacity limitations of computational resources. An integrated attention control for robot operation in complex environments is missing. The key issues of the challenges for an advanced exploration of various aspects of robot attention considered in this thesis are illustrated in Fig. 1.2 and summarized in this section.

How to perceive “task-irrelevant” stimuli while performing a task

An ordinary robot system usually performs a specific task with quantifiable purposes. Computational resources have been mainly applied in the primary robot task. In recent years, the deployment of mobile robotic systems in an unstructured environment has become a trend in robotic research. Since both robot mobility and dynamic environment are key factors increasing task performance uncertainty, robot attention should also be paid to task-irrelevant stimuli such as abrupt appearing and disappearing of objects, dynamic

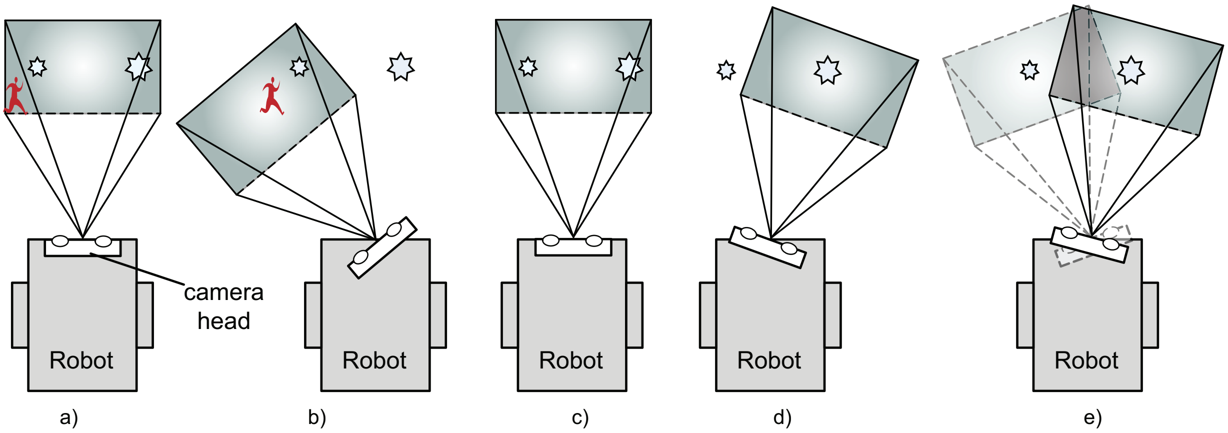

Fig. 1.2: Various aspects of robot attention behaviors: a) initial state; b) attending to a task-irrelevant (implicitly task-relevant) stimulus; c) promoting task-relevant information; d) attending to one of task-relevant objects; e) temporal attention selection of task-relevant objects. Stars: task-relevant information/objects; human-shaped symbols: task-irrelevant stimuli.

objects, or abrupt variation of object appearance. This stimuli-driven bottom-up perception means a lot to an operating robot for its own and users' safety, and contributes to an adaptive task accomplishment as well as a complete environment modeling. Therefore, it can also be regarded as an implicitly task-relevant aspect.

Two essential questions arise in this context. The first one is how to define stimuli with respect to robot applications. Most works consider bottom-up stimuli from a static perspective at a given time point. However, temporal novelty is a more interesting perspective for an uncertain environment than static saliency. Furthermore, continuous or random attending to task-irrelevant stimuli is ineffective in terms of task accomplishment. Therefore, the second question is how to select the best moment to perceive task-irrelevant stimuli, in order to avoid severe loss of task-relevant information. A stimulus-dependent property of robot attention control should be explored.

How to promote task-relevant information

Considering robot applications, an efficient enhancement and prediction of task-relevant visual information in the early vision processing are the most goal-directed improvements to a robot system that attention can provide. Thereby, top-down information is integrated into bottom-up attention processing to replace the conventional exhaustive search in the large amount of visual information. This integration is commonly conducted by applying a previous offline training to find a best representation of a target object in terms of low-level features, which fails, if appropriate top-down information is not available.

The environment changing due to robot mobility is another big issue, since the top-down information of one target object resulting from an offline training is inflexible. For mobile robots, a flexible and efficient environment adaptation of top-down information enhancement is envisioned.

In addition, for a complete robot system, the robot attention behavior dealing with task-relevant information may vary in different internal robot states. Decision making in searching for and operating more than one object should be considered, which is also a key issue for robot autonomy.

How to plan robot attention when facing multiple task-relevant objects

For a robot system with a limited field of view (FOV) in an application scenario containing multiple task-relevant objects, in a multi-robot system for instance, attention selection in a temporal sense strongly influences the evaluation of task performance. Conventional attention selection has only been considered in the 2D image space. A spatio-temporal robot attention control has been accomplished in a way that a sequential scan path is determined to process the information with a higher priority first. The priority is usually decided using 2D appearance in visual data input, which, however, cannot be always consistent with the task relevance. A task-oriented quantitative evaluation of robot attention behavior in the 3D task space is still missing.

1.2 Main Contributions and Outline of the Thesis

In this thesis, various aspects of robot visual attention according to the aforementioned challenges are explored, in which a general and integrated attention control concept is developed. Fig. 1.3 illustrates the outline of the thesis. The information contained in visual data input can be classified into three categories: “task-irrelevant” stimuli, task-relevant information, and task-irrelevant non-stimuli such as background, etc. Above all, attention distribution on task-relevant information can be studied from a spatial and a temporal point of view. After the state of the art is surveyed extensively in Chapter 2, Chapters 3, 4 and 5 investigate robot attention control from the stimulus-dependent aspect, the task-relevant spatial aspect, and the task-relevant temporal aspect, respectively.

Stimulus-Dependent Aspect: Bottom-Up Perception Considering Environment Dynamics

To deal with uncertainty caused by environment dynamics, a mobile robot should be endowed with the ability to be aware of the environment changing while performing its task. Since this kind of environment changing is not always explicitly correlated with the current robot task, it is considered in bottom-up perception. Inspired by the expectation-based perception of humans, two metrics are proposed: *local surprise* and *global surprise*. Local surprise is defined as Bayesian surprise of two consecutive saliency maps. A maximum local surprise in a 2D image indicates a large temporal novelty and/or a large spatial saliency in comparison to the other image regions. Since consistent attending towards local surprise is ineffective for robot tasks, global surprise is defined to represent the current environment dynamics and used to alert the robot system reasonably and economically when a shift of attention onto local surprise is necessary. In Chapter 3, a surprise-driven vision system is

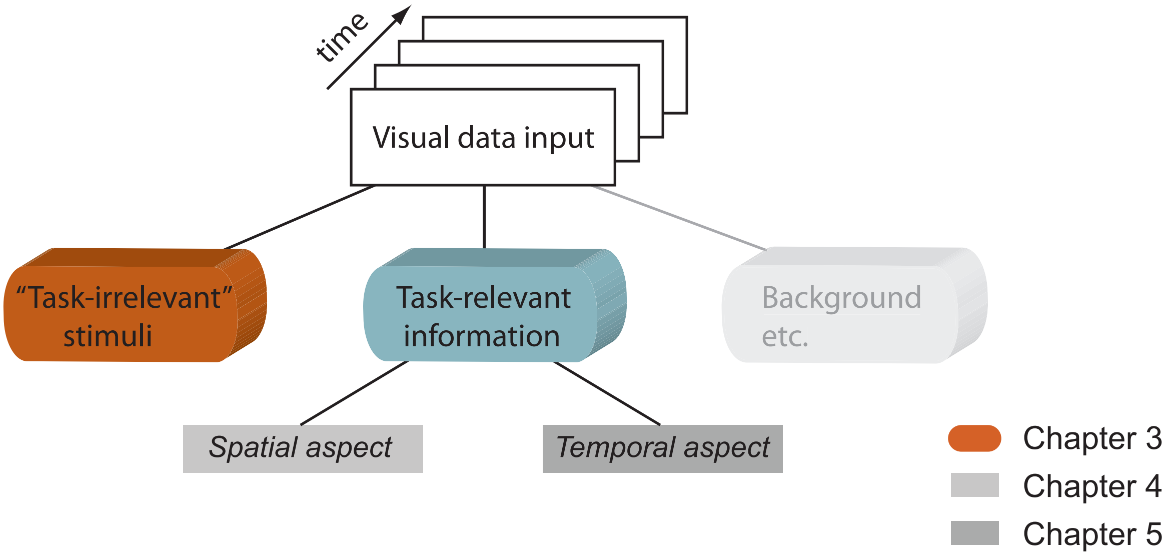

Fig. 1.3: Outline of the thesis.

described based on the interconnection of local surprise and global surprise. A significant extension and improvement of robotic perception and cognition in terms of high sensitivity is realized.

Spatial Aspect: Combination of Top-Down and Bottom-Up Attention Control for Task-Relevant Information

For an efficient promotion of task-relevant information, the recent tendency is to use top-down information to bias bottom-up attention selection. In order to overcome the challenges such as envisioned flexibility, adaptation to changing environment, and autonomous task changing, two complementary visual attention selection strategies for the detection of task-relevant objects are proposed in Chapter 4, in which the combination and coordination of top-down and bottom-up mechanisms are explored. The first one is a variation of top-down biased bottom-up attention selection, considering target objects with similar appearance and changing backgrounds due to robot locomotion. The conventional offline training of task-relevant information is replaced by an online extraction of top-down information of the first recognized target object. Successively, adaptation of model parameters on environments using a Kalman-filter (KF) is developed, which manifests itself in an improved efficiency in terms of fewer necessary fixations. The second one is an autonomous switching between top-down and bottom-up attention selection, which fills the gap in the first combination strategy for the situation where totally different targets are searched for while contexts vary. This application oriented robot attention system makes a further step towards efficient visual information selection and cognitive visual behavior planning in the robotics domain in terms of efficiency, flexibility, and autonomy.

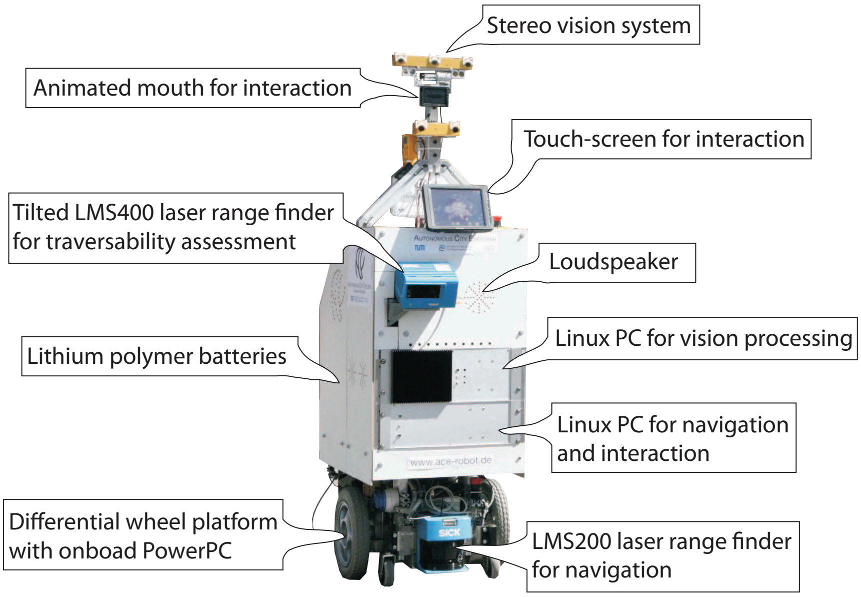
Stereo vision system
Animated mouth for interaction
Touch-screen for interaction
Tilted LMS400 laser range finder for traversability assessment
Loudspeaker
Lithium polymer batteries
Linux PC for vision processing
Linux PC for navigation and interaction
Differential wheel platform with onboad PowerPC
LMS200 laser range finder for navigation

Fig. 1.4: The Autonomous City Explorer robot [213]

Temporal Aspect: Human-Inspired Temporal Attention Control When Facing Multiple Task-Relevant Objects

After task-relevant information is enhanced in the 2D image space in Chapter 4, Chapter 5 addresses the temporal planning issue of robot attention control. Conventional visual scan paths are conducted according to saliency value indicating relative importance of the salient objects in a Winner-Take-All (WTA) manner, not considering more than one target with the same importance to tasks. In this chapter, an optimal fixation sequence is to be determined in terms of task-relevant quantitative evaluation in 3D task space including how long the current focus of attention (FOA) is to be fixated and which task-relevant environment fraction is selected for the next fixation. Here, an experimental investigation of human behavior is conducted, to study how humans behave (gaze and body) in the state with or without dominant intent when they are facing more than one target object. Human eye movement and body movement are recorded and analyzed. Inspired by the experimental results, a temporal attention planning algorithm for visual sensor with limited FOVs in single- and multi-robot systems is proposed in Chapter 5, achieving significant improvements in reduced perception uncertainty and extended FOV.

High-Speed Implementations on an Autonomous Mobile Robot

A vision-guided mobile robot, the Autonomous City Explorer (ACE) developed at the Institute of Automatic Control Engineering of the Technische Universität München (see Fig. 1.4), was used to demonstrate the strategies and evaluate the performance experimentally. The ACE robot is equipped with a high-performance active multi-focal camera system, which can be used to resemble visual behaviors such as scan, saccade and fixation. Details about ACE can be found in Appendix A.

In dynamic robot vision, high-speed processing of early vision can enable high-speed perception and recognition of sudden events, which reduces the overall latency of image processing and ensures real-time decision making. Another practical advantage of a high-speed image processing is to reduce the influence of inter-frame motion such that the motion blur or the ego-motion component can be ignored in computation. Bottom-up attention selection is implemented on a platform containing multiple Graphics Processing Units (GPUs) to significantly accelerate the compute-intensive but highly parallelizable computation of bottom-up attention. In this implementation, the Compute Unified Device Architecture (CUDA) technology is used. The implementation details can be found in Appendix B.

The various aspects addressed in this thesis contribute to a robot-centered visual attention system. The objective is to bridge the gap between fundamental studies in cognitive psychology/neuroscience and technical realizations in the robotics domain by developing biologically plausible and technically applicable robot attention strategies. A variety of applications and examples are presented to highlight the integrated and applicable characteristics of the proposed robot attention control concept.

2 Related Work

Attention is a general term for selectivity in perception, which is important for selecting and inhibiting visual information over space and over time [34]. It plays an essential role in perception and cognition and has already been studied intensively in the cognitive psychology and neuroscience area. However, visual attention selection is a relatively young research area in the robotics domain. Since the topic of this thesis is an interdisciplinary problem, it is considered how the findings from the fundamental studies can be mapped into the robotics domain and facilitate robot task performance.

In this chapter related works around visual attention are reviewed in four different parts. The first part gives a brief introduction to the terminology of human eyes. Then, an overview about the fundamental theoretical findings in cognitive psychology and neuroscience is given. The third section reviews computational models of visual attention selection. Finally, technical realizations in the computer vision and robotic systems are surveyed from various application aspects. More specific methodological introductions of the state-of-the-art approaches strictly related to the concepts in this thesis are given and discussed in the following three chapters.

2.1 Biological Terminology

Firstly, a brief introduction to the biological system is given, taking human eyes as an example. Definitions of some terms related to this thesis are summarized.

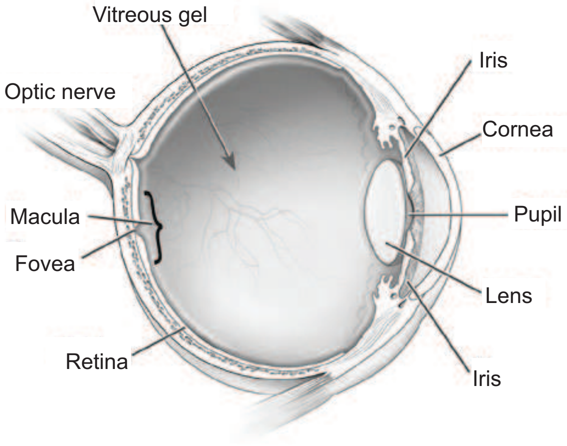

Fig. 2.1: Human eye diagram. Source: NEI Catalog number NEA09, National Eye Institute (URL: www.nei.nih.gov).

Eyes are organs that provide visual information of the environment to the brain. The lights reflected from the surroundings are refracted by the *lens* and imaged on the *retina*. The retina photo-receptors provoke nerve impulses which are transmitted by the *optic nerve* from the retina to the brain (see Fig. 2.1).

The small central area of the retina is called *macula*. The central pit in the macula which provides the sharpest vision is called *fovea*, while the vision extracted from the stimuli perceived by other retinal areas is named *peripheral vision* [30]. The foveal vision with a high visual acuity (resolution) is sensitive to color and shape, while the peripheral vision with a low visual acuity has a better ability to detect motion [76].

Due to the limited processing capacity in biological systems, an attention mechanism is deployed to determine "where-to-look", such that the most interesting parts of the surroundings can be projected on to the fovea and can be processed at a high resolution. Redirecting attention is realized by shifting of attention, which can be *overt* or *covert*. The overt attention indicates attention shifts with the eye moving, while the covert attention indicates attention shifts with the eye remaining fixated [204]. In this thesis, overt attention accompanied by eye saccades is mainly focused on.

2.2 Fundamental Theories in Cognitive Psychology and Neuroscience

In 1890 a principle was proposed, suggesting that two factors determine the distribution of visual attention: the properties of the image and the goals and expectations of the observer [86]. The former is today called *bottom-up* or *stimulus-driven* attention selection, while the latter is called *top-down* or *goal-directed* attention selection. Fundamental theories have studied how both attention selection mechanisms perform and interplay to guide human visual attention. In those works, *visual search* is a conventional experimental paradigm, in which the reaction time is investigated when subjects search for a target item among various distractor items. Three representative theories are introduced below, serving as foundations for many computational models of attention selection.

Treisman's Feature Integration Theory (1980)

Feature Integration Theory (FIT) [185] suggests two different kinds of visual search: feature search and conjunction search. Features are assumed to exhibit activations on specific retinotopic feature maps. If the target item owns a unique feature in comparison to the other distractors, feature search occurs, which performs a parallel and fast visual search. If the target item cannot be distinguished based on a unique feature but based on a conjunction of features, attention must be directed serially to each stimulus, in order to characterize objects. Therefore, conjunction search is more costly than feature search.

Duncan & Humphreys's Attentional Engagement Theory (1989)

Another theory of search and visual attention, the *Attentional Engagement Theory* [45], argues that visual search efficiency is not just subject to parallel or serial search. The difficulty of visual search is a continuum that increases with increased similarity of targets to distractors. The larger the difference between the target and the distractors is, the more efficiently the visual search is. Moreover, the more heterogenous the distractors are, the harder visual search performs. This theory supports perceptual grouping and parallel display segmentation and serves as foundations for object-based attention models.

Wolfe's Guided Search Theory (1990)

Guided search theory [34, 201, 202] explores how attention can be guided and be more efficient due to limited resources. Like FIT proposed in [185], the original model also has a pre-attentive stage executing a parallel process and a following attentive stage deploying a serial process. The contribution of this model is to show that the attentional deployment of limited resources is guided by the output of the earlier parallel processes. The feature maps in the parallel stage are combined into a general activation map. The item with higher activation value is attended to firstly. Here, a weighted sum of top-down and bottom-up components for attention control becomes possible.

Other Findings of Visual Attention

– Bottom-Up Attention First of all, bottom-up visual attention is attracted by salient stimuli that pop out from their surroundings which are potentially important to observers. It is shown in [36] that focal attention has a strong spatial component at the physiological level. The attentional response enhancement extends to behaviorally irrelevant objects around the target object. In addition, attention involves a complex modulation of responses to other stimuli in the surrounding visual space. Different cues and binding problems influencing visual search are experimentally investigated [65, 182], while the influence of distracters has also been considered [70]. Unexpected events are also proved to be an essential factor for attention control [87, 131].

– Top-Down Attention Moreover, visual attention is also guided to task-relevant information which is currently important to the observers. In [33] shows that top-down perceptual knowledge limits expectations and guides the eye in deciding where to attend. Thereby, the learned associations between novel visual shapes and regularities in dynamic visual environments facilitate search behavior.

– Top-Down Biased Bottom-Up Attention Recent neurophysiological experiments explore the biasing effect of top-down information on bottom-up visual search and show that attention sometimes appears as a non-linear property that results from a top-down biasing effect. A dynamical analysis of this biased competition and cooperation of neuronal spiking mechanisms is made in [42], in which the interaction between top-down attention and bottom-up stimulus contrast effects is modeled. It is shown that top-down attentional effects bias neurons by changing their nonlinear activation functions. A review of the interconnection of top-down and bottom-up is given in [93].

2.3 Computational Models of Visual Attention

Originating in computational neuroscience, various attention selection models have been proposed to resemble and implement human visual attention distribution. Above all, attention models combining and coordinating top-down and bottom-up attention selection mechanisms vary in different aspects: static vs. dynamic, space-based vs. object-based, bottom-up vs. top-down, feature saliency vs. information representation, etc. A brief overview of some representative attention models is given here.

Koch & Ullman (1985) The rudiment of many feature-based attention models has been proposed in [94]. They suggest that elementary features are processed in parallel in different topographical maps during an early representation. Locations with outstanding features with respect to their neighborhood are encoded in these feature maps. Then, the feature maps are combined in a central saliency map, representing the relative conspicuity of each location. A WTA network selects the most salient location. Accompanied by an *Inhibition-Of-Return* (IOR) mechanism, WTA guides the attention to shift to the next most salient location.

Tsotsos et al. (1990) The central thesis of [186, 187] is that attention acts to optimize the search procedure inherent in a solution to vision, stated by Tsotsos et al. The primate visual attention performance is computationally explained by using the concept of selective tuning of visual processing network: Based on the divergent feed-forward pathways activated by stimuli in the visual field, task guidance or bias is selected at the top level of the processing architecture (a processing pyramid). Then, the competition losers are inhibited by the feed-back pathways. Features such as luminance, orientation, and opponent colors have been implemented.

Milanese (1993) Based on [94], one of the earliest implementations of visual attention model is proposed in [125, 126]. Three subsystems are contained: a bottom-up subsystem, an alerting subsystem and a top-down subsystem. In the bottom-up subsystem, feature maps and conspicuity maps considering orientation, curvature, and color contrast difference with respect to pixels surroundings are computed and combined into a saliency map. Moreover, an object moving against a static background is integrated in the alerting subsystem, which can control attention movement. Top-down cues are analyzed in *Distributed Associated Memories* (DAM) through previous trainings and used in object recognition.

Itti et al. (1998) Based on [94], one of the standard bottom-up computational attention models is proposed in [84]. In this model, an input image is sub-sampled into dyadic Gaussian pyramids in three channels (intensity, orientation, and opponent color). Then, center-surround differences are calculated for the images in the Gaussian pyramids between the fine scales and the coarse scales. In this phase, feature maps (FM) are generated in which the salient pixels with respect to their neighborhood are highlighted. Using across-scale combinations, the FMs are combined and normalized into a conspicuity map (CM) in each channel. CMs are combined linearly into a final map, in which the bright pixels are the salient and interesting pixels with respect to their backgrounds. The so-called *saliency map model* is widely used in many research groups as reference and serves as a basis for many visual stimulus-driven attention control algorithms. Moreover, a Bayesian definition of surprise is proposed in [79], in order to combine temporal novelty and spatial saliency. Further integration of top-down bias, extensions, and detailed evaluations are also introduced in [83, 134]. Since both the saliency map model and the surprise model are relevant for this thesis, a detailed description will be given in Chapter 3.

Hamker (2000) The bottom-up part of the attention model proposed in [67, 68] is similar to the saliency map model in [84]. Top-down factors are however also considered here. The system can learn feature values of a stimulus and memorize them in a working memory. Moreover, *match detection units* are applied on the fixation candidate region to compare this region with the target template pattern and determine if an eye movement towards this region is needed.

Sun & Fisher (2003) A hierarchical object-based visual attention model is presented in [179], consisting of two new mechanisms: grouping-based visual salience computation of objects and hierarchical selectivity of attentional shifts. The first one is responsible for attentional competition among features, objects, or groupings of features and objects. In this process, object-based and feature-based visual attention are combined. The second mechanism is used to guide covert attentional movements, considering spatial locations, features, objects, and their conjunctions.

Backer & Mertsching (2003) Considering dynamic vision, another novel object-based model for efficient attention selection is introduced in [10]. A semi-attentive stage is imported between the pre-attentive stage and the attentive stage. A symbolic representation of each selected item is generated in which objects are labeled with an object file containing position, size, trajectories, selection histories, feature values, and the result of object recognition. *Dynamic neural fields* are used for multiple objects selection and tracking.

Ouerhani & Hügli (2003) In the attention model implemented in [138], a saliency map related to static features and a saliency map related to dynamic scene features are weightedly combined into a final saliency map to predict stimuli locations. Static features used are opponent colors and intensity, while they use optical flow at different scales to compute motion component. The most salient point is then found and tracked.

Frintrop (2005) A computational attention system *VOCUS* is proposed in [53], improving the saliency map model of [84] by merging top-down and bottom-up attention into a single system. From feature maps in intensity, color, and orientation, a bottom-up map and a top-down map are computed and integrated into a global saliency map by a linear weighting. The top-down map is a combination of an excitation map and an inhibition map weighted by known target information. The system is able to select regions of interest in a bottom-up way and detect predefined target objects in a top-down manner.

Gao & Vasconcelos (2005) A hypothesis that all saliency decisions are optimal in a decision-theoretic sense is suggested in [59, 60], denoted as *discriminant saliency*, which means minimum probability of error. Two classes of stimuli are defined: stimuli of interest and null hypothesis, the latter consisting all the non-salient stimuli. The computational measure for saliency is defined as the mutual information between features and the class label indicating one of the classes. The consistency with psychophysics and the plausibility of this simple and generic definition of saliency detectors are proved in features such as color, intensity, orientation, and motion field as well.

Bruce & Tsotsos (2006) In [25, 26], Attention based on Information Maximization (AIM) is proposed using information theoretic formulation in attention modeling. They claim that the localized saliency computation serves to maximize information of the environment. An information measure is defined as an estimate of the likelihood of content within a central patch on the basis of its surroundings. The results show that AIM behaves more like the human visual system than the model proposed by [84].

Zhang et al. (2008) Another information-based approach for saliency evaluation is presented in [212]. Two measures are defined: the self-information for the bottom-up saliency and pointwise mutual information between the features and the target for overall saliency, namely top-down biased bottom-up saliency. People's fixations in free viewing natural images are well predicted using this efficient algorithm with few free parameters.

Other attention models with minor differences to the aforementioned models can be found in [85, 107, 118]. Above all, the robotic attention concept proposed in this thesis is principally based on [84] and [79], which is not only based on primary features in a static image, but also based on information variation in an image sequence.

2.4 Technical Realizations in the Robotics Domain

Most traditional robotic vision systems apply a task-oriented information selection, given the predefined task-relevant features in 2D images such as color, geometry, motion, etc. An exhaustive search or search in a defined search window is conducted to find the current target object. Due to the large amount of visual information, visual attention has recently been playing an essential roll in robot vision. Enhancing task-relevant visual information while inhibiting the others, visual attention endows robots with the ability to process information quickly and efficiently with limited resources. Robotic applications of visual attention are reviewed below.

Demonstration of Visual Interest

One of the earliest implementations of visual attention on robots is introduced in [164]. A camera is mounted passively on a mobile robot. A segregation of visual stimuli based on *connectionist model* by means of synchronization of spiking neurons is used to bind image features corresponding to objects. Then, the largest one of the segregated objects, is selected and approached by the robot. Although only edge features are used, this system exhibits a primary version of visual attention of mobile robots.

Visual attention for eye and head animation of a realistic virtue human head is applied in [81], using an extended version of the neurobiological model proposed in [84]. In this animation, flicker features and motion features are included as well to deal with the temporal changes and moving objects. Moreover, coordination of eye and head movement is also concerned to achieve a realistic animation and rendering.

In [163, 194], a saliency-driven vision system is also applied to a robot head, which uses a bottom-up visual attention mechanism to focus on interesting objects in the environment in real time and attends to them.

In [13], a biologically motivated saliency map model based on a stereo saliency map is presented. Two bottom-up static saliency maps are first separately computed on two monocular cameras. Then, top-down information of human-like preference and refusal trained by a fuzzy ART network supervised by an interacting human is applied. The final saliency maps are used to compute the depth information for vergence control such that the vision system attends to the closest objects.

Active Visual Attention and Gaze Control

In [123], a neural active vision system is proposed which explores the environment using an attention model considering symmetry, eccentricity, and color contrast to locate interesting objects and recognize them. Gaze control is conducted differently in different modes: In a hypothesis generation mode, an inhibition map is brought into consideration to inhibit attending to an already fixated image region. In a hypothesis validation mode, an excitation map is also considered, in order to highlight the points of a generated object hypothesis. If a moving object is detected, the gaze control is converted into a tracking mode.

A few active vision systems in the context of humanoid development have been implemented. In [177], a generic, real-time scalable visual attention system is built. A number of human visual attributes such as log-polar mapping, feature maps, and *featureGate* are considered for the system to determine fixation points. Another overt attention system based on visual flow for a humanoid robot is proposed in [195]. In [147], a humanoid robot is demonstrated, which is able to grasp a waving object using a biologically inspired active vision system. Four basic human-like visual behaviors are implemented: saccade, smooth pursuit, vergence, and vestibulo-ocular reflex. A sensory processing module processing bottom-up and top-down signals to create a saliency map for fixation guidance, a motion planning module driving oculomotor system, and an interaction issues module dealing with robot self-motion are contained. Various features such as color, intensity, edges, stereo, and motion are used in [189] to drive the gaze of a humanoid head toward potential regions of interest, where a distributed implementation on a computer cluster ensures the real-time ability of the multi-focal vision system containing foveal vision and peripheral vision.

In [18], another multi-focal camera system is presented, consisting of foveal vision and peripheral vision. This system is able to locate and recognize objects in the real world using the top-down object characteristics: hue saliency and 3D size. The attentional process is performed in a relatively wide FOV, while recognition is conducted in the high-resolution foveal center.

Facilitation in Computer Vision

– Object Detection in the 2D Image Space Visual attention is commonly used as a front-end for object recognition to reduce computational cost. Bottom-up attention provides a prediction of potential location of target objects. Then, further processing can be applied in the pre-selected image regions to verify the existence of the target.

In [197], a marriage between bottom-up attention based on a saliency map and *Scale Invariant Feature Transform* (SIFT) feature-based object recognition is applied to demonstrate that bottom-up attention can contribute to object detection and reduce computation time. This paper serves as a basis for attention-based object detection. However, as discussed in this paper, many points regarding a complete detection system should be improved, for instance, top-down feedbacks and foveated vision. In [196] a salient proto-object detection model is suggested, where hierarchical models of object recognition in cortex based on Max-like operation on the inputs to certain cortical neurons are used, such that the objects are attended to before recognized. In [135], target detection speed is maximized, defined as the ratio between the strength of the signal detecting the target over that detecting the distracting background, such that the weights between top-down and bottom-up attentional influences are optimized.

In [43], a Toolkit “SAFE” is proposed to deal with the problem of invariance of 2D similarity transformation in the selective attention system, the Neuromorphic Vision Toolkit (NVT) developed at CalTech and USC (see ilab.usc.edu/bu). This improvement enables SAFE to be more suitable for an object recognition system.

In contrast to feature-based attentional object detection, an object recognition approach based on an information theoretic saliency measure is proposed in [57]. Local saliency is determined by information content using entropy computation and used to model sparse object representation.

An approach combining visual attention and a cascade classifier is proposed in [128] for ball recognition. The classifier is trained previously and used directly on input images to find a ball. Then the feature weights of the detected ball are learned by the attention system to predict regions of interest. Afterwards, the classifier is only applied in the regions of interest to avoid false positives.

In [46], a robot explores the environment, discovers actively controllable perceptual categories such as its hand or fingers, and adapts its controllers to place the hand within the visual field or uses arm information to predict 2D fingertip location. A visual attention system is used here to select a few salient image patches corresponding to fast moving regions for further hand detection.

In addition, an attention system as a front-end for image processing such as scene decomposition or traffic sign detection in a driving assistance system is introduced in [124].

– Attentional Face Detection Specifically, biologically inspired attention models also benefit human face detection. Face detection is accomplished in two steps in [14]. In the first step, a saliency map is computed considering face-related features such as color opponent, edge, intensity, and symmetry information. Then, face selection is conducted on the saliency map by using an auto associative multilayer perception model to classify face and non-face. In [171], instead of searching for faces in an exhaustive way, face part detection is conducted on a saliency map at a certain scale, which is predicted by the image gist computed by using a Discrete Fourier Transform of the whole image. Starting from the most salient position, some local salient points are selected to be candidates of face parts, until a face is declared to be detected. Using this approach, computational cost is reduced significantly.

– Object Segmentation A segmentation approach using visual attention is proposed in [69]. The bottom-up saliency map computation proposed in [84] is used firstly to encode the attention value. Then, a few salient locations are regarded as attention seeds. A *Markov random field* model is used to grow the attention objects from those attention seeds and extract attention objects without the need to understand the image semantic or help from an interacting user.

– Object Tracking Aided by visual attention, object tracking can be conducted before an object is precisely recognized [10, 55, 123, 138]. Objects are now labeled by primitive features obtained by a previous training or an online selection and pop out from bottom-up or top-down biased bottom-up saliency maps.

– Object Detection in the 3D Task Space To solve the problem of visual search for a given target in an arbitrary 3D space for robot vision systems, the probability of finding the target is optimized in [188], given a fixed cost limit in terms of total number of robotic actions the robot requires to find its visual target, facilitated by attentive processes.

Action/Intention Understanding

In [74], a pointing gesture recognition is conducted using a bottom-up feature map based on entropy, symmetry, corners, as well as skin color, and a top-down propagated recognition based on a pointing gesture classifier. Then, the 2D pointing angle is estimated, through which the pointed object is recognized.

The saliency of top-down elements and the saliency of bottom-up components are combined in [90], in a way that the top-down part is initialized by the bottom-up part, hence resulting in a selection of the behaviors that rightly require the limited computational resources.

In developmental robotics, attention is also essential for intention prediction and recognition [62, 66, 95, 132, 207], in order to establish a joint attention between communicating agents and imitations.

Context Guided Robot Attention

Context guided visual attention is also beneficial for robotic applications, especially for object detection tasks in terms of reduction of false positives and improvement of efficiency.

Bottom-up saliency, scene context, and top-down mechanisms are combined by the contextual guidance model of attention in [183, 184] at an early stage of visual processing, which profits object search in real-world scenes.

Similarly, in [16] the COBA (COntext-BAsed) model of attention is proposed, in which the spatial context of an object is important for its localization. In experiments for face detection in natural images, false positives are reduced.

Since the spatial contextual information is essential for robot attention selection, various algorithms are proposed for complex scene understanding and categories using "gist" [172], Bayesian network and SIFT [77], Bayesian hierarchical model [106], spatial pyramid matching [102], human action analysis [127], informative features for city-scale location recognition [166], and Hidden Markov Model [170].

In [153], the weights of top-down and bottom-up factors are combined, in which an offline optimization of the top-down weighting and a context learning based on a neural network are conducted using a large set of examples. The balance between top-down and bottom-up components is not fixed, but influenced by a simple context vector, which improves object searching tasks.

Assistance for Robot Navigation

In comparison to conventional landmark selection in the robot localization and navigation tasks, more and more applications consider visual attention approaches, in order to select landmarks more naturally, robustly, and efficiently.

For a reliable vision-based control of an autonomous vehicle, a saliency map, which is based upon a computed expectation of the input contents at the next time step, is used to emphasize the important task-relevant features in [12].

Information sampling is used in [200] to select the most interesting image data representing highly attentive regions of the environment. Based on that, the qualitative position of the robot in a topological context can be determined.

In [129], a visual attention control algorithm for a legged mobile robot is proposed. It functions by observing the direction which has the largest expected information gain calculated by a decision tree constructed using information gain by time, and the compensation mechanism for walking and locomotion.

Robot self-localization using visual attention is also introduced in [139], in which the robustness of the selected salient landmarks is evaluated during navigation. Only the most robust features remain as landmarks.

In [53], a biologically motivated attention system is proposed for landmark selection of visual simultaneous localization and mapping (SLAM). In the selected region of interest, Harris corners are detected. The re-detectability of the selected regions and the stability of Harris corners are combined, enhancing the whole task performance. Moreover, active gaze control for visual SLAM using features detected by an attention system is applied in [54], which supports the system with tracking, re-detection, and exploration behaviors.

Robot navigation based on active multi-focal vision is elaborated in [96], where the localization accuracy, events or objects of interest to be observed and the predicted visibility of objects are considered. High accuracy and large FOV are combined using this novel multi-focal view direction planning strategy.

Behavior-Based Robot Systems

In behavior-based robot systems, robot attention is commonly adapted to the predefined robot behaviors such as top-down or bottom-up attention selection, searching for different target objects in different behaviors, as well as attention enabled or inhibited processes.

In [180], a *recurrent neural network* (RNN) is used to learn the sequence of events encountered during navigation and to make predictions for the future. Attention between object recognition and wall-following tasks is switched by the top-down prediction made by the RNN.

Another visual attention system, *VOCUS*, is proposed in [53] for object detection and goal-directed search. This system can detect regions of interest in images in an exploration mode with no specified target and can search for a specific target using top-down information obtained from previous training process as well.

In [209], a task-driven object-based visual attention model for robot applications is proposed, which involves five components: pre-attentive object-based segmentation, bottom-up still attention, bottom-up motion attention, top-down object-based biasing, and contour-based object representation. Task-specific moving object detection and still object detection are operated based on this model.

A highly competent object recognition system on a mobile robot is proposed in [51], which is capable of locating numerous challenging objects amongst distracters. The potential objects are ranked using a bag-of-features technique and identified using an attention mechanism in a limited time. Three visual behaviors are defined: exploration behavior, coverage behavior, and view point selection behavior. The first behavior is more a robot exploration behavior in terms of path planning than a visual behavior. In the second behavior, the potential objects are scanned by the peripheral vision. After the environment is fully covered, novel perspectives of the objects are captured and object recognition is conducted in view point selection behavior.

Human-Robot Interaction

A context-dependent attention system for a social robot is proposed in [23]. This attention system integrates perceptions (motion detection, color saliency, and face pop-out) with habituation effects and influences from the robot's motivational and behavioral state to create a context-dependent attention activation map, which is used to direct eye movements. Using an image size of 64 × 64 pixels, the processing is in real-time.

An epigenetic model is proposed, consisting of the acquisition of intentionality, identification, and social communication. To recognize human intentions, robots are able to detect human eyes and track human gaze [207].

In [178], human-robot interaction is simulated for learning a sensorimotor map for joint attention, investigating the causality inherent in face-to-face interaction quantified by transfer entropy.

Multi-Modal Attention Systems

Multi-modal attention systems have been proposed in [27, 56, 121, 161]. Various sensor modalities such as vision, haptic sensor, lasers, sonars, and auditory sensors are applied and integrated to decide on the current attention of the robots. Most systems imitate human behavior and the objective is to achieve a cognitive and natural interaction between human and the robots. Attention architectures for machine vision and mobile robots with an emphasis on multi-modal information fusion are reviewed in [143].

Multi-Robot Systems

Visual attention is also a key point in multi-robot interaction [89]. A two-robot cooperation scenario is investigated in [58], in which one robot observes the hand movement of the other robot, who is going to manipulate an object, and try to recognize how the other robot will grasp the target object. However, the intention recognition is not autonomously started.

High-Speed Implementations of Visual Attention

A high-speed implementation of the early vision means a lot to robotics applications. Therefore, researchers are making great efforts to achieve this using various hardwares, algorithms, or implementation structures [109, 120, 140, 148, 189, 203]. A detailed survey can be found in Appendix B.

2.5 Summary

The state-of-the-art visual attention research can be mainly summarized into two different categories.

In the first category, researchers from biology, cognitive psychology, and neurosience backgrounds are aiming at developing a human-like attention selection model and normally focusing on static images or video clips. The models are sophisticated, compute-intensive, and sometimes not appropriate for robot applications. Many implications and findings from their perspective cannot be easily transferred and implemented. Only few works consider robot task-oriented evaluations.

In the second category, researchers from the computer vision and the robotics domain are aiming at developing application-oriented attention systems for robotic applications. However, most works are too specific and not generalizable. Generally, higher-level functions are missing. Most works have not considered implicitly task-relevant factors such as unexpected events and environment safety, while only few works have considered mobile robot applications including locomotion.

Although visual attention has been intensively studied in the fundamental research, it is still a young research area in the robotics domain. Many aspects of robotic vision have not been comprehensively and integratedly considered, such as robot mobility, adaptability, flexibility, reactivity, and applicability. Task- or application-oriented performance evaluation of robot attention is limited. Therefore, the objective of this thesis is to map the cognitive neuroscientific models onto robot visual attention design and to develop biologically plausible and technically applicable visual attention strategies for mobile robots, bridging the gap between these two categories.

3 Information-Based Bottom-Up Perception for Attention Control

3.1 Introduction

Visual perception is the ability to obtain, represent, and interpret visual sensor information about the external environment. The starting point of this process in a robot system is the input images captured by visual sensors – cameras. From visual data input, a large amount of information about the robot's operating environment is obtained. Conventional robot applications have only considered the extraction of task-relevant information, while stimulus-dependent perception has been ignored. However, due to robot mobility and environment dynamics, task-irrelevant stimuli such as abrupt appearing/disappearing of objects (e.g. a person entering a room), dynamic characteristics (e.g. a car passing by) or appearance variation of objects (e.g. changing neon lights or billboards), etc, can influence robot task accomplishment strongly, increasing environment uncertainty or even inhibiting robot tasks. Although they are not directly related to robot tasks, they definitely play a key role in cognitive technical systems. Being sensitive to unexpected stimuli is a critical problem for task-driven systems [87]. Moreover, if a robot is accomplishing a task, when and how should the attention be modified, scheduled, or distributed to the unexpected events that will obviously impair the on-going task?

In this chapter, two metrics for image sequences understanding are proposed – *local surprise* (LS) and *global surprise* (GS) – to describe dynamic characteristics of an image sequence and provide a more sensitive cue for robot attention control in dynamic environments. The concept of *surprise* defined in [79, 80] is borrowed, which measures the distance between the prior belief and the posterior belief of the environment computed by Kullback-Leibler (KL) divergence of the posterior and prior distributions.

LS and GS are complementary to each other. LS is computed to detect unexpected stimuli in the environment which provides a possible front-end for further recognition of the stimuli and avoids information loss, while GS is used to determine when a robot should attend to the unexpected stimuli during its on-going task and provide an economical timing to take a saccade behavior towards the unexpected stimuli. Combining LS and GS, a surprise-driven active vision system is developed based on the high-performance implementation of bottom-up attention. A perception-decision-action loop is conducted.

The remainder of this chapter is organized as follows. In Sections 3.2 and 3.3 LS and GS are defined, respectively. Relevant experiments are conducted and discussed. A surprise-driven robot attention control using the active vision system based on the interconnection of LS and GS is proposed in Section 3.4. A summary containing discussions of the contributions and limitations is given in Section 3.5.

3.2 Local Surprise for Unexpected Stimuli

Spatial salient stimuli have already been considered in bottom-up attention selection models for a human-like prediction of attentional allocation in natural scenes [84]. In dynamic environments, temporal novelty plays a more important role for robot applications. LS is proposed as a measure of unexpected stimuli, which is a combination of the spatial saliency and the temporal novelty.

3.2.1 State of the Art

To integrate unexpected stimuli, especially dynamic variations, into attention selection, the intuitive approach is to integrate motion detectors into saliency map computation [15, 29, 38, 103, 117, 209]. However, appropriate weighting between a static saliency map and a motion map has not yet been explored.

Bayesian modelling of visual perception has been proved to be a convenient and natural framework to study perceptual decision, consisting of the task, prior knowledge about the environment, and knowledge of the way the environment is sensed [49, 116, 152, 210]. Applications in video compression using Bayesian foveation [20], image change detection [114], and landmark detection [151] based on Bayesian theorem have been proposed.

The definition of *Bayesian surprise* was firstly proposed in [79, 80] by Itti & Baldi to detect temporal novelty in video clips. Surprise is defined as the distance between the posterior and prior belief distributions about the environment computed by using the relative entropy or Kullback-Leibler divergence. Surprise measures how data affects an observer, in terms of difference between posterior and prior beliefs about the world. The hypothesis that Bayesian surprise attracts human attention in dynamic natural scenes is proved by comparing Bayesian surprise locations with human saccades.

Itti & Baldi's surprise model is aiming at a best explanation of human gaze behavior. Each feature map according to color, orientation, intensity, flicker, or motion at different scales is considered to contain surprise detectors. In this thesis, the focus is to detect the unexpected event/stimulus such as abrupt appearing, disappearing, motion, and appearance variation of foreground objects. Objects which are totally obscure and do not appear as spatial stimuli themselves are not taken into account due to limited attentional resources. Therefore, in contrast to Itti's model, the saliency map [84] is first used here as a segmentation of interesting objects from their background. Then, the Bayesian surprise definition proposed in [79, 80] is applied directly on consecutive static saliency maps and merge the spatial saliency and temporal novelty together.

Since the LS computation is conducted directly on static saliency maps and, therefore, has a tight link to saliency map computation, an introduction to the saliency map model is given in the following section.

3.2.2 The Saliency Map Model

Among all the attention selection models and applications, the saliency map model proposed in [84] is one of the most well-known standard computational models for bottom-up

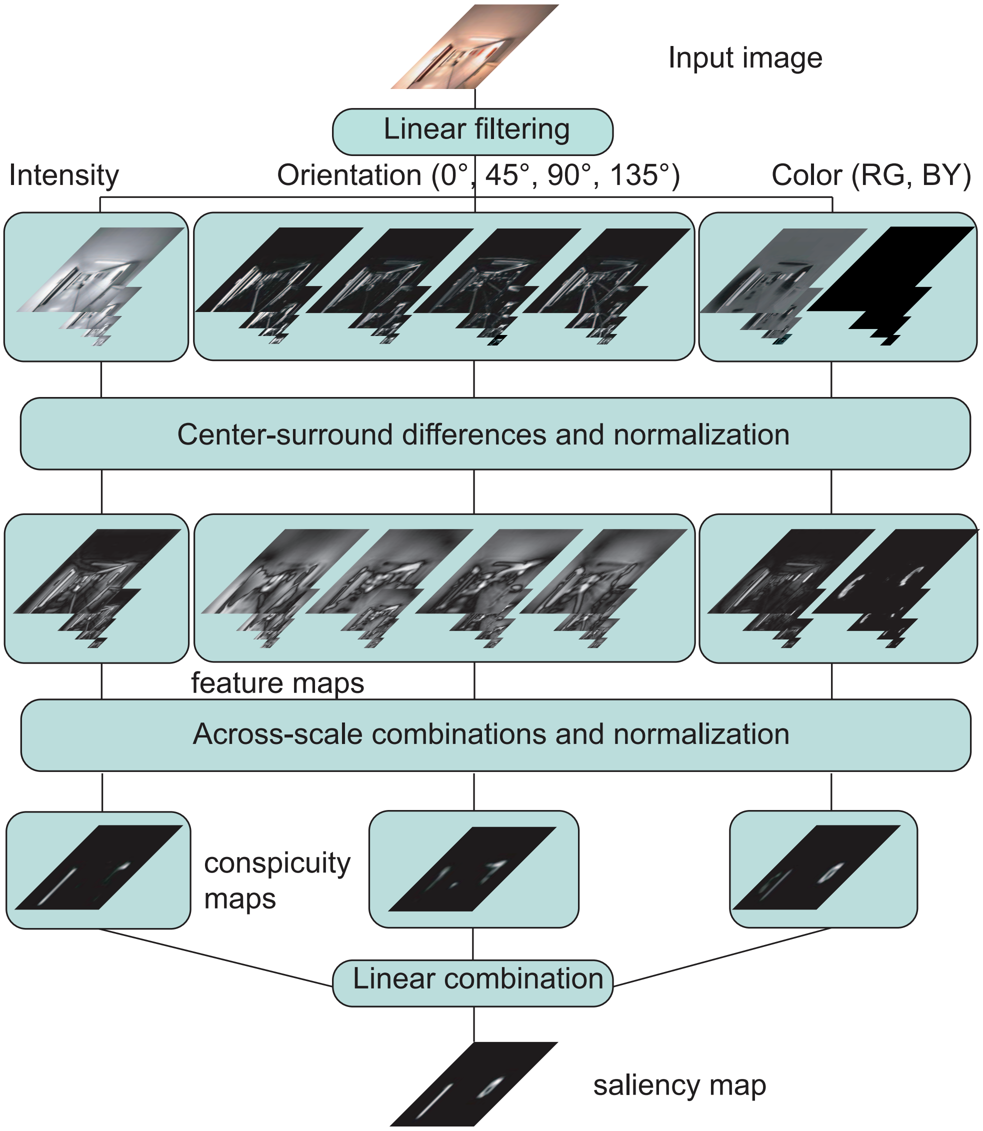
Input image
Linear filtering
Intensity
Orientation (0°, 45°, 90°, 135°)
Color (RG, BY)
Center-surround differences and normalization
feature maps
Across-scale combinations and normalization
conspicuity maps
Linear combination
saliency map

Fig. 3.1: The saliency map computation model

attention selection, illustrated in Fig. 3.1. From an input image, a saliency map is generated to predict the salient positions which potentially attract human visual attention. The saliency map computation is described here.

Dyadic Gaussian Pyramids

Firstly, the input image is subsampled into dyadic Gaussian pyramids in three channels (intensity (I), orientation (O) for $0°, 45°, 90°, 135°$, opponent color (C) in red/green (RG) and blue/yellow (BY)). A separable kernel of a 6×6 Gaussian kernel $[1\ 5\ 10\ 10\ 5\ 1]/32$ is used for the successive image size reduction. In the end, the input image at a resolution of for example 640×480 pixels is subsampled into 8 other scales: 320×240 (scale $\sigma = 1$), 160×120 ($\sigma = 2$), ..., 2×1 ($\sigma = 8$) (see Fig. 3.2).

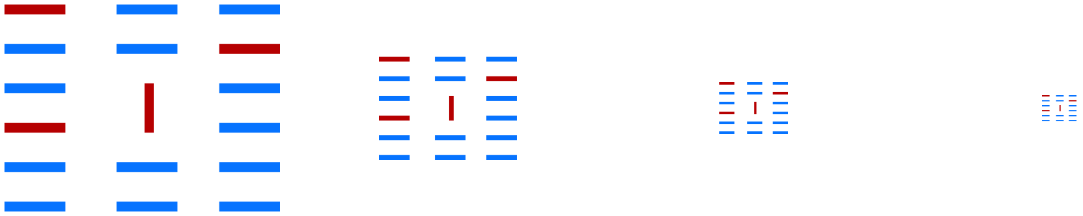

Fig. 3.2: Example images in a dyadic Gaussian pyramid. From left to right: 640×480 at scale 0; 320×240 at scale 1; 160×120 at scale 2; 80×60 at scale 3.

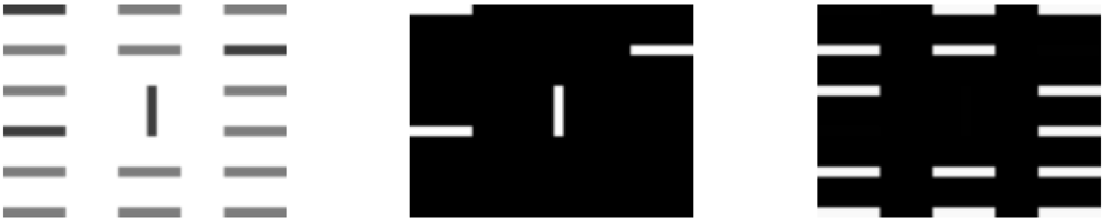

Fig. 3.3: Example images from left to right: I-map, RG-map and BY-map at scale 3 in the Gaussian pyramids.

I-, C-, O-Maps Computation

Then, the I-, RG- and BY-maps in the dyadic Gaussian pyramid in each channel are computed. According to [196], the I-map $M_I(\sigma)$ at scale σ is computed as follows:

$$M_I(\sigma) = \frac{r+g+b}{3}, \tag{3.1}$$

where r, g, b are the pixel values for red, green, and blue in RGB color space.

The opponent C-maps are computed as follows:

$$M_{RG}(\sigma) = \frac{r-g}{\max(r,g,b)}, \tag{3.2}$$

$$M_{BY}(\sigma) = \frac{b-\min(r,g)}{\max(r,g,b)}. \tag{3.3}$$

The image regions in which $\max(r,g,b) < 0.1$ are set to 0. Example resulting images are shown in Fig. 3.3.

To compute the O-maps in different scales, a Gabor-filter truncated to 19×19 pixels is used [196], which is formulated as follows:

$$G_\psi(x,y,\theta) = exp\left(\frac{x'^2 + \gamma^2 y'^2}{2\epsilon^2}\right) \cdot cos\left(2\pi\frac{x'}{\chi} + \psi\right), \tag{3.4}$$

with

$$x' = x\,cos(\theta) + y\,sin(\theta), \quad y' = -x\,sin(\theta) + y\,cos(\theta), \tag{3.5}$$

where (x,y) is the pixel coordinate in the Gabor-filter. The parameter values of this implementation are set according to [196], where γ stands for the aspect ratio with the

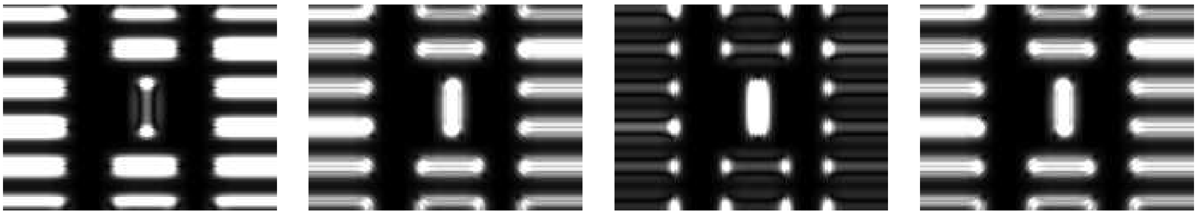

Fig. 3.4: Example images from left to right: O-maps at scale 3 in the Gaussian pyramids for orientations 0°, 45°, 90°, and 135°.

value 1 and χ is the wavelength and has the value of 7 pixels. The standard deviation ϵ of the Gaussian envelope is equal to 7/3 pixels, and the phase offset $\psi \in \{0, \frac{\pi}{2}\}$. θ stands for the orientation angles with $\theta \in \{0°, 45°, 90°, 135°\}$.

As defined in Eq. 3.4, the Gabor-filter consists of a combination of a 2D Gaussian bell-shaped curve and a sine ($\psi = \pi/2$) and cosine function ($\psi = 0$). In each direction, the image should be filtered twice and summed as follows:

$$M_\theta(\sigma) = |M_I(\sigma) * G_0(\theta)| + |M_I(\sigma) * G_{\pi/2}(\theta)|, \tag{3.6}$$

with $M_I(\sigma)$ the I-maps at scale σ. Fig. 3.4 illustrates the O-maps in different orientations at scale 4.

After the steps above, 9 I-maps, 18 C-maps and 36 O-maps are generated. It is followed by center-surround differences and across-scale combinations. In these two steps, images at different scales are subtracted and combined.

Center-Surround Differences

In center-surround differences, 6 feature maps (FMs) in the I-channel, 12 FMs in the C-channel and 24 FMs in the O-channel are computed as follows:

$$I(c, s) = |I(c) \ominus I(s)|, \tag{3.7}$$

$$RG(c, s) = |(R(c) - G(c)) \ominus (R(s) - G(s))|, \tag{3.8}$$

$$BY(c, s) = |(B(c) - Y(c)) \ominus (Y(s) - B(s))|, \tag{3.9}$$

$$O(c, s, \theta) = |O(c, \theta) \ominus O(s, \theta)|, \tag{3.10}$$

with c referring to the fine scale and s indicating the coarse scale: $c = \{2, 3, 4\}$; $\delta = \{3, 4\}$; $s = c + \delta$. θ is the orientation of the Gabor-filter. $\ominus$ means the subtraction between two images at different scales c and s. To execute this subtraction, the images should be enlarged or reduced into the same size and then a point-by-point subtraction is accomplished. Fig. 3.5 shows an example of center-surround differences.

After an iterative normalization of the output images of the subtraction, feature maps are generated in which the distinctive pixels with respect to their neighborhood are highlighted.

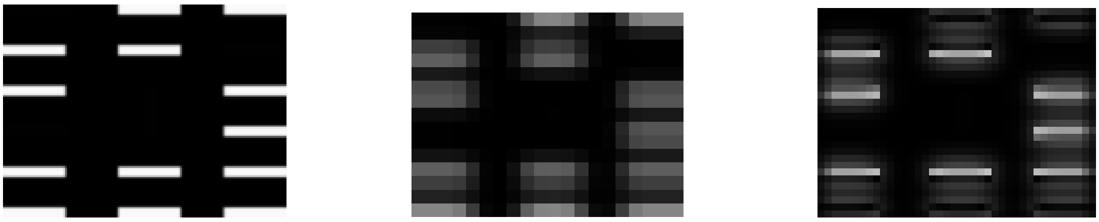

Fig. 3.5: Example images. Left: a BY-map at scale 3; Middle: a BY-map at scale 6; Right: $BY(3,6)$, a BY feature map at scale 3-6.

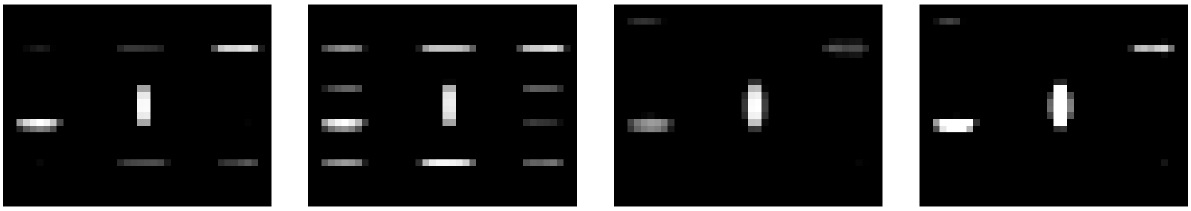

Fig. 3.6: Example images from left to right: Conspicuity maps $\bar{I}$, $\bar{C}$, and $\bar{O}$ in intensity, color and orientation channels as well as the resulting saliency map S.

Iterative Normalization

The iterative normalization $N(\cdot)$ is an important component in the whole computation. It simulates local competition between neighboring salient locations [84]. Each iteration contains self-excitation and neighbor-induced inhibition, which can be implemented using a difference-of-Gaussian (DoG) filter [82]:

$$DoG(x,y) = \frac{c_{ex}^2}{2\pi\sigma_{ex}^2} e^{-\frac{x^2+y^2}{2\sigma_{ex}^2}} - \frac{c_{inh}^2}{2\pi\sigma_{inh}^2} e^{-\frac{x^2+y^2}{2\sigma_{inh}^2}}, \tag{3.11}$$

with $\sigma_{ex} = 2\%$ and $\sigma_{inh} = 25\%$ of the input image width, $c_{ex} = 0.5$, $c_{inh} = 1.5$. At each iteration the given image M is computed as follows [82]:

$$M \leftarrow |M + M * DoG - C_{inh}|_{\geq 0}, \tag{3.12}$$

with the constant inhibitory term $C_{inh} = 0.02$, where $|\cdot|_{\geq 0}$ discards negative values.

Across-Scale Combinations

Using across-scale combinations, the FMs at different scales are combined and normalized into a conspicuity map (CM) in each channel. It is only a question of point-by-point integration of the FMs into CMs $\bar{I}$, $\bar{C}$ and $\bar{O}$ as follows [84]:

$$\bar{I} = \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} N(I(c,s)), \tag{3.13}$$

$$\bar{C} = \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} [N(RG(c,s)) + N(BY(c,s))], \tag{3.14}$$

$$\bar{O} = \sum_{\theta \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}} N(\bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} N(O(c,s,\theta))). \tag{3.15}$$

Final Saliency Map

The saliency map S is a linear combination of the normalized conspicuity maps, defined by

$$S = \frac{1}{3}(N(\bar{I}) + N(\bar{C}) + N(\bar{O})). \tag{3.16}$$

The conspicuity maps, shown in Fig. 3.6, illustrate the conspicuous pixels regarding color, intensity, and orientation. The bright pixels in the saliency map (the right one in Fig. 3.6) are the most salient and interesting pixels predicted by the saliency map model.

The saliency map computation is implemented on a multi-GPU platform in order to achieve a high-speed performance. The implementation details can be found in Appendix B.

3.2.3 Local Surprise Definition

The saliency map model computes spatial saliency in a static image which potentially attracts human attention. In technical systems, the saliency map can be used as a segmentation of potentially interesting objects from their background. Since dynamic characteristics of the current environment are also an essential cue for robot vision control, temporal novelty in an image sequence should be integrated. Temporal novelty can be modeled as a Bayesian Surprise proposed in [79], evaluated by the difference between the belief and the perceived information about the world, which measures how novel, surprising, or unexpected the new information is observed. The image region with a higher surprise value is worth being further processed.

Inspired by [79], an LS map is constructed, computing the LS value of salient image regions which have already been predicted in the saliency map computation. LS is defined by applying the Bayesian surprise definition directly on two consecutive saliency maps (see. Fig. 3.7).

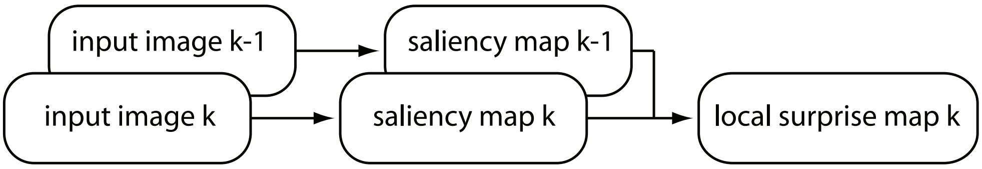

Fig. 3.7: Computation structure of LS maps.

The computational details of the surprise map are described as follows. From two consecutive input images, two saliency maps are computed as described in the previous section. Each pixel i with its normalized saliency value λ_i in the saliency map is regarded as a detector for LS and is modeled as a probability distribution, representing the observed saliency value and the observation uncertainty. The choice of probability distribution is trivial. For instance, Gaussian mixture is used in [19]. Since the data input of the local detector can be regarded as Poisson distributed events (neuron spikes) [80], Gamma probability density function (pdf), which maintains its functional form when Poisson-distributed data is observed, is used to model the saliency of a pixel as follows:

$$p_i \quad = \quad \gamma(\lambda_i, \alpha_i, \beta_i) = \frac{\beta_i^{\alpha_i} \lambda_i^{\alpha_i - 1}}{\Gamma(\alpha_i)} \cdot e^{-\beta_i \lambda_i}, \tag{3.17}$$

with the shape $\alpha_i > 0$ with an initial value A, the inverse scale $\beta_i > 0$ with an initial value B, and $\Gamma(\cdot)$ the Euler Gamma function.

At time step k, an observation is conducted in which an input image is captured and a respective saliency map is computed. Each pixel in the saliency map is considered as an LS-detector and obtains a saliency value of $\lambda_{i,k}$. Due to the new data input, the belief or observation of each detector is changed, in which the parameters α_i and β_i evolve as follows [80]:

$$\alpha_{i,k} = A + N_{1D}(\lambda_{i,k}), \text{ and } \beta_{i,k} = \xi \cdot B + 1. \tag{3.18}$$

where N_{1D} is a 1D-normalization and $\xi \in (0, 1)$ is a forgetting factor.

At the next time step $k + 1$, before a new observation is conducted, it is assumed that the environment does not change. The belief is based on the observation at time step k. The sensed information at each detector or each image pixel is the prior belief distribution and is formulated as follows:

$$p_{i,k} = \gamma(\lambda_{i,k}, \alpha_{i,k}, \beta_{i,k}). \tag{3.19}$$

After a new image is captured at time step $k+1$, the detector/pixel i has a new saliency value of $\lambda_{i,k+1}$. The belief of the environment is updated, in which the parameters α_i and β_i of this pixel evolve as follows:

$$\alpha_{i,k+1} = \alpha_{i,k} + N_{1D}(\lambda_{i,k+1}), \text{ and } \beta_{i,k+1} = \xi \cdot \beta_{i,k} + 1, \tag{3.20}$$

Then, the posterior belief distribution can be formulated as follows:

$$p_{i,k+1} = \gamma(\lambda_{i,k+1}, \alpha_{i,k+1}, \beta_{i,k+1}). \tag{3.21}$$

To quantify the distance of the prior belief and the posterior belief of a detector about the sensed environment, the LS for the pixel i is defined as Kullback-Leiber-divergence $\tau(x, y)$ (also the relative entropy) between the posterior and prior saliency distributions, formulated as follows:

$$\begin{aligned} \tau_{i,k+1} \quad &= \quad KL(p_{i,k+1} || p_{i,k}) \\ &= \quad -\alpha_{i,k} + \alpha_{i,k} \log \frac{\beta_{i,k+1}}{\beta_{i,k}} + \log \frac{\Gamma(\alpha_{i,k})}{\Gamma(\alpha_{i,k+1})} \\ &\quad\quad +\beta_{i,k} \frac{\alpha_{i,k+1}}{\beta_{i,k+1}} + (\alpha_{i,k+1} - \alpha_{i,k}) \Psi(\alpha_{i,k+1}) \text{ in [bit]}, \end{aligned} \tag{3.22}$$

where $\Psi(\cdot)$ is the Digamma function.

Example: Given two pixels with the saliency values λ_1 and λ_2, $\lambda_{1,2} \in [0, 255]$ in the saliency maps, the computed LS values at time step $k+1$ are listed in Tab. 3.1.

	Group 1		Group 2		Group 3		Group 4	
Time step	λ_1	λ_2	λ_1	λ_2	λ_1	λ_2	λ_1	λ_2
k	200	100	200	100	200	200	200	150
$k+1$	200	100	255	155	255	230	210	250
LS [bit]	0.9228	0.4583	1.1878	0.7121	1.1878	1.0666	0.9705	1.1655

Tab. 3.1: LS values computed using different sample saliency values of two pixels λ_1 and λ_2. Group 1 and 2: If the saliency value variations are the same, the pixel with a higher saliency value wins a higher LS value; Group 3 and 4: The pixel with a higher saliency value variation wins a higher LS value.

The LS is an indicator of a bottom-up robot gaze control. It combines the temporal novelty and the spatial saliency in a way that the larger the inter-frame saliency variation of the pixel i is and the higher the saliency value the pixel i contains, the higher the LS value $\tau_{i,k+1}$ is. For applications on mobile robots, the saliency map is rescaled to a small dimension to simplify the computation and ensure the real-time capability.

3.2.4 Performance Evaluation

Three experiments are conducted to demonstrate and investigate LS in different aspects: LS induced by salient object onset, LS induced by temporal novelty and spatial saliency, and how LS guides robot attention flexibly to facilitate robot tasks.

Experiment 1: LS Induced by Salient Object Onset

Fig. 3.8 illustrates an example of LS map computation. In the second input image, a human entered into the scene suddenly. Consequently, a large intensity value appears in the respective position in the surprise map. The other bright pixels show the LS value due to higher saliency value and small saliency value variation.

Experiment 2: LS Induced by Temporal Novelty and Spatial Saliency

Fig. 3.9 shows the performance of this strategy combining the spatial saliency and the temporal novelty. A red cylinder with high spatial saliency and a cup held by a hand with relatively low spatial saliency were investigated. The cylinder was mounted on the wall and had no self-motion at all during this experiment.

At first, the camera's attention was directed to the cylinder. Then, the hand and the cup shifted in front of the camera at time steps k and $k+1$ without camera ego-motion (see Fig. 3.9, left). Although the cylinder had a larger saliency, the hand/cup had a high LS value evaluated by the surprise map at this time. The camera head then moved towards the hand/cup to bring them into the center of the view. After the change of the camera gaze direction, the hand/cup stopped moving at time step $k+2$ and $k+3$ (see Fig. 3.9, middle) and lost, therefore, their high LS value. Due to the higher saliency value,

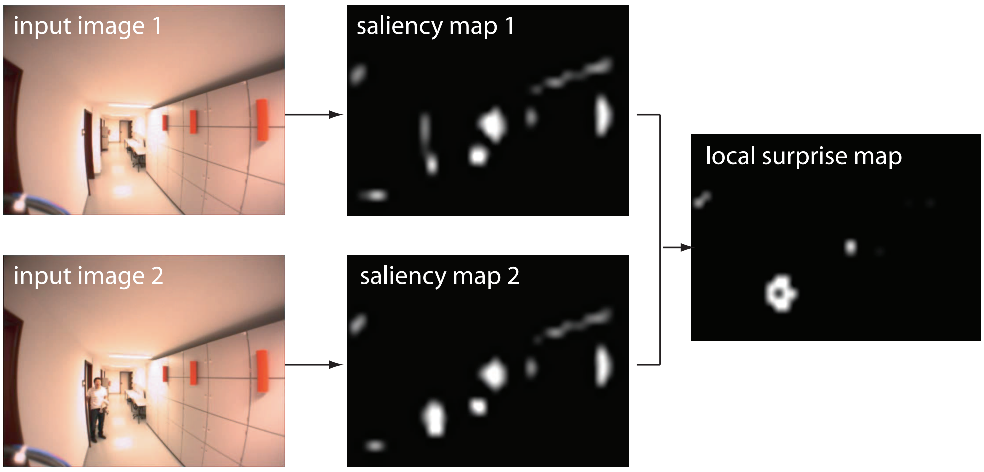

Fig. 3.8: LS induced by salient object onset – a human entering the scene.

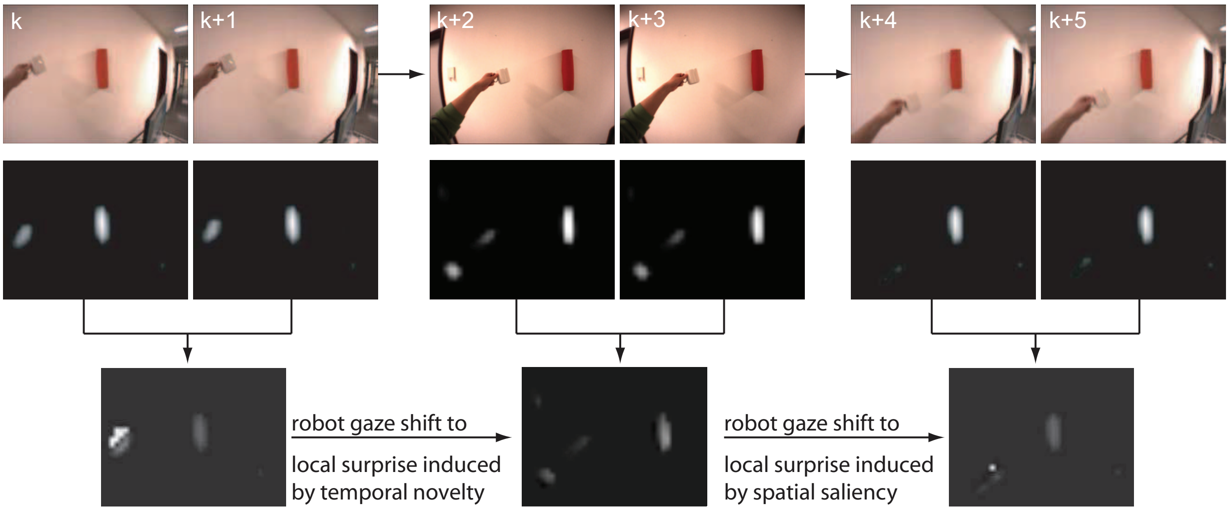
k
k+1
k+2
k+3
k+4
k+5
robot gaze shift to
local surprise induced by temporal novelty
robot gaze shift to
local surprise induced by spatial saliency

Fig. 3.9: Robot gaze shift towards the position of the LS maximum, which is induced by temporal novelty (hand/cup movement in the left two columns) and by spatial saliency (salient red cylinder mounted on the wall in the middle two columns). Upper row: original images; Middle row: the respective saliency maps; Lower row: LS maps.

the cylinder succeeded in attracting the camera's attention again. After the camera gaze direction changed at time steps $k+4$ and $k+5$, the cup was moved by the hand again and acquired a high LS value in the LS map (see Fig. 3.9, right).

This experiment shows evidently that this strategy successfully combines the spatial saliency and the temporal novelty for robot gaze control.

Experiment 3: LS Facilitating a Robot Task

In this experiment, an object search task is combined with the surprising event detection. Fig. 3.10 shows the image sequence in this experiment. The objective was to detect a traffic sign, which was not located in the FOV at the beginning. The left columns show the

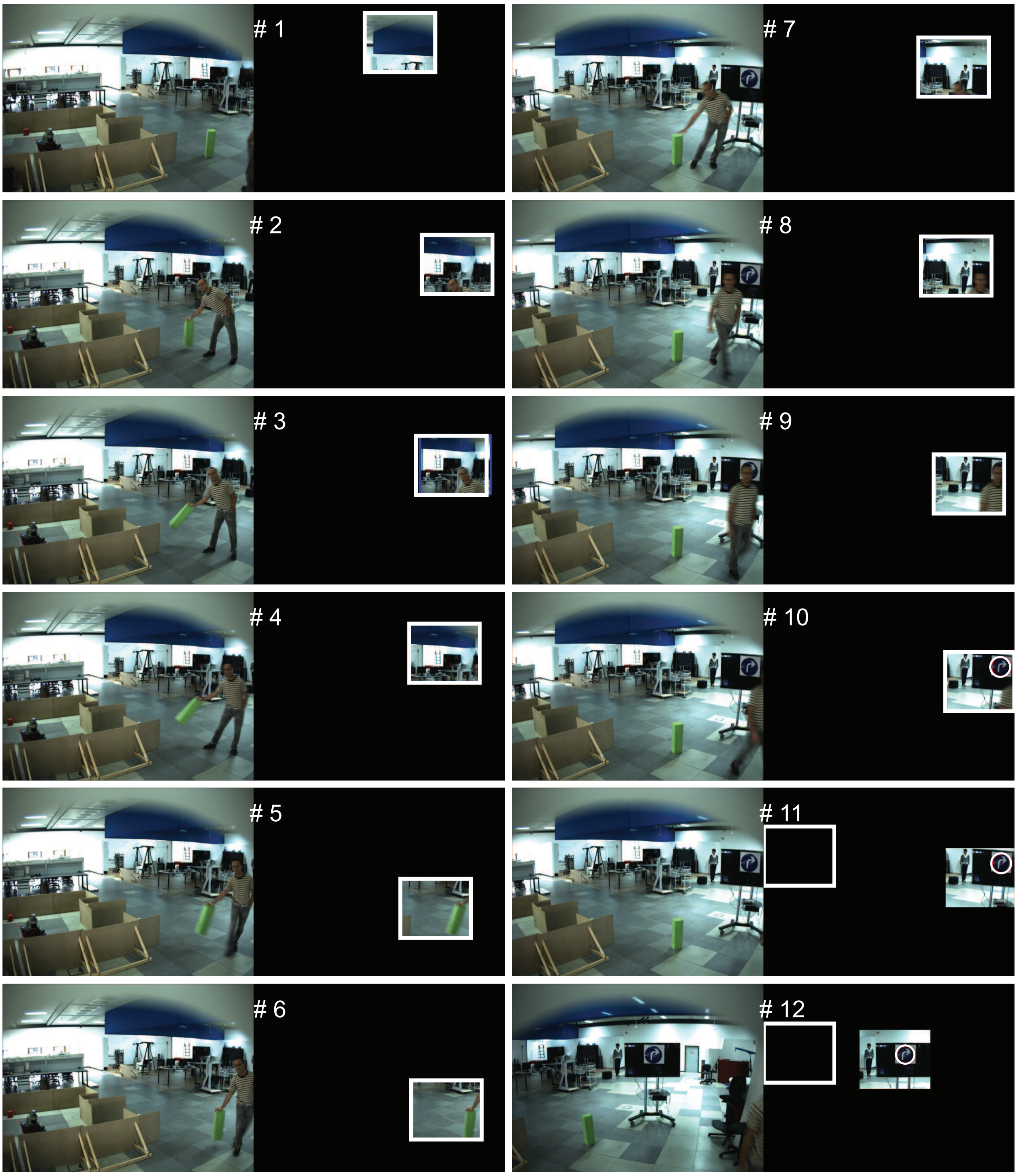

Fig. 3.10: Image sequence for object detection task. Left sub-columns: the original input images. Right sub-columns: the respective masked result images with the frame number. Masked regions: the robot FOA; Rectangles: image regions containing the maximum LS; Circles: the detected objects.

original input images. Aiming at efficient information selection and processing, the object detection algorithm ran only in the preselected regions limited by the white rectangles in the images of the right columns which indicate regions with maximum LS values in the surprise maps. A human tried to bring the traffic sign into the camera FOV by moving a salient green cylinder near to the traffic sign. The human and the cylinder were detected successfully as a surprising event. The camera platform was controlled to look straightly towards the surprising event in every five images. Through tracking of an LS, the limited FOV is flexibly extended, which made object detection in this case possible.

3.2.5 Discussion

From the experimental results above, the performance of LS in terms of combining spatial saliency and temporal novelty is demonstrated. This information-based extension of saliency map model integrates motion cue into bottom-up attention and detects the unexpected changes in the environment, which is advantageous for world model exploration and update. The preference for surprising events in the bottom-up attention model enables safe operations of mobile robots or manipulations in dynamic environments.

Other Definitions of Surprise

LS has been already defined in [111] to quantify the difference between the perception and the knowledge of a *Belief-Desire-Intention-Surpise* (BDIS) agent about his intention. An abstract model of an agent's mental state considering surprise-based filter of belief update is established. An agent should update his beliefs only with surprising and intention-relevant inputs. The definition is general and not specific for visual perception. Other Bayesian modellings of visual perception have been defined in different applications such as video compression using Bayesian foveation [20], image change detection [114], and landmark detection [151]. Generally, all the works have only considered goal-directed surprise.

LS and Motion Map

The LS can be regarded as another interpretation of motion map, considering the inter-frame saliency variation of each pixel instead of the inter-frame intensity variation. The main difference between LS and explicitly combining a motion map into saliency map computation is that no optimized weight for motion map is needed to be explored. Moreover, from common motion maps the temporal change of intensity is obtained, which is usually the contour of a dynamic object (see motion map in Fig. 3.11). Compared with a motion map, in an LS map the object with varying appearance is detected as a whole, which is more interesting for robot attention control. Compared with a difference map of two consecutive saliency maps, an LS map also considers spatial saliency, if no temporal novelty occurs at that time (see middle column in Fig. 3.9).

LS and Itti & Baldi's Surprise

Itti & Baldi's Bayesian surprise model [80] is aiming at a best explanation of human behavior in an active search for non-specific information of subjective interest, namely purely bottom-up attention. It is a general surprise definition. In this thesis, this concept is used for detection of onset, offset, and motion of salient foreground objects, focusing on

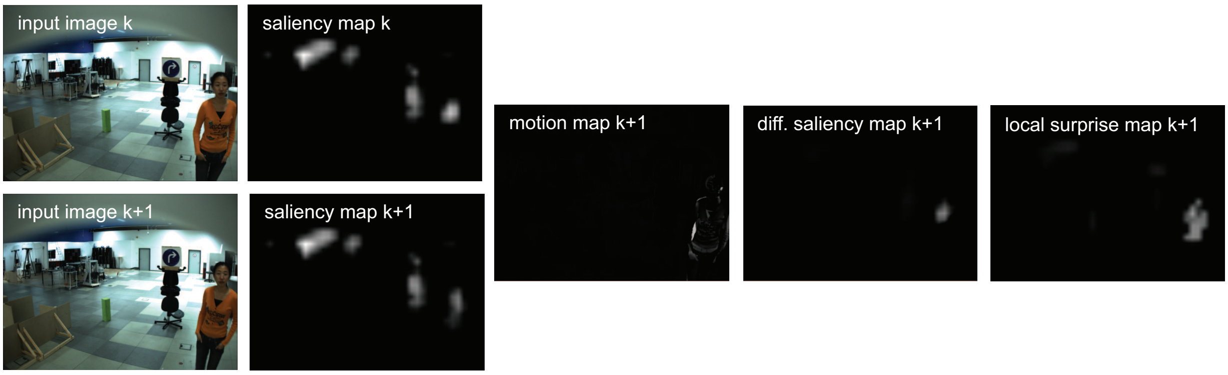

Fig. 3.11: Comparison of LS map, motion map, and difference map of consecutive saliency maps computed using two consecutive input images.

robot applications. A saliency map is used to segment the scene and determine interesting image regions, namely the salient regions.

Furthermore, in the computation, in Itti & Baldi's model, the distribution parameters (α and β in Eq. 3.18) are updated consistently to resemble a short-term, human-like surprise damping behavior. In this model, the initial values of the distribution parameters are constant to emphasize saliency variation at the current time step and enhance the sensitivity of awareness of dynamic environments on the one hand. On the other hand, another metric GS is defined in the next section to describe the global dynamics of the environment and resemble the damping effect. In this way, LS and GS are independent of each other and can be flexibly applied alone or together in different contexts.

Limitations

Since a robot is supposed to operate in the environment, a permanent attending to the maximum LS is inefficient and will cause possible loss of task-relevant information. The maximum LS is the maximum compared with the other image regions in the FOV. Two maximum LS values in consecutive input images can not be compared with each other along the time scale, since they are relative measures in terms of importance for attention in comparison to other image regions. Therefore, a measure is needed to trigger attention towards the current maximum LS. Aiming at this, GS is defined in the following section.

3.3 Global Surprise for Representation of Environment Dynamics

As discussed, if a robot always directs its FOA to the LS determined in the input image, the on-going task accomplishment will be impaired. Therefore, a metric for a good timing to take this saccade behavior should be derived, which the robot can quickly rely on to determine how uncertain the environment is and how it should behave in this situation; for example, to stop moving in a chaotic environment to avoid crashes.

3.3.1 State of the Art

Psychological research has studied the human attention control considering unexpected stimuli while one or more tasks are to be conducted concurrently. Some works suggest that high perceptual load reduces interference from distractors [101]. Top-down guidance tends to dominate in real-world visual search [32]. Unexpected/task-irrelevant stimuli can be detected if percetual/cognitive resources are available. The *inattentional blindness theory* [113] states that when subjects are engaged in one task, they are remarkably insensitive to other perceptual tasks due to limited human attention capacity. This indicates the relationship between the attention distribution and the subject internal state. This inspiration can be applied to robots; for instance, if the task is accomplished satisfactorily according to certain task-oriented measures, a gaze control towards the LS can be conducted as the decision making strategy proposed in [168]. Since the focus in this chapter is image-based information perception, internal robot states are not considered at this step. Further information about attention control relying on internal robot state can be found in Chapter 4.

Other works suggest that subjects are sensitive to environmental events and modify the distribution of attention accordingly to detect unexpected stimuli, because peripheral vision can actively monitor other moving objects in the field [87]. This indicates that the external environment also plays an important factor in attention selection. A global metric should be proposed to describe the environment dynamics and be applied as a trigger for LS.

By now, scene or context representation can be divided into two categories: object-based [192], and gist-based [172, 183, 184]. The former pre-supposes recognized objects and their pre-defined relations, while the latter provides a stable prediction on what kind of object could be expected in this scene or context [156]. However, the object-based and gist-based scene representations are static. Another approach for context recognition is behavior-based [127, 170], in which behavior or intention of manipulators is recognized and guides the attention allocation. However, behavior-based context representation focuses on intention recognition of the local motion and can not represent the global environment dynamics.

An effect called *mindsight* is reported in [157], in which people can have a strong feeling about the environment changing without seeing it. This feeling can then alert the attention system to be aware of the change. Inspired by this phenomenon, another metric is defined, the GS, to interpret how surprising or uncertain the current environment is. Based on this interpretation, an economical timing for robot attention shift towards LS is determined.

3.3.2 Global Surprise Definition

In humans, feature contrast affects the speed of performance more than feature values themselves [187]. GS is computed as the relative entropy between the prior image variation and posterior image variation along the time scale to compute the variation rate of motion in the environment between two consecutive time steps.

Firstly, a static environment is regarded as an unsurprising, certain, and familiar environment for the robot and the dynamic environment as a surprising environment because of changes and danger caused by the moving objects. Three consecutive input images without camera ego-motion are converted from RGB color space into grayscale, denoted by I_{k-2}, I_{k-1} and I_k, indicating the image intensity. Large intensity variation can be regarded as motion. Therefore, motion maps are calculated as the difference between two consecutive input images with pixel number i:

$$M_{i,k-1} = I_{i,k-1} - I_{i,k-2}, \tag{3.23}$$

and

$$M_{i,k} = I_{i,k} - I_{i,k-1}. \tag{3.24}$$

Then, the histograms $H_{k-1}(j)$ and $H_k(j)$ for each motion map are computed, in order to represent the motion distribution in the pixel value range $j \in [0, 255]$.

Each normalized histogram of motion map can be regarded as a discrete distribution. The relative entropy T of the histograms of the two consecutive motion maps is computed as the GS of the current scene as follows:

$$T_k = \mathrm{KL}(N_{1D}(H_k)||N_{1D}(H_{k-1})) = \sum_i N_{1D}(H_k) \log \frac{N_{1D}(H_k)}{N_{1D}(H_{k-1})} \text{ in [bit]}. \tag{3.25}$$

where $N_{1D}(\cdot)$ is the 1D-normalization function.

GS indicates the rate of motion variation in the current environment. The larger T is, the more chaotic the current scene is. This metric can be used to alert the robot system when the current environment contains unpredictable dynamics, whereas an FOA towards the current LS may be necessary. A threshold is needed to distinguish between the relatively dangerous environment and relatively tame environment.

3.3.3 Experimental Investigation

The threshold is determined experimentally. Different representative scenes were gathered, shown in Fig. 3.12, and their GS values were calculated.

- Scene 1: a floor without moving objects present
- Scene 2: a square with crowded people
- Scene 3: a floor with people suddenly appearing
- Scene 4: a street with a vehicle moving very fast

The rows show the consecutive time steps $k-2$, $k-1$ and k at a frame rate of 30 fps. It is obvious that the environment almost does not change in the first scene. Therefore, the GS value is very small, namely 0.0526 bit. In comparison to the first scene, in scene 4 the environment changes very significantly because of the vehicle movement. Hence, the GS value in this environment is large, namely 0.1761 bit. For the second scene a small GS value is obtained, namely 0.0473 bit, because the scene dynamics change is relatively small although there is motion. This environment can be regarded as a non-chaotic environment, since no surprise exists. In the third scene the GS value is large, because a person entered suddenly into the FOV. Therefore, the scene dynamics change is relatively large.

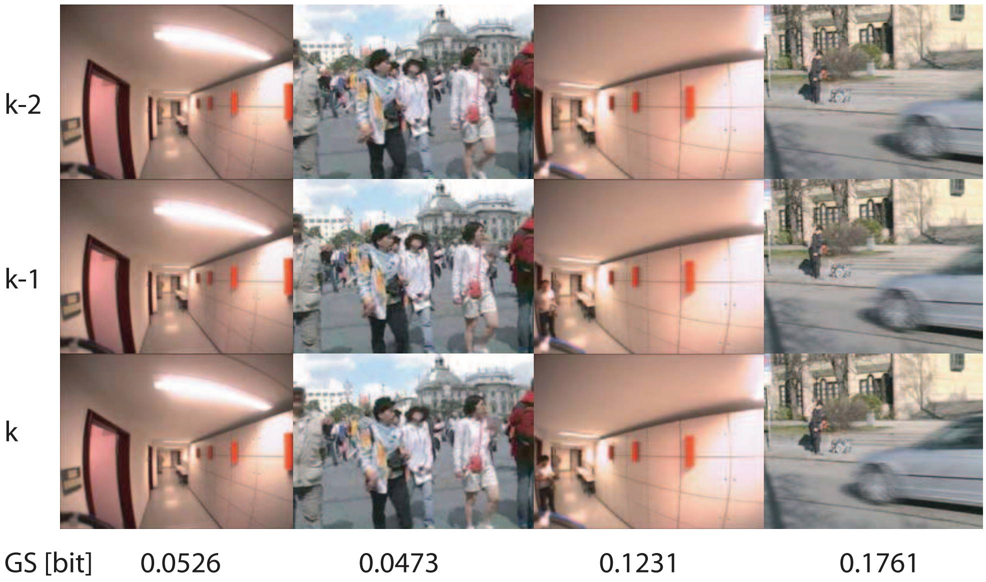

Fig. 3.12: Experimental investigation of GS using four different scenes with different scene dynamics. From left to right: a static indoor scene; a city-center scene with slight pedestrian movement; an indoor scene with an unexpected event – a human entering the scene; a street scene with a high-velocity vehicle.

The threshold is empirically determined and may differ in different contexts. For the experiment in the next section, it is equal to 0.1 bit.

3.3.4 Discussion

As mentioned, permanent looking at LS can cause a loss of the current task-relevant target and is computationally inefficient for robot task. Moreover, LS, as a relative and normalized measure, cannot be used temporally to direct robot attention towards the maximum LS itself. Therefore, it is reasonable and necessary to introduce a global metric to alert the robot system about the current environment dynamics.

GS can also be classified into a gist-based description of the environment. In comparison to the other gist-based scene recognition approaches [172, 183], GS does not provide a typical scene classification for, for example, object position prediction. The focus of this metric considers environment uncertainty, which is a key factor for robot operation in an unstructured real world. Compared to this definition, another GS is defined in [111], regarded as the global maximum of the LS, which is in a totally different domain.

Although the definition of GS is still limited in some aspects at the current step, e.g. sensitive to lighting conditions, only an empirical determination of the threshold and difficult quantitative evaluation, the necessity, and possibility of using such a global metric to facilitate robot tasks are shown.

A more sophisticated concept may be to apply an extra vision sensor with a wide FOV, e.g. omni-directional camera, to monitor the operating environment. Similar research topics are multi-object tracking and scene classification.

3.4 Surprise-Driven Robot Attention Control

LS is an extension of bottom-up attention selection, which interprets input images without any pre-defined task information. However, it is not easy to apply this metric for a conventional robot operation. For a robot task, in which LS detection and tracking only hold a secondary meaning, a novel combination of LS and GS is designed for this kind of robot attention control.

3.4.1 Robot Decision-Action Design

The whole robot decision-action design is shown in Fig. 3.13. In the perception/decision block, the GS of the current environment is computed along with task-oriented information perception and interpretation. A decision is made in the way that if the GS value is smaller than a pre-defined threshold, which means the environment is safe enough, the robot performs its primary task for the next time point. If the value of the GS is larger than the pre-defined threshold, the robot should stop its primary task and attend to the position with maximum LS for its own safety. In the control application block, the determined action is carried out, which results in an update of camera orienting or robot locomotion.

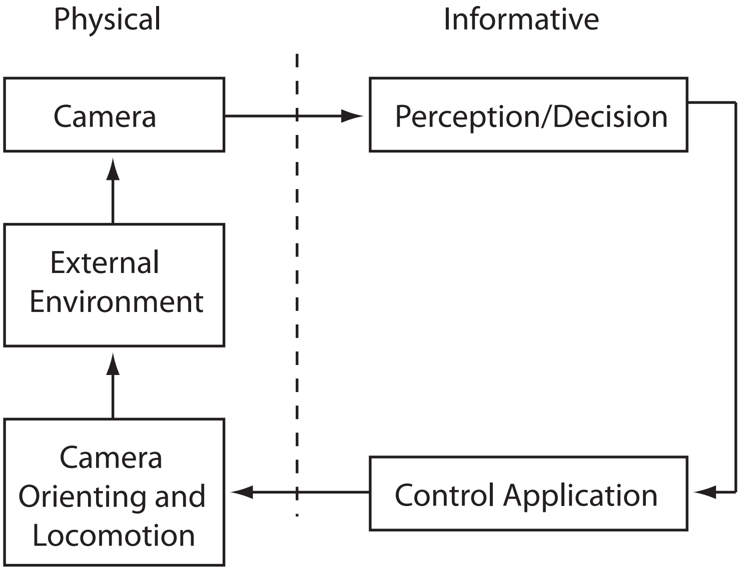

Fig. 3.13: Robot decision-action design.

3.4.2 Performance Demonstration

An experiment is conducted using the ACE robot. The primary task is vision-based robot self-localization. The GS value is used as an indicator to alert the ACE robot to distribute attention to the LS in the environment during its task.

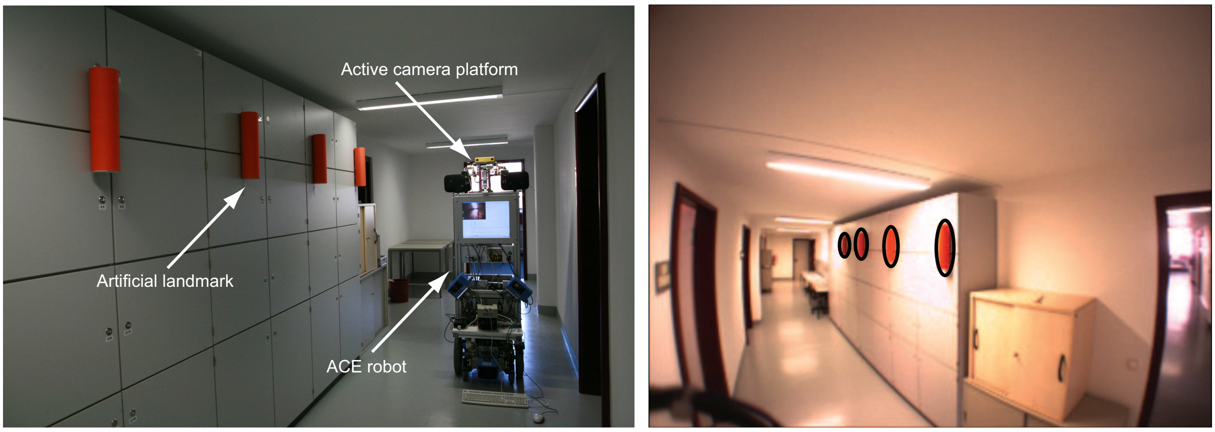

Fig. 3.14: Left: the ACE robot with an active camera head and four artificial landmarks mounted on the wall. Right: color-based landmark detection in 2D input images.

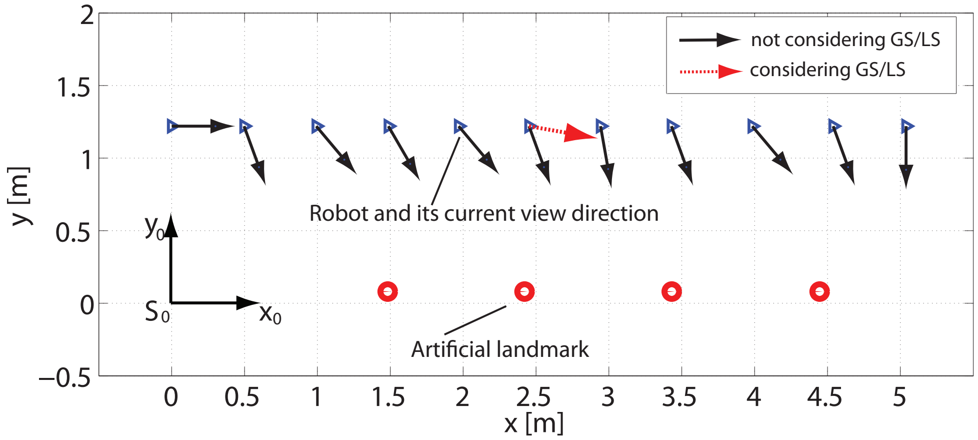

Fig. 3.15: A top view of the robot locations and view directions during the robot locomotion, considering or not considering the GS. Triangles: the actual robot positions; Solid circles: the artificial landmarks; Solid lines: robot view directions for the self-localization task; Dashed line: robot view direction toward the maximum LS.

Experimental Setup and Task Description

The experiment was executed in a corridor (see Fig. 3.14 left). An active stereo camera (Bumblebee I with focal lengths of 2 mm each from Point Grey Research Inc., see Appendix A) was used for this experiment. Four artificial landmarks were installed at the same height as the camera, such that only robot attention distribution in a horizontal plane was considered. The landmark positions were known.

A world frame $\boldsymbol{S}_0$ is defined, shown in Fig. 3.15. The ACE robot, illustrated as triangles in Fig. 3.15, moved from its start point $(0, 1.25)$ m to $(5.0, 1.25)$ m straight forward. About every 0.5 m a view direction planning was executed and an optimal view direction was applied for the next 0.5 m. Three measurements for position estimation were executed at each time step. The view direction Ω is defined as the angle between the locomotion direction and the camera optical axis in the horizontal plane. At the start point $(0, 1.25)$ m the camera had an initial view direction 0° towards the locomotion direction.

Landmark Detection

In each measurement, a color-based landmark detection was conducted on the stereo input images. The color images were converted first from the RGB color space into the HSV color space. In the hue-channel, the lower and upper bounds were determined to classify image pixels into landmark pixel and background pixel. After dilation and erosion processes, an ellipse was fitted for each landmark. The landmark positions in the left and right input images were represented by the ellipse center points. Using stereo triangulation, the 3D landmark positions with respect to the camera were obtained. Through coordinate transformations, the robot location with respect to the world frame $\boldsymbol{S}_0$ was calculated.

Information-Based View Direction Planning

For vision-based self-localization using landmarks, the view direction of a robot has an important impact on the accuracy of the estimation. An active vision system should decide autonomously where to pay its attention to. The view direction planner for self-localization uses an information-based strategy derived from [169]. The goal is to achieve a maximum information gain after each view direction change.

The information measure ν_k for this task at step k is defined as follows:

$$\nu_k = \frac{1}{2} \sum_{l=1}^{2} \sqrt{e_{l,k}}, \tag{3.26}$$

where $e_{l,k}$ are the eigenvalues of the robot position covariance matrices and l is the index for x- and y-direction. A *Kalman-filter* (KF) is used to predict the robot position and the covariance matrices. For possible view directions $\Omega_{k+1|k}$ of the camera, the respective information measures $\nu_{k+1|k}$ at step $k+1$ will be predicted. The view direction $\Omega^*_{k+1|k}$ with the strongest increase of the information content, which is defined as the maximum decrease of the covariance, is regarded as the optimal view direction and applied to the active vision system at the next time step $k+1$.

$$\Omega^*_{k+1|k} = \underset{\Omega_{k+1|k}}{\arg\max}(\nu_k - \nu_{k+1|k}(\Omega_{k+1|k})). \tag{3.27}$$

Experimental Results

Without considering the possible surprising events in the environment, the robot should concentrate itself on the localization task and attend to the landmarks in order to achieve an accurate position estimation [220]. In this experiment the information content for 10 different view directions $\Omega_{k+1|k} \in \{0°, 10°, ..., 80°, 90°\}$ (because all the landmarks are allocated in the right hand side of the mobile platform) was estimated and compared with the information content of the last view direction. The view direction which could provide the maximum information gain was chosen. In Fig. 3.15 the solid lines with arrows show the view directions, while Fig. 3.16 shows an image sequence captured in the self-localization task, with the respective optimal view directions at the respective robot positions in x-direction.

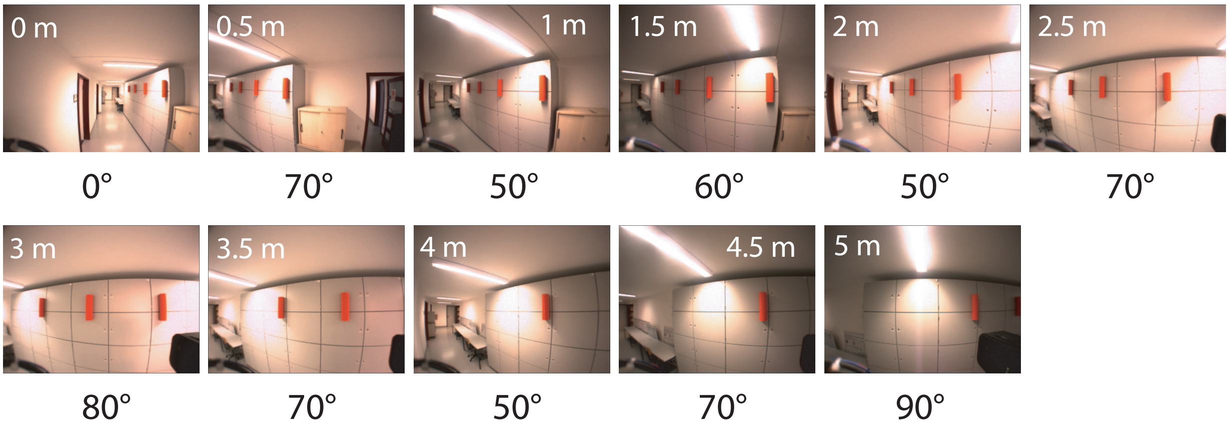

Fig. 3.16: Image sequence captured at different locations (x-coordinate on the images) for the self-localization task without consideration of GS and LS. The numbers under the images denote the camera view directions with respect to the locomotion direction.

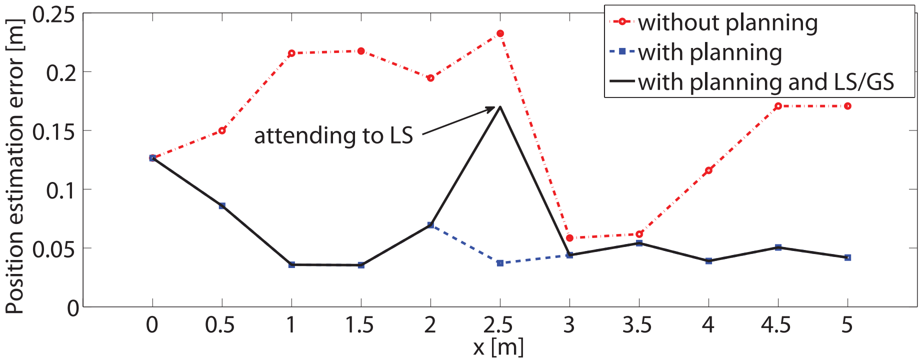

Fig. 3.17: Robot position estimation errors using constant view directions without planning (red, dash-dot line with point markers), using planned view directions (blue, dashed line with square markers), and using planned view direction and considering LS/GS as well (black solid line).

The active vision system calculated the robot position using the optimal view directions and stereo-vision triangulation. The odometry data is taken as the ground truth, which is very accurate on ideal indoor floor. However, systematic errors mainly due to coordinate transformation errors are not considered here. Fig. 3.17 visualizes the position estimation results. $\Omega = 0°$ was used as reference for the position estimation to show the improvement of estimation accuracy using active vision. The red, dash-dot line with point markers shows the estimated robot positions without active vision, while the blue, squared-dashed line shows the results with the planned view directions. The robot position estimation result using active vision is better than that using the constant view direction.

Taking GS and LS into consideration, the robot also computed the GS value from the three consecutive input images during the position estimation. If the GS value in the current environment was higher than a threshold defined empirically, set to 0.1 in this experiment, the robot attended to the LS at the next time step. Fig. 3.18 illustrates the GS value at each time step. At position $(1.969, 1.19)$ m the GS value was higher than the

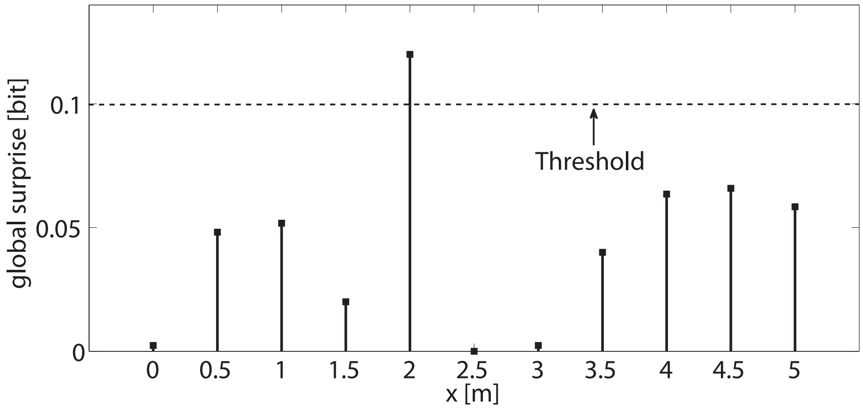

Fig. 3.18: GS values during robot locomotion. The threshold was set to 0.1 bit.

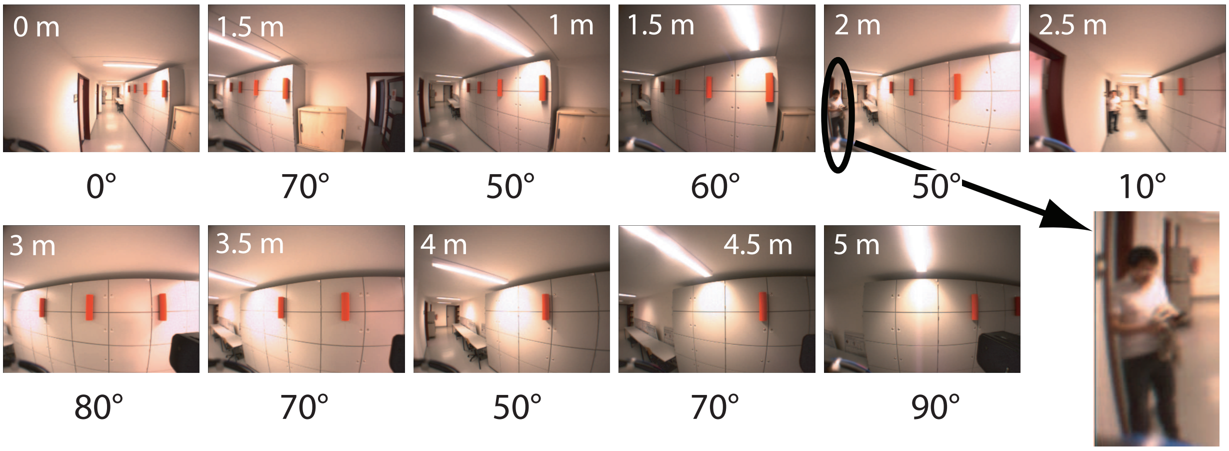

Fig. 3.19: Image sequence captured at different locations (x-coordinate on the images) for the self-localization task taking visual stimuli (LS and GS) into consideration. The numbers under the images denote the camera view directions with respect to the locomotion direction.

threshold. Then, the robot attended to the LS at position $(2.452, 1.19)$ m. Fig. 3.19 shows the view directions for self-localization tasks with consideration of environment uncertainty. The human that suddenly entered into the FOV was attended to by the robot.

Fig. 3.17 also illustrates the robot position estimation result considering surprising events in a black, solid line. Without considering surprising events, the robot planned its view direction towards the task-relevant information – the landmarks and the position estimation error is about 0.001 m in x-direction and 0.0417 m in y-direction. With consideration of a surprising event at the previous time step – a human entering into the FOV – the camera tried to locate the surprising event in the environment and changed its view direction from 50° to 10°. The position estimation errors in x- and y-direction are 0.1538 m and 0.0441 m. The self-localization task is impaired by considering GS and LS at this moment.

3.4.3 Discussion

Although the self-localization task is impaired by considering GS and LS, the surprising event in the environment should also attract the robot's attention considering environment exploration and safety. Through interaction between LS and GS, a robot has the chance to perceive the not-explicitly task-relevant stimuli while performing its primary task.

Both metrics are indispensable. Since LS indicates the relative importance of an image location in comparison to other locations, without GS the robot has to compute LS consistently, which is a waste of computation capacity. Moreover, if the robot always attends to the LS, the robot task is ruined. Without LS, the robot does not know the surprise origin. Not attending towards the LS means that, at the very least, an information loss occurs. Worst of all, the robot may become damaged. For the situation that the robot has a primary task and is also envisioned to be aware of its operation environment, these two metrics are combined to give the robot the possibility to choose an adaptive behavior, and detect and track the surprising event without weakening its primary task all the time. A high GS indicates a high environment uncertainty and alerts the robot system to attend to the current LS maximum, which has probably caused the uncertainty increase. Through interconnections of LS and GS, the sensitivity of robot systems to the operating environment is highly improved while preserving primary robot tasks.

The main limitation is that a quantitative evaluation is still missing. However, under different task loads, in different environments (indoor/outdoor, familiar/strange surroundings), different subjects would also behave differently when facing the same unexpected stimuli. Therefore, this ability to attend to unexpected stimuli in an uncertain environment is not a question about "right", "wrong", or "how much", but concerning whether a robot can or cannot have this kind of cognitive ability. Therefore, although a quantitative evaluation is not easy, the reasonability and necessity of the existence of GS and LS to reduce information loss and computation waste are experimentally demonstrated. This pre-attention ability enables a high sensitivity to the environment and benefits a concurrent world exploration during robot operation. The more sensitive the perception is conducted, the more accurately and quickly a response can be achieved.

3.5 Summary

Conventional robot applications emphasize task-relevant information explicitly. Task-irrelevant (not directly task-relevant) stimuli in the environment are commonly ignored, which are, however, a critical point for perception of unexpected events and a guarantee of the robot's working order in an uncertain environment. This chapter addresses this bottom-up perception problem during a robot performing a task by solving two main issues: definition of task-irrelevant stimuli and determination of a reasonable and economical time point to attend to task-irrelevant stimuli considering environment dynamics. Two metrics, local surprise and global surprise, are defined in this context.

Local surprise combines static saliency and temporal novelty in the 2D image space using an information-based approach, while global surprise emphasizes the dynamic changing of the robot's operating environment. A high global surprise indicates a high environment uncertainty and alerts the robot system to attend to the current local surprise maximum, which has probably caused the uncertainty increase. Both metrics are complementary to each other and can also be separately applied in different contexts.

Through interconnections of local surprise and global surprise, the sensitivity of robot systems to the operating environment is highly improved while preserving primary robot tasks. These results open up the possibilities and future directions of developing cognitive technical systems, which can also concern their own existence and safety issues in addition to scheduled tasks. Limitations are mainly located in expensive quantitative evaluations and the need for a more sophisticated means of environment monitoring, which are subject to future work.

4 Integrated Approaches to Top-Down and Bottom-Up Attention Control

4.1 Introduction

Considering goal-directed robotic applications, visual attention has become a popular topic of robotics research to deal with the limited processing capability and the real-time requirement of technical systems, especially autonomous and/or mobile robots. A pure bottom-up attention selection is neither sufficient nor efficient for task-relevant information enhancement. Goal-directed guidance of gaze control based on coordinated task and stimulus parameters plays a key role in robot attention development.

Currently, the related works about visual attention in the robotics domain can be mainly divided into two different categories: computer vision aiming at perfecting bottom-up attention selection models in the 2D image space, and task-oriented robotics applications in the 3D task space. The former category usually ignores robot characteristics such as locomotion in the 3D space, the real-time requirement, or the goal-directed evaluation, while the latter commonly deals with specific tasks and uses simple features in structured work spaces to reduce system complexity. In addition, to search for task-relevant target objects, most works are tightly based on a costly offline training procedure. An optimal representation of a target object is learned from the training procedure, which is, however, not always the best representation of the current environment.

Two complementary visual attention selection strategies are proposed, to deal with the aforementioned problems and realize a complete robot attention system concerning top-down information, bottom-up stimuli, and robot behaviors such as active vision control and locomotion. The combination and coordination of top-down and bottom-up mechanisms especially in a changing environment due to robot mobility are explored here.

The first strategy is a variation of top-down biased bottom-up (*TBB*) attention selection, considering target objects with a similar appearance. In TBB, the conventional offline training of task relevant top-down information is replaced by an online extraction of top-down information of the first recognized target object. Successively, adaptation of model parameters to the changing environment using a *Kalman-filter* (KF) is developed, which shows improved efficiency in terms of fewer necessary fixations.

The second strategy is autonomous switching between top-down and bottom-up attention selection (*TOB*, abbreviated for Top-down Or Bottom-up), considering target objects with different appearances. In TOB, autonomous switching between a pure top-down and a pure bottom-up attention selection mechanism is proposed for the first time, which enables a vision-guided mobile robot to be “autonomous” in the aspect of visual behavior planning. Moreover, three different internal robot modes – exploring, searching and operating – are also considered. The visual behavior is then properly adapted to the internal robot modes.

These two strategies complement each other. TOB fills the gap in TBB for the situation in which totally different targets are searched for while contexts vary. Moreover, integrating TBB into TOB can improve the overall task performance.

This chapter is organized as follows. In Section 4.2 and 4.3, attention selection strategies TBB and TOB are presented. The experimental results are presented and discussed, respectively. A summary is given in Section 4.4.

4.2 Top-Down Biased Bottom-Up Attention Strategy (TBB)

In this section, the scenario is considered, in which several objects of the same type are located in the environment and searched for. The central problem is how to promote task-relevant objects using bottom-up attention without a previous training process. A variation of the top-down biased bottom-up attention selection strategy is proposed to realize object representation on the one hand and adaptation of the representation to the changing environment on the other.

4.2.1 State of the Art

Bottom-up attention selection has been applied as a front-end for object detection in a few works. In [197], a marriage between bottom-up attention based on a saliency map and SIFT feature-based object recognition is applied to demonstrate that bottom-up attention can contribute to object detection and reduce computation time. This paper serves as a basis for attention-based object detection. An improved version considering similarity transformations for object recognition is proposed in [43]. However, as mentioned in those works, some points regarding a complete system need to be improved, for instance, top-down feedbacks and foveated vision.

If the features of the searched target object are known, top-down information can be used to bias the attention selection, conventionally named top-down biased bottom-up attention. The effect of attentional weighting of a target-defining dimension has been investigated in cognitive psychological and neuroscientific studies [149, 196]. When computing a bottom-up saliency map, weighting the features contained in the target objects can accelerate the searching process. A few works have assigned weights for top-down and bottom-up attentional signals and conducted offline learning to achieve the optimal value of the weights for different feature dimensions such as color, orientation, and intensity [9, 22, 57, 199], considering maximized target detection speed [135], context sensitivity [71, 153], and color invariance [128]. In [135], target detection speed is maximized, defined as the ratio between the strength of the signal detecting the target over that detecting the distracting background, such that the weights between top-down and bottom-up attentional influences are optimized. By now, a previous offline training for the target object has become an inevitable prerequisite. Common offline learning processes are conducted using optimization algorithms [21, 22, 135], *neural network* [153, 180], or *reinforcement Learning* [142].

Without previous training, top-down information can be acquired from the first input image containing the target object, such as for object tracking in [55] or for object recognition in [47]. However, adaptation of the top-down information has not yet been applied. If the target object in the first input image has a different appearance than that in the other images, the detection in the following images will probably fail.

Furthermore, active multi-focal camera systems with peripheral vision and foveal vision or active zooming cameras aiming at assembling visual attention behavior have been developed [18, 51, 189]. Only limited functions such as saliency map computation and saccades as well as fixation on salient objects are currently available. Moreover, attention systems are usually studied decoupledly. Few works have applied concurrent locomotion or manipulation.

4.2.2 Model of TBB

Consider a scenario that a robot is assigned a task to bring four beer mugs lettered with "Munich". The target objects can be in different colors and forms. The only feature in common is the letters on them, which cannot be directly used in bottom-up attention models. Here, it is desired to avoid conventional offline training, which consists of capturing images containing target objects and manual selection of the target objects from the background. In this system, a robot is given a sample image of a certain kind of target object and can start to search for all the target objects in a room. A similar approach to acquire a sample image is described in [47]. It is not wise to use the information in the sample image directly, since the environment in the sample image can be different from the one in which the target objects are searched for.

A variation of top-down biased bottom-up attention selection, TBB, is proposed. Once an object is recognized as the target object, the bottom-up attention model is adapted to the current environment, using the top-down information extracted from this target object. A KF is used here to estimate the model parameters based on the previous knowledge and the current measurement. Moreover, bottom-up attention is applied to a wide-angle stereo camera to select a sequence of fixation points. Successive snapshots of high foveal resolution using a telephoto camera enables highly accurate object recognition.

Fig. 4.1 illustrates the operating structure of a multi-focal vision system, searching for M target objects with similar appearances. Before detailed information processing, the vision system first scans the environment. A wide-angle stereo camera is used to acquire the rough information due to its wide FOV. Bottom-up attention selection is computed on the low-resolution wide-angle image to predict potentially interesting objects, the target object candidates, at first glance. On the saliency map, thresholds T_{min} and T_{max} for the grayscale value of each pixel are set, to achieve a binary map. Based on this binary map, an object map consisting of target object candidates is constructed. In the object map, the candidates are numbered in an order that the more salient a candidate in the saliency map is, the earlier this candidate is processed in detail, to ensure that the most likely object candidate has the highest priority if the time condition is critical. Since object recognition is highly dependent on image resolution, object recognition is executed on the telephoto images. A telephoto camera with high resolution focuses on and processes the previously selected areas consecutively. This saccade/fixation behavior is facilitated by a

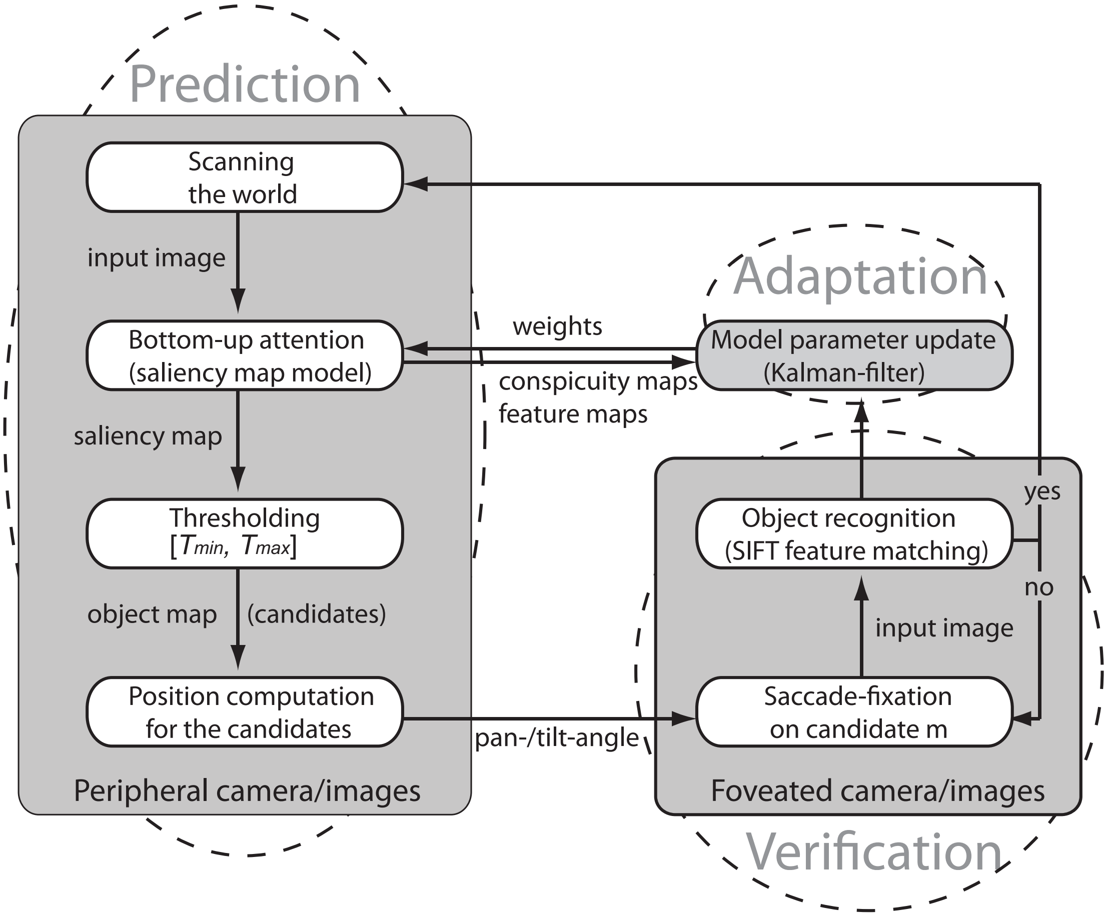
Prediction
Scanning the world
input image
Bottom-up attention (saliency map model)
saliency map
Thresholding [T_{min}, T_{max}]
object map
(candidates)
Position computation for the candidates
Peripheral camera/images
weights
conspicuity maps
feature maps
Adaptation
Model parameter update (Kalman-filter)
pan-/tilt-angle
Object recognition (SIFT feature matching)
yes
no
input image
Saccade-fixation on candidate m
Foveated camera/images
Verification

Fig. 4.1: Overview of the TBB model consisting of prediction, verification and adaptation.

pan-tilt platform. Once a candidate is verified as a target object, the bottom-up attention selection model parameters are newly estimated using the top-down information extracted from this object. The parameter adaptation to environments is accomplished online by using a KF. No previous training is needed and the whole process is more efficient in this perception-verification-action loop.

It is worth mentioning that the target objects are not assumed to be salient. The vision system starts searching in the most salient positions. If the salient positions firstly determined in the object map do not contain any target object, the threshold of the saliency value for determination of the binary map is reduced to the next $[T_{min}, T_{max}]$. The most salient positions only have a higher priority to be attended to than the other positions.

For the bottom-up attention selection, a standard computational model, the saliency map model proposed in [84], is used. Since the object recognition algorithm is not the focus of this strategy, *Scale-Invariant Feature Transform* (SIFT) feature matching between the sample image and the high-resolution images is chosen to verify whether a pre-selected attentional allocation contains a target object. The saliency map model and the SIFT algorithm are implemented using the CUDA technology on the multi-GPU platform, which highly accelerates image processing. Further details about the multi-GPU implementation of bottom-up attention selection can be found in Appendix B.

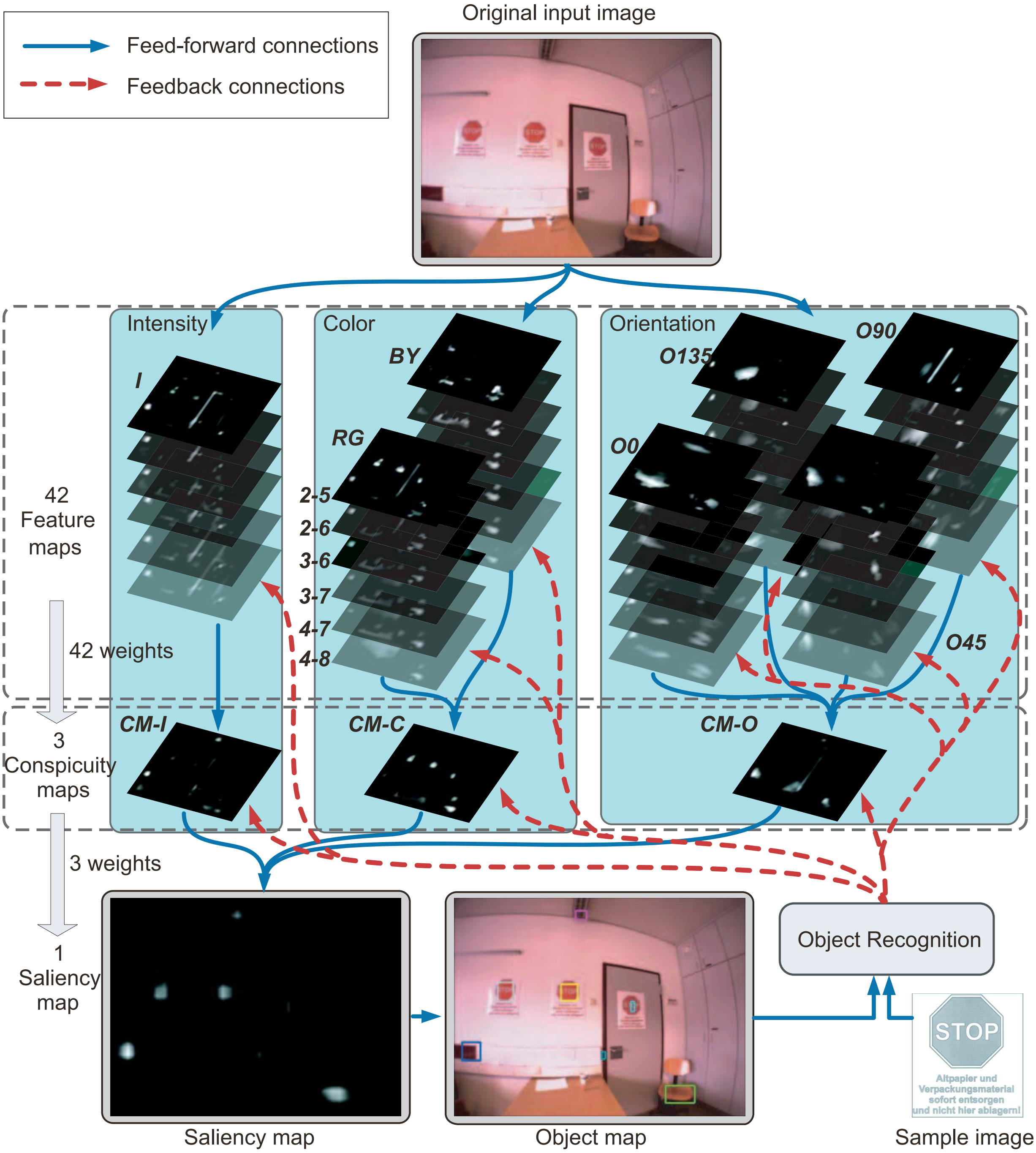

Fig. 4.2: Saliency map computation illustrated in the feed-forward connections and model parameter update illustrated in the feedback connections.

Bottom-Up Attention

For the bottom-up attention, the saliency map model proposed in [84] is applied in this model, which is illustrated by the feed-forward connections in Fig. 4.2. As introduced in Chapter 3, an input image is sub-sampled into a dyadic Gaussian pyramid with 9 scales in three channels (intensity (I), orientation (O) for $0°, 45°, 90°, 135°$, opponent color (C) in red/green (RG) and blue/yellow (BY)). Then, center-surround differences are calculated for the images in the Gaussian pyramids between the fine scale $\{2, 3, 4\}$ and the coarse scale $\{5, 6, 7, 8\}$. In this phase, 42 feature maps (FM) are generated in which the salient pixels with respect to their neighborhood are highlighted. Using across-scale combinations the FMs are combined and normalized into a conspicuity map (CM) in each channel. The saliency map is a linear combination of the CMs. The bright pixels are salient and interesting pixels with respect to their backgrounds. If no previous knowledge is available, the saliency map predicts purely bottom-up attention selection.

Model Parameter Definition

To combine top-down information into the saliency map model, 45 weights are defined in the saliency map model, which represent the importance of the contributions of 3 CMs and 42 FMs in building a saliency map. They are divided into 8 groups, namely the CM group containing 3 maps $CM\text{-}I$, $CM\text{-}C$ and $CM\text{-}O$, as well as 7 FM groups: $FM\text{-}I$, $FM\text{-}RG$, $FM\text{-}BY$, $FM\text{-}O_0$, $FM\text{-}O_{45}$, $FM\text{-}O_{90}$, and $FM\text{-}O_{135}$, containing 6 center-surround difference maps between different scales (2-5, 2-6, 3-6, 3-7, 4-7, 4-8) each. A weighting vector $\boldsymbol{w}$, representing the 45 weights for 45 maps in the model, can be formulated as follows:

$$\boldsymbol{w} = \begin{pmatrix} w_{CM\text{-}I} \\ w_{CM\text{-}C} \\ w_{CM\text{-}O} \\ w_{FM\text{-}I[\text{2-5}]} \\ \vdots \\ w_{FM\text{-}RG[\text{2-5}]} \\ \vdots \\ w_{FM\text{-}BY[\text{2-5}]} \\ \vdots \\ \vdots \end{pmatrix}. \tag{4.1}$$

If there is no top-down information available, which means the model works as bottom-up, $\boldsymbol{w}$ is a vector of ones. If top-down information should be integrated into the saliency map, the components of $\boldsymbol{w}$ will be adjusted to certain values to present the characteristics of the task-relevant information.

As shown in Fig. 4.2, once a candidate region m in the object map is verified as a target object, this candidate's coordinate information is fed back to the 45 maps. Then, the corresponding region in these maps can be ascertained. An average gray value V in those regions in each map is reckoned, to identify how much each map (CM or FM) contributes to the saliency of this location. Fig. 4.3 illustrates the conspicuity maps $CM\text{-}I$, $CM\text{-}C$, and $CM\text{-}O$ of an input image. The pixels limited by the rectangles are involved in the contribution computation. The contribution (c) of each map can be computed through Eq. (4.2) and (4.3).

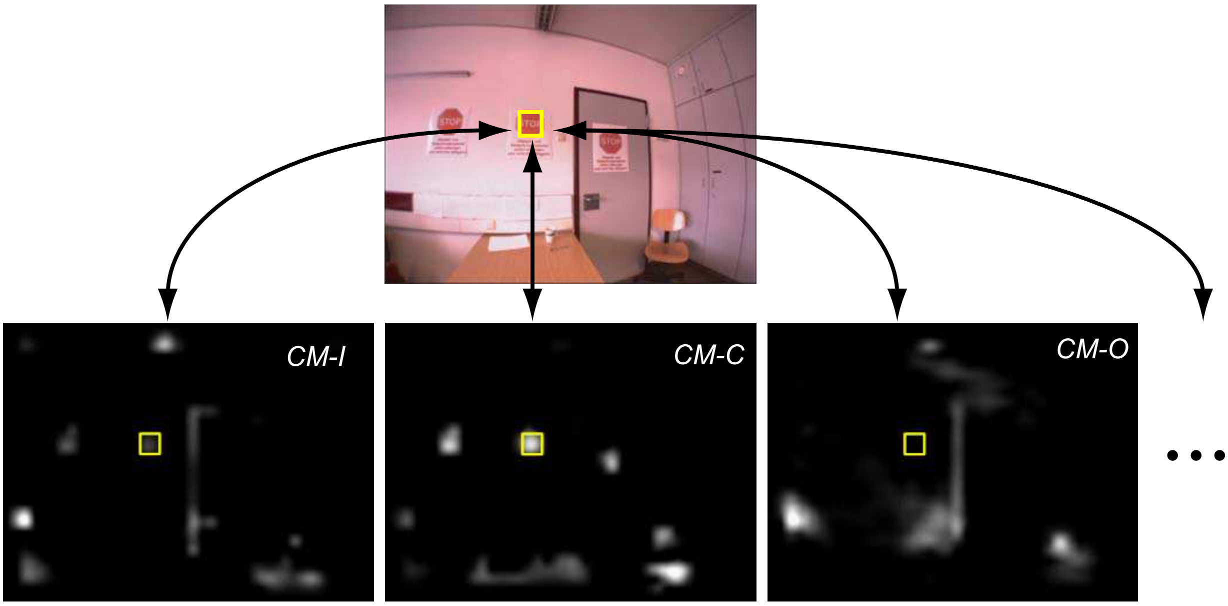

Fig. 4.3: Contribution of CMs in building a salient image region, illustrated by the squares. Upper image: the object map; Lower images: the conspicuity maps in I-, C-, and O-channels from left to right.

For CMs

$$c_{CM\text{-}i}(n) = \frac{V_{CM\text{-}i}(n)}{\sum\limits_{i} V_{CM\text{-}i}(n)} \quad i \in \{I, C, O\}; \tag{4.2}$$

For FMs

$$\begin{aligned} \text{Intensity:} \quad & c_{FM\text{-}I[j]}(n) = \frac{V_{FM\text{-}I[j]}(n)}{\sum\limits_{j} V_{FM\text{-}I[j]}(n)}, \\ \text{Color:} \quad & c_{FM\text{-}C[j]}(n) = \frac{V_{FM\text{-}C[j]}(n)}{\sum\limits_{C}\sum\limits_{j} V_{FM\text{-}C[j]}(n)}, \\ \text{Orientation:} \quad & c_{FM\text{-}O[j]}(n) = \frac{V_{FM\text{-}O[j]}(n)}{\sum\limits_{O}\sum\limits_{j} V_{FM\text{-}O[j]}(n)}, \end{aligned} \tag{4.3}$$

where $C \in \{RG, BY\}$, $O \in \{O_0, O_{45}, O_{90}, O_{135}\}$, and $j \in \{2\text{-}5, 2\text{-}6, 3\text{-}6, 3\text{-}7, 4\text{-}7, 4\text{-}8\}$.

In system initialization, to build a saliency map, three CMs are weighted equally, namely $w_{CM\text{-}i} = w_{CM\text{-}c} = w_{CM\text{-}o} = 1$. To build a CM, the different features are also weighted equally. Therefore, the sum of weights remain constant as follows:

$$\begin{aligned}\sum_i w_{CM\text{-}i,k} &\equiv 3,\\ \sum_j w_{FM\text{-}I[j],k} &\equiv 6,\\ \sum_C \sum_j w_{FM\text{-}C[j],k} &\equiv 12,\\ \sum_O \sum_j w_{FM\text{-}O[j],k} &\equiv 24,\end{aligned} \tag{4.4}$$

where k is the current time step.

If an interesting area in the current saliency map is selected as target object candidate and also confirmed to be a target object, the more a CM or an FM contributes to building the current saliency map in this area, the more weight this map should be assigned for the next step, such that the characteristics of the target object are enhanced in the next saliency map. The weights of maps for the next saliency map computation are proportional to the contributions of the maps in the current step, formulated as follows:

$$\begin{aligned}\text{CMs:}\quad & w_{CM\text{-}i,k+1} = 3 \times c_{CM\text{-}i,k};\\ \text{FMs:}\quad & w_{FM\text{-}I[j],k+1} = 6 \times c_{FM\text{-}I[j],k},\\ & w_{FM\text{-}C[j],k+1} = 12 \times c_{FM\text{-}C[j],k},\\ & w_{FM\text{-}O[j],k+1} = 24 \times c_{FM\text{-}O[j],k}.\end{aligned} \tag{4.5}$$

Parameter Adaptation Using Kalman Filtering

Eq.(4.5) means that the adaptation of the new weights for next cycle is completely according to the top-down information in the current cycle. In other words, the system learns only once from the current result. However, instead of "one-shot" adaptation, the model parameter is updated not only based on the latest measurement but also considering previous measurements.

Investigating a sequential attentional task, a phenomenon called *attentional priming* is reported [115]. In visual search tasks, trial-to-trial repetition of a target-defining feature or target location substantially reduces the reaction time. Two arguments can be implied [130]: First, based on past experience, a probabilistic model of the environment can be dynamically constructed by the perceptual system; second, control parameters of the attentional system are tuned so as to optimize the performance under the current environmental model. Dealing with the attentional priming phenomenon, in [130] a probabilistic model of the environment is proposed which is updated after each trial. A memory constant is introduced to represent how much the past experience affects the current result. Based on this only free parameter, results from diverse experimental paradigms are explained.

For a mobile robot, the background and the light conditions are always changing. The changes due to the movement can be continuous, while the changes due to entering or facing a totally new environment can be very sudden. Therefore, the parameter update cannot only be based on the latest measurement, since it will be difficult to find a new target if the last measurement is unique. Moreover, the more recently the measurement was taken, the more representative the contributions/parameters according to the current environment are.

KF is an efficient recursive filter that estimates the state of a dynamic system from a series of incomplete and noisy measurements. Here, the memory constant proposed in [130] is replaced by using the Kalman gain K_k, which evolves dynamically in the correction phase in the Kalman filtering and bias the weights between the past experience $\boldsymbol{w}_{k-1}$ and the new measurement $\boldsymbol{c}_k$.

In this case, the system state is the weight vector of the bottom-up attention model at time k:

$$\boldsymbol{x}_k = \boldsymbol{w}_k = \begin{pmatrix} w_{CM\text{-}I,k} \\ w_{CM\text{-}C,k} \\ w_{CM\text{-}O,k} \\ \vdots \\ \vdots \end{pmatrix}, \tag{4.6}$$

where $\boldsymbol{x}_k$ is assumed to be constant for one kind of object. The system equation can be formulated as follows:

$$\boldsymbol{x}_k = \boldsymbol{A} \cdot \boldsymbol{x}_{k-1} + \boldsymbol{z}_{k-1}, \tag{4.7}$$

where $\boldsymbol{z}_{k-1}$ is process noise and the state transition matrix $\boldsymbol{A}$ is a unit matrix of a dimension of 45×45. There is no control input in this case.

The measurement is the contributions of CMs and FMs:

$$\boldsymbol{y}_k = \boldsymbol{c}_k = \begin{pmatrix} c_{CM\text{-}I,k} \\ c_{CM\text{-}C,k} \\ c_{CM\text{-}O,k} \\ c_{FM\text{-}I[2\text{-}5],k} \\ \vdots \\ c_{FM\text{-}RG[2\text{-}5],k} \\ \vdots \end{pmatrix}, \tag{4.8}$$

with

$$\boldsymbol{y}_k = \boldsymbol{H} \cdot \boldsymbol{x}_k + \boldsymbol{v}_k, \tag{4.9}$$

where $\boldsymbol{v}_k$ is the measurement noise and the measurement matrix and

$$\boldsymbol{H} = \begin{pmatrix} \frac{1}{3}\boldsymbol{I}_3 & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \frac{1}{6}\boldsymbol{I}_6 & \boldsymbol{0} & \vdots \\ \vdots & \boldsymbol{0} & \frac{1}{12}\boldsymbol{I}_{12} & \boldsymbol{0} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \frac{1}{24}\boldsymbol{I}_{24} \end{pmatrix}, \tag{4.10}$$

where $\boldsymbol{I}_n$, $n \in \{3, 6, 12, 24\}$, is a unit matrix with n-dimension.

$\boldsymbol{z}_k$ and $\boldsymbol{v}_k$ are assumed to be zero mean Gaussian white noise with covariance matrices $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$ obtained empirically.

The main contributions of this strategy based on the prediction-verification-adaptation loop are as follows:

- This is a general concept which can be applied for various objects and scenarios. The top-down information is extracted from the detected target objects in the current scenario. Therefore, no previous training is necessary.
- The KF-aided model parameter tuning enables autonomous adaptation to the operating environment that changes along with the robot locomotion.
- The proposed adaptation strategy, the implementation on multi-GPU platform, and the multi-focal camera system facilitate an efficient and high-speed object detection.

4.2.3 Performance Evaluation

Experiments using the ACE robot were conducted for performance evaluation. An object detection task was applied. Since object recognition is not the focus of this strategy, for simplicity, posters with "emergency exit" written on them were chosen as target objects. Four posters were hung around the initial robot position. Because of the low resolution and the limited effective range of the wide-angle stereo camera, the average distance between the posters and the robot was 3 m. The robot rotated 90° after investigating one side of the room. Four rotations were needed to accomplish the object detection task.

Four different strategies are considered: exhaustive searching without attentional pre-selection (abb. *B0T0K0*), purely bottom-up attentional pre-selection (abb. *B1T0K0*), top-down biased bottom-up attentional pre-selection but without KF estimation (abb. *B1T1K0*), and the proposed TBB (abb. *B1T1K1*). The symbol definition is shown in Tab. 4.1. The "0" in the symbols indicates "without", while "1" indicates "with". To be consistent with the other strategies, the proposed TBB is referred to as B1T1K1.

Symbol	Bottom-up	Top-down	KF
B0T0K0	−	−	−
B1T0K0	+	−	−
B1T1K0	+	+	−
B1T1K1 (TBB)	+	+	+

Tab. 4.1: Symbol definition for different strategies. "+": with; "−": without.

The object detection result using B1T1K1 is shown first. Then, the performance enhancement of Kalman filtering is discussed by comparing strategies B1T1K1 and B1T1K0. After that, a comparison of four strategies in terms of detection rate and computation time is conducted.

Object Detection Using Online Top-Down Information Update

The left two columns in Fig. 4.4 show the object maps and the respective saliency maps using B1T1K1. In the first image, eight target candidates in the image were selected using the initialized saliency map without top-down information. After the first candidate

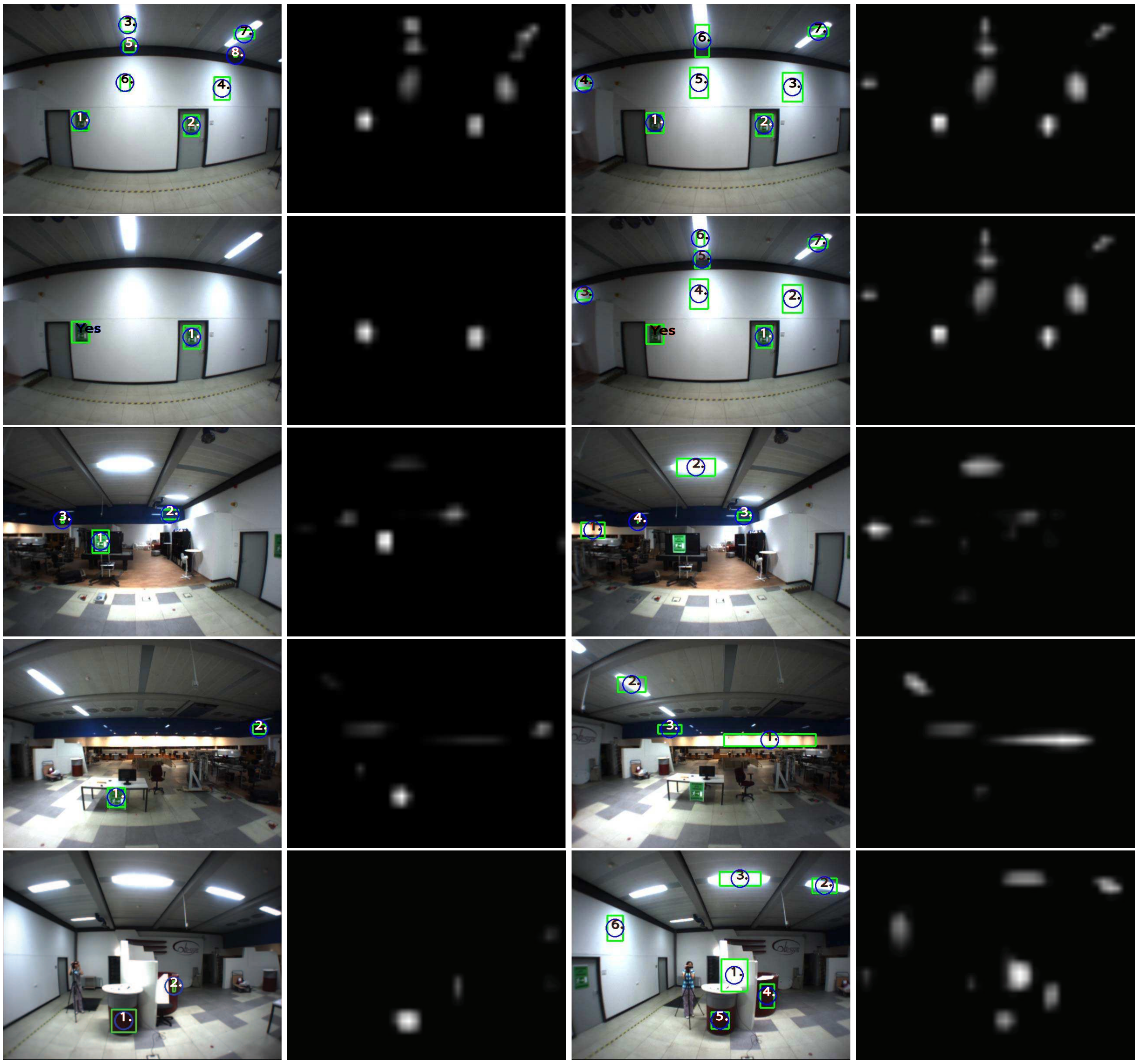

Fig. 4.4: Column 1: object maps predicted using B1T1K1; Column 2: saliency maps computed using B1T1K1; Column 3: object maps predicted using B1T0K0; Column 4: saliency maps computed using B1T0K0. Numbers on the object maps indicate the fixation sequence along with a descending saliency value of the selected image region candidates. "Yes" on the object maps indicate that an image region candidate contains a target object.

was fixated by the telephoto camera and recognized as a target, this candidate region was marked by "Yes" and the weight vector was adapted. Only two target candidates remained in the newly computed saliency map (in the second row) and were investigated further. For the following three totally different scenes (the last three rows) the target objects had always been selected for a detailed processing.

If B1T0K0 is used, shown in the right two columns in Fig. 4.4 for the first scene, all the seven candidates were processed in more detail. For the following three scenes, the target objects were not even selected for a saccade/fixation.

The sample image used for the object recognition is shown in the left-most image in

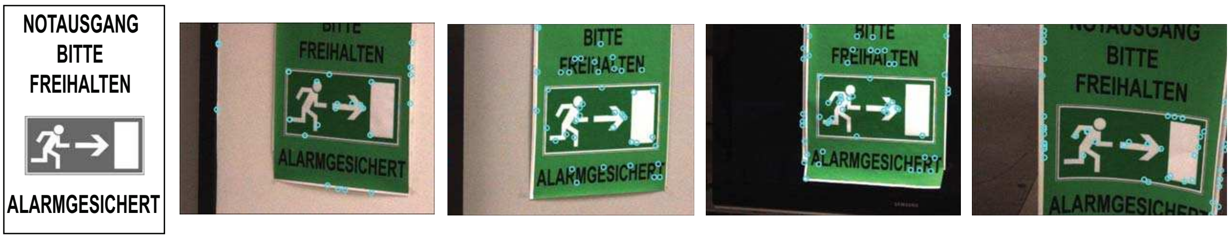

Fig. 4.5: The sample image of the target object used in the experiment (left) and four images of the target objects captured by the high-resolution telephoto camera during the experiment using B1T1K1. Blue circles: the matched SIFT feature points.

Fig. 4.5, . Since the proposed strategy should be general, a grayscale image was used which contains no top-down information such as color. The high-resolution images captured by the telephoto camera are also shown in Fig. 4.5. The blue circles are the matched SIFT features with the sample image. In each object recognition cycle, once the matched feature number is beyond the predefined threshold, it means a target object was detected. Otherwise, up to 10 SIFT feature extractions and matchings are computed to reduce the influence of noise.

Fig. 4.6 shows a new update of the thresholds T_{min} and T_{max} in building a new object map, if no target object was found in the previously selected task-relevant image region candidates (the left column). In the right column, the less salient image regions were investigated. The threshold update is defined in this experiment only once to save time and energy. It can be adapted to different tasks with different specifications, which require either time/energy minimization or maximization of the number of the detected target objects.

Performance Enhancement of Kalman Filtering

To show the performance of Kalman filtering, strategy B1T1K0 and B1T1K1 are compared here. Fig. 4.7 shows the changing of the weights for $CM\text{-}I$, $CM\text{-}C$, and $CM\text{-}O$. The weights were initialized to be 1. After a target object was recognized, the weights were updated. $w_{CM\text{-}C}$ increased from 1 to 2.980217. Respectively, $w_{CM\text{-}I}$ and $w_{CM\text{-}O}$ decreased. At time steps 2, 3, 4, and 7, target objects were detected. Using B1T1K1, $w_{CM\text{-}C}$ converged to about 2.99, illustrated by the lines with circular markers.

The lines without circular markers in Fig. 4.7 are the results using B1T1K0 using the same input images. Without Kalman filtering, the model parameter for the third fixation totally depends on the second measurement, which has a lower weight for $CM\text{-}C$ and a higher weight for $CM\text{-}I$. After a rotation of 90°, the target position was not chosen as the first candidate if KF was not used (see the right column in Fig. 4.8). In contrast, using B1T1K1, the target position was chosen to be first processed, marked with “1” in the object image (see Fig. 4.8, top left). In summary, the utility of the KF is one of the possibilities to find an efficient updated parameter value to represent the target object itself and the current environment by weighting the past experience and the new measurement.

Another example is illustrated in Fig. 4.9. A stop sign was searched for in this exper-

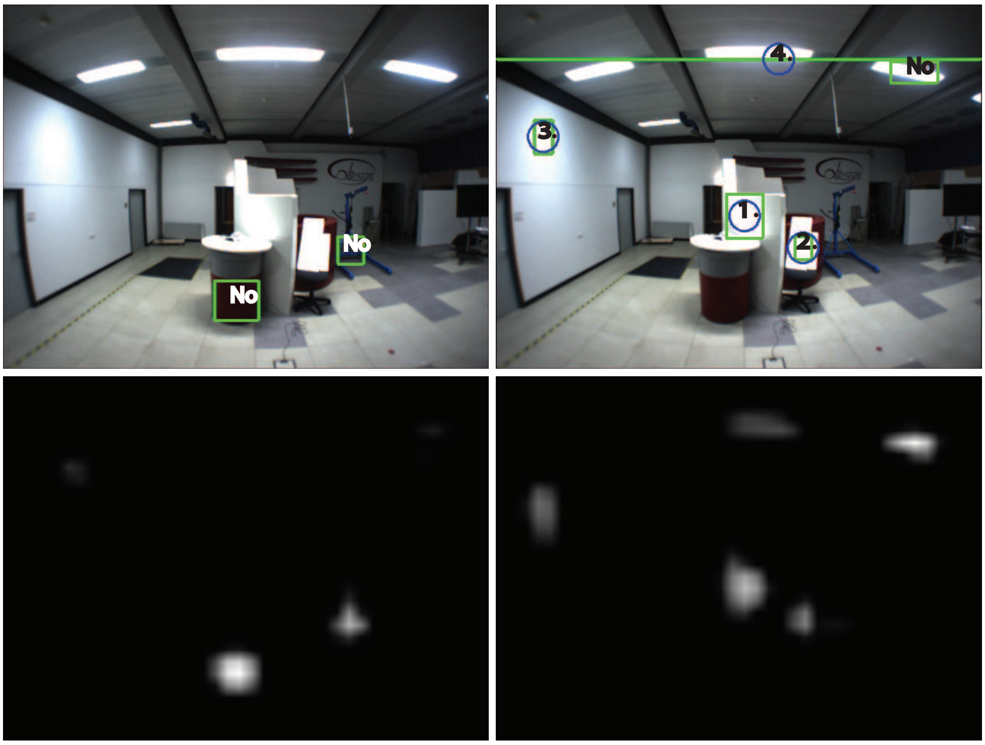

Fig. 4.6: Update of thresholds for object map. Left column: object map (upper) and saliency map (lower) computed using the initial thresholds; Right column: object map (upper) and saliency map (lower) using the updated thresholds. Numbers on the object maps indicate the fixation sequence along with a descending saliency value of the selected image region candidates. "No" on the object maps indicate that an image region candidate contains a target object.

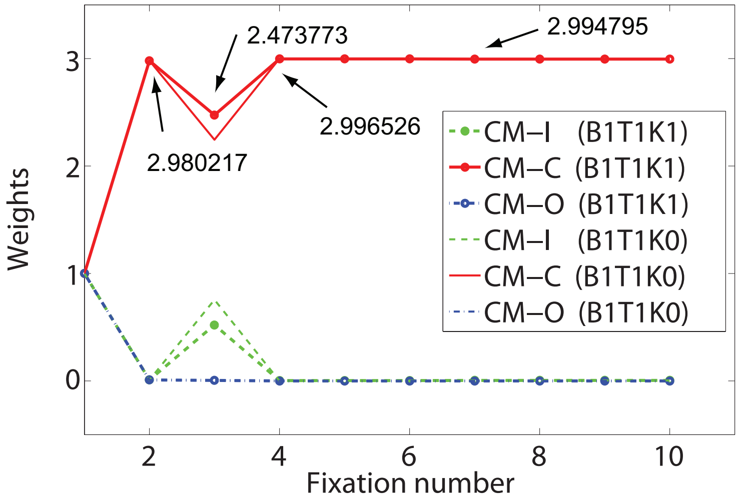

Fig. 4.7: The weights variation for *CM-I*, *CM-C*, and *CM-O* using and not using KF.

iment. The upper row shows three successive object maps, while the respective saliency

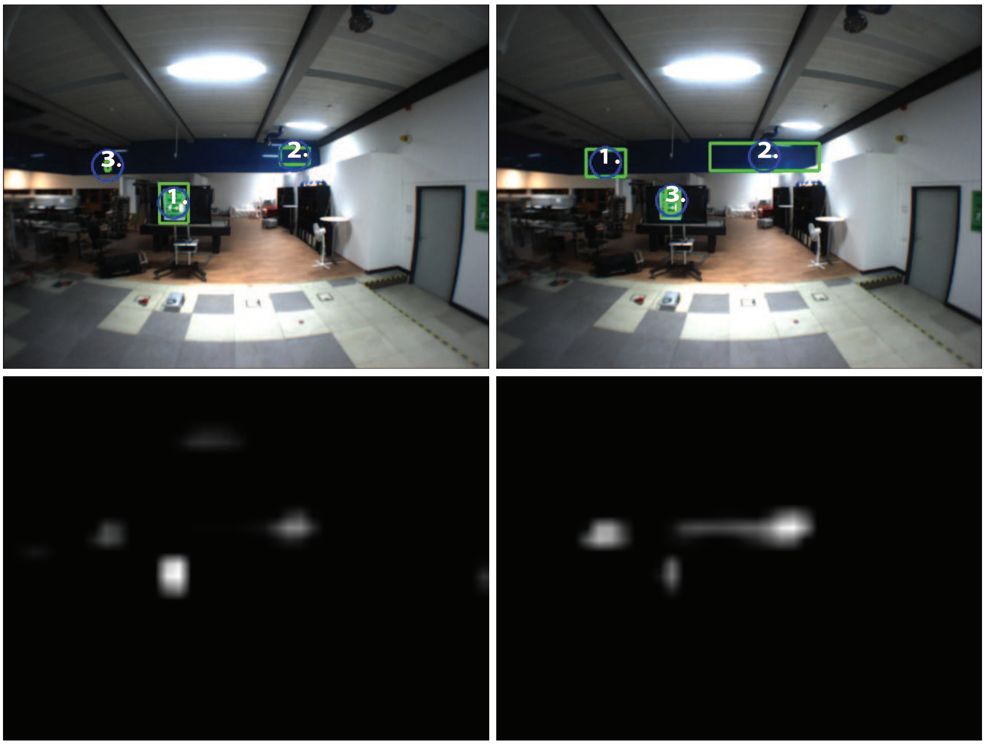

Fig. 4.8: Left column: object map (upper) and saliency map (upper) with Kalman filtering; Right column: object map (upper) and saliency map (upper) without Kalman filtering. Numbers on the object maps indicate the fixation sequence along with a descending saliency value of the selected image region candidates.

maps are shown in the lower row. In each image of column a and b, one sign was detected. Between two consecutive maps, a parameter update is conducted. The circles drawn in the object maps indicate the image region with a descending saliency value from a previous image to a current image, while the circles drawn in the saliency maps indicate the image region with an ascending saliency value. Through the KF-aided parameter update, the task-relevant regions are enhanced, while the task-irrelevant regions are inhibited.

Investigation of the Computational Cost

The performances of four strategies defined in Tab. 4.1 are compared in terms of the average detection rate and the approximate necessary fixation times for this task in Fig. 4.10. The performance was experimentally evaluated in three different scenarios. In each scenario three to five experiments were conducted. The detection rate is defined as the ratio of the detected and actual target object number M in the environment. In this comparison, the adaptation of T_{min} and T_{max} is not considered.

In B0T0K0, the telephoto camera would have to scan the whole environment and process the object recognition for each input image. Therefore, more than 500 fixations of the telephoto camera would be needed, which indicates a high computational cost. If the target objects are not captured completely in a telephoto image, the recognition could fail, which causes a detection rate of approximately 80%.

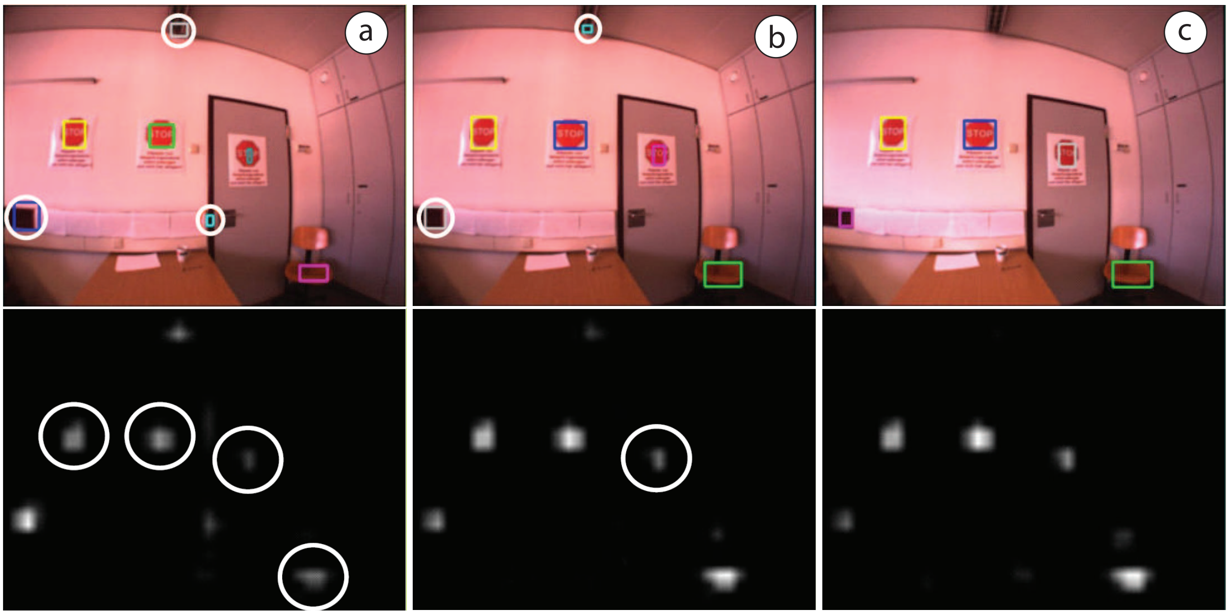

Fig. 4.9: From left to right: update of the object maps (upper) and the saliency maps (lower) aided by a KF. Rectangles: image regions selected as candidates; Circles in the object maps: image regions with a descending saliency value from a) to c), inhibited by the Kalman filtering; Circles in the saliency maps: image regions with an ascending saliency value from a) to c), enhanced by the Kalman filtering.

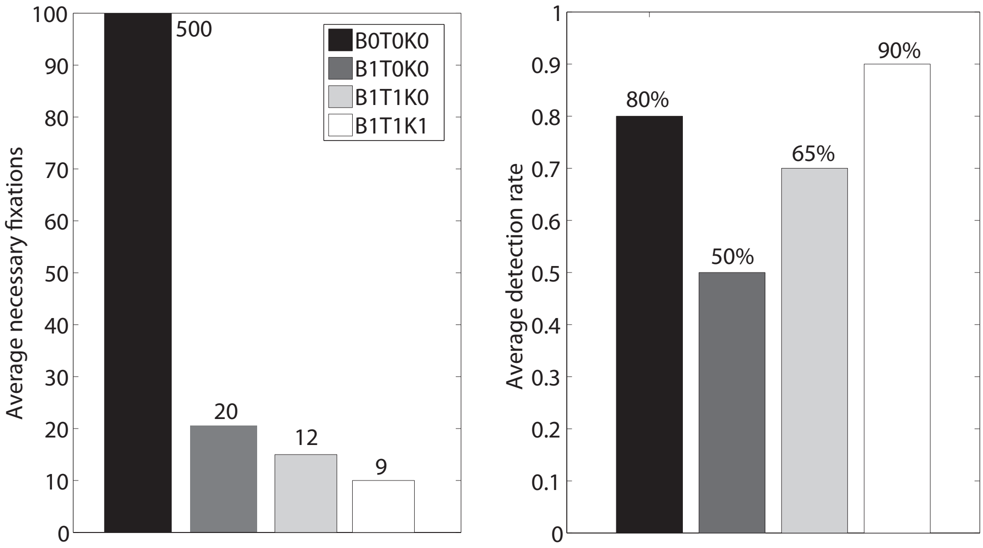

Fig. 4.10: Comparison of the approximate necessary fixations (left) and detection rates (right) for 4 target objects using four different strategies.

Only with the bottom-up attention model B1T0K0 is it difficult to detect all the target objects. The detection rate is only 50%, although the computational cost is low. Only the positions selected by the bottom-up model need to be focused on and be further processed.

Using B1T1K0 and B1T1K1, the detection rate is higher, namely about 65% and 90%. However, without KF, the weights vary strongly after each recognition, causing a difficult selection for next step or that the target object is selected but not numbered to be firstly processed, if the top-down information totally depends on the last measurement. Aided by KF, the telephoto camera has only fixated nine times in 3D room to detect all the four target objects.

Process	Computation time	Mechanical time
Initialization	T_0	T_1
Saliency map	$T_2(n_1+1)$	
SIFT	$T_2(n_2 \cdot n_3)$	
Saccade		$n_2 \cdot T_3$
Robot motion		T_4
Sum	$T_1 + T_2(n_1 + 1 + n_2 \cdot n_3)$	$n_2 \cdot T_3 + T_4$

Tab. 4.2: Computational and mechanical time cost for object detection. The constants $T_0 + T_1 = 6$ s, $T_2 = 0.033$ s, $T_3 = 1$ s, $T_4 = 20$ s. n_1: total number of saliency maps computed using top-down information; n_2: total number of target object candidates; n_3: the average times for SIFT computation in one candidate region.

Tab. 4.2 shows the time cost for one experiment including both computation and mechanical times. n_1 indicates the number of the detected target objects. After a target object is detected, the saliency map will be computed with the updated weights. In addition to the first saliency map with equal weights, there are n_1 saliency maps computed. As mentioned in Appendix B, using cameras at 30 Hz, no time delay is noticed for saliency map computation and SIFT algorithm, since they are implemented on a multi-GPU platform. Therefore, $T_2 = 0.033$ s is taken for image capturing. The total number of the candidates predicted in the saliency maps, as well as the necessary fixation number, is denoted by n_2, while n_3 means the average times of SIFT computation and matching for one candidate. Here, n_3 is inversely proportional to n_1/n_2. The more target objects there are under the total target candidates, the shorter the average time for SIFT computation is. For each saccade a time delay of 1 s was manually added to stabilize the telephoto image. For the robot motion, 4 rotations of 90° cost $T_4 = 20\ s$ in total.

To sum up, the computation time will decrease if the total number of the candidates n_2 decreases and n_1/n_2 increases for the same number of target objects M. Using B1T1K1, an improvement in computational cost is achieved.

4.2.4 Discussion

Contributions

Repeated object detection is solved by integrating top-down information of the recognized target object into the bottom-up attention model, such that the most likely objects are promoted by the bottom-up attentional pre-selection. The approach proposed here is a general concept for object detection, which can be applied for various objects and scenarios. No previous training of model parameters is necessary. The model parameters can be adapted to the changing environment and tuned online. A KF facilitates the parameter

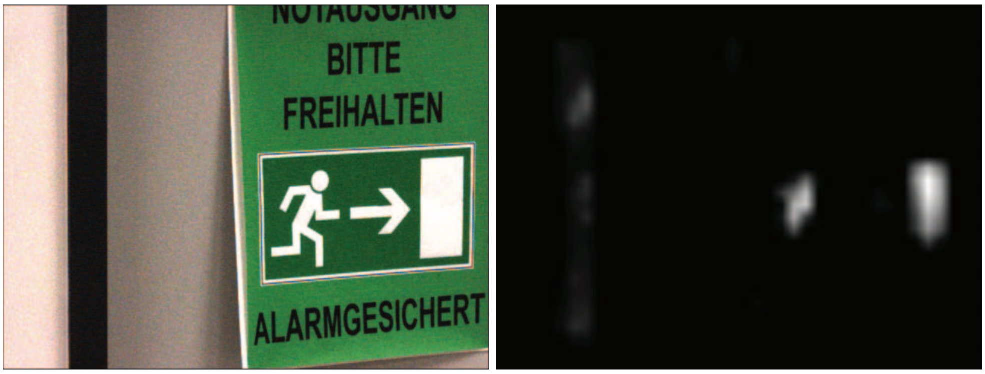

Fig. 4.11: Saliency map computed directly from a sample image. Left: sample image. Right: the respective saliency map.

estimation and provides a rational combination of the current measurement and the previous knowledge. Significant improvements in terms of accuracy, flexibility, and efficiency are achieved.

TBB(B1T1K1) can be regarded as an online training process. For the object recognition, a sample image about the target object is available. But the top-down information, which can be directly integrated into the bottom-up attention, is not available. From the sample image, the detailed SIFT features can be extracted for instance, but not the colors, intensity, and orientation needed in a bottom-up attention model. The sample image can be a gray image or an image containing a target object and a totally different background than the later searched environment. If the top-down information is directly extracted from the sample image, more detailed features may be detected which can not represent the whole object (see Fig. 4.11). To avoid manual selection of the target in the temporarily unknown environment, the attention system is initialized using a purely bottom-up attention. After the first target is found, the system works the same as the one with an initialization using top-down information. It is more costly than using a conventional top-down biased bottom-up strategy before the first target is found, but more flexible since no bottom-up feature related top-down information is needed. Moreover, if there is no obvious task-relevant objects in the current FOV, several fixations on other salient objects may be also informative.

Possible Overfitting

Fig. 4.12 illustrates the weights for feature maps at different scales in channel I, RG, BY, and O_0. The influence of scale on the attention model is investigated. As previously mentioned, the same kind of target objects could also have the similar scale with respect to its surroundings. It can be seen that the FM with scale 2-5 and 2-6 have the maximal weights in each channel, which indicates that it is meaningful to define weights for feature maps at different scales.

However, the different distances of the object positions may cause an overfitting due to the weighted scales. Possible solutions are zoom camera utilization or robot motion controlled to achieve the best view of the target [188].

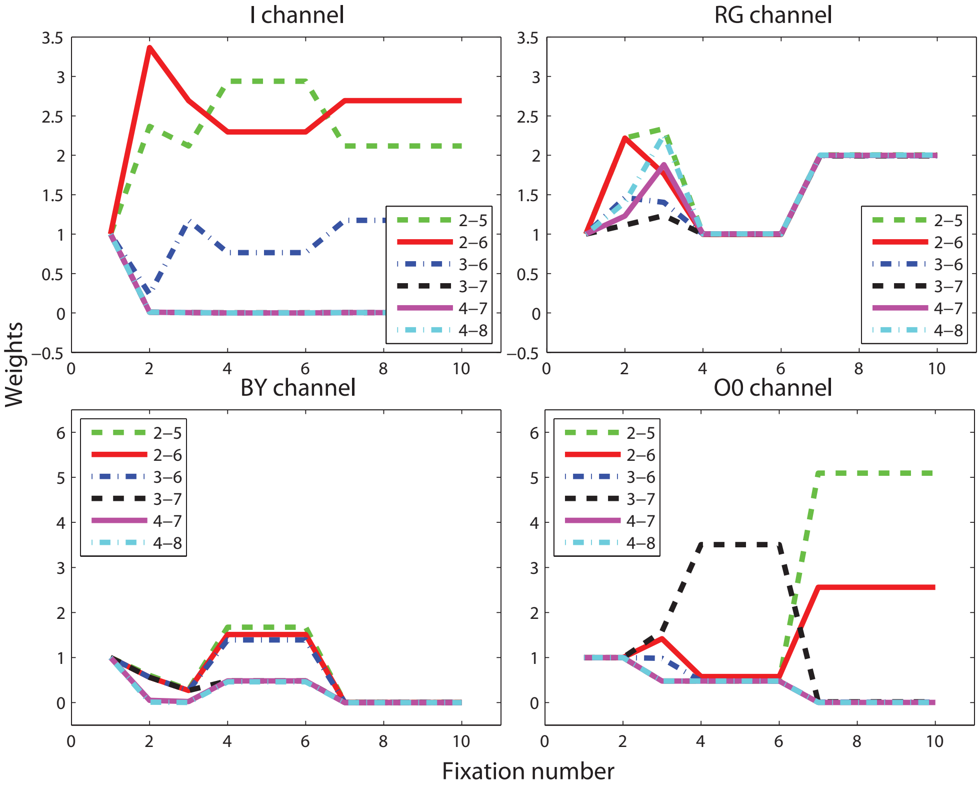

Fig. 4.12: Weights update of FMs at different scales in different channels: I-channel (top left), RG-channel (top right), BY-channel (bottom left), and O_{0°-channel (bottom right).

Experimental Setup

Occluded scenes are not considered in the experiments, since the matched SIFT feature number is the only crucial factor to verify if an image region contains the target object. If only a small part of the target object can be seen, the recognition rate decreases strongly. The background is complicated with distractors such as the blue ceilings and the red table which are also significantly salient with respect to colors, which can be ignored for humans attention, since the knowledge is available that the target objects can not be in the ceiling from the context. A context recognition approach can be integrated [172, 183].

Limitations

This efficient strategy can only be applied if the searched targets have a similar appearance, for example, a kind of objects is repeatedly searched for. If the targets change with the context, a top-down biased bottom-up strategy could impair the search process, since the top-down information does not converge.

As with other works using top-down biased bottom-up attention selection for object detection, there is no guarantee that the weightings for one object are unique to that object. A set of objects may be represented by the same weighting vector. An increase of the number of the feature dimensions in the bottom-up attention model could improve the performance but cannot solve this problem absolutely. The work could be improved by modeling the distractors, which are also fixated by the telephoto camera.

Currently, the scan path of the telephoto camera is selected according to the priority of a candidate image region, namely the task-relevant saliency value. Task-oriented evaluations and rewards should be considered for further improvements.

4.3 Autonomous Switching of Attention Mechanisms (TOB)

As mentioned previously, a top-down biased bottom-up attention strategy can help a lot in terms of efficiency. However, it fails if a group of objects is searched whose appearances can not be uniquely described by low-level features used in a bottom-up computation model. For example, different traffic signs are all salient in color but different in geometry and have different patterns on them. They are, therefore, not distinguishable from each other only relying on low-level features used in bottom-up attention selection. An exhaustive search is still needed. To lower the computational cost, a search window is usually defined for exhaustive search as the robot FOA, in which the exhaustive search is conducted.

A search window based on bottom-up attention can predict image regions with higher probability to contain a target object, while a search window based on top-down attention is efficient for task accomplishment. Both bottom-up attention and top-down attention are essential for robot attention control. On the one hand, if a task-relevant object is not located in the robot FOV, pure top-down attention selection can also use position data in 3D task space to direct robot attention towards the target, while bottom-up or top-down biased bottom-up attention selection only relies on the 2D image data. On the other hand, if there is no task-relevant information in the FOV at all, a pure bottom-up attention can guide the robot attention to explore the environment in a flexible way. In this section, autonomous switching between top-down and bottom-up attention mechanisms is proposed, which are also adapted to the internal robot states.

4.3.1 State of the Art

Only a few works have up to now considered switching between top-down and bottom-up visual behavior. In [53], top-down object search and bottom-up environment exploration using the same saliency map model and robot platform are proposed. However, the switching between them is manual. In [90], a top-down part is initialized by a bottom-up part to recognize actions, track the actions, and determine the current context. In [23], visual attention is switched between different targets. Instead of a pure bottom-up or top-down state, visual attention allocation is determined by varyingly weighted top-down and bottom-up signals to demonstrate the robot gaze preference. In [209], a task-driven object-based visual attention model for robot applications is proposed which involves five components: pre-attentive object-based segmentation, bottom-up still attention, bottom-up motion attention, top-down object-based biasing, and contour-based object representation. Task-specific moving object detection and still object detection are operated based on this model. In [51], three visual behaviors are defined: exploration behavior, coverage behavior, and view point selection behavior. The first behavior is more a robot exploration behavior than a visual behavior. In the second behavior, potential objects are explored by the peripheral vision using bottom-up attention. After the environment is fully covered, novel perspectives of the objects are captured and the object recognition is conducted in view point selection behavior. The top-down state has been started only once. Strictly speaking, none of the above works have applied autonomous switching between pure top-down and bottom-up attention.

Moreover, most visual attention systems are studied decoupledly, where a goal-directed robot operation is commonly ignored. A robot should always be supposed to do something with the target object, such as approaching or manipulating. Considering this, robot visual attention behavior should be adapted to the internal robot state to achieve a complete system.

Therefore, a switching between top-down visual state and bottom-up visual state is proposed to deal with different situations, which enables autonomy of robots in terms of visual behavior. This autonomous switching between these two kinds of attention selection mechanisms is also adapted to different internal robot states and fills the gap for object searches not solvable using a conventional combination of them, such as TBB.

4.3.2 Model of TOB

The switching mechanism of attention selection for an autonomous robot is illustrated in Fig. 4.13. Three different robot internal modes are considered:

- *Exploring* mode, in which the robot has no specific task and just explores the world by looking at interesting parts of the environment;
- *Searching* mode, in which the robot has a specific task and searches for its current target object;
- *Operating* mode, in which the robot is accomplishing its task, e.g. moving to or manipulating the detected target object.

Four attention selection states are assigned to the robot visual behavior: the bottom-up state in the exploring mode (abb. BU_e), the bottom-up state in the searching mode (abb. BU_s), the top-down state in the searching mode (abb. TD_s), and the top-down state in the operating mode (abb. TD_o). Seven transitions are defined. In this section it is discussed how the robot FOA is determined in each state and how the autonomous switching between the states is conducted.

Bottom-Up State

In the bottom-up state, the robot focuses on an interesting area in the FOV, which is computed using the local surprise map introduced in Chapter 3. The FOA is directed towards the image region which is salient or surprising and is, therefore, attractive.

In the example shown in the left column in Fig. 4.14, the rectangles in solid lines are the FOA predicted by the surprise map. A moving human is selected as the FOA because

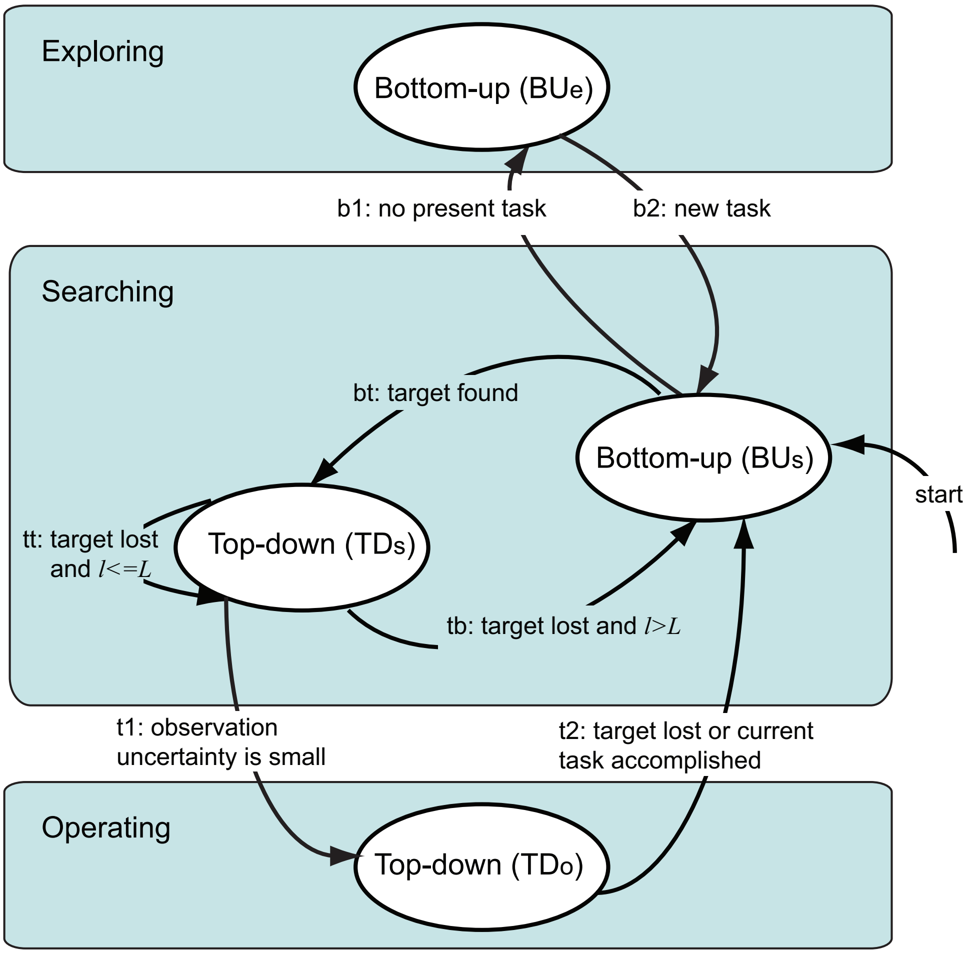

Fig. 4.13: Finite state machine of the autonomous switching mechanism. Three internal robot modes: exploration, searching and operation. Four different attention states: bottom-up state in exploring mode (BU_e), bottom-up state in searching mode (BU_s), top-down state in searching mode (TD_s), and top-down state in operating mode (TD_o).

of its high surprise value. In the bottom-up state, the robot attends to the image region limited by the rectangle in solid lines, although no robot task such as human detection is assigned to the robot. The FOA (the masked image region) and the most salient/surprising position (the rectangle) indicate the same position. More examples of the surprise map can be found in [225]. In the bottom-up state the salient/surprising image regions in the input image are viewed sequentially according to their descending saliency/surprise values.

The difference between the visual behaviors in the state BU_s and BU_e is whether exhaustive search is applied in the selected FOA. In the BU_s state, the object detection algorithm is applied in the selected FOA, since the robot has a specific task in the searching mode. In the BU_e state, the robot only attends towards the salient/surprising region. No further information processing has been applied at the current step.

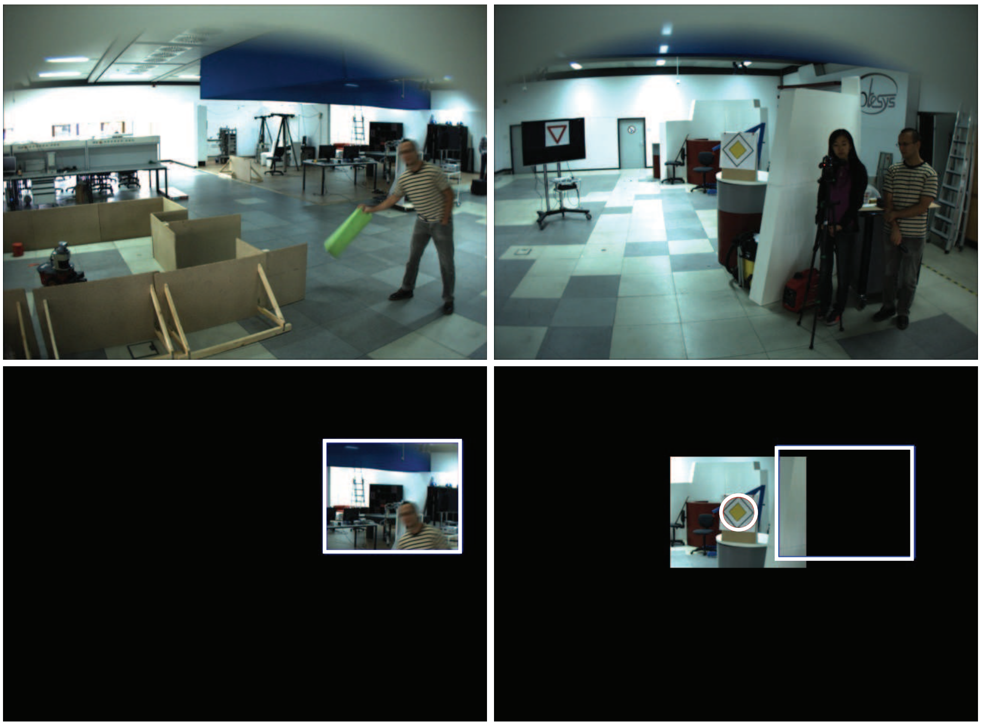

Fig. 4.14: Left column: bottom-up state; Right column: top-down state; Upper images: original input images; Lower images: the resultant robot attention windows. Rectangles: the salient/surprising image regions (the same as the masked region in bottom-up state); Masked regions: the current robot FOA; Circle: the detected target object.

Top-Down State

In the top-down state, the robot concentrates on the image region containing task-relevant information. The conventional robot tasks can be approaching, avoiding, or grasping an object in which the position estimation of the object is the main objective. To perform this task, the robot should attend to the region which contains the target object to get a better accuracy.

The right column in Fig. 4.14 shows an example of the FOA selection in the top-down state. A robot is supposed to detect a traffic sign and approach it. The region around a target object, the masked region in the right-bottom image, is selected as the current robot FOA and is further processed in detail, although this region is not the most salient/surprising region at this moment; this is in fact the region in the rectangle.

In short, in the top-down state, the position of the target object is known. No matter how salient and surprising the other features are, to perform its task, the robot attends to the detected target object.

The difference between the behaviors in the state TD_s and TD_o is that in the TD_s state the observation of the target object has a higher priority, while in state TD_o the robot starts to accomplish its task based on the complete observation acquired in TD_s state.

Switching Mechanism

The main contribution of this section is to realize autonomous switching between the top-down and the bottom-up visual attention selection considering robot task performance. The transition conditions illustrated in Fig. 4.13 are defined as follows.

After initialization, the image region to be further processed is selected in the BU_s state, since the position of the target object is unknown at this moment. Once a target is found in the selected FOA, the TD_s state is activated (bt). In this state, the image region around the target is selected constantly, ignoring the other salient features. If the target is lost, for example due to lighting condition change or humans and vehicles hiding the target object, the robot should initially continue focusing on the last region for L frames to see if the target object is re-detectable (tt). If the robot stays in top-down state for l frames, $l > L$, and the target is still unseen, the BU_s state is triggered again to search for the previous target (tb).

If the observation of the target object in the TD_s state is accurate enough, the robot starts to operate ($t1$). To evaluate the observation uncertainty, the n-dimensional system state $\boldsymbol{x} \in \mathbb{R}^n$ of the current robot task is modeled as a n-dimensional Gaussian distribution with mean value $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{P}_{\boldsymbol{x}}$ in the task space computed using a KF. The system state $\boldsymbol{x}$ is chosen according to the current task and can be the robot position and velocity for a self-localization task or object position and velocity for an object tracking task. The distribution at the previous time step $k-1$ is regarded as the prior probability density function (pdf) p_{k-1}, while the posterior belief distribution about the system state at the current time step k is p_k with a continuous variable $\boldsymbol{x}$ for specific tasks. Both of them are defined as follows:

$$p_{k-1} = \frac{1}{(\sqrt{2\pi})^n (\det \boldsymbol{P}_{\boldsymbol{x},k-1})^{1/2}} \exp(-\frac{1}{2}(\boldsymbol{x}_{k-1} - \boldsymbol{\mu}_{k-1})^T (\boldsymbol{P}_{\boldsymbol{x},k-1})^{-1} (\boldsymbol{x}_{k-1} - \boldsymbol{\mu}_{k-1})), \quad (4.11)$$

and

$$p_k = \frac{1}{(\sqrt{2\pi})^n (\det \boldsymbol{P}_{\boldsymbol{x},k})^{1/2}} \exp(-\frac{1}{2}(\boldsymbol{x}_k - \boldsymbol{\mu}_k)^T (\boldsymbol{P}_{\boldsymbol{x},k})^{-1} (\boldsymbol{x}_k - \boldsymbol{\mu}_k)). \quad (4.12)$$

Then, the observation uncertainty is defined as the Kullback-Leibler divergence or relative entropy computed as follows:

$$KL(p_k||p_{k-1}) = \int_{-\infty}^{\infty} p_k \cdot \log \frac{p_k}{p_{k-1}} \mathrm{d}\boldsymbol{x} \ \text{ in [bit]}. \quad (4.13)$$

An empirical threshold is defined for the relative entropy between the predicted and the updated state estimate as one of the criteria for evaluating the observation uncertainty. The smaller the observation uncertainty is, the less the estimation and its expected value vary, and therefore, the more certain the position estimation is. If the observation uncertainty at the k-th step is smaller than this threshold, the observation at this step is regarded as successfully executed. Upon this value the robot takes the decision what action to perform next: operating or observing. Correspondingly, if the task is finished or the target is lost, the robot stops the current operation, turns into the BU_s state, and observes ($t2$).

If the predefined task is accomplished in total, the robot explores the world by directing

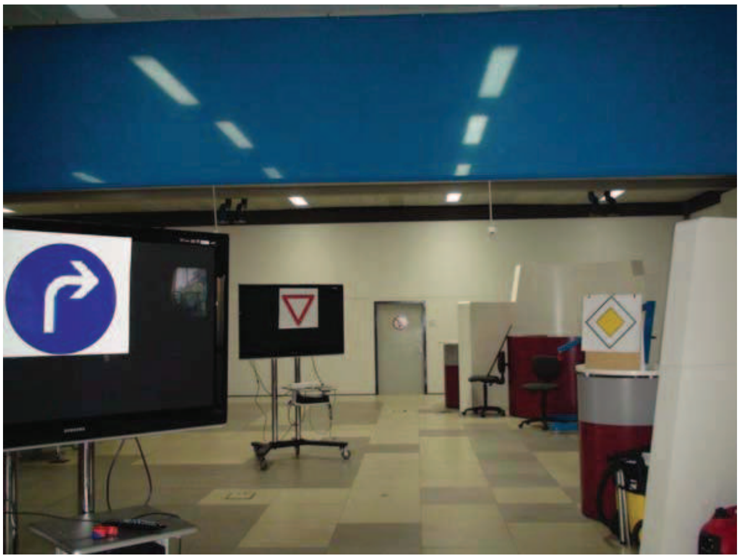

Fig. 4.15: Experimental setup consisting of three different signs.

its attention towards interesting areas in the environment selected in a pure bottom-up state BU_e ($b1$). If a new task with new target objects arrives, the robot attention selection is in the BU_s state again ($b2$).

To sum up, the robot's visual behavior with different emphases of information acquisition is now adapted to the internal robot state. The switching of robot FOA selection mechanisms is autonomously conducted.

4.3.3 Performance Evaluation

To demonstrate the strategy, experiments were conducted using the ACE robot. Sign detection and approaching tasks were assigned to the ACE robot. Fig. 4.15 shows the experimental scenario in the institute laboratory from the robot perspective. Three different signs were placed in different distances to the initial robot position.

These signs can not be uniquely described by low-level features used in the saliency map model and therefore can not be easily recognized and distinguished by enhancing certain bottom-up features using top-down information. Previously trained classifiers based on Haar-like features are used for object recognition [217]. To lower the computational cost of object recognition, the classifiers were only applied in the FOA selected in the input images. The whole input image represents a peripheral sensor input, while the focus region represents a foveated sensor input with a higher resolution.

Experiment 1: Searching → Operating → Exploring

In the first experiment, the robot was supposed to detect the blue sign and move toward this sign. If the robot reached its desired position, namely 1 m in front of the sign, it should turn 180° and move back to its initial position.

Fig. 4.16 illustrates some representative original input images and their respective results of the attention selection (the masked region). The frame number and the attention

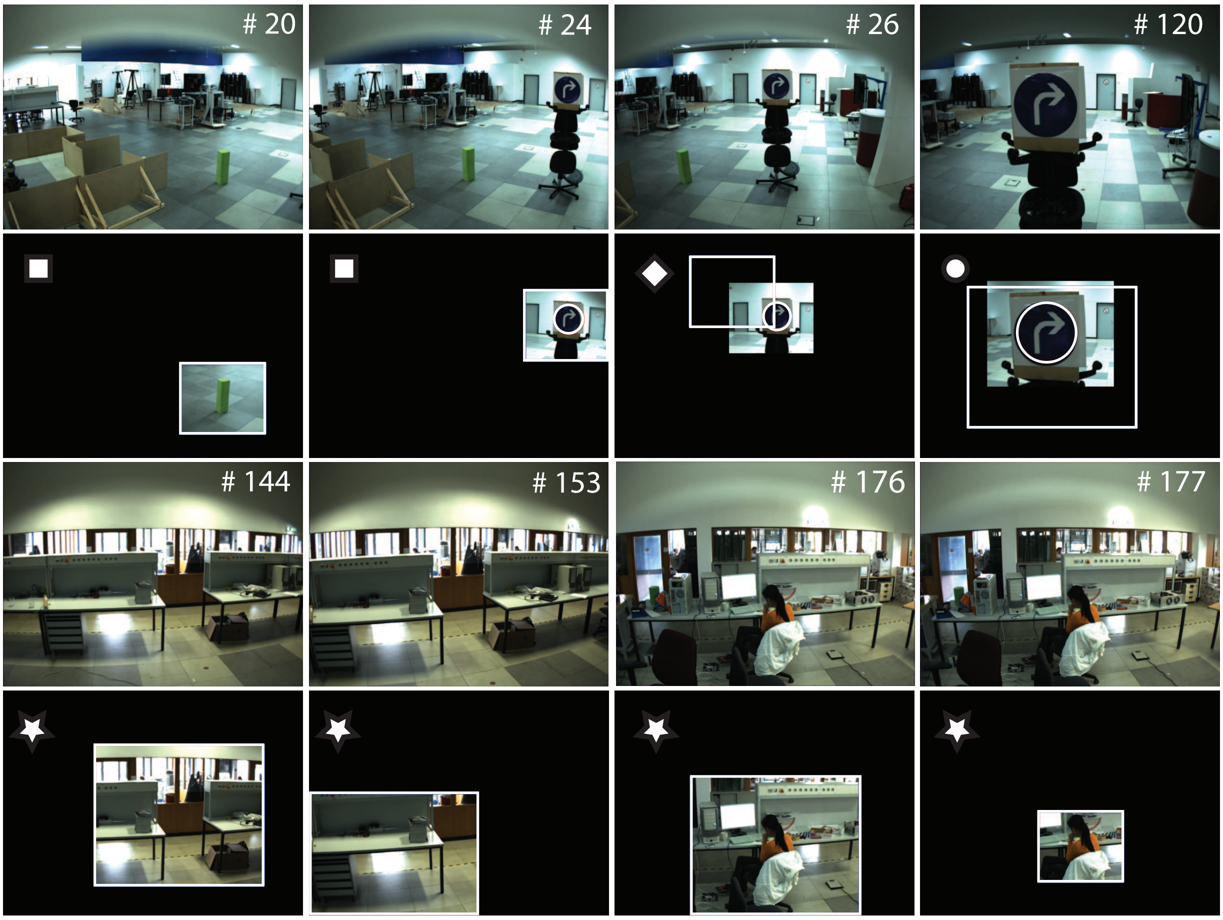

Fig. 4.16: Results of experiment 1 comprising original input images (the first and third rows) and their respective resultant images with the robot FOA (the second and the fourth rows). Numbers on the original images indicate the frame number. The markers on the left top corner of the resultant images are defined as follows: Rectangle: bottom-up attention in searching state (BU_s); Diamond: top-down attention in searching state (TD_s); Circle: top-down attention in operating state (TD_o); Star: bottom-up attention in exploring state (BU_e). Rectangles: the salient/surprising image regions; Masked regions: the current robot FOA; Circle: the detected target object.

selection state of each image are also given. The robot first searched for the traffic sign in the BU_s state. The object detection algorithm was applied in the selected salient/surprising image regions (frame 20 and 24). After the sign was detected in the FOA in frame 24, the robot kept focusing on the sign and computed the relative position (frame 26) in the TD_s state. After the position estimation of the sign was accurate enough, the robot started to move towards the sign and tracked the sign during the movement (frame 120) in the TD_o state. The threshold for the observation uncertainty was set to 0.12bit. After the task was accomplished, the robot turned back and looked at the salient/surprising parts in the environment (frame 144, 153, 176 and 177) in the BU_e state. The size of FOA varied with the size of the detected target object or the size of the salient/surprising image regions.

Experiment 2: Searching ↔ Operating

In the second experiment, the ACE robot was supposed to detect three different signs one after another. The positions of the signs were unknown. Once a sign was detected and the position of this sign was satisfyingly estimated, ACE moved straight ahead and tracked the sign using the active camera head during the movement, until it reached the position one meter in front of the sign. Then, the head of the robot should turn to another direction randomly and search for another signs and so on.

Fig. 4.17 illustrates the experimental results. Images with the FOA (the masked region) and salient/surprising region (the region in the rectangle in solid lines) as well as the frame number are shown. At the first step, ACE looked straight ahead and the BU_s state was activated. In frame 1, the blue sign was detected. The FOA changed into the TD_s state. The image region around the blue sign was selected in the following frames, until the robot reached the position one meter in front of the blue sign (frame 44). Then, the robot turned its head randomly to the right side and detected the yellow sign (frame 45). After the position estimation was satisfyingly accomplished, the robot started to move and track the yellow sign. In frame 111 the sign was lost and the BU_s state was activated after several frames. In frame 127, the yellow sign was re-detected in the FOA. The TD_s state was triggered again. After the robot reached the position one meter in front of the yellow sign, the head was randomly directed and the state was the BU_s state again (frame 149). In frame 151 and 214 the red sign was detected and tracked. For 228 frames in total, there are 18 frames in the bottom-up state and 210 frames in the top-down state.

Fig. 4.18 illustrates the evolution of the observation uncertainty and the switching mechanism between the top-down and the bottom-up states. The semi-transparent time intervals indicate the operating state in which the robot was moving. The blank areas indicate the time intervals in which the robot was in the searching mode. The frame numbers near the arrows show several representative time points. In frame 1, the first sign was detected. The observation uncertainty reached its maximum in frame 4 and decreased in the searching mode, since repeated viewing of the same object reduced the observation uncertainty. In frame 36, the observation uncertainty reached its threshold, here 0.12 bit. The robot was triggered into the operating mode and started to move towards the first sign. In frame 45, the second sign was detected coincidently in the top-down search window of the first sign in the operating mode. Therefore, the robot attention state was switched from this top-down state to the top-down state in the searching mode for the second sign. The observation uncertainty reached a local maximum in frame 50. The second sign was lost in N frames before frame 120 and re-detected in frame 127. A local maximum of the observation uncertainty was reached in frame 128. In frame 156, a local maximum of the observation uncertainty was reached again, after the third sign was detected.

To evaluate the visual guidance performance separately, the other sensors on ACE such as laser range finders were deactivated. To avoid possible crashes with the signs, a very low threshold value was set to the observation uncertainty, which caused a relatively long period in the searching mode before the robot started to operate. However, this can be easily improved if other sensor modalities are used for obstacle avoidance as well.

Tab. 4.3 shows the average computation time which was taken in different phases. Since the bottom-up attention selection was implemented on the multi-GPU platform,

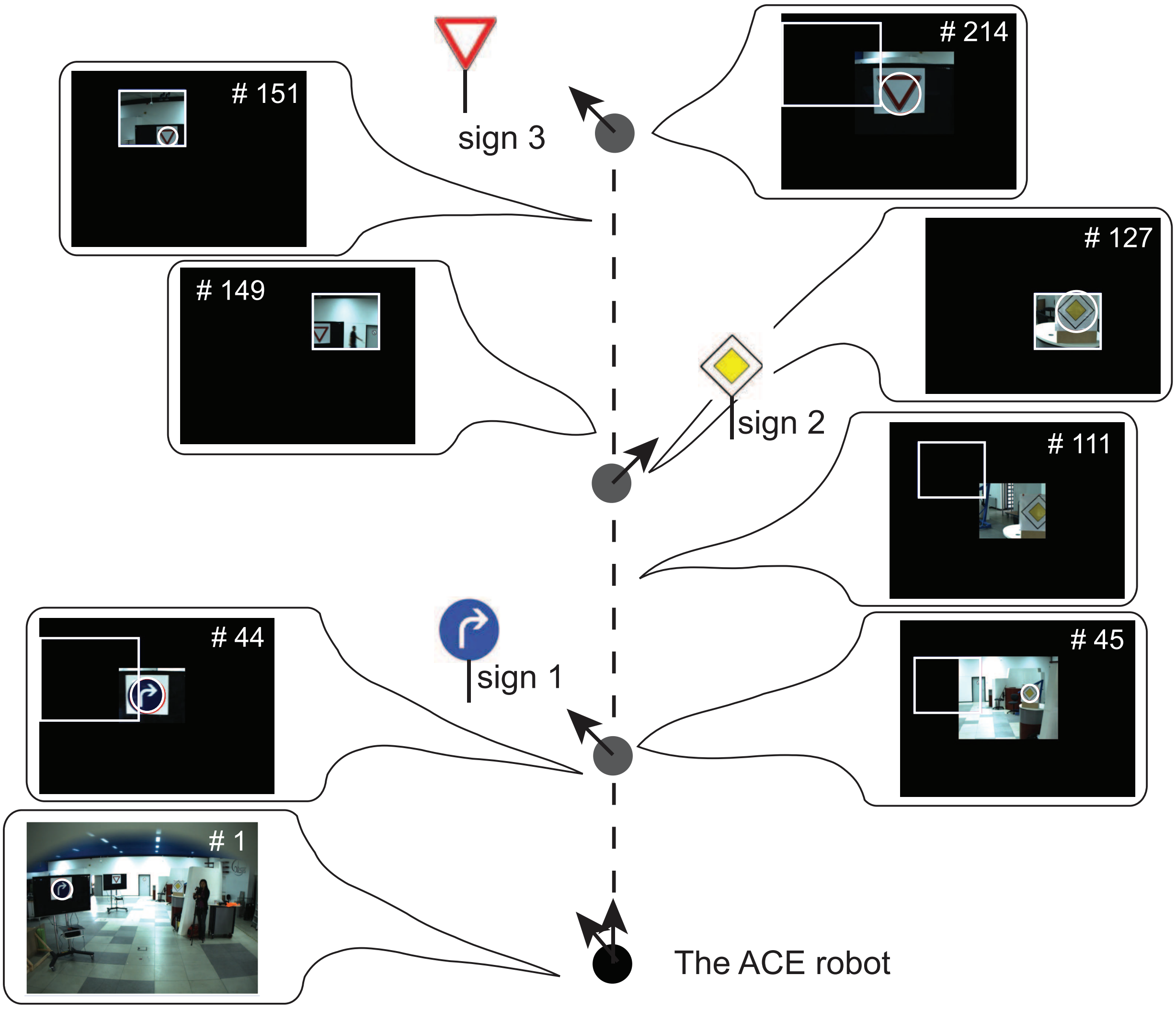

Fig. 4.17: Results of experiment 2 comprising the resultant images of robot FOA. Numbers indicate the frame numbers. The solid circles: the ACE robot. The arrows on the robot: the view direction of the active camera head. The dashed line: the robot trajectory. Rectangles: the salient/surprising image regions; Masked regions: the current robot FOA; Circle: the detected target objects.

real-time processing in this part is ensured. The most expensive processing is the object recognition using the previously trained classifiers. There is a large improvement in the performance if the robot searches for the signs only in the FOA but not in the whole image.

4.3.4 Discussion

In this experiment, the searched targets, namely three different signs, have different appearances. However, it is impossible to use uniform or similar model parameters such as the weights of feature maps in bottom-up attention selection models to represent and distinguish between them. Purely bottom-up attention facilitates the robot task accomplishment by providing FOA candidates and reducing the detection time. Image regions

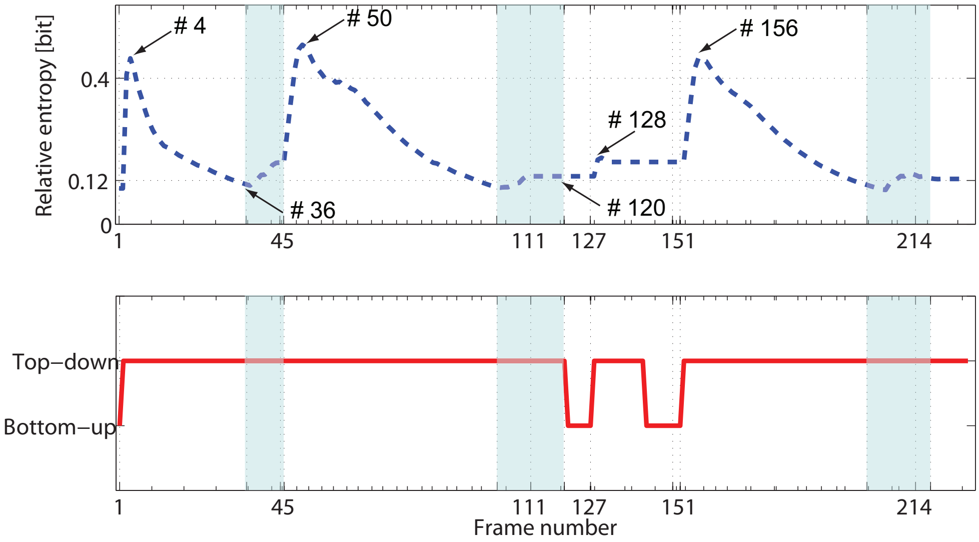

Fig. 4.18: Relative entropy (observation uncertainty) evolution (dashed line) and the respective attention state (solid line). The semi-transparent areas indicate the time intervals in which the robot was in operating state. The blank areas indicate the time intervals in which the robot was in searching state. Some representative time points are shown with their frame numbers.

Task	Time [ms]
Image capture (approximately)	67
Surprise map computation	20
Search for a sign in the FOA	31
Search for 3 signs in the FOA	33
Search for a sign in the whole image	183
Search for 3 signs in the whole image	373

Tab. 4.3: Average computation time in the experiment.

with higher saliency are regarded as positions with a higher probability of containing a target object and are processed first. Moreover, *inhibition-of-return* (IOR) is used here to extend the robot FOA to less salient regions.

In this experiment, the resolution of the vision sensor is still sufficient for sign recognition. If more resolution is required to further process the selected region, the bottom-up state is a must for efficient utilization of high-resolution cameras, providing potential image region candidates before a target object is found. Otherwise, the high-resolution camera has to search for objects in the environment randomly and inefficiently.

In addition, the pure bottom-up state guides the robot attention to explore the environment in a flexible way if top-down information does not exist in the current FOV at all.

To accelerate the whole task performance, it is obvious that the pure bottom-up attention selection should be used as little as possible, although the bottom-up state is necessary. Three solutions are suggested:

- Reduce the computation time for the bottom-up state, which has already been achieved using the multi-GPU implementation.
- Use TBB for a more efficient search in the top-down state after a target object has been found.
- Apply IOR in the 3D task space to avoid repeated view of the positions which have already been observed. Currently, a simple IOR is integrated in our implementation in the way that the current FOA is suppressed in the searching and exploring modes where the robot is not in motion.

4.4 Summary

Visual attention has become a popular topic of robotics research to deal with the limited processing capability and the real-time requirement of technical systems. Especially for autonomous and/or mobile robots, goal-directed guidance of gaze control based on coordinated task and stimulus parameters plays a key role in robot attention development. To enhance the ability of the bottom-up attention in facilitating robot task performance and solve the problem of resource limitations, two integrated approaches, TBB and TOB, are proposed.

In TBB, conventional offline training of task-relevant top-down information is replaced by the online extraction of top-down information of the first recognized target object. Successively, adaptation of model parameters to changing environments using a KF is developed. Compared to the state-of-the-art approaches, TBB displays improved efficiency in terms of reduced necessary fixations, higher flexibility with reference to unnecessary offline training, and enabled adaptability in relation to changing backgrounds mainly caused by the robot mobility.

In TOB, autonomous switching between top-down and bottom-up attention selection is realized, which fills a gap in TBB for the situation where totally different targets are searched for while contexts vary. The capability of autonomous switching of visual attention selection mechanisms enables a vision-guided mobile robot to be “autonomous” in this aspect for the first time. Visual behavior, the selection mechanism of the robot FOA, is adapted to the internal robot state.

To demonstrate the strategies, the active camera system placed on the ACE robot was used in the experiments to imitate peripheral and foveal vision as well as scan, saccade, and fixation behaviors. Here, the cooperation of the hardware and the parallel implementation aiming at real-time robot visual attention control during robot motion is established sufficiently. This application-oriented robot attention system makes a step forward in efficient visual information selection and also contributes to a further development towards cognitive visual perception in the robotics domain.

5 Human-Inspired Temporal Attention Control for Multi-Object Tasks

5.1 Introduction

In the preceding chapters, robot attention is studied in the spatial aspect: Chapter 3 contributes to the selection of task-irrelevant stimuli from the environment, while task-relevant information is selected in Chapter 4. Robot attention focuses on the winner(s) in the current image data in terms of 2D appearances.

However, in most robot tasks, multiple task-relevant target objects exist concurrently, which may even have different appearances and different saliency values in 2D images, but each of them earns a detailed observation for task performance. Some example scenarios such as environment monitoring or multi-robot manipulation/cooperation are shown in Fig. 5.1. For robots with limited FOVs, the temporal aspect of robot attention is a key factor, since the task-relevant target objects may not be located in the FOV concurrently. The central problem is how robot attention should be distributed along the time scale. While the preceding chapters consider the spatial aspect of attention, this chapter considers the temporal aspect of fixation.

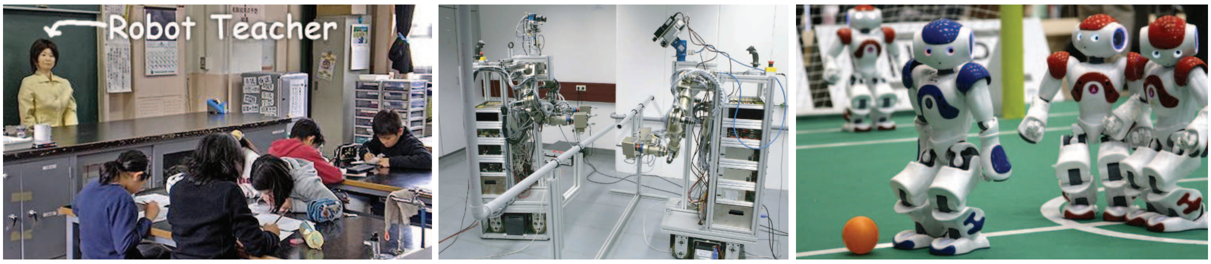

Fig. 5.1: Example scenarios for robots with limited FOV facing more than one task-relevant object. Left: the robot teacher SAYA looking at her students [208]. Middle: the cooperation of two robots for repairing a broken water pipe [28]. Under restricted communication, each robot should not only observe the water pipe but also monitor the manipulators of the other robot. Right: Nao robots in RoboCup 2009 [158].

Fundamental research in cognitive psychology and neuroscience has investigated sequential attention planning of human subjects, but few works consider it during coordinated multi-agent motion. Therefore, an experimental investigation of human eye movement during body movement is conducted. The results show that, in addition to the switched eye and body movements, repeated viewing of multiple task-relevant objects and the preference for dynamic task-relevant objects are investigated in human eye movements.

Inspired by the analytical results of this experiment, a human-inspired temporal attention planning strategy for a multi-object task is proposed, which is formalized as an optimization problem considering multiple objectives such as perception uncertainty, coverage of objects, and acquisition of new information under mechanical constraints such as the limited FOV of the camera, solving the "*when and where*" components of camera movement control. A significantly improved perception uncertainty and a similarity to human behaviors are achieved.

Furthermore, inspired by the human attention behavior, the temporal aspect of a coordinated behavior between visual attention and body motion in a multi-robot system is studied exploratively. The new contribution to existing studies is the first-time utilization of active vision sensors with limited FOV in a multi-robot formation task, deploying a coordination of attention and motion.

The remainder of this chapter is organized as follows. In Section 5.2, human motion and gaze behavior are experimentally investigated. In Section 5.3, a human-inspired sequential attention selection strategy is proposed to solve the problem of multi-object tasks. In Section 5.4, a behavior-based robot formation strategy is designed, combining attention control and motion control. The performance of the strategy is simulated and discussed. A summary is given in Section 5.5.

5.2 Experimental Study of Human Attention during Motion

Human behavior is assumed to be efficient in many tasks [44, 133]. Especially in this aforementioned context, a human eye with the limited FOV is a suitable biological example. Therefore, human attention behavior is to be studied for inspirations.

In cognitive psychology and neuroscience, experimental studies about visual scan path selection under task demands have been conducted extensively, especially in recent years since modern eye trackers have become available. Most studies investigate eye saccades facing artificial scenes [108, 211] or static pictures [52, 165] in the 2D image space according to image saliency, while some others use virtual environments [87, 160]. Only a few studies put subjects into a real-world scene to perform a natural task and investigate human visual behavior during subjects' activities, such as tea making [99, 100], arm movement [91], making a sandwich [73] or navigating on a sidewalk [72]. Furthermore, differences may be observed in a scenario where other unpredictable, autonomous humans or objects exist.

Therefore, an experiment is designed to investigate human eye movements, considering various relevant aspects for the objective of robot attention planning, such as existence of multiple dynamic or static task-relevant objects as well as eye movements during body movements. Subjects are actively and naturally involved in the task and perform a goal-directed behavior. For the sake of future work such as realization of joint attention or coordinated attention in robotic systems, a scenario is chosen, where subjects also cooperate with each other in the experiments.

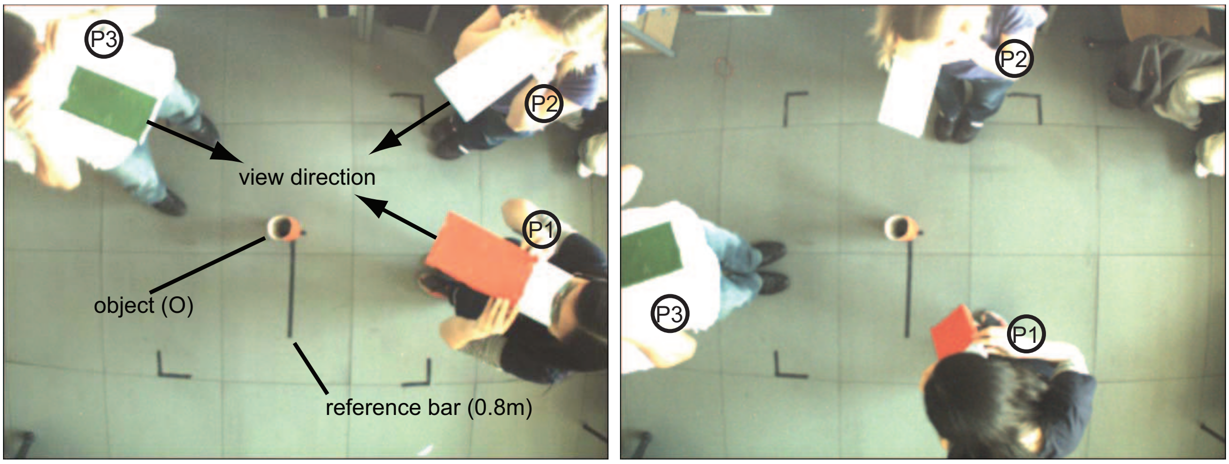

Fig. 5.2: Experiment overview captured by the overlooking camera. Left: Start positions of participant P1, P2, P3, object (O), as well as a reference bar of 0.8 m; Right: final positions of the participants. An equal distribution around the object is built.

5.2.1 Experiment: Materials and Procedure

In this experiment, three participants were asked to perform a formation task. First they were located around an object at a distance of 0.8 m, as illustrated by a reference bar in Fig. 5.2. The participants were supposed to move around the object and to distribute themselves equally, which means that any two of them should build an angle of 120° with the target object in the center as the endpoint. During the movement, they should also keep the same distance to the object. Seven healthy participants were divided into three groups. Each group performed the task three times with different participants at different initial positions. The first trial in each group was a test trial. The starting points were fixed. During the experiments, the participants' eye movements while performing this task were recorded.

In the experiments, conventional (mobile) eye trackers are abandoned for the following reasons:

- The set-up of more than one portable head-mounted eye tracker [72] is costly.
- In addition to the eye movement, the body movement for the formation task should also be investigated, which may be different when subjects act *in* a scene or *in front of* a scene (virtual environment) with *real* partners or *virtual* partners.
- A perfect modeling and illustration of human behavior in a virtual environment is difficult and costly at the current step.

To record the eye and body movements, each participant used a rectangular paper scroll to cover his/her eyes, such that each participant only had a narrow FOV during the experiment. A similar set-up was used in [112], in order to investigate human visual behavior while avoiding obstacles in movement. The paper scrolls were in different colors (white, green, and red) for each person. An overlooking camera (Firefly camera from Point Grey Research Inc. with a resolution of 640 × 480 pixels at 30 Hz) was mounted on the ceiling and used for recording the whole process. Two example input images are shown in Fig. 5.2. The gaze directions and the body positions of each participant were collected frame by frame manually.

It is worth mentioning here that the focus of this experiment is not an investigation of the targeting of the eye/head movement like the studies in other conventional visuomotor research [72, 73, 91, 99], but the scheduling of the eye/head movement. It is assumed that the subjects know what the task-relevant objects in this task are, namely the static object in the center and the other participants. The focus examined here is when and which of the task-relevant objects is attended to by which subject. Therefore, the relatively lower spatial resolution compared to a conventional eye tracking system is not a severe problem.

Moreover, this artificial delimitation can effectively

- inhibit the concurrent view of two or more task-relevant objects/participants in order to ease the identification of the present FOA,
- slow down the saccade velocity by registering the head movement instead of the eye movement,
- and reduce the dimension of the gaze location from 3D to 2D,

such that the participants' gaze behavior is goal-oriented, intended, and recordable. Furthermore, the direction and the length of the longer borders of the rectangular paper scrolls in 2D images indicates the 3D gaze direction. Different colors denote different initial position, such that the body position and gaze direction could be automatically recorded.

5.2.2 Results and Inspirations for Robotic Systems

The hypothesis is that humans' gaze behavior during movement is different to that while they are standing still.

In the experiments, three participants were involved and named P1, P2, and P3. In the following analysis, three different states of the participants were observed, namely *actor*, *observer*, and *idler* with the following characteristics:

- An *actor* is acting towards his/her temporal goal.
- An *observer* is standing still, observing the other participants, the object or the environment, and making decisions for their own future action.
- An *idler* approaches the final position and is investigating the environment without a dominant intent.

Firstly, the participants' movement and the switching of their gaze direction were analyzed. Then, a further investigation of the gaze distribution was conducted. The body movements and gaze distributions showed a high similarity across subjects.

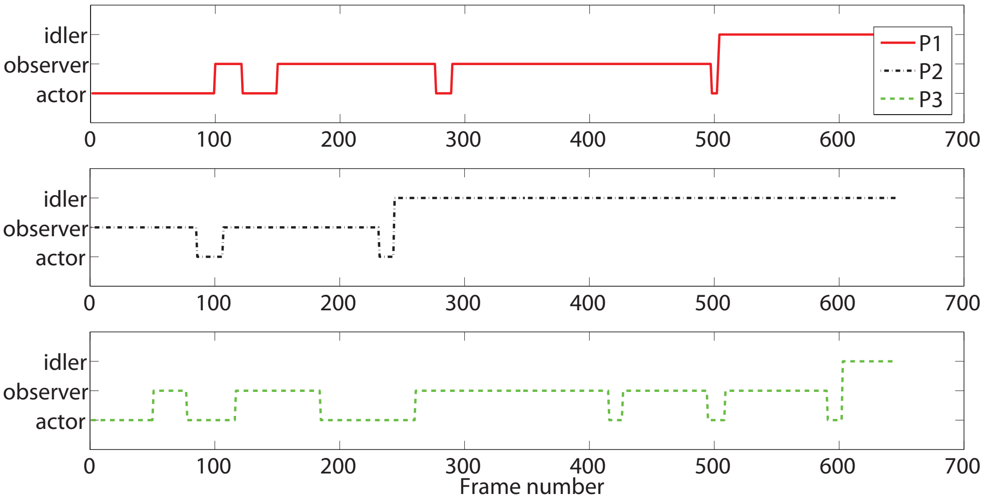

Fig. 5.3: Participants' state variation among *actor*, *observer* and *idler*. The solid line: P1; the dash-dot line: P2; The dashed line: P3.

Motion and Attention

A typical participants' state variation in an experimental trial is shown in Fig. 5.3. Since the view angle of the participants was limited, the states of the participants switched between *actor* and *observer*. A possible reason could be that the task-relevant objects including the static object (O) and the other participants were not able to be located in the FOV concurrently. Along with the increasing time, the measurements of the object positions became uncertain due to self-motion and motion of others, etc. The participants had to stop and observe the environment to reduce perception uncertainty, such that the task could be performed at all.

Moreover, the motion of the participants was totally asynchronous and decentralized, since no communication occurred in the experiments. In 9 trials, P2 always started as an *observer* and then become an *actor* later than P1 and P2. A possible reason is that the initial angle built by P1 and P3 with O was the largest. Therefore, P1 and P3 started to move earlier than P2. The relative positions could be an important factor for the determination of the movement. A small position difference may cause a higher probability of collision.

In addition, each *actor* started to move toward the direction where the furthest participant was located, in order to balance the relative position between him-/herself and the other participants and achieve the desired equal distribution (see Fig. 5.4). For instance, as illustrated in Fig. 5.2, when facing the target object, P3 moved to his/her right, P1 moved to his/her left and P2 moved to his/her right. Each participant only moved about the half of the absolute difference between his/her left and right angle, spanned with the neighbored participants to the left or right and the target object. It is suggested that the participants have assumed that the neighbored participant would coordinately help reduce the formation error.

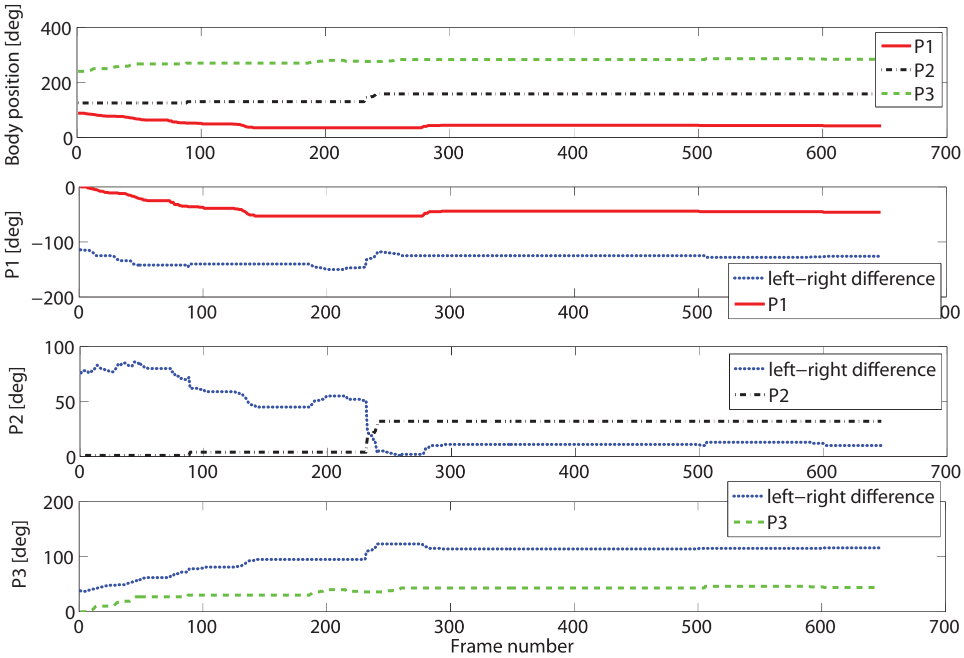

Fig. 5.4: Participants' body movements computed as the relative angle between the reference bar and the line formed by each participant and the target object in [deg]. Solid lines: P1; Dash-dot lines: P2; Dashed lines: P3; Dotted lines: difference between the left relative angle and the right relative angle of each participant (the formation error).

FOA Switching

Fig. 5.5 shows the gaze direction of P1, P2 and P3 in the same trial as illustrated in Fig. 5.3. The participants' gaze directions in each frame were recorded in this figure. Possible targets are the other participants (P), the object (O), and other task-irrelevant distractors (X). During the movement, the FOA of the participants switched mostly and repeatedly among the task-relevant objects (P and O). The reason for this repeated switching of FOA may be that repeated viewing of multiple target objects can reduce perception uncertainty, which is mainly caused by discontinuous existence of targets in the FOV.

The data from 6 formal trials are normalized over *trial time*, defined as the time interval from the beginning to the time point that all of the participants reached the final positions and stayed. Fig. 5.6 left shows the mean value and the standard deviation of the *percentage switching number of the FOA* pSN_ξ in state $\xi = \{a(ctor), o(bserver)\}$ given by:

$$pSN_\xi = 100\% \cdot \frac{SN \text{ in state } \xi}{SN \text{ in the trial time}}, \tag{5.1}$$

where SN denotes FOA switching number.

The pSNs in different states differ from each other significantly, indicated by a one-way ANOVA with $F_{1,34} = 55.95$, $p = 1.1119e\text{-}008 < 0.05$. About 85% of attention switching occured when the participants were in the *observer* state, which means that the participants barely changed their FOA when they were in motion, probably to avoid motion blur or

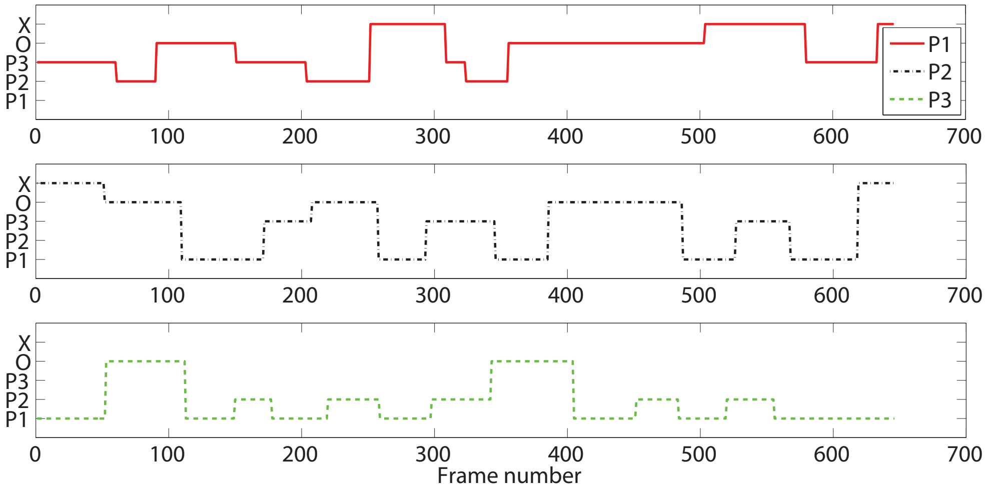

Fig. 5.5: Participants' gaze direction switching among other participants (P1, P2, or P3), the target object (O), and other distractors (X). Solid lines: P1; Dash-dot lines: P2; Dashed lines: P3.

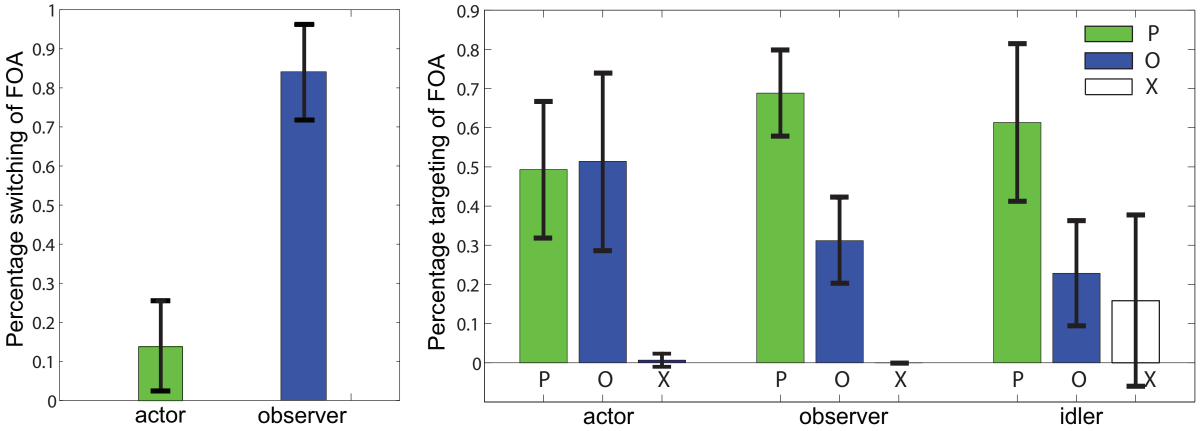

Fig. 5.6: Left: Percentage number of the participants' FOA switchings pSN_a in the *actor* state and pSN_o in the *observer* state, shown in the mean values and their standard deviations; Right: Percentage FOA duration attending to the other participants (P) $pFOA_{P,a/o/i}$, to the object (O) $pFOA_{O,a/o/i}$, and to other distractors (X) $pFOA_{X,a/o/i}$ in the state *actor*, *observer* or *idler*, shown in the mean values and their standard deviations.

difficult estimation of relative motion. The motion task added to the participants in the *actor* state occupied the computation resources of the perception behavior.

FOA Targeting

The *percentage FOA duration* $pFOA_{\xi,\eta}$ in each system state $\xi = \{a(ctor), o(bserver), i(dler)\}$ on object $\eta = \{P, O, X\}$ are illustrated in Fig. 5.6 right and computed as follows:

$$pFOA_{\xi,\eta} = 100\% \cdot \frac{\text{fixation time on } \eta \text{ in state } \xi}{\text{fixation time in state } \xi}. \tag{5.2}$$

As an *actor*, the participants attended to the task-relevant targets P and O, while as an *observer*, the participants attended to the dynamic targets, namely the other participants, much longer than to the static target, namely the object. $pFOA_{P,o}$ and $pFOA_{O,o}$ differ significantly in the *observer* state, denoted by ANOVA $F_{1,10} = 35.54$, $p = 0.00014 < 0.05$, while the difference between $pFOA_{P,a}$ and $pFOA_{O,a}$ is not significant in the *actor* state (ANOVA $F_{1,10} = 0.27$, $p = 0.6160 > 0.05$). In the state *actor* and *observer*, few distractors were attended to, possibly due to higher workload. For the *idler* state, part of the attention of the participants was drawn by task-irrelevant distractors such as the other persons in the room who were not involved in the experiments. The investigation corresponds with cognition studies about human visual attention in [101, 113].

If the relative angle between the view direction and the line spanned by the observing participant and the observed participant/object is smaller than 5°, the FOA is regarded to be on the observed participant/object. The fixation duration illustrated in Fig. 5.5 includes a small fraction of time for searching for task-relevant objects besides the real fixation duration, which is excluded in the computation in Eq. 5.2.

5.2.3 Discussion

From the results above, the hypothesis is verified. It can be concluded that the distinguishing of the behavior in states *observer*, *actor*, and *idler* is reasonable and significant. Macroscopically, the *observer* and the *actor* state switch with each other until the *idler* state is reached. The detailed results are summarized as follows:

In the *observer* state

- FOA switches frequently and repeatedly among different target objects;
- Attention is directed towards dynamic objects more frequently than static objects.

In the *actor* state

- FOA rarely switches;
- There is almost equal attention distribution towards dynamic objects and static object;

In the *idler* state

- Evidently, more attention is distributed to distractors.

Although a mapping of all the results directly on robotic implementations is not possible, this experimental investigating of human gaze behavior during movement provides inspirations for robot attention control development. Based on these results, strategies are proposed in the following sections to deal with temporal attention planning of robots in a multi-object system. Human-inspired *observer* and *actor* states are applied to robotic systems.

5.3 Temporal Attention Planning for Multiple Target Objects

As discussed in the previous section, humans switch their FOA from one target object to another when they are dealing with more than one task-relevant target object, in order to update task-relevant information. Inspired by this kind of attention selection behavior, an overt attention selection strategy for visual sensors with limited FOVs is developed and described in this section. The attention selection along the time scale is embodied as a view direction planning in the 3D task space for active vision systems.

5.3.1 State of the Art

Most conventional active vision systems are aiming at bringing the target into the center of the FOV, e.g. object tracking [18, 137]. Some others consider active search for features or objects with a pre-defined view direction sequence [41] or continuous camera panning [167]. The mechanisms in those works can hardly be regarded as view direction planning, since no re-planning mechanism is presented.

It is indicated that there is a tight link between eye movements and goal-directed motor actions [176]. The programming of eye movements can be understood within a framework of sequential information maximization [155]. View direction planning can be formulated as an optimization problem to maximize the information gain [98, 129, 155, 165, 168], to minimize the ambiguity of recognition [8], to optimize the probability of finding the target in a fixed cost limit with relation to the number of robotic actions needed [188], to achieve a maximum reward using reinforcement learning [176], or to optimize multiple objectives such as tracking accuracy and joint comfort [48], the physical, subjective, and perceptive situations [146], as well as detection probability, new informations and motion cost [162].

From the experimental results of the human attention investigation in the previous section, it is found that humans as *observers* switch their FOA towards task-relevant objects frequently and repeatedly. More attention is paid to moving target objects and less attention is paid to static target objects. The possible reason is that humans try to reduce the perception uncertainty and obtain a continuously updated modeling of the environment. Then, reduction of the overall perception uncertainty can be considered as one of the criteria for answering the attention-shift-in-time problems raised in [187]:

- In which order should the regions be selected?
- When should a previously selected region be re-selected?
- If the visual world is time-varying, how are the changes in the image contents taken into account in determining the selected regions?

In order to solve these problems, the view direction planning problem is formulated as a multi-objective optimization problem and an optimal solution considering the evaluation criteria of a technical system such as perception precision and accuracy under system constraints, e.g. limited FOV of the visual sensor, is searched for.

5.3.2 System Definition

If M equally task-relevant objects are located in the environment, a complete and certain observation of the internal representations of all target objects is desired. The representations could be concrete information such as color, shape, position, motion, etc, or abstract information such as saliency, surprise, identity, etc. The information can be static or vary dynamically. Through a repeated fixation, limited resources are allocated to the task-relevant objects to update the information.

Since conventional technical tasks consider object position estimation or manipulator position estimation relative to a reference position, positions of the task-relevant objects are regarded here as the major task-relevant information for further consideration. It is worth mentioning that the following modeling is not limited to this assumption and can be extended to consider other aforementioned features.

A scenario containing a robot equipped with an active camera with multiple task-relevant objects is illustrated in Fig. 5.7. Frames of reference are defined conventionally: the world frame $\boldsymbol{S}_0$, the robot frame $\boldsymbol{S}_r$, the camera frame $\boldsymbol{S}_c$, as well as the image plane $\boldsymbol{S}_i$ (not illustrated in Fig. 5.7). The camera on the robot has a limited FOV of $\Pi(\boldsymbol{\Psi}, \boldsymbol{L})$, constrained by limited view angle $\boldsymbol{\Psi} = (\Psi_{\text{pan}}, \Psi_{\text{tilt}})^T$ and limited confident sensing range along the optical axis of the camera $\boldsymbol{L} = (L_{\min}, L_{\max})^T$.

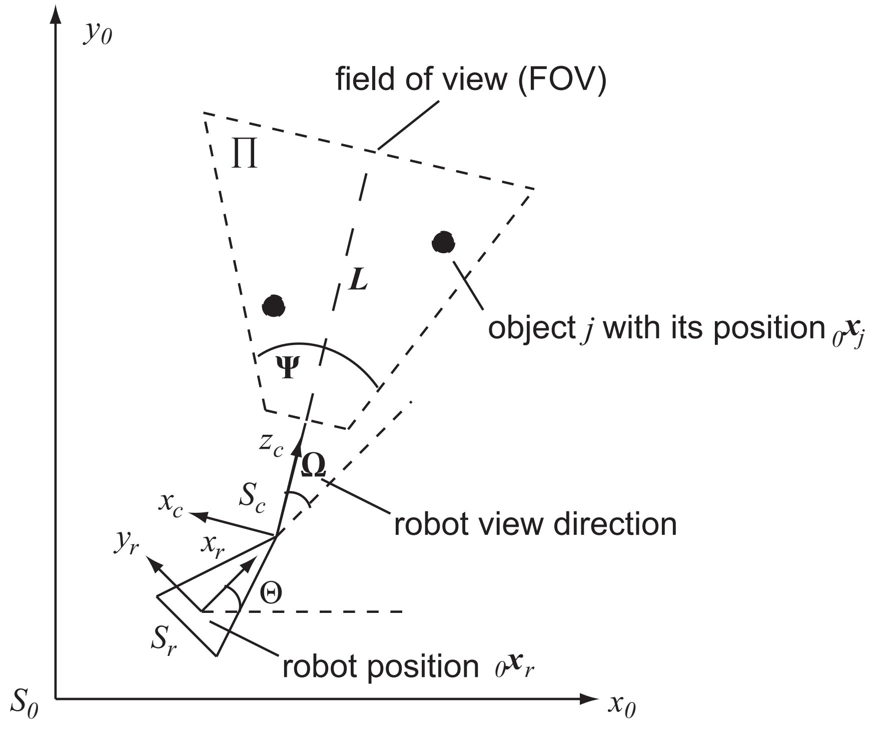

Fig. 5.7: Definition of frames of reference.

The robot position in the world frame is denoted by ${}_0\boldsymbol{x}_r \in \mathbb{R}^n$ ($n = 2,$ or 3 for 2D or 3D task space), while the robot orientation in the world frame and the view direction angle with respect to the robot frame are indicated by Θ and $\boldsymbol{\Omega} = (\Omega_{\text{pan}}, \Omega_{\text{tilt}})^T$. M task-relevant objects are located in the surrounding with the coordinate ${}_0\boldsymbol{x}_j$ for object j in the world frame. The relation between the object position in the world frame ${}_0\boldsymbol{x}_j$ and it position in the camera frame ${}_c\boldsymbol{x}_j$ can be described by the following equation:

$$ {}_0\hat{\boldsymbol{x}}_j = {}^0\boldsymbol{T}_r \cdot {}^r\boldsymbol{T}_c \cdot {}_c\hat{\boldsymbol{x}}_j, \tag{5.3} $$

where ${}_0\hat{\boldsymbol{x}}_j$ and ${}_c\hat{\boldsymbol{x}}_j$ denote homogeneous coordinates and ${}^0\boldsymbol{T}_r$ and ${}^r\boldsymbol{T}_c$ homogeneous transformation matrices.

The objects 1 to M are assumed to be static or to move with a slightly varying velocity between two consecutive time steps k and $k+1$. Then, the system state is defined as the object position and velocity in the world frame as follows:

$$ \boldsymbol{x} = (\boldsymbol{x}_1^T, \cdots, \boldsymbol{x}_j^T, \cdots, \boldsymbol{x}_M^T)^T, $$

where $\boldsymbol{x}_j = ({}_0\boldsymbol{x}_j^T, {}_0\dot{\boldsymbol{x}}_j^T)^T$ for object j. Therefore, the linear dynamic system equation for object j can be written as follows:

$$ \boldsymbol{x}_{j,k+1} = \boldsymbol{A}_j \cdot \boldsymbol{x}_{j,k} + \boldsymbol{w}_{j,k}, \tag{5.4} $$

with the $2n \times 2n$ state transition matrix

$$ \boldsymbol{A}_j = \begin{pmatrix} \boldsymbol{I}_n & \boldsymbol{I}_n \\ \boldsymbol{0}_n & \boldsymbol{I}_n \end{pmatrix}, \tag{5.5} $$

where $\boldsymbol{I}_n$ denotes a unit matrix of dimension $n \times n$ and $\boldsymbol{w}_{j,k}$ is the process noise, which is assumed to be Gaussian white noise with covariance $\boldsymbol{Q}_{j,k}$ of a dimension of $2n \times 2n$. A matrix of zeros with a dimension of $n \times n$ is denoted by $\boldsymbol{0}_n$.

From the visual data, only object position can be calculated. Therefore, the system measurement is the measured positions of the objects in the camera frame

$$ \boldsymbol{y} = (\boldsymbol{y}_1^T, \cdots, \boldsymbol{y}_j^T, \cdots, \boldsymbol{y}_M^T)^T, $$

with $\boldsymbol{y}_j = {}_c\boldsymbol{x}_j$. Then, the system measurement equation is conducted as:

$$ \boldsymbol{y}_{j,k} = \boldsymbol{H}_j \cdot {}_c\boldsymbol{x}_{j,k} + \boldsymbol{v}_{j,k}, \tag{5.6} $$

where $\boldsymbol{H}_j = (\boldsymbol{I}_n, \boldsymbol{0}_n)$ is the $n \times 2n$ observation model, and $\boldsymbol{v}_{j,k}$ the observation noise of a dimension of n, which is assumed to be Gaussian white noise with covariance $\boldsymbol{R}_{j,k}$ of a dimension of $n \times n$. From the Eq. 5.3, an equation with homogeneous coordinates ${}_c\hat{\boldsymbol{x}}_j$, and ${}_0\hat{\boldsymbol{x}}_j$ can be rewritten in

$$ {}_c\hat{\boldsymbol{x}}_j = \left({}^0\boldsymbol{T}_r \cdot {}^r\boldsymbol{T}_c\right)^{-1} \cdot {}_0\hat{\boldsymbol{x}}_j. \tag{5.7} $$

5.3.3 Temporal Attention Planning

The fundamental problem is to predict an optimal view direction $\Omega^*_{k+1|k}$ for time step $k+1$ based on the object position estimation result at time step k. Considering task-relevant evaluation criteria, a decomposed multi-objective optimization problem is defined as follows:

$$\boldsymbol{\Omega}^*_{k+1|k} = \underset{\boldsymbol{\Omega}_{k+1|k}}{\arg\min} \left(J_1(\boldsymbol{\Omega}_{k+1|k}), J_2(\boldsymbol{\Omega}_{k+1|k}), J_3(\boldsymbol{\Omega}_{k+1|k}) \right), \tag{5.8}$$

where $\boldsymbol{\Omega}_{k+1|k}$ denotes the predicted possible view directions the robot can provide at time step $k+1$. The objective functions J_1, J_2, and J_3 favor a low overall perception uncertainty, a large number of covered objects by the visual sensor, and low energy cost vs. a large amount of newly-arriving information, respectively. The definitions of $\boldsymbol{\Omega}_{k+1|k}$, J_1, J_2, and J_3 are presented as following in detail.

Optimal View Direction Candidates $\boldsymbol{\Omega}_{k+1|k}$

Due to limited capacity, attentional resources should be allocated to task-relevant objects. Therefore, the optimal view direction candidate $\boldsymbol{\Omega}_{k+1|k}$ comprises all the possible view directions for individual task-relevant objects:

$$\boldsymbol{\Omega}_{k+1|k} \in \{\boldsymbol{\Omega}^*_{1,k+1|k}, \cdots, \boldsymbol{\Omega}^*_{j,k+1|k}, \cdots, \boldsymbol{\Omega}^*_{M,k+1|k}\}.$$

Experiments were conducted to investigate visual sensor error models (see Section 5.3.4). The results in Fig. 5.8 show that the 3D locations with the minimum perception uncertainty are along the optical axis of the visual sensor. Therefore, the optimal view direction candidates $\boldsymbol{\Omega}^*_{j,k+1|k}$ are predicted using the predicted (a priori) object position $\boldsymbol{x}^-_{j,k+1|k}$ based on the previously estimated object position $\boldsymbol{x}_{j,k}$:

$$\boldsymbol{x}^-_{j,k+1|k} = \boldsymbol{A}_j \cdot \boldsymbol{x}_{j,k}. \tag{5.9}$$

Overall Perception Uncertainty J_1 using Adaptive Kalman-Filter Concept

The overall perception uncertainty J_1 depends on both accuracy and precision. It is assumed that no systematic error exists in measurements and therefore no accuracy problem occurs. Before an optimal view direction for time step $k+1$ is found and applied, multiple KFs are applied to predict the perception uncertainty for each possible view direction $\boldsymbol{\Omega}_{k+1|k}$ and for each task-relevant object. The overall perception uncertainty of the system can then be formulated similarly to [168] as follows:

$$J_1(\boldsymbol{\Omega}_{k+1|k}) = \frac{1}{n} \sum_{j=1}^{M} \sum_{l=1}^{n} \sqrt{e^2_{j,l}(\boldsymbol{\Omega}_{k+1|k})}, \tag{5.10}$$

where $e_{j,l}(\boldsymbol{\Omega}_{k+1|k})$ are the first n eigenvalues of the system state estimation covariance matrix $\boldsymbol{P}_{j,k+1|k}$ of object j predicted using $\boldsymbol{\Omega}_{k+1|k}$ (position components only). The dimension of the object position is denoted by n, while index l denotes x-, y-, and/or z-direction. Note that the velocity estimation covariance is not considered here. A conventional KF for object j consists of a prediction phase using the dynamic system model:

$$\boldsymbol{P}^{-}_{j,k+1|k} = \boldsymbol{A}_j \cdot \boldsymbol{P}_{j,k} \cdot \boldsymbol{A}_j^T + \boldsymbol{Q}_{j,k}. \tag{5.11}$$

and a correction phase using the predicted optimal Kalman gain $\boldsymbol{K}_{j,k}$:

$$\begin{aligned} \boldsymbol{K}_{j,k+1|k} &= \boldsymbol{P}^{-}_{j,k+1|k} \cdot \boldsymbol{H}^T_{j,k}(\boldsymbol{H}_{j,k}\boldsymbol{P}^{-}_{j,k+1|k}\boldsymbol{H}^T_{j,k} + \boldsymbol{R}_{j,k})^{-1}, & (5.12)\\ \boldsymbol{P}_{j,k+1|k} &= (\boldsymbol{I} - \boldsymbol{K}_{j,k} \cdot \boldsymbol{H}_{j,k})\boldsymbol{P}^{-}_{j,k+1|k}. & (5.13) \end{aligned}$$

Here, $\boldsymbol{P}^{-}_{j,k+1|k}$ denotes the predicted, prior state estimation covariance, while $\boldsymbol{K}_{j,k+1|k}$ and $\boldsymbol{P}_{j,k+1|k}$ denote the predicted Kalman gain and the posterior state estimation covariance computed based on the true state estimation covariance $\boldsymbol{P}_{j,k}$ at time step k and using the predicted possible view direction $\boldsymbol{\Omega}_{k+1|k}$.

Note The prior and posterior system state $\boldsymbol{x}_{k+1|k}{}^{-}$ and $\boldsymbol{x}_{k+1|k}$ are only predicted for computation of $\boldsymbol{\Omega}_{j,k+1|k}$ (see Eq. 5.15), but not corrected at this step (see 5.15), since no measurement $\boldsymbol{y}_{j,k+1|k}$ is conducted at all in the planning phase.

$$\begin{aligned} \boldsymbol{x}^{-}_{j,k+1|k} &= \boldsymbol{A}_j \cdot \boldsymbol{x}_{j,k}, & (5.14)\\ \boldsymbol{x}_{j,k+1|k} &= \boldsymbol{x}^{-}_{j,k+1|k} + \boldsymbol{K}_{j,k} \cdot (\boldsymbol{y}_{j,k+1|k} - \boldsymbol{H}\boldsymbol{x}^{-}_{j,k+1|k}). & (5.15) \end{aligned}$$

The state error covariance matrix $\boldsymbol{P}_{j,k+1|k}$ can be predicted and corrected based on the dynamic system model $\boldsymbol{A}_j$, the process and measurement noise $\boldsymbol{Q}_{j,k}$ and $\boldsymbol{R}_{j,k}$ using various view directions $\Omega_{j,k+1|k}$.

When using conventional KFs, the process noise $\boldsymbol{Q}_{j,k}$ and the measurement noise $\boldsymbol{R}_{j,k}$ are normally measured and tuned offline [198]. However, if they are constant, the state error covariance matrix $\boldsymbol{P}_{j,k}$ usually converges to a constant along with the increasing prediction/correction cycles, although the state estimate may be far away from the real value. In this case, J_1 would be the same for all possible view directions $\boldsymbol{\Omega}_{k+1|k}$.

To deal with this problem, an adaptive KF (AKF) concept is considered. AKF solves the problem of balancing the contributions of the system model and the measurements on the state estimation by adjusting the stochastic properties online [75, 206]. Here the predicted state estimation covariance matrix $\boldsymbol{P}_{k+1|k}$ is biased by modifying the process noise covariance $\boldsymbol{Q}_{j,k}$ and the measurement noise covariance $\boldsymbol{R}_{j,k}$.

- **Measurement Noise $\boldsymbol{R}_{j,k}$ Biased by Object Position**

For different view direction $\boldsymbol{\Omega}_{k+1|k}$, the predicted location ${}_c\boldsymbol{x}_{j,k+1|k}$ of object j relative to the camera is different. Different sensors have different sensing range and perception process error models [110, 181]. Taking this into account, the measurement noise $\boldsymbol{R}_{j,k}$ of $n \times n$ is strongly influenced by the location of the object j:

$$\boldsymbol{R}_{j,k} = \boldsymbol{R}_{j,k}({}_c\boldsymbol{x}_{j,k}), \tag{5.16}$$

while in Eq. 5.7

$$_c\hat{\boldsymbol{x}}_{j,k+1|k} = \left({}^0\boldsymbol{T}_r \cdot {}^r\boldsymbol{T}_c(\boldsymbol{\Omega}_{k+1|k})\right)^{-1} \cdot {}_0\hat{\boldsymbol{x}}_{j,k+1|k}. \tag{5.17}$$

Commonly, the magnitude of the measurement error covariance matrix $\boldsymbol{R}_{j,k}$ decreases if the object position with respect to the vision sensor $_c\boldsymbol{x}_{j,k}$ is near to the optical axis and the vision sensor, and increases if the object position $_c\boldsymbol{x}_{j,k}$ is near to the sensor limitation of the FOV. The relationship between $\boldsymbol{R}_{j,k}$ and $_c\boldsymbol{x}_{j,k}$ is determined offline and illustrated in Fig. 5.8 later in Section 5.3.4.

An extreme case occurs if the object j would not be located in the FOV when $\boldsymbol{\Omega}_{k+1|k}$ is applied. Then, the posterior state error covariance matrix $\boldsymbol{P}_{j,k+1|k}$ is equal to the prior state error covariance matrix $\boldsymbol{P}^-_{j,k+1|k}$ and will not be corrected, which results in a significant increasing of $\boldsymbol{P}_{j,k+1|k}$.

- **Process Noise $\boldsymbol{Q}_{j,k}$ Biased by Object Dynamics**

The process noise $\boldsymbol{Q}_{j,k}$ consists of process noise in position $\boldsymbol{Q}_{j,k,\mathrm{pos}}$ of $n \times n$ and process noise in velocity $\boldsymbol{Q}_{j,k,\mathrm{vel}}$ of $n \times n$ as follows:

$$\boldsymbol{Q}_{j,k} = \begin{pmatrix} \boldsymbol{Q}_{j,k,\mathrm{pos}} & \boldsymbol{0}_n \\ \boldsymbol{0}_n & \boldsymbol{Q}_{j,k,\mathrm{vel}} \end{pmatrix}. \tag{5.18}$$

While the measurement noise is dependent on the visual sensor characteristics and the object location with respect to the visual sensor, the process noise is usually biased dynamically by the current system state dynamics. In this system, the process noise is proportional to the velocity and acceleration of the respective object as follows:

$$\boldsymbol{Q}_{j,k,\mathrm{pos}} \propto {}_c\dot{\boldsymbol{x}}_{j,k}, \quad \text{and} \quad \boldsymbol{Q}_{j,k,\mathrm{vel}} \propto {}_c\ddot{\boldsymbol{x}}_{j,k}. \tag{5.19}$$

Then, the magnitude of the process noise increases if the object velocity increases or if the changes of the object velocity increases.

It can be seen from Eq. 5.10 that the overall perception uncertainty J_1 increases if the magnitude of the predicted posterior state error covariance matrix $\boldsymbol{P}_{j,k+1|k}$ increases. Together with Eq. 5.11, 5.12, and 5.13, the following conclusions can be made:

- The overall perception uncertainty J_1 increases if the process noise $\boldsymbol{Q}_{j,k}$ increases;
- The overall perception uncertainty J_1 increases if the measurement noise $\boldsymbol{R}_{j,k}$ increases;
- The overall perception uncertainty J_1 is large if the state error covariance $\boldsymbol{P}_{j,k}$ (at the previous time step) is large.

For each possible view direction $\boldsymbol{\Omega}_{k+1|k}$, the predicted overall perception uncertainty $J_1(\boldsymbol{\Omega}_{k+1|k})$ is computed. The view direction minimizing the predicted overall perception uncertainty is desired.

Visual Coverage J_2

Achieving the same J_1 using more than one view direction, it is also envisaged having a wide visual coverage J_2, which is proportional with the ratio between the number of objects located in the FOV M_{seen} and the total number of the target objects M:

$$J_2(\boldsymbol{\Omega}_{k+1|k}) = -M_{\text{seen}}(\boldsymbol{\Omega}_{k+1|k})/M, \tag{5.20}$$

In J_2 the maximization of the object number located in the FOV is considered, which implicitly integrate the mechanical constraints, such as the limited FOV of the visual sensor, since M_{seen} is dependent on the camera FOV.

Energy Cost or New Information J_3

The energy J_3 which would be consumed by changing view direction from the previous view direction $\boldsymbol{\Omega}_k^*$ to the predicted view directions $\boldsymbol{\Omega}_{k+1|k}$ is proportional to the view direction variation:

$$J_3(\boldsymbol{\Omega}_{k+1|k}) = \varsigma \cdot |\boldsymbol{\Omega}_k^* - \boldsymbol{\Omega}_{k+1|k}|, \tag{5.21}$$

while the parameter $\varsigma = \{-1, 1\}$, favoring constantly attending to different targets while $\varsigma = -1$ or attending to the same object to lower energy cost while $\varsigma = 1$. The choice of ς may vary in different application scenarios. Here, it is emphasized to view a different side of the environment to acquire new information and, therefore, $\varsigma = -1$.

Decomposed Optimization

Three aspects are considered for an optimal solution: the overall perception uncertainty J_1, visual coverage J_2, and new information J_3. Because of the different importance and priorities for task accomplishment, $J_1 > J_2 > J_3$, this optimization problem is decomposed and the minimization of the objective functions is solved one by one. A list is constructed for each view direction candidate containing the predicted objective function values at the time step k as follows:

$$\begin{bmatrix} J_1(\boldsymbol{\Omega}_{1,k+1|k}^*) & J_2(\boldsymbol{\Omega}_{1,k+1|k}^*) & J_3(\boldsymbol{\Omega}_{1,k+1|k}^*) \\ \vdots & \vdots & \vdots \\ J_1(\boldsymbol{\Omega}_{j,k+1|k}^*) & J_2(\boldsymbol{\Omega}_{j,k+1|k}^*) & J_3(\boldsymbol{\Omega}_{j,k+1|k}^*) \\ \vdots & \vdots & \vdots \\ J_1(\boldsymbol{\Omega}_{M,k+1|k}^*) & J_2(\boldsymbol{\Omega}_{M,k+1|k}^*) & J_3(\boldsymbol{\Omega}_{M,k+1|k}^*) \end{bmatrix}. \tag{5.22}$$

The first column is sorted descendingly, to find the optimal candidate with the minimum overall uncertainty J_1. If more than one view direction candidate is determined, the J_2 values of those candidates are compared with each other. If more than one view direction candidate still remains after this sorting, the J_3 values of those remaining view direction candidates are compared with each other. After the third sorting, if no uniquely optimal view direction is found, a random selection from those remaining optimal view direction candidates is conducted, which happens very seldom.

After the predicted optimal view direction $\mathbf{\Omega}^*_{k+1|k}$ is found, it is applied at the next time step $k+1$. The position and velocity estimation of the task-relevant objects are also accomplished by using another KF. If object j is located in the FOV now, prediction and correction using the KF is performed. If object j is not located in the FOV now, only a prediction using the KF is carried out.

Using this attention planning algorithm, the overall perception uncertainty from an *observer's* point of view is minimized, the number of the target objects located in the FOV is maximized, and the new information content, which is embodied by a large view direction change, is also maximized.

5.3.4 Simulation Results

To evaluate the proposed attention planning strategy from different perspectives, simulations are conducted. The first two simulations are implemented in Matlab, in which two attention planning strategies are compared in terms of the position estimation performance of multiple objects:

- *Attention Planning* (AP): the proposed attention planning strategy;
- *Round-Robin Algorithm* (RR): FOA is equally distributed, namely directed toward the left-most object and switched towards the right objects one by one and in reverse order.

A multi-robot system is then further investigated in the third simulation using Player/Gazebo in C++ [4]. Player provides a simple interface to robot sensors and actuators and allows the simulation codes to work on the real hardwares without any changes required. Gazebo is a multi-robot simulator capable of simulating a population of robots, sensors, and objects in a 3D world. It generates realistic sensor feedback and physically plausible interactions between objects. Further information can be found at `playerstage.sourceforge.net`. Real input images from a simulated camera were processed to provide the position measurements of the task-relevant objects/robots.

The sensor error model of the simulated camera used in the third simulation was established and applied in all the simulations. Moreover, task-relevant objects are assumed to be located on the ground surface. Therefore, only the horizontal pan-angle of the camera Ω_{pan} is considered in the simulations. The object position has a dimension of 2, namely in the x- and y-direction of the world frame.

Sensor Error Modeling

To estimate the perception uncertainty of a visual sensor, its characteristics should be taken into account and serve for the sensor planning. A biological model is human vision, which consists of foveal vision in the center of the FOV and peripheral vision. The foveal vision, with a high visual acuity (resolution), is sensitive to color and shape, while the peripheral vision, with a low visual acuity, has a better ability to detect motion [76].

To explore the sensor model, experiments were conducted using a simulated single camera (Sony VID30 PTZ model in Player/Gazebo simulation environment with a horizontal

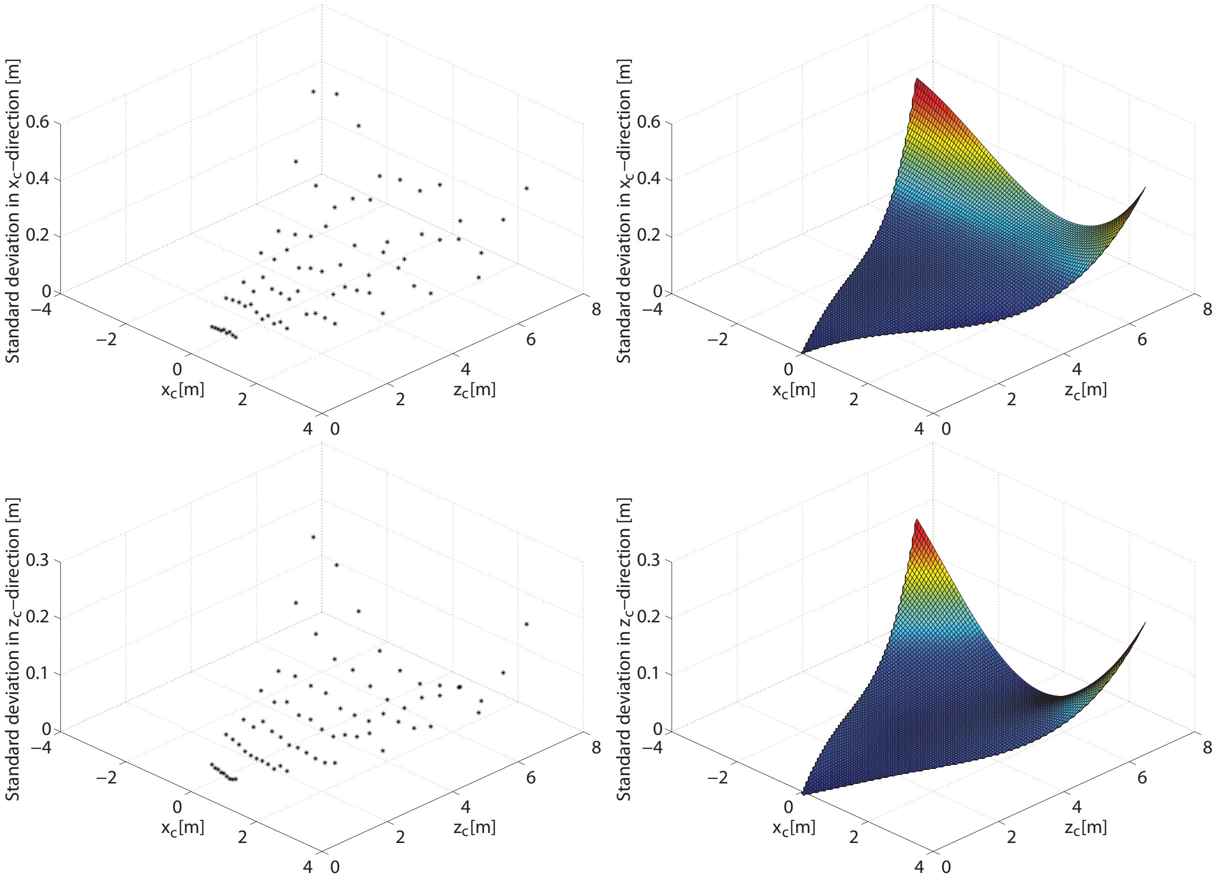

Fig. 5.8: Standard deviations of the vision-based position measurement errors $\sigma_x({}_c\boldsymbol{x}_j)$ in the x_c-direction (upper row) and $\sigma_z({}_c\boldsymbol{x}_j)$ in z_c-direction (lower row) at different positions ${}_c\boldsymbol{x}_j$. Left column: real measurements; Right column: interpolated surfaces $\boldsymbol{f}$ based on the real measurements.

FOV of 60° with a resolution of 320 × 240 pixels [4]). The positions of three reference objects at different positions ${}_c\boldsymbol{x}_j$ in front of the camera were measured: spheres with known radiuses and different colors. The systematic errors were corrected first. Then, the standard deviation of the position measurement errors (σ_x, σ_z) at different positions in front of the camera were calculated. A 3-degree polynomial surface $\boldsymbol{f}$ is fitted to the data obtained from the experiments:

$$(\sigma_x, \sigma_z) = \boldsymbol{f}({}_c\boldsymbol{x}_j), \tag{5.23}$$

indicating the standard deviations of the position measurement errors of object j with the coordinate ${}_c\boldsymbol{x}_j$ in the camera frame $\boldsymbol{S}_c$ (see Fig. 5.8). The standard deviation (σ_x, σ_z) is used to model the measurement noise $\boldsymbol{R}_{j,k}$ in the simulations (see Eq. 5.12).

It is worth mentioning that the standard deviations of the position estimation errors illustrated here are not only due to sensor noise or sensing range but also affected by the shape and size of the objects used in the experiments as well as the image processing algorithms. Experiments were also conducted using a real stereo camera (Bumblebee stereo camera from Point Gray Research Inc. with focal lengths of 2 mm each and at resolutions of 640 × 480 pixels each [5]) and a planar chess board pattern with known square size.

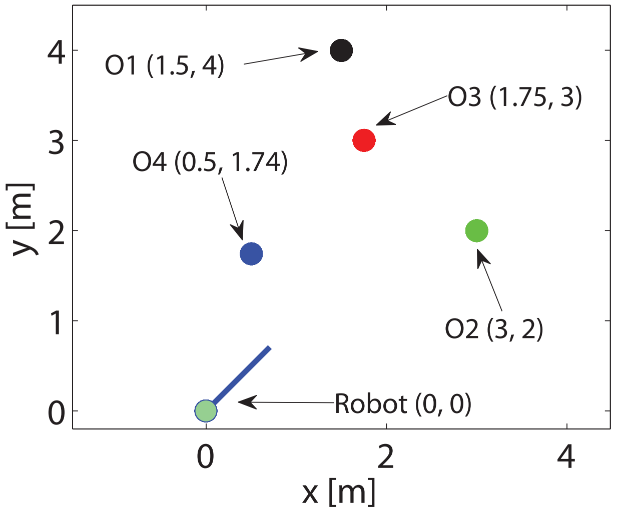

Fig. 5.9: Overview of simulation 1. A robot with position $(0, 0)$ m and four static objects are located in this scenario. The blue solid line indicates the initial view direction of the robot.

The standard deviation of position measurement errors shows a high similarity with the simulated ones except a much higher standard deviation due to varying lighting conditions in the environment.

Simulation 1: Static Objects

In this simulation, a robot is facing four static objects in the environment. The objects are distributed in a way that the rightmost object can not be seen if the robot attends towards the leftmost object, and vice versa. Possible applications are surveillance/monitoring using active cameras or human-robot interaction such as the robot teacher shown in Fig. 5.1.

Four objects O1 to O4 are located at $(1.5, 4)$ m, $(3, 2)$ m, $(1.75, 3)$ m, and $(0.5, 1.74)$ m, respectively, while the robot is located at $(0, 0)$ m (see Fig.5.9). The initial view direction is 45° with respect to the x-direction of the world frame. The horizontal FOV is 60°. At the initial position, all the objects can been seen by the robot. The simulation takes 20 time steps. The actual positions of the robot and the objects are shown in Fig. 5.10 as solid circles. The solid lines indicate the camera valid sensing range with limited horizontal view angle. The stars denote the simulated position estimations using a KF.

The top sub-figure in Fig. 5.11 shows the predicted optimal angle towards each object and the actual pan-angle of the robot at each time step. The middle and the bottom sub-figures in Fig. 5.11 illustrate the respective values of the estimation uncertainty variation of each object computed as the sum of the square root of the system state estimation covariance eigenvalues using the selected pan-angle and the respective position estimation error at each time step.

At the beginning, the robot attends to the objects 1, 3, and 4 at time step 2 to reduce the estimation uncertainty of the objects 1, 3, and 4, in order to reduce the overall perception uncertainty. Then, the robot regards the objects 1, 3, and 4 as a group, since they are near to each other and can be seen concurrently if one of them is attended to. The robot

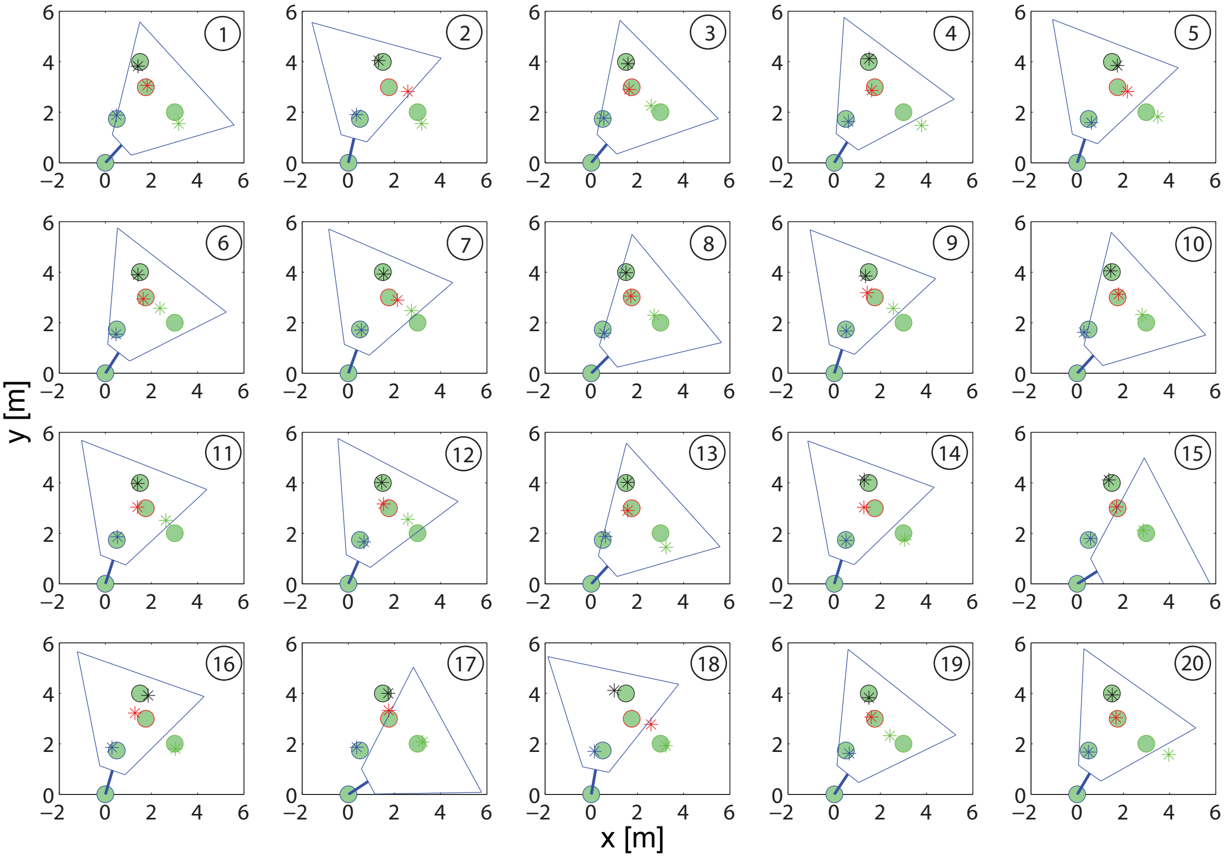

Fig. 5.10: Observation at each time step. Solid circles: the robot and the static objects; Solid lines: limited FOV of the robot; Stars: estimated object positions; Top-right number in each sub-figure: frame number (time step).

switches its attention between them and the object 2. The reason for this switching is that if the robot focuses on object 2, the uncertainty of the position estimation of objects 1, 3, and 4 increases. Attending to one of the objects 1, 3, and 4, the other two are also visible. Then, the uncertainty of their position estimation also decreases. The robot then decides to turn towards object 2, which was not located in the FOV and had a higher uncertainty at the previous time step.

Ten simulations were conducted using the same parameters. The performances of the proposed AP strategy and RR algorithm are compared in terms of position estimation uncertainty, position estimation error, and the object coverage computed as the ratio of the object number seen by the camera and the total number M in Tab. 5.1. The mean values (mean) and the standard deviations (std) are listed. Using the proposed AP strategy, a lower position estimation uncertainty with a much lower standard deviation and a smaller position estimation error with a much lower standard deviation are achieved. The object coverage is slightly smaller using the proposed AP strategy than that using the RR algorithm. The reason is that by using the RR algorithm, objects 1, 3 and 4 are not regarded as a group and attended to one by one. Therefore, object 2, which is located further from them, is not attended to that often, which results in a higher estimation uncertainty and a larger position estimation error.

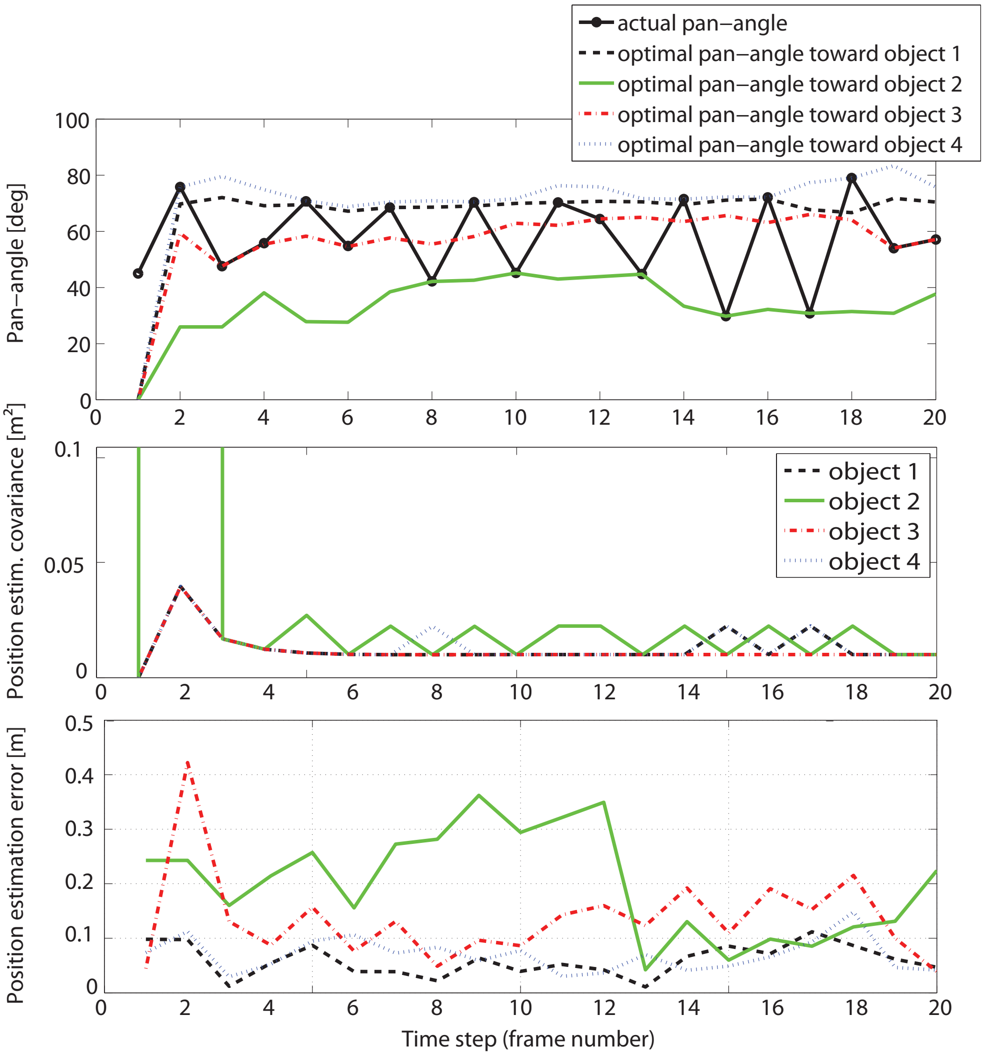

Fig. 5.11: Results of simulation 1. Top: predicted optimal view direction for individual object and the actual view direction at each time step; Middle: individual uncertainty of object position estimation at each time step; Bottom: position estimation error of each object at each time step. The solid line with circular markers: actual robot view direction; Dashed lines: object 1; Solid lines: object 2; Dash-dot lines: object 3; Dotted lines: object 4.

Item	Position Estimation Uncertainty		Position Estimation Error		Coverage	
	mean [m^2]	std [m^2]	mean [m]	std [m]	mean	std
AP	1.4434	0.0029	0.2814	0.0988	0.7675	0.0581
RR	1.7510	0.5004	0.7075	0.5549	0.7900	0.0219

Tab. 5.1: Comparison of the AP strategy and the RR algorithm in terms of position estimation uncertainty, position estimation error, and visual coverage. Data are averaged over ten simulations.

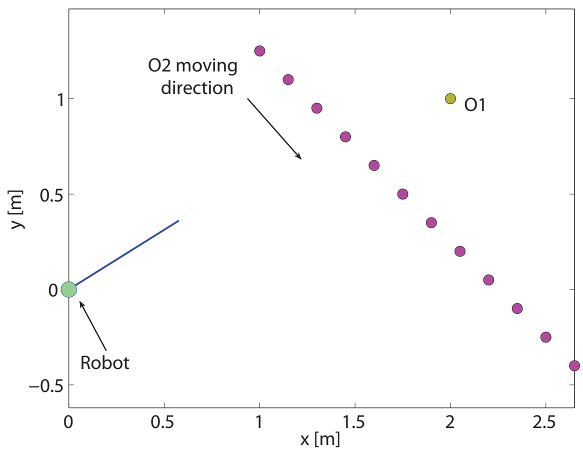

Fig. 5.12: Overview of simulation 2. A robot with two objects is located in this scenario. Object 2 (O2) moves at a constant velocity.

Simulation 2: Static and Dynamic Objects

In the second simulation, the situation where static and dynamic objects exist is considered. Possible applications are robot manipulation, multi-robot cooperation, and human-robot interaction etc (see Fig. 5.1).

Fig. 5.12 illustrates an overview of this simulation. A robot is located at position $(0, 0)$ m with the initial view direction of 45°, while two objects are located at position $(2, 1)$ m and $(1, 1.25)$ m. Object 2 moves from its initial position with a constant velocity of 0.15 m per time step in x-direction and -0.15 m per time step in y-direction. The simulation takes 12 time steps.

At the first steps, the two objects can be located concurrently in the FOV, if the robot attends to one of them. Therefore, the position measurements of both objects are always updated. The robot switches its FOA between them.

At the last three steps, the two objects can not be located concurrently in the FOV. The object position estimation errors and velocity estimation errors are compared using the AP strategy and the RR algorithm in Fig. 5.13. Using the AP strategy, the robot attends towards the moving object O2, which explains the experimental result of human overt attention in *observer* state in the previous section: Attention is distributed on dynamic objects more frequently. Using AP strategy, a better position/velocity estimation is obtained in comparison with that using the RR algorithm, especially for the last three steps. From five simulations, the mean values of the position estimation errors using the AP strategy and the RR algorithm are 0.3832 m and 0.8547 m, respectively.

Simulation 3: an Application in a Multi-Robot System

In this simulation, a multi-robot system is investigated, which contains three pioneer mobile robots (see Fig. 5.14 left). One of them is the observing robot, robot 0 at $(0, 0)$ m, equipped with the aforementioned Sony VID30 camera on a pan/tilt platform (see Fig. 5.14 top right). Here, only the view direction change in the horizontal direction is considered. The maximal pan-angle is approximately 55°. The servo gain term of P-controller for pan-angle

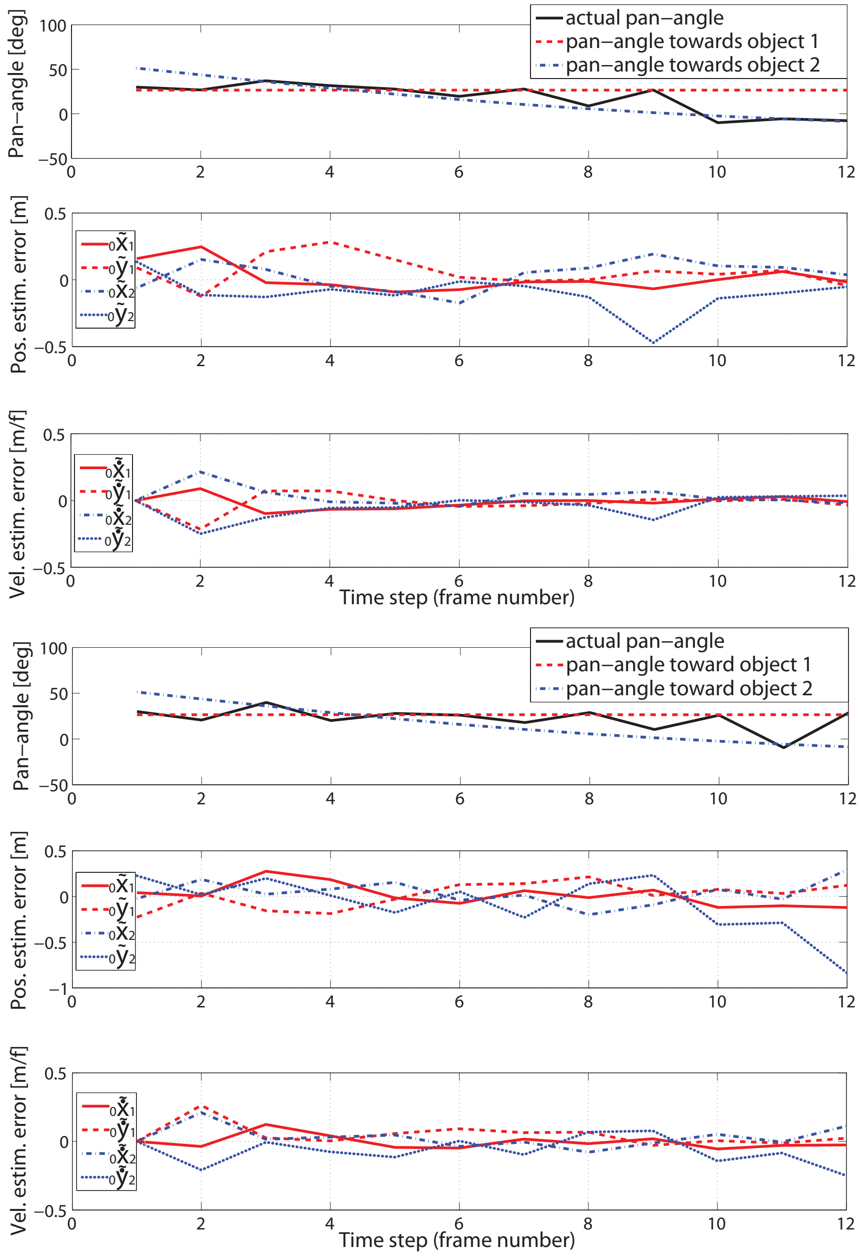

Fig. 5.13: Results of simulation 2. The actual robot view directions, position estimation errors, and velocity estimation errors of object 1 and 2 using the AP strategy (row 1 to 3) and the RR algorithm (row 4 to 6).

control is 5.

The other two robots are equipped with a colored marker each: robot 1 with a magenta marker and robot 2 with a cyan marker. Robot 1 is moving with a linear velocity of 0.1 m/s and an angular velocity of 0.3 rad/s from its initial position (3, -1) m. Robot 2 is moving with a linear velocity of 0.1 m/s and an angular velocity of 0.5 rad/s from its initial position (3, -1) m. Therefore, the trajectory of each robot is a circle.

To identify the markers and compute the relative position and orientation of robot 1 and 2 with respect to robot 0, the marker detection algorithm is implemented according

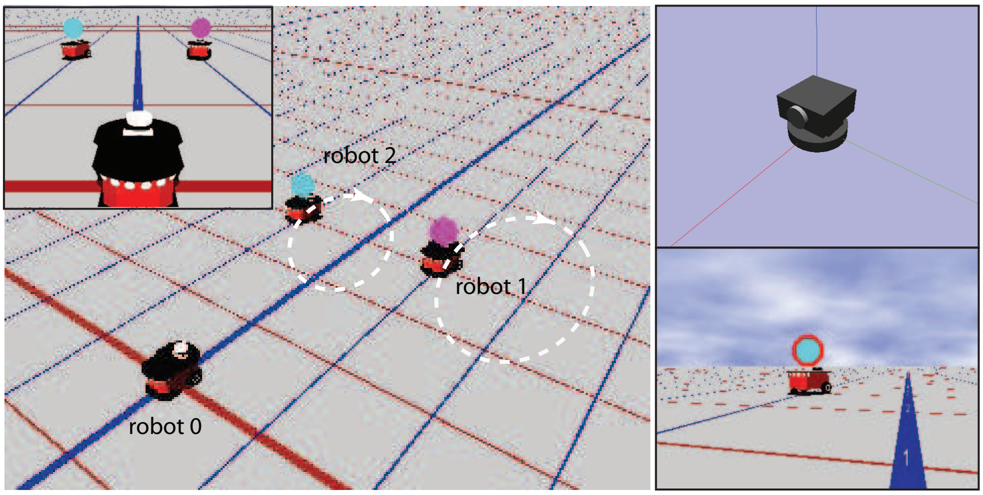

Fig. 5.14: Overview of simulation 3. Left: initial positions of robot 0, 1, and 2 and the predefined trajectories of robot 1 and 2, as well as a view from the robot perspective (top left); Top right: Sony VID30 camera with a pan/tilt platform; Bottom right: an input image and marker detection result.

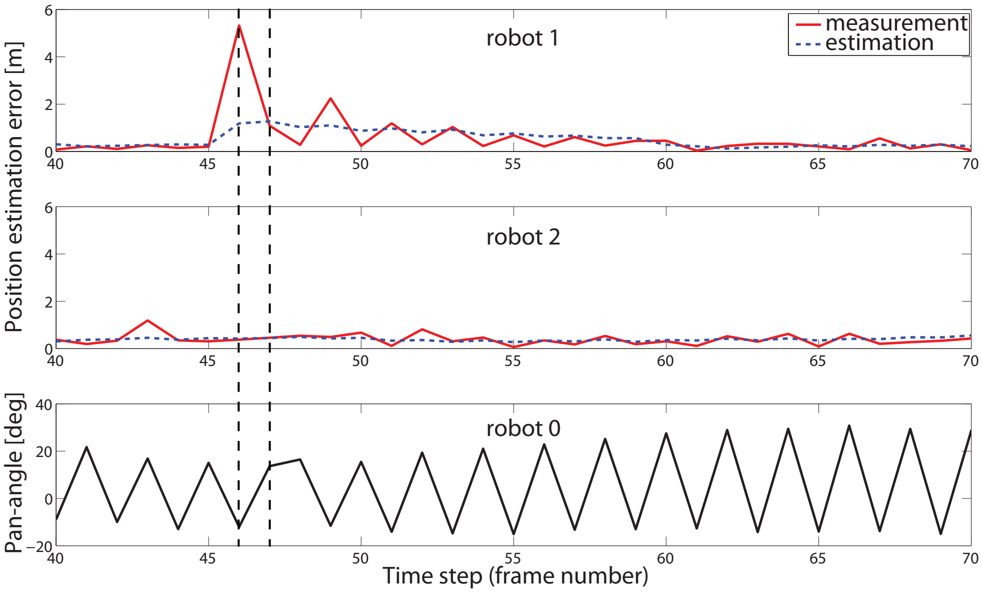

Fig. 5.15: Results of simulation 3. Row 1 and 2: the position estimation errors of robot 1 and 2. Solid lines: vision-based object position measurements or predictions (if the respective robot is not in the FOV); Dashed lines: position estimation errors. Row 3: the actual robot view direction at each time step. Positive pan-angle denotes that robot 0 is attending to robot 1. Negative pan-angle denotes that robot 0 is attending to robot 2.

to [6]. Since the radiuses of the markers are known, the 3D robot positions are computed from the 2D marker appearances (see Fig. 5.14 bottom right). The overall frame rate is approximately 3 fps.

Fig. 5.15 illustrates the position estimation errors of robot 1 and 2 as well as the pan-angles of robot 0. In general, robot 0 switches its FOA towards robot 1 and 2 repeatedly. At time step 46, the FOA of robot 0 is on robot 2. The position estimation error of robot 1 is very large, while the covariance of the estimation is also very large, since robot 1 is at the furthest position with respect to robot 0 at this moment. Therefore, robot 0 attends to robot 1 at time step 47 and tries to reduce the observation uncertainty of robot 1. Since the position estimation error of robot 1 at time step 46 is large, robot 1 is not brought into the FOV center at time step 47, which causes a repeated view of robot 0 on robot 1 at time step 48.

5.3.5 Discussion

Robotic Implementations of Human Behavior Inspirations

In the traditional computer vision and robotics domain, the ultimate objective of this attention planning can be regarded as simultaneous tracking of multiple objects. Tracking of multiple objects is more regarded as a parallel process in early visual processing than a serial process [31, 150]. However, a human-inspired computational models has yet to be proposed.

Based on the results of the experimental investigation of human gaze behavior during movement, human-inspired temporal attention planning is proposed here, to deal with multiple task-relevant objects which may not be located in the FOV concurrently. The inspirations from the previous section for robotic implementations are shown in Tab. 5.2. In an *observer* state, humans' FOA is mainly focused on task-relevant objects. Therefore, to reduce computational cost, only the optimal view directions towards individual task-relevant objects $\boldsymbol{\Omega}^*_{j,k+1|k}$ are chosen as candidates for the next time step. Humans are assumed to reduce perception uncertainty by switching their FOA frequently in the *observer* state. Therefore, the main objective for an optimal view direction is to minimize the overall position estimation uncertainty of multiple task-relevant objects. Moreover, by modifying the stochastic information in the KF, robots, like humans, prefer to attend to dynamic objects.

Human *observer*	Robot *observer*
Repeated targeting of FOA on task-relevant objects	$\boldsymbol{\Omega}_{k+1\|k} \in \{\cdots, \boldsymbol{\Omega}^*_{j,k+1\|k}, \cdots\}$ as view direction candidates
Frequent FOA switching	Minimization of the overall perception uncertainty J_1
Preference for dynamic objects	Process noise $\boldsymbol{Q}_{j,k}$ varying with object dynamic

Tab. 5.2: Inspirations from human behavior in the previous experimental investigation.

Multi-Object Tracking Using an Active Camera

In technical systems, various concepts of vision-based object tracking have been proposed such as tracking multiple objects using passive cameras [88, 104, 136], single-object tracking using an active camera or a multi-focal camera [35, 39, 175], and tracking multiple objects using multiple active/passive cameras [92, 190], where no attention planning is needed.

For tracking multiple objects using an active camera with a limited FOV, concurrent objects or tasks are present for vision resource allocation. Dealing with this scheduling problem, information-based multi-agent selection and reinforcement learning algorithms are used in [168] and [176], respectively. However, dynamic environments are not considered in those two works and attention is usually distributed on the nearest object first. Moreover, the RR algorithm, online Dynamic Vehicle Routing Problem (DVRP) with deadlines, and greedy scheduling policies of network packet scheduling are used to plan the camera switching in [64] [17] and [37], respectively, where multiple objects are either equally treated or assigned to pre-defined processing deadlines.

Potential Extensions

Based on this strategy, several extensions can be made:

- Although this section is dealing with the attention strategies if the robot is a static *observer*, robot motion can also be integrated. This will be explored in the next section.
- In practice, the mechanical limitations of camera panning and tilting should also be considered.
- Extensions considering other features such as object color or saliency can also be explored using the same attention planning strategy, in order to combine stimulus-driven and goal-directed attention selection mechanisms.

5.4 Towards Attention and Motion Coordination in Multi-Robot Systems

In the preceding section, a human-inspired temporal attention planning strategy is proposed to reduce the overall perception uncertainty in a multi-object system, where a robot acts as an *observer*. In addition to the *observer*, an *actor* behavior is also necessary for performing robot tasks and should be integrated. Therefore, a human-inspired extension is made here, in which a behavior-based combination of *actor* and *observer* is applied on robots with limited FOV.

One common scenario where multiple target objects exist is a multi-robot system (see Fig. 5.1). From the perspective of an individual robot in those systems, both the objects to be manipulated and the other robots are task-relevant. The possibility of utilization of the proposed attention strategy in a multi-robot system is illustrated in the previous section. In this section, formation problems are taken as examples to demonstrate the performance of *actor* and *observer* coordination in multi-robot systems.

5.4.1 State of the Art

For certain tasks beyond the capability of one single robot and for an improved task performance, more and more multi-robot systems are used [78, 144]. Multi-robot systems can be classified into heterogeneous systems containing different types of robots with different capabilities, and homogeneous systems containing robots of the same type. The organization and control can be centralized or distributed.

One of the tasks attracting much attention in the field of multi-robot systems is formation task. A formation task can be solved in a leader-referenced, neighbor-referenced, or unit-center referenced structure [11]. The task is to reduce the formation error to zero, while the formation error is computed using distributed vision sensors on multiple robots in this section.

In most vision-based multi-robot systems, the robots are equipped with on-board omni-directional cameras, such that they can update the position information of the other robots consistently [40, 50, 122, 141, 145, 193]. However, omni-directional cameras have the disadvantages of high cost, difficult calibration, as well as low resolution on the image borders, and therefore, a small sensing range. Furthermore, for humanoid robots or other systems where the anthropomorphismus is desired, an omni-directional camera is not appropriate. Passive cameras with limited FOV have also been applied in some systems, in which communication and data transfer among the robots are needed [63, 159, 205]. In those systems, no active actions of vision systems are considered, such as searching, attending, or view direction planning. Some other systems deploy global vision to acquire the position information of robots [7, 11, 24]. Furthermore, active vision agents pre-supposing communication among robots [119] and pan-tilt-zoom cameras combined with odometry [154] have been proposed.

To sum up, few decentralized multi-robot systems using active cameras exist without communication in the robot team. One of the challenging problems using active cameras is that each on-board camera cannot provide all the information of the environment all of the time due to limited FOV, especially for multiple mobile robots.

The multi-robot system considered here is a homogeneous, distributed system, in which each robot is equipped with an active camera with mechanical constraints such as limited FOV and limited pan-/tilt-angles. A totally distributed formation task is to be accomplished without communication among robots or global sensory facilities.

5.4.2 Behavior-Based Attention and Motion Coordination

From the experimental investigation of human behavior in Section 5.2, two macroscopic conclusions are made: 1) Due to limited FOV and multiple target objects, observing and moving are mainly separated in the *observer* and the *actor* state; 2) Attention switching is mainly conducted in the *observer* state. Moreover, it is suggested in [100] that eyes generally guide manipulation, in which the relevant eye-movements usually precede the respective motor acts.

Inspired by these, a behavior-based formation task is accomplished, in which the robot performance is based on a set of predefined behaviors [11, 61]. Two major behaviors are defined for each robot in a formation task: an *observer* behavior and an *actor* behavior, illustrated in Fig. 5.16. In the *observer* behavior, attention is distributed on task-relevant objects. Attention planning is conducted according to the proposed strategy. The operating robots search for and attend to multiple task-relevant objects, including static target objects or the other task-related robots. The formation error is then calculated. Activi-

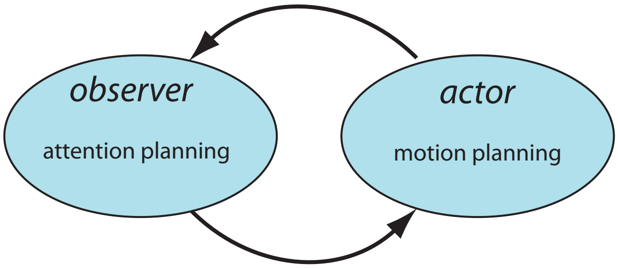

Fig. 5.16: Combination of two human-inspired robot behaviors *observer* and *actor*.

ties such as motion are conducted mainly in the *actor* behavior, aiming at reducing the formation error to zero.

The control of the robots is distributed and independent, while the switching of behaviors of each robot is asynchronous. Using this behavior-based attention and motion coordination, formation tasks can be solved in a distributed manner.

5.4.3 Simulation Results

To demonstrate the human-inspired behavior-based multi-robot attention/motion coordination, two formation tasks are simulated in Player/Gazebo simulator. In the first simulation a leader-referenced formation task is performed, while a neighbor-/unit-center referenced formation task is conducted in the second simulation.

In each simulation, the same conditions are used as in simulation 3 of Section 5.3.4: Robots are equipped with the active cameras Sony VID 30 and colored markers (see Fig. 5.14). Only the view direction change in the horizontal direction is considered; the servo gain term of P-controller for pan-angle control is set to 5. Since the control design and the motion planning are not the focus of the simulations, a simple position-based P-controller is used for the robot motion control. The P-term is proportional to the control error. Image processing is at a frequency of 3 Hz.

Simulation 1: a Leader-Reference Formation Task

The first simulation is an extension of simulation 3 in Section 5.3.4. Two robots are equipped with colored markers (robot 1 with a magenta marker and robot 2 with a cyan marker, see Fig. 5.17 left) and perform pre-defined trajectories as leaders: robot 1 moves along a sinus-shaped trajectory from its initial position at $(3.5469, -0.9401)$ m, while robot 2 moves along a straight line from its initial position at $(3.3891, 1)$ m (see Fig. 5.18). The average velocities of robots 1 and 2 in x-direction are approximately 0.23 m/s. The range of the velocity of robot 2 in the y-direction is from -0.1672 m/s to 0.29995 m/s. Robot 0, equipped with an active Sony camera at the initial position $(0, 0)$ m, is a follower and desired to form an isosceles triangle with the other two robots. The length of the side formed by robot 0 and robot 1 should be equal to the length of the side formed by robot 0 and robot 2, while the interior angles of these two sides are equal to 65°.

Fig. 5.18 illustrates the trajectory of robot 0 (circles on the dotted line) and the rectangles formed by robots 0, 1, and 2 (large triangles in solid lines) at the every third time

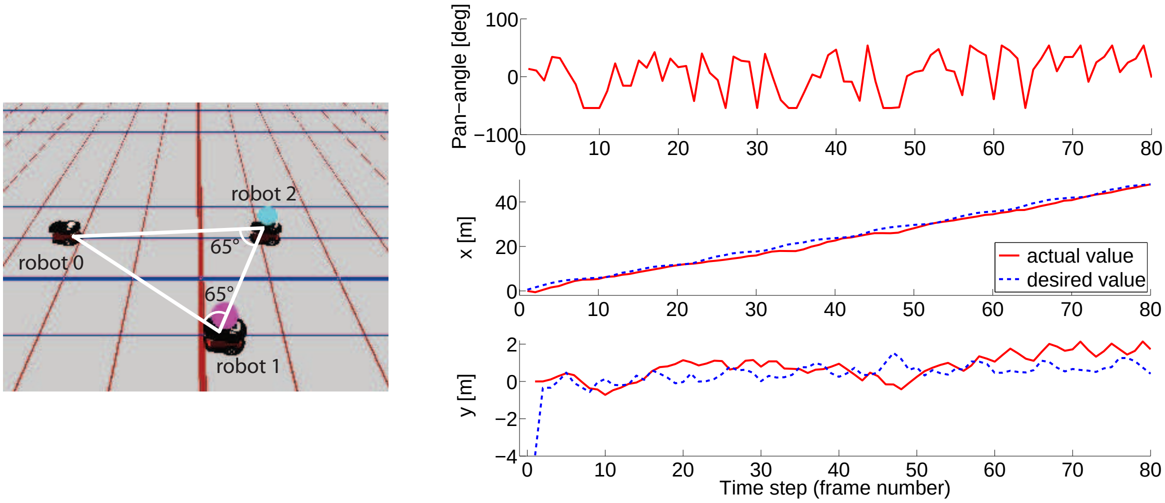

Fig. 5.17: Simulation results in a leader-referenced formation task. Left: desired formation of robot 0 with robot 1 (magenta) and robot 2 (cyan); Right: pan-angle, x- and y-position (from top to bottom) of robot 0. Solid lines: actual value; Dashed lines: desired value.

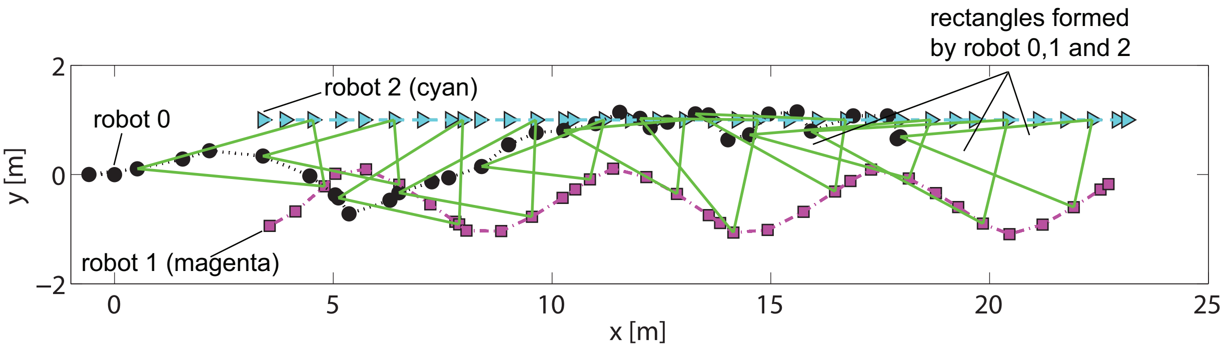

Fig. 5.18: Overview of the leader-referenced formation task (until time step 35). Circles on the dotted line: trajectory of robot 0 at each time step; Squares on the dash-dot line: trajectory of robot 1 at each time step; Triangles on the dashed line: trajectory of robot 2 at each time step; Large triangles in solid lines: rectangles formed by robot 0, 1, and 2.

step. The pan-angle and x-/y-positions of robot 0 are shown in Fig. 5.17 right. The solid lines indicate the actual positions, while the dashed lines indicate the desired positions computed from the real positions of robots 1 and 2. Over a distance of 50 m, the average position error in the x-direction is 1.2484 m and the average position error in the y-direction is 0.6321 m. The leader-referenced formation task is successfully performed by robot 0.

Simulation 2: a Neighbor-/Unit-Center Referenced Formation Task

In the second simulation, a neighbor-/unit-center referenced formation task is investigated. Three to six mobile robots are used, which are equipped with the active Sony cameras and colored markers (see Fig. 5.19 left and Fig. 5.20 left). The x-/y-directions of the world

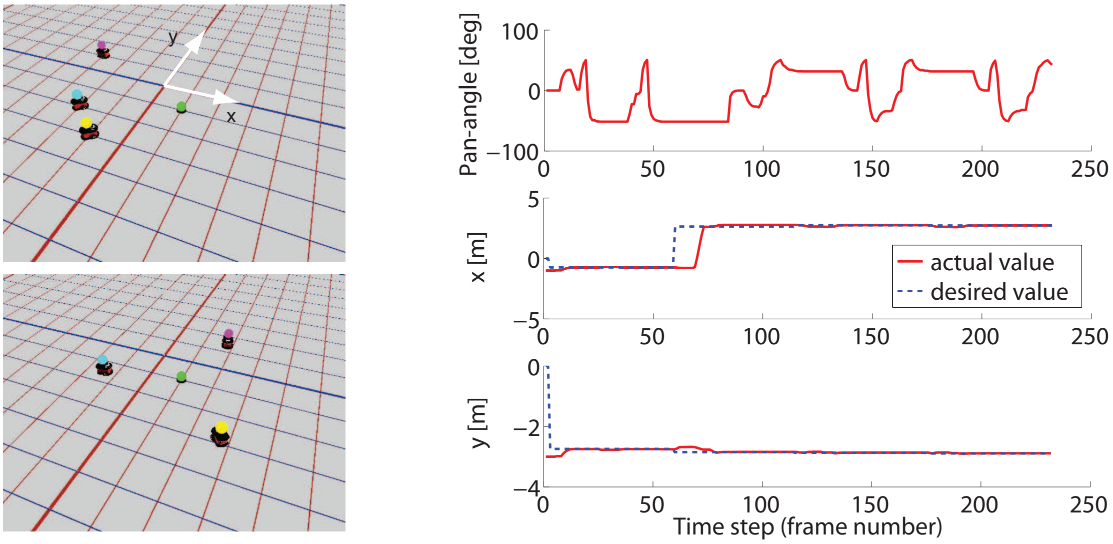

Fig. 5.19: Simulation results in a 3-robot system for a neighbor-/unit-center referenced formation task. Left: initial positions of the robots (upper) and final positions of the robots (lower); Right: pan-angle, x- and y-position (from top to bottom) of one robot. Solid lines: actual value; Dashed lines: desired value.

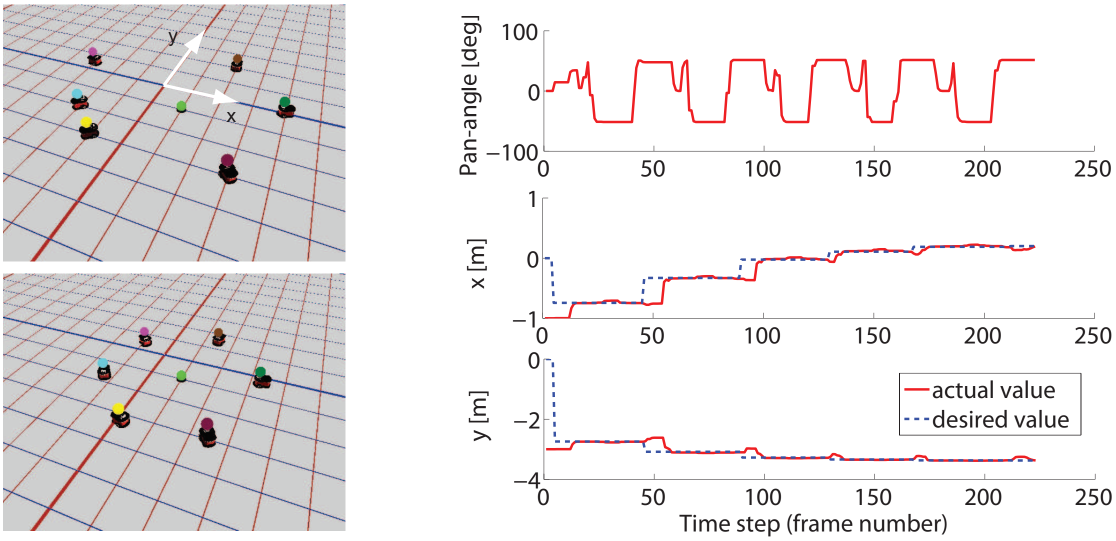

Fig. 5.20: Simulation results in a 6-robot system for a neighbor-/unit-center referenced formation task. Left: initial positions of the robots (upper) and final positions of the robots (lower); Right: pan-angle, x- and y-position (from top to bottom) of one robot. Solid lines: actual value; Dashed lines: desired value.

frame are also illustrated. The robots should be equally distributed around the target object, a green sphere at position (1, -1) m, at a distance of 2.5 m each from their randomly chosen initial positions.

The simulation results in a 3-robot system are illustrated in Fig. 5.19, while Fig. 5.20 shows the results in a 6-robot system. In the sub-figures from top to bottom, the pan-angle and x-/y-position of one of the deployed robots are illustrated. The blue lines denote the desired values, while the red lines denote the actual values. The robots reached the desired positions in the end. Due to the limited FOV and the limited mechanical constraints of pan-angles, robots have to stop to turn back to the neighbored robots in their opposite directions, resulting in a relatively long waiting time. Since each robot only considers the positions of the neighbored robots, the convergence velocity is independent of the number of the robots deployed in the system theoretically. Therefore, this strategy can be generalized.

5.4.4 Discussion

A team of mobile robots equipped with active vision systems with limited FOVs and limited pan-angles is firstly deployed to accomplish a formation problem in a totally distributed manner. Inspired by human behavior, a simple behavior-based solution is proposed, where attention planning in the *observer* state and motion planning in the *actor* state are integrated.

Comparison to RR Algorithm

The main advantage of the proposed algorithm compared to the RR algorithm is that a replanning is conducted if the vision-based position estimation is not satisfied, for instance if the target object, which should be attended to at the current time step, is located far away from the center of the camera FOV due to erroneous estimation of the previous time step or self-motion. In this case, this target object will be attended to again. Applying the RR algorithm to a mobile robot in a dynamic environment often causes divergent performance.

Comparison to Conventional Vision-Based Multi-Robot Systems

Compared to the conventional vision-based multi-robot systems containing omnidirectional cameras, global sensory units, or communication, the main challenge of deploying robots with limited FOVs is the insufficient information update, which slows down the task performance and may cause instabilities in a more complex scenario. However, the fully decentralized, active attention planning is also desired in many systems (see Fig. 5.1), which is a promising future research direction.

Limitations

For the formation tasks, mathematical proof of the system stability and convergence is still lacking. Furthermore, multi-modal attention should be considered in order to overcome the insufficient information update in dynamic environments.

5.5 Summary

In this chapter, the problem of how to plan a scan path in a sequential aspect and solve a temporal attention shift problem, which is embodied as view direction planning in the 3D task space under limitation of visual FOV, is investigated . For tasks containing multiple task-relevant objects which may not be located in the FOV concurrently, attention planning is strictly required to ensure overall task accomplishment.

An experimental study of human subjects was conducted to examine and analyze human eye movement and body movement while performing a coordinated formation task with more than one task-relevant objects. The results imply that human FOA switches repeatedly and frequently among task-relevant objects, with a preference for dynamic objects. Moreover, gaze behavior and body movement are also combined in a switching manner. The contribution of the experimental study is not only limited in the application of robotic systems, but also means a lot for human-robot cooperation, such that humans feel more confident and natural when working together with robots.

Based on the experimental results of human behavior investigation, a spatio-temporal attention planning for multi-object systems is proposed, resulting in a significantly reduced overall perception uncertainty and a high similarity of temporal attention distribution between robots and human subjects. Furthermore, the temporal aspect of a coordinated behavior between visual attention and body motion in a multi-robot system is studied exploratively. Contribution to the state of the art is the first-time utilization of active vision sensors with limited FOV in a multi-robot formation task, deploying a coordination of attention and motion planning.

This human-inspired attention planning completes the development of robot attention control in the temporal aspect, considering evaluation of a technical system in the task space. The results bring essential insights not only into active vision-based robot/multi-robot system development but also into human-robot system research in the context of efficient and human-like visual information acquisition and processing. High-level combination of attention strategies in the image space and in the task space is envisioned for future work.

6 Conclusions and Future Directions

6.1 Concluding Remarks

Nowadays, the deployment of technical systems in complex and unstructured everyday environments has become a tendency of robotic research, in which the limited computation capacity and real-time constraints become the bottleneck of the system development. Cognitive abilities to select essential information from a large amount of sensory data are important and necessary for an autonomous mobile robot.

From the extensive works of human visual attention in biology, cognitive psychology, and neuroscience, it is known that human attention is one of the most powerful cognitive processes dealing with visual information selection. Considering the challenges arising in the aforementioned context, it is envisioned to develop both biologically plausible and technically applicable robot visual attention strategies and to bridge the gap between fundamental studies and specific technical realizations.

This thesis focuses on the investigation of various aspects of robot attention control. Robot goal-directed visual attention strategies are explored from three different perspectives: the stimulus-dependent aspect, the task-relevant spatial aspect, and the task-relevant temporal aspect. Applications and examples are presented for demonstration and evaluation. The main approaches along with the main results are highlighted below.

From visual data input, a large amount of information about robots' operating environments is obtained. Conventional robot applications have only considered extraction of the task-relevant information, while stimulus-dependent perception has been ignored. However, due to robot mobility and environment dynamics, task-irrelevant stimuli such as abrupt appearing/disappearing of objects, dynamic characteristics, or appearance variation of objects, etc. can influence robot task accomplishment significantly, increasing system uncertainty or even inhibiting robot tasks. Although they are not directly related to robot tasks, they definitely play a key role in cognitive technical systems. Chapter 3 addresses this bottom-up perception problem during a robot performing a task by solving two main issues: the definition of task-irrelevant stimuli and the determination of a reasonable and economical time point to attend to the task-irrelevant stimuli considering environment dynamics. Two metrics, local surprise and global surprise, are defined in this context. Local surprise combines static saliency and temporal novelty in the 2D image space using an information-based approach, while global surprise emphasizes the dynamic changing of the robots' operating environment. A high global surprise indicates a high environment uncertainty and alerts the robot system to attend to the current local surprise maximum, which has probably caused the uncertainty increase. Through interconnections of local surprise and global surprise, the sensitivity of robot systems to the operating environment is greatly improved while preserving primary robot tasks. Limitations are mainly located in expensive quantitative evaluations and the need for a more sophisticated means of environment monitoring.

The major improvement provided by robot attention control is efficient task-relevant information acquisition and processing. Conventional approaches in the technical realizations use top-down information to bias bottom-up perception in the early vision. Offline training is commonly an unavoidable process to find a best representation of the task-relevant information using primary features in bottom-up attention selection models. Compared to the state of the art, a more flexible and adaptive variation of top-down biased bottom-up attention selection is presented in Chapter 4. A prediction-verification-adaptation loop is established using a multi-focal vision sensor configuration. A prediction-correction inner loop for estimation of the representation of top-down information in the current environment using a Kalman-filter enables a reasonable weighting of past experience and current environment modeling and manifests itself in an improved efficiency in terms of fewer necessary fixations for a number of target objects. Furthermore, a complementary robot attention approach is proposed, in which an autonomous switching between top-down and bottom-up attention mechanisms is applied for the first time in an application scenario, in which more than one target object with totally different appearances are searched for. The robot attention mechanism is adapted to the internal robot states including searching, operating, and exploring. Efficient decision making completes the system autonomy at task changing and robot state changing. Further improvements such as the ego-motion compensation issue and the integration of other sensor modalities can be considered.

After Chapters 3 and 4 explore the determination of task-irrelevant stimuli and task-relevant information from a spatial perspective, Chapter 5 addresses the temporal aspect of robot attention control. The central problem is how robot attention should be distributed along the time scale if more than one task-relevant object is located in the environment, especially in a multi-robot system containing both static and dynamic targets of the same or different importance from an individual robot's point of view. Fundamental research in cognitive psychology and neuroscience have investigated sequential attention planning of human subjects, but few works consider it during coordinated multi-agent motion. The hypothesis that attention distribution differs significantly in different states of human behavior is verified. Based on the experimental results of human behavior investigation, a spatio-temporal attention planning for multi-object system is proposed, resulting in a significantly reduced overall perception uncertainty and a high similarity of temporal attention distribution between robots and human subjects. Furthermore, inspired by the human attention behavior, the temporal aspect of a coordinated behavior between visual attention and body motion in a multi-robot system is studied exploratively. Contribution to the state of the art is the first-time utilization of active vision sensors with limited FOVs in a multi-robot formation task, deploying a combination of attention and motion.

Summarizing, the overall advantages of robot attention strategies proposed in this thesis are qualitative improvements of sensitive awareness of environment dynamics, efficient and flexible task-relevant information enhancement, and adaptation to changing environments or tasks, as well as the reduced perception uncertainty and the extended FOV through the temporal attention planning. Application-oriented attention control considering robot characteristics has been studied in a general and integrated manner for the first time. The contributions advance the state of the art in robot attention development and provide valuable insights for future research.

6.2 Outlook

Vision is one of the most powerful tools for environment perception. Along with the rapid development of sensor technology and data transfer systems, the demand on high-speed information processing is increasing for applications in mobile technical systems or in dynamic environments. Biologically inspired visual attention systems have been proposed to be one of the most efficient solutions for this challenge. Although much is known from cognitive psychology and systemic neuroscience, visual attention is still a relatively young research field in the robotics domain. There is still a large number of open questions and interesting future directions remaining, some of which are suggested below.

- *Multi-modal attention* - The main advantage of visual attention in a technical system is to achieve high efficiency compared to conventional goal-directed approaches. However, it does not guarantee a successful task accomplishment. Multi-modal attention is then entered into the agenda. With an elaborated sensor fusion together with other sensor modalities such as haptic sensor, lasers, sonars, and auditory sensors, the system capability and the extent of functioning can be enhanced and complemented. To date, multi-modal attention has been applied to a limited extent in humanoid robots or human-robot interaction areas and mainly exhibits a redundant system structure for resources allocation. Coordination and cooperation of multi-modal sensors have not yet been considered. Interesting future directions for multi-modal attention also exist in multi-robot systems such as development of joint attention between robots [89].

- *High-level semantic perception* - Robotic applications in everyday life require a high-level cognitive ability of technical systems. A very interesting direction is attentional semantic perception. Scene interpretation, context recognition, and also semantic visual SLAM among others are challenging and exciting research areas of artificial intelligence (AI), for which visual attention can provide a highly efficient solution.

- *Bi-directional improvement of fundamental studies and robotic research* - From an engineering point of view, the implications from fundamental studies cannot be easily transferred and implemented in robotic applications. A tighter link between the two research fields is needed. A reciprocal relationship is envisioned, from which both bio-inspired robots and AI-inspired biology and psychology can benefit.

Research on robot visual attention will have a large impact on the development of cognitive abilities of technical systems. This kind of biologically inspired perception system is expected to be an inevitable component of modern robotics technology.

A Experimental Platform: The Autonomous City Explorer (ACE) Robot

In the project Autonomous City Explorer (ACE), an interactive robot is designed to find its way to a given destination in unknown urban environments by interacting with pedestrians. In a recent experiment the robot managed to successfully travel a distance of 1.5 km from the campus of the Technische Universität München to Marienplatz, the central square of Munich [213, 217].

In this thesis, the ACE robot was used in the diverse experiments to demonstrate various attention strategies. The robot platform, the active vision system, and a goal-directed attention control strategy developed and applied in the framework of the ACE project are introduced here.

A.1 Hardware Components

In its current setup, the ACE robot comprises a differential drive mobile platform with wheel encoders, developed by BlueBotics SA, two laser range finders for navigation and traversability assessment, a loudspeaker, a touch-screen, an animated mouth, as well as a sophisticated stereo vision system based on a multi-focal active camera head for image processing (see. Fig. 1.2). The complete system measures 78 cm in length, 56 cm in width, and 178 cm in height, including the camera head, and weighs approximately 160 kg.

The mobile platform has a maximum payload of 150 kg and is moved by two wheelchair drive wheels (30 cm diameter) with differential drives and treads. It has two castor wheels (12 cm diameter) at the rear and two castor wheels on springs at the front (10.5 cm diameter). The maximum velocity is 1.4 m/s, the maximum acceleration 1.35 m/s^2. It has an autonomy of up to 10 km depending on the paving. The climbing ability of the platform has been thoroughly tested, since this is an essential factor for outdoor navigation. The robot is capable of climbing a slope of 6° and steps of 35 mm. For urban environments this means that the robot can safely navigate on sidewalks and smooth surfaces but must avoid larger steps, such as the curbside.

The software is run on two on-board Linux PCs (one PC for navigation and interaction and one for vision processing) with four 2.2 GHz cores each, powered by an array of rechargeable lithium polymer batteries that provide power for up to 8 hours. A third PowerPC independently controls the differential wheel platform and receives asynchronous driving commands from the navigation PC. All processes run at fixed update rates in a pull architecture fashion, meaning data is queried from sensors and processes are refined at fixed intervals.

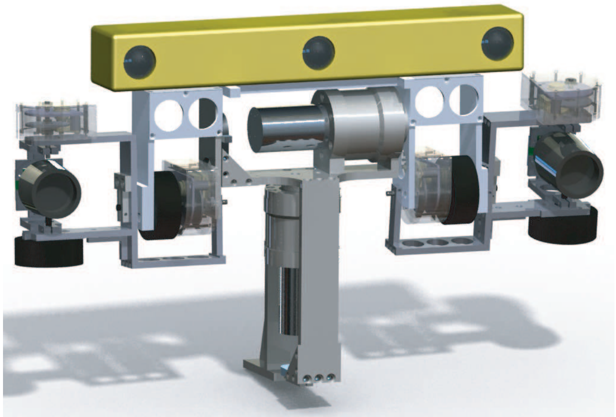

Fig. A.1: New revision of the high-performance active camera platform [97, 225]

A.2 The Active Multi-Focal Vision System

The design of the multi-focal high-performance vision system is based on the multi-focal vision system, which has been developed for the humanoid robot *LOLA* [97]. It comprises several vision sensors with independent motion control which strongly differ in fields of view and measurement accuracy. High-speed gaze shift capabilities provide fast situational attention changes of the individual sensors. Thereby, large and complex dynamically changing environments are perceived flexibly and efficiently.

This multi-focal vision system generalizes the foveated vision concept by introducing independent motion control of several vision sensors, thus adding more flexibility in sensor resources allocation [97]. This feature is particularly beneficial in robot navigation and scene observation, providing higher robot localization accuracy and tracking performance than conventional systems.

The vision system consists of a wide-angle stereo camera mounted on a central pan/tilt-platform, see Fig. A.1. As an upgrade from the previous vision system, the main camera is now a 3-sensor, multi-baseline Bumblebee XB3 by Point Grey Research Inc., with enhanced flexibility and accuracy because of the switchable baseline [225].

In addition, two telephoto cameras are gimbal-mounted on the central platform with 2 DoF each. Aperture angles of approximately 85° (wide) and 20° (telephoto) and focal-lengths of 2 mm and 25 mm, respectively, are provided. The central platform is driven by DC drives with harmonic drive gears, the gimbal-mounted cameras by brushless DC direct drives providing high torques and accelerations at small dimensions and weights. Top open-loop speeds and accelerations measured are 8,400°/s and 100,000°/s^2. An embedded RISC processor (MPC555, Motorola) controls the camera motions on joint levels. The position feedback for the control loop is provided by incremental magnetical encoders (512 counts per motor-revolution) on the dc motor side and processed in the RISC processor. For the brushless-motor side, position is measured by light-weight and small optical absolute encoders, which were developed specifically for this camera head. The position is encoded in a 16-bit gray code on the encoder disc, processed directly in the respective sensor and can be requested via I2C.

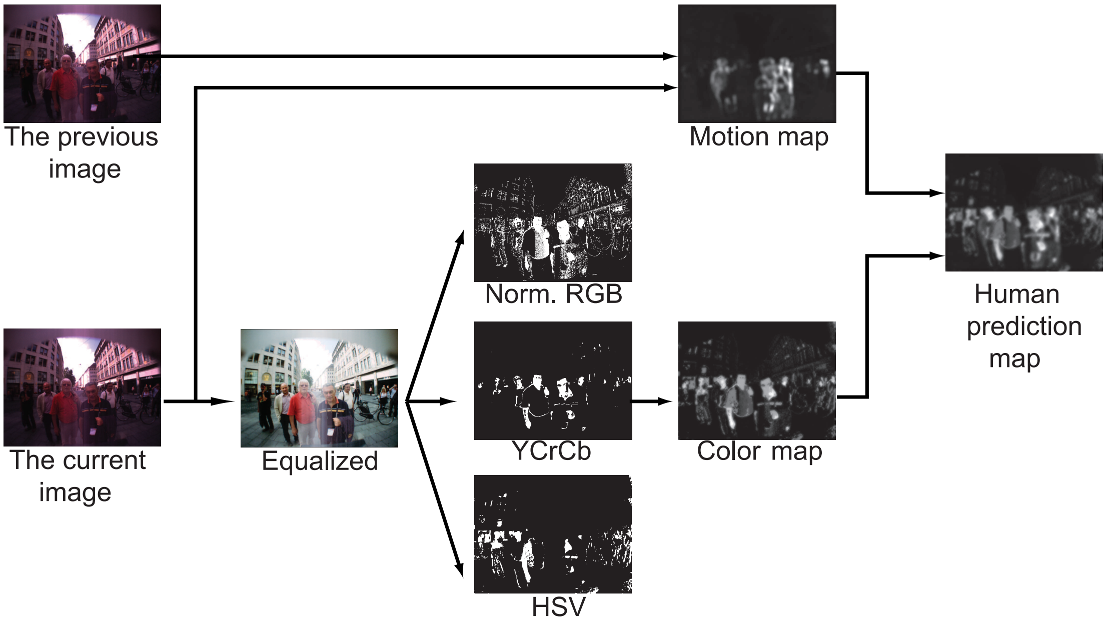

Fig. A.2: Human prediction map model based on color and motion maps.

The system is encapsulated and accepts camera pose commands from a higher-level decision and planning unit via a CAN-based interface. The system body is made of aluminum alloy. Overall dimensions are 37 × 30 × 5 cm and the weight is 2.2 kg.

In the experiments described in this thesis, the wide-angle stereo camera mounted on the pan/tilt-platform was mainly used for visual perception.

A.3 Attention Control for Human-Robot Interaction

A top-down biased bottom-up attention control has been proposed for human detection in the framework of the ACE project.

For successful human-robot interaction, pedestrians should be detected first. Most proposed human detection models are based on feature extraction and classification [105]. They are robust but not real-time capable, or are highly dependent on high resolution, which is not suitable for applications in highly dynamic outdoor environments. Some strategies based on skin color are also proposed in [174] and [191] which can work in real-time but not robustly enough. Therefore, simple algorithms are combined to achieve a relatively robust and real-time capable human detection approach. Considering the common characteristics of pedestrians, for instance, the skin color and motion, a goal-directed attention system is proposed for human detection and human tracking.

Human Detection

The attention model for human detection is illustrated in Fig. A.2. From an incoming image sequence two consecutive images are taken into account to compute a human prediction map, which is derived from feature maps of the skin color and motion. Feature maps are normalized and weightedly combined into the final prediction map.

– Color map To reduce the influence of different lighting conditions, an equalization in the R-, G-, and B-channels of the input images is executed. To achieve the robustness, the color feature map is the weighted sum of three color maps in different color spaces, inspired by [174] and [191]. For each color space the corresponding grayscale result image is given rules by which a certain pixel is either determined to be skin color (pixel set to 255, white) or not (pixel set to 0, black). Color spaces used by this model are normalized RGB, HSV, and YCrCb.

– Motion map The input data for the motion feature map is the absolute value of the difference between the grayscale values for each pixel in two consecutive images. The result is one grayscale image showing intensity changes from the previous image to the current image. To compensate for the small motion caused by shaking of the camera, which results in little offsets l and k in the image horizontal and vertical directions between images in the sequence, the grayscale motion image as described above is computed several times while the two input images are shifted towards each other by one pixel in one direction at a time. Then, an optimization problem is solved as follows:

$$\min_{k,l} \sum_{i=1}^{N} \big(I_1 \left(x_i, y_i \right) - I_2 \left(x_i - k, y_i - l \right) \big), \tag{A.1}$$

to compute the offset $k \in (-k_{max}, k_{max})$ and $l \in (-l_{max}, l_{max})$. The total pixel number is denoted by N, where each pixel i can be addressed by a pair (x_i, y_i). In order to use this method more efficiently regarding the computational cost, N is set to a smaller value representing several smaller areas in the image, while k_{max} and l_{max} are chosen within reasonable limits.

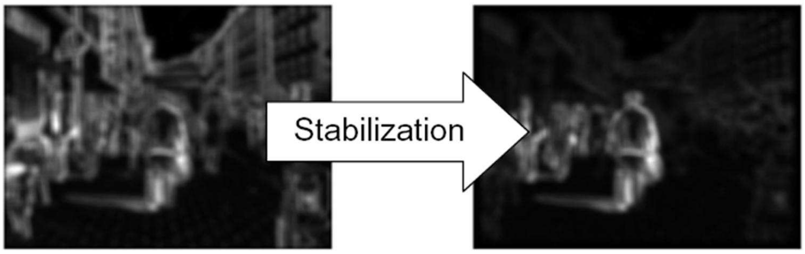

Fig. A.3: Motion maps before and after the stabilization algorithm.

Human Tracking

The most bright position in the final human prediction map is the position with the highest probability to contain a pedestrian, and, therefore, becomes the robot FOA. For multiple positions containing the same brightness in the human prediction map, the position with the maximum area in the 2D image is chosen. The camera platform is controlled to locate this position into the image center, which also shows the robot current visual interest in interacting with the selected pedestrian.

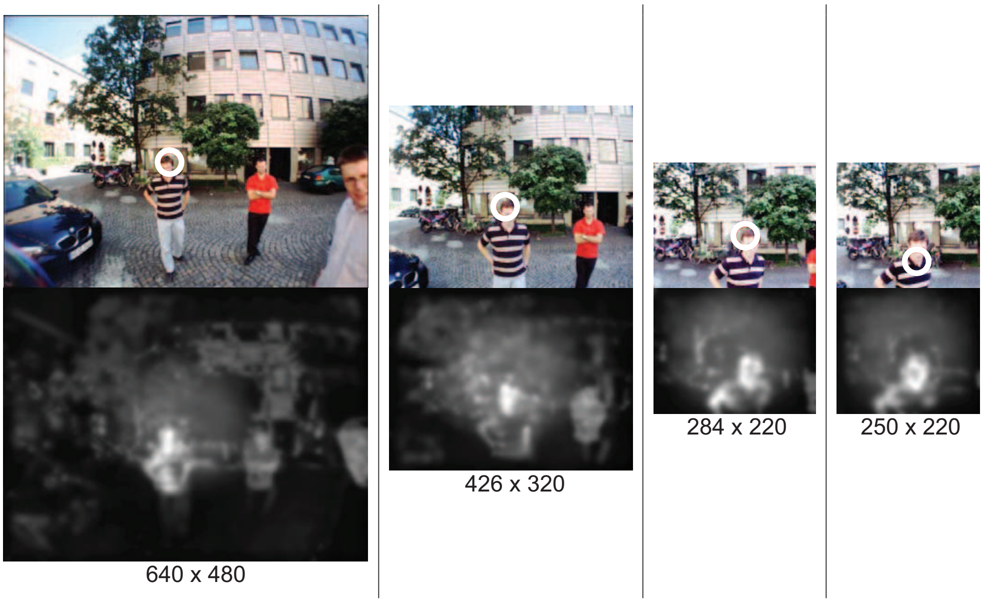

Fig. A.4: Determination of search windows for efficient image processing. Upper images: the search windows on the original images with current robot FOA denoted by circles; Lower images: the search windows on the human prediction maps.

To lower the computational cost, a search window (area of interest) is constrained for human tracking as follows. If the selected position is very close to the principle point, a small search window is placed around the image center for the next step. In contrast, if the search window has already been reduced and the FOA position is close to the search window boarders, the size of the search window is enlarged. The size of the search window varies between 250 × 220 to 640 × 480 pixels (see Fig. A.4). This method also facilitates keeping the robot's attention focused on one target/pedestrian and not to switch back and forth between several points of interest.

Experimental Evaluation

During an 11-minute test run, pedestrians were usually located in a range up to 8 meters away from the camera. The results in the outdoor environment were very pleasing, namely a detection rate of 92.6%, approximately. The detection rate in indoor environments is relatively low due to the large number of distractors with similar colors to skin color.

Since the search window for human detection varies, the computational cost for each step also varies. Working on the hardware described previously, the maximum computation time for one input image of 640 × 480 pixels is 0.7 s, while the minimum computation time using images of 250 × 200 pixels is 0.2 s. The color map computation is also implemented using Graphics Processing Unit (GPU), which brings an additional speed-up of approximately 30%.

Fig. A.5 left illustrates the results of human detection. The images in the first row are input images before camera motion, while the images in the second row are input images after camera motion. The detected humans regarded as the robot FOA are indicated by circles. The camera head attended to those humans in order to bring them into the image

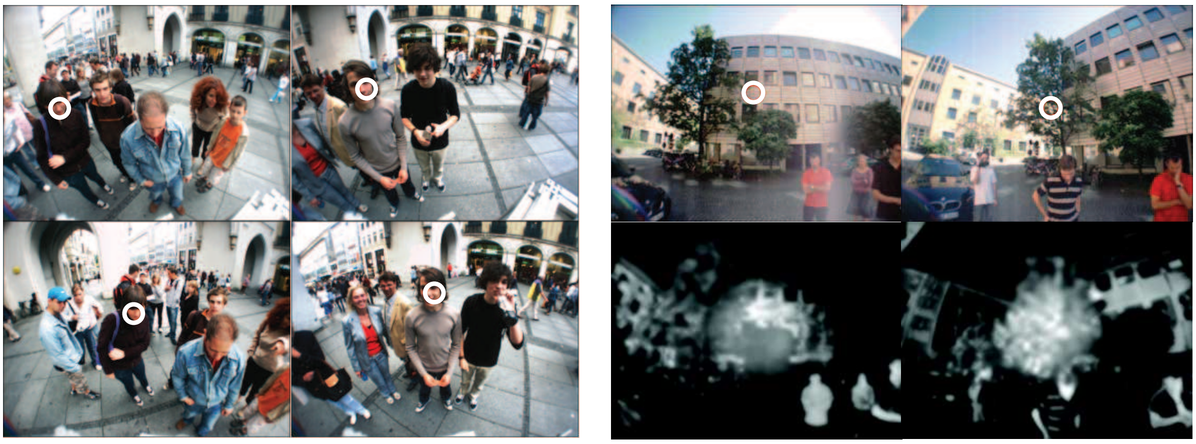

Fig. A.5: Left: attention selection before (upper) and after (lower) the camera motion control. Right: possible false positive errors on the original images (upper) and their respective human prediction maps (lower). Circles: current robot FOA.

center. Through this behavior, ACE shows its interest in the current interaction partner.

Two possible false positive errors are shown in Fig. A.5 right. In the left column a part of a building in the background is detected due to a similar color to skin color. In the right column, swinging leaves are detected. To avoid those errors, the 3D positions of the image region candidates should be constrained.

Further descriptions of hardware and software design in the other aspects such as path planning, human-robot interaction, gesture recognition, and the diverse results of the field experiment can be found in [213–216].

B A High-Speed Multi-GPU Implementation of Bottom-Up Attention

In the dynamic robot vision, high-speed early visual processing can enable high-speed perception and recognition of sudden events, which reduces the overall latency of image processing and ensures real-time decision making. Another practical advantage of high-speed image processing is to reduce the influence of inter-frame motion such that the motion blur or ego-motion can be ignored in computation. Bottom-up attention selection is implemented on a platform of multiple Graphics Processing Units (GPUs) to significantly accelerate the compute-intensive but highly parallelizable saliency map computation.

B.1 State-of-the-Art Implementations

Various implementations have been proposed. The details of the implementations may differ, but most works are based on the saliency map model proposed in [84].

A real-time implementation of the saliency-based model of visual attention on a low power, one board, highly parallel Single Instruction Multiple Data (SIMD) architecture called Protoeye is proposed in [140]. Protoeye consists of a 2D array of mixed analog-digital processing elements (PE). The operation of visual attention computation is optimally distributed to the analog and digital parts. The analog part is used to implement the spatial filtering-based transformations such as the conspicuity operator and the normalization, while the digital part is used for the logical and arithmetical operations such as the integration of conspicuity maps. The implemented attention process runs at a frequency of 14 fps at a resolution of 64×64 pixels.

Another real-time implementation of a selective attention model is proposed [203], in which intensity features, edge features, red-green opponent features, and blue-yellow opponent features are considered. To achieve real time ability they implement a Gaussian pyramid with only 5 layers on an input image of 160×120 pixels. They use a look-up table (LUT) to replace the Gaussian pyramid operation to save the calculation time and also use retina-topic sampling to calculate symmetry information. For each channel, four feature maps are computed instead of six feature maps in the saliency map model proposed in [84]. Their model can perform within 280 ms at Pentium-4 2.8 GHz with 512 MB RAM.

A distributed visual attention approach on a humanoid robot is proposed in [189]. In this system five different modalities including color, intensity, edges, stereo, and motion are used. The attention processing is distributed on a computer cluster which contains eight PCs. Four run Windows 2000, three Windows XP and one Linux. Five of the PCs are equipped with 2×2.2 GHz Intel Xeon processors, two with 2×2.8 GHz Intel Xeon processors, and one with 2 Opteron 250 processors. All of the computers are connected to a single switch via a Gigabit Ethernet. A frequency of 30 fps with input images of 320×240

pixels is achieved.

A GPU-based saliency map for high-fidelity selective rendering is proposed [109]. This implementation is also based on the saliency map model proposed in [84]. In this implementation a motion map and a depth map as well as habituation are also integrated. However, they use a Sobel filter instead of the complex Gabor-filter to produce the orientation maps. No iterative normalization is computed. For an input image at a resolution of 512×512 pixels the saliency map generation takes about 34 ms using NVIDIA 6600GT graphics card. No CUDA technology is used.

Another high-performance visual attention system handling invariants in the optical array is proposed in [120]. Computation time of 21.8 ms for a saliency map computation at VGA resolution is achieved using GF 8800 GTX at precision of 32 bit using OpenGL 2.0.

The most comparable implementation to this implementation is [148] of iLab, USC, because it also uses the same parameter values as those set in [84] [196]. For a 640×480 color input image, running in a single-threaded on a GNU/Linux system (Fedora Core 6) with a 2.8 GHz Intel Xeon processor, the CPU time required to generate a saliency map is 51.34 ms at a precision of floating-point arithmetic and 40.28 ms at a precision of integer arithmetic. Computed on a cluster of 48 CPUs a 1.5-2 times better result is achieved.

B.2 Graphics Processing Units

In the last few years, programmable GPUs have become more and more popular. GPUs are specialized for compute-intensive, highly parallel computation. Moreover, Compute Unified Device Architecture (CUDA), a new hardware and software architecture issued by NVIDIA in 2007, allows the issuing and managing computations on the GPU as a data-parallel computing device without the need for mapping them in a graphics API [2]. CUDA software development kit includes a standard C compiler, hardware debugger tools, and a performance profiler for simplified application development. It is the only C-language development environment for GPUs. A wide range of applications can be accelerated by using GPUs, such as matrix multiplication, optical flow computation, and so on. More information can be found at www.nvidia.com.

The saliency map computation consists of compute-intensive filtering in different scales, which is highly parallelizable. For real-time application the computation of saliency map is implemented on GeForce 8800 (GTX) graphics cards of NVIDIA, which support the CUDA technology. Here, GeForce 8800 cards are taken as an example. The other new products, which are compatible with the CUDA technology, can also be used. The GeForce 8800 (GTX) consists of 16 multi-processors which consist of 8 processors each. All the processors in the same multi-processor always execute the same instruction, but with different data. This concept enables a highly-gradely parallel computation of a large amount of similar data. The GeForce 8800 (GTX) has a core clock frequency of 575 MHz and a 768 MB memory. The multi-GPU performance is strongly dependent on an efficient usage of the thread-block concept and different memories.

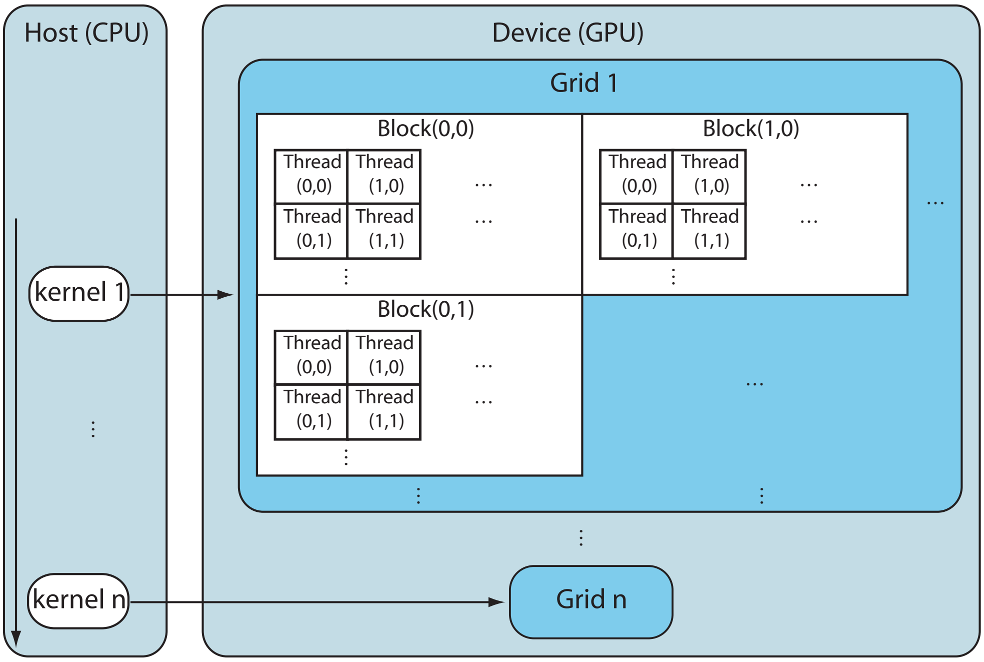

Fig. B.1: The GPU thread batching model.

Thread batching

Programming with CUDA, a GPU is called *compute device*. It contains a large amount of threads which can execute an instruction set on the device with different data in parallel. A function which is compiled to those instruction sets is called *kernel*. In comparison with the GPU, the main CPU is called *host*. The goal is to execute the data-parallel and compute-intensive portions of applications on the GPU instead of on the CPU.

Fig. B.1 shows the thread batching model of a GPU. For each *kernel* function the GPU is configured with a number of threads and blocks. The respective grid of a *kernel* consists of two dimensional blocks. Each block contains up to 512 threads. The input data are divided into the threads. All the threads in a grid execute the same kernel functions. With the thread index *threadIdx* and the block index *blockIdx*, it is configured which data will be processed in which thread. With this structure an easy programming and a good scalability are realized.

Memory

The memory access is also a focus for an efficient programming on GPU. There are six different memories in GPUs:

- Read-write per-thread registers
- Read-write per-thread local memory
- Read-write per-block shared memory
- Read-write per-grid global memory

- Read-only per-grid constant memory
- Read-only per-grid texture memory

Above all, the shared memory and the texture memory are cached, while the read or write access in the non-cached global memory always takes 400-600 clock cycles. Only the texture memory and the global memory can be used for a large amount of data. Moreover, the texture memory is optimized for 2D spatial locality and supports many operations such as interpolation, clamping, data type conversion, etc. However, the texture memory is read-only. The results must be saved in the global memory, which requires data copy between memories.

B.3 Multi-GPU Implementation Details

In Fig. B.2 a data flow diagram of the GPU implementation is illustrated. After an initialization, an input image is firstly converted into 32-bit floating point such that high accuracy and high efficiency will be achieved in the following computation phases. The dyadic Gaussian pyramids are created in the shared memory together with the generation of the intensity maps (I-maps), the opponent red-green (RG-maps), and blue-yellow maps (BY-maps). A Gabor-filter is used to calculate the orientation-maps (O-maps). The Gabor-filter kernel is firstly calculated in the CPU. To save computational cost, the convolution of the sub-sampled images with the Gabor-filter in the space domain is displaced by the multiplication in the frequency domain using Fast Fourier Transform (FFT). Here a Cuda-image is constructed which contains all the images to be filtered by the transformed Gabor-filter such that only one FFT and eight IFFT are needed for the convolution. The images should be assembled before the transformation and disassembled after the transformation in the texture memory. After that, 9 I-maps, 18 C-maps and 36 O-maps are generated.

Furthermore, to ease the center-surround differences and the cross-scale combinations, the available maps at different scales are rescaled into the same size. A point-to-point subtraction followed by an iterative normalization is calculated. On the resulting feature maps (FMs), a point-to-point addition and its following normalization are executed. One conspicuity map (CM) in each channel is obtained. At the end, a summation of the conspicuity maps into the saliency map is completed. A detailed description is provided below.

Initialization

Firstly, the GPU should be initialized. For the reason that the memory allocation in GPUs takes a very long time, the memory is firstly allocated for different images such as the input images, the images in the dyadic Gaussian pyramids, the FMs, the CMs, and the rescaled FMs/CMs at the same size as well as the saliency map.

Since the filter kernel will not be changed during the saliency map computation, the Gabor-filter is calculated in the initialization phase in the CPU and then transformed into

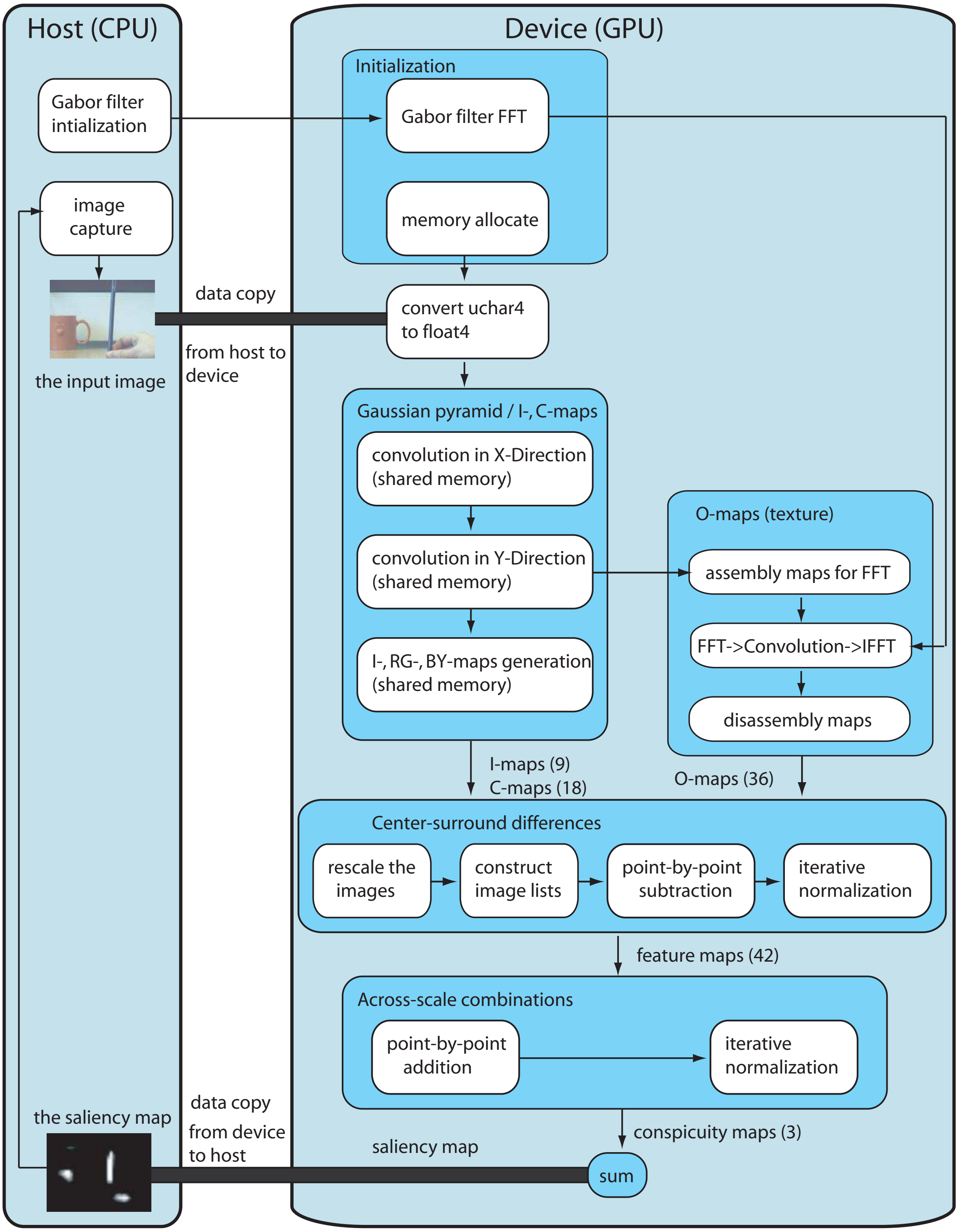

Fig. B.2: Data flow diagram for GPU implementation of saliency map computation

the frequency domain. The implementation of the Gabor-filter and the FFT transformation of the Gabor-filter will be described in Section B.3 in detail.

Data Type Conversion

Input images of 640 × 480 pixels and three 8-bit channels, namely red, green, and blue, are taken as an example. The image data are copied from the CPU into the global memory of the GPU. Since the global memory is not cached, it is essential to follow the right access pattern to get maximum memory bandwidth. The data type must be such that `sizeof`(*type*) is equal to 4, 8, or 16 and the variables of type *type* must be aligned to `sizeof`(*type*) bytes [2]. If the alignment requirement is not fulfilled, the access to the device memories is very costly. The image width fulfills the alignment requirement, while the data amount of each pixel is $3 \times 8 = 24\,\text{bit}$, which does not fulfill the alignment requirement. Therefore, the pixel width should be extended with *padding* and insert an extra 8-bit channel (see Fig. B.3).

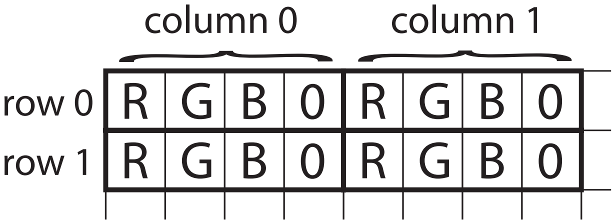

Fig. B.3: Image data padding

After the *padding* the image data type is converted from *uchar4* into *float4* to achieve a high precision for the following computation. The texture memory provides an implicit possibility to do type conversions by means of 2D-texture. Firstly, the input image is bound with a 2D-texture. Then, the kernel function reads each pixel with the function `tex2D`(·) as normalized (between 0.0 and 1.0) *float4* data from the texture and saves them into the global memory.

Dyadic Gaussian Pyramid Computation

In [196] a 6 × 6 separable Gaussian kernel [1 5 10 10 5 1]/32 is used for the image size reduction. A two-dimensional convolution contains $6 \times 6 = 36$ multiplications for each output pixel, while a convolution with separable filters only requires $6 + 6 = 12$ multiplications for each output pixel. Therefore, the dyadic Gaussian pyramid computation is separated into two convolutions: one convolution in the horizontal direction to reduce the horizontal dimension, and one convolution in the vertical direction.

Since each access in the uncached global memory takes 400-600 clock cycles, it is necessary to compute the convolutions in the faster texture memory or shared memory. Bounding the images to a texture requires the data copy between the global memory and the texture memory. Moreover, the data are only readable by kernels through texture fetching. It is more costly than loading the data into the shared memory and computing the convolutions there. Therefore, the convolution is computed in the shared memory.

The convolution in the horizontal direction works, for instance, as follows:

- Specify the thread and block number for the kernel function

- Load the data of an image row from the global memory into the shared memory
- Synchronize the threads to make sure that all the pixels are loaded in the shared memory
- Each thread computes the convolution in the same way
- Copy the result from the shared memory into the global memory

For the convolutions in the horizontal direction, the thread and block number are so specified that a block consists of as many threads as the number of the output image columns and a grid has as many blocks as the number of the output image rows. For example, for the subsampling from an input image at 640 × 480 into an output image at 320 × 480, each block has 320 threads, while each grid has 480 blocks. Each thread computes only one pixel in the output image.

Attention must be paid to the threads' synchronization, because the convolution in the thread n is dependent on the pixels loaded by thread $n - 1$ and $n + 1$.

To deal with the convolutions on the image borders, [10 10 5 1]/26 is used on the left border and [1 5 10 10]/26 on the right border (see Fig. B.4).

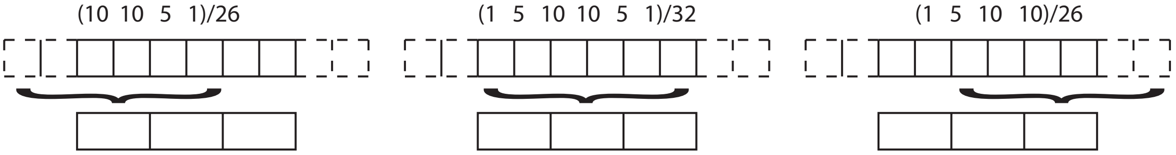

Fig. B.4: The convolution in the horizontal direction

After that, a following subsampling in the vertical direction can be similarly solved. The input image at 640 × 480 (scale $\sigma = 0$) is subsampled into 8 other scales: 320 × 240 ($\sigma = 1$), 160 × 120 ($\sigma = 2$), ..., 2 × 1 ($\sigma = 8$).

C-maps and I-maps Computation

In the saliency map computation I- and C-maps including RG- and BY-maps are required. According to [196], the I-maps are computed as follows:

$$M_I(\sigma) = \frac{r + g + b}{3}, \tag{B.1}$$

where r, g, b are the pixel values in the red, green, and blue channels of the input image.

The opponent C-maps are computed as follows:

$$M_{RG}(\sigma) = \frac{r - g}{\max(r, g, b)}, \tag{B.2}$$

$$M_{BY}(\sigma) = \frac{b - \min(r, g)}{\max(r, g, b)}. \tag{B.3}$$

The regions in which $\max(r, g, b) < 0.1$ are set to 0.

To make the computation more efficient, the computation of the I-maps and the C-maps is integrated into the Gaussian filter convolutions in the vertical direction. Thus, the time for loading the data from the global memory can be spared, since the image data are already in the shared memory after the convolutions.

O-maps Computation

Gabor-filter

To compute the O-maps in different scales, a Gabor-filter G truncated to 19×19 pixels is used [196], which is formulated as follows:

$$G_{\psi}(x, y, \theta) = exp\left(\frac{x'^2 + \gamma^2\, y'^2}{2\epsilon^2}\right) \cdot, cos\left(2\pi\frac{x'}{\lambda} + \psi\right), \tag{B.4}$$

with

$$x' = x\, cos(\theta) + y\, sin(\theta), \quad y' = -x\, sin(\theta) + y\, cos(\theta), \tag{B.5}$$

where (x, y) is the pixel coordinate. The parameter values of this implementation are set according to [196], where γ stands for the aspect ratio with the value 1, while λ is the wavelength and has the value of 7 pixels. The standard deviation ϵ is equal to 7/3 pixels, and $\psi \in \{0, \frac{\pi}{2}\}$. Here, θ stands for the orientation angles with $\theta \in \{0°, 45°, 90°, 135°\}$.

As defined in Eq. B.4, the Gabor-filter consists of a combination of a 2D Gaussian bell-shaped curve and a sine ($\psi = \pi/2$) and cosine function ($\psi = 0$). In each direction, the image should be filtered twice and summed as follows:

$$M_{\theta}(\sigma) = |M_I(\sigma) * G_0(\theta)| + |M_I(\sigma) * G_{\pi/2}(\theta)|, \tag{B.6}$$

with $M_I(\sigma)$ the I-maps at scale σ.

FFT and IFFT

Since a convolution with the 19×19 Gabor-filter is too costly, FFT and IFFT are used to accelerate this process significantly. The Gabor-filter and the images to be convoluted should be first converted into the frequency domain using FFT, and multiplied with each other. Then, the result is converted from the frequency domain into the space domain using IFFT. In doing this, the complexity sinks from $O(n^4)$ (2D convolution) to $O(n^2 \log n)$ (2D FFT).

As mentioned in B.3, the FFT of the Gabor-filter should be computed in the initialization, because it will never be modified in the saliency map generation. Using the CUFFT library [1], eight FFTs with four different orientations and two different forms (sine and cosine) are computed from the original Gabor-filter.

Due to the fact that the input image (640×480) and the subsampled image at scale 1 (320×240) are not used for the following saliency map computation, $7 \times 4 \times 2 = 56$ convolutions for the O-maps are needed (7 scales, 4 orientations and 2 forms). The images in 7 scales are assembled together into a Cuda-image (see Fig. B.5, left) such that just one FFT and eight IFFTs instead of seven FFTs and 56 IFFTs are computed. For an input image at 640 × 480, an image with 256 × 256 is big enough to assemble all the images into itself.

Using the texture a modus named "clamp-to-border" is supported, which makes the image copy very simple. If a pixel outside the texture border is accessed, this pixel has the same color as the border. Therefore, instead of copying the pixel from $(0,0)$ to $(n-1, n-1)$, the pixel is copied from $(-9,-9)$ to $(n+8, n+8)$ of an image with $n \times n$ pixels. In doing this, the border extension for the convolutions is obtained.

Before the FFT of the Gabor-filter is computed, the Gabor-filter kernel (19 × 19) should be rescaled into the same size as that of the image to be convoluted (256 × 256), because the convolution using FFT only can be applied on the input data of the same size [3]. The expansion of the Gabor-filter kernel to the image size should be executed as shown in Fig. B.5 right: cyclically shift the original filter kernel such that the kernel center is at $(0,0)$.

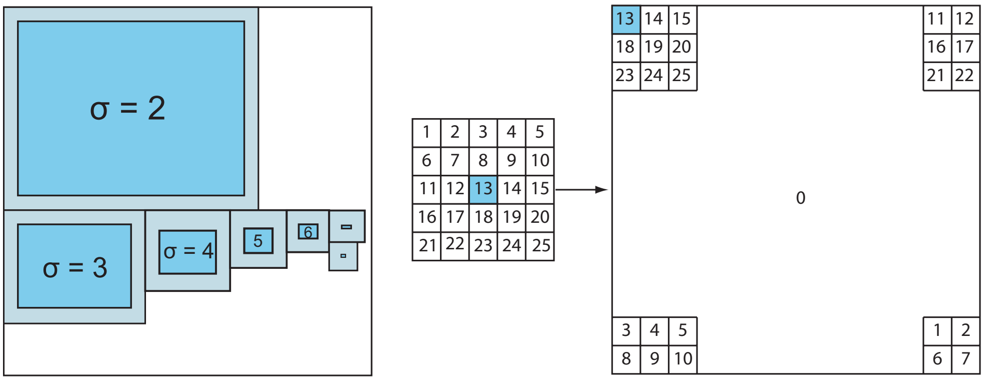

Fig. B.5: The Cuda-image of 256 × 256 pixels (left) and the filter kernel (right) prepared for the FFT

Center-Surround Differences

After the steps above, 9 I-maps, 18 C-maps and 36 O-maps are generated. This is followed by the generation of the center-surround differences and cross-scale combinations. In these two steps, images at different scales are subtracted and combined.

In the center-surround differences, 6 feature maps in the intensity channel, 12 feature maps in the color channel and 24 feature maps in the orientation channel are computed as follows:

$$I(c,s) = |I(c) \ominus I(s)|, \tag{B.7}$$

$$RG(c,s) = |(R(c) - G(c)) \ominus (R(s) - G(s))|, \tag{B.8}$$

$$BY(c,s) = |(B(c) - Y(c)) \ominus (Y(s) - B(s))|, \tag{B.9}$$

$$O(c,s,\theta) = |O(c,\theta) \ominus O(s,\theta)|, \tag{B.10}$$

with c referring to the fine scale and s indicating the coarse scale: $c = \{2,3,4\}$; $\delta = \{3,4\}$; $s = c+\delta$. The orientation of the Gabor-filter is denoted by θ. The subtraction between two images at different scales c and s is denoted by $\ominus$. To execute this subtraction, the images should be enlarged or reduced into the same size and then a point-by-point subtraction is undertaken. The images at scale 2 and 3 are rescaled into scale 4 and the images at scales 5, 6, 7, and 8 are enlarged into scale 4. At the end all the images are at scale 4 and have 40×30 pixels.

For those enlargements and reductions the texture concept is used again. Firstly, the images are bound to the textures. The advantage of this method is that the images using float-coordinates can be accessed and the GPU can compute one new pixel by the interpolation of the four pixels nearby using `tex2D`$(\cdot)$. The step size is computed by dividing the source image size through the goal image size. For example, if an image of 20×15 pixels is rescaled into 40×30 pixels, the step sizes are 0.5 and 0.5 in the horizontal and vertical directions. For the thread and block configuration, 40 threads per block and 7×30 blocks per grid are used, so that the 7 images from I-, C- and O-channels at the same scale are rescaled concurrently, which provides a speed-up for the computation.

Since the images are rescaled into 40×30 pixels at this step, three lists are constructed to make the computation as parallel as possible. Fig. B.6 shows the configuration of the lists. Each list contains $6 \times 7 = 42$ images with different scale numbers (but in the same size 40×30) and channels. The threads and blocks are so parametrized that 42 blocks are configured. Each block is responsible for one image in the list. 42 images are processed in only one kernel function in parallel. This list-concept is also used for the iterative normalization and the cross-scale combinations.

	I-maps							O-maps					
list center	2	2	3	3	4	4	...	2	2	3	3	4	4
list surround	5	6	6	7	7	8	...	5	6	6	7	7	8
list difference	2-5	2-6	3-6	3-7	4-7	4-8	...	2-5	2-6	3-6	3-7	4-7	4-8

Fig. B.6: The image lists configuration

Iterative Normalization

Iterative normalization $N(\cdot)$ is an important component in the whole computation. It simulates local competition between neighboring salient locations [84]. Each iteration contains self-excitation and neighbor-induced inhibition, which can be implemented using a difference-of-Gaussian (DoG) filter [82]:

$$DoG(x,y) = \frac{c_{ex}^2}{2\pi\sigma_{ex}^2} e^{-\frac{x^2+y^2}{2\sigma_{ex}^2}} - \frac{c_{inh}^2}{2\pi\sigma_{inh}^2} e^{-\frac{x^2+y^2}{2\sigma_{inh}^2}}, \tag{B.11}$$

with $\sigma_{ex} = 2\%$ and $\sigma_{inh} = 25\%$ of the input image width, $c_{ex} = 0.5$, $c_{inh} = 1.5$ and the constant inhibitory term $C_{inh} = 0.02$. At each iteration the given image M is computed as follows [82]:

$$M \leftarrow |M + M * DoG - C_{inh}|_{\geq 0}. \tag{B.12}$$

The inseparable DoG filter is divided into two separable convolution filters, one Gaussian filter for excitation of 5 × 5 pixels and one Gaussian filter for inhibition of 29 × 29 pixels for an input image of 40 × 30 pixels. The larger the input image is, the bigger the filter kernels are. The kernel size is computed as follows:

$$size_{(ex|inh)} = 2 \cdot floor\left(\sigma_{(ex|inh)} \cdot \sqrt{-2 \cdot ln(1/100)}\right) + 1. \tag{B.13}$$

Although the shared memory size is limited, the images at 40 × 30 and the respective filter kernels (4916 Byte) can fit into it. In doing this, a 10 times acceleration is obtained, where the lists mentioned in B.3 are also used.

Combination into the Saliency Map

In the following across-scale combinations, no image rescaling is needed. It is only a question of point-by-point integration of the feature maps into conspicuity maps $\bar{I}$, $\bar{C}$, and $\bar{O}$ as follows [84]:

$$\bar{I} = \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} N(I(c,s)), \tag{B.14}$$

$$\bar{C} = \bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} [N(RG(c,s)) + N(BY(c,s))], \tag{B.15}$$

$$\bar{O} = \sum_{\theta \in \{0°,45°,90°,135°\}} N(\bigoplus_{c=2}^{4} \bigoplus_{s=c+3}^{c+4} N(O(c,s,\theta))). \tag{B.16}$$

The final saliency map is a linear combination of the normalized conspicuity maps.

$$S = \frac{1}{3}(N(\bar{I}) + N(\bar{C}) + N(\bar{O})). \tag{B.17}$$

The weighting possibility of the feature maps and the conspicuity maps is also implemented such that top-down visual attention selection can be easily integrated later (see Section 4.2).

Multi-GPU Utilization

A parallel utilization of multi-GPU is one of the highlights of this implementation, through which a significant acceleration of the saliency map computation is achieved.

One of the parallelization possibilities is the pipeline structure (see Fig. B.7). The whole computation is divided into several parts with similar computational load. Each GPU computes only one part of the whole computation. The first GPU reads the input images. After the former GPU finishes its job, the data will be transfered to the next GPU. New data is then loaded into the former GPU. The last GPU provides the final result. The advantage of the pipeline structure is that all the partial works are executed in parallel, such that the frame rate can be increased. However, in this case, it is difficult to divide the saliency map computation into exactly equal parts. Furthermore, if the GPU number is changed, the computation must be redivided. Finally, the data copy from one GPU into another GPU is also costly.

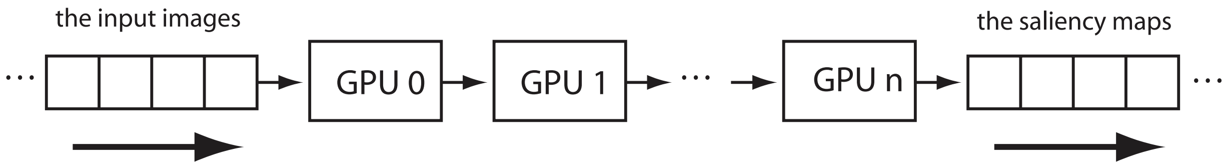

Fig. B.7: The pipeline structure

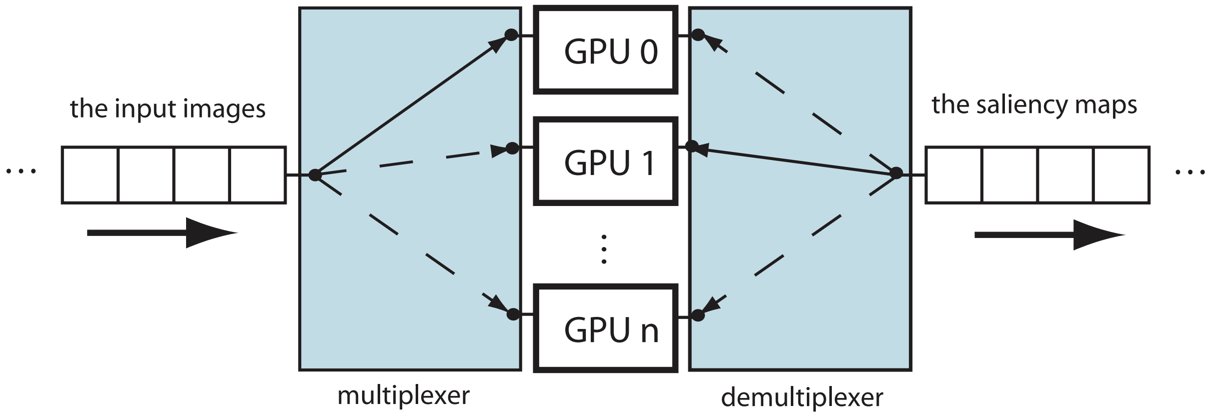

Fig. B.8: The multiplexer and demultiplexer structure

Because of these disadvantages of the pipeline structure, a multiplexer structure is used in this implementation. Fig. B.8 shows how a multiplexer works. The input images are transferred by the multiplexer, also a 1-to-n switch, to different GPUs. Each GPU accomplishes the whole saliency map computation. A following demultiplexer, also an n-to-1 switch, decides from which GPU a saliency map should be taken. It must be pointed out that the multiplexer and the demultiplexer can not work synchronously. Otherwise, the GPU which just starts the computation would provide a wrong result. This multiplexer/demultiplexer structure provides a good scalability on multi-GPU platforms.

To avoid the intricateness of a multi-process mode, a multi-threaded mode is used to manage the multi-GPU utilization. In a multi-threaded mode, in addition to a main thread, several other threads are also utilized. Each thread is responsible for one GPU (see Fig. B.9). In CUDA it is designed that each thread has its own 32-bit address, such that most CUDA functions are encapsulated and do not influence each other. Two exceptions are the modules and the texture references. Since texture reference is defined globally, it is possible that all the threads access the texture reference concurrently. Here, mutual exclusion (mutex) is used to solve this problem. The functions which use a texture reference are blocked by a mutex. In this case there is no significant performance loss, because it is impossible that more than one thread accesses the same texture reference concurrently due to the processing delay.

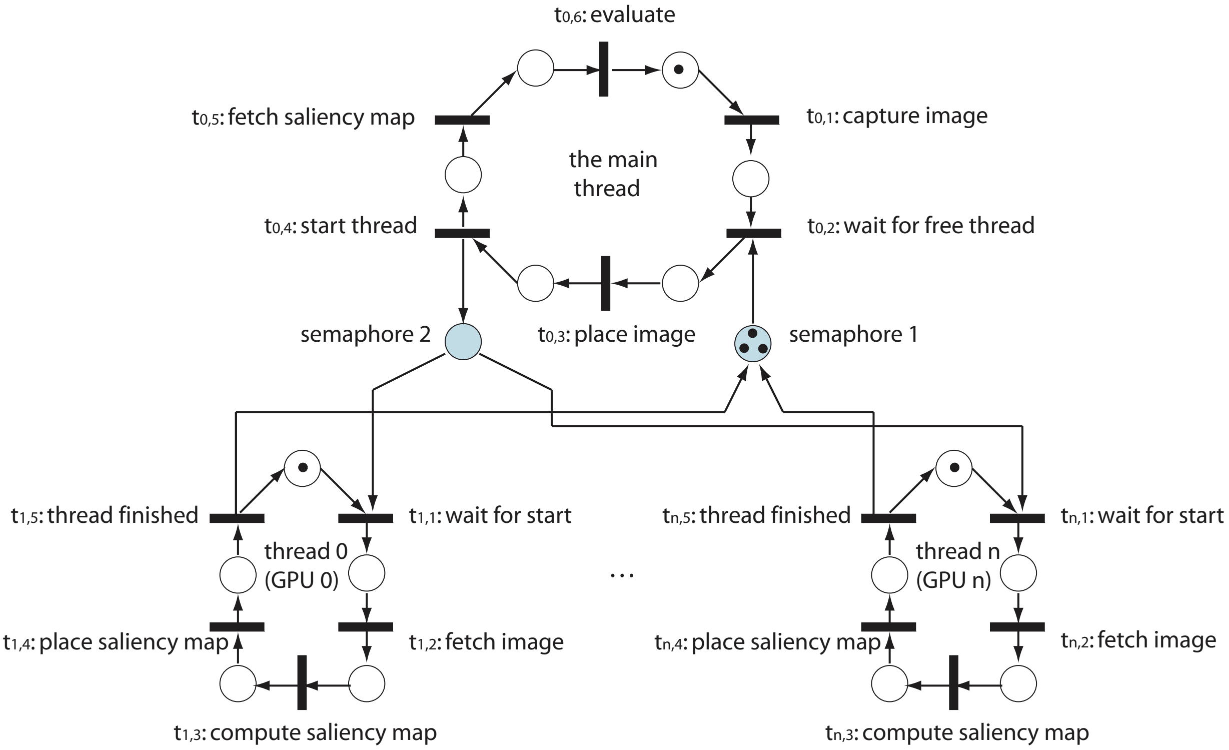

Fig. B.9: The petri-net structure for the multi-threaded mode

Fig. B.9 illustrates the multi-threaded mode in a petri-net. Two semaphores are used to ensure the synchronization of the threads. Semaphore 1 sends a signal to the main thread if one or more GPUs are idle, and is initialized with the number of the applied GPUs. Semaphore 2 starts one of the GPU threads. It is worth mentioning that the transitions $t_{1,1}$, ..., $t_{n,1}$ stand in a deliberate branch conflict, which means it is not possible to determine which transition will be connected. It means in practice that the linux-scheduler decides which thread will be started. It simplifies the implementation and also takes care of an equal utilization of all the threads.

Interestingly, in the main thread, at $t_{0,4}$ a thread is started, while at $t_{0,5}$ a saliency map is ready to be taken. Using this multi-threaded mode the frame rate is significantly increased.

B.4 Results and Discussion

The multi-GPU implementation was tested using 1 to 4 NIVDIA GeForce 8800 (GTX) graphics cards. The computers are equipped with different CPUs and 64-bit linux systems. The computational time is the average processing time of an image over the time processing 1000 input images at a resolution of 640 × 480 pixels.

Tab. B.1 shows the detailed processing time protocol. The most costly step is the initialization, which has a computational time of 328 ms. The memory allocation happens only once and needs almost 50MB RAM. The saliency map computation takes only about 10.6 ms at a frame rate of 94.3 fps. In the GFLOPS performance estimation, only the

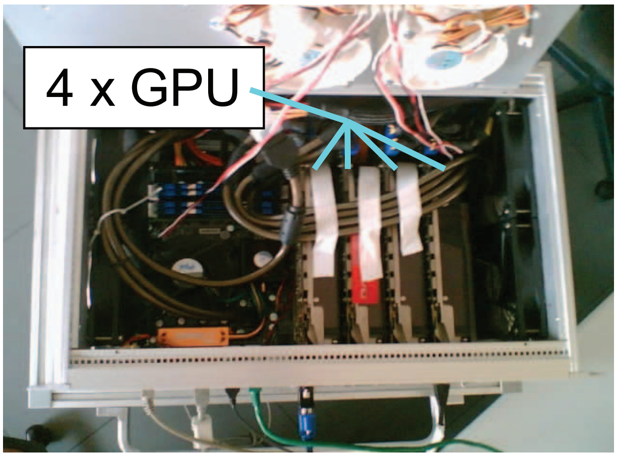

Fig. B.10: Four GPUs installed in a PC.

Saliency map computation	Time [ms]	FLOP	GFLOPS
Initialization	328		
Gaussian pyramid/I-, C-maps	2,10	6.482.049	3,09
FFT, convolution, IFFT	2,39	27.867.923	11,66
Image rescaling	0,89	294.000	0,33
Center-surround differences	0,16	151.200	0.95
Iterative normalization	4,74	34.876.690	7,36
Integration into saliency maps	0,33	62.390	0,19
Total	10,61	69.734.252	6,57

Tab. B.1: Computational time registration using 1 GPU.

floating-point operations are considered. The address-pointer arithmetic, the starting of the CUDA functions, and the memory copy/accesses, which are very time-consuming and have therefore a strong influence on the computational time, are not considered.

Fig. B.11 illustrates the computational time using 1 to 4 GPUs. Using 1 GPU, the saliency map generation takes 10.61 ms, while using 4 GPUs it takes approximately 3.3 times less than that, namely 3.196 ms. This shows a very good scalability of the multi-GPU implementation. The computation performance of GPUs is almost independent of the CPU.

The implementation is also evaluated using a high-speed camera (Dragonfly Express of Point Grey Research Inc., IEEE-1394b, 640 × 480 pixels at 200 Hz). A frame rate of 77 fps is achieved using 1 GPU, while using 2 GPUs a frame rate of 134 fps is obtained. Only 2 ms extra computational time for the saliency map generation in addition to the camera capturing time is required.

In Tab. B.2, the performance of the iLab's implementation and this implementation is compared. Working on the images with the same resolution and the same precision, iLab uses the 2.8 GHz Intel Xeon processor and achieves a frequency of 19.48 Hz, while using this implementation a frequency of 313 Hz is obtained. Using the multi-threaded mode, the maximum speed of iLab's implementation is about 37 fps, which is still about 8.5 times slower than this implementation.

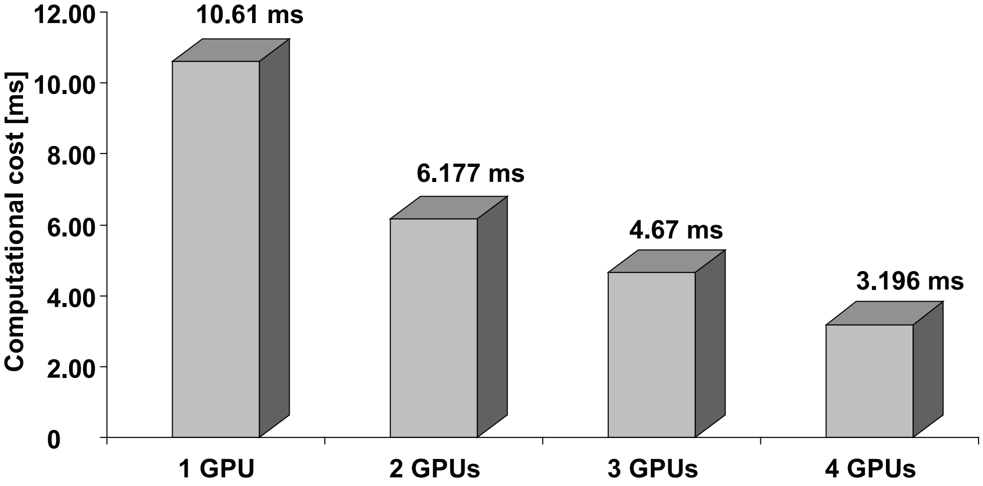

Fig. B.11: Comparison of computation times using 1 to 4 GPUs.

	iLab's implementation	This implementation
Resolution	640 × 480 pixels	640 × 480 pixels
Hardware	2.8 GHz Intel Xeon processor	4 GPUs NVIDIA GeForce 8800 (GTX)
Precision	floating-point	floating-point
Computational time	51.34 ms	3.196 ms
Frequency	19.48 Hz	313 Hz

Tab. B.2: Comparison between iLab's implementation [148] and this implementation.

However, the disadvantage of multi-GPU utilization on mobile robots is the power demand. To power a single GeForce 8800 GTX card, a 450-Watt power supply is needed. To solve this problem, ACE is equipped with eight high power Polymer Li-Ion Modules with 51.8 V and 21 Ah. Thereby, the power demand is totally fulfilled. Without critical real-time conditions, slower GPUs with lower power demand can also be used, such as GeForce 8600 cards.

Bibliography

[1] *CUDA CUFFT Library, NVIDIA, 2007.*

[2] *CUDA Programming Guide Version 1.1, NVIDIA, 2007.*

[3] *FFT-based 2D convolution, NVIDIA, 2007.*

[4] *Player/Stage/Gazebo simulation environment. http://playerstage.sourceforge.net/.*

[5] *Point Grey Research Inc. www.ptgrey.com.*

[6] M. Achtelik, T. Zhang, K. Kühnlenz, and M. Buss. Visual tracking and control of a quadcopter using a stereo camera system and inertial sensors. In *Proceedings of the 2009 IEEE International Conference on Mechatronics and Automation (ICMA), Changchun, China*, pages 2863–2869, 2009.

[7] G. Antonelli, F. Arrichiello, S. Chakraborti, and S. Chiaverini. Experiences of formation control of multi-robot systems with the null-space-based behavioral control. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA), Roma, Italy*, pages 1068–1073, 2007.

[8] T. Arbel and F. P. Ferrie. Viewpoint selection by navigation through entropy maps. In *Proceedings of the Seventh IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 248–254, 1999.

[9] J. C. Baccon, L. Hafemeister, and P. Gaussier. A context and task dependent visual attention system to control a mobile robot. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Lausanne, Switzerland*, pages 238–243, 2002.

[10] G. Backer and B. Mertsching. Two selection stages provide efficient object-based attentional control for dynamic vision. In *Proceedings of Internatonal Workshop on Attention and Performance in Computer Vision*, 2003.

[11] T. Balch and R. C. Arkin. Behavior-based formation control for multirobot teams. *IEEE Transactions on Robotics and Automation*, 14 (6):926–939, 1998.

[12] S. Baluja and D. A. Pomerleau. Expectation-based selective attention for visual monitoring and control of a robot vehicle. *Robotics and Autonomous Systems*, 22:329–344, 1997.

[13] S. W. Ban and M. Lee. *Scene Reconstruction, Pose Estimation and Tracking*, chapter Biologically Motivated Vergence Control System Based on Stereo Saliency Map Model, pages 513–530. I-Tech, 2007.

[14] S. W. Ban, M. Lee, and H. S. Yang. Face detection using biologically motivated saliency map model. *Neurocomputing*, 56:475–480, 2004.

[15] A. Belardinelli, F. Pirri, and A. Carbone. Motion saliency maps from spatiotemporal filtering. In L. Paletta and J. K. Tsotsos, editors, *Attention in Cognitive Systems: 5th International Workshop on Attention in Cognitive Systems (WAPCV)*, volume 5395/2009 of *Lecture Notes in Artificial Intelligence*, pages 112–123. Springer-Verlag, Berlin Heidelberg, 2009.

[16] N. Bergboer, E. Postma, and J. van den Herik. A context-based model of attention. In *Proceedings of the European Conference on Artificial Intelligence (ECAI), Valencia, Spain*, 2004.

[17] A. Del Bimbo and F. Pernici. Towards on-line saccade planning for high-resolution image sensing. *Pattern Recognition Letters, Special issue on vision for crime detection and prevention*, 27 (15):1826–1834, 2006.

[18] M. Björkman and J. O. Eklundh. Vision in the real world: Attending and recognizing objects. *International Journal of Imaging Systems and Technology*, 16:189–208, 2007.

[19] G. Boccignone. Nonparametric bayesian attentive video analysis. In *Proceedings of the 19th International Conference on Pattern Recognition (ICPR), Tampa, FL, USA*, pages 1–4, 2008.

[20] G. Boccignone, A. Marcelli, P. Napoletano, G. D. Fiore, G. Iacovoni, and S. Morsa. Bayesian integration of face and low-level cues for foveated video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 18 (12):1727–1740, 2008.

[21] E. Borenstein, E. Sharon, and S. Ullman. Combining top-down and bottom-up segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, pages 46 – 46, 2004.

[22] A. Borji, M. N. Ahmadabadi, and B. N. Araabi. Offline learning of top-down object based attention control. In *Workshop on Vision in Action: Efficient strategies for cognitive agents in complex environments, Marseille, France*, 2008.

[23] C. Breazeal and B. Scassellati. A context-dependent attention system for a social robot. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1146–1153, 1999.

[24] M. Brezak, I. Petrovic, and E. Ivanjko. A robust and accurate global vision system for real-time tracking of multiple mobile robots. *Robotics and Autonomous Systems*, 56 (3):213–230, 2008.

[25] N. Bruce and J. K. Tsotsos. *Advances in Neural information processing information*, chapter Saliency Based on Information Maximization, pages 155–162. Cambridge, MA: MIT Press, 2006.

[26] N. D. B. Bruce and J. K. Tsotsos. Saliency, attention, and visual search: An information theoretic approach. *Journal of Vision*, 9 (3): 5:1–24, 2009.

[27] C. Burghart, R. Mikut, R. Stiefelhagen, T. Asfour, H. Holzapfel, P. Steinhaus, and R. Dillmann. A cognitive architecture for a humanoid robot: A first approach. In *Proceedings of the 5th IEEE-RAS International Conference on Humanoid Robots*, pages 357–362, 2005.

[28] M. Buss, A. Peer, T. Schauß, N. Stefanov, U. Unterhinninghofen, S. Behrendt, G. Fäber, J. Leupold, K. Diepold, F. Keyrouz, M. Sarkis, P. Hinterseer, E. Steinbach, B. Farber, and H. Pongrac. Video: Multi-modal multi-user telepresence and teleaction system. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Nice, France*, pages 4137–4138, 2008.

[29] R. Carmi and L. Itti. Visual causes versus correlates of attentional selection in dynamic scenes. *Vision Research*, 46:4333–4345, 2006.

[30] B. Cassin and M. L. Rubin, editors. *Dictionary of Eye Terminology.* Triad Publishing Company, 2006.

[31] P. Cavanagh and G. A. Alvarez. Tracking multiple targets with multifocal attention. *TRENDS in Cognitive Sciences*, 9 (7):349–354, 2005.

[32] X. Chen and G. J. Zelinsky. Real-world visual search is dominated by top-down guidance. *Vision Research*, 46:4118–4113, 2006.

[33] M. M. Chun and Y. Jiang. Top-down attentional guidance based on implicit learning of visual covariation. *Psychological science*, 10 (4):360–365, 1999.

[34] M. M. Chun and J. M. Wolfe. Visual attention. In E. B. Goldstein, editor, *Blackwell Handbook of Perception.* Wiley-Blackwell, 4 edition, 2001.

[35] R. T. Collins, O. Amidi, and T. Kanade. An active camera system for acquiring multi-view video. In *Proceedings of International Conference on Image Processing*, volume 1, pages 520–527, 2002.

[36] C. E. Connor, D. C. Preddie, J. L. Gallant, and D. C. Van Essen. Spatial attention effects in macaque area v4. *The Journal of Neuroscience*, 17(9):3201–3214, 1997.

[37] C. J. Costello, C. P. Diehl, A. Banerjee, and H. Fisher. Scheduling an active camera to observe people. In *Proceedings of the ACM 2nd international workshop on Video surveillance & sensor networks*, pages 39–45, 2004.

[38] N. Courty and E. Marchand. Visual perception based on salient features. In *Proceedings of the IEEE/RSJ Intl. Conference on Intelligent Robotics and Systems (IROS)*, volume 1, pages 1024–1029, 2003.

[39] K. Daniilidis, C. Krauss, M. Hansen, and G. Sommer. Real time tracking of moving objects with an active camera. *Real-Time Imaging, Special Issue on computer vision motion analysis*, 4 (1):3–20, 1998.

[40] A. K. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer, and C. J. Taylor. A vision-based formation control framework. *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION*, 18 (5):813–825, 2002.

[41] A. J. Davison. *Mobile Robot Navigation Using Active Vision.* PhD thesis, Robotics Research Group, Department of Engineering Science, University of Oxford, 1998.

[42] G. Deco and E. T. Rolls. Neurodynamic of biased competition and cooperation for attention: A model with spiking neurons. *Journal of Neurophysiology*, 94:295–313, 2005.

[43] B. A. Draper and A. Lionelle. Evaluation of selective attention under similarity transformations. *Computer Vision and Image Understanding, Special issue: Attention and performance in computer vision*, 100 (1-2):152–171, 2005.

[44] B. R. Duffy. Authropomorphism and robotics. In *The Society for the Study of Artificial Intelligence and the Simulation of Behaviour (AISB), Imperial College, England*, 2002.

[45] J. Duncan and G. W. Humphreys. Visual search and stimulus similarity. *Psychological Review*, 96:433–458, 1989.

[46] A. Edsinger and C. C. Kemp. What can i control? a framework for robot self-discovery. In *Proceedings of the 6th International Workshop on Epigenetic Robotics (EpiRob), Paris, France*, 2006.

[47] S. Ekvall, P. Jensfelt, and D. Kragic. Integrating active mobile robot object recognition and slam in natural environment. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Beijing, China*, pages 5792–5797, 2006.

[48] F. Faber, M. Bennewitz, and S. Behnke. Controlling the gaze direction of a humanoid robot with redundant joints. In *Proceedings of the 17th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), Munich, Germany*, pages 413–418, 2008.

[49] M. Ferraro, G. Boccignone, and T. Caelli. Entropy-based representation of image information. *Pattern Recognition Letter*, 23:1391–1398, 2002.

[50] R. Fierro, A. Das, J. Spletzer, J. Esposito, V. Kumar, J. P. Ostrowski, G. Pappas, C. J. Taylor, Y. Hur, R. Alur, I. Lee, G. Grudic, and B. Southall. A framework and architecture for multi-robot coordination. *International Journal of Robotics Research*, 21 (10-11):977–995, 2002.

[51] P. E. Forssen, D. Meger, K. Lai, S. Helmer, J. J. Little, and D. G. Lowe. Informed visual search: Combining attention and object recognition. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA), Pasadena, USA*, pages 935–942, 2008.

[52] T. Foulsham and G. Underwood. What can saliency models predict about eye movements? spatial and sequential aspects of fixations during encoding and recognition. *Journal of Vision*, 8 (2): 6:1–17, 2008.

[53] S. Frintrop. *VOCUS: A Visual Attention System for Object Detection and Goal-directed Search.* PhD thesis, Institute of computer science, Rheinische Friedrich-Wilhelms-Universität Bonn, 2005.

[54] S. Frintrop and P. Jensfelt. Active gaze control for attentional visual slam. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA), Pasadena, USA*, pages 3690–3697, 2008.

[55] S. Frintrop and M. Kessel. Most salient region tracking. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan*, pages 1869–1874, 2009.

[56] S. Frintrop, A. Nüchter, H. Surmann, and J. Hertzberg. Saliency-based object recognition in 3d data. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 2167–2172, 2004.

[57] G. Fritz, C. Seifert, L. Paletta, and H. Bischof. *Attention and Performance in Computational Vision*, volume 3368/2005 of *Lecture Notes in Computer Science*, chapter Attentive Object Detection Using an Information Theoretic Saliency Measure, pages 29–41. Springer Verlag, 2005.

[58] T. Fujita. Hand movement detection using monocular camera for robot cooperation. In *Proceedings of 10th International Conference on Control, Automation, Robotics and Vision (ICARCV), Hanoi, Vietnam*, pages 2188–2191, 2008.

[59] D. Gao, V. Mahadevan, and N. Vasconcelos. On the plausibility of the discriminant center-surround hypothesis for visual saliency. *Journal of Vision*, 8 (7):1–18, 2008.

[60] D. Gao and N. Vasconcelos. *Advances in neural information processing systems 17*, chapter Discriminant saliency for visual recognition from cluttered scenes, pages 481–488. Cambridge, MA: MIT Press, 2005.

[61] S. S. Ge, F. L. Lewis, M. Dekker, C. Jones, and M. J. Mataric. *Autonomous Mobile Robots: Sensing, Control, Decision-Making, and Application*, chapter Behavior-Based Coordination in Multi-Robot Systems, pages 549–572. CRC Press, 2006.

[62] I. Goga and A. Billard. Attention mechanisms for the imitation of goal-directed action in developmental robots. In *Proceedings of the 6th International Conference in Epigenetic Robotics (EPIROB), Paris, France*, 2006.

[63] D. Göhring and H. D. Burkhard. Multi robot object tracking and self localization using visual percept relations. In *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Beijing, China*, pages 31–36, 2006.

[64] H. Goto. Surveillance camera system for mobile computing environments using an active zooming camera and mac address tracking. In *Proceedings of Image and Vision Computing New Zealand (IVCNZ)*, pages 108–113, 2005.

[65] H. S. Greenwald, D. C. Knill, and J. A. Saunders. Integrating visual cues for motor control: A matter of time. *Vision Research*, 45(15):1975–89, 2005.

[66] V. V. Hafner and F. Kaplan. Learning to interpret pointing gestures: experiments with four-legged autonomous robots. *Biomimetic Neural Learning for Intelligent Robots*, 3573:225–234, 2005.

[67] F. H. Hamker. Distributed competition in directed attention. In *Dynamische Perzeption, Workshop der GI-Fachgruppe, Proceedings in Artificial Intelligence*, pages 39–44, 2000.

[68] F. H. Hamker. *Attention and Performance in Computational Vision*, volume 3368/2005, chapter Modeling Attention: From Computational Neuroscience to Computer Vision, pages 118–132. Springer Verlag, 2005.

[69] J. Han, K. N. Ngan, M. Li, and H. Zhang. Unsupervised extraction of visual attention objects in color images. *IEEE Transactions on Circuit and Systems for Video Technology*, 16 (1):141–145, 2006.

[70] J. Hasic and A. Chalmers. Visual attention for significantly influencing the perception of virtual environments. In *Proceedings of the 15th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG)*, 2007.

[71] N. Hawes and J. Wyatt. Towards context-sensitive visual attention. In M. Vincze and L. Paletta, editors, *Proceedings of the Second International Cognitive Vision Workshop (ICVW), Graz, Austria*, 2006.

[72] M. Hayhoe and D. Ballard. Eye movements in natural behavior. *TRENDS in Cognitive Sciences*, 9 (4):188–194, 2005.

[73] M. M. Hayhoe, A. Shrivastava, R. Mruczek, and J. B. Pelz. Visual memory and motor planning in a natural task. *Journal of Vision*, 3:49–63, 2003.

[74] G. Heidemann, R. Rae, H. Bekel, I. Bax, and H. Ritter. *Computer Vision Systems*, volume 2626/2003 of *Lecture Notes in Computer Science*, chapter Integrating Context-Free and Context-Dependent Attentional Mechanisms for Gestural Object Reference, pages 22–33. Springer Verlag, 2003.

[75] C. Hide, T. Moore, and M. Smith. Adaptive kalman filtering for low-cost ins and gps. *The Journal of Navigation*, 56:143–152, 2003.

[76] H. Hunziker. *In the eye of the reader: foveal and peripheral perception.* Transmedia Stäubli Verlag Zürich, 2006.

[77] S. B. Im and S. B. Cho. *Advanced Concepts for Intelligent Vision Systems*, volume 4179/2006 of *Lecture Notes in Computer Science*, chapter Context-Based Scene Recognition Using Bayesian Networks with Scale-Invariant Feature Transform, pages 1080–1087. Springer-Verlag Berlin Heidelberg, 2006.

[78] L. Iocchi, D. Nardi, and M. Salerno. *Balancing Reactivity and Social Deliberation in Multi-Agent Systems*, volume 2103/2001 of *Lecture Notes in Computer Science*, chapter Reactivity and Deliberation: A Survey on Multi-Robot Systems, pages 9–32. Springer Berlin / Heidelberg, 2001.

[79] L. Itti and P. Baldi. A principled approach to detecting surprising events in video. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 631–637, 2005.

[80] L. Itti and P. Baldi. Bayesian surprise attracts human attention. *Vision Research*, 49 (10):1295–1306, 2009.

[81] L. Itti, N. Dhavale, and F. Pighin. Realistic avatar eye and head animation using a neurobiological model of visual attention. In *Proceedings of SPIE 48th Annual International Symposium on Optical Science and Technology*, pages 64–78, 2003.

[82] L. Itti and C. Koch. A comparison of feature combination strategies for saliency-based visual attention systems. In *Proceedings of SPIE human vision and electronic imaging IV (HVEI), San Jose, CA, USA*, volume 3644, pages 473–482, 1999.

[83] L. Itti and C. Koch. Computational modelling of visual attention. *Nature Reviews: Neuroscience*, 2:194–203, 2001.

[84] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *Pattern Analysis and Machine Intelligence*, 20:1254–1259, 1998.

[85] M. Jägersand. Saliency maps and attention selection in scale and spatial coordinates: an information theoretic approach. In *Proceedings of International Conference on Computer Vision (ICCV), Cambridge, MA, USA*, pages 195–202, 1995.

[86] W. James. *The principles of psychology*. New York: Holt, 1890.

[87] J. Jovancevic, B. Sullivan, and M. Hayhoe. Control of attention and gaze in complex environments. *Journal of Vision*, 6:1431–1450, 2006.

[88] M. Juza, K. Marik, J. Rojicek, and P. Stluka. 3d template-based single camera multiple object tracking. In O. Chum and V. Franc, editors, *Computer Vision Winter Workshop, Czech Pattern Recognition Society*, 2006.

[89] F. Kaplan and V. V. Hafner. The challenges of joint attention. *Interaction Studies*, 7:2:135–169, 2006.

[90] B. Khadhouri and Y. Demiris. Compound effects of top-down and bottom-up influences on visual attention during action recognition. In *Proceedings of Nineteenth International Joint Conference on Artificial Intelligence (IJCAI), Edinburgh, Scotland*, pages 1458–1463, 2005.

[91] J. M. Kilner, Y. Paulignan, and S. J. Blakemore. An interference effect of observed biological movement on action. *Current Biology*, 13:522–525, 2003.

[92] N. Kim, I. Kim, and H. Kim. Video surveillance using dynamic configuration of mutiple active cameras. In *Proceedings of IEEE International Conference on Image Processing*, pages 1761–1764, 2006.

[93] E. I. Knudsen. Fundamental components of attention. *Annual Review of Neuroscience*, 30:57–78, 2007.

[94] C. Koch and S. Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. *Human Neurobiology*, 4:219–227, 1985.

[95] H. Kozima. Infanoid: A babybot that explorers the social environment. In *Socially Intelligent Agents*, pages 157–164. Springer US, 2002.

[96] K. Kühnlenz. *Aspects of Multi-Focal Vision*. PhD thesis, Institute of Automatic Control Engineering, Technische Universität München, 2007.

[97] K. Kühnlenz, M. Bachmayer, and M. Buss. A multi-focal high-performance vision system. In *Proceedings of the International Conference of Robotics and Automation (ICRA), Orlando, USA*, pages 150–155, 2006.

[98] K. Kühnlenz and M. Buss. Multi-focal vision and gaze control improve navigation performance. *Journal of Humanoids*, 1 (1):33–44, 2008.

[99] M. Land, N. Mennie, and J. Rusted. The roles of vision and eye movements in the control of activities of daily living. *Perception*, 28:1311–1328, 1999.

[100] M. F. Land and M. Hayhoe. In what ways do eye movements contribute to everyday activities? *Vision Research*, 41:3559–3565, 2001.

[101] N. Lavie, J. W. Fockert, and E. Viding. Load theory of selective attention and cognitive control. *Journal of Experimental Psychology*, 133:339–354, 2004.

[102] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2169–2178, 2006.

[103] I. Lee, S. W. Ban, K. Fukushima, and M. Lee. *Artificial Intelligence and Soft Computing (ICAISC)*, volume 4029/2006 of *Lecture Notes in Computer Science*, chapter Selective Motion Analysis Based on Dynamic Visual Saliency Map Model, pages 814–822. Springer-Verlag Berlin Heidelberg, 2006.

[104] B. Leibe, N. Cornelis, K. Cornelis, and L. Van Gool. Dynamic 3d scene analysis from a moving vehicle. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*, pages 1–8, 2007.

[105] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision, Special Issue on Learning for Vision and Vision for Learning*, 77 (1-3):259–289, 2008.

[106] F. F. Li and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 524–531, 2005.

[107] Z. Li. A saliency map in primary visual cortex. *Trends in Cognitive Sciences*, 6:9–16, 2002.

[108] S. P. Liversedge and J. M. Findlay. Saccadic eye movements and cognition. *Trends in Cognitive Sciences*, 4 (1):6–14, 2000.

[109] P. Longhurst, K. Debattista, and A. Chalmers. A gpu based saliency map for high-fidelity selective rendering. In *Proceedings of the 4th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa (AFRIGRAPH)*, pages 21–29, 2006.

[110] O. Lorch, J. F. Seara, K. H. Strobl, U. D. Hanebeck, and G. Schmidt. Perception errors in vision guided walking: Analysis, modeling and filtering. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA), Washington, DC, USA*, pages 2048– 2053, 2002.

[111] E. Lorini and M. Piunti. The benifits of surprise in dynamic environments: From theory to practice. *Affective Computing and Intelligent Interaction, LNCS, Springer Verlag*, 4738:362–373, 2007.

[112] O. Ludwig. *Untersuchungen zur Schrittphasenabhängigkeit von Hindernisvermeidungsreaktionen bem menschlichen Gang.* PhD thesis, Naturwissenschaftlich-Technische Fakultät III der Universität des Saarlandes, Saarbrücken, Germany, 2002.

[113] A. Mack and I. Rock. *Inattentional blindness.* Cambridge, MA: MIT Press, 1998.

[114] W. Maier, E. Mair, D. Burschka, and E. Steinbach. Visual homing and surprise detection for cognitive mobile robots using image-based environment representation. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan*, 2009.

[115] V. Maljkovic and K. Nakayama. Priming of pop-out: I. role of features. *Memory and Cognition*, 22:657–672, 1994.

[116] P. Mamassian, M. Landy, and L. T. Maloney. Bayesian modelling of visual perception. In R. Rao, B. Olshausen, and M. Lewicki, editors, *Probabilistic Models of the Brain.* Cambridge, MA: MIT Press, 2002.

[117] S. Marat, T. Ho Phuoc, L. Granjon, N. Guyader, D. Pellerin, and A. Guerin-Dugue. Modelling spatio-temporal saliency to predict gaze direction for short videos. *International Journal of Computer Vision*, 82 (3):231–243, 2009.

[118] R. Marfil, A. Bandera, J. A. Rodriguez, and F. Sandoval. A novel hierarchical framework for object-based visual attention. *Attention in Cognitive Systems, LNCS 5359/2009*, pages 27–40, 2009.

[119] T. Matsuyama. *KI-99: Advances in Artificial Intelligence*, volume 1701, chapter Cooperative Distributed Vision: Dynamic Integration of Visual Perception, Action, and Communication, pages 75–88. Springer Verlag Berlin Heidelberg, 1999.

[120] S. May, M. Klodt, E. Rome, and R. Breithaupt. Gpu-accelerated affordance cueing based on visual attention. In *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA*, pages 3385–3390, 2007.

[121] P. McGuire, J. Fritsch, J. J. Steil, F. Röthling, G. A. Fink, S. Wachsmuth, G. Sagerer, and H. Ritter. Multi-modal human-machine communication for instructing robot grasping tasks. In *Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Lausanne, Switzerland*, pages 1082–1089, 2002.

[122] E. Menegatti and E. Pagello. Omnidirectional distributed vision for multi-robot mapping. *Distributed Autonomous Robot System*, 5:279–288, 2002.

[123] B. Mertsching, M. Bollmann, A. Massad, and S. Schmalz. Recognition of complex objects with an active vision system. In *Proceedings of International ICSC/IFAC Symposium on Neural Computation (NC)*, 1998.

[124] T. Michalke, R. Kastner, J. Adamy, S. Bone, F. Waibel, M. Kleinhagenbrock, J. Gayko, A. Gepperth, J. Fritsch, and C. Goerick. An attention-based system approach for scene analysis in driver assistence. *at-Automatisierungstechnik*, 56:575–584, 2008.

[125] R. Milanese. *Detecting Salient Region in an Image: from Biological Evidence to Computer Implementation.* PhD thesis, University of Geneva, 1993.

[126] R. Milanese, H. Wechsler, S. Gil, J. M. Bost, and T. Pun. Integration of bottom-up and top-down cues for visual attention using non-linear relaxation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA*, pages 781–785, 1994.

[127] M. Mitani, M. Takaya, A. Kojima, and K. Fukunaga. Environment recognition based on analysis of human actions for mobile robot. In *Proceedings of the 18th International Conference on Pattern Recognition (ICPR)*, volume 4, pages 782–786, 2006.

[128] S. Mitri, S. Frintrop, and A. Nüchter. Robust object detection at regions of interest with an application in ball recognition. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation Barcelona, Spain*, pages 125– 130, 2005.

[129] N. Mitsunaga and M. Asada. Visual attention control for a legged mobile robot based on information criterion. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and System (IROS), EPFL, Switzerland*, pages 244–249, 2002.

[130] M. Mozer, M. Shettel, and S. Vecera. Top-down control of visual attention : a rational account. In *Proceedings of the 9th Annual Conference on Neural Information Processing Systems (NIPS)*, volume 18, pages 923–930, 2005.

[131] H. J. Müller, B. Reimann, and J. Krummenacher. Visual search for singleton feature targets across dimensions: Stimulus and expectancy-driven effects in dimensional weighting. *Journal of Experimental Psychology: Human Perception and Performance*, 29 (5):1021–1035, 2003.

[132] Y. Nagai, M. Asada, and K. Hosoda. Learning for joint attention helped by functional development. *Advanced Robotics*, 20 (10):1165–1181, 2006.

[133] Y. Nakamura. From human behavior to robot intelligenz. Accademia Nazionale Del Lincei Convegno Internazionale – Robotics: A New Science, 2008.

[134] V. Navalpakkam and L. Itti. Modeling the influence of task on attention. *Vision Research*, 45:205–231, 2005.

[135] V. Navalpakkam and L. Itti. An integrated model of top-down and bottom-up attention for optimizing detection speed. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2049–2056, 2006.

[136] H. Nguyen and B. Bhanu. Tracking multiple objects in non-stationary video. In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, pages 1561–1568, 2009.

[137] K. Nishiwaki, S. Kagami, J. J. Kuffner, K. Okada, Y. Kuniyoshi, M. Inaba, and H. Inoue. Online humanoid locomotion controll by using 3d vision information. *Springer Tracts in Advanced Robotics, Experimental Robotics VIII*, 5:85–94, 2003.

[138] N. Ouerhani and H. Hügli. *Computational Methods in Neural Modeling*, volume 2686/2003 of *Lecture Notes in Computer Science*, chapter A Model of Dynamic Visual Attention for Object Tracking in Natural Image Sequences, pages 702–709. Springer Verlag Berlin Heidelberg, 2003.

[139] N. Ouerhani and H. Hügli. Robot self-localization using visual attention. In *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), Espoo, Finland*, pages 309–314, 2005.

[140] N. Ouerhani, H. Hügli, P. Y. Burgi, and P. F. Ruedi. *Pattern Recognition*, chapter A Real Time Implementation of the Saliency-Based Model of Visual Attention on a SIMD Architecture, pages 282–289. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2002.

[141] E. Pagello, A. D'Angelo, and E. Menegatti. Cooperation issues and distributed sensing for multirobot systems. *Proceedings of The IEEE*, 94 (7):1370–1383, 2006.

[142] L. Paletta, G. Fritz, and C. Seifert. Reinforcement learning of informative attention patterns for object recognition. In *Proceedings of the 4-th IEEE International Conference on Development and Learning*, pages 188–193, 2005.

[143] L. Paletta, E. Rome, and H. Buxton. *Neurobiology of attention*, chapter Attention Architectures for Machine Vision and Mobile Robots, pages 642–648. Elsevier Academic Press, 2005.

[144] L. E. Parker. *Current Research in Multirobot Systems*, volume 7 (2-3). Springer Japan, 2003.

[145] L. Paya, O. Reinoso, A. Gil, and J. Sogorb. Multi-robot route following using omnidirectional vision and appearance-based representation of the environment. *Lecture Notes in Computer Science, Hybrid Artificial Intelligence Systems (HAIS)*, LNAI 5271:680–687, 2008.

[146] M. Pellkofer and E.D. Dickmanns. Ems-vision: Gaze control in autonomous vehicles. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Dearborn, USA*, pages 296–301, 2000.

[147] J. Peng, A. Peters, X. Ao, and A. Srikaew. Grasping a waving object for a humanoid robot using a biologically-inspired active vision system. In *Proceedings of the 2003 IEEE International Workshop on Robot and Human Interactive Communication (RO-MAN), Millbrae, USA*, pages 115– 120, 2003.

[148] R. J. Peters and L. Itti. Applying computational tools to predict gaze direction in interactive visual environments. *ACM Transactions on Applied Perception*, 5(2), 2008.

[149] S. Pollmann, R. Weidner, H. J. Müller, and D. Yves von Cramon. Neural correlates of visual dimension weighting. *Visual cognition*, 14 (4-8):877–897, 2006.

[150] Z. W. Pylyshyn and R. W. Storm. Tracking multiple independent targets: evidence for a parallel tracking mechanism. *Spatial Vision*, 3 (3):1–19, 1988.

[151] A. Ranganathan and F. Dellaert. Bayesian surprise and landmark detection. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan*, 2009.

[152] R. P. N. Rao. Bayesian inference and attentional modulation in the visual cortex. *Cognitive Neuroscience and Neuropsychology*, 16 (16):1843–1848, 2005.

[153] B. Rasolzadeh, A. Tavakoli, and J-O. Eklundh. An attentional system combining top-down and bottom-up influences. In *Workshop on Attention and Performance in Computational Vision (WAPCV07), Hyderabad, India*, 2007.

[154] P. Renaud, E. Cervera, and P. Martiner. Towards a reliable vision-based mobile robot formation controll. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sendai, Japan*, volume 4, pages 3176– 3181, 2004.

[155] L. W. Renninger, J. Coughlan, and P. Verghese. An information maximization model of eye movements. *Advances in neural information processing systems*, 17:1121–1128, 2005.

[156] R. A. Rensink. The dynamic representation of scenes. *Visual Cognition*, 7 (1/2/3):17–42, 2000.

[157] R. A. Rensink. Visual sensing without seeing. *Psychological Science*, 15 (1):27–32, 2004.

[158] Reuters. Robocup 2009: Kein weltmeistertitel für österreich. *Kleine Zeitung*, 2009.

[159] R. Rocha, J. Dias, and A. Carvalho. Cooperative multi-robot systems a study of vision-based 3-d mapping using information theory. In *Proceedings of the 2005 IEEE Internatinal Conference on Robotics and Automation (ICRA), Barcelona, Spain*, pages 384–389, 2005.

[160] C. A. Rothkopf, D. H. Ballard, and M. M. Hayhoe. Task and context determine where you look. *Journal of Vision*, 7 (14):1–20, 2007.

[161] J. Ruesch, M. Lopes, A. Bernardino, J. Hörnstein, J. Santos-Victor, and R. Pfeifer. Multimodal saliency-based bottom-up attention a framework for the humanoid robot icub. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA), Pasadena, CA, USA*, pages 962–967, 2008.

[162] F. Saidi, O. Stasse, and K. Yokoi. Active visual search by a humanoid robot. *Recent Progress in Robotics: Viable Robotic Service to Human, Lecture Notes in Control and Information Sciences*, 370:171–184, 2009.

[163] S. Schaal and L. Itti. Learning and attention with a humanoid robot head. 2005.

[164] C. Scheier and S. Egner. Visual attention in a mobile robot. In *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE), Guimaraes, Portugal*, volume 1, pages 48–52, 1997.

[165] K. Schill. *Neurobiology of Attention*, chapter A Model of Attention and Recognition by Information Maximization, pages 671–676. Elsevier, 2005.

[166] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–7, 2007.

[167] F. Schubert, T. Spexard, M. Hanheide, and S. Wachsmuth. Active vision-based localization for robots in a home-tour scenario. In *Proceedings of the 5th International Conference on Computer Vision Systems (ICVS), Bielefeld, Germany*, 2007.

[168] J. F. Seara. *Intelligent Gaze Control for Vision-Guided Humanoid Walking*. PhD thesis, Technische Universität Müenchen, Munich, Germany, 2004.

[169] J.F. Seara and G. Schmidt. Gaze control strategy for vision-guided humanoid walking. *at-Automatisierungstechnik*, 2:49–58, 2005.

[170] K. Shinoda, N. H. Bach, S. Furui, and N. Kawai. Scene recognition using hidden markov models for video database. In *Proceedings of Symposium on Large-Scale Knowledge Resourses (LKR), Tokyo, Japan*, pages 107–110, 2005.

[171] C. Siagian and L. Itti. Biologically-inspired face detection: Non-brute-force-search approach. In *Proceddings of International Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, pages 62–62, 2004.

[172] C. Siagian and L. Itti. Gist: A mobile robotics application of context-based vision in outdoor environment. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) – Workshops*, pages 88–88, 2005.

[173] H. A. Simon. *Computers, Communication, and the Public Interest*, chapter Designing Organizations for an Information-Rich World, pages 37–72. The Johns Hopkins Press, 1971.

[174] S. K. Singh, D. S. Chauhan, M. Vasta, and R. Singh. A robust skin color based face detection algorithm. *Tamkang Journal of Science and Engineering*, 6 (4):227–234, 2003.

[175] K. Smith, D. Gatica-Perez, J. Odobez, and B. Sileye. Evaluating multi-object tracking. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 36–36, 2005.

[176] N. Sprague and D. Ballard. Eye movements for reward maximization. *Advances in Neural Information Processing Systems 16: Proceedings of 2003 Neural Information Processing Conference*, 16:1467–1474, 2003.

[177] O. Stasse, Y. Kuniyoshi, and G. Cheng. *Biologically Motivated ComputerVision*, volume 1811/2000 of *Lecture Notes in Computer Science*, chapter Development of a Biologically Inspired Real-Time Visual Attention System, pages 779–785. Springer Berlin/Heidelberg, 2000.

[178] H. Sumioka, Y. Yoshikawa, and M. Asada. Learning of joint attention from detecting causality based on transfer entropy. *Journal of Robotics and Mechatronics*, 20 (3):378–385, 2008.

[179] Y. Sun and R. Fisher. Object based visual attention for computer vision. *Artificial Intelligence*, 146 (1):77–123, 2003.

[180] J. Tani. *Artificial Neural Networks – ICANN'97*, volume 1327/1997 of *Lecture Notes in Computer Science*, chapter Visual Attention and Learning of a Cognitive Robot, pages 697–702. Springer Verlag Berlin Heidelberg, 1997.

[181] B. Telle, O. Stasse, T. Ueshiba, K. Yokoi, and F. Tomita. Three characterisations of 3d reconstruction uncertainty with bounded error. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3894– 3899, 2004.

[182] J. Theeuwes, B. Reimann, and K. Mortier. Visual search for featural singletons: No top-down modulation, only bottom-up priming. *Visual Cognition*, 14 (4-8):466–489, 2006.

[183] A. Torralba, A. Oliva, M. S. Castelhano, and J. M. Henderson. Contextual guidance of eye movements and attention in real-world scenes: The role of global features in object search. *Psychological Review*, 113 (4):766–786, 2006.

[184] A. Torralba and P. Sinha. Statistical context priming for object detection. In *Proceedings of the International Conference on Computer Vision (ICCV), Vancouver, Canada*, volume 1, pages 763–770, 2001.

[185] A. M. Treisman and G. Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12:97–136, 1980.

[186] J. K. Tsotsos. Analyzing vision at the complexity level. *Behavioral and Brain Sciences*, 13 (3):423–445, 1990.

[187] J. K. Tsotsos, S. M. Culhane, W. Y. K. Wai, Y. Lai, N. Davis, and F. Nuflo. Modeling visual attention via selective tuning. *Artificial Intelligence*, 78:507–545, 1995.

[188] J. K. Tsotsos and K. Shubina. Attention and visual search: Active robotic vision systems that search. In *Proceedings of the 5th International Conference on Computer Vision Systems*, 2007.

[189] A. Ude, V. Wyart, L. H. Lin, and G. Cheng. Distributed visual attention on a humanoid robot. In *Proceedings of 2005 5-th IEEE-RAS International Conference on Humanoid Robots*, pages 381–386, 2005.

[190] N. Ukita and T. Matsuyama. Real-time multi-target tracking with cooperative communication among active vision agents. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2*, pages 664–671, 2002.

[191] V. V. Vassili, V. Sazonov, and A. Andreeva. A survey on pixel-based skin color detection techniques. In *Proceedings of International Conference on Computer Graphics*, 2003.

[192] S. Vasudevan, S. Gächter, V. Nguyen, and R. Siegwart. Cognitive maps for mobile robots – an object based approach. *Robotics and Autonomous Systems*, 55:359–371, 2007.

[193] R. Vidal, O. Shakernia, and S. Sastry. Formation control of nonholonomic mobile robots with omnidirectional visual servoing and motion segmentation. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 584–489, 2003.

[194] S. Vijayakumar, J. Conradt, T. Shibata, and S. Schaal. Overt visual attention for a humanoid robot. In *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, volume 4, pages 2332–2337, 2001.

[195] S. Vijayakumar, J. Conradt, T. Shibata, and S. Schaal. Overt visual attention for a humanoid robot. In *Proceedings of 2001 IEEE International Conference on Intelligent Robot and Systems*, volume 4, pages 2332–2337, 2001.

[196] D. Walther and C. Koch. Modeling attention to salient proto-objects. *ScienceDirect. Neural Networks*, 19:1395–1407, 2006.

[197] D. Walther, U. Rutishauser, C. Koch, and P. Perona. Selective visual attention enables learning and recognition of multiple objects in cluttered scenes. *Computer Vision and Image Understanding*, 100:41–63, 2005.

[198] G. Welch and G. Bishop. An introduction to the kalman filter. SIGGRAPH 2001 Course.

[199] H. Wersing, S. Kirstein, M. Götting, H. Brandl, M. Dunn, I. Mikhailova, C. Goerick, J. Steil, H. Ritter, and E. Körner. Online learning of objects and faces in an integrated biologically motivated architecture. In *Proceedings of International Conference on Computer Vision Systems (ICVS), Bielefeld, Germany*, 2007.

[200] N. Winters and J. Santos-Victor. Visual attention-based robot navigation using information sampling. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Maui, Hawai, USA*, volume 3, pages 1670–1675, 2001.

[201] J. M. Wolfe. Guided search 2.0: A revised model of visual search. *Psychonomic Bulletin & Review*, I (2):202–238, 1994.

[202] J. M. Wolfe. Visual search. In H. Pashler, editor, *Attention*. University College London Press, 1998.

[203] W. J. Won, S. W. Ban, and M. Lee. Real time implementation of a selective attention model for the intelligent robot with autonomous mental development. In *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE), Dubrovnik, Croatia*, volume 3, pages 1309– 1314, 2005.

[204] R. D. Wright and L. M. Ward. *Orienting of Attention.* Oxford University Press, 2008.

[205] L. Yang, Z. Cao, C. Zhou, and M. Tan. *Intelligent Robotics and Applications*, volume 5314/2008 of *Lecture Notes in Computer Science*, chapter Observation-Based Multi-robot Cooperative Formation Control, pages 1117–1126. Springer Verlag Berlin Heidelberg, 2008.

[206] Y. Yang and W. Gao. An optimal adaptive kalman filter. *Journal of Geodesy*, 80 (4):177–183, 2006.

[207] T. Yonezawa, H. Yamazoe, A. Utsumi, and S. Abe. Gaze-communicative behavior of stuffed-toy robot with joint attention and eye contact based on ambient gaze-tracking. In *Proceedings of the 9th International Conference on Multimodal Interfaces*, pages 140–145, 2007.

[208] A. Yorita, T. Hashimoto, H. Kobayashi, and N. Kubota. *Progress in Robotics*, volume 44 of *Communications in Computer and Information Science*, chapter Remote Education Based on Robot Edutainment, pages 204–213. Springer Verlag Berlin Heidelberg, 2009.

[209] Y. Yu, G. K. I. Mann, and R. G. Gosine. A task-driven object-based attention model for robots. In *Proceedings of IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1751–1756, 2007.

[210] A. Yuille and D. Kersten. Vision as bayesian inference: analysis by synthesis? *TRENDS in Cognitive Sciences, Special Issue: Probabilistic models of cognition*, 10 (7):301–308, 2006.

[211] A. Zaharescu, A. L. Rothenstein, and J. K. Tsotsos. *Attention and Performance in Computational Vision*, volume 3368/2005 of *Lecture Notes in Computer Science*, chapter Towards a Biologically Plausible Active Vision Search Model, pages 133–147. Springer Verlag Berlin Heidelberg, 2005.

[212] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, and G. W. Cottrell. Sun: A bayesian framework for saliency using natural statistics. *Journal of Vision*, 8 (7): 32:1–20, 2008.

Own Publications

[213] A. Bauer, K. Klasing, G. Lidoris, M. Mühlbauer, F. Rohrmüller, S. Sosnowski, **T. Xu**, K. Kühnlenz, D. Wollherr, and M. Buss. The autonomous city explorer: Towards natural human-robot interaction in urban environments. *International Journal of Social Robotics*, 1(2):127–140, 2009.

[214] A. Bauer, K. Klasing, **T. Xu**, S. Sosnowski, G. Lidoris, Q.Mühlbauer, T. Zhang, F. Rohrmüller, D. Wollherr, K. Kühnlenz, and M. Buss. The autonomous city explorer project. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan*, pages 1595–1596, 2009.

[215] G. Lidoris, K. Klasing, A. Bauer, **T. Xu**, K. Kühnlenz, D. Wollherr, and M. Buss. The autonomous city explorer project: Aims and system overview. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Diego, CA, USA*, pages 560–565, 2007.

[216] Q. Mühlbauer, S. Sosnowski, **T. Xu**, T. Zhang, K. Kühnlenz, and M. Buss. The autonomous city explorer project: Towards navigation by interaction and visual perception. In *Proceedings of the 1st International Workshop on Cognition for Technical Systems, Munich, Germany*, 2008.

[217] Q. Mühlbauer, S. Sosnowski, **T. Xu**, T. Zhang, K. Kühnlenz, and M. Buss. Navigation through urban environments by visual perception and interaction. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA) , Kobe, Japan*, pages 3558–3564, 2009.

[218] Q. Mühlbauer, **T. Xu**, A. Bauer, K. Klasing, G. Lidoris, F. Röhrmüller, S. Sosnowski, K. Kühnlenz, D. Wollherr, and M. Buss. Wenn roboer nach dem weg fragen. *at – Automatisierungstechnik*. under review.

[219] **T. Xu**, K. Kühnlenz, and M. Buss. Information-based gaze control adaptation to scene context for mobile robots. In *Proceedings of the 19th International Conference on Pattern Recognition (ICPR), Tampa, USA*, pages 1–4, 2008.

[220] **T. Xu**, K. Kühnlenz, and M. Buss. A view direction planning strategy for a multi-camera vision system. In *Proceedings of IEEE International Conference on Information and Automation (ICIA), Zhangjiajie, China*, pages 320–325, 2008.

[221] **T. Xu**, K. Kühnlenz, and M. Buss. Autonomous switching of top-down and bottom-up attention selection for vision guided mobile robots. In *Proceedings of 2009 IEEE/RSJ International Conference on Intelligent RObots and Systems (IROS), St. Louis, MO, USA*, pages 4009–4014, 2009.

[222] **T. Xu**, K. Kühnlenz, and M. Buss. Coordinated multi-focal feature tracking for an interactive mobile robot. In *Proceedings of 18th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), Toyama, Japan*, pages 861–866, 2009.

[223] **T. Xu**, K. Kühnlenz, and M. Buss. A multi-camera view direction planning strategy for mobile robots. *International Journal of Information Acquisition (IJIA)*, 5 (4):309–320, 2009.

[224] **T. Xu**, K. Kühnlenz, and M. Buss. Two integrated approaches to top-down and bottom-up controlled active vision of mobile robots. *Transactions on Robotics*, 2009. under review.

[225] **T. Xu**, Q. Mühlbauer, S. Sosnowski, K. Kühnlenz, and M. Buss. Looking at the surprise: Bottom-up attention control of an active camera system. In *Proceedings of the 10th International Conference on Control, Automation, Robotics and Vision (ICARCV), Hanoi, Vietnam*, pages 637–642, 2008.

[226] **T. Xu**, T. Pototschnig, K. Kühnlenz, and M. Buss. A high-speed multi-gpu implementation of bottom-up attention using cuda. In *Proceedings of International Conference on Robotics and Automation (ICRA), Kobe, Japan*, pages 41–47, 2009.

[227] **T. Xu**, H. Wu, T. Zhang, K. Kühnlenz, and M. Buss. Environment adapted active multi-focal vision system for object detection. In *Proceedings of International Conference on Robotics and Automation (ICRA), Kobe, Japan*, pages 2418–2423, 2009.

[228] **T. Xu**, T. Zhang, K. Kühnlenz, and M. Buss. *Computer Vision*, chapter Towards High-Speed Vision for Attention and Navigation of Autonomous City Explorer (ACE), pages 189–214. IN-Tech, 2008.

[229] **T. Xu**, T. Zhang, K. Kühnlenz, and M. Buss. Attentional object detection with an active multi-focal vision system. *International Journal of Humanoid Robotics (IJHR)*, 7(2):223–243, 2010.