Original Article

SHM Structural Health Monitoring

Structural Health Monitoring
2014, Vol. 13(2) 205–218

DOI: 10.1177/1475921713517289
shm.sagepub.com
SAGE

Concrete crack detection through full-field displacement and curvature measurements by visual mark tracking: A proof-of-concept study

Zhaozheng Yin[1], Chenglin Wu[2] and Genda Chen[2]

Abstract

In this study, a noncontact vision-based sensing method is proposed to measure surface displacements and curvatures and to detect cracks in a reinforced concrete slab. The proposed method includes five independent modules for structure boundary identification by a dynamic programming algorithm, boundary movement tracking by a contour tracking algorithm, distinguishable surface feature detection by speed-up-robust features, feature (visual mark) tracking by a three-stage data association algorithm, and displacement interpolation from those at visual marks by a Delaunay triangulation algorithm. The displacement field was used to evaluate the slab curvature that functioned as a crack indicator. The proposed data association algorithm for visual mark translation, linking, and connection was successfully applied for visual mark tracking of concrete slab images. The proposed algorithms used in five modules are computationally efficient, making them viable tools for real-time structural health monitoring. By persistently tracking the features and positions of spatially distributed visual marks in time-lapse videos, the displacement time histories at mark locations are successfully evaluated. The relative error of displacement measurements for the tested concrete slab is approximately 1.24%. The proposed method was applied to successfully detect cracks of a full-scale reinforced concrete slab from image analysis. Unlike contact measurements, the proposed noncontact measurement is not affected by concrete cracking.

Keywords

Noncontact measurement, visual marks, feature detection and tracking, data association algorithm, displacement and curvature, crack detection

Introduction

The responses of structures under external loads are conventionally measured with sensors that are directly attached to the structures. For example, linear variable differential transformers (LVDTs) and strain gauges are typically used for displacement and strain measurements. Each of the attached sensors only provides one local measurement at a time, and a significant number of them are often required for a complete structural evaluation of a large-scale civil infrastructure, which could amount to a cost-prohibitive sensing system in practical applications. In addition, installing sensors on a large structure requires hours of labor from professional technicians particularly for difficult-to-access areas, including the physical attachment and protection of sensors, and power and data transmission cabling. The long cablings, instruments, and operators of the sensing system may be displeased and even disruptive to the daily operation of the instrumented structure. Furthermore, the maintenance and associated reliability of the sensing system for long-term monitoring is a challenge.

With the rapid development of image acquisition systems/cameras and computational algorithms, noncontact measurements with image analysis have gained increasing acceptance in experimental mechanics. For example, the digital image correlation (DIC) technique has been applied to monitor the dynamic characteristics of structures with video images.[1–5] For proper DIC

[1]Department of Computer Science, Missouri University of Science and Technology, Rolla, MO, USA
[2]Department of Civil, Architectural, and Environmental Engineering, Missouri University of Science and Technology, Rolla, MO, USA

Corresponding author:
Genda Chen, Department of Civil, Architectural, and Environmental Engineering, Missouri University of Science and Technology, Rolla, MO 65409, USA.
Email: gchen@mst.edu

measurements, the surface of a test specimen must be well prepared (e.g. painted) to achieve a random gray intensity distribution on the specimen surface. Virtual grid lines are applied to the first captured image, and intensity patches are extracted around the intersections of the grid. At each loading level, the location of each patch in the subsequent recorded image is searched iteratively and exhaustively within a small area centered at the original position by image correlation and peak-finding algorithms. The high computational cost in the iterative and exhaustive search may make the DIC technique impractical for long-term and real-time structural health monitoring. More importantly, technical difficulties can arise when the deformation/rotation of an intensity patch between two consecutive images becomes excessive, when images are captured in uncontrolled field environments with intermittent visual blocks, and when cameras are temporarily malfunctional.[5]

An alternative vision-based measurement technique was recently attempted with preliminary results.[6] In their study, the vertical displacement of concrete/steel beams was determined from images taken from video cameras. Each beam was painted white for easy image segmentation from its background color in LabVIEW. Its edge was detected from each image, and the edge difference between two consecutive images represented the displacement increment over time. The preliminary study showed the potential of noncontact measurements by cameras in structural health monitoring. However, point correspondences between consecutive images were not attempted, and thus, the effectiveness of the alternative technique for measurements of any displacement perpendicular to the beam edge is yet to be seen.

In this study, a new noncontact sensing method with off-the-shelf cameras is proposed to measure the full-field displacement of a structure. The profiles/boundaries of a structure are extracted and tracked by using a dynamic programming algorithm and a contour tracking algorithm, respectively. Visual marks on the structure are identified at distinctive pixel locations in the images and represented by Speeded-Up Robust Feature (SURF) descriptors.[7] A three-stage data association algorithm based on the linear programming optimization is developed to persistently track multiple visual marks in time-lapse image sequences. The movement trajectories of the visual marks in the spatial–temporal domain represent the displacement of their corresponding physical points on the structure surface. Finally, by a simple spatial interpolation from the movement of several nearby marks, the full displacement field of the structural surface can be constructed.

Compared with point sensors such as strain gauges, the proposed vision-based sensing method provides remote and noninvasive measurements for the full-field displacement. The method is more cost-effective in practical applications. It fundamentally differs from the DIC technique in that various visual marks instead of a patch of pixels are tracked so that the deformed marks can be detected and tracked accurately even under excessive displacement or intermittent occlusion.

The proposed visual mark tracking method

To measure the displacement of a structure, a video camera was set in front of the structure to take images perpendicular to a surface of the structure in time-lapse sequences as the structure was deformed under loading. If a camera is set near a structure in certain angle with the exterior face of the structure in practical applications, all images must be adjusted to account for the effect of the imaging angle. The proposed sensing method consists of the following five modules as illustrated with testing of a 6.1-m-long, 2.743-m-wide, and 0.254-m-thick concrete slab under three-point loading.

Profile initialization

The profiles of a structure are defined as the boundaries between the structure surface and its surrounding background environment. For example, the concrete slab as shown in Figure 1 has top and bottom profiles. Due to their relatively high image contrast, the profiles were extracted in the gradient domain of an image where the image pixels with high gradients can be located. In this module, a color image was converted to a gray scale image for gradient computation.

Let (x, y) be the Cartesian coordinates of an image pixel with intensity $I(x, y)$. The partial derivatives of $I(x, y)$ with respect to x and y give a two-component gradient vector, $[g_x(x, y), g_y(x, y)]$, respectively, which were approximated in this study by the finite central differences along grid lines of the image. The magnitude $mag(x, y)$ and normal direction $\theta(x, y)$ of the gradient vector can be computed by

$$mag(x, y) = \sqrt{g_x(x, y)^2 + g_y(x, y)^2} \tag{1}$$

$$\theta(x, y) = tan^{-1}\left(\frac{g_y(x, y)}{g_x(x, y)}\right) \tag{2}$$

where $tan^{-1}()$ is the inverse tangent function. As shown in Figure 1(b) or (c), the gradient magnitude of the concrete slab image in Figure 1(a) indicated that (a) some pixels with high gradients are close to each other, forming a short line, and (b) some short lines can be connected into a long curve, for example, along a profile of the slab. To search for the long profiles, dynamic

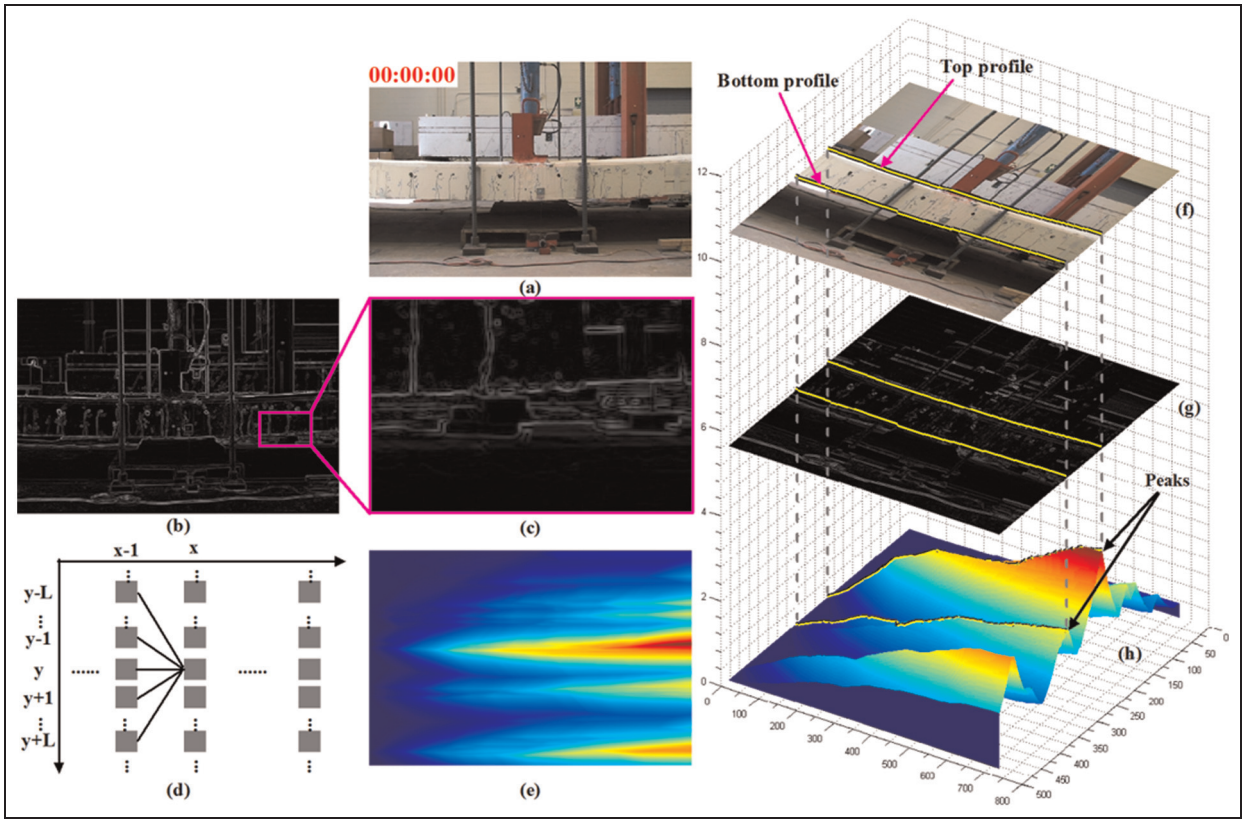

Figure 1. Profile extraction: (a) the first image of an image sequence, (b) the gradient magnitude of (a), (c) a zoom-in portion of (b), (d) a regular grid graph at a pixel coordinate system, (e) the state values of all pixels and (f–h) the back-traced optimal image paths or profile locations in yellow.

programming was applied to find the optimal image paths with high image gradients.

The location of each pixel as shown in Figure 1(d) is referred to as a node in a grid graph. The state of each node was defined by the maximum cumulative gradient magnitude along a path from the leftmost to the current node. In the case of vertical or inclined profiles, the images must be rotated accordingly before this technique can be applied. Specifically, let matrix $S_{N_x \times N_y}$ store the state values of all nodes where N_x and N_y denote the grid sizes in horizontal (x-) and vertical (y-) directions (i.e. image resolution), respectively. Each element of the matrix represents the state value of corresponding node $S(x, y)$ and can be computed by

$$S(x,y) = \max_{\delta \in [-L,L]} \left\{ S(x-1, y+\delta) + mag(x,y) \times e^{-\frac{(\theta(x,y)-\theta(x-1,y+\delta))^2}{\mu}} \right\} \quad (3)$$

where $[-L, L]$ denotes the search range and μ is the mean of all possible differences among the normal directions of neighboring nodes. Equation (3) represents a weighted summation of gradient magnitudes in search of a path with the maximum cumulative gradient magnitude with the weight defined as the exponential function of the difference in normal directions. In other words, if $\theta(x, y)$ is similar to $\theta(x-1, y+\delta)$, $e^{-(\theta(x,y)-\theta(x-1,y+\delta))^2/\mu}$ approaches to 1. Otherwise, the weight is much smaller. Figure 1(e) shows the computed states of all nodes with $L = 1$.

As shown in Figure 1(h), multiple peaks exist on the last column of the grid graph. Some of the peaks correspond to image paths in the cluttered background. For example, the bottom peak in Figure 1(h) corresponds to the electrical cord on the ground. In this case, human interaction is needed to identify the structural profiles. Finally, the optimal image path was back-traced from the right to left side of the image in a reverse order of the path search direction by equation (3). As shown in Figure 1(f) to (h), the identified structure profiles of Figure 1(a) are closely correlated with the actual boundaries of the structure, indicating good performance of the developed algorithm.

Profile tracking

A contour tracking algorithm was developed to automatically track the structure profiles from time-lapse

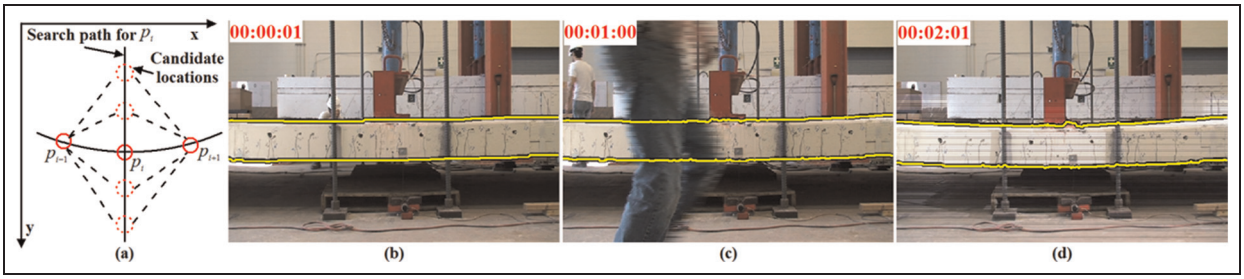

Figure 2. Structure profiles tracking: (a) the search path of a point on the contour and (b–d) example tracking results on the concrete slab profiles from images.

images in sequence, starting from the profile in the previous image. The current position of the profile was iteratively adjusted until it matched the high-contrast boundaries of the structure in the current image. To achieve this goal, an energy function was introduced and associated to each possible location and shape of the profile.

Two forces are considered to change the profile and thus the energy function: (a) image gradient (called external force) and (b) profile curvature (called internal force). The external force attracts the profile to the pixel locations with high gradients and is thus defined by the gradient term of the energy function $E_g(p_i)$ as

$$E_g(p_i) = -mag(p_i) \tag{4}$$

where $p_i = (x_i, y_i)$, $i \in [1, N_x]$, defines the ith pixel along a horizontal profile. The larger the gradient, the lower the energy. The internal force penalizes the high curvature and preserves the smoothness of a curve. Thus, the smoothness term of the energy function $E_s(p_i)$ is defined as

$$E_s(p_i) = \|p_{i-1} - 2p_i + p_{i+1}\|^2 \tag{5}$$

where the second derivative of the pixel locations is a measure of the profile curvature. The total energy of a profile can then be calculated by

$$E = \sum_{i=1}^{N_x} (E_g(p_i) + \omega E_s(p_i)) \tag{6}$$

where ω is a weight factor to balance the influences of the gradient and smoothness terms.

Given its initial position, the profile is updated by minimizing the total energy in sequence, one point at a time from the left to right of an image as shown in Figure 2(a). At each point p_i, several up-down moves were considered to find the new location of the profile that corresponds to the minimum energy. The profile was iteratively updated until the change between the energies of two consecutive profiles is within a predetermined tolerance level.

To avoid numerical instability, the two energy terms are individually normalized in the same fashion so that both vary from 0 to 1. For example, E_g is normalized as $(E_g - m)/(M - m)$, where M and m denote the maximum and minimum gradients along the search path, respectively. Figure 2(b) to (d) shows the tracked structure profiles of time-lapse images with $\omega = 1$. Since both gradient and smoothness terms are considered in the energy function, the profiles are accurately tracked even when the image in Figure 2(c) was partially blocked as people walked by the test structure, and when the illumination condition in Figure 2(d) was changed flashing lights were used during video recording.

Visual mark detection

To measure the displacement of a concrete surface, reference points (i.e. marks) on the surface must be selected and tracked over time. The marks must be distinctive to their surrounding background.[8,9] Otherwise, they cannot be reliably tracked over time. For example, a point on a textureless surface is very hard to track. Manual selection of marks is inefficient and prone to inter- and intra-person variability. More importantly, the manually selected marks are difficult, if not impossible, to distinguish and evaluate objectively and quantitatively.

In this study, an image feature extraction algorithm called SURF[7] is adopted to automatically select distinctive marks in images (referred to as visual marks in this article). Visual marks are selected at pixel locations and discriminative and identifiable in images. To describe each visual mark, a distinctive feature vector is then generated based on the information around it so that the vector is robust to imaging noise, geometric and photometric deformation such as rotation and illumination changes.

Figure 3 shows examples of detected visual marks. The centroid, size and short line within each circle

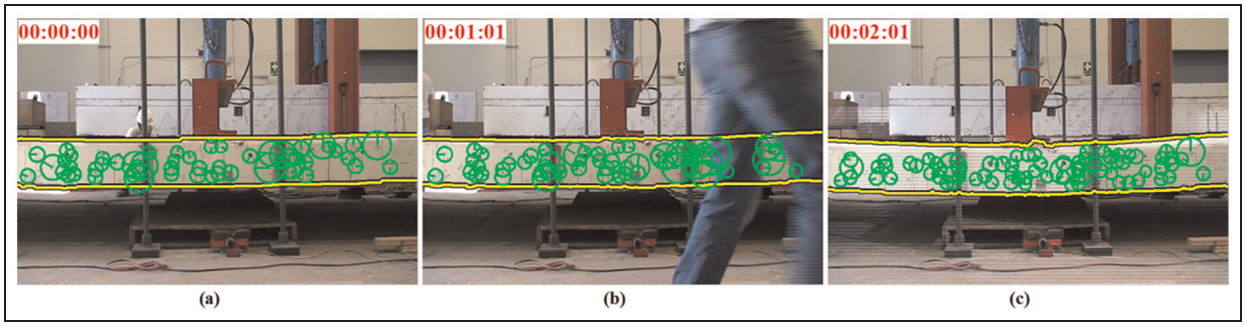

Figure 3. Extracted marks under (a) normal condition, (b) visual obstruction, and (c) illumination change.

represent each visual mark's location, scale and dominant gradient orientation, respectively. The detected visual marks at pixel locations are easy to identify, and physically, they are coincident with special patterns on the concrete surface such as notable image contrast changes and surface texture discontinuous. It can be observed from Figure 3 that some marks are consistently extracted from image sequences, but others may temporarily disappear due to visual obstructions as illustrated in Figure 3(b) or sudden illumination changes as illustrated in Figure 3(c).

Visual mark tracking

In this study, a three-stage data association algorithm is proposed as schematically illustrated in Figure 4. The overall image sequence is divided into many subsequences $[N_{k-1}^{frame}+1,\ N_k^{frame}]$, where $N_0^{frame}=0$ and $k = 1, 2, 3, \ldots$. In the first stage, visual marks between two consecutive images within an image subsequence are matched and connected into a short trajectory, referred to as a *tracklet*. Due to potential vision obstruction or illumination change, some marks temporarily disappear in the current image and reappear in the later images, resulting in broken short trajectories in the spatial–temporal domain. For example, the visual marks from image 1 to image N_1^{frame} are matched and connected to form 6 tracklets $\{T_1, \ldots, T_6\}$. In the second stage, the broken trajectories or tracklets within the subsequence are linked into a long trajectory, referred to as a *trunklet*. For example, the broken tracklets $\{T_4, T_5, T_6\}$ from the first stage are linked into a long Trunklet U_4. In the third stage, the trunklets between two consecutive image subsequences are matched and connected in sequence. The trunklets from all image subsequences are related into a complete trajectory over time or a displacement time history.

Dividing a long video sequence into a set of short subsequences allows the implementation of a divide-and-conquer strategy to (a) reduce a large-scale optimization problem into multiple small-scale solvable optimization problems and (b) enable the fast parallel processing for the first and second stages. For example, a video captured at 60 frames/s (fps) over 24 h has about 5.2×10^6 images, which would result in millions of tracklets and make it unsolvable on a normal workstation to optimally link all the tracklets in a single optimization problem with millions of variables.

The first stage—visual mark association. Three possible associations between two visual marks in the kth image subsequence $[N_{k-1}^{frame}+1,\ N_k^{frame}]$ were considered: translation, disappearing, and reappearing. For brevity, the subscript k is omitted in this section without causing any confusion. Let N_p^{Mark} and N_c^{Mark} be the numbers of visual marks extracted in the previous and current images, respectively. Each visual mark is represented by a vector $\vec{v}=[\vec{l},\vec{\psi}]$, where $\vec{l}$ and $\vec{\psi}$ denote its location and SURF feature descriptor. In the following, the costs of the above three associations are defined and applied to track various visual marks.

1. *Translation*. A visual mark moves from one position in the previous image to another in the current image due to the surface deformation of a structure. Since the deformation of a structure between two time instants is governed by Newton's laws, the search area for visual mark correspondence must be subjected to a physical constraint. Let the maximum linear deformation between two consecutive images be Δ_s, which can be estimated by dividing the load increment between the two images by stiffness of the structure. The cost to match Mark $\overrightarrow{v_i}$ in the previous image to Mark $\overrightarrow{v_j}$ in the current image is defined as

$$c\left(\overrightarrow{v_i}\rightarrow\overrightarrow{v_j}\right)=\frac{\left\|\vec{l}_i-\vec{l}_j\right\|}{\Delta_s}+\frac{\left\|\overrightarrow{\psi_i}-\overrightarrow{\psi_j}\right\|}{M_\psi},\quad \text{when } \vec{l}_i-\vec{l}_j\leqslant\Delta_s \tag{7}$$

where $i\in\left[1,N_p^{Mark}\right]$, $j\in\left[1,N_c^{Mark}\right]$, $\|.\|$ represents L_2-norm of the vector in the bracket, and

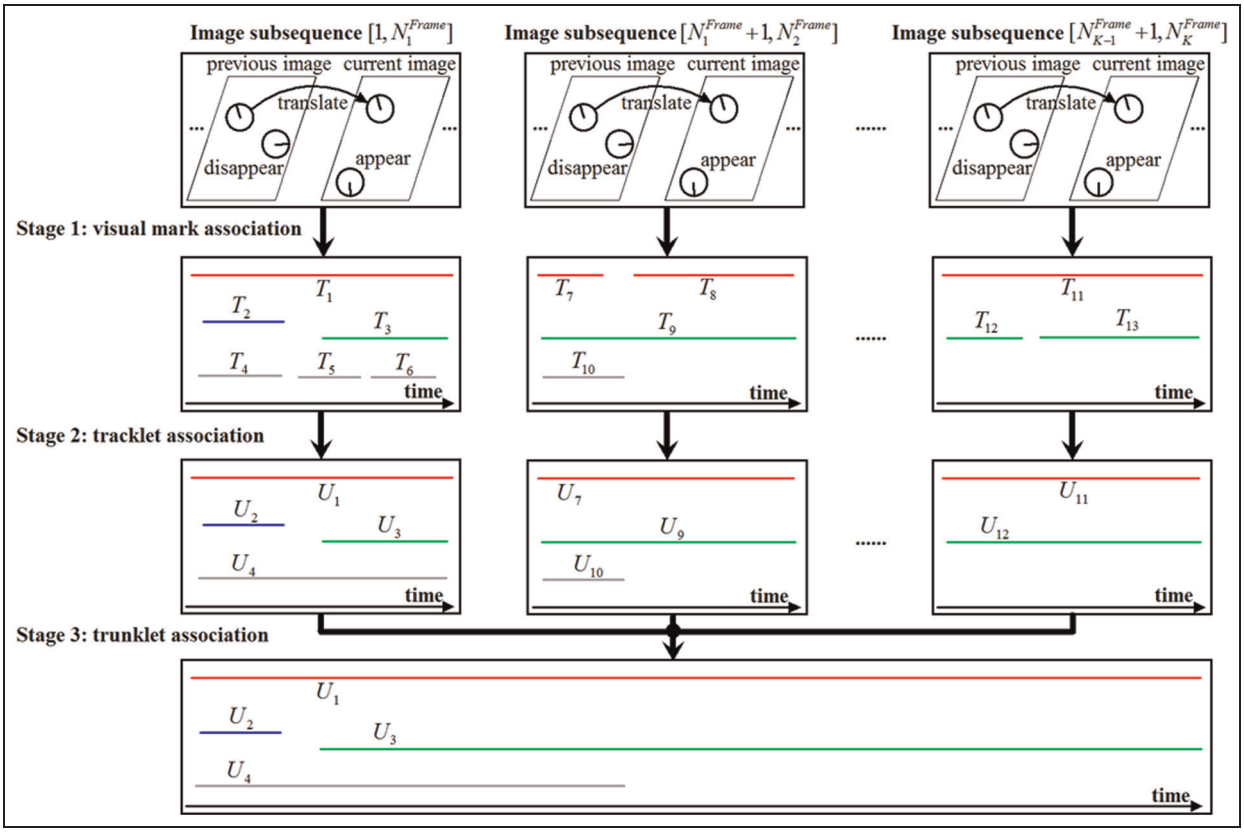

Figure 4. Schematic view of the three-stage data association.

$M_\psi = \max_{ij}\left\|\overrightarrow{\psi_i} - \overrightarrow{\psi_j}\right\|$. The cost function considers both spatial distance and feature similarity between two visual marks. The closer the two marks and the more similar the two features, the smaller the cost of their correspondence. When two marks are separated by more than Δ_s spatial distance, they do not have any translation association.

2. *Disappearing*. A visual mark in the previous image disappears in the current image due to potential blocked vision and illumination change. The cost of a visual mark disappearance from the previous image is defined as

$$c\left(\overrightarrow{v_i} \to \phi\right) = \sum_i \sum_j c\left(\overrightarrow{v_i} \to \overrightarrow{v_j}\right) - \sum_j c\left(\overrightarrow{v_i} \to \overrightarrow{v_j}\right) \tag{8}$$

When the total cost to associate $\overrightarrow{v_i}$ with all possible $\overrightarrow{v_j}$, $\sum_j c(\overrightarrow{v_i} \to \overrightarrow{v_j})$, is low, a translation association from $\overrightarrow{v_i}$ has occurred with high probability. In this case, the probability of a disappearing association with $\overrightarrow{v_i}$ must be low (e.g. 0.1) or the cost to associate $\overrightarrow{v_i}$ as a disappearing case is high as indicated in equation (8).

3. *Reappearing*. A visual mark not present in the previous image appears in the current image. The cost of a visual mark reappearance in the current image is defined as

$$c\left(\phi \to \overrightarrow{v_j}\right) = \sum_i \sum_j c\left(\overrightarrow{v_i} \to \overrightarrow{v_j}\right) - \sum_i c\left(\overrightarrow{v_i} \to \overrightarrow{v_j}\right) \tag{9}$$

Similar to disappearance, when the total cost to associate all possible $\overrightarrow{v_i}$ with $\overrightarrow{v_j}$, $\sum_i c(\overrightarrow{v_i} \to \overrightarrow{v_j})$, is low (e.g. 0.1), a translation association to $\overrightarrow{v_j}$ has occurred with high probability. In this case, the probability of a reappearing association with $\overrightarrow{v_j}$ must be low or the cost to associate $\overrightarrow{v_j}$ as a reappearing case must be high as indicated in equation (9).

4. *Association cost minimization*. The possible numbers of translation, disappearance, and reappearance associations between the two images are represented by $N_{Translation}^{Mark}$, N_p^{Mark}, and N_c^{Mark}, respectively. The number of possible translation associations is determined by the inequality in equation (7) or the number of (i,j) pairs within $\left\|\vec{l}_i - \vec{l}_j\right\| \leqslant \Delta_s$.

Therefore, the dimension of a cost vector $\vec{c}$ for the three associations is $N_{Translation}^{Mark} + N_p^{Mark} + N_c^{Mark}$. The actual associations between two images are determined by solving the following argument-of-minimum (argmin) optimization problem

$$\arg\min_{\vec{a}} \vec{c}^T \vec{a}, \quad \text{s.t.} \quad Q^T \vec{a} = \vec{1} \tag{10}$$

where $\vec{a}$ is a binary vector with $N_{Translation}^{Mark} + N_p^{Mark} + N_c^{Mark}$ elements, and $a_n = 1$ means that the nth association is selected in the optimal solution; Q is a $(N_{Translation}^{Mark} + N_p^{Mark} + N_c^{Mark}) \times (N_p^{Mark} + N_c^{Mark})$ binary matrix, with the nonzero elements of each row indicating which visual marks are potentially involved in that association; $\vec{1}$ represents a vector with all elements one; and the superscript "T" represents the transpose of a vector or a matrix. The constraint $Q^T \vec{a} = \vec{1}$ is introduced to ensure that each visual mark appears in no more than one association in the optimal solution. The solution process of the linear programming optimization problem is executed from the first to the last image of an image subsequence to generate all tracklets in that subsequence. Note that equation (10) is a linear optimization problem with convex functions. Its local minimum is also a global minimum whenever available. Therefore, the optimization solution is always unique.

Figure 5 shows an example optimization process with $N_p^{Mark} = 2$ and $N_c^{Mark} = 2$. The distance between Mark A/B in the previous image and Mark C in the current image was considered within Δ_s so that their possible translation associations were established. However, the distance between Mark A/B and Mark D in the current image exceeded Δ_s so that no translation associations were established. Thus, $N_{Translation}^{Mark} = 2$. After the linear programming optimization, Mark A matched with Mark C, Mark B disappeared from the previous image, and Mark D reappeared in the current image.

The second stage—tracklet association. The tracklets extracted within the kth subsequence can be associated in three ways: linking, disappearing, and reappearing. Again for brevity, the subscript k for the specified subsequence is omitted without causing any confusion. Let $N^{Tracklet}$ represents the total number of tracklets in the subsequence and the ith tracklet be $T_i = \{\overrightarrow{v_i^{s_i}}, \overrightarrow{v_i^{s_i+1}}, \ldots, \overrightarrow{v_i^{e_i}}\}$, where s_i and e_i represent the first and last time instants of T_i, respectively.

1. *Linking*. Two tracklets T_i and T_j that were separated in the first stage due to temporarily blocked vision and/or illumination change are connected when generated by the same visual mark. The cost to link the two tracklets is defined as

$$c(T_i \rightarrow T_j) = \frac{\left\| \vec{l}_i^e - \vec{l}_j^s \right\|}{\Delta_s} + \frac{\left\| e_i - s_j \right\|}{\Delta_t} + \frac{\left\| \overrightarrow{\psi_i^e} - \overrightarrow{\psi_j^s} \right\|}{M_\psi},$$
$$\text{when } \vec{l}_i^e - \vec{l}_j^s \leqslant \Delta_s \text{ and } 0 \leqslant s_j - e_i \leqslant \Delta_t \tag{11}$$

where $i,j \in [1, N^{Tracklet}]$; $\vec{l}_i^e$ is the tail location of Tracklet T_i or Mark $\overrightarrow{v_i^{e_i}}$; $\vec{l}_j^s$ is the head location of Tracklet T_j or Mark $\overrightarrow{v_j^{s_j}}$; $\overrightarrow{\psi_i^e}$ is the last SURF feature descriptor of Tracklet T_i or Mark $\overrightarrow{v_i^{e_i}}$; and $\overrightarrow{\psi_j^s}$ is the first SURF feature descriptor of Tracklet T_j or Mark $\overrightarrow{v_j^{s_j}}$. The linking cost between the two tracklets is valid only when (a) the spatial distance between the tail location of T_i and the head location of T_j does not exceed the maximum translation distance, that is, $\left\| \vec{l}_i^e - \vec{l}_j^s \right\| \leqslant \Delta_s$ and (b) Tracklet T_j starts after Tracklet T_i in time and their time difference does not exceed Δ_t, that is, $0 \leqslant s_j - e_i \leqslant \Delta_t$, where Δ_t represents the maximum duration a visual mark can possibly be lost from images over time. For example, $\Delta_t = 50$ means that the visual mark may continue to disappear from 50 images.

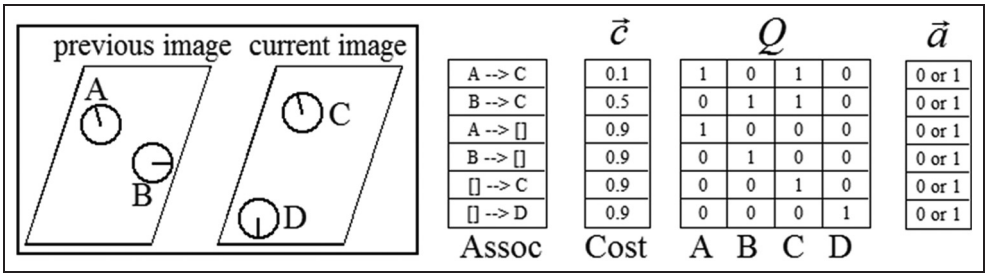

Figure 5. Visual mark association by linear programming optimization.

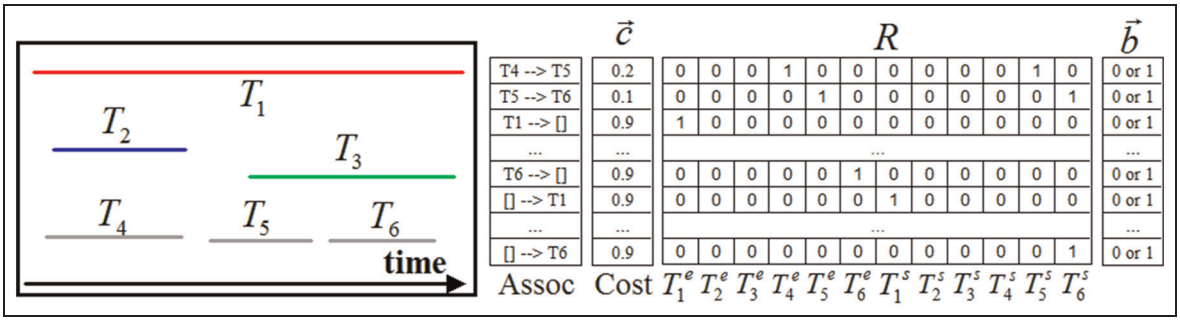

Figure 6. Tracklet association by linear programming optimization.

2. *Disappearing*. A tracklet in the previous image disappears when it cannot be connected to any later tracklet within the subsequence. The cost for a tracklet disappearance is

$$c(T_i \to \phi) = \sum_i \sum_j c(T_i \to T_j) - \sum_j c(T_i \to T_j) \quad (12)$$

Any tracklet within the subsequence can potentially disappear and become the last tracklet of a trunklet. Therefore, there are $N^{Tracklet}$ possible disappearances within the image subsequence.

3. *Reappearing*. A tracklet in the current image reappears when it cannot be connected to any previous tracklet within the subsequence. The cost for a tracklet reappearance is

$$c(\phi \to T_j) = \sum_i \sum_j c(T_i \to T_j) - \sum_i c(T_i \to T_j) \quad (13)$$

Any tracklet within the subsequence can potentially reappear and become the first tracklet of a trunklet. Therefore, there are $N^{Tracklet}$ possible appearances within the image subsequence.

4. *Association cost minimization*. The number of linking associations, denoted by $N_{Linking}^{Tracklet}$, can be determined from the spatial and temporal constraints in equation (11). Therefore, the dimension of a cost vector for the three associations is equal to $N_{Linking}^{Tracklet} + 2N^{Tracklet}$. Similar to equation (10), the following minimization problem can be formulated for the cost vector

$$\arg\min_{\vec{b}} \ \vec{c}^T\vec{b}, \quad \text{s.t. } R^T\vec{b} = \vec{1} \quad (14)$$

where $\vec{b}$ is a binary vector with $N_{Linking}^{Tracklet} + 2N^{Tracklet}$ elements, and $b_n = 1$ means that the nth association is selected in the optimal solution, and R is a $(N_{Linking}^{Tracklet} + 2N^{Tracklet}) \times 2N^{Tracklet}$ binary matrix, with the nonzero elements of each row indicating which tracklets with starting and ending time information are involved in that association.

Figure 6 shows an example optimization process to link 6 tracklets within a subsequence. Due to spatial and temporal distance constraints (i.e. Δ_s and Δ_t), only two linking associations ($T_4 \to T_5$ and $T_5 \to T_6$) are possible in this subsequence. After the linear programming optimization, Tracklets T_4, T_5, T_6 were linked to a long trunklet $\{T_4, T_5, T_6\}$. Tracklets T_1, T_2, and T_3 represented three single-tracklet trunklets $\{T_1\}$, $\{T_2\}$, and $\{T_3\}$.

The third stage—trunklet association. The next step is to connect trunklets between two consecutive subsequences. For brevity, the consecutive subsequences are referred to as the previous and current subsequences without the explicit use of subscripts. Let the numbers of trunklets in the previous and current subsequences be $N_p^{Trunklet}$ and $N_c^{Trunklet}$, respectively. The ith trunklet with M tracklets in the previous subsequence, $i \in \left[1, N_p^{Trunklet}\right]$, can be represented by

$$U_i = \left\{ T_{i_1}, T_{i_2}, \ldots, T_{i_M} \right\} = \left\{ \left\{ \overrightarrow{v_{i_1}^{s_{i_1}}}, \overrightarrow{v_{i_1}^{s_{i_1}+1}}, \ldots, \overrightarrow{v_{i_1}^{e_{i_1}}} \right\}, \ldots, \left\{ \overrightarrow{v_{i_M}^{s_{i_M}}}, \overrightarrow{v_{i_M}^{s_{i_M}+1}}, \ldots, \overrightarrow{v_{i_M}^{e_{i_M}}} \right\} \right\} \quad (15)$$

where T_{i_1} and T_{i_M} are the first and last tracklets in the trunklet, each consisting of a series of visual mark descriptors. The jth trunklet in the current subsequence can be expressed similar to equation (15) and $j \in [1, N_c^{Trunklet}]$. Like visual marks and tracklets, trunklets can be related with three associations: connection, disappearing, and reappearing.

1. *Connection*. Trunklet U_i in the previous subsequence is related to Trunklet U_j in the current subsequence when generated from the same visual mark. The cost to connect the two trunklets is defined as

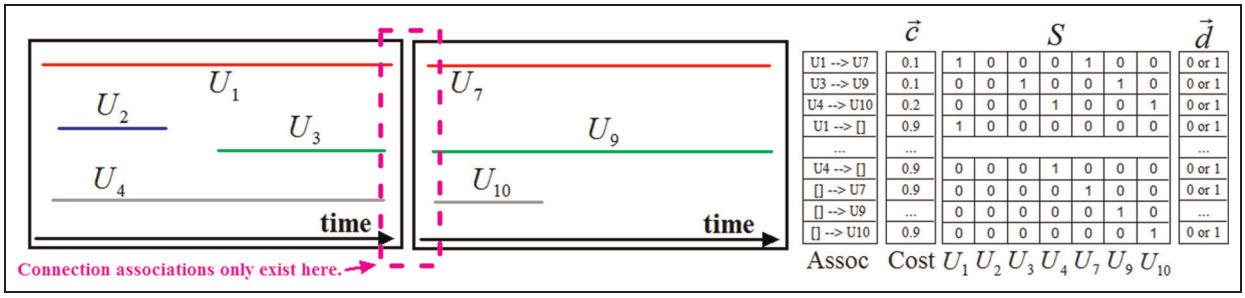

Figure 7. Trunklet association by linear programming optimization.

$$c(U_i \to U_j) = \frac{\left\|\vec{l}^{\,e}_{i_M} - \vec{l}^{\,s}_{j_1}\right\|}{\Delta_s} + \frac{\left\|e_{i_M} - s_{j_1}\right\|}{\Delta_t} + \frac{\left\|\overrightarrow{\psi^{e_{i_M}}_{i_M}} - \overrightarrow{\psi^{s_{j_1}}_{j_1}}\right\|}{M_\psi},$$
$$\text{when } \left\|\vec{l}^{\,e}_{i_M} - \vec{l}^{\,s}_{j_1}\right\| \leqslant \Delta_s \text{ and } 0 \leqslant s_{j_1} - e_{i_M} \leqslant \Delta_t \quad (16)$$

where $\vec{l}^{\,e}_{i_M}$ represents the tail location of Trunklet U_i or Tracklet T_{i_M} or Mark $\overrightarrow{v^{e_{i_M}}_{i_M}}$; $\vec{l}^{\,s}_{j_1}$ is the head location of Trunklet U_j or Tracklet T_{j_1} or Mark $\overrightarrow{v^{s_{j_1}}_{j_1}}$; e_{i_M} is the last time instant of Trunklet U_i; s_{j_1} is the first time instant of Trunklet U_j; $\overrightarrow{\psi^{e_{i_M}}_{i_M}}$ is the last SURF feature descriptor of Trunklet U_i; and $\overrightarrow{\psi^{s_{j_1}}_{j_1}}$ is the first SURF feature descriptor of Trunklet U_j.

2. *Disappearing*. A trunklet in the previous subsequence disappears when it cannot be related to any trunklet in the current subsequence. The cost for the disappearance is defined as

$$c(U_i \to \phi) = \sum_i \sum_j c(U_i \to U_j) - \sum_j c(U_i \to U_j) \quad (17)$$

3. *Appearing*. A trunklet in the current subsequence reappears when it cannot be related to any trunklet in the previous subsequence. The cost for the trunklet reappearance is defined as

$$c(\phi \to U_j) = \sum_i \sum_j c(U_i \to U_j) - \sum_i c(U_i \to U_j) \quad (18)$$

4. *Association cost minimization*. The number of connection associations between two subsequences, denoted by $N^{Trunklet}_{Connection}$, can be determined by the spatial and temporal constraints in equation (16). Therefore, the dimension of a cost vector for the three associations is equal to $N^{Trunklet}_{Connection} + N^{Trunklet}_{P} + N^{Trunklet}_{C}$. The overall cost can be minimized by

$$\arg\min_{\vec{d}} \vec{c}^{\,T}\vec{d}, \quad \text{s.t. } S^T\vec{d} = \vec{1} \quad (19)$$

where $\vec{d}$ is a binary vector with $N^{Trunklet}_{Connection} + N^{Trunklet}_{P} + N^{Trunklet}_{C}$ elements, and $d_n = 1$ means the nth association is selected in the optimal solution, and S is a $(N^{Trunklet}_{Connection} + N^{Trunklet}_{P} + N^{Trunklet}_{C}) \times (N^{Trunklet}_{P} + N^{Trunklet}_{C})$ binary matrix, with the nonzero elements of each row indicating which trunklets are involved in that association.

Figure 7 shows an example minimization process to relate the trunklets between two consecutive subsequences. Due to the temporal distance constraint, the trunklet relations only exist in a transition period from the end of the previous subsequence to the beginning of the current subsequence. After the minimization process, Trunklets U_1 and U_7; U_3 and U_9; and U_4 and U_{10} were related. Trunklet U_2 disappeared.

Application of the three-stage data association algorithm. Figure 8 illustrates the application process of the proposed three-stage data association algorithm. Corresponding to Figure 4, a total of 19,100 frames were divided into 10 image subsequences; each of the first 9 subsequences having 2000 frames and the last subsequence including 1100 frames. First, visual marks were matched between any consecutive images as shown in Figure 8(a). Those visual marks that were tracked continuously for a series of images within a subsequence formed tracklets. Then, the broken tracklets in the subsequence were linked using the tracklet association algorithm to form long trunklets as shown in Figure 8(b). Finally, the trunklets in various subsequences were connected using the trunklet association algorithm to form the complete trajectories of visual marks as indicated in Figure 8(c). In this application, $\Delta_s = 3$ pixels and $\Delta_t = 50$ images. The average computational time for visual mark detection and first stage association was 0.12 s on a common desktop. On an average, each subsequence had approximately 3000 tracklets that were linked into trunklets in 98 s. The

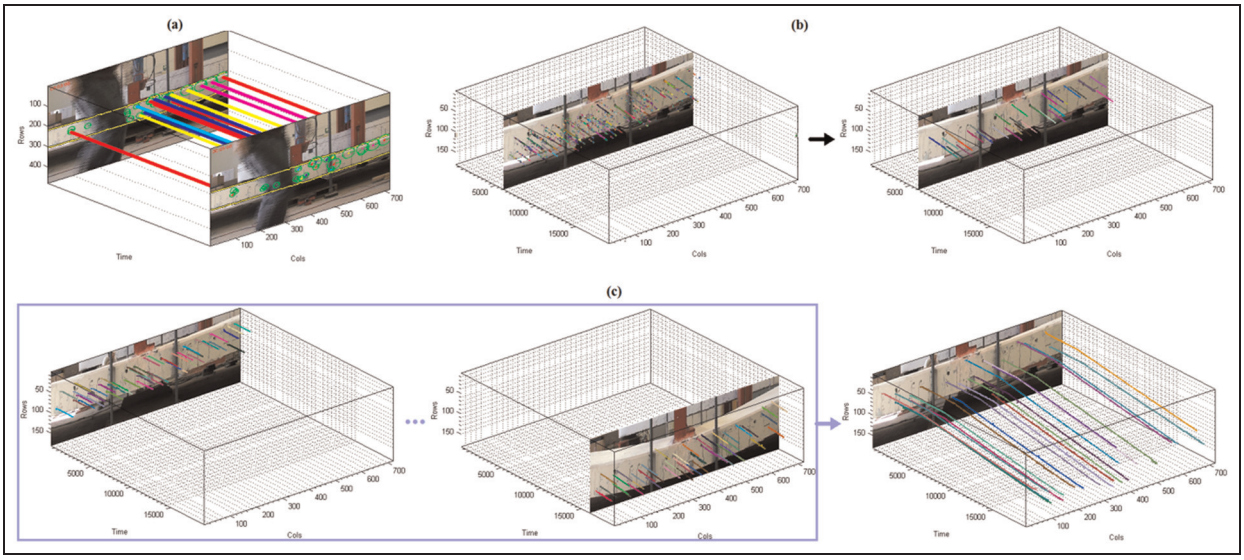

Figure 8. Three-stage data association application: (a) visual mark association, (b) tracklet association, and (c) trunklet association.

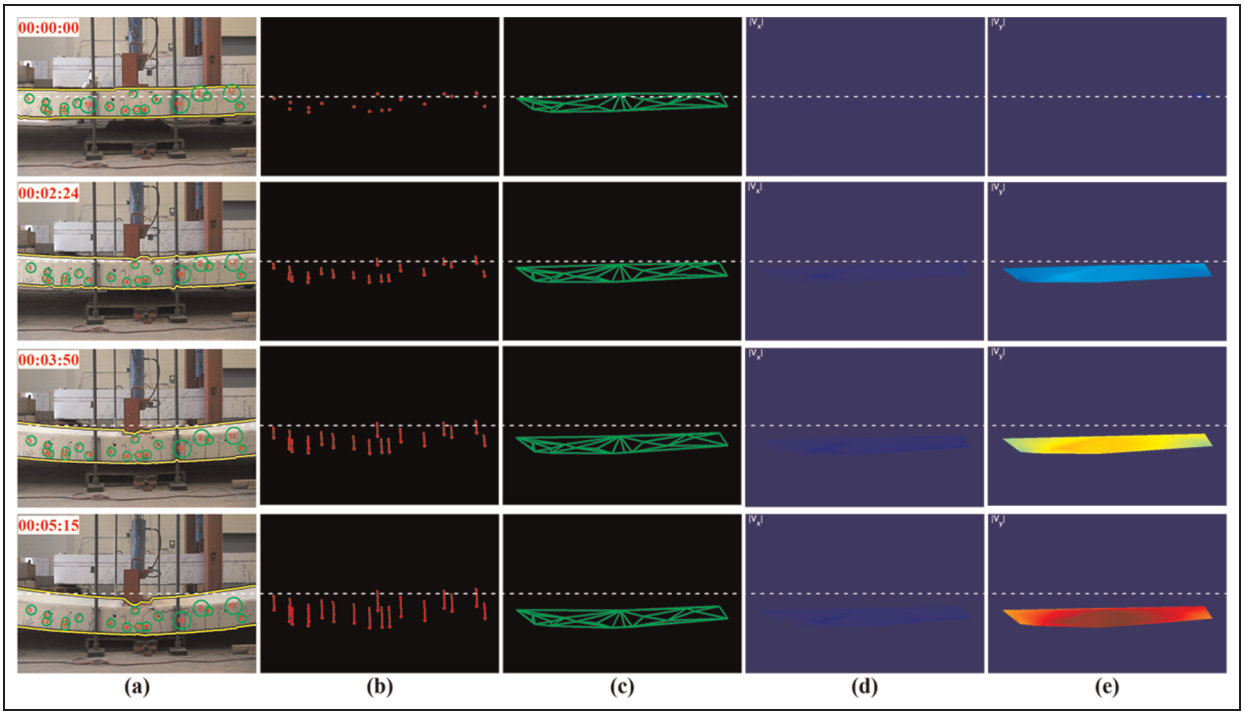

Figure 9. Displacement field construction at various time instants: (a) tracked visual marks, (b) visual mark displacements, (c) triangle elements, (d) *x*-displacement field, and (e) *y*-displacement field.

trunklets in the 10 subsequences were connected to the complete trajectories of surface displacements in 7.4 s.

Construction of displacement field and its derived fields

After the complete trajectory of each visual mark has been found, the displacement of its corresponding point on a structure can be determined by subtracting the initial position of the visual mark (under no loading at time $t = 0$) from the current trajectory value (under loading at time t). For example, Figure 9(a) and (b) shows the tracked visual marks and their corresponding displacements, respectively. To construct the full displacement field of the structure from the scattered trajectories at discrete visual marks, the Delaunay

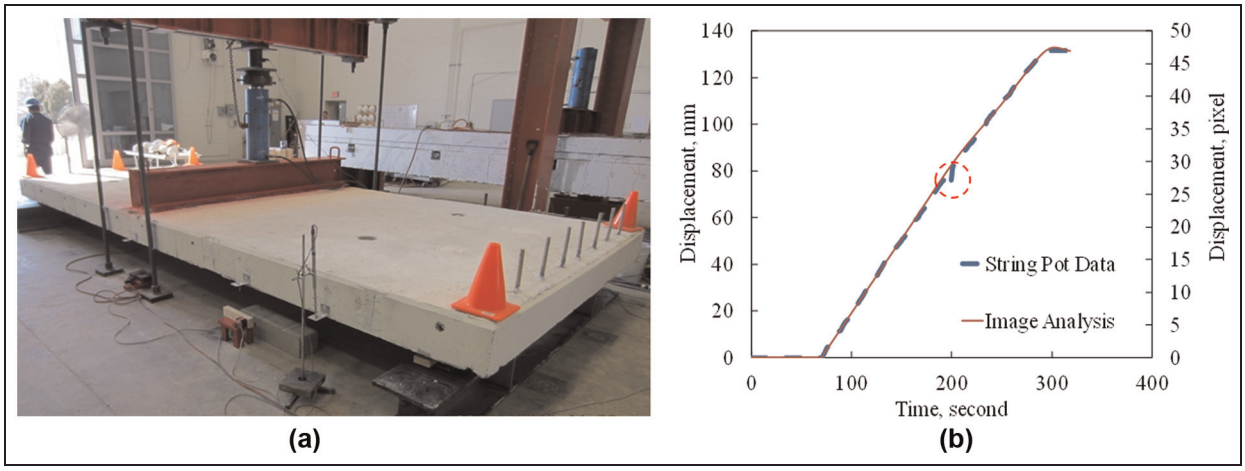

Figure 10. Test setup and results of the concrete slab: (a) test setup and (b) mid-span displacement over time.

triangulation algorithm[10] is applied as shown in Figure 9(c). First, three nearby visual marks detected were selected. Their represented displacements were then used to interpolate the displacement at any point inside the triangle formed by the three marks. In essence, the displacement inside the triangle is considered as a linear function of the displacements at three vertices of the triangle in finite element analysis.[11]Figure 9(d) and (e) shows the x- and y-displacement fields, respectively, based on the interpolation with the triangle element. The obtained displacement field at any time instant can be used to determine the surface strain field of the structure through the procedure commonly used in finite element analysis.[11] The displacement time history obtained can also be used to evaluate the velocity or acceleration time histories by taking first and second derivatives.

Experimental validation of the proposed visual mark tracking method

Test setup of the concrete slab

The reinforced concrete (RC) slab as used for illustrations in Figures 1 to 4, 8, and 9 was tested under three-point loading as shown in Figure 10(a). A point load was applied at mid-span of the slab and uniformly distributed through a transverse spread beam. The applied load was measured through a load cell installed between the hydraulic jack and the reaction frame in Figure 10(a). A string pot was attached to the bottom surface of the slab at mid-span to measure the mid-span deflection of the slab at 50 samples/s. A SONY video camera (Model HDR-CX190) was set 2 m in front of the simply supported concrete slab to capture 720 pixels $\times$ 480 pixels video images at 60 fps.

Results and discussion

The mid-span displacement was determined from video images with the proposed visual image tracking method. The results are presented as a function of time and compared in Figure 10(b) with the test data obtained from the laboratory tests after the video camera was set up. Overall, the displacement from image analysis is in good agreement with the experimental results, demonstrating the accuracy of the proposed visual image tracking method.

The displacement sampling rate from the string pot (50 samples/s) is not in multiplier of the frame rate from the camera (60 fps). Therefore, the displacement from the image analysis was compared with the nearest experimental data in terms of time. At each time instant, the displacement error between the image analysis and experimental data and the relative error by dividing the displacement error by the image analysis displacement were calculated. Over the test duration of 318 s (19,100 frames), the average relative error was approximately 1.24%. In addition, the root mean square (RMS) error between the image analysis and experiment was determined to be 0.124 mm or 0.042 pixels, which is only 0.095% the maximum displacement obtained from image analysis.

Figure 10(b) indicates that a "kink" in red circle was observed from the experimental displacement time history. This phenomenon was likely attributed to the occurrence and propagation of significant cracks near the string pot since the mid-span velocity notably decreased from the pre- to post-kink. Due to the non-contact nature, the proposed visual mark tracking method can accurately find the change in velocity without being disturbed by the cracking process in concrete. This observation clearly showed the advantage of the

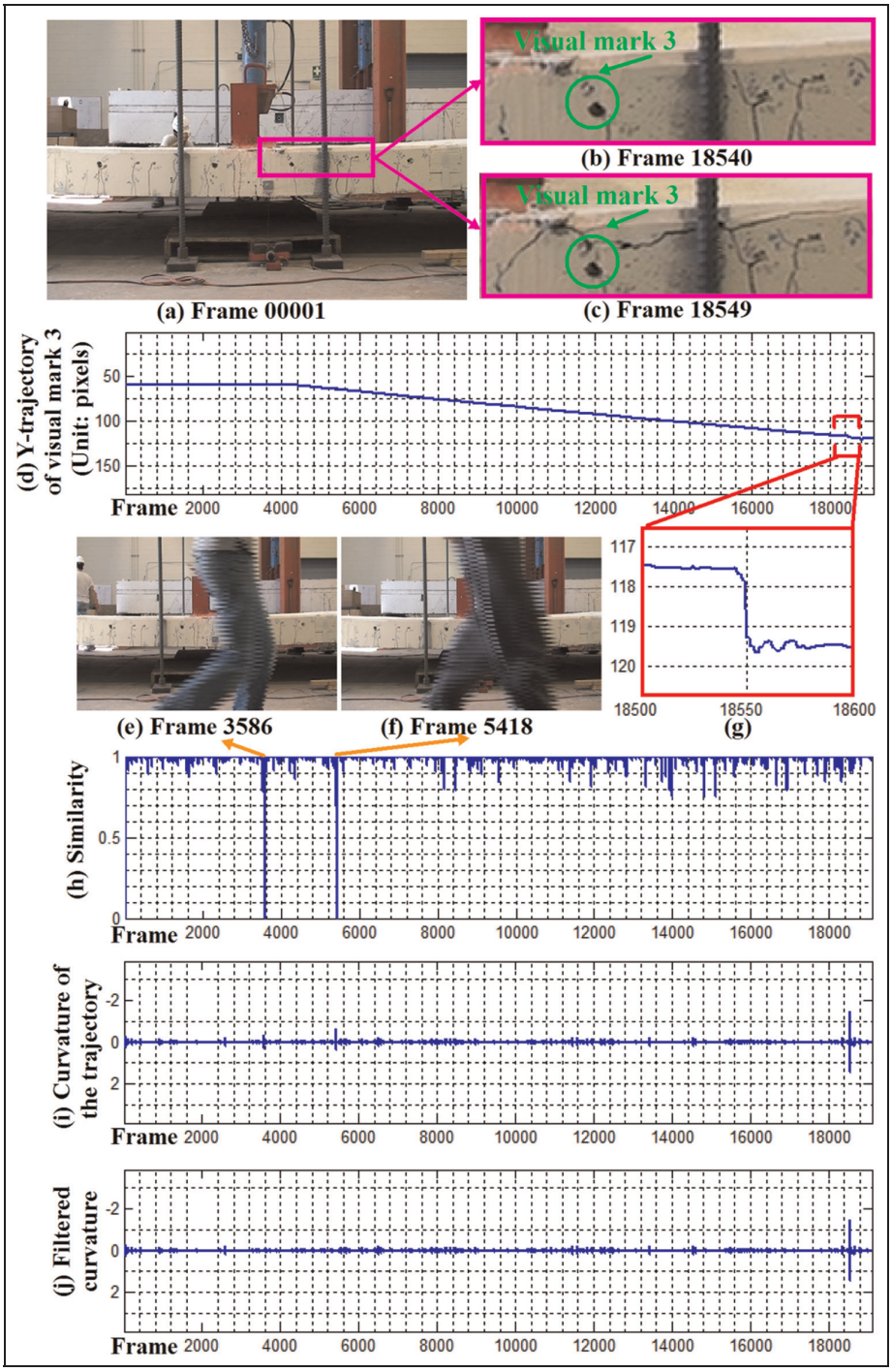

Figure 11. Computing the curvature of visual mark's trajectories: (a) frame 00001, (b) frame 18450, (c) frame 18549, (d) *y*-trajectory of visual Mark 3, (e) frame 3586, (f) frame 5418, (g) change on the visual mark's trajectory, (h) similarity, (i) curvature of the trajectory, and (j) filtered curvature.

proposed noncontact methodology in displacement measurement.

Application on crack detection

The visual mark tracking system can be applied to detect where and when cracks appear on a structure and then determine whether the detected cracks have structural implications by comparing them with the threshold specified in design codes. For the RC slab as shown in Figure 10(a), new cracks occurred in the highlighted area at 90 tons loading around frame 18,549 after the camera had been set up. Note that other cracks

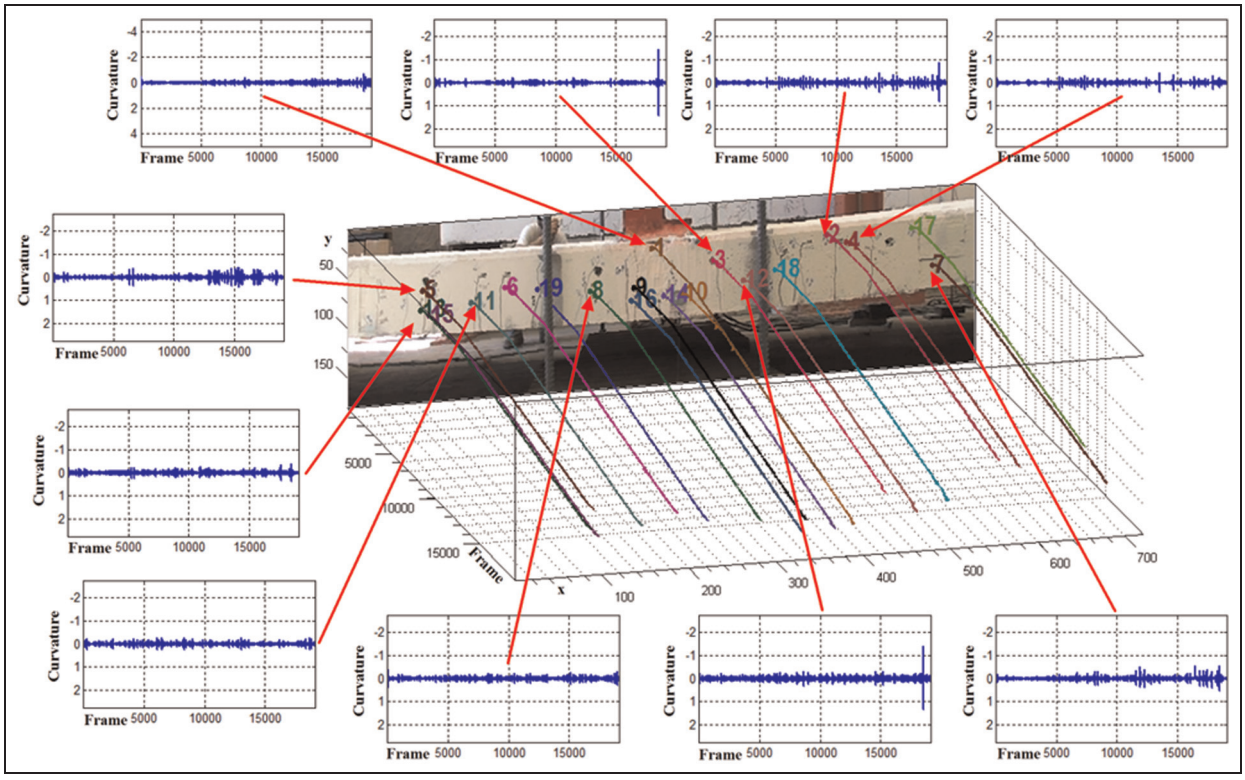

Figure 12. Curvature trajectories of the tracked visual marks.

in Figure 10(a) were already in existence before the frames had been taken in this study due to previous loading. Cropped from the full image coordinate as shown in Figure 11(a), Figure 11(b) and (c) shows zoomed-in details around the new crack region before and after their occurrence. Figure 11(d) shows the y-component of a visual mark's trajectory or vertical displacement near the region.

There was an abrupt change on the visual mark's trajectory when cracks occurred, as indicated in Figure 11(g). The curvature of the trajectory, computed by the second-order finite difference, was used to detect the abrupt change as presented in Figure 11(i). Since occlusion may happen during the visual mark tracking as exemplified in Figure 11(e) and (f), similarity of visual marks between consecutive frames was computed as shown in Figure 11(h). When the visual mark was occluded, the similarity was low. The filtered curvature by weighting the original curvature with the similarity is shown in Figure 11(j), where the location of the highest magnitude of curvature indicates the occurrence of cracks.

Figure 12 shows a summary of the curvatures of various tracked visual marks. Marks 2, 3, and 12 have abrupt changes around the same time instant, indicating the occurrence of cracks near that region as verified by visual observations.

The sparse set of time-series curvature trajectories such as those in Figure 12 were interpolated by Delaunay triangulation to estimate the full-field curvature trajectory over time, as shown in Figure 13. The time-lapse heat maps in Figure 13 clearly indicate where and when the cracks appeared during the loading test.

Conclusion

Based on image analyses and experimental validations, the following conclusions can be drawn:

- The three-stage data association algorithm for visual mark translation, linking, and connection was successfully applied to visual mark tracking of concrete slab images.
- The proposed algorithms in five modules are computationally efficient, making them viable tools for real-time structural health monitoring. They are applied to successfully detect cracks in a full-scale RC slab from the curvature trajectories obtained from image analysis.
- By tracking spatially distributed visual marks from time-lapse videos, the displacement time histories at mark-associated structure surface are successfully evaluated.

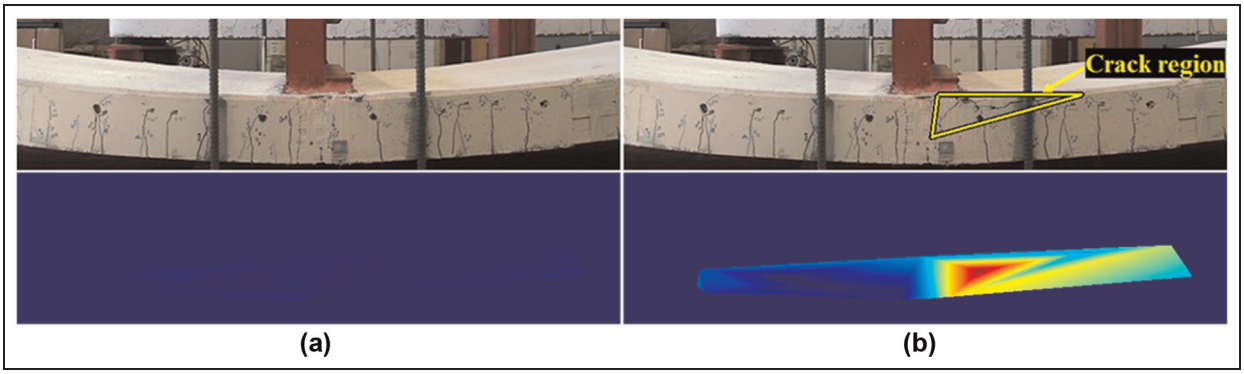

Figure 13. Detection of crack region and occurrence from time-lapse images: (a) frame 18540 and (b) frame 18549.

- The relative error of displacement measurements for the tested concrete slab is approximately 1.24%, demonstrating the accuracy of the proposed sensing method.
- Unlike contact measurements with LVDTs, the proposed noncontact measurement is not affected by concrete cracking.

This article presents a new methodology for full-field displacement and curvature measurements with some preliminary results. For practical applications, a camera can be set near a bridge with a line of sight on the face of the bridge structure (there is no need to set the camera right in front of the bridge surface to be monitored). In these cases, the captured video must be adjusted to take into account the effect of the video angle between the line of sight and the bridge face before the procedure presented in this article can be used directly. The proposed method must be further investigated and validated to understand the sensitivity of various parameters and develop their selection criteria. For example, a bridge carries moving traffic vehicles. Therefore, capturing images with a camera under the moving load requires a certain minimum speed or the frame rate per second. Due to limited space, parametric studies will be summarized in another article in the near future.

Declaration of conflicting interests

The results and opinions expressed in this article are those of the authors only and they do not necessarily represent those of the sponsors.

Funding

Financial support for this study was provided in part by the US National Science Foundation (NSF) under Award No. CMMI-1030399 and by the Mid-America Transportation Center (MATC) under Award No. 25-1121-0003-197. Special thanks are due to Dr. Zhibin Lin, Mr. Mostafa Fakharifar, and Mr. Jason cox for preparing the slab specimen tested in this study.

References

1. Jurjo DLBR. Development of a vision system for the dynamic analysis of structures. Internal Report, Department of Civil Engineering, COPPE/UFRJ, Rio de Janeiro, Brazil, 2004.
2. Jurjo DLBR. Stability analysis of a column under self-weight using digital image processing techniques. Internal Report, Department of Civil Engineering, COPPE/UFRJ, Rio de Janeiro, Brazil, 2004.
3. Ji YF and Chang CC. Nontarget image-based technique for small cable vibration measurement. *J Bridge Eng* 2008; 13(1): 34–42.
4. Lee JJ and Shinozuka M. Real-time displacement of a flexural bridge using digital image processing technique. *Exp Mech* 2006; 46: 255–260.
5. Pan B, Qian K, Xie H, et al. Two-dimensional digital image correlation for in-plane displacement and strain measurement: a review. *Meas Sci Technol* 2009; 20(6): 062001 (17 pp.).
6. Chan THT, Ashebo DB, Tam HY, et al. Vertical displacement measurements for bridges using optical fiber sensors and CCD cameras—a preliminary study. *Struct Health Monit* 2009; 8(3): 243–247.
7. Bay H, Ess A, Tuytelaars T, et al. Speeded-Up Robust Features (SURF). *Comput Vis Image Und* 2008; 110: 346–359.
8. Mikolajczyk K and Schmid C. A performance evaluation of local descriptors. *IEEE T Pattern Anal* 2005; 27(10): 1615–1630.
9. Mikolajczyk K, Tuytelaars T, Schmid C, et al. A comparison of affine region detectors. *Int J Comput Vision* 2005; 65(1): 43–72.
10. Watson DF. Contouring: a guide to the analysis and display of spatial data. Oxford, UK: Elsevier Science Ltd., 1992, p. 340.
11. Reddy JN. An introduction to finite element method (McGraw-Hill Series in Mechanical Engineering). New York, USA: McGraw-Hill 2004, p. 757.