Automatic Detection of Shaft Locations in Site Plans

Timo Santehanser[1] and Phillip Schönfelder[2]
[1]Department of Computer Science, Ruhr University Bochum,
Universitätsstraße 150, 44801 Bochum, Germany
[2]Department of Civil and Environmental Engineering, Ruhr University Bochum,
Universitätsstraße 150, 44801 Bochum, Germany
E-mail(s): timo.santehanser@rub.de, phillip.schoenfelder@rub.de

Abstract: Site plans play a crucial role in construction projects, providing detailed layouts of structures and infrastructure components. Extracting specific information, such as sewage system details including shafts and pipeline routes, from these plans is essential for accurate cost estimation. However, it remains a labor-intensive task, especially for pixel-based drawings lacking machine-interpretable vector geometries and machine-readable text elements. In this study, we conceptualize an automated method to streamline this process, leveraging object detection and optical character recognition (OCR) techniques. The proposed approach involves three main steps: (1) locating the legend region in which the shaft symbol is specified using a specialized OCR method, (2) identifying the relevant shaft symbol from the plan's legend, and (3) detecting shaft locations within site plans using state-of-the-art object detection algorithms. Developing this method aims to significantly increase efficiency in construction cost estimation by automating the tedious task of extracting and analyzing site plan data. The preliminary results included in this study demonstrate candidate techniques for symbol processing in site plans.

Keywords: site plan, automation, computer vision, deep learning, drawing analysis

Erschienen in Tagungsband 35. Forum Bauinformatik 2024, Hamburg, Deutschland, DOI: 10.15480/882.13530

1 Introduction

Site plans constitute a central component of building applications in Germany, in accordance with the building code regulations [1]. They consist of a written and a graphical section, with this work focusing on the latter. The graphical section provides a scaled top-down view of the planned object, typically at 1:500 in densely built areas and 1:1000 in sparsely built areas [2]. In particular, site plans depict pipelines, including utility lines (water, gas, electrical), as well as phone and internet cables. The visualization of pipelines adheres to DIN 2425 [2], which defines symbols and line types to denote different pipeline types, directions, and functions. Site plans also depict shafts, which are structural elements typically embedded in the ground, serving various functions such as the intake and/or discharge of water, as well as providing access to underground pipelines. The visualization

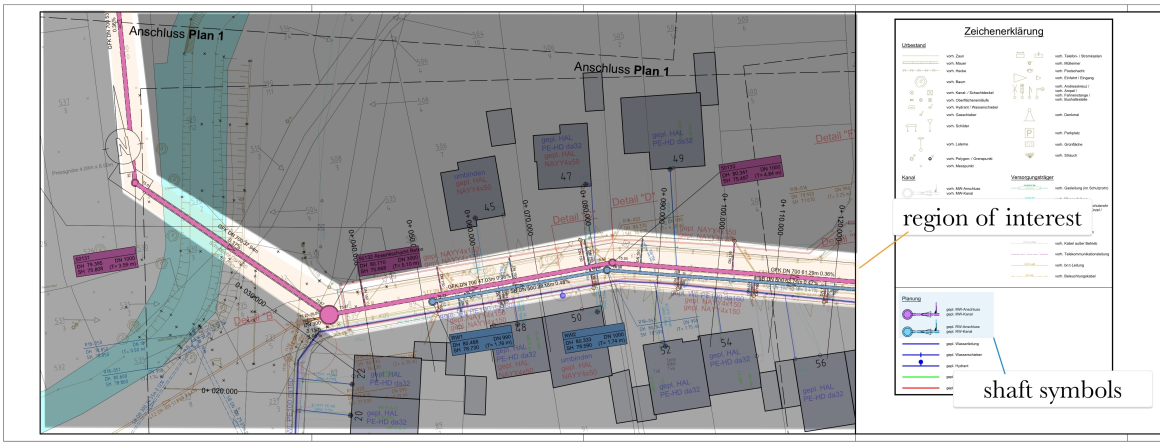

Figure 1: An example site plan, with highlighted areas for shaft symbols within the legend, as well as the region of interest containing planned shafts and pipelines

of shafts typically includes additional details like depth, dimensions, and connections to other shafts within the plan. An example site plan is shown in Figure 1.

Extracting information about sewage systems, including the routes of pipelines and locations of shafts, is beneficial for multiple reasons. First, this information is essential for accurate quantity estimation, and, thus, for the cost calculation during a project's planning phase. Second, in the construction phase, this information can be useful for creating 3D models of the construction site, which can help coordinate construction machinery to avoid conflicts with existing components through smart visualization systems. Third, such 3D models can be used in the later stages of a building's life-cycle as well, for instance, for scheduling maintenance tasks or planning adjacent buildings.

However, actually extracting the information from the plans remains a labor-intensive task, especially for pixel-based drawings that lack machine-interpretable vector geometries and machine-readable text elements. Currently, human experts manually retrace pipeline routes and locate shafts in the original plans to reconstruct the geometry. Although specialized software can aid this process, it is still time-consuming and highlights the need for an automated solution.

Therefore, in this study, we aim to conceptualize an automated system for extracting and interpreting information from site plans. The three main steps of the proposed system notably leverage recent advancements in the symbol recognition research for spatial diagrams, e.g., regarding specialized Optical Character Recognition (OCR) and legend processing [3], [4] as well as shafts and symbol detection [5], [6]. The system is meant to automatically process a scanned site plan, thereby streamlining the digitization process and improving the efficiency and accuracy of construction planning and management.

The remainder of this paper is organized as follows: Section 2 revisits previously proposed methods for diagram processing, addressing their strengths and limitations. Section 3 presents details on the proposed system. Section 4 reflects on the implications and shortcomings of this study. Finally, Section 5 concludes the paper.

2 Related Work

The current method for extracting pipeline route information from site plans heavily relies on human interaction with the document. Specifically, individual shafts are identified manually, and the pipeline routes are reconstructed by connecting these shafts. Additionally, details such as the depth of the shafts must be recognized in a separate section of the site plan and manually evaluated. This approach is not only time-consuming but also prone to errors [7]. Consequently, the capture and analysis of data from site plans require a high level of manual effort and expertise.

Previous works on automating similar symbol recognition tasks have shown that existing methods often require a predefined set of symbols, particularly when using neural network-based computer vision approaches [6], [8], [9]. This requirement stems from the learning phase of neural networks, which necessitates an annotated training dataset and, thus, a predefined list of symbols to be detected. This dependency limits the applicability of such methods to site plans, where symbol representations can vary due to factors such as the creator, the standards followed, or other contextual differences. This limitation highlights the need for more flexible and adaptive approaches. Few-Shot Learning (FSL) offers a potential solution to the problem of large variety in real site plan collections. FSL refers to a model's ability to learn new tasks or classes from only a few examples. The goal is for the model to generalize quickly from these few examples. [10]

One common method in FSL is the use of prototypical networks, which calculate a prototype in the feature space for each class and classify new examples based on their similarity to these prototypes. [11] Another popular approach is Model-Agnostic Meta-Learning (MAML), which aims to initialize a model so that it can be optimized for new tasks with minimal adjustment steps. [12] One-Shot Learning is a specialization of FSL where the model requires only a single example per class to recognize that class successfully. This approach is particularly challenging, as it demands extreme efficiency in transfer learning and generalization. [13] Techniques used in One-Shot Learning are often similar to those in Few-Shot Learning but impose even stricter constraints on the number of available examples. A recently developed method by Gupta et al. [14] describes how symbols in Piping and Instrumentation Diagrams (P&IDs) can be recognized using FSL. In this method, first, all symbols in the diagrams are detected, and, second, the symbols are classified using a Siamese network. This approach demonstrates the effectiveness of FSL in scenarios where annotated training data is limited and symbols may vary, underscoring its potential for application to site plans where symbol variability and limited training data are common challenges as well.

3 Method

3.1 Overall Concept

Since site plan symbols might not be known a priori, the automatic extraction of shaft symbols and pipelines on site plans must be divided into two stages: (1) extracting shaft symbol classes from the legend and (2) locating their respective positions on the site plan itself. Addressing both stages, we propose a concept for an automated site plan analysis method, which is illustrated in Figure 2. First, text detection is performed on the site plans using a pretrained YOLOv8 [15] model. The detected text

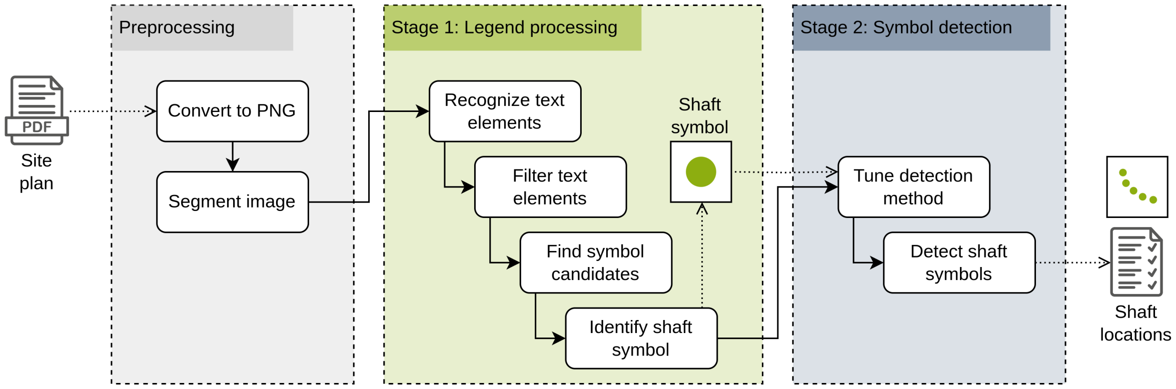

Figure 2: Proposed shaft localization concept.

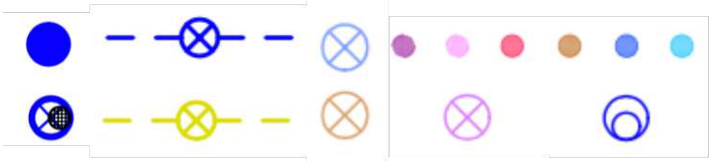

Figure 3: Examples of shaft representations in site plans.

objects are then converted to strings using Tesseract OCR [16], and filtered for keywords to locate the legend and identify the shaft symbols within it. Using morphological operations, nearby candidate symbols are found to select one of them as the shaft symbol. Implementation details of this stage are given in Section 3.3. Second, these symbols are then used to find the shaft symbols in the site plan, determine the pipelines, and calculate the positions of the shafts. For this stage, preliminary results suggest which methods are suitable for the final development, as detailed in Section 3.4. Additionally, preprocessing steps are outlined in Section 3.2.

3.2 Preprocessing

Site plans possess certain characteristics that necessitate preprocessing. First, in the data at hand, they are available as PDF files, requiring conversion to a format suitable for image processing, e.g., by Convolutional Neural Networks (CNN). Therefore, the files are converted to PNG. Second, since the YOLOv8 architecture was developed for images of size up to 1280 pixels in side length, this site plan images are segmented to into several smaller images. To ensure that segmentation boundaries do not render parts of the site plan unreadable, which could lead to missed symbols, segments are created with a 40% overlap with their neighboring segments.

3.3 Symbol detection in legends

Due to the various symbols for shafts on plans, with some examples depicted in Figure 3, training a learning-based model for automated symbol detection presents difficulties. The national DIN standard [2] does not specify how shafts must be represented, so the symbols' appearances are largely up to the plan's draftsman or company standard. However, since the symbols for the documented shafts are always indicated in a legend on the plan, the proposed concept involves recognizing the symbols for shafts in the legend and then identifying these in the plan's graphical section, indicated as region of interest in 3.

Symbol class recognition is achieved through the following steps:

1. *Text detection.* To perform text detection on the site plan, a YOLOv8 model pretrained by Schönfelder et al. [3] is employed. Originally trained on floorplan drawings, it is expected to generalize to site plans, as the characteristic text element features are similar.
2. *Text recognition.* To convert the text regions to strings, Tesseract OCR is applied to the identified regions. It offers fast processing and shows high performance on good quality site plan images. To safely include the entire text element, the detected regions are extended by a two pixel margin.
3. *String matching.* The recognized strings are searched for key terms like `Shaft` and `Connector`, to identify the relevant regions in the legend.
4. *Horizontal cropping.* The image is cropped to the region where the corresponding symbol is assumed to be located, which is the smallest rectangle containing the text region, and spanning the entire width of the image segment. This decreases the likelyhood of false positive matches.
5. *Morphological processing.* With the exception of the text region itself, the crop undergoes several processing steps, including binary thresholding and contour approximation to identify symbol region candidates.
6. *Symbol identification*. The symbol is finally identified by selecting the candidate region that is closest to the text region, while also being larger than 10x10 pixels.

The described steps, in particular the choice of key terms and the symbol identification step parameters, are validated with a set of 50 (German) example site plans.

3.4 Symbol detection in the graphical section

This section discusses some approaches for symbol recognition outside of the legend on the site plan that are evaluated in the course of this study.

3.4.1 Template matching

Template matching relies on the symbol template as provided by the legend, and includes a brute-force search of this template in the entire image. While this offers a potential solution to the symbol detection steps, it is found to be impractical for two main reasons. First, symbols might be partially occluded by other lines, rendering the symbol undetectable for symbol matching, which requires a near perfect match with the template. Second, the symbols in the plan might have been subject to affine transformations, and appear rotated or scaled. Even in the case of axisymmetric symbols, this necessitates an unreasonably large number of additional templates to include rotated and scaled templates to the search, which proves computationally extensive while still often inaccurate. Therefore, one should rather focus on techniques invariant to scale and rotation, like homography finding or CNNs.

3.4.2 Homography finding

With regards to homography finding, cv2 implements the `findHomography` function, which computes a 3x3 homography matrix describing a perspective transformation between two images. This matrix allows for the projection of points from one image into another, considering perspective differences, inluding rotation and scaling. The homography matrix H is a 3x3 matrix describing the relationship between the homogeneous coordinates of the template symbol in the legend and the candidate symbol in a plan segment. It is defined as

$$s\begin{bmatrix}x'\\y'\\1\end{bmatrix}=H\begin{bmatrix}x\\y\\1\end{bmatrix}, \tag{1}$$

where (x, y) are the template symbol's coordinates, (x', y') are the coordinates of the candidate symbol, and s is a scaling factor. To compute H, at least four point correspondences between the two symbols are needed. In our implementation, we use the Scale-Invariant Feature Transform (SIFT) to extract keypoints and descriptors from both the template symbols and the sliding windows within the plan segments. To match the keypoints between the template and candidate symbols, we employ a FLANN-based matcher (Fast Library for Approximate Nearest Neighbors), optimized for fast and efficient high-dimensional feature matching. Lowe's ratio test is applied to filter out weak matches, retaining only those with a significantly better best match, thereby reducing false positives.

Once the matches are identified, the RANSAC algorithm within the `findHomography` function is used to compute the homography matrix H. RANSAC iteratively selects random subsets of matches to estimate the homography and evaluates the number of inliers—points that fit well with the estimated transformation. The number of inliers serves as a metric for match strength, with valid matches exceeding a predefined threshold. This process is applied across sliding windows in the plan, with the best homography matrix for each window selected based on the highest number of inliers, ensuring robustness even under rotation, scaling, or minor distortions.

By combining SIFT for feature extraction, FLANN for efficient matching, and RANSAC for robust homography estimation, our method achieves accurate and reliable symbol detection across complex plan images, with a precision of 82% and a recall of 78%.

4 Discussion

The extraction of shaft symbols from the legend is an effective solution for the non-uniform symbols in the plans, which make training a dedicated model difficult. While this approach is effective for the data sample at hand, it might however require algorithmic adjustment for plans that exhibit unseen formats. However, the fundamental process is expected to remain functional for most legend formats. In our evaluation of the FSL approach as demonstrated by the recent study of Gupta et al. [14], we found that while the method shows promise in the context of P&IDs, it does not generalize to the domain of site plans. One of the primary challenges is that, unlike P&IDs, where symbols typically exist in isolation without overlapping with other elements like pipelines, site plans present a more complex environment. In site plans, symbols often overlap with various other content, including pipelines, structures, and roads, making reliable symbol detection significantly more challenging.

Regarding the learning-free attempts towards symbol detection in the graphical section of the plan, implementations of template finding show poor results in the sample dataset. Especially the large image sizes of the site plans and the potential rotation and scaling of the symbols increases the runtime by multiple orders of magnitude. The investigated homography matching approach successfully identified the majority of symbols extracted from the legend within the site plans. However, challenges remain, particularly with low-contrast symbols, such as yellow symbols on a white background. These cases may require further optimization or the application of pre-processing filters to enhance symbol visibility and improve detection accuracy. Another significant challenge arises with the recognition of less complex, symmetric symbols, such as simple geometric shapes or symbols with uniform color distributions. The homography-based approach, while effective for more detailed and feature-rich symbols, struggles to detect these simpler symbols due to the lack of distinctive keypoints.To address this limitation, a hybrid approach could be a viable solution. Such an approach would combine the strengths of different methods to ensure robust symbol detection across a variety of symbol types. For instance, the process could begin with SIFT-based keypoint detection for complex symbols, followed by template matching for simpler, geometric symbols that lack sufficient keypoints.

5 Conclusion

As a first step towards the automatic extraction of shaft locations and pipeline routes in site plans, this study achieves (1) a concept for and end-to-end extraction method for shaft symbols, (2) a proof-of-concept for a legend processing method that enables extraction of plan-specific symbol class templates, and (3) considerations about candidate techniques for symbol detection on site plans with few available training examples. In terms of learning-free symbol detection, a homography-based approach shows promising in accurately detecting symbols within the provided site plans, even though the symbols are subjected to variations in scale and orientation. However, it is important to note that parameter optimization remains a critical step in this pipeline, requiring further fine-tuning to maximize detection accuracy. Additionally, the performance of this approach on symbols with suboptimal contrast ratios remains an area that needs further development. Symbols that blend with the background, such as those with low contrast, present challenges that necessitate the application of advanced pre-processing techniques or adjusted detection parameters to improve reliability. Overall, while the homography-based method has shown substantial potential, ongoing refinement is necessary to fully address the nuances of symbol detection in diverse site plan contexts.

Acknowledgments

The authors would like to express their gratitude to isl-kocher GmbH and IB Becker for providing the sample dataset used in this study.

References

[1] Bauministerkonferenz, *Musterbauordnung (MBO), zuletzt geändert durch Beschluss vom 23.09.2022*. [Online]. Available: https://www.dibt.de/fileadmin/dibt-website/Dokumente/Rechtsgrundlagen/MBO.PDF (visited on 07/07/2024).

[2] *DIN 2425-4:2022-11, Planwerke für die Versorgungswirtschaft, die Wasserwirtschaft und für Fernleitungen_- Teil_4: Entwässerungssysteme außerhalb von Gebäuden*. DOI: 10.31030/3384593.

[3] P. Schönfelder, F. Stebel, N. Andreou, and M. König, “Deep learning-based text detection and recognition on architectural floor plans”, *Automation in Construction*, vol. 157, p. 105 156, 2024. DOI: 10.1016/j.autcon.2023.105156.

[4] T. Hassan, J. Verges-Llahi, and A. Gonzalez, “High-Performance Preprocessing of Architectural Drawings for Legend Metadata Extraction via OCR”, in *Proceedings of the 2017 ACM Symposium on Document Engineering*, 2017, pp. 197–200. DOI: 10.1145/3103010.3121042.

[5] S. Sarkar, P. Pandey, and S. Kar, “Automatic Detection and Classification of Symbols in Engineering Drawings”, *arXiv*, 2022. DOI: 10.48550/arXiv.2204.13277.

[6] A. Rezvanifar, M. Cote, and A. B. Albu, “Symbol Spotting on Digital Architectural Floor Plans Using a Deep Learning-based Framework”, in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020, pp. 2419–2428. DOI: 10.1109/CVPRW50498.2020.00292.

[7] S. Paliwal, A. Jain, M. Sharma, and L. Vig, “Digitize-PID: Automatic Digitization of Piping and Instrumentation Diagrams”, in *Proceedings of the 2021 Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2021, pp. 168–180. DOI: 10.1007/978-3-030-75015-2_17.

[8] L. Jamieson, C. Francisco Moreno-García, and E. Elyan, “A review of deep learning methods for digitisation of complex documents and engineering diagrams”, *Artificial Intelligence Review*, vol. 57, no. 6, p. 136, 2024. DOI: 10.1007/s10462-024-10779-2.

[9] E. Elyan, L. Jamieson, and A. Ali-Gombe, “Deep learning for symbols detection and classification in engineering drawings”, *Neural Networks*, vol. 129, pp. 91–102, 2020. DOI: 10.1016/j.neunet.2020.05.025.

[10] X. Li, Z. Sun, J.-H. Xue, and Z. Ma, “A concise review of recent few-shot meta-learning methods”, *Neurocomputing*, vol. 456, pp. 463–468, 2021. DOI: 10.1016/j.neucom.2020.05.114.

[11] J. Snell, K. Swersky, and R. S. Zemel, “Prototypical Networks for Few-shot Learning”, in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 4080–4090. [Online]. Available: https://dl.acm.org/doi/abs/10.5555/3294996.3295163#.

[12] C. Finn, P. Abbeel, and S. Levine, “Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks”, in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, 2017, pp. 1126–1135. [Online]. Available: https://dl.acm.org/doi/10.5555/3305381.3305498.

[13] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, “Generalizing from a few examples: A survey on few-shot learning”, *ACM Computing Surveys*, vol. 53, no. 3, p. 63, 2020.

[14] M. Gupta, C. Wei, and T. Czerniawski, “Semi-supervised symbol detection for piping and instrumentation drawings”, *Automation in Construction*, vol. 159, p. 105 260, 2024. DOI: 10.1016/j.autcon.2023.105260.

[15] G. Jocher, A. Chaurasia, and J. Qiu, *YOLOv8*, 2024. [Online]. Available: https://github.com/ultralytics/ultralytics (visited on 07/07/2024).

[16] R. Smith, “An Overview of the Tesseract OCR Engine”, in *Proceedings of the 9th International Conference on Document Analysis and Recognition (ICDAR)*, 2007, pp. 629–633. DOI: 10.1109/ICDAR.2007.4376991.