LiLoc – Defect Image Localization in Laser Scans

Patrick Herbers[1], Lisa Freifrau von Rössing[1], and Markus König[1]

[1]Chair of Computing in Engineering, Ruhr University Bochum, Bochum, Germany

patrick.herbers@rub.de, lisa.freiinvonroessing@rub.de, koenig@inf.bi.rub.de

Abstract -

Deterioration and maintenance are a constant part of the operational phase of a structures life cycle. Managing and analyzing a structure's history of damage and repair is essential for present operational decisions. Images are a central way of documenting the location of damages during maintenance, but often lack references to the structure. Mapping defects not only spatially, but also over time, can give inspectors a deep insight into the health of a structure. Therefore, we present *LiLoc*, a module for localizing images of inspection documentation in laser scans. Images are matched to the panoramic images of a laser scan, and matrix transformations for mapping points to laser scans are determined. Extracted matrices can be used to transform damage masks and other overlaying information. We then compare the algorithms on two bridge data sets.

Keywords -

Localization, Maintenance, Laser Scanning, Inspection, Defect Mapping

1 Introduction

The maintenance effort required for infrastructure is at an unprecedented high with many concrete structures in Europe and North America reaching the end of their life span. Among the various activities involved in infrastructure upkeep, inspections stand out as the most critical component for assessing the health and integrity of structures. This is especially true for concrete bridges, for which regular inspections throughout their lifespan are an absolute necessity for public safety.

Currently, the inspections in Germany are in most cases conducted by traversing the bridge, identifying old and new defects, documenting the bridge's current state with images and written notes, and finally summarizing the inspection results in a standardized report, consisting of text and images. The images document defects visually, and are used for referencing the defects in later inspections to enable the evaluation of defect progression. Textual information is used to further describe the defect, such as material and size. Usually, the textual information also describes the location of the defect, such as the building component identifier, and less often, road kilometer or cardinal direction. Due to a lack of detail, this recorded location is often imprecise. Finding the defect again is therefore difficult, even though it is required as a basis for the analysis of the structural condition and the defect progression. Enhancing the largely analog and inefficient inspection process through digital tools enables earlier detection of defects and defect progression, thereby providing a stronger foundation for making informed operational decisions.

One possible solution is creating a defect map, meaning a module for a Digital Twin (DT) of the structure that stores defect information and location over time. The planning and execution of maintenance measures for bridge structures benefits from the concept of DTs. Inspectors can easily locate defects with the defect map in a 3D model, while operators can efficiently analyze large amounts of data on a holistic level. Especially the comprehensive map of all defects on a structure over time improves the informational value of inspections by also giving exact spatial information. This defect map will save time during inspections and allow operators to make more informed decisions during the life cycle of a bridge. While a lot of data is available with many inspections reaching back decades, mapping defects from older or current inspections manually is however inefficient.

The major hurdle to overcome for automatically creating a defect map is transforming the data from existing databases. Since the location of defects is often only documented in inconsistent and imprecise textual descriptions, automatically placing a defect on a 3D model becomes challenging. Inspection images may contain hints to the location of a defect, but often lack context, consisting of close ups and zoomed-in pictures for a detailed documentation of defects. Thus, a way to localize images from historical databases is an essential step for the creation of a defect map.

Furthermore, a 3D model into which defects can be mapped is required. Capturing the as-built state of a structure by creating a digital model using laser scanning has been used in practice for several years. As laser scanning allows for the efficient generation of an as-built model, and images of defects are readily available, combining the two methods enables comprehensive defect mapping in conjunction with detailed visual and textual information.

Therefore, we propose an approach to map defects using one-time laser scans and images from multiple inspections through feature matching.

2 Related Works

Improving the mapping of defects is an important topic in structural health research, with a focus on accurately and automatically localizing defects in a 3D model. An often-used tool for concurrent 3D model creation and defect mapping are Unmanned Aerial Systems (UAS). For example, Lin et al. [1] use an UAS to take images of a bridge. The images are used to create a 3D model with Structure from Motion (SfM), and therefore first clustered based on GPS coordinates for to prevent mismatches due to the inherent self-similarity of bridge structures. Defects are then detected on the same images. Since the location of the latter within the 3D model is known, the detected defects can be mapped into the model easily. However, extensive flight planning is necessary to account for GPS inaccuracy. A similar approach is used by Tan et al. [2] for external building walls. Here, GPS is used to compute the UAS position, which is then used to map defects onto a Building Information Modelling (BIM) model.

Research has shown that localization with GPS can lack the accuracy required for defect mapping, especially in areas with little reception, such as underneath a bridge. Therefore, additional technology has been tested to increase localization and thereby mapping accuracy. Yoon et al. [3] utilize an UAS with LiDAR, GPS, and IMU to achieve an accuracy of 10 cm even underneath a bridge. The system is however only evaluated on flat surfaces, such as one side of a bridge column, and issues due to GPS inaccuracy are reported, such as image curving, distortion, and areas missed by the UAS. In another example, Zhao et al. [4] use an UAS with LiDAR, Ground Control Points (GCP) and Check Points (CP) to take images of a concrete dam. Next, cracks larger than 5 mm are detected and mapped with Scale-Invariant Feature Transform (SIFT) [5] and SfM. However, the authors note that the GCPs are critically necessary for reliably mapping the defects, but the setup of GCP and CP is time-intensive and therefore inefficient.

Alternative, non-GPS methods for mapping defects have been tested, mainly relying on camera data. Simultaneous localization and mapping (SLAM), specifically ORB-SLAM, is used on images taken by an UAS to localize defects underneath a longer bridge structure by Jiang et al. [6]. The images were taken from a distance of 2 m and the method results in a localization error of around 25 cm. Bartczak et al. [7] use images taken by an UAS for photogrammetric reconstruction of a 3D model by employing sequential image pairing, explicitly foregoing GPS due to inaccuracies. Defects are detected in the UAS images and thereby mapped to the resulting 3D model.

The previous literature shows the general suitability of using UAS for defect mapping. If an image is used for 3D model creation and defect detection concurrently, the defects are mapped into the resulting model automatically. However, UAS rely on GPS or other methods to localize themselves and thereby images. The previously presented studies show that these methods are still time- and/or cost-intensive, and prone to localization errors. Additionally, although the research is promising, there are further issues with the usage of UAS for defect mapping. First, UAS require a trained user for utilization, which adds further costs to the inspection process, as they cannot replace inspectors. UAS have to keep a safe distance to walls, which decreases the defect resolution in an image. In Xiao et al. [8] for example, close-up shots were taken, but could not be used for model generation successfully and were therefore discarded, leaving their data unavailable for defect detection. The detection of fine cracks with a width of 0.1 mm is required during inspections. Second, usage of UAS is highly regulated and can be restricted in environments such as the vicinity of critical infrastructure, which highway bridges are part of.

In summary, while the most common method of 3D model creation and defect mapping for bridges with UAS does have accuracy issues with regards to localization and resolution, it's not an unsuitable method, and it's apparent that it will play a major role in future inspection processes. However, due to usage restrictions and efficiency considerations, other methods of defect mapping are necessary. A simple alternative for 3D model creation are laser scans with portable scanners, which require little training and are explicitly encouraged in German inspection regulations [9]. Laser scanners usually record either panoramic or stitched images of their scanning pose to capture color information in the point cloud. While panoramic images can be taken during laser scans and are mapped to the scans point cloud, the images are not suitable for defect detection. Therefore, scans are usually not used for defect mapping.

Both laser scanning and UAS are used for as-built model creation but only consider the current state of the structure. A wealth of historical defect images exists, and new images are taken regularly during in-person inspections. Automatically mapping this defect data into a laser scan fulfills the need for a defect map. It also offers additional benefits, as the historical data is very valuable during inspections, e.g., to assess defect progression.

This historical data is not considered by any of the studies described previously, but the textual location descriptions have been considered the two following studies. Göbels and Beetz [10] localize defects via text from bridge defect inspection reports with an ontology called RELOC.

The localization is accurate to the structural component and the directions left/right/top/bottom, with the level of detail corresponding to the detail of the inspectors notes. Heise et al. [11] show the importance of the spatial location of defects in reports, and propose a linked data approach to road and bridge infrastructure data. Through mapping defect location between cross-domain systems, such as road and bridge databases with differing spatial reference systems, defect information can be referenced into other systems and historical data can be integrated. It is obvious that neither of these two works consider image data, which are not as prone to human error as text-based descriptions, or achieve an exact localization. As mentioned before, none of the other works discussed earlier consider historical data at all. Therefore, we focus on efficiently and accurately localizing current and historical defect images in present laser scans.

Since the historical data consists predominantly of individual images, image registration is a possible solution for mapping this data to the panoramic images in the laser scans. Image registration is the process of aligning two images of the same subject with different camera angles, resulting in a homography. One approach for this is key point based image matching. SIFT and adjacent methods extract comparable key points in both images, which are then used for image alignment. SIFT has previously been used for image retrieval in panoramic images, for example by Zamir and Shah [12].

In recent years, image matching using neural networks has become a dominant area of research. Neural network based key point extractors and matching methods, such as XFeat [13] and LightGlue [14] show a significant improvement in speed and accuracy over previous methods. A common research problem is image matching for panoramic images, such as street scenes or from laser scanners. Orhan and Bastanlar [15] use Convolutional Neural Networks (CNNs) to generate searchable feature sets on panoramic images by using a sliding window approach instead of rectifying multiple images. Kendall et al. [16] use a CNN for 6-DoF camera pose regression on a learned environment.

Image registration can be applied to all kinds of images. It is not limited to matching defect images with panoramic images, but can also map a historical image to a current image taken from a different angle. This is especially relevant considering UAS-image-based 3D models, into which in the future, historical defects can also be mapped.

3 LiLoc

We present *LiLoc*, a Python module for localizing images in LiDAR scans with the explicit purpose of creating a defect map for structures. The module is made up of three stages: *Image Extraction*, *Image Matching*, and *Projection*.

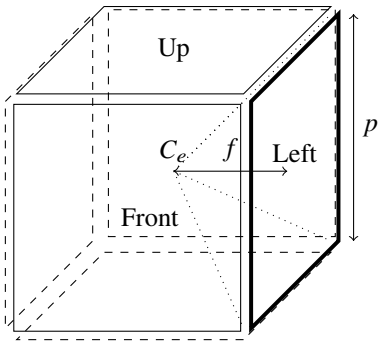

Figure 1. Cube configuration of the laser scan images. Every global camera pose C_e is associated with six images (Up, Down, Left, Right, Front, Back). f is the focal length, p is the image width.

3.1 Image Extraction

First, laser scan images and poses are extracted from a LiDAR scan. As input, we assume point clouds in the open general-purpose *E57* format [17] with multiple scan locations and RGB images. For best results, the scan locations should already be geo-referenced or registered among each other.

The images are extracted as rectified images in a cube configuration (see Figure 1). Every image is saved with an associated pose frame as a transformation matrix with translation and rotation. Pose frames are exported as JSON files with references to the associated JPEG file. We also attempt to extract camera information, such as focal length, from the *E57* file.

3.2 Image Matching

Image matching is done by first identifying visually distinct keypoints in defect and scan images. *LiLoc* supports keypoint detection using Scale-Invariant Feature Transform (SIFT), as well as XFeat [13] for finding keypoints with significant features. XFeat uses a convolutional neural network for finding features, which provides faster and more accurate keypoint detection than older methods. While SIFT produces a variable amount of keypoints, XFeat produces a maximum amount of 4096 keypoints, which are filtered by keypoint quality. All keypoints are saved per image and can be cached to disk for later reuse.

The second stage of image matching is finding keypoint matches. For image matching, we provide two matching methods: Exhaustive matching and cross matching. Exhaustive matching matches every image with every other

image, such as to find matches between all images of an inspection, or a defect instance, if one is specified. Cross matching matches every image of a set A with every image of a set B. This is useful for finding matches between laser scan images and inspection images, or between different capture dates.

Supported matching algorithms are k-Nearest-Neighbour (kNN) and LightGlue [14], a learned feature matching algorithm using deep neural networks. Both matching algorithms return a set of keypoint matches.

If the number of matches exceeds a threshold value, *LiLoc* attempts to find a homography between image pairs using OpenCV. This results in a perspective transformation matrix T. Results are written out in JSON format.

3.3 Projection

In the case of laser scan matching, we can use the match transform T to project points from the defect image onto the laser scan or a 3D model. For this, the intrinsic and extrinsic camera matrix (C_i and C_e respectively) are required for the transformation. The extrinsic camera matrix can be constructed from the position of the scanner and the direction of the scan image. The formula for the intrinsic camera matrix is as follows:

$$C_i = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

We assume a pinhole camera model with a cube projection arrangement using square images with resolution P for width and height. Due to the cube arrangement, the focal length is equal to the half the image width/height $f_x = f_y = P/2$, resulting in a 90° field of view. Since the camera center is also the image center, it also follows $c_x = c_y = P/2$. Thus, we can simply write the intrinsic camera matrix as:

$$C_i = \begin{bmatrix} P/2 & 0 & P/2 \\ 0 & P/2 & P/2 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

The extrinsic matrix $C_e = (R|t)$ encodes rotation R and translation t of the camera. Combining the extrinsic and intrinsic coordinates, we can obtain a directional vector $\vec{r}$ from an image coordinate $\vec{p}$.

$$\vec{r} = C_e C_i \vec{p} \tag{3}$$

The image coordinate is equal to the image center for now, but can be expanded when the defect position in the image is given. From position t towards direction $\vec{r}$, we use a cylinder cast (analogous to a ray cast) with a radius of 10 cm to find the closest point in the laser scan. We use the median of the closest 10 points to reduce the impact of outliers. This results in a 3D position for the defect image.

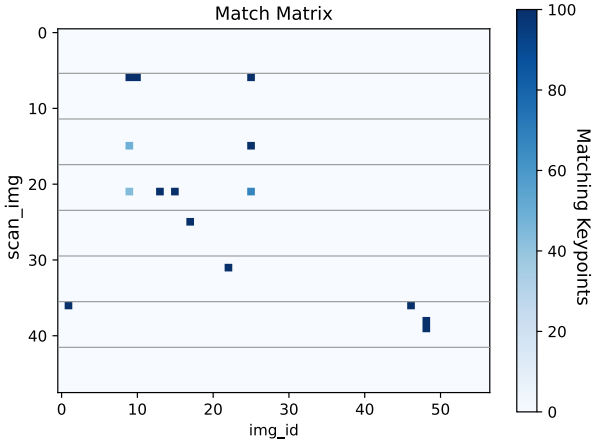

Figure 2. Match matrix of 57 defect images of to 8×6 laser scan pose images of Bridge A (scan-to-defect) using XFeat + LightGlue.

4 Case Study Results

Our approach was tested on two case studies consisting of two different highway bridges. For the lasers scans, a Leica RTC 360 was used, and photos were taken with Android and iOS smartphones.

Bridge A is a concrete bridge with a highway on-ramp. A full laser scan with 8 poses was taken, and images of concrete-based defects were collected on the same day. The individual scans were combined to a point cloud, and the point cloud was then registered to a pre-existing BIM model. For Bridge B is a 600 m pre-stressed concrete bridge. Partial laser scans were taken at five poses and 504 pictures of present defects in different camera angles were collected.

For Bridge A, we compare SIFT + kNN-Matching with XFeat + LightGlue. We only focus on XFeat + LightGlue for Bridge B.

4.1 Scan matching

For localization of the defect images in the laser scans, we attempt a match of every defect image with every scan image. For this, we extracted the scan images as described in Section 3.1. The matching threshold for all tests was set to 30 matches.

For this test we use XFeat as the keypoint extractor and LightGlue as the keypoint matcher. Bridge A with 8 scan locations (resulting in 48 rectified scan images) and 57 defect images results in 15 matches (see Figure 2). 10 images were correctly matched to at least one scan image, allowing for localization in the point cloud. Even with limited context, the matching performed well. Figure 3

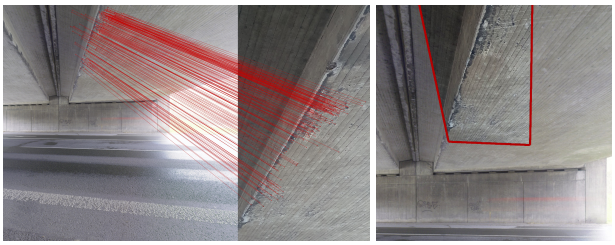

Figure 3. Example of a positive match between a panoramic image (left) and a defect documentation image (middle). The defect image is projected onto the scan pose (right).

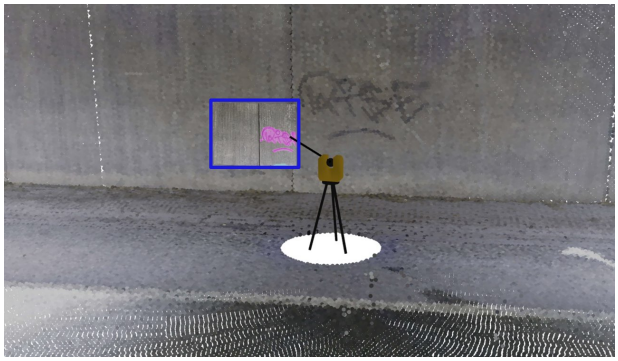

Figure 4. Projection of a defect image of graffiti on the concrete wall below Bridge A.

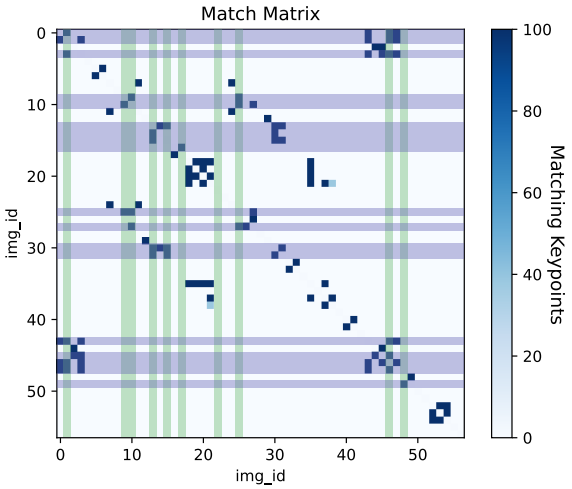

Figure 5. Match matrix of 57 defect images of Bridge A (defect-to-defect). Green lines show images that could be localized to a scan position (10 total). Blue lines include images that match to at least one localized image (18 total).

shows an example of spalling on the ceiling of the highway bridge. The detailed image of the defect was correctly matched to the scan image. Images that could not be matched to any scan image consist of extreme close-ups or of areas that the scanner did not cover. No false positives occurred in this limited data set.

We tested the projection capabilities with example images. Figure 4 shows the point cloud below the bridge. From the laser scan position, the matching defect image was projected onto the nearby wall. This links the defect to the point cloud with a 3D position.

As a baseline comparison, this data set was tested with SIFT features and kNN-matching. This resulted in 67 matches. Only four of the matches were correct (6% precision). Surprisingly, these four matches were matches that XFeat and LightGlue did not recognize. The low precision shows that classic keypoint matching methods are not suitable for finding matches in larger databases.

Bridge B with 9 scan positions (54 scan images) and 504 defect images resulted in 17 matches. We assume that the low number of matches is due to the limited scan coverage of the 600 m bridge. Again, no false positives were found. Notable is that most matches contain text written with chalk on the structure, like demarcations and measurements conducted by workers. Every matched defect image also corresponds to only one scan position, since the laser scan was limited to only significant locations with almost no overlap.

4.2 Defect matching

We used the exhaustive matching method for matching all defect images. This $n \times n$ matching allows for finding images of the same defect in the database.

Bridge A amounts to 58 matches in total (see Figure 5). Again, no false positives were found in this limited data set. Considering the scan matches from Section 4.1, we can subsequently localize overlapping and zoomed images of the same defect. 18 images are first removed to a direct scan match, and 22 images can be traced to at least one scan position in total. Some sets of images form a cluster, where 4–5 images of the same defect could be localized among each other, but could not be matched to a scan position.

For Bridge B we matched 504 images against each other. Overall, 904 matches were detected (see Figure 6). Since the image data set had up to ten pictures of the same defect from different perspectives, the matching clusters are larger than in Bridge A. Even though images were sometimes taken at extreme perspectives, the images were matched correctly (see Figure 7).

171 matches for Bridge B were manually classified as false positives (81% precision). The largest portion of false positives (120) turned out to be images with a crack

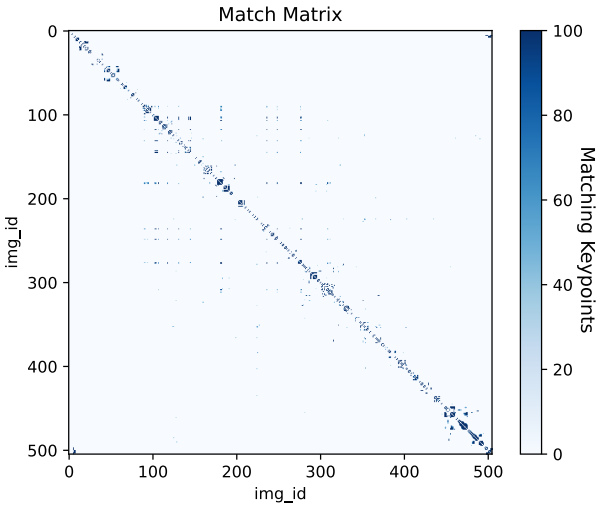

Figure 6. Match matrix of 504 defect images of Bridge B (defect-to-defect).

Figure 7. Example of a positive match of two images of the same defect, one with context, one with detail.

Figure 8. False positives for defect matching. Top: A crack width reference card is recognized as a match. Bottom: A similar scaffolding element is recognized as a match.

width reference card, a tool often used by inspectors to measure cracks (see Figure 8). The other false positives were mainly due to scaffolding present at the bridge, which shows a high degree of self-similarity. Increasing the keypoint match threshold to 100 reduces the number of total matches to 667 and the false positives to 47 (93% precision).

Tracing all images of Bridge B to at least one scan position allows localizing more defects. Filtering for false positives, 92 images can be traced to at least one scan position, allowing for localization in the point cloud.

4.3 Discussion

Using XFeat and LightGlue for matching shows significantly better results than using SIFT and kNN matching. SIFT and others have been developed for image stitching or live image tracking, such as SLAM, instead of recalling images taken at separate times with different cameras. While still useful in some of these situations (e. g. loop closure), matching entire databases produces too much false positives to be useful. Meanwhile, AI matching methods prove to be sensitive enough to detect significant points even in uniform concrete surfaces.

While only a small portion of defect images could be matched to scan images in our data set directly, exhaustive matching could fill some gaps. Using exhaustive matching to detect defect image clusters allowed for tracing these clusters to at least one scan image. Despite this, some images still could not be traced to a scan position. This is possibly due to incomplete laser scans or defect images that did not provide enough context for a successful localization.

False positives were mostly due to similar objects in the images, such as scaffolding or instruments. Still, some false positives are due to the self-similar nature of bridges. We have also neglected counting false negatives, as defining a false negative in terms of localization can be difficult.

5 Conclusion

We presented *LiLoc*, a tool for localizing defect images in laser scans, and tested it on two self-collected data sets. The tool supports extracting scan images from point clouds, matching images in exhaustive and cross-matching mode, and mapping defect images to a 3D coordinate. The code is documented on GitHub[1].

This tool paves the way for researchers, engineers and on-site inspectors utilizing the historical data of inspections to make maintenance easier and more comprehensible. With this image matching tool, historical data can be used in conjunction with modern digital tools. This approach can be integrated into existing inspection processes as historical data can be linked to new digital technologies. Databases can be enriched with location specific data, closing the gap between image data, laser scans, and DTs.

The results show that with AI keypoint extraction, reliable image localization is achievable. Learned features perform better on concrete structures than classic methods such as SIFT. Even difficult matches of pure concrete walls can reliably be matched based on the imperfections in the concrete. This has previously been a challenge with classic methods.

Figure 9. Example of a match between two images of the same defect, taken in 2005 and 2014 respectively.

Although the results are promising, some limitations are evident. Some objects or artifacts in the image can confuse the matching algorithm, such as measurement instruments or burnt in text on the image (such as the capture date). One option would be to detect these objects using a semantic segmentation network filter out keypoints with a mask. Lighting conditions are also a significant impact factor. In this paper, we have not compared large-scale data sets with images under different lighting conditions or camera modes (e. g. flash photography). The data set should be expanded upon for realistic applications. In this study, we have also only quantified the method's performance with data that was recorded concurrently. Figure 9 shows an example of a successful match of real historical inspection data with a time difference of nine years. Due to the difficulty of obtaining historical data sets in a large scale, this study focused on data captured without a large time difference. In future research, the data set should be expanded with more data recorded at different times and conditions. Crafting data sets that reflect these challenges is a necessary step in future work.

Future research will focus on improving the algorithms used in this paper, and adding new keypoint and matching algorithms. Both XFeat and LightGlue allow for training the network to adjust to specific conditions. Training these algorithms specifically for the materials used in the construction industry may improve precision significantly. Removing keypoints on equipment, as well as other camera artifacts such as burnt-in dates may be masked out in a pre-processing step. Other local feature extractors such as GlueStick [18] work with lines in addition to points, which may prove suitable for the built environment.

[1] https://github.com/RUB-Informatik-im-Bauwesen/LiLoc

Acknowledgment

This research was funded by the mFUND research programme of the Federal Ministry for Digital and Transport (BMDV) (funding code: 01F2259A). This paper is based on parts of the research project carried out at the request of the BMDV, represented by the Federal Highway Research Institute, under research project No. 69.0017/2023. The authors are solely responsible for the content.

References

[1] Jacob J. Lin, Amir Ibrahim, Shubham Sarwade, and Mani Golparvar-Fard. Bridge Inspection with Aerial Robots: Automating the Entire Pipeline of Visual Data Capture, 3D Mapping, Defect Detection, Analysis, and Reporting. *Journal of Computing in Civil Engineering*, 35(2):04020064, March 2021. ISSN 1943-5487. doi:10.1061/(ASCE)CP.1943-5487.0000954.

[2] Yi Tan, Geng Li, Ruying Cai, Jun Ma, and Mingzhu Wang. Mapping and modelling defect data from UAV captured images to BIM for building external wall inspection. *Automation in Construction*, 139:104284, July 2022. ISSN 0926-5805. doi:10.1016/j.autcon.2022.104284.

[3] Sungsik Yoon, Gi-Hun Gwon, Jin-Hwan Lee, and Hyung-Jo Jung. Three-dimensional im-

age coordinate-based missing region of interest area detection and damage localization for bridge visual inspection using unmanned aerial vehicles. *Structural Health Monitoring*, 20(4):1462–1475, July 2021. ISSN 1475-9217, 1741-3168. doi:10.1177/1475921720918675.

[4] Sizeng Zhao, Fei Kang, and Junjie Li. Concrete dam damage detection and localisation based on YOLOv5s-HSC and photogrammetric 3D reconstruction. *Automation in Construction*, 143: 104555, November 2022. ISSN 09265805. doi:10.1016/j.autcon.2022.104555.

[5] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2): 91–110, November 2004. ISSN 1573-1405. doi:10.1023/B:VISI.0000029664.99615.94.

[6] Shang Jiang, Yuyao Cheng, and Jian Zhang. Vision-guided unmanned aerial system for rapid multiple-type damage detection and localization. *Structural Health Monitoring*, 22(1): 319–337, January 2023. ISSN 1475-9217. doi:10.1177/14759217221084878.

[7] E. T. Bartczak, M. Bassier, and M. Vergauwen. RTK-Based Photogrammetric Reconstruction and Spatial Filtering. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVIII-1/W2-2023:1873–1880, December 2023. ISSN 2194-9034. doi:10.5194/isprs-archives-XLVIII-1-W2-2023-1873-2023.

[8] Jing-Lin Xiao, Jian-Sheng Fan, Yu-Fei Liu, Bao-Luo Li, and Jian-Guo Nie. Region of interest (ROI) extraction and crack detection for UAV-based bridge inspection using point cloud segmentation and 3D-to-2D projection. *Automation in Construction*, 158:105226, February 2024. ISSN 09265805. doi:10.1016/j.autcon.2023.105226.

[9] Bundesministerium für Verkehr und digitale Infrastruktur. Richtlinie zur einheitlichen Erfassung, Bewertung, Aufzeichnung und Auswertung von Ergebnissen der Bauwerksprüfungen nach DIN 1076 (RI-EBW-PRÜF), 2017-02-22, 2017. URL `https://www.bast.de/DE/Publikationen/Regelwerke/Ingenieurbau/Erhaltung/RI-EBW-PRUEF-Erhaltung.html`.

[10] Anne Göbels and Jakob Beetz. Relative Location Ontology: An ontological model for representing directional topological relationships between spatial entities in oriented space. In *Proceedings of the 2nd Linked Data in Architecture and Construction Workshop 2024*, Bochum, Germany, June 2024.

[11] Ina Heise, Anne Göbels, Andre Borrmann, and Jakob Beetz. Enabling Comprehensive Querying of Road and Civil Structure Data using Graph-based Methods. In *Proceedings of the 41st International Conference of CIB W78*, Marakesh, Morrocco, October 2024.

[12] Amir Roshan Zamir and Mubarak Shah. Accurate Image Localization Based on Google Maps Street View. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision – ECCV 2010*, Lecture Notes in Computer Science, pages 255–268, Berlin, Heidelberg, 2010. Springer. ISBN 978-3-642-15561-1. doi:10.1007/978-3-642-15561-1_19.

[13] Guilherme Potje, Felipe Cadar, Andre Araujo, Renato Martins, and Erickson R. Nascimento. XFeat: Accelerated features for lightweight image matching. In *2024 IEEE / CVF Computer Vision and Pattern Recognition (CVPR)*, 2024.

[14] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local feature matching at light speed. In *International Conference on Computer Vision (ICCV)*, 2023.

[15] Semih Orhan and Yalin Bastanlar. Efficient Search in a Panoramic Image Database for Long-term Visual Localization. In *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 1727–1734, Montreal, BC, Canada, October 2021. IEEE. ISBN 978-1-6654-0191-3. doi:10.1109/ICCVW54120.2021.00198.

[16] Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2938–2946, Santiago, Chile, December 2015. IEEE. ISBN 978-1-4673-8391-2. doi:10.1109/ICCV.2015.336.

[17] Daniel Huber. The ASTM E57 file format for 3D imaging data exchange. In *Proceedings of SPIE Electronics Imaging Science and Technology Conference (IS&T), 3D Imaging Metrology*, volume 7864, 2011-01, 2011.

[18] Rémi Pautrat, Iago Suárez, Yifan Yu, Marc Pollefeys, and Viktor Larsson. GlueStick: Robust image matching by sticking points and lines together. In *International Conference on Computer Vision (ICCV)*, 2023.