

Automatic Identification and Vectorization of Traffic Infrastructure Features from Orthophoto Images

Zdeněk Svatý¹, Pavel Vrtal², Luboš Nouzovský³, Jakub Nováček⁴

¹ CTU in Prague, FTS, Department of Forensic Experts in Transportation, Konviktská 20, 110 00, Prague - svaty@fd.cvut.cz

² CTU in Prague, FTS, Department of Forensic Experts in Transportation, Konviktská 20, 110 00, Prague - vrtalpav@fd.cvut.cz

³ CTU in Prague, FTS, Department of Forensic Experts in Transportation, Konviktská 20, 110 00, Prague - nouzovsky@fd.cvut.cz

⁴ CTU in Prague, FTS, Department of Forensic Experts in Transportation, Konviktská 20, 110 00, Prague - jakub.novacek@fd.cvut.cz

Keywords: Orthophoto, Road Infrastructure, Colour Segmentation, Morphological Operations, Automation, Vectorization

Abstract

Orthophoto imaging of the Earth's surface using unmanned aerial systems have in recent years become a common and efficient method for acquiring highly detailed orthophoto maps. These are widely used in transportation and civil engineering fields. In the context of traffic accidents and technical documentation, such imagery can be applied for accurate reconstruction of the scene. However, this process often requires manual vectorization of selected road infrastructure features. This task is time-consuming and demanding, especially in more complex scenarios. The presented paper introduces a newly proposed method for semi-automatic vectorization of road infrastructure features from raster imagery. The method was implemented in MATLAB and consists of several sequential steps. These include selection of the area of interest, colour identification, noise reduction, clustering, and generation of vector contours. The entire process emphasizes simplicity, computational efficiency, and ease of use without the need for machine learning or extensive training data. Statistical evaluation using a paired t-test ($p = 0.0022$) confirmed that the automated approach is significantly faster than manual processing. On average, the proposed semi-automatic vectorization process was 2.15 times faster. In realistic scenarios, such as entire intersection areas, a speed increase of up to 3.1 times was achieved. These results confirm the practical benefit of the proposed method for efficient and rapid processing of traffic infrastructure image documentation.

1. Introduction

Imaging of the Earth's surface represents one of the fundamental methods for collecting geospatial data. Obtaining these image data is not only dependent on satellite images or aerial photography, but increasingly also on unmanned aerial vehicles (Svatý et al., 2023). Low-altitude image capture enables the acquisition of very high-resolution surface information (McGlone et al., 2004). Another undeniable advantage of UASs is the speed of data collection and the timeliness of the results. Using image correlation techniques, orthographic images can be generated (Knyaz et al., 2016) and subsequently used for further analysis.

Orthophotos are commonly applied in transportation engineering and construction-related disciplines. For example, sections of road networks can be mapped to provide up-to-date visual overviews of specific locations (Pinto et al., 2020). Aerial image documentation can also be captured during traffic accident investigations. These image materials can then serve as a base for situational plans in design studies or as a mapping background for graphical reconstruction of accident events.

Reconstruction of such sites is typically carried out by manually redrawing selected road infrastructure objects into vector form, while the raster image continues to serve only as a background layer. These objects usually include road markings, road edges, and, in some cases, roadside barriers. Manual vectorization can become increasingly time-consuming with the growing complexity or size of the site, requiring users to spend considerable time redrawing existing features that add little value to the analytical phase of the work. However, this step is essential for most subsequent processes.

The aim of this paper is to present the individual steps of a newly proposed method for identifying and vectorizing selected elements of road infrastructure with minimal user input. Emphasis is placed on process automation, user-friendly control, and computational speed. By simplifying the user's workload, more time can be devoted to the actual problem analysis of the given site. Additionally, the method allows for the generation of quick image-processing outputs even by users with limited technical skills, such as police officers preparing investigation report of a traffic accident.

The proposed method does not rely on machine learning techniques and thus does not require large volumes of training or testing data. The outcomes of the method will be demonstrated through sample use cases and compared with traditional approaches based on manual vectorization of selected road infrastructure features. This comparison aims to determine whether the use of automated processes provides a more efficient approach to processing image-based documentation.

2. Related work

The review section is primarily focused on methods currently used for image processing, as this aspect is fundamental to the present work.

The initial detection of information from images began in the 1980s (Zhang and Suen, 1984). At that stage, the process could not yet be described as vectorization of raster images. However, methods were introduced for identifying the skeletal structure of a graphical representation, i.e., image skeletonization. The principle of the method was based on iterative cycles in which unnecessary points were eliminated in each iteration.

These methods were gradually improved, and by the 1990s, actual image vectorization could be observed in the processing of cadastral map documentation (Janssen and Vossepoel, 1997). This involved progressive line smoothing and guiding curves through areas with dark pixels. A binary mask was applied, which could be efficiently handled computationally and programmatically, as working with matrices reduced computational demands. In this way, it was possible to generate an editable image.

Over the years, a large number of publications have focused primarily on sketch analysis and the vectorization of hand-drawn illustrations. For example, methods have been proposed for raster image vectorization based on simple lines and the detection of node points (Guo et al., 2019), as well as techniques for evaluating the quality of extracted paths within an image (Yan et al., 2020).

In the context of road analysis, it is then possible to present, for example (Chaudhuri et al., 2012). In this study, the authors focused on the extraction of roads, rivers, and buildings from satellite imagery using semi-automatic methods. Noise is reduced, followed by the extraction of the image skeleton. Similarly, (Wang et al., 2025) presented a method aimed at improving the quality of road segmentation from satellite images.

Satellite imagery is also used by (Dey and Aithal, 2024), who introduced more advanced methods employing machine learning and deep learning. The development of urban databases was supported, and contributions were made to sustainable transportation planning.

Vectorization of road markings was addressed by (Xia et al., 2024), who introduced a sophisticated machine learning approach for generating standardized vector map features, particularly for lane prediction. Similarly, advanced techniques were also presented by (Chen et al., 2024).

Feature identification and vectorization are also performed on point clouds, which, although inherently different from raster imagery, can serve as a valuable source of methodological inspiration. One such approach involves the analysis of RGB colour intensity (Gao et al., 2017). In the case of point clouds, feature identification can be further refined using laser return intensity or point density (Rastiveis et al., 2020). Additional methods for assessing point clouds include height-based analysis (Pu et al., 2011) or feature extraction within cross-sectional slices (Yan et al., 2016).

Identified features within point clouds may also be subjected to techniques such as the alpha shape method. In a later stage, shortest path algorithms can be applied to define object trajectories or envelopes, thus enabling subsequent vectorization (Prochazka et al., 2018).

Several publications can also be mentioned that deal more generally with image processing methods. These methods can be universally applied beyond the domain of road infrastructure. These include educational materials e.g., (Hájkovský et al., 2012). Similarly, there are also important algorithms that have not yet been mentioned, e.g., (Douglas and Peucker, 1973), which focuses on point reduction and curve simplification. Finally, bitmap adjustments can also be mentioned (Stanko et al., 2020; Shen et al., 2021).

Based on the reviewed literature, it was possible to understand the principles of various processing techniques and draw inspiration from them for the present work. This research differs

from existing approaches primarily in its direct adaptation to the generation of schematic site plans, which are applicable in transport practice. Moreover, it is specifically focused on inputs derived from orthographic images captured by UASs.

3. Object identification and vectorization method

The process of vectorizing elements in an image is designed in the MATLAB software environment and is divided into several modular parts, which are further divided by their own functions. It is possible to return to these cycles at any time and repeat the given part of the calculation if necessary. In general, the method can be summarized in several consecutive steps:

The first step involves defining the area of interest and identifying the relevant colour. Colour is the only visual indicator that users can reliably interpret from a standard raster image.

This is followed by steps focused on defining specific colour shades, as well as detecting and removing unwanted noise. Noise presents a major challenge in the process and tends to increase proportionally with broader tolerance ranges for the selected colour shades.

Subsequent steps involve refining selected regions of interest by clustering and grouping relevant points. After identifying complete objects, it is possible to proceed to the creation of envelopes and final vectorization of the image.

As the method is still being optimized to achieve the most accurate results for identifying road infrastructure features, minor changes to the current procedure may still occur.

To demonstrate the current functionality of the proposed process, horizontal road markings at an intersection will be identified. The orthophoto image of the examined location is shown in the following Figure 1. The image size is 1321×802 px (1,059,442 pixels).

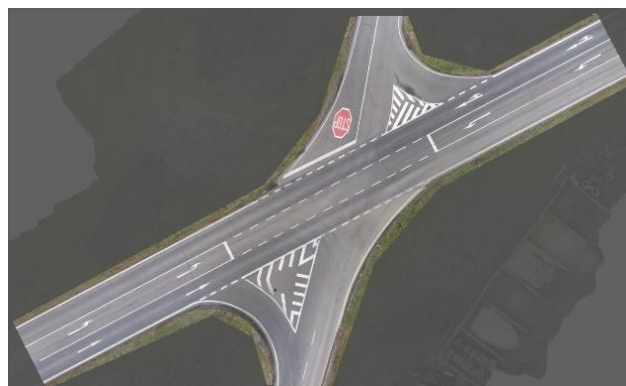


Figure 1. Orthophoto image of the evaluated site with a polygon marking the area of interest.

In the given image, RGB colours are identified. For each colour component, an appropriate tolerance range is defined to allow for slight variations in hue. Depending on the user's needs, this tolerance is set in such a way that it sufficiently covers the desired colour area while minimizing the generation of excessive noise. When identifying light colours, a common issue at high RGB tolerances is the inclusion of light gray shades, which need to be reduced. To enable efficient image processing, the input image is converted into a binary mask. Pixels falling within the user-defined colour range are assigned a value of 1, while the rest of the image is represented by 0. This process significantly simplifies the analysed image, as the bit depth is reduced from

the original 24 bits per pixel (RGB) to just 1 bit per pixel (Hájkovský et al., 2012).

Noise elimination can be addressed through several techniques. One approach involves the removal of small pixel clusters below a defined size threshold (in this specific case, clusters smaller than 50 pixels are eliminated). The pixels are removed or recalculated from a value of 1 to a value of 0 if the maximum size condition is met. At the same time, no connectivity is detected along diagonal or vertical axes with other previously identified pixels having a value of 1.

An inverse approach can also be used to fill holes in the mask. Areas with horizontal road markings often exhibit signs of wear due to frequent vehicle traffic or the presence of dirt on the road surface. Nevertheless, the underlying light colour, typically white, tends to dominate in these regions, and the corresponding pixels have mostly already been identified. The filling of unidentified areas is once again carried out if two conditions are met: the size of the hole is below a defined threshold and there is sufficient connectivity to the surrounding mask.

To further reduce noise in the binary mask and achieve more accurate object identification, morphological operations such as dilation and erosion can be applied (Said et al., 2021; Hájkovský et al., 2012). Morphological transformation can be imagined as the movement of a structural element (B) across the binary mask image (X). The structuring element represents a predefined shape. In this case, a circle with a defined radius was selected as the most suitable option. Dilation then represents the composition of two sets of points using vector addition. After dilation is applied, the objects in the image are enlarged by the size of the circle's radius. This operation helps fill gaps or smooth irregularities along the edges of previously segmented pixel clusters.

The formal definition of dilation can be expressed by the following equation (1):

$$X \oplus B = \{d \in E^2; d = x + b, x \in X, b \in B\} \quad X \oplus B = \bigcup_{b \in B} X_b \quad (1)$$

where B = structural element (circle)
 X = binary mask image
 E^2 = Euclidean space

Erosion represents the inverse operation. It is defined as the composition of two point sets using vector difference and is typically used to simplify the structure of an object. The relationship can be defined by the following equation (2):

$$X \ominus B = \{d \in E^2; d + b \in X, \forall b \in B\} \quad X \ominus B = \bigcap_{b \in B} X_{-b} \quad (2)$$

where B = structural element (circle)
 X = binary mask image
 E^2 = Euclidean space

By combining these two morphological operations, the closing operation is achieved (Jähne, 2005). This process makes it possible to smooth edges and eliminate small holes within individual clusters in the mask. If the size of the structuring element is the same, the final mask retains approximately the same overall dimensions as before the operations. However, irregularities along the edges and small gaps within pixel clusters are significantly reduced. The result after applying these operations can be seen in Figure 2.

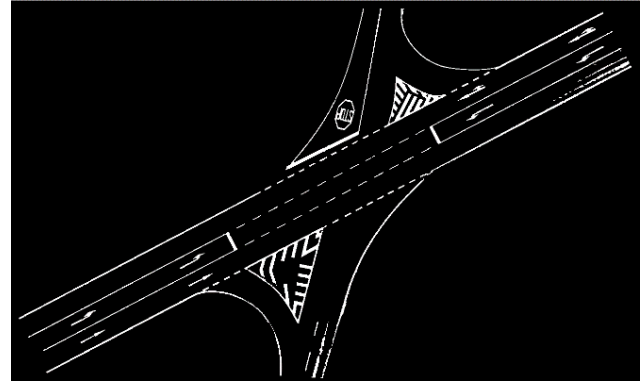


Figure 2. Modification of a binary mask using the closure method (dilation and erosion).

It can be seen that most of the noise has been successfully reduced. At this stage, objects are still not precisely identified in the image. The results are still a representation in the form of a binary mask. To transform the mask into individual, editable objects, the DBSCAN method was applied (Beer et al., 2023; Ester et al., 1996). DBSCAN is a non-parametric density-based clustering algorithm. The clustering process is based on the identification of core points within clusters (3). Core points are defined if there is a defined minimum number of neighbouring (non-core) points within a distance of ϵ around them. Each cluster then forms at least one core point. If non-core points are also present, they must be directly accessible from the core. The identified clusters can subsequently be optimized to minimize their number while still meeting the conditions defined by the method.

$$\min_{C \subset C, d_{db}(p, q) \leq \epsilon, \forall p, q \in C_i \forall C_i \in C} |C| \quad (3)$$

where C = set of clusters
 ϵ = threshold distance
 $d_{db}(p, q)$ = distance between core and non-core points

The resulting clustering can be seen in Figure 3.



Figure 3. Application of the DBSCAN method to a binary mask and identification of objects.

This method now allows individual clusters to be treated as separate objects. It is possible to manually remove excess objects that were not previously detected as noise. At the same time, individual clusters can be merged into a single entity. Due to the

absence of machine learning, this semi-automatic mode is necessary. Although the algorithm mathematically considers the clustered objects to be valid, they may not always make logical sense in context.

As part of the final vectorization of the image, two output options are available. If the emphasis is on the overall envelope of individual objects, it is possible to detect and smooth the boundaries of objects. If the user wants to work only with a simplified version, it is possible to skeletonize the image and work only with the unit skeleton of the object.

The first option is more suitable for complex use cases, as the outputs contain the complete envelope of each cluster. The envelope is easily identifiable, as it defines the transition between the object and the rest of the binary mask area. The detected edges are then smoothed using the Savitzky–Golay filter (Schafer, 2011). This digital filter is capable of smoothing object boundaries with minimal distortion and preserves corner points more effectively than, for example, a moving average filter. The smoothing process is based on convolution. Successive subsets of points along the object boundary are fitted with a low-degree polynomial, approximated using the method of linear least squares (Kumar, 2017). The definition of the method can be described by the following equation (4):

$$Y_j = \sum_{i=\frac{1-m}{2}}^{\frac{m-1}{2}} C_i Y_{j+i} \quad \frac{m+1}{2} \leq j \leq n - \frac{m-1}{2} \quad (4)$$

where
 Y_j = centre of the convolution window
 C = convolution coefficient
 m = window size

The second method for analysing objects is the application of the skeletonization process (Abu-Ain et al., 2013). This process reduces objects to a width of one pixel. The coordinates of the image skeleton are then recalculated into identified clusters. The principle of skeletonization is based on erosion followed by morphological opening (erosion followed by dilation) (Jähne, 2005). Its mathematical expression can be described by the following equation (5):

$$S(X) = \bigcup_{\rho>0} \bigcap_{\mu>0} [(X \ominus \rho B) - (X \ominus \rho B) \circ \mu \bar{B}] \quad (5)$$

where
 S = skeleton
 X = binary mask image
 B = structuring element

The skeletonized image can then be approximated using geometric primitives. At the current stage, the authors focused primarily on fitting line segments and arcs using the least squares approximation method (Kumar, 2017). This is based on the polynomial function defined in equation (6):

$$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + ax + a_0 \quad (6)$$

A first-degree polynomial is used for line detection (7):

$$P_1(x) = ax + a_0 \quad (7)$$

If the distribution of points within a skeletonized cluster suggests the presence of an arc, the approximation is performed using the equation of a circle (8). A segment of the arc is then extracted, defined by the endpoints of the given cluster.

$$(x - m)^2 + (x - n)^2 = r^2 \quad (8)$$

In the final stage, individual objects are vectorized and exported to the .dxf graphic format. During export, they are automatically divided into individual layers. Both the boundaries of individual clusters and the skeleton itself are exported. An example can be seen in Figure 4. The smoothed envelopes of individual objects are shown in white, and the unit skeleton is shown in red.

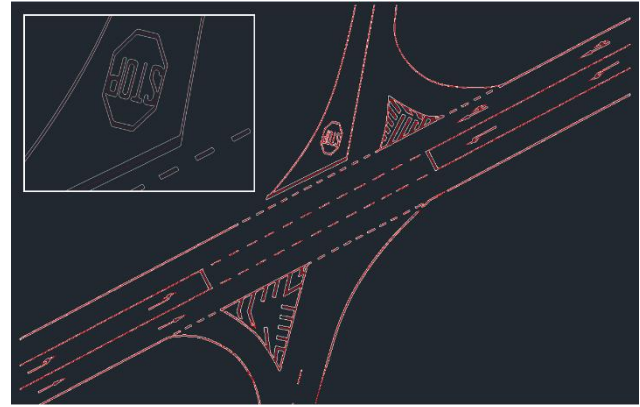


Figure 4. Image vectorization.

4. Comparison of automated methods with manual image processing

A comparison of the automated process and manual vectorization was performed on a total of 10 samples listed in Table 1. The test images varied in both size and level of detail. Additionally, the number of individual objects present differed across the images. The aim of this evaluation was to compare the time efficiency of manual and automated segmentation processes.

Both approaches were tested by all authors of this paper (number of tests performed – 4 x 2 x 10). The authors have experience in design, so it can be said that the efficiency of manual processing was performed at a high level. Predefined templates with individual horizontal road marking lines and directional arrow templates were also used in the manual work. It can therefore be said that the time required for manual vectorization sufficiently corresponds to the user's real-life practice.

Table 1 lists the overall average time spent by all authors on a specific image and type of processing.

Image	No.	Image size	No. of objects	Processing method	
		[px]		Manual [s]	Auto [s]
	1	193 x 276	1	225	13
	2	514 x 446	2	223	47
	3	1809 x 753	3	38	31

	4	1813 x 1375	8	214	41
	5	2561 x 1493	20	596	90
	6	2813 x 1605	14	398	43
	7	4381 x 2625	19	187	94
	8	4445 x 3069	37	1078	204
	9	12121 x 6737	62	1475	728
	10	17700 x 10350	86	1900	1649

Table 1. Comparison of automated and manual object vectorization processes.

The number of objects is always defined by a specific shape or line type. For example, if only the road line is analysed, three objects are defined in the form of left, right, and centre continuous lines.

The individual measured values are presented in Figures 5 and 6 below.

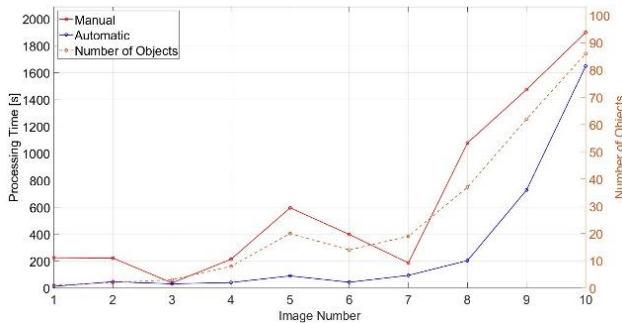


Figure 5. Dependency of automatic and manual processing on time and the number of objects in individual images.

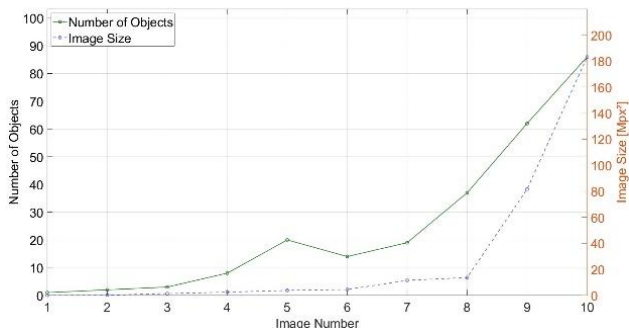


Figure 6. Dependency of image size and object count across the evaluated dataset.

Even though the table shows a clear difference between automated and manual processing, a more detailed statistical

analysis was performed. his analysis will also be useful in the further development and testing of the algorithm.

The measured values were evaluated using the statistical method of the paired t-test (Shier, 2004). It was assessed whether the average difference between these two methods of evaluation was statistically significant or whether it was a coincidence.

Two hypotheses were defined. The null hypothesis (H_0) states that there is no statistically significant difference between automated and manual image processing. The alternative hypothesis (H_1), on the other hand, states that manual processing is slower than automatic processing.

Mathematically, a one-sided paired t-test can be expressed as follows:

Two sets of data (X_1 , X_2) are obtained from n samples. Subsequently, the difference between the measured values z_i (9) is defined, as well as the mean $\bar{z}$ (10) and standard deviation s_z (11).

$$z_i = x_{1,i} - x_{2,i} \quad (9)$$

$$\bar{z} = \frac{1}{n} \sum_{i=1}^n z_i \quad (10)$$

$$s_z = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (z_i - \bar{z})^2} \quad (11)$$

The test criterion then corresponds to relation (12):

$$t = \frac{\bar{z}}{s_z} \sqrt{n} = -3,7696 \quad (12)$$

The definition of the critical region for the one-sided paired t-test is as follows (13):

$$K = \{t < t_{\alpha, n-1}\} \quad (13)$$

$$t_{0.05, 9} \approx -1,8331$$

where

$$\alpha = 0,05$$

n = count of images

For the given results, it is also possible to determine the p-value (14):

$$p = P(T \leq t) \quad T \sim t(n-1) \quad (14)$$

The resulting p-value is 0,0022.

These calculations confirmed the hypothesis that manual processing is slower than automatic processing. Another way to interpret the result is that the time required for processing using the automatic method is significantly faster than objects identified and vectorized manually.

In order to determine how many times faster automated processing is, it is possible to calculate the ratio (17) of their average times (15, 16).

$$\bar{x}_1 = \frac{1}{n} \sum_{i=1}^n x_{1,i} = 294 \text{ s} \quad (15)$$

$$\bar{x}_2 = \frac{1}{n} \sum_{i=1}^n x_{2,i} = 633.4 \text{ s} \quad (16)$$

$$R = \frac{\bar{x}_2}{\bar{x}_1} = \frac{633.4}{294} = 2.15 \quad (17)$$

The results for these specific cases indicate that automated processing and vectorization of selected road infrastructure objects is (in its current implementation) on average 2.15 times faster. It is evident that this value is influenced primarily by large images (e.g., Image no. 10) or images with simple shapes (e.g., straight lines in Image 3).

In realistic scenarios, analysis would more likely be conducted on images such as 5, 8, and 9, as their areas correspond more closely to the spatial extent of an intersection. These images clearly show the entire intersection area, including its arms. If the efficiency calculation is performed on these three images, the difference in speed between the automated process and the manual process is on average 3.1 times faster.

5. Conclusion

Based on the presented analysis, it can be concluded that the developed method clearly leads to a higher degree of efficiency related to image processing. Automated vectorization of selected road features can save the user 2 to 3 times more time compared to manual processing.

However, the current procedure needs to be further modified and optimized. One of the key areas for future development is more accurate identification of object boundaries. A future objective is the implementation of predefined envelopes (e.g., rectangles), which could be automatically fitted to features known to represent lines. This would help smooth the representation of linear elements and reduce the number of interest points.

Simplification will also be applied to curves that cannot be directly described using basic geometric shapes. In such cases, the use of methods like the Douglas–Peucker algorithm (Douglas and Peucker, 1973) is being considered. If this method is already applied to modified objects, it will be possible to simplify their envelope point by point without compromising accuracy.

Another essential criterion for improving the precision of object definition is the development of a method robust against shadows or curve discontinuities caused by factors such as vegetation above the roadway. The current design allows two or more objects to be merged into one, but this merger is not linked in a binary mask. Only groups are merged, not points recalculated or continuous lines created between these objects. However, the skeletonization method makes it possible to find a vector along which points can be added until the shapes connect with each other.

Lastly, the authors plan to evaluate the proposed method in comparison with more modern approaches such as machine learning or artificial intelligence. Testing will be conducted to assess the effectiveness of both approaches, with the potential for their integration in future developments.

Acknowledgements

This work has been funded by a grant from the Programme Johannes Amos Comenius under the Ministry of Education, Youth and Sports of the Czech Republic [CZ.02.01.01/00/23_021/0009003]. As set out in the Legal Act, beneficiaries must ensure that the open access to the published version or the final peer-reviewed manuscript accepted for publication is provided immediately after the date of publication via a trusted repository under the latest available version of the

Creative Commons Attribution International Public Licence (CC BY) or a licence with equivalent rights. For long-text formats, CC BY-NC, CC BY-ND, CC BY-NC-ND or equivalent licenses could be applied.

References

- Abu-Ain, W., Abdullah, S. N. H. S., Bataineh, B., Abu-Ain, T., & Omar, K. (2013). Skeletonization Algorithm for Binary Images. In *Procedia Technology* (Vol. 11, pp. 704–709). Elsevier BV. <https://doi.org/10.1016/j.protcy.2013.12.248>
- Beer, A., Draganov, A., Hohma, E., Jahn, P., Frey, C. M. M., & Assent, I. (2023). Connecting the Dots – Density-Connectivity Distance unifies DBSCAN, k-Center and Spectral Clustering. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 80–92). <https://doi.org/10.1145/3580305.3599283>
- Chen, P., Jiang, X., Zhang, Y., Tan, J., & Jiang, R. (2024). MapCVV: On-Cloud Map Construction Using Crowdsourcing Visual Vectorized Elements Towards Autonomous Driving. *IEEE Robotics and Automation Letters*, 9(6), 5735–5742. <https://doi.org/10.1109/lra.2024.3396096>
- DOUGLAS, D. H., & PEUCKER, T. K. (1973). ALGORITHMS FOR THE REDUCTION OF THE NUMBER OF POINTS REQUIRED TO REPRESENT A DIGITIZED LINE OR ITS CARICATURE. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2), 112–122. <https://doi.org/10.3138/fm57-6770-u75u-7727>
- Dey, & Aithal, B. H. (2024). Mapping urban road networks using semantic approach. In *IOP Conference Series: Earth and Environmental Science*, 1412(1), 012033. <https://doi.org/10.1088/1755-1315/1412/1/012033>
- Ester, M., Kriegel, H., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96)*, 226–231.
- Gao, Y., Zhong, R., Tang, T., Wang, L., & Liu, X. (2017). Automatic extraction of pavement markings on streets from point cloud data of mobile LiDAR. *Measurement Science and Technology*, 28(8), 085203. <https://doi.org/10.1088/1361-6501/aa76a3>
- Guo, Y., Zhang, Z., Han, C., Hu, W., Li, C., & Wong, T. (2019). Deep Line Drawing Vectorization via Line Subdivision and Topology Reconstruction. *Computer Graphics Forum*, 38(7), 81–90. <https://doi.org/10.1111/cgf.13818>
- Hájkovský, R., Pustková, R., Kutálek, F. (2012). Image processing in measurement and control technology: Textbook. Technical University of Ostrava. ISBN 978-80-248-2596-0.
- Jähne, B. (2005). Digital Image Processing. Springer Science & Business Media.
- Janssen, R. D. T., & Vossepoel, A. M. (1997). Adaptive Vectorization of Line Drawing Images. *Computer Vision and Image Understanding*, 65(1), 38–56. <https://doi.org/10.1006/cviu.1996.0484>

- Knyaz, V. A., & Chibunichev, A. G. (2016). Photogrammetric Techniques for Road Surface Analysis. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLI-B5, XXIII ISPRS Congress, Prague. <https://doi.org/10.5194/isprs-archives-xlvi-5-w2-2023-111-2023>
- Kumar, M., Rao, G., Shah, A., Davergave, N. (2017). Statistical Techniques for Transportation Engineering (1st ed.). Butterworth-Heinemann, USA.
- McGlone, J., Mikhail, E. M., Bethel, J. S., Mullen, R. (2004). Manual of Photogrammetry. 5th ed. American Society for Photogrammetry and Remote Sensing. ISBN 15-708-3071-1.
- Pinto, L., Bianchini, F., Nova, V., Passoni, D. (2020). Low-Cost UAS Photogrammetry For Road Infrastructures' Inspection. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B2-2020, 1145–1150. <https://doi.org/10.5194/isprs-archives-XLIII-B2-2020-1145-2020>
- Prochazka, D., Prochazkova, J., & Landa, J. (2018). Automatic lane marking extraction from point cloud into polygon map layer. *European Journal of Remote Sensing*, 52(sup1), 26–39. <https://doi.org/10.1080/22797254.2018.1535837>
- Pu, S., Rutzinger, M., Vosselman, G., & Oude Elberink, S. (2011). Recognizing basic structures from mobile laser scanning data for road inventory studies. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(6), S28–S39. <https://doi.org/10.1016/j.isprsjprs.2011.08.006>
- Rastiveis, H., Shams, A., Sarasua, W. A., & Li, J. (2020). Automated extraction of lane markings from mobile LiDAR point clouds based on fuzzy inference. *ISPRS Journal of Photogrammetry and Remote Sensing*, 160, 149–166. <https://doi.org/10.1016/j.isprsjprs.2019.12.009>
- Said, K. A. M., & Jambek, A. B. (2021). Analysis of Image Processing Using Morphological Erosion and Dilation. *Journal of Physics: Conference Series*, 2071(1), 012033. <https://doi.org/10.1088/1742-6596/2071/1/012033>
- Schafer, R. (2011). What Is a Savitzky-Golay Filter? *IEEE Signal Processing Magazine*, 28(4), 111–117. <https://doi.org/10.1109/msp.2011.941097>
- Shen, R., Tang, B., Liberti, L., D'Ambrosio, C., & Canu, S. (2021). Learning discontinuous piecewise affine fitting functions using mixed integer programming over lattice. *Journal of Global Optimization*, 81(1), 85–108. <https://doi.org/10.1007/s10898-021-01034-x>
- Shier, R. (2004). Statistics: 1.1 Paired t-tests. Mathematics Support Centre, Loughborough University. <https://www.statstutor.ac.uk/resources/uploaded/paired-t-test.pdf>
- Stanko, T., Bessmeltsev, M., Bommers, D., & Bousseau, A. (2020). Integer-Grid Sketch Simplification and Vectorization. *Computer Graphics Forum*, 39(5), 149–161. <https://doi.org/10.1111/cgf.14075>
- Svatý, Z., Vrtal, P., Kouhout, T., & Nouzovský, L. (2023). Automated Detection and Vectorization of Road Elements in High Resolution Orthographic Images. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVIII-5/W2-2023, 111–116. <https://doi.org/10.5194/isprs-archives-xlvi-5-w2-2023-111-2023>
- Xia, D., Zhang, W., Liu, X., Zhang, W., Gong, C., Huang, J., Yang, M., & Yang, D. (2024). DuMapNet: An End-to-End Vectorization System for City-Scale Lane-Level Map Generation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 6015–6024). <https://doi.org/10.1145/3637528.3671579>
- Yan, C., Vanderhaeghe, D., & Gingold, Y. (2020). A benchmark for rough sketch cleanup. *ACM Transactions on Graphics*, 39(6), 1–14. <https://doi.org/10.1145/3414685.3417784>
- Yan, L., Liu, H., Tan, J., Li, Z., Xie, H., & Chen, C. (2016). Scan Line Based Road Marking Extraction from Mobile LiDAR Point Clouds. *Sensors*, 16(6), 903. <https://doi.org/10.3390/s16060903>
- Zhang, T. Y., & Suen, C. Y. (1984). A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 27(3), 236–239. <https://doi.org/10.1145/357994.358023>