Spatial-temporal motion field analysis for pixelwise crack detection on concrete surfaces

Subhajit Chaudhury, Gaku Nakano, Jun Takada and Akihiko Iketani

NEC Central Research labs,
1753, Shimonumabe, Kawasaki City, Japan 211-0011
{s-chaudhury@ap, g-nakano@cq, j-takada@bc, iketani@cp}.jp.nec.com

Abstract

Crack development in concrete structures starts at the micro-crack stage and proceeds to the macro-crack stage due to repeated cyclic loading, like ongoing vehicles on bridges. Automatic detection of early stage cracks is required for both safety and economic reasons. We present an automatic crack detection method that scans a captured concrete area and provides a pixel-wise localization of both visible macro-cracks and early stage micro-cracks from video sequences. The key component in the proposed method is a spatial-temporal non-linear filtering on frame-wise dense 2D motion field combined with Conditional Random Fields based crack localization refinement. We evaluate our method against labeled ground truth data provided by an expert crack inspector. Experimental results show that our method can produce high accuracy automatic crack localization having F1 score improvement of 0.14-0.22 compared to conventional image based detectors. The proposed method is also shown to detect cracks at an earlier stage which enables early preventive measures for repair operations.

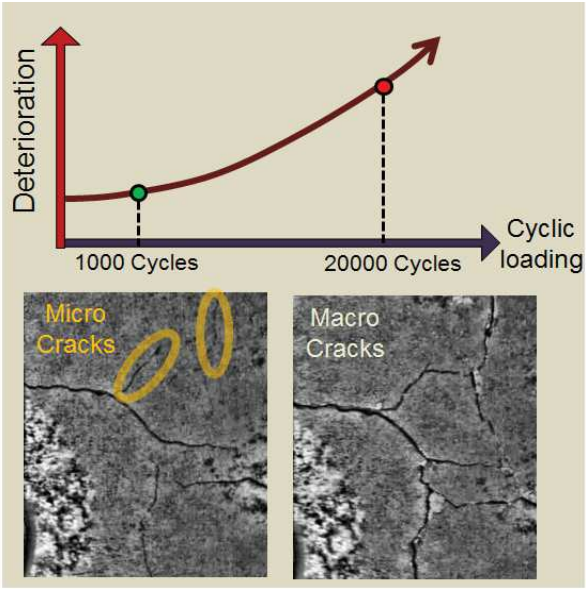


Figure 1: Showing growth of cracks from early stage with 1000 cycles to severe stage with 20,000 cycles. Micro-cracks (shown by yellow circles) are not easy to find in early stage. Images are enhanced for ease of viewing.

1. Introduction

Many concrete structures like roads, bridges etc. are subjected to repeated cyclic loadings which result in fatigue in these structures. Fatigue loading causes structural failures in the form of cracks. Research on fatigue loading of concrete structures [5, 6, 13, 16] suggest that crack development starts with micro-cracks and then propagates to macro-crack stage due to repeated cyclic loading. Ultimately after a finite number of loading cycles, the structure fails and is considered unsafe for use. This process of crack growth in real concrete is demonstrated in figure 1. Successful detection of early stage cracks is very important to ensure the safety of concrete structures. Such early detections could help predict future deteriorations, reduce investments by scheduling repairs at early stage and facilitate optimal scheduling of limited repair and maintenance resources.

In this work, we focus our study on bridge surfaces since ensuring safety of bridges is extremely important because of its daily usage by general public. Failure to detect damage in bridges may lead to disastrous accidents which might cause loss of life. Typically, cracks in bridges are monitored by certified inspectors and structural engineers by manually inspecting the bridge site. Such procedures are often

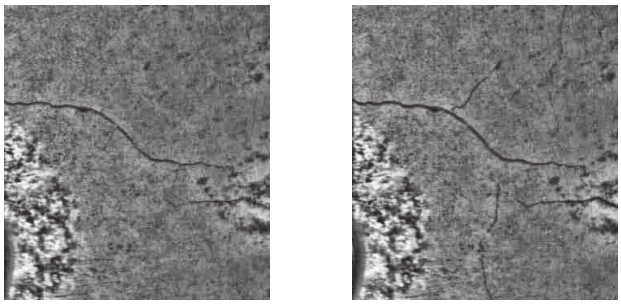
The final published version is available at https://doi.org/10.1109/WACV.2017.44.

slow and expensive because it requires setting up of scaffoldings, close inspection of each part of the bridge surface and stopping traffic on the bridge. Due to these reasons, automatic detection of cracks are gaining importance in the recent times.

Traditionally, crack detection algorithms proposed in literature are single image based methods. A survey by Koch *et al.* [11] gives a summary of recent computer vision based crack detection methods in reinforced concrete bridges, precast concrete tunnels, underground concrete pipes, and asphalt pavements. Some methods in literatures for solving this problem include measuring texture anisotropy [17], continuous wavelet transform based crack labeling of pixels [20], preprocessing images for noise removal to enhance crack localization response [4]. Percolation-based image processing techniques [22, 23] which considers connectivity of cracks for superior performance have also been developed. Patch based methods [7, 8, 21] which partition the image into small patches and assign binary labels to patches have also been proposed. A method for morphological transformations based crack candidate retrieval and subsequent crack detection by learning geometrical features of cracks was proposed by Jahanshahi *et al.* [10]. It extracts well designed features from each patch and train a Support Vector Machine (SVM) classifier to judge whether a patch is a crack or not. An image processing toolbox for crack detection was released by [14], which performs a comprehensive set of image processing algorithms for detection and characterization of pavement cracks. Recently Convolutional Neural Network (CNN) based crack classification combined with spatial-temporal grouping based refinement was proposed by Schmugge *et al.* [19] which labeled if a particular image has crack or not and provided coarse crack localization.

Although conventional methods attempt to find cracks from a single image, it is not suited for detecting micro-cracks. This is because early stage micro-cracks are not visible without load applied to concrete as shown in figure 2 and it is difficult to automatically capture images when load is applied above the bridge. Thus we propose crack detection from videos captured when the concerned concrete is under active dynamic loading. For example, videos captured from bottom surface of a bridge during regular traffic might be used for crack detection.

A simple way of detecting cracks from videos is to apply single image based methods directly to the image frames. However, detecting micro-cracks using "single image only" based detectors poses several problems. Firstly, as already mentioned, micro-cracks are visible only under load. This requires choosing frames where all cracks are visible in a single frame, which is difficult to obtain automatically. Secondly, even if we assume micro-cracks are visible in a frame, their width is typically sub-pixel level which makes



(a) Micro-cracks not visible in frame

(b) Micro-cracks visible in frame

Figure 2: (a) Micro-cracks not visible without loading (b) Cracks become visible when there is loading above the region being captured. This illustrates the need for spatial-temporal analysis for micro-crack detection. Images are enhanced for better viewing.

differentiating them from noisy structures such as scratches, concrete textures, weldings etc. even more difficult. Furthermore for machine learning methods, a huge amount of labeled training data is required for good generalization performance for crack detection in the wild. This is because concrete structures present large variation in appearance, texture, crack patterns, lighting conditions, loading condition and other factors. Obtaining sufficient labeled image data with all possible variations might be a time consuming and expensive process.

A popular method for image based strain detection from digital images in the field of civil and mechanical engineering is Digital Image Correlation (DIC) [2, 15, 18]. Cracks are regions on the image where stress concentration is high. In DIC, images of concrete before and after loading are compared to find the displacement field and its derivatives. Typically, locations in the image having high displacement derivatives are considered as crack pixels. A feasibility study for DIC based crack detection is presented in [9]. The following problems are characteristic of DIC based crack detection: (1) DIC based methods, which threshold motion gradient computed by simple block based correlation technique, produces crack localization which is usually coarse without fine-details (2) These methods cannot find cracks where the crack width is sub-pixel level or the heavy loading is not applied. (3) Special setup conditions with speckle patterns on test surface is required for accurate displacement fields.

In this paper, we present an automatic crack detection method which can detect both micro-cracks and macro-cracks on bridge surfaces with high crack localization accuracy. Figure 3 gives the flowchart for our motion based crack detection system. Our algorithm takes simple video

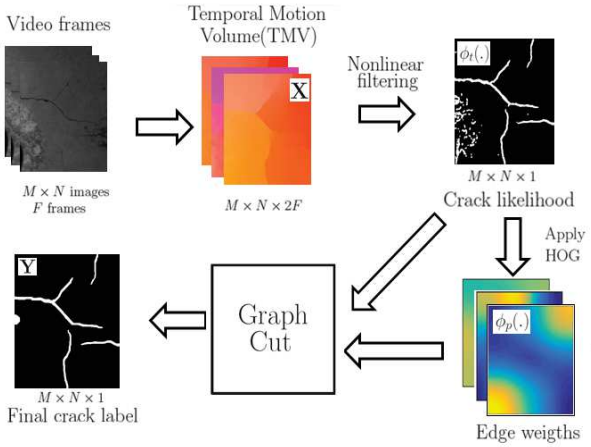


Figure 3: Overview of the proposed algorithm

sequences of bridge surface during wheel loading due to vehicles and gives pixel-wise detection which we refer to as "crack map". Accurate crack maps are required by inspection engineers to predict the growth of cracks in the future and thus take preventive measures at an early stage. Our method is heuristic in nature which does not require prior training procedure and utilizes physical properties of cracks on loading for detection. Inspired from DIC based methods, we use motion information to make our method robust to noisy image structures like scratches, weldings, etc. We frame the problem of crack detection as a binarization problem based on temporal motion information. The basic idea is that crack pixels are those regions where there is local motion discontinuity along the crack line. This is achieved by a non-linear spatial-temporal filtering technique which produces likelihood of crack at every pixel. The next stage is a Conditional Random Field (CRF) based refinement stage, where false detections are removed using a prior crack shape probability term which assumes that local shape of cracks are spatially linear. This two step method produces crack localization with higher accuracy than single image based methods by detecting early stage micro-cracks and removing false detections.

2. Proposed Method

2.1. Problem formulation

Given a video sequence of a sample concrete region with image size (M, N) and F number of frames, our goal is to find the spatial pixel-wise location of cracks on the concrete sample using inter-frame 2D motion temporal data. Let $\boldsymbol{X} \in \mathbb{R}^{M \times N \times 2F}$ represent the dense 2D motion for all frames in the video sequence computed with respect to a fixed reference frame. We refer to this frame-wise motion data as Temporal Motion Volume (TMV). Our goal is to infer pixel-wise crack label, given by $\boldsymbol{Y} \in \{0,1\}^{M \times N}$ from the TMV. This is a binary labeling problem which is solved on a graph structure. We propose a non-linear spatial-temporal filtering technique for generating crack likelihood map followed by CRF inference which solves the maximum-a-posteriori (MAP) problem. We find the optimum crack label $\hat{\boldsymbol{y}}$ by maximizing the posterior probability $P(\boldsymbol{Y} = \boldsymbol{y}|\boldsymbol{X})$.

We make the Markov assumption that the crack label at a particular pixel is influenced only by the labels of neighboring pixels and independent of other pixels. Following conventional approaches in CRF literature [1], we take negative log of the posteriori probability and convert the maximization problem to a minimization problem,

$$\begin{aligned} \hat{\boldsymbol{y}} &= \arg\max_{\boldsymbol{y}} \; P(\boldsymbol{Y} = \boldsymbol{y}|\boldsymbol{X}) \\ &= \arg\max_{\boldsymbol{y}} \frac{1}{Z(\boldsymbol{X})} \exp(-E(\boldsymbol{y}|\boldsymbol{X})) \\ &= \arg\min_{\boldsymbol{y}} E(\boldsymbol{y}|\boldsymbol{X}). \end{aligned} \tag{1}$$

Here $E(\boldsymbol{y}|\boldsymbol{X})$ is the energy of the configuration $\boldsymbol{y} \in \{0,1\}^{M \times N}$ and $Z(\boldsymbol{X})$ is the partition function. From now on, we drop the conditioning on $\boldsymbol{X}$, for convenience. The configuration energy $E(\boldsymbol{y})$ is composed of two terms. The first term assigns crack likelihood of local patch observing the temporal motion and the second term encodes prior knowledge about crack shape and acts as an regularization term for outlier rejection. We represent the above binary minimization problem as a graph structure, $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$ with $\boldsymbol{V}$ as vertices and $\boldsymbol{E}$ as edges of graph. The likelihood term is manifested as terminal weights and prior term influences edge-weights of the graph. We solve a energy minimization problem corresponding to the negative log of posterior probability in equation 1 with respect to the crack label $\hat{\boldsymbol{Y}}$ that can take binary values at each pixel. The energy term is given as,

$$E(\boldsymbol{y}) = \sum_{v \in \boldsymbol{V}} \phi_t(y_v) + \sum_{(u,v) \in \boldsymbol{E}} \phi_p(y_u, y_v), \tag{2}$$

where the unary term $\phi_t(y_v)$ gives the inverse likelihood of the pixel v taking the label y_v, and pairwise term $\phi_p(y_u, y_v)$ gives the cost of assigning labels y_u, y_v to neighboring pixel pairs u, v simultaneously. The unary and pairwise potential terms are described in details in the following sections.

2.2. Non-linear spatial-temporal filtering

This section describes the spatial-temporal filtering which is used as the unary potential to assign terminal weights of the graph. Using insight that cracks lie on motion discontinuity, we develop a non-linear spatial temporal filtering (NLSTF) scheme that produces high response at regions of motion discontinuity.

Given a kernel h, of size $B \times B$, where B is an odd number, we perform a non-linear filtering which computes the likelihood of crack at a given vertex v in the graph. Consider $\boldsymbol{A}_v$ represents the spatially local patch of temporal motion around the pixel represented by vertex v which has a size of $B \times B \times 2F$, if F image frames are used. We find a dissimilarity score between the central pixel with all the other pixels in the local window. For similar pixels, the score is close to zero and for dissimilar pixels the score has high value. A weighted summation of the dissimilarity scores is performed based on the kernel weights, which assigns the value of unary potential $\phi_t(y_v = 0)$ as given below,

$$\phi_t(y_v = 0) = \sum_i \sum_j \left(h(i,j) f_d\Big(\boldsymbol{T}^v_{i,j} - \boldsymbol{T}^v_{0,0}\Big)\right), \quad (3)$$

where $\boldsymbol{T}^v_{i,j} = \boldsymbol{A}_v(i,j,:)$, is a $2F$ length vector containing the motion information for all frames at a particular pixel location (i,j) relative to the central pixel. $\boldsymbol{T}^v_{0,0}$ is the pixel represented by vertex v in the graph and $h(i,j)$ represents the corresponding kernel weights. The function $f_d(.)$ computes the dissimilarity between two temporal motion vectors. For our implementation, we use the function given by,

$$f_d(\vec{x}) = \frac{1 - e^{-\gamma \|\vec{x}\|_2^2}}{1 + e^{-\gamma \|\vec{x}\|_2^2}}. \quad (4)$$

The proposed function for calculating dissimilarity score is robust to large outliers because the response flattens out for large values. We also tested with other functions like linear, L2 response etc. and found that re-descending M-estimators, like equation 4, show best performance due to their ability for rejecting large outliers. γ is the controlling parameter for outliers rejection with larger values indicating detection of minute motion discontinuities and smaller value producing high response only for large motion discontinuities. The output from NLSTF can be directly thresholded to obtain a preliminary crack labeling. However, in our CRF setting, we use this filter response for assigning the terminal weights of the graph.

2.3. Local shape based edge weighting

The non-linear spatial-temporal filtering assumes that the crack probability at each pixel is independent of that at other pixels. However, cracks show some known structure that can be utilized to act as regularization for rejecting outliers. If neighboring pixels form a local structure typical of cracks, it is highly probable for these to be crack pixels. Such prior knowledge about structure, can be used to prevent over-fitting from classifying all motion discontinuities as cracks. Also small gaps between dis-jointed cracks can be filled by using prior knowledge of shape. Due to local conformity of crack-like structure, the "gap" pixels, which were missed by NLSTF, might have high prior crack probability, leading to positive labeling by CRF inference.

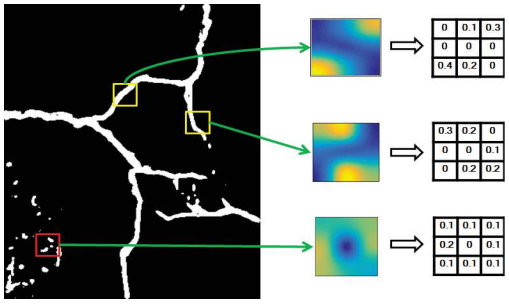


Figure 4: Deciding edge-weighting based on local crack structure. The yellow color depicts cells with dominant direction which are used to assign high edge weights in the neighboring pixels.

To encode prior information, we assume that cracks in a local patch locate on a line. We use the histogram of oriented gradients (HOG) [3] on every local patch of the motion discontinuity map, obtained from the NLSTF response, to locate the dominant direction which is used to assigning edge weights as shown in figure 4. Edge weight assignment of this fashion preserves locally linear structure and removes false detections.

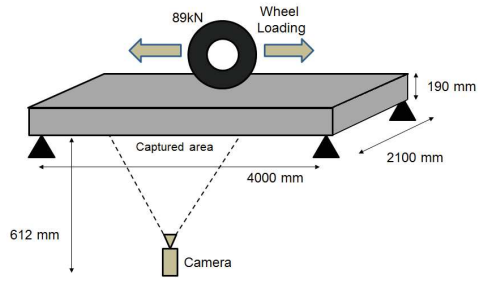
Let $\boldsymbol{L}_v$ represent the local motion discontinuity patch obtained from non-linear filter response. From this patch we compute the HOG feature on the motion discontinuity and obtain the gradient strength in each of the 8 bins oriented towards the 8-neighbors of the current graph vertex v which is represented as a vector $\boldsymbol{H}_v$ for the current graph vertex. Let $u \in N(v)$ be neighboring vertex, and $H_v(u)$ represent the element of gradient strength at vertex v in the direction of vertex u. Then the weights assigned to the edge (u,v) is given as,

$$\phi_p(y_u \neq y_v) = \frac{H_u(v) + H_v(u)}{2}. \quad (5)$$

The above equation suggests that graph edges lying in the dominant gradient direction are assigned higher weights and the other edges are given lower weights. A comparatively high cost has to be paid, if neighboring pixels in the dominant direction are assigned different labels. Such a weighting scheme favors removing edges that have lower edge weights and hence preserves those structures in the motion discontinuity map that have locally linear structure.

After computing the terminal and edge weights, inference on the graph is performed by standard graph-cut algorithm [1] to produce the final crack labeling.

(a) Wheel loading setup on concrete bridge sample



(b) Illustration of wheel loading on concrete block and video capture by camera

Figure 5: Experimental setup to simulate traffic loading.

3. Experimental Results

3.1. Experimental setup

We performed controlled experiments on real concrete block to verify our algorithm's effectiveness. A concrete block having width 2.1 meters , length 4 meters and thickness 190mm, was used for experiment purpose. The block was loaded with a moving wheel load (89KN) that traversed its length back and forth to simulate the effect of traffic. We refer to each such to-and-fro traversal of the moving wheel as a cycle. Camera was setup at a distance of 612mm from the lower surface of the concrete block perpendicular to the plane of bridge, which recorded the movement of slab as wheel was moving on the upper surface. Experimental setup is shown in figure 5. With each cycle, the deterioration of concrete slab will increase and the developed cracks and faults will become more prominent. For our experiments we captured the image sequences of the lower surface of the concrete block at roughly 1000, 2000, 3000, 4000, 5000 and 20000 cycles.

3.2. Implementation details

We now discuss the details on image processing methods and parameters used for realizing the discussed algorithm in section 2. For dense 2D motion extraction we used MIT CSAIL Optical flow [12] with default parameters. For each experiment, a base frame (first frame of the video) was fixed and motion in all frames was computed with respect to this base frame. The video sequence had $F = 200$ frames which were used to create the temporal motion volume (TMV) having 400 optical flow fields for x, y motion fields. The image dimensions were $(M, N) = (640, 480)$. For computing the response from NLSTF, the filter kernel (h) in equation 3, used for dissimilarity computation was a simple 8-neighbor Laplacian 3×3 kernel. Larger block-size may be used for more diffuse motion discontinuity across the crack. Parameter γ was set at 0.01 for the experiments. We used local patch size of $B = 32$ for edge weighting scheme. An implementation graph-cut by Boykov *et al.* [1] was used to perform inference. Some isolated false detections which might be detected by CRF inference were removed by connected components analysis. We follow a similar technique described in [10], by finding connected components and removing ones with number of pixels less than 244. We observed that this step gives us a small improvement in detection accuracy.

3.3. Evaluation metric

We test our proposed algorithm against crack labels provided by an expert crack inspector. The inspector was presented with the videos of concrete slab which were captured during the wheel loading experiments and also with single images having maximum crack opening for each cycle. We found out from the inspector that crack labeling using video sequences was much easier than using a single image, which is in line with our claim of improved crack detection from motion fields.

To evaluate crack localization accuracy, we divide the image into non-overlapping patches of size 32×32 and assign crack label to each such image cell in the expert labeled ground truth map and the detected crack map. Patches having number of crack pixels more than a certain threshold were declared as crack patch. We use standard true positive rate (recall), false positive rate and F1 score metric for quantitative evaluation of our method against other methods in literature. The overall detection quality is determined by F1 score.

3.4. Crack detection accuracy

Videos for 1000 (early stage), 5000 (intermediate stage) and 20000 (damaged) cycles were used for comparing crack detection performance. Labeling obtained by simple thresholding of non-linear filter response (NLSTF), CRF-refined label (CRF) and cleaned detection with non-connected components removal (CRF+CC), as discussed in section 3.2, were compared. These are shown in the last three rows of table 1. For simple thresholding, pixels with NLSTF response greater than 0.5 were considered as cracks. Qualitative comparison for these outputs are shown in figure 6 which also shows clear motion discontinuity at the location of cracks.

Table 1: Quantitative comparison of various methods. Results have been rounded up to 2 decimal places.

	1000 Cycles			5000 Cycles			20000 Cycles		
Method	**TPR**	**FPR**	**F1**	**TPR**	**FPR**	**F1**	**TPR**	**FPR**	**F1**
Schmugge [19]	0.37	0.15	0.32	0.48	0.12	0.46	0.84	0.31	0.49
Jahanshahi [10]	0.78	0.12	0.61	0.80	0.08	0.73	0.92	0.09	0.76
Proposed (NLSTF)	0.83	0.16	0.62	0.91	0.09	0.80	0.93	0.06	0.85
Proposed (CRF)	0.85	0.05	0.80	0.95	0.05	0.87	0.93	0.04	0.89
Proposed (CRF+CC)	**0.85**	**0.04**	**0.83**	**0.95**	**0.03**	**0.92**	**0.93**	**0.03**	**0.90**

Results indicate that NLSTF output, although has high recall, produces false positive detections which reduces overall F1 score. The CRF refinement process removes such outliers and reduces false positive detections. Recall is also improved by CRF refinement which joins formerly disjointed crack regions to form continuous linear elements. Further post-processing, which removes isolated detections, reduces false positive rate improving overall F1 score by a small amount.

3.5. Comparison with other methods

In the literature, while there are many image based crack detection algorithms, there has not been considerable work on motion based crack detection except DIC based methods [2, 15]. Thus we evaluated our method against the image based crack detection methods presented in [19] and [10] because they provide state-of-the-art results in image based crack detection. We also wanted to compare with crackIT [14], however we could not obtain license for using the toolbox. For fair comparison with our method, which uses multiple frames, image based crack detection was applied to 5 manually picked images where cracks were visible and the union of detections was computed to be used for comparison. We also tried taking union of detection for all 200 frames of each sequence, but that produced increased false positive detections at almost similar recall thus reducing overall F1 score. Images were preprocessed with adaptive histogram equalization to improve detections for both methods. The quantitative comparison results are provided in table 1. Figure 7 gives the visualization of crack detection all methods. Sequences for 1000 (early), 5000 (intermediate) and 20000 (damaged) cycles were chosen to show the performance of each method.

We also compared with DIC based methods. However for the case of bridge surface, proper texture was missing, unlike artificially sprayed metal structures where DIC is predominantly used. This led to erroneous 2D motion extraction and thus local gradient thresholding methods for crack detection did not produce satisfactory results. Hence we do not report the comparison with DIC.

3.5.1 Comparison with Schmugge [19]

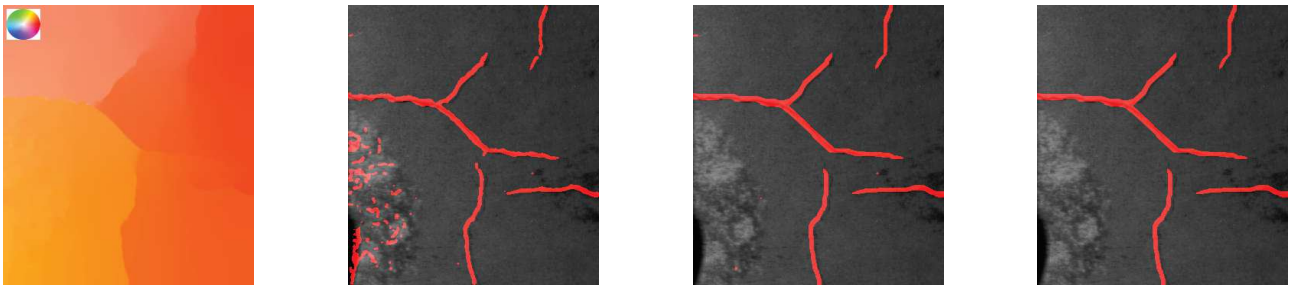
We obtained the CNN model described in the paper [19], trained on nuclear plant crack dataset, from the authors. We did not train the network on bridge images. Overlapping image patches of size 224×224 were chosen with stride of 16 pixels. Softmax response greater than 0.5 was declared as crack and a neighborhood of 16×16 around the current pixel was labeled as crack.

Quantitative results from table 1 shows this method has low recall and high false positive rate compared to our method and Jahanshahi [10] because the CNN was trained on nuclear power plant dataset and it did not generalize well to detect cracks on the bridge surface. Furthermore, the coarse nature of detection resulted in higher false positive detections thereby reducing F1 score. Our method achieves F1 scores improvement of 0.41-0.51 over this method. Qualitative comparison can also be seen in figure 7 which shows coarse detection compared to our method which produces much finer quality detection. These results indicate that good generalization performance for crack detection "in-the-wild" dictate that machine learning methods be trained with large amount of diverse data which is difficult, time-consuming and expensive to obtain.

3.5.2 Comparison with Jahanshahi [10]

We followed the algorithmic details mentioned in the paper [10] to implement the method assuming that depth variation of the crack surface was negligible. Initial detection by morphological transformations were refined by removing connected components, as mentioned in the paper, having less than 244 pixels. Further refinement was obtained by discriminative model which learned geometric feature of cracks. A polynomial SVM kernel of degree 3 was used. Training was done on synthetic crack dataset using the technique described in the paper.

For all cycles, our method outperforms Jahanshahi [10] in F1 score due to false positive detections and failure to detect micro-cracks at early stage. With increasing number of cycles, the F1 score and true positive rate for this method increases which indicates that it is well suited for macro-crack detection clearly visible in the image. Our method achieves F1 scores improvement of 0.14-0.22 over this method.



(a) Motion Visualization (b) NLSTF (c) CRF (d) CRF+Post processing(CC)

Figure 6: From left to right - (a) Optical flow visualization for a particular frame showing motion discontinuity at crack locations, (b) NLSTF response having false positive detections, (c) CRF output successfully removing outliers, (d) Post-processing which removes isolated detections for 1000 cycles.

Figure 7 shows that while our method successfully detects micro-crack in the top right corner of the image for 1000 cycles while this method fails to do so. Moreover, this algorithm confused texture patterns in the bottom left as crack pixel thus producing false positive detections which our method successfully rejects yielding better performance.

3.6. Early stage crack detection

Our method can accurately detect cracks at an early stage which is evident from the high recall and F1 score even at 1000 cycles case. Qualitative analysis from figure 7 indicate that the obtained crack map is very similar to the ground truth map even at an early stage. Furthermore, for both 5000 and 20000 cycles, F1 score is high with good qualitative detection, which provides evidence that our method can detect cracks at an early stage before serious damage occurs. In addition, as demonstrated in previous sections, our motion based system provides accurate and finer crack map detection which can be used by inspectors and engineers to predict crack growth patterns in the future.

4. Conclusion

We have presented an automatic crack detection methods for detecting both visible cracks and micro-cracks at an early stage using motion analysis on concrete bridge structures. We show that motion features on the crack surface can be used to successfully detect cracks that are not clearly visible and hence cannot be detected by conventional image based crack detection techniques. Experimental results demonstrate the effectiveness of our CRF based crack detection algorithm. The proposed method can detect cracks at an early stage using just videos from generic cameras which facilitates economical and frequent maintenance scheduling. It also gives detailed crack map which can be used to predict crack growth in the future. For these reasons, our proposed automatic crack detection system provides a faster, more practical and economical alternative to manual crack inspection. In the future, we hope to design systems that provide high crack localization accuracy using cameras at large distance as well.

Acknowledgment

We are grateful to Research Association for Infrastructure Monitoring System (RAIMS)[1] for sharing the concrete crack dataset. This work was partly supported by Strategic Innovation Promotion Program (SIP), a Japanese project led by the Cabinet Office's Council for Science, Technology and Innovation.

References

[1] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, Nov. 2001.

[2] H. Bruck, S. McNeill, M. A. Sutton, and W. Peters Iii. Digital image correlation using newton-raphson method of partial differential correction. *Experimental mechanics*, 29(3):261–267, 1989.

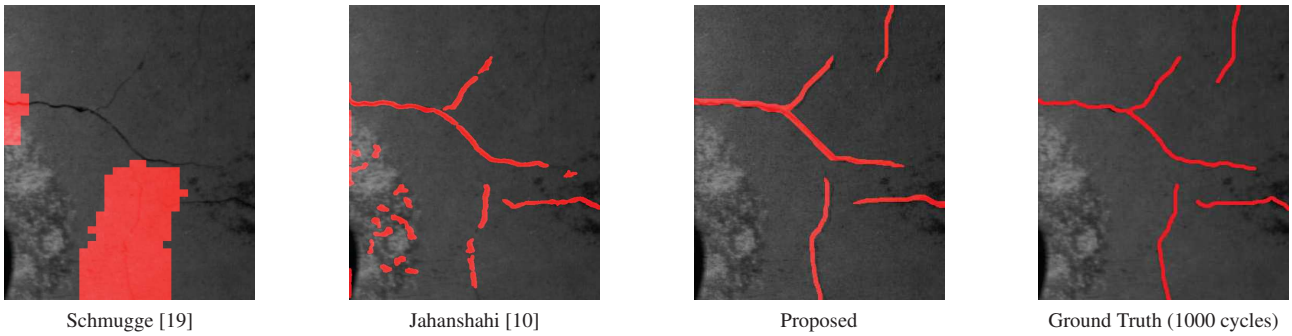
[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1, June 2005.

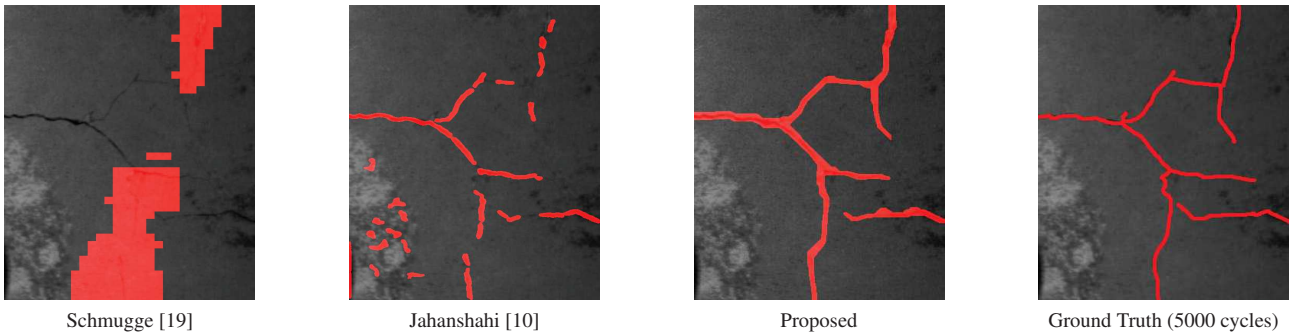
[4] Y. Fujita, Y. Mitani, and Y. Hamamoto. A method for crack detection on a concrete structure. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 3, pages 901–904, 2006.

[5] D. A. Hordijk and H. W. Reinhardt. Growth of discrete cracks in concrete under fatigue loading. In *Toughening mechanisms in quasi-brittle materials*, pages 541–554. Springer, 1991.

[6] T. T. Hsu. Fatigue of plain concrete. In *Journal Proceedings*, volume 78, pages 292–305, 1981.

[1] http://www.raims.or.jp/en/





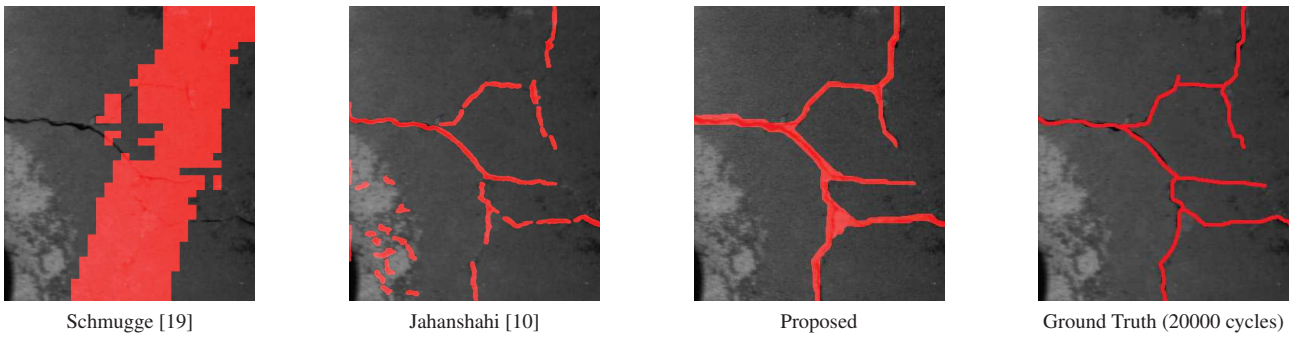


Figure 7: Crack localization by various methods for 1000 cycles (1^{st} row), 5000 cycles (2^{nd} row) and 20000 cycles (3^{rd} row)

[7] H. Hu, Q. Gu, and J. Zhou. Htf: a novel feature for general crack detection. In *2010 IEEE International Conference on Image Processing*, pages 1633–1636, Sept 2010.

[8] Y. Huang and B. Xu. Automatic inspection of pavement cracking distress. *Journal of Electronic Imaging*, 15(1):013017–013017–6, 2006.

[9] T. Hutt and P. Cawley. Feasibility of digital image correlation for detection of cracks at fastener holes. *Ndt & E International*, 42(2):141–149, 2009.

[10] M. R. Jahanshahi, S. F. Masri, C. W. Padgett, and G. S. Sukhatme. An innovative methodology for detection and quantification of cracks through incorporation of depth perception. *Machine vision and applications*, 24(2):227–241, 2013.

[11] C. Koch, K. Georgieva, V. Kasireddy, B. Akinci, and P. Fieguth. A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. *Advanced Engineering Informatics*, 29(2):196–210, 2015.

[12] C. Liu, W. T. Freeman, and E. H. Adelson. *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, Massachusetts Institute of Technology, 2009.

[13] J. Mordock. A critical review of research on the fatigue of plain concrete. *Illinois Univ Eng Exp Sta Bulletin*, 1965.

[14] H. Oliveira and P. L. Correia. Crackit—an image processing toolbox for crack detection and characterization. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 798–802. IEEE, 2014.

[15] J. Poissant and F. Barthelat. A novel "subset splitting" procedure for digital image correlation on discontinuous displacement fields. *Experimental mechanics*, 50(3):353–364, 2010.

[16] J. Roesler and E. Barenberg. Fatigue and static testing of concrete slabs. *Transportation Research Record: Journal of the Transportation Research Board*, (1684):71–80, 1999.

[17] F. Roli. Measure of texture anisotropy for crack detection on textured surfaces. *Electronics Letters*, 32(14):1274–1275, Jul 1996.

[18] J. Rupil, S. Roux, F. Hild, and L. Vincent. Fatigue microcrack detection with digital image correlation. *The Journal of Strain Analysis for Engineering Design*, 46(6):492–509, 2011.

[19] S. J. Schmugge, L. Rice, N. R. Nguyen, J. Lindberg, R. Grizzi, C. Joffe, and M. C. Shin. Detection of cracks in nuclear power plant using spatial-temporal grouping of local patches. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–7. IEEE, 2016.

[20] P. Subirats, J. Dumoulin, V. Legeay, and D. Barba. Automation of pavement surface crack detection using the continuous wavelet transform. In *2006 International Conference on Image Processing*, pages 3037–3040, Oct 2006.

[21] T. Tomikawa. A study of road crack detection by the metagenetic algorithm. In *Africon, 1999 IEEE*, volume 1, pages 543–548 vol.1, 1999.

[22] T. Yamaguchi and S. Hashimoto. Improved percolation-based method for crack detection in concrete surface images. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4, Dec 2008.

[23] T. Yamaguchi, S. Nakamura, R. Saegusa, and S. Hashimoto. Image-based crack detection for real concrete surfaces. *IEEJ Transactions on Electrical and Electronic Engineering*, 3(1):128–135, 2008.