Combining airborne images and open data to retrieve knowledge of construction sites

Katrin Jahr[1], Dorota Iwaszczuk[2], André Borrmann[1]
[1]Technical University of Munich, Germany
[2]Technical University of Darmstadt, Germany
katrin.jahr@tum.de

Abstract. Construction site planning is based on both explicit knowledge, as retrieved from regulations, and implicit knowledge, arising from experience. To retrieve and formalize rules from implicit knowledge, past construction projects can be analyzed. In this paper, we present an image analysis pipeline to retrieve information on past construction sites from airborne images. We fuse machine learning based image analysis with georeferencing and openly available geospatial data to retrieve a detailed description with true dimensions of the construction site at hand.

1. Introduction

The digitization of the AEC (Architecture, engineering, construction) industry offers various new possibilities for designing, planning, and monitoring buildings. In recent years, many research projects focused on using methods of computer-aided engineering, such as building information modelling or structural simulations, to facilitate and enhance the planning process. However, as of now, not many of the advantages of using digital support are carried on after structural designing is concluded. Specifically, construction site layout planning (CSLP) is still mostly realized based on planner's experience and rules of thumb, if at all. Using scientifically proven digital support tools offers significant potential for improvement regarding planning accuracy and time efficiency.

A large set of information, traditionally acquired in late planning stages, must be considered during site equipment (SE) planning. To reduce planning efforts and prevent repetitive re-planning phases due to changes in construction design or construction methods, SE planning is usually conducted only after decisions on design and construction have been finalized (without in-depth information on required equipment). To address this issue and enable efficient CSLP, several semi-automatic solutions have been proposed in previous research. Key point to formulating these planning algorithms is extensive domain knowledge, both explicit and implicit. Explicit knowledge can be retrieved from regulations, guidelines, and local boundary conditions. However, in practice, numerous SE variants might be applicable for each construction project. To support the decision, implicit knowledge is vital. Implicit knowledge arises from expert knowledge and practical experience and is hard to extract and formulate in a strict, logical, and computable manner. Executing companies dread competitive disadvantages, and often hesitate when asked to give out delicate information on design decision on past construction sites. Therefore, we propose gathering information by analyzing large numbers of aerial photographs containing construction sites.

In this paper, we present an image analysis pipeline for retrieving information on past construction sites. We fuse information retrieved from convolutional neural networks for image analysis on airborne images with georeferencing and external data sources to retrieve a detailed description of the construction site at hand. The applicability of our approach is presented in a case study.

2. Related work

Image analysis, as part of computer vision, is a heavily researched topic, that got even more attention through recent advances in autonomous driving and machine learning related topics. For effective and efficient image analysis and object recognition, machine learning algorithms have been increasingly used during the last decades. In 2012, the convolutional neural network (CNN) "AlexNet" (Krizhevsky, Sutskever, and Hinton, 2012) achieved a top-5 error of 15.3% in the prestigious ImageNet Large Scale Visual Recognition Challenge (Russakovsky et al., 2015). These results were surprisingly accurate at the time, proving the advantages of using CNN. On this account, the software industry shifted towards using CNN for all machine learning based image processing tasks (LeCun, Bengio, and Hinton, 2015).

Image analysis on construction sites, on the other hand, is a rather new topic. Since one of the key aspects of machine learning is the collection of large datasets, current approaches focus on data gathering. Tajeen and Zhu (2014) present an image dataset containing numerous annotated images of construction equipment, however centering on excavation phase (excavator, loader, dozer, roller and backhoe) and images taken from ground. Kropp, Koch, and König (2018) detect indoor construction elements based on similarities, focusing on radiators. In the scope of automated construction progress monitoring, Han et al. published an approach for Amazon Turk based labelling (Han and Golparvar-Fard, 2017). Bügler et al. (2017) combined photogrammetric methods and video analysis to assess the progress of earthworks. To this end, they created point clouds to measure the volume of excavated soil and detected truck dumpers on images using foreground detection. Jahr, Braun, and Borrmann (2018) use an artificial intelligence approach to detecting formwork elements on UAV imagery of construction sites.

3. Methodology

To generalize implicit knowledge from airborne images, we propose an image analysis pipeline utilizing machine learning algorithms, georeferencing, as well as data retrieval (see Figure 1). In a first step, we detect construction sites by using an object detection algorithm on the airborne images. If at least one construction site is detected, an instance segmentation algorithm is used to detect individual elements of the construction. In this paper, we concentrate on detecting tower cranes, as they highly affect construction progress. To enable georeferencing, surveying information on the images is required. To be able to estimate element dimensions, the images are orthorectified. Additional information relevant to the construction site, such as building dimensions, property lines, or neighboring constructions, can be retrieved by georeferencing the image and linking it to spatial information retrieved from the cadastral map or other geoinformation services, such as OpenStreetMap or city models. Finally, all information retrieved is stored in a database for reliable data management and access.

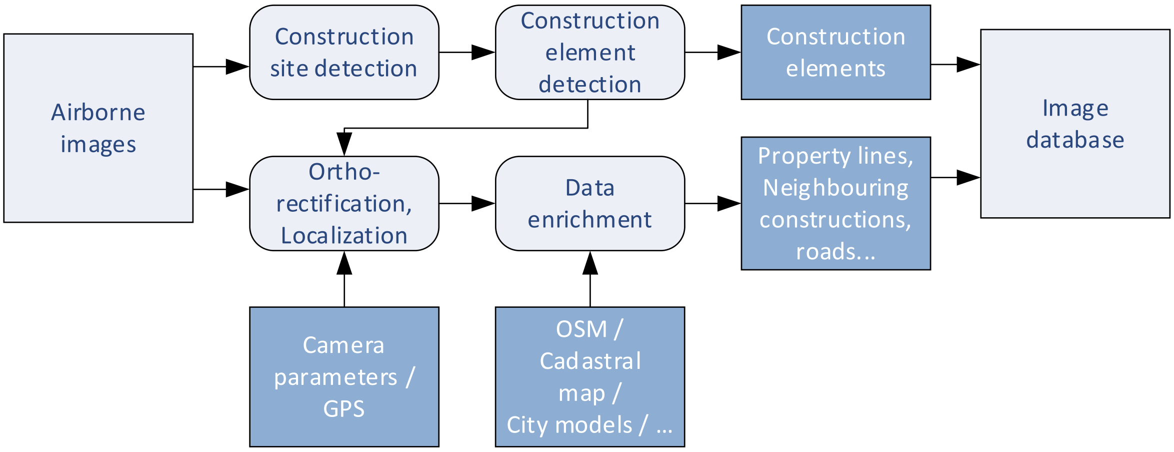

Airborne images
Construction site detection
Construction element detection
Construction elements
Image database
Ortho-rectification, Localization
Data enrichment
Property lines, Neighbouring constructions, roads...
Camera parameters / GPS
OSM / Cadastral map / City models / ...

Figure 1: Image analysis pipeline: In a first step, airborne images are scanned for construction sites. If any were found, individual construction elements are detected. To retrieve correct dimensions, the images are orthorectified and georeferenced. Additional data sources are used to add context.

3.1 Image analysis on airborne images

To evaluate the photographs, we use two different CNNs: the first network detects construction sites on airborne images, the second network segments the resulting cropped images.

Image analysis using convolutional neural networks. There are different tasks to be solved by image processing algorithms. Well known tasks include classification, where classes of single-object images are recognized; object detection, where several objects in one image may be classified and localized within the image; and image segmentation, where individual pixels are classified (Rusk, 2015). In this paper, we focus on object detection and instance segmentation using CNNs.

CNNs are structured in locally interconnected layers with shared weights (see Figure 2). Each layer comprises multiple calculation units, or neurons. The neurons of the first layer (input layer) represent the pixels of the analyzed image, the last layer (output layer) comprises the predictable object classes. Between input and output layer, any number of hidden layers can be arranged. To adapt to different problem domains, such as recognizing construction site elements, CNNs are trained. During training, the connections between certain neurons are increased, while the connections between other neurons are reduced—the weights connecting consecutive layers are weighted. The training is usually carried out using supervised backpropagation, meaning the network is fed with exemplary input-output pairs (Rusk, 2015). The expected output, viz. correct solution, for each input is called ground truth. To train a CNN towards reliable predictions, a significant amount of training data is required, which has to be prepared in a preprocessing step. To accelerate the training processes, weights of previously trained CNNs can be used. To adapt pretrained CNNs, usually the last layers are replaced with layers representing the new problem domain before training with new data.

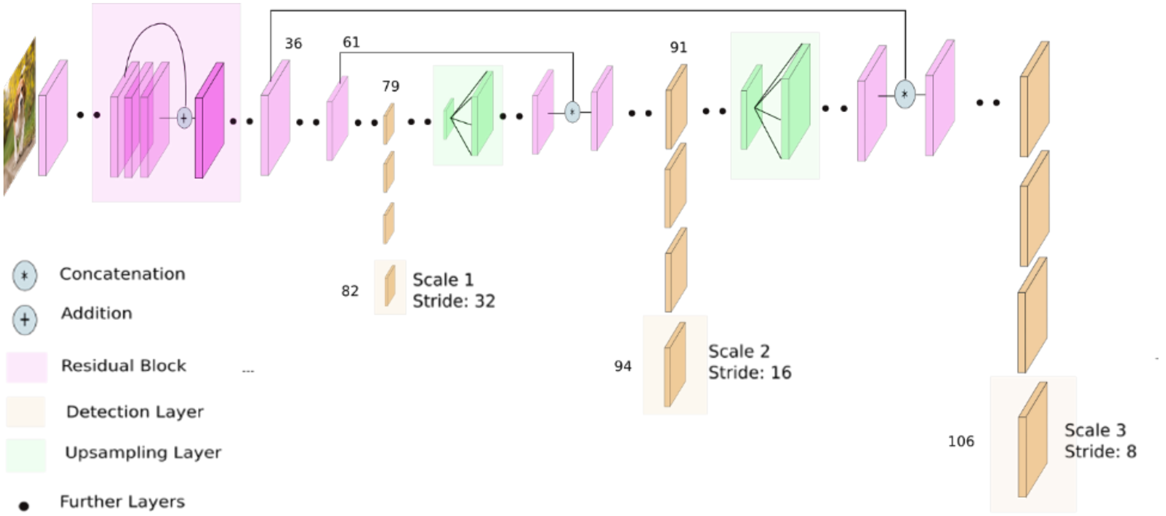


Figure 2: Layered structure of YOLOv3 (slightly modified from https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b)

Detecting construction sites using convolutional neural networks. To collect information on construction sites, first, they have to be localized. The task to classify and localize objects on images containing several objects of different types is called object detection (Rusk, 2015). To solve this task, algorithms usually predict rectangular areas (bounding boxes) with high probabilities of object occurrences, as well as the corresponding object class. To measure the performance of an algorithm, the intersection of union between prediction and correct solution is examined. Acknowledged measures include precision (How many of the predictions are correct?), recall (How many relevant items are predicted?) and mAP (mean average precision, calculated from recall and precision).

In this paper, we used an “you only look once” network (YOLOv3, see Figure 2) in Darknet framework (Redmon & Farhadi, 2018). YOLOv3 is a single shot detector, which enables reasonable prediction rates with very fast training and prediction times compared to other leading algorithms. YOLOv3 divides the input image in a grid, where each cell predicts only one object. Predictions of objects of varying sizes are enabled by a feature pyramid network—YOLOv3 makes predictions at three different scales for each location. To predict the bounding box of the detected object, YOLOv3 uses anchor boxes with dimensions tailored to the specific problem domain.

Segmenting construction elements using convolutional neural networks. As exact information on the whereabouts and dimensions of the site equipment is desired, each equipment has to be detected as precisely as possible. We want to know the exact shape of the object rather than its bounding box. Therefore, we need an instance segmentation algorithm that labels images pixelwise and is able to distinguish not only between several classes, but between several objects of the same class.

A very capable algorithm for instance segmentation is Mask R-CNN (He, Gkioxari, Dollár, and Girshick, 2017). Mask R-CNN predicts instance masks for detected objects in two stages: firstly, it uses a RoI (region of interest) Align network to locate bounding boxes and classes for possible objects. Secondly, a semantic segmentation model is used to determine the exact object outlines within the bounding box. Since only one object should be contained in each bounding box, a binary classifier mapping the pixels 1/0 is sufficient—1 representing the presence, 0 the absence of an object.

3.2 Orthorectification of images

Aerial images are not only source of information about the captured scene regarding the content but can also deliver information about localization of the objects and metric information. To be able to measure and localize objects in aerial images, the following information is needed:

- Exterior orientation parameters for each image (position and orientation of the camera during acquisition)
- Interior orientation parameters of the camera (obtained in the calibration process)
- Meshed digital Surface Model (DSM) generated from the images itself, if stereo pairs are provided, or DSM from another source.

Georeferencing. Determining the exterior orientation of the camera in the world coordinate system is called georeferencing. This can be done either by using ground control points (GCPs), or by using the GNSS (global navigation satellite system) position together with inertial measurement unit (IMU) and system calibration for camera orientation (direct georeferencing). Direct georeferencing has the advantage that the manual effort of measuring the GCPs is avoided. The accuracy of the direct georeferencing depends on the quality of the GNSS signal and can vary from few decimeters to few meters (Pfeifer, Glira, and Briese, 2012).

Ground Sample Distance. To estimate true dimensions in aerial photographs, the distance between two pixels on ground (Ground Sample Distance, GSD) must be known. In traditional aerial photogrammetry, images are acquired using nadir view. Assuming a locally flat Earth surface and knowing flight altitude, sensor size and camera constant (focal length), the GSD can be calculated. The altitude is determined relatively to the reference surface and not the terrain (Figure 3). Therefore, changes in the terrain height and presence of other objects (e.g. high buildings) lead to varying GSD. Furthermore, the flight altitude does not remain constant over the whole flight campaign, as well as vertical acquisition geometry not always can be ensured. Accordingly, only approximate GDS can be indicated. Modern aerial photogrammetric camera systems use a combination of nadir and oblique view cameras, delivering additional views on building facades and other 3D objects. GSD in oblique images, however, cannot be determined.

Orthophoto. During orthorectification, aerial images in central projection are transformed into orthogonal projection in order to unify the GSD and allow direct measurements in the images (Figure 4). For each cell of the DSM mesh, the corresponding part of the image is identified and transformed onto the DSM. The resulting mesh is then ortho-projected on a regular grid created on the reference surface with defined GSD. We orthorectify not only the color image, but also obtained labels.

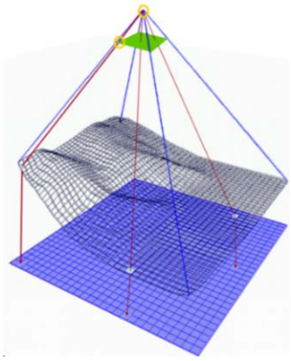


Figure 3: Flight altitude, terrain height and reference surface

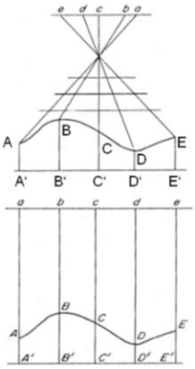


Figure 4: Orthorectification of a central projection

3.3 Additional sources and data enrichment

Surrounding amenities and conditions of construction sites pose big influences on the site equipment. For example, neighboring buildings might influence tower crane positions and minimum dimensions to ensure safe operation without interference; access roads are highly relevant for material supply and will influence construction roads and storage area positioning. If the location of a construction site has been determined through georeferencing the image, the information on that construction site can be enriched using spatial information on the surrounding amenities. Spatial information is available from different sources, e.g. cadastral maps. Cadastral maps show property lines and ownerships and may include additional information such as parcel numbers and existing structures.

A wide selection of digital geodata is available on OpenStreetMap[1] (OSM). OSM aims to collect and provide data under an open license. The geodata provided includes, inter alia, roads, parks, building outlines, and amenities such as fire hydrants and post boxes. While the data can be viewed as map representation, several APIs are available for data access, of which Overpass API is currently well maintained. Overpass works with queries either in xml or in its native language, Overpass QL. In this paper, we used the Overpass API with Overpass QL to retrieve information on neighboring buildings and roads.

Additional 3-dimensional information could be retrieved from city models. Depending on the model's Level of Detail (LoD), buildings are represented as 3D blocks (base area and height) or with increasing detailing, such as roof shape, window areas and even interior construction. City models are widely available. For storing and exchanging city models, the open standard CityGML, among other data models, might be used.

4. Case study

The presented image analysis steps were implemented individually to prove the proposed approach. The results are presented in the following sections.

4.1 Generating ground truth data for image analysis

To create ground truth data for both CNNs, we used airborne images provided by the German Aerospace Center (DLR) (Kurz et al., 2012). The aerial photographs were not commissioned for this paper, but rather repurposed from other applications, therefore not leading to additional data recording efforts. The images were gathered in Germany, both by airplane and helicopter. In total, approximately 4.500 high resolution images with varying image sizes have been sighted and labeled. In a first step, we extracted manually all images that contain construction sites within the construction phase (characterized by the use of tower cranes; visible material storage; first construction elements have been erected. See Figure 5). Subsequently, to generate the object detection dataset, we added bounding boxes for all construction sites using the labeling platform “Labelbox”[2]. We translated the labels from Labelbox format to YOLO format and split the data in training, testing and validation data set (Table 1).

Table 1: Number of images used for training, testing and validation, and number of construction sites

Images total	Number of construction sites	Training (70%)	Testing (20%)	Validation (10%
4443	1727	1209	345	173

[1] https://www.openstreetmap.org

[2] https://labelbox.com/

To prepare the segmentation dataset, we used images of tower cranes taken by UAVs, and images taken by hand cameras (mostly from below) (Figure 6). Additionally, we further processed images from the object detection dataset. We added a margin around the construction site bounding boxes to ensure that all relevant information is contained (especially tower cranes reaching outside construction field) and cropped along the labels. Corresponding image areas for neighboring construction sites with overlapping bounding boxes are contained in both crops. We again used Labelbox to add polygonal labels for all tower cranes. For training the Mask R-CNN network, we decided to use the COCO data format. In this case, too, we split the data in 70% training data, 20% testing data and 10% validation data.

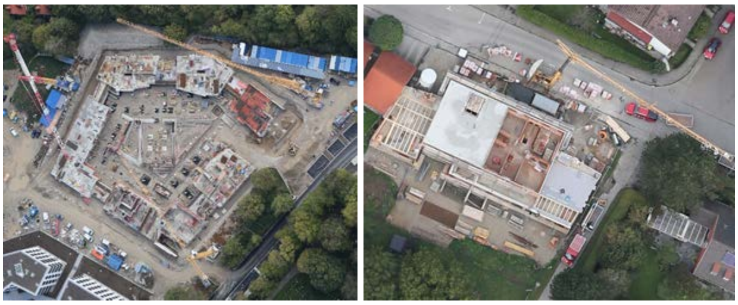


Figure 5: Samples for cropped images containing construction sites

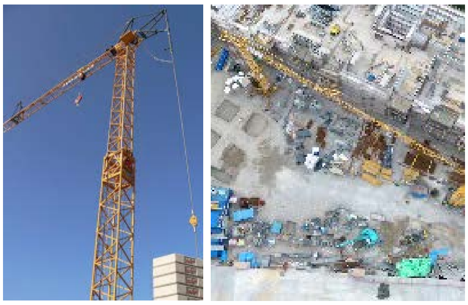


Figure 6: Samples for hand held camera and UAV images of tower cranes

4.2 Training CNNs for object detection and instance segmentation

Object detection with YOLOv3. For training the construction detection network, we use a YOLOv3 architecture in Darknet (Redmon and Farhadi, 2018). To better adapt the network to the construction site dataset, we regenerated the anchor sizes. To that end, we used k-means clustering on the aggregate of bounding boxes in the construction site dataset. The resulting 9 clusters for bounding box width and length, normalized on the respective image size (see Figure 7), are used as length and width of the anchors.

To reduce training time and retrieve better results with our limited data set, we use the pretrained weights of the Darknet53 network. Training for 1000 epochs took approximately 10h on an Nvidia DGX-1. Examples for resulting bounding boxes are depicted in Figure 8. While bounding boxes for smaller construction sites are very well predicted, the CNN is not yet sufficiently adapted for larger construction sites. When sighting the dataset, it gets apparent, that smaller construction sites are predominant in residential areas and make up for a majority of the dataset. To a broader variety of construction sites, the training dataset is currently expanded further. Another step to improve the object detection algorithm entails more extensive preprocessing of the data.

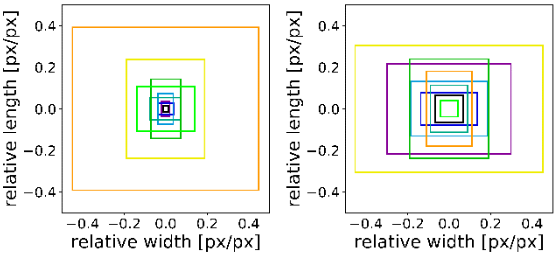


Figure 7: Original YOLOv3 anchor boxes (left) and construction site anchor boxes (right)

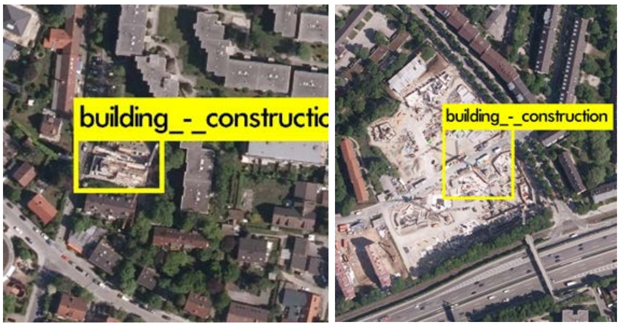


Figure 8: Examples for resulting bounding boxes from validation set

Tower crane segmentation with Mask R-CNN. For image segmentation, we used Mask R-CNN in Keras with TensorFlow backend (He et al., 2017). We used pretrained weights from the COCO Dataset (Lin et al., 2014). Mask R-CNN adjusted very fast to the tower crane dataset, leading to low loss after few epochs (Figure 9). Examples for resulting instance bounding boxes and masks are depicted in Figure 10. Object bounding boxes are predicted reliably, while, in some cases, masks tend to disconnections. To improve the predictions, the training data set is increased continuously.

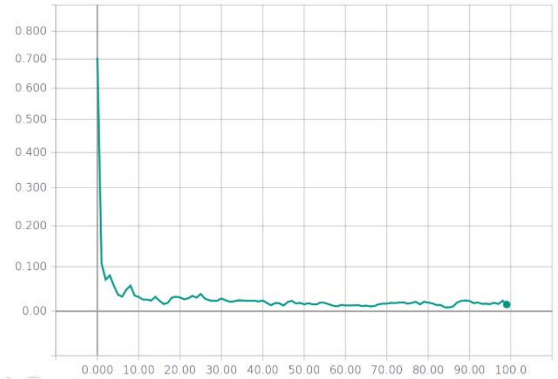


Figure 9: Loss for training Mask R-CNN on tower crane dataset

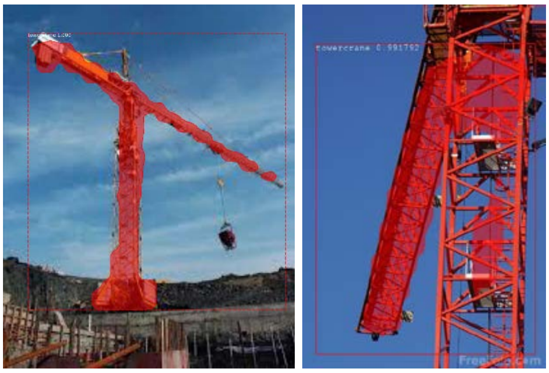


Figure 10: Examples for resulting masks (in red) from validation set.

4.3 Georeferenced information

In Figure 11, we present results for orthorectification. First, the DSM was generated (left) and then the orthophoto was calculated (right). Here, the seamline dividing the areas covered by color information from two images cuts a tower crane into two parts, which makes it difficult to detect tower cranes directly in the orthophoto. Therefore, cranes were labeled in original aerial images (Figure 12, left) and the labels were orthoprojected on the DSM (Figure 12, right). As we see in Figure 11, crane 1 is not present in the DSM at all, and from crane 2 only the tower of the crane is mapped in the DSM. This is because the crane structure is not easy to reconstruct from images. As a very thin structure, cranes do not provide many points in the point cloud. Points that are detected are on much higher level than the surrounding, therefore many algorithms detect those points as outliers and remove them from DSM. In addition, cranes may move during data acquisition, which leads to further difficulties in the reconstruction. The remedy for this situation is using true orthophotos based on high-density and high-accuracy point clouds, which contain entire tower cranes. True orthophotos, however, require high overlaps between the images (80% in flight direction and 60% between the image stripes). Here not only the difficult geometry of the crane must be considered, but also its dynamic behavior. This aspect should be a subject of our future investigations. Currently, we select for orthorectification labels which are located close to the image center and orthorectify labels object-wise, which means that we prevent seamlines splitting the labels.

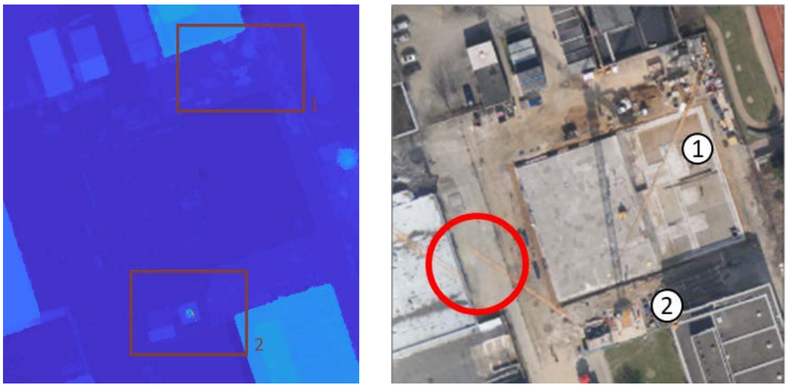


Figure 11: Difficulties with orthorectification of cranes: DSM (left); orthophoto sections with cranes (right)

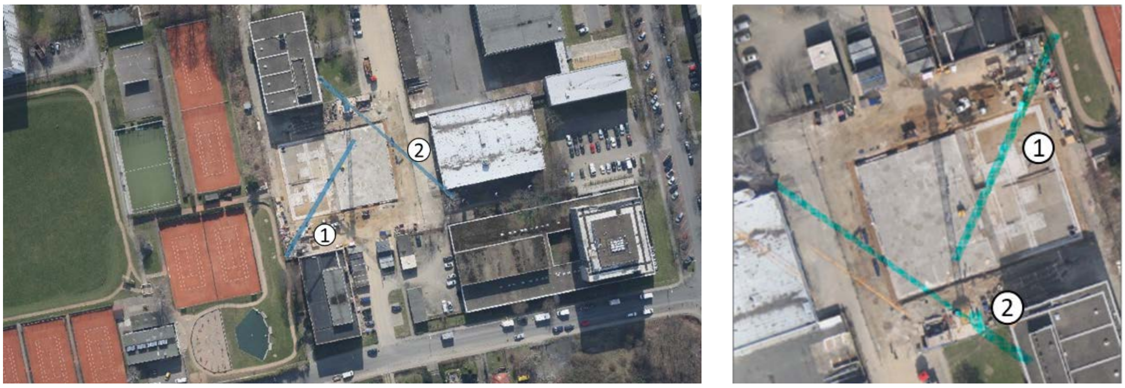
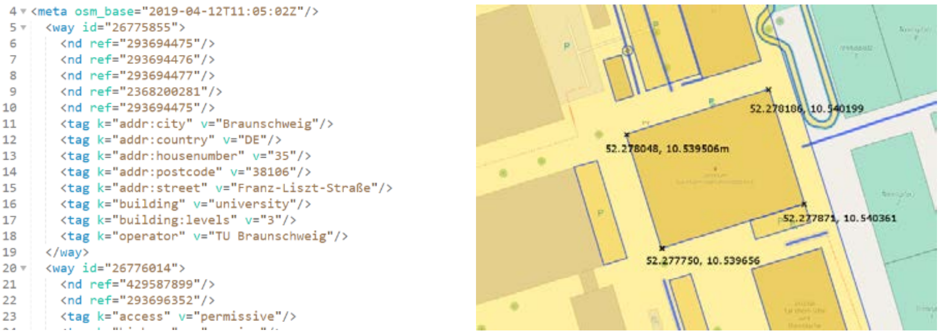


Figure 12: Orthorectification and georeferencing of labels: aerial image with labeled tower crane (left); orthorectified and georeferenced labels stored as GeoTIFF (right)

4.4 Data enrichment using Overpass API

To retrieve surrounding amenities to the construction site in question from OSM, we used Overpass API. Using the GPS coordinates of the construction's outer corners, we queried adjacent roads and footways as well as neighboring buildings. Overpass API returns the queried data text based (i.e. as json or xml). For monitoring the results, we used Overpass turbo[3] (Figure 13). Overpass turbo is a web-based tool able to run Overpass API queries. The results are shown as interactive map, where further information on the queried nodes can be retrieved. The additional data collected from OSM is added to the information retrieved from the images.



```
<meta osm_base="2019-04-12T11:05:02Z"/>
  <way id="26775855">
    <nd ref="293694475"/>
    <nd ref="293694476"/>
    <nd ref="293694477"/>
    <nd ref="2368200281"/>
    <nd ref="293694475"/>
    <tag k="addr:city" v="Braunschweig"/>
    <tag k="addr:country" v="DE"/>
    <tag k="addr:housenumber" v="35"/>
    <tag k="addr:postcode" v="38106"/>
    <tag k="addr:street" v="Franz-Liszt-Straße"/>
    <tag k="building" v="university"/>
    <tag k="building:levels" v="3"/>
    <tag k="operator" v="TU Braunschweig"/>
  </way>
  <way id="26776014">
    <nd ref="429587899"/>
    <nd ref="293696352"/>
    <tag k="access" v="permissive"/>
```

Figure 13: Neighboring elements (footways, highways, buildings) of the construction site as retrieved by Overpass API. Left: XML scheme, right: Visualization using Overpass turbo, queried elements are marked with blue outlines.

[3] https://overpass-turbo.eu/

5. Summary

In this contribution, we presented an image analysis pipeline capable of detecting construction sites as well as construction elements on airborne images. To gain further information on the situation on site, we retrieve real dimensions of the construction field and tower cranes by orthorectifying the images. Further data sources can be used to enrich the information. Using the overpass API, we retrieve information on the site's surroundings from OpenStreetMap. In the end, we gained a knowledge database for construction sites in Germany, which will be dynamically extended. Next steps include the extension with further construction site elements such as containers or vehicles, connection to city models to retrieve 3D information and the advancement of available algorithms using the knowledge database for solving the CSLP.

References

Bügler, M., Borrmann, A., Ogunmakin, G., Vela, P. A., & Teizer, J. (2017). Fusion of Photogrammetry and Video Analysis for Productivity Assessment of Earthwork Processes. *Computer-Aided Civil and Infrastructure Engineering*, *32*(2), 107–123. https://doi.org/10.1111/mice.12235

Han, K. K., & Golparvar-Fard, M. (2017). Potential of big visual data and building information modeling for construction performance analytics: An exploratory study. *Automation in Construction*, *73*, 184–198. https://doi.org/10.1016/j.autcon.2016.11.004

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, 2961–2969.

Jahr, K., Braun, A., & Borrmann, A. (2018). Formwork detection in UAV pictures of construction sites. *Proc. of the 12th European Conference on Product and Process Modelling, Copenhagen, Denmark.*

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 1097–1105.

Kropp, C., Koch, C., & König, M. (2018). Interior construction state recognition with 4D BIM registered image sequences. *Automation in Construction*, *86*, 11–32. https://doi.org/10.1016/J.AUTCON.2017.10.027

Kurz, F., Meynberg, O., Rosenbaum, D., Türmer, S., Reinartz, P., & Schroeder, M. (2012). Low-cost optical camera system for disaster monitoring. *Int. Archives of the Photogrammetry, Remote Sens. and Spatial Information Sci*, *39*, B8.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., … Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. *European Conference on Computer Vision*, 740–755.

Pfeifer, N., Glira, P., & Briese, C. (2012). Direct georeferencing with on board navigation components of light weight UAV platforms. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *39*(B7), 487–492.

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *ArXiv Preprint ArXiv:1804.02767*.

Rusk, N. (2015). Deep learning. *Nature Methods*, Vol. 13, p. 35. https://doi.org/10.1038/nmeth.3707

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., … Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, *115*(3), 211–252. https://doi.org/10.1007/s11263-015-0816-y

Tajeen, H., & Zhu, Z. (2014). Image dataset development for measuring construction equipment recognition performance. *Automation in Construction*, *48*, 1–10. https://doi.org/10.1016/J.AUTCON.2014.07.006