2016 14th International Conference on Control, Automation, Robotics & Vision
Phuket, Thailand, 13-15th November 2016 (ICARCV 2016)

Su43.1

Computer Vision-based Method for Concrete Crack Detection

Tran Hiep Dinh, Q. P. Ha
School of Electrical, Mechanical and Mechatronic Systems
University of Technology Sydney
Sydney, Australia
Email: TranHiep.Dinh@uts.edu.au, Quang.Ha@uts.edu.au

H. M. La
Department of Computer Science and Engineering
University of Nevada
Reno, USA
Email: hla@unr.edu

***Abstract*— This paper presents a computer vision-based method to automatically detect concrete cracks. We focus on images containing the concrete: background and crack, where the background is the major mode of the gray-scale histogram. Therefore, we address the detection problem of potential concrete cracks by dealing with histogram thresholding to extract regions of interests from the background. We first employ line emphasis and moving average filters to remove noise from concrete surface images obtained from an inspection robot. The developed algorithm is then applied for automatic detection of significant peaks from the gray-scale histogram of the smoothed image. The biggest peak and its corresponding valley(s) are consequently identified to calculate the threshold value for image binarization. The effectiveness of our proposed method was successfully evaluated on various test images, where cracks could be identified without the requirement of some heuristic reasoning.**

Keywords- Concrete crack; histogram thresholding; image segmentation

I. Introduction

With the development of robotic and computer vision systems, tedious, expensive and information intensive tasks of infrastructure monitoring such as road or bridge inspection could be done by robots. Effectively and periodically monitoring could help in early detection of cracks to avoid structural damage, and hence, potential accidents.

Many vision-based approaches have been proposed to deal with robotic color tracking and image segmentation [1-4] and employed to solve the crack detection problem [5-14]. Most of these crack identification methods are based on employing edge-detection techniques, such as the fast Haar transform, fast Fourier transform, or the Sobel and Canny operator [5]. Yu [6] proposed a semi-automatic strategy by combining Sobel and Laplacian operators to find crack edges and employing a graph search method to extract cracks based on user's input. Another semi-automatic approach is introduced in [7], where the detected cracks are traced bi-directionally before being merged to accomplish the final result.

Simple edge detection techniques are however only robust to high contrast images with a nearly uniform background [12]. Various hybrid approaches have been developed to enhance the automatic detection results from concrete images [8-12]. Sinha [8] fused data obtained from a ratio edge detector [9] and a variance comparison [10] to successfully detect cracks from underground pipe images. A gradient based crack detector was developed in [11] and demonstrated promising results in highly noisy concrete images. The machine vision and optimization approach are applied in [12], where cracks in various appearances could be successfully detected without any adjustments needed for threshold parameters. A multi-scale line emphasis filter based on a 3-D Hessian matrix is suggested in [13] for medical images. An automatic method was developed to deal with the problem of concrete crack detection using 2-D images in [14]. Therein, the eigenvalues and eigenvectors of the Hessian matrix are combined to extract line structures from noisy concrete surface images.

In this paper, we propose an automatic peak detection algorithm for image segmentation to apply to the concrete crack detection problem. Scanned images with a crack potential are firstly processed using the line emphasis filter developed in [14]. The gray-scale histogram of the processed image is then smoothed by using a moving average filter, and analyzed to detect significant peaks based on two dynamic parameters: the offset distance and crossover index. These parameters are determined from the detected initial peaks and each updated mobile origin, without any heuristic criterion or user's supervision. The segmentation result is then employed for image binarization. The effectiveness of our proposed algorithm is evaluated through various test data, from high-contrast to very noisy background images.

This paper is organized as follows. Section II presents a comprehensive review of the proposed approach with two main steps: preprocessing and automatic peak detection. In Section III, we present the experimental results on test data using a variety from high-contrast to high-clutter images. Finally, Section IV summarizes the contribution and draws a conclusion of this paper.

978-1-5090-3549-6/16/$31.00 ©2016 IEEE

II. Methodology

At the first stage, the image of a concrete surface is preprocessed to eliminate the blob-like noise. The corresponding histogram is then smoothed using a moving average filter, before being analyzed to detect significant peaks. After the preprocessing step, the resulting image with background and potential structural cracks is to be segmented by employing our automatic peaks detection algorithm. The segmentation result is then enhanced using morphological operations to remove small uninterested regions. Figure 1 illustrates the processing steps of our proposed approach.

A. Preprocessing

1) Line emphasizing

In this step, we implement a line emphasis filter, proposed in [13] and [14], based on the Hessian matrix, to remove blob-like and sheet-like noises and feature the line structure corresponding to a crack. The Hessian matrix of an image *I(x)*, where $\boldsymbol{x} = (x, y)$, is given by:

$$\nabla^2 I(\boldsymbol{x}) = \begin{bmatrix} I_{xx}(\boldsymbol{x}) & I_{xy}(\boldsymbol{x}) \\ I_{yx}(\boldsymbol{x}) & I_{yy}(\boldsymbol{x}) \end{bmatrix}, \tag{1}$$

in which

$$\begin{cases} I_{xx}(\boldsymbol{x}) = \dfrac{\partial^2}{\partial x^2} I(\boldsymbol{x}) \\ I_{xy}(\boldsymbol{x}) = \dfrac{\partial^2}{\partial x \partial y} I(\boldsymbol{x}) \\ I_{yy}(\boldsymbol{x}) = \dfrac{\partial^2}{\partial y^2} I(\boldsymbol{x}) \\ I_{yx}(\boldsymbol{x}) = \dfrac{\partial^2}{\partial y \partial x} I(\boldsymbol{x}). \end{cases} \tag{2}$$

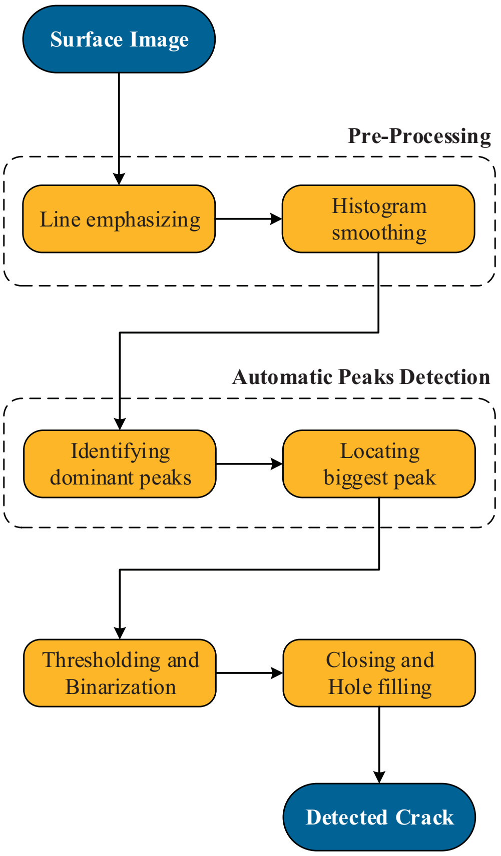


Figure 1. Proposed approach for concrete crack detection.

The eigenvalues $\lambda_1(\boldsymbol{x}), \lambda_2(\boldsymbol{x})$ of $\nabla^2 I(\boldsymbol{x})$ (where $\lambda_1 \geq \lambda_2$) are adopted to acquire a generalized measure of similarity to a line structure as follows:

$$\lambda_{12} = \begin{cases} |\lambda_2|(1 + \dfrac{\lambda_1}{|\lambda_2|}) = |\lambda_2| + \lambda_1 \\ \text{if} \quad \lambda_2 \leq \lambda_1 \leq 0 \\ |\lambda_2|(1 - \alpha\dfrac{\lambda_1}{|\lambda_2|}) = |\lambda_2| - \alpha\lambda_1 \\ \text{if} \quad \lambda_2 < 0 < \lambda_1 < \dfrac{|\lambda_2|}{\alpha} \\ 0 \quad \text{otherwise}, \end{cases} \tag{3}$$

where $0 < \alpha \leq 1$.

The line-, blob- or sheet structure in the image could be expressed by combining two eigenvalues λ_1 and λ_2 as described in Table I.

In order to recover line structures of various widths, the partial second derivatives of the image *I(x)* in (2) can be combined with the Gaussian convolution as,

$$I_{xx}(\boldsymbol{x}; \sigma_f) = \frac{\partial^2}{\partial x^2}\{G(\boldsymbol{x}; \sigma_f)\} * I(\boldsymbol{x}), \tag{4}$$

where $G(\boldsymbol{x}, \sigma_f)$ is an isotropic Gaussian function with standard deviation σ_f. The maximum among the normalized multiple scales will be selected from the multi-scale integration of the filter responses λ_{12} of a pixel $\boldsymbol{x}$ within a region defined by $R(\boldsymbol{x})$ as:

$$R(\boldsymbol{x}) = \max_{\sigma_i} \sigma_1^2 \lambda_{12}(\boldsymbol{x}; \sigma_i), \tag{5}$$

where $\sigma_i = s^{i-1}\sigma_1 \quad (i = 1, 2, ..., n)$ is a discrete sample of σ_f, with σ_1 and s being the minimum scale and scale factor of the sampling interval of σ_f, respectively.

TABLE I. COMBINATION OF EIGENVALUES AND CORRESPONDING SHAPE STRUCTURE

Relationships between eigenvectors	Structure
$\lvert\lambda_1\rvert \geq \lvert\lambda_2\rvert \approx 0$	Line
$\lvert\lambda_1\rvert \approx \lvert\lambda_2\rvert \geq 0$	Blob
$\lvert\lambda_1\rvert \approx \lvert\lambda_2\rvert \approx 0$	Sheet

2) *Histogram smoothing*

The gray-scale histogram of the processed image is then smoothed using a moving average filter with the kernel width equal to 3, taking into account the previous, current and next intensity level. This is the best filter candidate for our automatic peak detection approach because of its compactness and robustness in eliminating random noise while retaining significant peaks of the histogram. Let $h(i)$ be the pixel number at intensity level i for $i = 0, 1, ..., L$; where L is the maximum intensity level. After applying the moving average filter, the pixel number at intensity level i is determined as:

$$h(i) = \frac{1}{3}\left[h(i-1) + h(i) + h(i+1)\right]. \tag{6}$$

A peak of the smoothed histogram can be identified if it meets the following condition:

$$\begin{cases} h(i) > h(i+1) \\ h(i) > h(i-1). \end{cases} \tag{7}$$

These initial peaks are then stored in a cluster vector δ. This cluster vector could be considered as a draft plan, from which the significant peaks detection strategy will be built. Condition (7) can therefore be represented as:

$$\begin{cases} h(\delta(k)) > h(\delta(k)+1) \\ h(\delta(k)) > h(\delta(k)-1). \end{cases} \tag{8}$$

B. *Automatic peak detection algorithm*

Our proposed algorithm is inspired by a practical problem in mountain exploration. When an explorer gets lost in an unknown forest or mountain, the realistic solution is to localize a nearest highest position to access an unobstructed view before planning for the next travel segment. By repeating this "finding and moving" process, the explorer could finally go through the mountain and accomplish the journey. A flowchart of the mentioned strategy is described in Fig. 2. Our proposed algorithm is to split into two main steps: i) searching for observing location, and ii) searching for the highest peak in the closest distance.

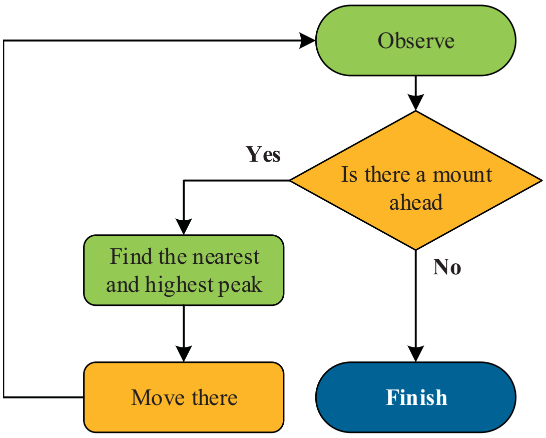


Figure 2. Strategy for a lost mountain explorer.

1) *Observing location*

Let consider the smoothed histogram as the mount to be explored. A point on the intensity axis could then be considered as the observing location $\alpha(m)$ if the following condition is fulfilled:

$$\alpha(m) < g(\delta(k)) - L(\delta(k)), \tag{9}$$

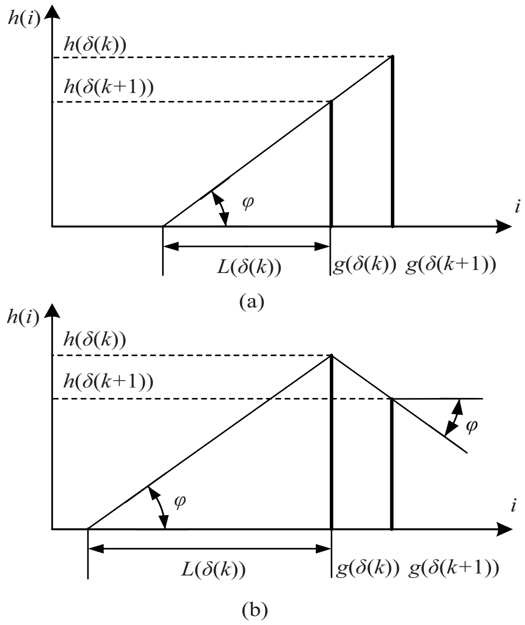
where $L(\delta(k))$ is a distance to be defined from the current peak to the observing location and dependent on the draft plan, $g(\delta(k))$ is the intensity level at the kth initial peak and p_D is the number of possible dominant peaks, $1 \le m \le p_D$. Now, let us define the offset distance:

$$L(\delta(k)) = \begin{cases} \dfrac{h(\delta(k))[g(\delta(k+1)) - g(\delta(k))]}{\left|h(\delta(k+1)) - h(\delta(k))\right|} \\ \text{if} \qquad h(\delta(k+1)) \neq h(\delta(k)) \\ \dfrac{h(\delta(k))[g(\delta(k+1)) - g(\delta(k))]}{\left|\dfrac{k+1}{k} h(\delta(k+1)) - h(\delta(k))\right|} \\ \text{if} \qquad h(\delta(k+1)) = h(\delta(k)). \end{cases} \tag{10}$$

It can be seen that this offset distance $L(\delta(k))$ is determined based on the two adjacent intensity levels and their corresponding pixel number, as illustrated in Fig. 3. Based on the height difference and distance between two adjacent peaks, a higher peak could always be identified from the calculated observing location.

In the implementation phase, the observing location is set at $g(\delta(k)) - L(\delta(k))$, hence:

$$\alpha(m) = g(\delta(k)) - L(\delta(k)), \tag{11}$$



(a)

(b)

Figure 3. Offset distance determination:
(a) $g(\delta(k+1)) > g(\delta(k))$
(b) $g(\delta(k+1)) < g(\delta(k))$.

and the observing location is then placed “higher” than the intensity axis to maintain the unobstructed view. For the *k*th detected peak in the cluster vector, the crossover index is proposed as follows:

$$\theta(\delta(k)) = \frac{d(\delta(k))}{L(\delta(k))}, \tag{12}$$

where

$$d(\delta(k)) = h(\delta(k)) - \frac{\min\{h(\delta(k)), h(\delta(k+1))\}}{2}.$$

2) Highest peak in the closest distance

A peak is considered as a nearest prominent peak, if it is high enough to be “observed” from the current observing location and to “block” the view to the next peak. This statement can be mathematically described as follows:

$$\begin{cases} \theta(\delta(k)) > \theta(\delta(k+1)) \\ \theta(\delta(k)) > \theta(\delta(k-1)). \end{cases} \tag{13}$$

The searching mechanism of our algorithm is illustrated in Fig. 4 when $g(\delta(k))$ and $g(\delta(k+3))$ are identified as dominant. Firstly, $g(\delta(k))$ is identified as a dominant peak. The new location for observation will be updated at $g(\delta(k)) - L(\delta(k))$. The crossover index of $g(\delta(k))$ is reset while this characteristic of $g(\delta(k+1))$, $g(\delta(k+2))$, … is re-calculated based on the updated observing location. A following dominant peak is then determined at $g(\delta(k+3))$ as demonstrated in Fig. 4.

A new observing location is calculated and the crossover index of the next peak is consequently updated according to (10–12), until the cluster vector is completely checked. The block of Fig. 5 illustrates the implementation of the proposed algorithm pseudo-code.

Once a set of dominant peaks of the processed image are identified, we locate the position of the biggest peak. Because all blob-like and sheet-like structures have been removed, the remaining objects are the background, potential crack and small uninterested objects. We calculate the valleys by determining the minimum value between two adjacent peaks and use the valley(s) of the highest peak as the threshold for image binarization. Morphological operations are then conducted to eliminate the small uninterested objects and finally extract the crack from the processed image.

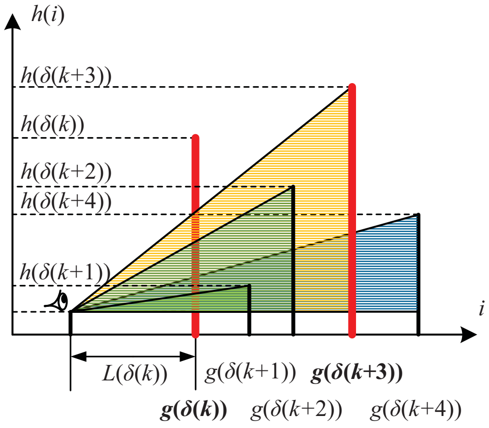


Figure 4. Illustration of searching mechanism.

```
$m = 1; n = 1; found = 0;$
for $k = n{:}length(\delta)$ do
  /* Calculate peak's crossover index
  for $j = (n+1){:}(k-1)$ do
    if there is a dominant peak at level $j$ then
      $n = j+1;$
      $found = 1;$
      /* Save the position of dominant peak in $\beta$
      $\beta(m) = \delta(j);$
      /* Update the position of dominant peak
      $m = m+1;$
      $\alpha(m) = g(\delta(j)) - (L(\delta(j));$
      /* Break out of for loop
      break;
    end
  end
  if found then
    fprintf('Dominant peak at position %d', j);
    fprintf('New observe position at %d', $\alpha(m)$);
  else
    fprintf('No dominant peak at this step');
  end
end
```

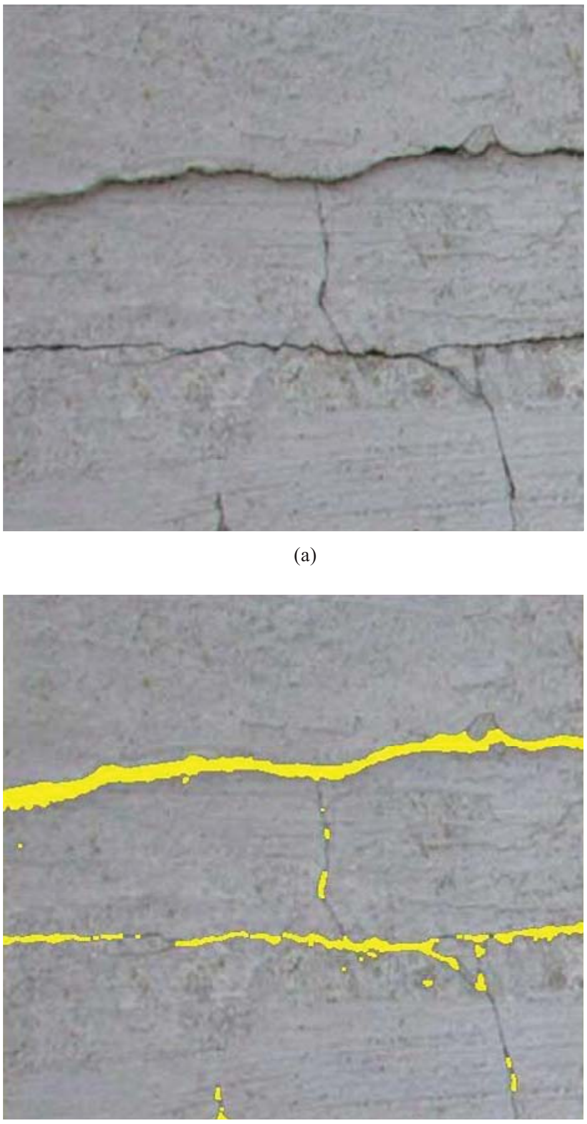
Figure 5. Implementation of automatic peak detection algorithm.

III. Results and Discussion

In this section, we evaluate our proposed algorithm on various concrete surface images, which were collected from the autonomous robotic NDE system [11]. The image backgrounds vary from nearly uniform to very noisy.

Figure 6(a) is the test image where the background is nearly uniform and has a high contrast against cracks. The cracks in this image are also visibly inherent and could be effectively detected with the thresholding technique mentioned in Section II.B. The detection result in Fig. 6(b) illustrates the extracted cracks, where all visible structures are successfully isolated.

Figures 7(a) and 8(a) are test images with low contrast between the background and cracks as well as an increased level of noise. Figures 7(b) and 8(b) highlight the extracted cracks from both images. Even with a decrease in the visibility and width of the cracks, these interested objects are still adequately isolated as displayed. Because there is more clutter in Fig. 8(a), some blebs and shadings still remain after the cleaning process.

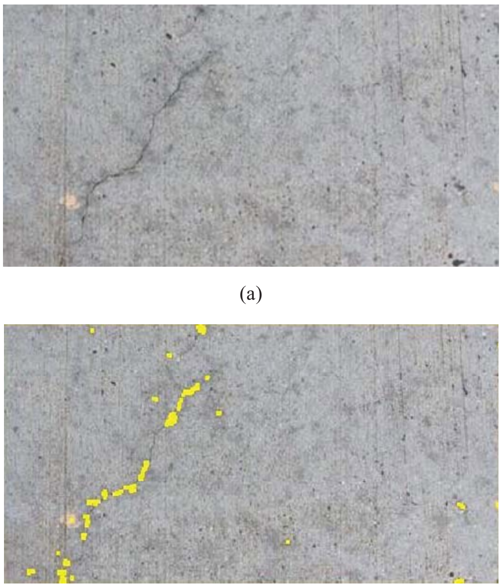


(a)

(b)

Figure 6. Test result on high contrast and low clutter image:
(a) Original image
(b) Crack detection result.

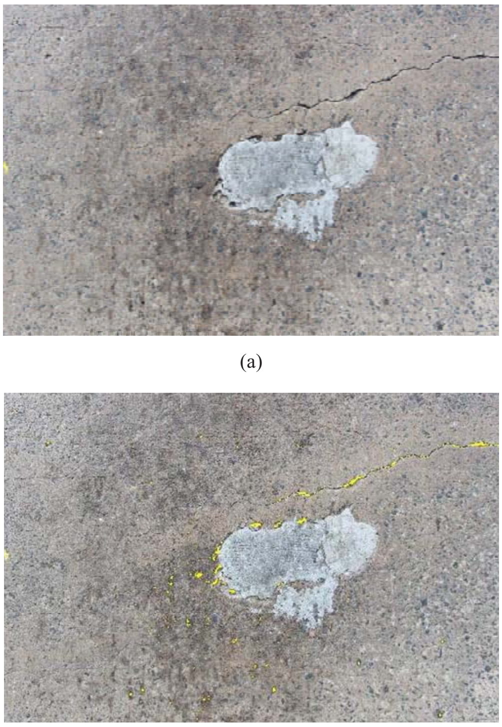
It is significant to see that the crack structure in all test images can be quite accurately detected from distinct surfaces, varying from high to low contrast and low to high clutter images. Many vision-based crack detection methods use a median filter in the pre-processing step [7, 14] to subtract the smoothed image from its origin. This approach is, however, only effective with a specific filter size for a specific image, which is not applicable for real-world images of concrete in construction. Moreover, the thresholding step to extract cracks from the background requires another phase of some manual adjustments, which leads to a non-robust and heuristic solution. Compared with these techniques, our proposed peak detection algorithm could overcome these disadvantages by employing image gray-scale histograms, to extract dynamic parameters for thresholding. The filter span for histogram smoothing described in Section II.A.2 is limited at a minimum level to illustrate the robustness and effectiveness of our proposed algorithm on rough data.



(a)

(b)

Figure 7. Test result on low contrast image
(a) Original image
(b) Crack detection result.



(a)

(b)

Figure 8. Test result on low contrast and high clutter image:
(a) Original image
(b) Crack detection result.

IV. CONCLUSION

This paper has presented an effective vision-based method for concrete crack detection. The proposed approach has been evaluated on various images, and can successfully extract visible cracks from the background of images captured by an inspection robot. The obtained results are consistent from different data sources with a distinct number as well as the visibility level of the detected crack. The threshold for image binarization is employed from a non-parametric peak detection algorithm. Therefore, our proposed method could be considered as fully automatic, and as such, it could substantially enhance the effectiveness of a robotic inspection process. More importantly, there is also an option for some crack classification by analyzing the remaining detected peaks after the thresholding step.

REFERENCES

[1] Y.-H Yu, N.M. Kwok, Q.P. Ha, “Color tracking for multiple robot control using a system-on-programmable-chip”, *Automation in Construction*, vol. 20, no. 6, pp. 669-676, 2011.

[2] P. K. Sahoo, S. Soltani and A. K. C. Wong, "A survey of thresholding techniques", *Computer Vision Graphics and Image Processing*, vol. 41, pp. 233-260, 1988.

[3] T. H. Dinh, M. T. Pham, M. D. Phung, D. M. Nguyen, V. M. Hoang and Q. V. Tran, “Image segmentation based on histogram of depth and an application in driver distraction detection”, in *Proceedings of the 13th International Conference on Control, Automation, Robotics and Vision (ICARCV14)*, pp. 969-974, 2014.

[4] 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*, vol. 29, no. 2, pp. 196-210, 2015.

[5] I. Abdel-Qader, O. Abudayyeh and M.E. Kelly, "Analysis of edge-detection techniques for crack identification in bridges", *Journal of Computing in Civil Engineering*, vol. 17, no. 3, pp. 255-263, 2003.

[6] S.N. Yu, J.H. Jang and C.S. Han, "Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel", *Automation in Construction*, vol. 16, pp. 255-261, 2007.

[7] J.K. Oh, G. Jang, S. Oh, J.H. Lee, B.J. Yi, Y.S. Moon, et al., "Bridge inspection robot system with machine vision", *Automation in Construction*, vol. 18, pp. 929-941, 2009.

[8] S.K. Sinha and P.W. Fieguth "Automated detection of cracks in buried concrete pipe images", *Automation in Construction*, vol. 15, no. 1, pp. 58-72, 2006.

[9] A. Lopes, E. Nezry, R. Touzi and H. Laur, "Structure detection and statistical adaptive speckle filtering in SAR images", *International Journal of Remote Sensing*, vol. 14, no. 9, pp. 1735-1758, 1993.

[10] Y. Yakimovsky, "Boundary and object detection in real world images", *Journal of ACM*, vol. 23, no. 4, pp. 599-618, 1976.

[11] H. M. La, N. Gucunski, Seong-Hoon Kee, J. Yi, T. Senlet and Luan Nguyen, "Autonomous robotic system for bridge deck data collection and analysis" in *Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1950-1955, 2014.

[12] P. Prasanna, K. J. Dana, N. Gucunski, B. B. Basily, H. M. La, R. S. Lim, and H. Parvardeh., “Automated crack detection on concrete bridges”, *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 591-599, 2016.

[13] Y. Sato, S. Nakajima, N. Shiraga, H. Atsumi, S. Yoshida, T. Koller, et al., "Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images", *Medical Image Analysis*, vol. 2, no. 2, pp. 143-168, 1998.

[14] Y. Fujita and Y. Hamamoto, "A robust automatic crack detection method from noisy concrete surfaces", *Machine Vision and Applications*, vol. 22, pp. 245-254, 2011.