Image-based Localization of Bridge Defects with AR Visualization

Liu Hong, Shayegan Shakeri Nasab, Majdi Hassan, Andre Gou
University of Illinois at Urbana-Champaign
{liuhong2, shakeri3, mhassa22, agou2}@illinois.edu

ABSTRACT

Visual inspection of bridges is customarily used to identify and evaluate faults. However, current procedures followed by human inspectors demand long inspection times to examine large and difficult to access bridges. To address these limitations, we investigate a computer vision-based approach that employs SIFT keypoint matching on collected images of defects against a pre-existing reconstructed 3D point cloud of the bridge. We also investigate methods of reducing computation time with ML-based and conventional CV methods of segmentation to eliminate redundant keypoints. Our project successfully localizes the defect images and achieves a savings in runtime from filtering keypoints.

Keywords:
Infrastructure, Structural Health Monitoring, Bridge Management Systems, SfM, Segmentation, SIFT

INTRODUCTION

Monitoring the condition of bridges is essential to ensuring structural integrity. The traditional method of assessment involves deploying inspectors who would manually scan the bridge for defects (rusts or cracks), take a picture, and manually record the details (including the location) on paper. After the inspection phase, the repair team will begin work, guided by the details reported by the inspectors.

The project goal would be to eliminate the paper-based part of the bridge inspection process by automatically determining the location of defects in relation to a predefined 3D reconstructed point cloud based only on the photo taken by the inspector. In the literature, direct 2D to 3D matching [4] has been shown to offer considerable improvement in the number of registered images over indirect methods through prioritized correspondence search and using visual word descriptors to reduce search space. Additionally automatic crack detection has been explored before [6] but the existing literature does not incorporate localization in a 3D reconstruction.

We propose an algorithm that functions based on keypoint detection of defect images and feature matching with an existing 3D point cloud. This involves reconstructing the camera position from the matched keypoints of the defect image. The position of the defect is localized using the reconstructed camera position and defect image. Lastly, using gradient features of bridges, background SIFT keypoints are automatically filtered out to reduce computation time when recovering the camera position of test images. At the end, an augmented reality visualization is created for the localized defect within the reconstructed 3D point cloud.

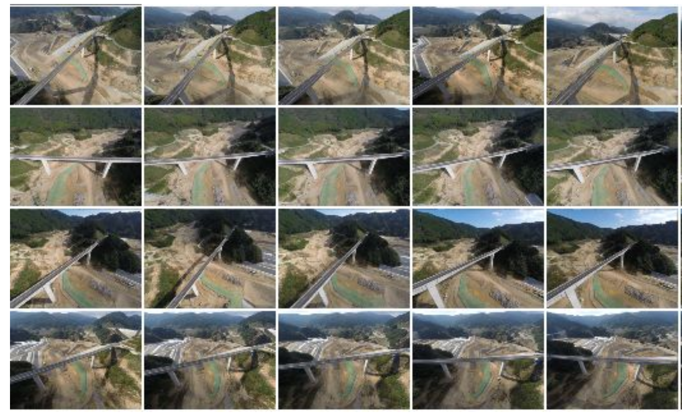

Figure 1: The dataset of bridge images of size 4797x3598 with pipeline positions of cameras

APPROACH

As shown in Figure 1, out data consists of 100 reference images and 15 test images taken by drone from multiple angles. The overall workflow can be summarized as follows (as well as in Figure 2):

1. Recover camera information and 3D dense cloud points from reference images by using VisualSFM [9], which incorporates SiftGPU, Bundle adjustment, CMVS dense reconstruction to perform SFM reconstruction.
2. Load and visualize camera position and orientation from bundle adjustment output file, and then run OpenCV SIFT keypoint detection for all reference images and test images.
3. Perform FLANN matching between test and reference images with distance filter 0.7. The matching scores are calculated based on the number of matching points, and the top 5 reference images are chosen.

4. Remove outlier 2D points from 5 reference images by calculating mean and standard deviation of descriptor distance and removing matches outside of 1 standard deviation.
5. Crop 5 reference images to bounding box containing only inliers. Run SIFT detection and FLANN matching using cropped images again, and use these results for the following steps.
6. Use projection matrix of 5 cameras from bundle adjustment and apply triangulation method to compute potential 3D point coordinates related to the feature in the test image.
7. Use mean and standard deviation outlier removal algorithm to remove the outlier points from proposed 3D points.
8. Compute mean and std. to decide the 3D cube location and size, and visualize them with dense reconstruction.

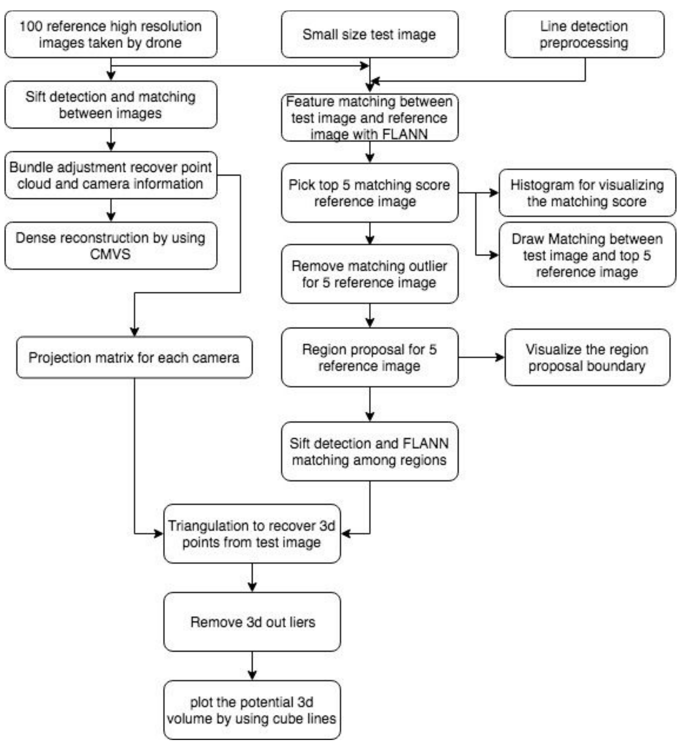

Figure 2: Diagram of overall project workflow

Improving Algorithm Efficiency

Keypoint matching is computationally expensive, taking a quadratic number of comparisons in the number of keypoints. This is especially applicable when matching keypoints of a test image against a large set of existing keypoints, so reducing the number of keypoints in the test image can have a large impact on computation speed. We propose a method to reduce the computational cost by identifying a Region of Interest (RoI) which constitutes the parts of the image that correspond to the bridge. The bounding box around the bridge can often be less than a third the size of the original image. By running the algorithm on only the RoI, we can reduce the number of redundant keypoints on the reference images used in the computation by a significant factor. We attempted both CNN-based segmentation and conventional methods of detection.

CNN-based Segmentation

For the CNN-based approach, 15 images were manually annotated with 10 used as the training set and 5 used as the test set. Since the problem scope is extremely limited and simple, a small dataset like this was sufficient to obtain an acceptable result. The images and annotations were resized to 256x256 to reduce the complexity of the network.

We adopted an architecture modified from VGGNet [8] where we replaced the fully connected layers at the end with convolutional layers and added a softmax layer to obtain probability maps for the background and the bridge. The network was trained for 200 iterations with a batch size of 16 using SGD of learning rate 0.0001 and momentum 0.9. A rough segmentation is obtained by taking the argmax of the two probability maps. To determine a bounding box for the crop, the mean across each row and column was taken, and the first and last rows and columns that had a mean greater than a threshold of 0.2 were chosen. In the end, given manual annotations of a few representative images, the entire dataset of reference images have RoI boxes extracted.

Edge Detection and Line Fitting

In this approach, edge detection and line fitting were combined to obtain a bounding box for the RoI. For edge detection, we used the Canny edge detector of OpenCV, since it has the best performance for segmentation according to an experiment by Muthukrishnan on all of existing proposed edge detectors [5]. After extracting the features, a probabilistic Hough transformation was used to draw the boundaries of the bridge [2]. This improved version of the Hough transformation is based on RANSAC principles and is thus faster than the original method. It starts with a random batch from the dataset, and based on tuning hyperparameters of minimum length of line and maximum allowed gap between line segments, it allows the user to have more accurate results according to his needs. Since different views of bridge were appeared in our dataset, it is also important for the boundaries to include the bridge columns. To obtain the box, thresholding only the lines with horizontal scope gave us satisfying results. To obtain the RoI rectangle, min and max values of all of the starting and ending points of lines in x and y direction were extracted to form the bounding box.

It should be mentioned that all the hyperparameters were tuned accordingly to get good results for this dataset, and it cannot be a good substitute for a generalized detector like

CNN. However, since in this algorithm all of the input would only need to be processed once, it is reasonable to use conventional methods.

RESULTS

In Figures 6, 7, and 8, the results generated from VisualSFM are plotted with the open3d package in Python.

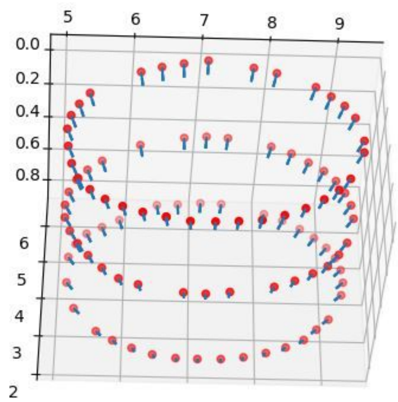

Figure 6:. Camera position (red dots) and orientation (blue arrows).

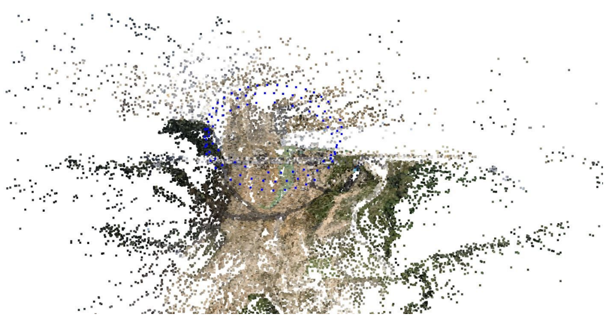

Figure 7:. Sparse reconstruction from bundle adjustment (blue points indicate camera positions)

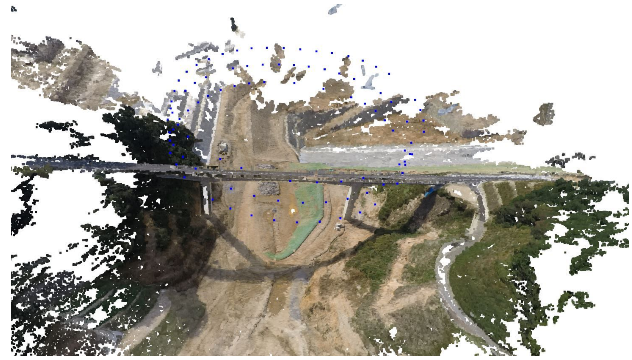

Figure 8:.Dense reconstruction from CMVS (blue points indicate camera positions)

Three test images are used to verify the result: Test 1(Figures 9 - 14), Test 2 (Figures 15 - 20) and Test 3 (Figures 21 - 26)

Test Image 1

Figure 9: Test image 1

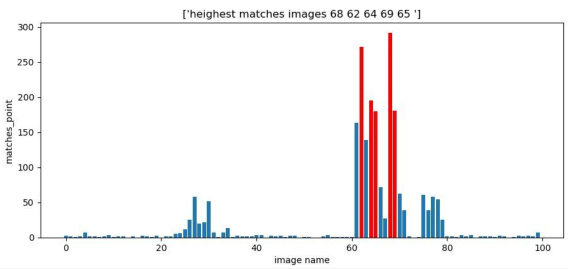

Figure 10:. Matching score of test image 1

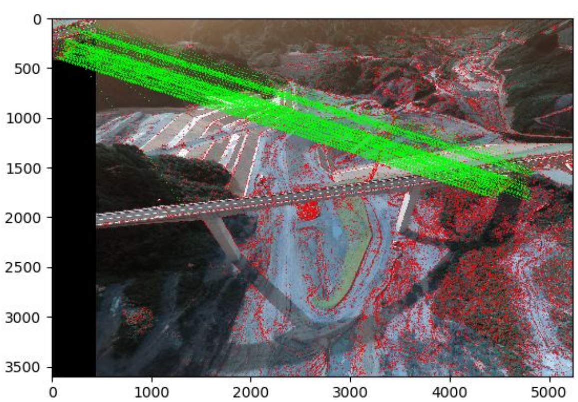

Figure 11: Highest matching score reference image matching result for test image 1 (green lines connect between points in reference image and test image in top left corner)

Figure 12: Region crop of reference image for test image 2

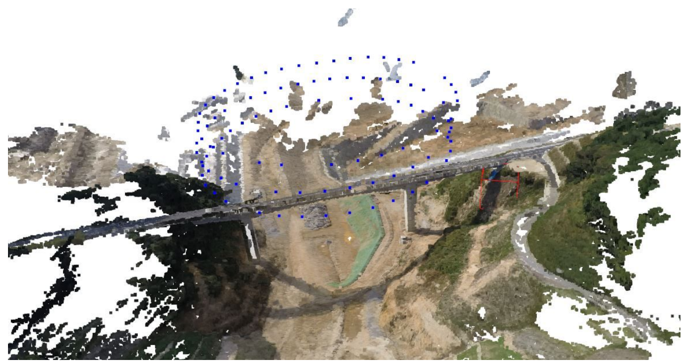

Figure 13: Final result of test 1 (red cube indicate the matching volume)

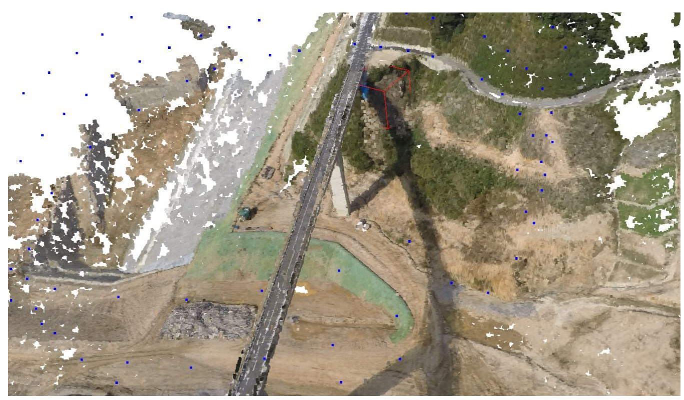

Figure 14: Final result of test 1 from another angle.

Test Image 2

Figure 15: Test image 2

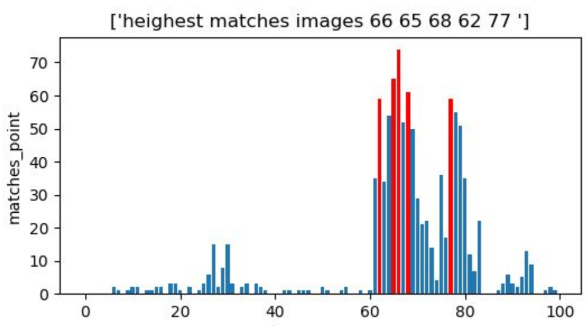

Figure 16: Matching score of test image 2

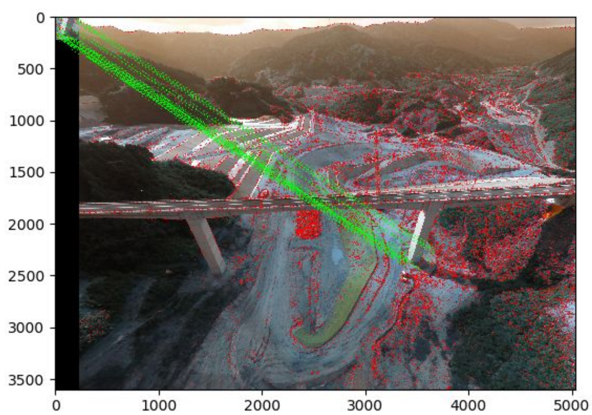

Figure 17: Highest matching score reference image matching result for test image 2 (green lines connect between points in reference image and test image in top left corner)

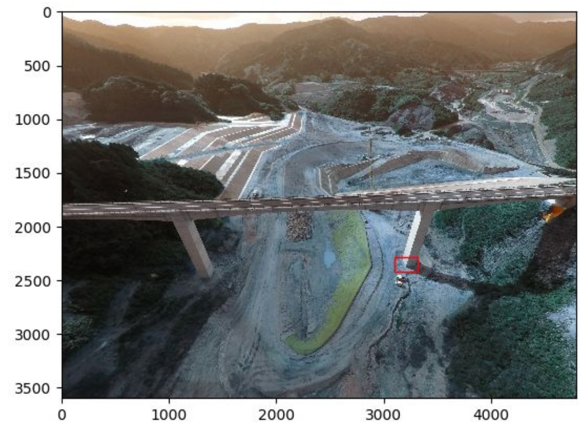

Figure 18: Region crop of reference image for test image 2

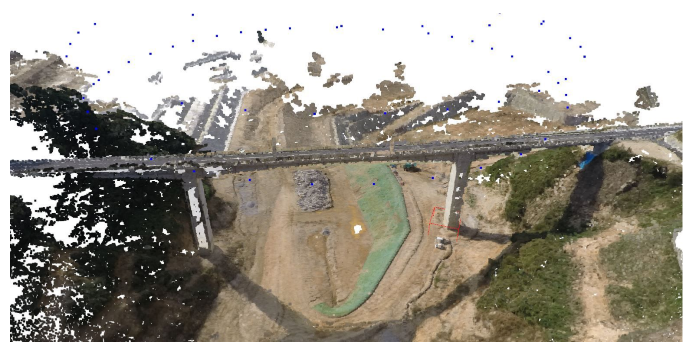

Figure 19: Final result of test 2 (red cube indicate the matching volume)

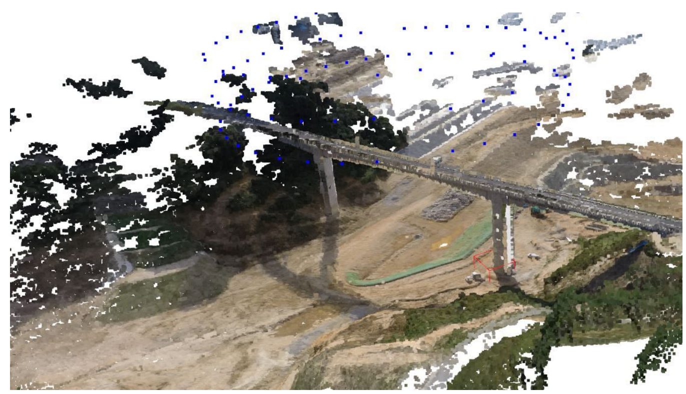

Figure 20: Final result of test 2 from another angle.

Test Image 3

Figure 21: Test image 3

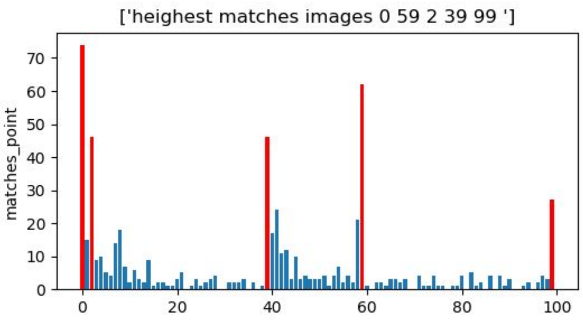

Figure 22: Matching score of test image 3

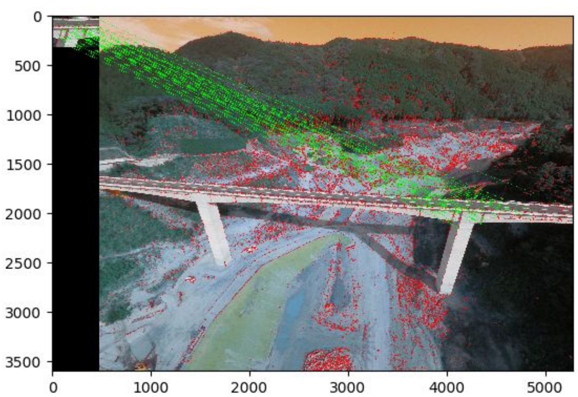

Figure 23: Highest matching score reference image matching result for test image 3 (green lines connect between points in reference image and test image in top left corner)

Figure 24: Region crop of reference image for test image 3

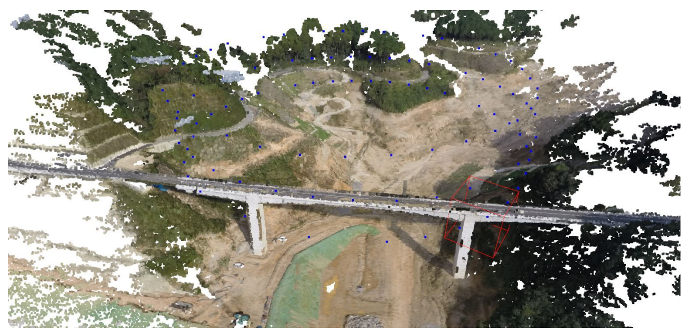

Figure 25: Final result of test 3 (red cube indicate the matching volume)

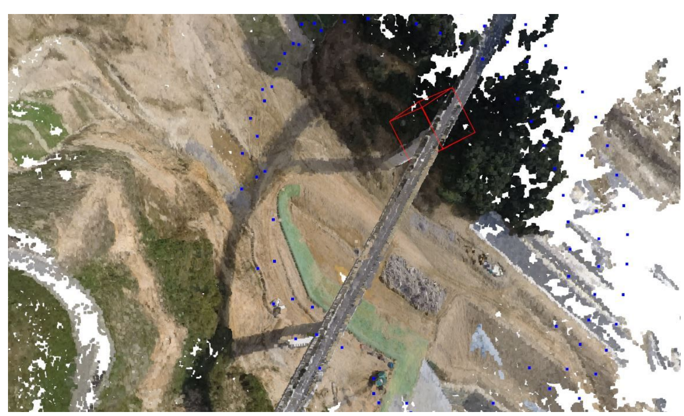

Figure 26: Final result of test 3 from another angle.

RoI Cropping

CNN

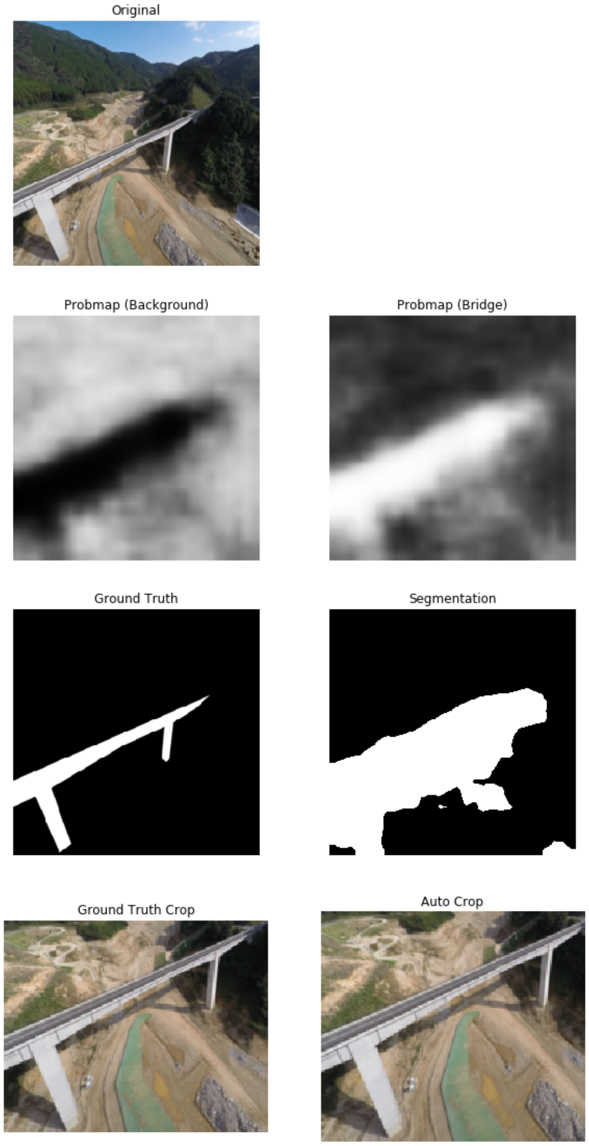

Figure 3: Results from CNN-based approach to bridge RoI cropping. First row: original image; Second row: probability maps of background and bridge; Third row: Ground truth and predicted segmentation; Last row: ground truth RoI crop and automatic RoI crop

For the CNN, across 5 annotated test images, the obtained RoI was on average 38.2% of the original image's size and had an mean IoU of 0.647 with the ground truth RoI. Excluding an outlier with particularly poor segmentation, the mean IoU was 0.726.

Edge Detection and Line Fitting

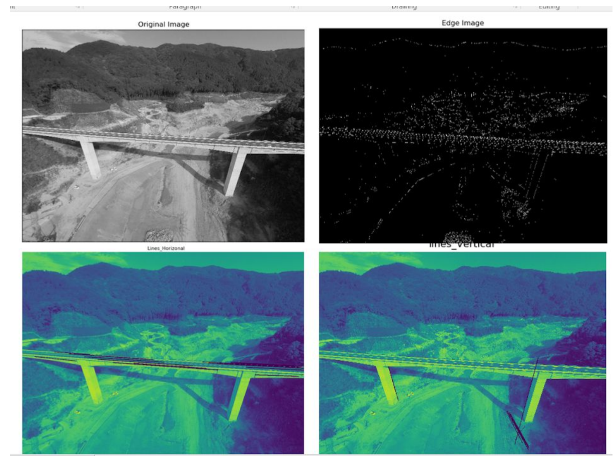

Figure 4: - Top row left: original image, top row right: detected edges, bottom row left: drawn horizontal lines, bottom row right: drawn vertical lines

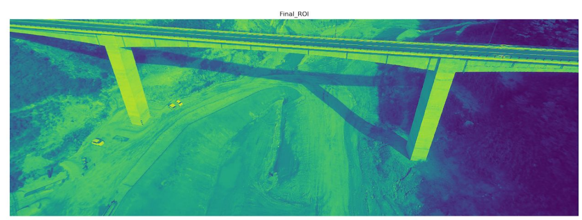

Figure 5:- Final region of interest of this sample

Table 1.Computation times for a single test image

	Computation time (sec)	**RoI size**
without RoI	137.398133039	-
with RoI (CNN)	96.30690050125122	59.48% height 81.69% width 48.59% area
with RoI (edge detection)	99.4285252094	64.99% height 67.44% width 43.82% area

Both RoI methods perform around 30% faster than the baseline. The edge detection method managed to create slightly smaller RoI boxes than the CNN method, but it is impossible to ascertain accuracy without RoI boxes generated from ground truth annotations of the reference images.

DISCUSSION

1. Although the quality of the 3D reconstructed point cloud is highly dependent on the collected data (which greatly benefits from a diverse range of camera angles and positions), the accuracy of the solution proposed for the problem is independent from the extracted projection matrices since the the target is the relative position of the defect in the 3d point cloud.
2. One potential caveat could be different lighting and shading conditions due to time of day or change over time between the test images and the reference images. In this particular case, the test images were chosen from the same dataset which were taken one the same day of the main dataset for point cloud. However, none of the test data used in the reconstruction procedure.
3. RoI selection using CNN segmentation proved surprisingly effective given the small size of the dataset (10 images). This is likely due a combination of reasons. First, the similarity of the conditions between the dataset used for training and the reference images means the training data is quite close to the test data. Second, the segmentation does not have to be very accurate to obtain an adequate bounding box on the RoI, so even a mediocre network can be sufficient. However, the CNN segmentation may very well prove less robust when applied to datasets that have reference images with varying conditions.
4. Augmented reality visualization of the defect location in the 3D point cloud was not in the original scope of this project but has been done successfully. The size of the red cube is based on the standard deviation of the descriptor distances between matches. For more practical usage, this visualization could be improved to contain more info describing the date and time of the taken inspection image, name of the inspector and any other information for more description of the defect.
5. Another idea for future work is to incorporate an classifier to automatically detect the type of the defect from out of a variety, each caused by different phenomenon [3] (A brief showcase of different types of concrete cracks can be found in the Figure 27). A successful implementation of such a system module is STRUM [6], which is recommended for further readings in the references.

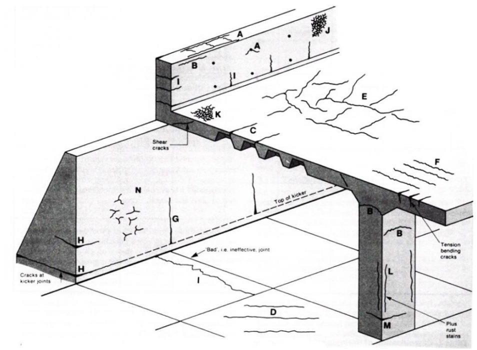

Figure 27: - different crack types

With such a classifier, it would be possible to localize the crack location and identify the type and cause of the crack without need for skilled judgment.

CONCLUSION

We have demonstrated a successful application of 2D-3D matching to the domain of bridge defect localization, successfully employed two different approaches to filter redundant keypoints for a sizeable algorithmic speedup, and created an augmented reality visualization of the bridge defect in relation to the 3D reconstruction. This project can be used to eliminate the use of paper reports and has the potential to serve as the basis for a platform that visually integrates technical data relating to bridge condition assessment.

STATE OF INDIVIDUAL CONTRIBUTION

Andre: investigate ML-based method of segmentation for bridge RoI detection and algorithm speedup, final report revision and proofreading

Majdi: Data handling and extracting information such as camera locations. Worked on the 3d reconstruction with keypoint detection and matching.

Liu Hong: Contribute to sift detection, outlier remove, FLANN matching between test and reference image. Worked on final result visualization and region crop.

Shayegan: Line fitting and edge detection to resize the reference image. Applied Hough transform to improve the efficiency of the matching progress.

REFERENCES

[1] A. Williamson, A. Chen, R. Sah R. J. Ortho. Research, 19(6)1113-1121, 2001.

[2].https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_houghlines/py_houghlines.html

[3] Srewil, Yaseen. (2006). Cracks of concrete and repair works & aase study. 10.13140/RG.2.2.12299.95522.

[4] Sattler, B. Leibe and L. Kobbelt, "Fast image-based localization using direct 2D-to-3D matching," *2011*

International Conference on Computer Vision, Barcelona, 2011, pp. 667-674.
doi: 10.1109/ICCV.2011.6126302
[5] Muthukrishnan.R, M.Radha, EDGE DETECTION TECHNIQUES FOR IMAGE SEGMENTATION International Journal of Computer Science & Information Technology (IJCSIT) Vol 3, No 6, Dec 2011
[6] P. Prasanna *et al*., "Automated Crack Detection on Concrete Bridges," in *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 591-599, April 2016.
[8] . Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. ICLR 2015
[9] Wu, C. (2011). VisualSFM: A visual structure from motion system. *http://www. cs. washington. edu/homes/ccwu/vsfm*.
[10] http://ronny.rest/tutorials/module/localization_001/iou/