Development of an Automatic Crack Inspection System for Concrete Tunnel Lining Based on Computer Vision Technologies

C K Nguyen¹, K Kawamura², M Shiozaki³, and A Tarighat⁴

¹Graduate School of Science and Engineering, Yamaguchi University, Japan
²Graduate School of Science and Technology for Innovation, Yamaguchi University, Japan
³Sumitomo Mitsui Construction Co., Ltd., Tokyo, Japan
⁴Dept. of Civil Engineering, Shahid Rajaee Teacher Training University, Iran

Abstract. This paper presents an automatic crack inspection system for concrete tunnel lining based on image processing technique and computer vision. The system includes a video image acquisition device, an image stitching software, and a crack detection software. Firstly, the video image acquisition device uses six video cameras and three illuminators mounted on a steel framework which is capable of sliding from side to top of the inspecting vehicle to shoot the full surface of the tunnel lining. Secondly, the image stitching software based on image matching technique was developed to create layout panorama from the tunnel lining surface images, making it easier to visualize a large and detailed section of the tunnel lining. Finally, we propose a semi-automatic crack detection software relied on a combination of image processing technique and interactive genetic algorithm (iGA) to crack detection for concrete surface images of the tunnel lining. Experimental results demonstrate the effectiveness of the proposed method.

1. Introduction

The number of deteriorated concrete components of existing civil infrastructure system has increased dramatically in many countries due to rapid aging. Therefore, the maintenance and management of existing concrete structures of the civil infrastructure system have become a major social concern worldwide. Especially, the existing tunnels were built over 50 years. To keep concrete components in good condition under all circumstances, the inspection plays an important role in the maintenance of infrastructure. Crack detection on concrete tunnel lining surface is one of the basic tasks of the inspection. Therein, the visual inspection is one of the most frequently adopted, which is carried out by inspector’s observation. It has a number of limitations, such as the high labour costs associated with carrying out tasks, and inaccuracy due to subjectivity, safety, slow inspection speed and disruption to traffic. In recent years, automatic crack inspection system for large scale concrete structures is a very active area of research due to popular digital video camera and computer vision developments. Especially, for tunnel inspection, the layout panorama generation of the tunnel lining is one of the important tasks for preliminary tunnel inspection [1]. Altogether, many crack detection systems have been developed. Kawamura et al. [2] proposed interactive genetic algorithm is applied to adjust the image processing parameters. It is developed to extract crack pattern effectively. Fujita et al. involved preprocessing steps (median filter and background subtraction) [3] and post processing (Hessian matrix and threshold) to extract crack
features. Miyamoto et al. calculated crack widths based on the difference of pixel brightness of each row [4]. Nishinkawa et al. [5] designed System for Automatic Construction of Image Filter (SACIF) based on a genetic program to detect cracks on concrete structures. Stent et al. [6], [7] used clustering and ranking methods for detecting changes probability on tunnel lining surface. Yeum and Dyke [8] applied threshold of Frangi filter and Canny edge detector to detect bolt cracks. Additionally, digital video and line-scan camera-based inspection devices have undergone substantial developments. However, the existing equipment requires a high level of expertise for effective inspection work to be performed. Therefore, the inspection cost is likely to be expensive, which the maintenance budgets of local governments often cannot handle.

This study contributes a system including automatic panorama generation and semi-automatic crack detection software with high accuracy and inexpensive inspecting cost because image processing expertise and exclusive vehicle are unnecessary. Moreover the system is based on computer vision technologies, to aid the visual inspection of tunnel lining.

2. Video image acquisition device

The video images of the entire tunnel lining surface are continuously scanned by the video acquisition device assembled on a car shown in Figure 1. Figure 1 (a) exposes the system consisted of six digital video cameras (from V1 to V6), three illuminators attached to the steel framework that its structure has shape appropriated with a half of cross-section of the tunnel, and a car. Further, this device is possible of sliding from the side to the top of the car in order to shoot the full tunnel lining surface under several passes through the tunnel. Figure 1 (b) shows how to obtain image data in moving forward three times of the inspection car in the longitudinal direction of the tunnel. To capture full cross-section of the tunnel, the inspection car has to run forward three times and backward three times. Quality of the image data depends on many factors such as distance between the tunnel lining and inspection car, the illumination, and specification of video camera. This device is simple and does not require an exclusive vehicle. The video data are converted into the consecutive images with the specific overlapped region depending on the scanning speed of the inspection car. Due to the characteristic feature of the imaging system, the consequence image motion is only the translation, so it is unnecessary to concern the rotation. Therefore, the direct method, which makes use of available information in the image-matching location, is applied to stitch images.

3. Image stitching method

Figure 2 shows an image stitching procedure for the entire tunnel lining including three steps. Captured image data are retrieved from six video cameras which each camera contains a number of images depended on the length of the tunnel. In Step 1, the dataset is connected to form a panoramic image in the longitudinal direction of tunnel for each camera. In Step 2, these panoramic images obtained in the step 1 are stitched in the circumferential direction of tunnel for each region corresponding to each pass.
Subsequently, in Step 3, the connected images of the six regions (R1-R6) in step 2 are stitched together to make a layout panorama which consists of a full view of the entire tunnel lining surface. The layout panorama of the tunnel lining is produced from the combination of the automatic image stitching method of Step 1, and semi-automatic methods of Steps 2 and 3. In this paper, Step 1 is presented in detail.

3.1 Similarity metric

Figure 3 shows a representation of automatic matching process of two consecutive images in the tunnel longitudinal direction (X-axis). Here, image 1 is the referenced image, and image 2 is the registered image of the image matching process. A search area is set in advance. Each the movement step of the search point is with respect to the location which enables image 2 to be shifted on image 1 to find the appreciable image-matching location by measuring the similarity and sharpness in the brightness of all pixel pairs in the overlapped region of the two images.

Furthermore, to accelerate the search and measurement process, the search point and similarity measurement are skipped in the search area and overlapped region with predefined values, respectively. Firstly, the matching cost functions are used to measure similarity of pixel-wise intensity in the overlapped region presented in detail:

\[
SAD = \sum_{i=1}^{K} \sum_{j=1}^{L} |(I_1 - I_2)|_{(i,j)}
\]

\[
(k,l) = \{ \arg \min_{\text{search area}} (SAD) \} | k \in [T_1, T_2] \}
\]

\[
T_{1,2} = \frac{v \max_{u \times t}}{u \times t}
\]

Where, \(I_1\) and \(I_2\) are the intensity of pixels at coordinates \((i, j)\) of images 1 and 2 respective in the overlapped region, \(SAD\) is determined from the sum of a difference between the pixel values of images 1 and 2 for each color channel (R, G, and B). Because standard \(SAD\) function has poor performance, image data are attributable to radiometric distortion and noise [9]. Authors propose a modified \(SAD\) function as follow: The numerator of the Equation (1) is divided by the area of the overlapped region (the total number of pixels) to normalize the overlapped region in each measurement. Furthermore, \((k, l)\) and \((K, L)\) are as the lower left and upper right coordinates (pixel) of the overlapped region of the image pairs, respectively. \(k\) coordinate is defined in image motion quantity (IMQ). Accordingly, the higher similarity of the overlapped region of two images obtains, the smaller score of \(SAD\) achieve. A
matching point at \((k, l)\) coordinates is corresponding to the minimum cost value of \(SAD\) function with winner-take-all strategy. 

\(T_1\) and \(T_2\) are the lower and upper threshold values determined from the (2). In that, \(v\) is the initial setting speed of inspection car. It is equal to 30 Km/h. \(\alpha\) is a parameter of the speed variance that is equal to 5 Km/h. In addition, \(u\) is image resolution in unit mm/pixel, and \(t\) defined frame rate is a number of frames per second in the scanning tunnel progress. To avoid too strict for keeping constant inspection speed, authors allow the speed change of inspection vehicle to 30 ± 5km/h.

### 3.2 Refinement

We use median filter to smooth some local bad matches which appear corresponding to the inappropriate IMQ of the neighbor images as the following equations:

\[
Q_i = \begin{cases} 
M_i & \text{if } |S_i - M_i| \leq T_i \\
S_i & \text{otherwise} 
\end{cases}
\]  

(3)

Where,

\[
S_i = \text{med}(M_{(i-j)}, \ldots, M_i, \ldots, M_{(i+j)})
\]

if \(i \leq j\) then \(M_{(i-j)} = 0\)

(4)

Here, \(Q_i\) is IMQ in unit pixels after refinement, and \(M_i\) is IMQ before refinement. \(S_i\) is a median value obtained from (4), and \(T_j\) is a threshold value depended on the speed change of the inspection car. The size of the median filter is \(2j+1\) (where \(j\) is an integer).

### 4. Experimental work

#### 4.1 Case study

The tunnel used in this experiment is an actual single-core circular tunnel with a length of 230 m in Yamaguchi Prefecture, Japan. Additionally, the image data sets captured for each camera from the image acquisition system consist of 1,558 images with a resolution of 1920 x 1080 pixels for each image. The initial setting inspection speed is \(v = 30\) Km/h, and image resolution is \(u = 0.231\) mm/pixel. Further, the frame rate is \(t = 60\) fps, so the overlapped region of each image pair is over 50%. To ensure the resolution, a photographic laser distance meter is used to maintain the distance between the tunnel lining and the inspection car.

Further, a prototype software based on the C++ language was developed to implement the full image stitching algorithm. The search range is set in large enough space to ensure the search result of matching point objectively, as shown in Figure 3. The skipped step for measuring similarity and searching matching point is set to 11x4 pixels. Additionally, the parameter values of refinement step are chosen as follows: size of the median filter is 100, and the threshold value is 50.

#### 4.2 Experimental results

Figure 4 shows IMQ results of camera V1 via the running direction of inspection vehicle before and after refinement. The IMQ result via the vertical direction (Y-axis) isn’t presented because its variance is not much different. In Figure 4(a), some local points show inappropriate matching results before refinement. After refinement is applied, IMQ graph presents smoother shown in Figure 4(b). The IMQ results of the other cameras are similar to camera V1.
Figure 4. The IMQ results before and after refinement.

Figure 5 shows a layout panorama segment resulted from the proposed image stitching method in both longitudinal and circumferential directions of a half of tunnel cross-section obtained from the image acquisition device. The panoramic pictures of the entire length of the tunnel lining include 22 segments. The length of each segment is 10.5 m excluded overlapped parts at the two ends of next segment. In addition, each mosaic region results from stitching panoramic image of each camera via circumferential direction. In that, each camera mosaic compressed with scale 1 per 32 has 45600x1920 pixels based on stitching 65 RGB (1080x1920) resolution images. The panorama shows high-accurate alignment in Figure 5. As a result, inspectors can detect tunnel lining abnormalities off-line easily.

Figure 5. A represented panorama segment of three imaging regions shown in Figure 1(b), and Step 3 in Figure 2.

5. Crack detection method

5.1 Image processing technique

The algorithm for accurate crack detection has composed of three major process parts shown in Figure 6. The first part was image input part which included image input and gray-scale image transformation. The second part was crack enhancement part which included the following pre-processing steps: median filter, subtraction, binarization. The final part was noise removal part, there were two steps: labelling and linear degree determination. These parts are presented in [10].

5.2 Application of iGA to the optimization of image processing parameters

In this study, the iGA is applied to searching optimal values of the image processing parameters (IPPs) because it has had many advantages such as the evaluation for each individual’s fitness easier than the
individual’s fitness function of GA, small population size, less number of generation [11]. Figure 7 shows an operated mechanism of genetic algorithm including into the crucial three stages, namely the initial population generation, fitness evaluation of each individual in the current population, and evolution operation to create the next generation. Therein, the evolution operation consisted of selection, crossover, and mutation is repeated until finding best solution. Namely, the detailed steps are presented as follow:

![Operated mechanism of genetic algorithm](image)

**Figure 7.** Operated mechanism of genetic algorithm.

| Variable | Range  | Steps | bits |
|----------|--------|-------|------|
| fsize    | [1 127]| 2     | 6    |
| binary   | [0 255]| 1     | 8    |
| linear   | [0 32.5]| 0.5   | 6    |

**Table 1.** The solution ranges of IPP parameter.

5.2.1 Initialize the first generation randomly.

Firstly, the IPPs are combined to create an individual in a population. Secondly, each individual is represented by a chromosome encoded to a binary string, as shown in Figure 8. Namely, the size of structuring element (fs) is assigned by 6 bits, the binary is expressed by 8 bits and the linear is expressed by 6 bits. Finally, an initial population was generated randomly with a predetermined size. Table 1 shows some parameter values based on the preliminary experiments.

5.2.2 Fitness evaluation of each individual.

The fitness of each individual (solution candidate) in a population is evaluated based on user’s preference. Therefore, the evaluation result is often a relative fitness. This point was different from traditional GA.

5.2.3 Stopping criterion.

Evaluation of each individual is determined by the user until the user finds the best individual corresponding to the optimal parameters.

![Input image window](image)

**Figure 8.** Coding presentation of a solution candidate.

**Figure 9.** Input image window.
5.2.4 Evolution operation.

When stopping criterion is not satisfied, evolution procedure is implemented for operating selection, crossover, and mutation until the user chooses the best parameters.

6. Crack detection prototype software development

In this study, the image processing algorithm shown in Figure 6 is developed in the C program language. In addition, an interactive interface of the software for extracting crack and searching optimal parameter values are developed in C# program language (see Figures from 9 to 11). The following describes the prototype software that includes five STEPs of semi-automatic crack extraction.

6.1 STEP 1

Firstly, the user inputs an original digital image by clicking "Input" button in Figure 9. The software is able to divide a large size image into many subimages. In this paper, the original image size is 800x800 pixels. Secondly, the user checks "Image cut out" button to adjust the image processing parameters. Here, the size of the cut-out image is 300x300 pixels. Next, the user chooses an interested region of the cracks. Accordingly, a red frame appears automatically as Figure 9. The purpose of the crack region selection is to search optimal image processing parameter values for the cut-out image.

6.2 STEP 2 (Adjustment of parameters)

The interactive interface is presented as Figure 10 so that the user adjusts the parameters. The software generates 15 sample images (individuals) as an initial population. The user can keep elites (better individuals) to next new generation. If the user cannot keep any individuals, the user has to operate upon “Initialization” to generate a new population. In this window, the user evaluates the fitness of each individual with five buttons from 1 to 5 corresponding to the fitness from the worst to the best. These buttons are located below sample images. Subsequently, the user chooses “Evolution” button to perform gene operators (crossover, mutation).

6.3 STEP 3 (Crack extraction)

To extract cracks from the original image smoothly, the user traces roughly the shape of the cracks by a finger on the touch panel as Figure 11 (a). This operation is also capable of processing with a computer mouse. The place in which the user traces on the touch panel is displayed as a red line. After limiting the crack region by a user’s fingertip, the cracks are automatically displayed in the selected region with a blue line shown in Figure 11 (b). The user has to trace region of the crack to extract it from the background image. Hence, the crack of the final result image will be extracted with reduced noises.

As a matter of course, depending on the chosen parameter value, the cracks can be appeared with noise pixels or loss pixels. If some crack parts are not displayed, then the user returns STEP 2 to choose other parameter values. If many noises are also displayed as a portion of the crack, then the user moves to next STEP 4 for noise removal.

6.4 STEP 4 (Noise removal)

When some noises appear with the cracks are extracted from STEP3, the user selects the “Eraser” button in Figure 11 to carry out a process of noise removal. If the noises are not presented, the user moves to STEP5 without selecting the “Eraser” button.
6.5 STEP5 (End determination)

When the user completes the crack extraction process, he/she presses “End” button at the bottom left of the window shown in Figure 11. On the other hand, when the user continues working crack extraction, he/she returns to STEP3. If the user wants to choose the parameters again, the user returns to STEP2. Consequently, the outputs of the software have two images automatically saved in the specified folder. Those are a crack extraction image as Figure 12 (a) and a crack map (final result image) as Figure 12 (b).

7. Evaluation and discussion

7.1 Evaluation method

The proposed method is compared with the Haar wavelet transform (HWT) method with GA to optimize the parameters of HWT [12]. The HWT method is a useful tool for crack detection with many advantages as multi-resolution properties. They are compared with the benchmark image traced by human operator based on loss and noise rate values expressed as the (10) and 11 equations:

\[ f_1 = \frac{m}{M} \times 100\%; \quad f_2 = \frac{n}{N} \times 100\% \]  

\[ f = 1 - \sqrt{\frac{1}{2} (f_1)^2 + \frac{1}{2} (f_2)^2} \quad f \in [0,1] \]  

Where \( m \) is the number of loss pixels, \( n \) is the number of noise pixels. \( M, N \) is the number of crack pixels and back pixels respectively. \( f_1, f_2 \) is percentage rate of loss and noise respectively. Additionally, \( f \) is evaluation value. The higher value is obtained, the higher accuracy of the software is obtained.

The process of wavelet transform method is firstly the wavelet decomposition level which is considered as a parameter which needs to be found optimum value. Secondly, hard threshold based all sub-band coefficients of HWT are adjusted. If the coefficient values are less than the hard threshold value, they are replaced by 0s (white pixels). Otherwise, they are kept unchanged values. The hard threshold also is taken account into the second parameter of solution candidates to search optimum value. Consequently, a filtering image is reconstructed by inverting wavelet transform. Next, the labelling is performed to connect 4-neighboring components. Finally the linear step is implemented as the part 3 of the proposed method, and it also is considered as the third parameter of solution candidate. As a result, the candidates of solution are encoded into binary string.
16 bits which include the level the composition (2 bits), the hard threshold (8 bits), the linear (6 bits). In this paper, the experiments are implemented for the two methods with 10 original images included various texture and uneven illuminating conditions as sample tests.

7.2 Discussion

As a result, Table 2 shows HWT method and the proposed method. In this table, the accuracy of crack detection results of the proposed method outperforms the ones of the HWT method for all sample tests. The crack detection of the HWT method is limited with respect to low contrast images. For illustration, Figure 13 shows the crack detection results of orig. Image 10. The loss and noise rate in Figure 13.(c) of The HWT method are too much compared with the one of the proposed method. The crack detection accuracy using proposed method is \( f \) equal to 0.9355. Meanwhile, the crack detection accuracy of the HWT method is \( f \) equal to 0.8140.

8. Conclusion

In this study, the crack inspection system included the useful imaging device, image stitching and image processing software. Therein, the imaging device was capable of collecting full enough tunnel lining surface images without constraint traffic to create an enlarged long panorama of tunnel lining aided to visual inspection easily. The proposed image matching method in this study implements the similarity metrics and the dual threshold values based on the imaging speed. Moreover, median filter is applied to refine local inappropriate matching points of the images in the experiment. Additionally, combination of image processing techniques and interactive genetic algorithm adjusted the optimized parameters of image processing procedure. The crack detection accuracy of the proposed method compared with the HWT method. The experimental results demonstrated that the effectiveness of the proposed method. The limitation of this system is that properties of the crack shapes and sizes aren’t classified as well as measured automatically. These problems would be implemented in the future research.

---

**Figure 13.** (a) Original mage No.10; (b): crack image result of proposed method; (c) crack image result of HWT method.

| No | \( f_1 \) | \( f_2 \) | \( f \) | \( f_1 \) | \( f_2 \) | \( f \) | \( f_1 \) | \( f_2 \) | \( f \) | \( f_1 \) | \( f_2 \) | \( f \) |
|----|------|------|-----|------|------|-----|------|------|-----|------|------|-----|
| 1  | 1.48 | 1.06 | 0.9887 | 1.06 | 1.54 | 0.9868 | 6    | 6.15 | 3.43 | 0.9502 | 11.71 | 5.48 | 0.9086 |
| 2  | 3.29 | 2.73 | 0.9698 | 2.36 | 3.04 | 0.9728 | 7    | 0.99 | 1.22 | 0.9889 | 6.5   | 10.11 | 0.915  |
| 3  | 1.67 | 1.86 | 0.9828 | 1.11 | 2.82 | 0.9786 | 8    | 2.06 | 2.45 | 0.9774 | 4.35  | 1.91  | 0.9664 |
| 4  | 2.72 | 2.55 | 0.9736 | 11.05 | 7.79 | 0.9044 | 9    | 10.07 | 12.8 | 0.8844 | 20.34 | 22.08 | 0.7877 |
| 5  | 6.98 | 10.16 | 0.9128 | 6.06 | 24.07 | 0.8245 | 10   | 7.79 | 5.26 | 0.9355 | 24.26 | 10.18 | 0.8140 |

---

**Table 2.** Result of the accuracy analysis of hwt transform, the proposed method.
9. References

[1] Zhu Z, Fu J, Zhang X, 2016 *Journal of Computer-Aided Civil and Infrastructure Engineering*, **00**, 1-19

[2] Kawamura K, Miyamoto A, Nakamura H, Sato R, 2003 *Proc. Japan Soc. Civ. Eng.* **742**, pp. 115-131

[3] Fujita Y, Mitani Y, and Y. Hamamoto 2006 *ICPR2006: IEEE 18th Int. Conf. on Pattern Recognition*, 3, New York, 901-904

[4] Miyamoto A, Konno M, and Bruhwiler E 2007 *Asian J.Inf Technol* **6(5)**, 553-561

[5] Nishikawa T, Yoshida J, Sugiyama T, and Fujino Y 2012 *Computer-Aided Civil and Infrastructure Engineering* **27(1)**, 29-47.

[6] Stent S, Gherardi R, Steneger B, Soga K, and Cipolla R 2013 *International association for pattern recognition (IAPR) Int. Conf. on Machine Vision Applications* Kyoto, Japan

[7] Stent S, Gherardi R, Steneger B, Soga K, and Cipolla R 2016 *Machine Vision and Applications* 27:319–330

[8] Yeum C. M., and S.J. Dyke 2015 *Computer-Aided Civil and Infrastructure Engineering* 30,759-70

[9] Kumar S, and Chatterji N 2004 *ISSN:0377-2063 IETE Journal of Research*, **50**, No. I, pp 49-61

[10] Cuong Nguyen Kim, Kei Kawamura, Amir Tarighat, Junji Matsumoto, and Masando Shiozaki. Development of Semi-Automatic Crack Detection Software For Concrete Structures. *Proceedings of The 11th fib International PhD Symposium in Civil Engineering*, Tokyo, Japan. pp. 459-466, 2016-August

[11] Kawamura K, Miyamoto A, Nakamura H, Sato R 2003 *Proc. Japan Soc. Civ. Eng.* **742**, pp. 115-131

[12] Mukhopadhyay S, Mandal J 2013 Procedia Technology doi 10.1016/j.protcy.10.680-689

Acknowledgments

The research in this article is supported by the financial support of JSPS KAKENHI Grant Number 15K0618.