Journal of Electronic Imaging

SPIEDigitalLibrary.org/jei

Efficient key frames selection for panorama generation from video

Mohammad Javad Fadaeieslam
Mohsen Soryani
Mahmood Fathy

Efficient key frames selection for panorama generation from video

Mohammad Javad Fadaeieslam
Mohsen Soryani
Mahmood Fathy
Iran University of Science and Technology,
School of Computer Engineering,
Narmak, Tehran, 16846-13114 Iran
E-mail: fadaei@iust.ac.ir

Abstract. *A video sequence consists of several hundred frames, and as a result, creating a panoramic image from these frames is a very time-consuming process. Consecutive frames have large overlap areas that do not provide much information. Therefore, some key frames must be extracted for better performance. There are a number of methods for key-frame selection that match all frames in a video sequence. We present a novel and efficient method to select key frames from video for creating a large panoramic mosaic without matching all frames. Consecutive frames are transformed and projected onto the common mosaic surface and the position of each corner of the next frame is predicted with a distinct Kalman filter on this surface. The overlap area between each predicted frame and its previous key frame is used as the criterion to select the next key frame. This method uses video information to reduce features and align frames with repeated structures more accurately. We show that this approach is an efficient preprocessing step and substantially reduces the time required to construct panorama from a video sequence.* © *2011 SPIE and IS&T.* [DOI: 10.1117/1.3591366]

1 Introduction

Creating a panoramic view from still images is an interesting field of research in computer vision and has found applications in several areas. For this purpose, a number of robust algorithms have been proposed thus far,[1,2] and several commercial software systems have been developed.[1]

However, the image-based mosaic algorithms do not work efficiently when directly applied to video frames. A video may include thousands of frames. Consecutive frames typically have large overlap. Thus, matching them has a high computational cost without acquiring noticeable information. Although a number of papers have proposed methods for creating mosaics from video,[3–6] they do not offer a solution for frame selection without the need to match all the frames.

To decrease computational cost, Steedly et al.[5] extract key frames based on the amount of overlap and stitch only these key frames for constructing the mosaic image. Each frame is matched to the previous frame and a previous key frame. If the amount of overlap is less than a threshold, then it is marked as a key frame. The adjacent key frames must have an appropriate balanced overlap area, because a large overlap increases the computational cost without giving much data and a small overlap results in less accurate image registration. Mei et al.[7] use motion vector field to select best frames for mosaicing by constructing a global motion path. In their approach, motion vectors from the MPEG video format is extracted and used directly. In Refs. 3, 8, and 9 each pair of consecutive frames are also aligned, which is a very time-consuming method.

This paper presents a novel and fast method to select key frames from video as an efficient preprocessing step to create panorama. The main contribution is the camera motion prediction process without matching all frames. All the research work in this area concentrates on a special kind of camera motion. However, in this work, camera movement can include any kind of rotation or translation. Furthermore, the camera does not need to be calibrated, and the focal length could change during capture of a video.

In this work, consecutive frames are transformed and projected onto the common mosaic surface and the position of each corner of the next frame is predicted with a distinct Kalman filter in this surface (one Kalman filter for each corner). A frame where its corners are predicted is labeled as a key frame if its overlap area with the last key frame is lower than a threshold. Then, this new key frame is matched and aligned with the previous key frame. Otherwise, corners of the next frame are predicted and the overlap area between this new frame and the previous key frame is computed. The corners of this aligned frame (new key frame) are used to update the Kalman filters.

The remainder of this paper is structured as follows: Sec. 2 reviews the related works. Section 3 describes the camera motion model and the image-matching technique. The proposed method is described in Sec. 4. The Kalman filter that is used in this paper is presented in Sec. 5. Threshold-updating and feature-reduction methods are introduced in Sec. 6 and 7. Section 8 explains the last step of the method to enhance key fame selection. Experimental results are shown in Sec. 9, and the conclusion is given in Sec. 10.

Paper 10082RR received May 12, 2010; revised manuscript received Apr. 12, 2011; accepted for publication Apr. 26, 2011; published online Jun. 8, 2011.

1017-9909/2011/20(2)/023015/10/$25.00 © 2011 SPIE and IS&T

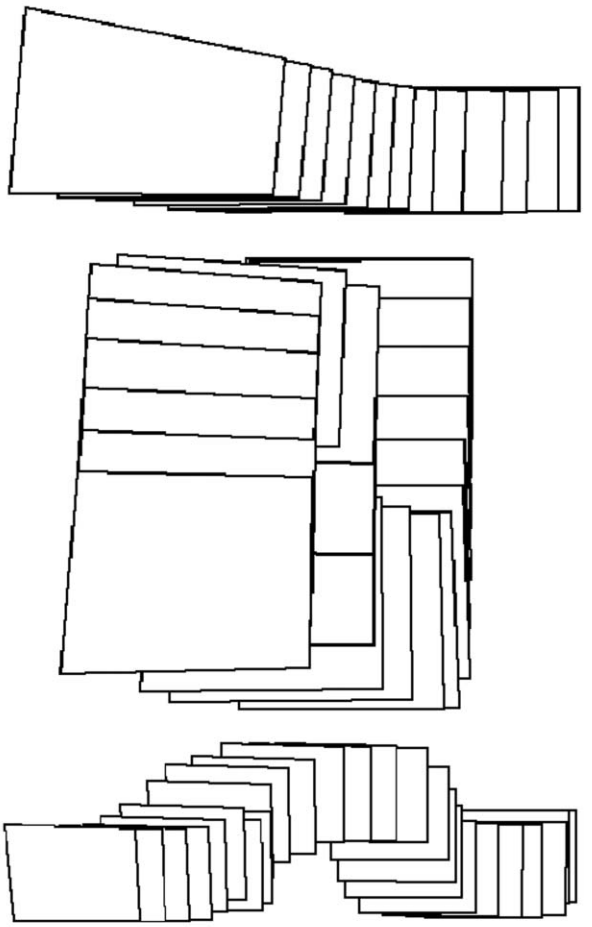

Fig. 1 Some frames of three different videos projected onto their reference frame (first frame plane). Corners compose trajectories on the common mosaic surface.

2 Related Works

Mosaicing methods for creating a panorama could be divided into two categories: In the first category, narrow strips are taken from consecutive frames, warped (if required), and placed onto the mosaic image.[10–13] Manifold projection[10] is a fast technique that is based on the alignment of the strips contributing to the mosaic. It is contrary to the alignment of the entire overlap area between frames. The final product of this technique is less accurate than those of static image-based batch-processing approaches.[5] In Ref. 11 more general types of camera motion, such as forward motion and zooming, is allowed by using strips whose shapes are determined adaptively during the mosaicing process. Wexler et al.[12] proposed a method that converts the geometrical alignment problem to an optimization mosaicing problem when the camera pans or translates mostly along one direction. In this approach, a graph is constructed in which its nodes represent strips (a one-pixel-wide column of pixels) and the edges are possible transitions from each strip to the strips of the next frame. Each path from a strip of the first frame to a strip of the last frame corresponds to a panoramic view. The weight of each edge encodes the transition cost. The Dijkstra algorithm is employed to solve this global optimization process efficiently. This method deals seamlessly with both static and dynamic

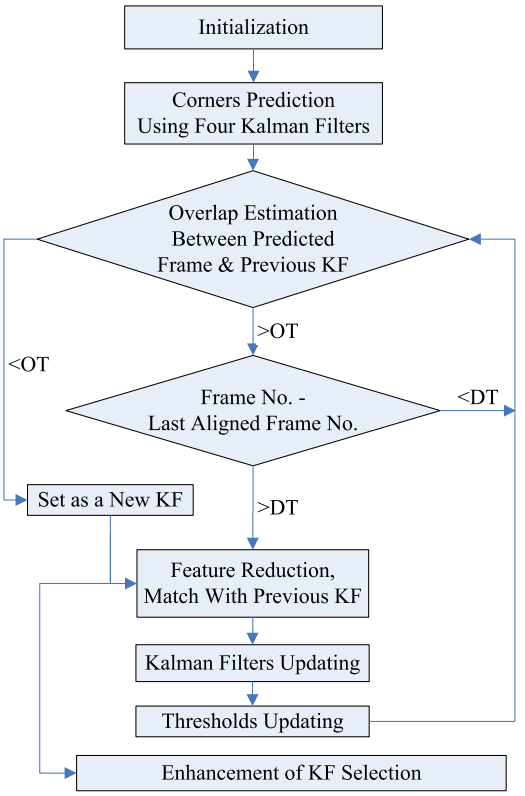

Fig. 2 Flowchart of the proposed method. KF stands for key frame.

scenes with or without 3-D parallax. The methods of the first category are suitable for video mosaicing, but they restrict the camera motion. For example, the camera cannot follow a zigzag path (go back onto a scene partly seen before).

Most existing mosaicing systems fall into the second category, which align and combine full images or video frames.[1,5,14–17] Two different methods are used in this category to align complete images: the direct method[16] and feature based method.[1,14,15,17] The direct method takes the advantage of using all the available image data (pixel-to-pixel matching) and hence can provide very accurate

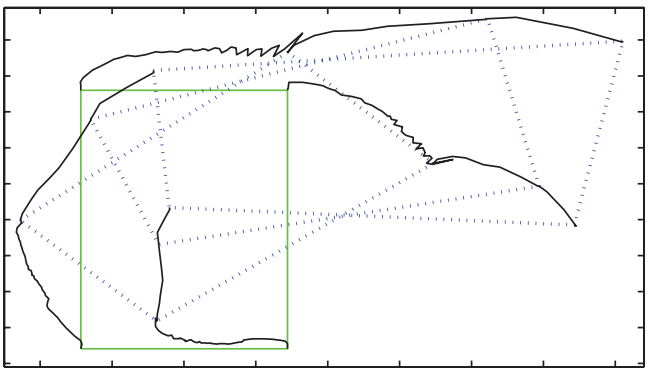

Fig. 3 trajectories of four corners on a common mosaic surface. The velocity of each corner is different. The first frame is shown by a solid rectangle. The main motion of the camera is a mixture of panning and rolling. Its panoramic view is shown in Fig. 10 (video from camera with roll motion).

registration, but their convergence depends heavily on the initialization process. In the feature-based approach, correspondence features between two images are used to estimate the geometric transformation between them. For this purpose, different kinds of features have been proposed. Among them, scale invariant feature transform (SIFT) features[18] that are invariant to rotation, zooming (scaling), and illumination changes have gained popularity more recently.[1,5,17,19,20] Feature-based approaches are more robust when there are moving objects in the scene, and they are potentially faster than the direct methods. Line features in Ref. 17 have been used to estimate lens-distortion parameters.

Direct or feature-based methods are used to register pairs of images. To minimize the misregistration error between all pairs of images, a global adjustment is necessary. Bundle adjustment is a photometric technique for combining multiple images of the same scene into an accurate 3-D configuration, as used in Refs. 1, 5, and 17. It is an iterative algorithm that computes optimal values for the 3-D coordinates of the scene and camera position by minimizing the overall feature projection errors using a least-squares algorithm. Bundle adjustment is an offline method to achieve global optimization. To construct real-time mosaicing, Civera et al.[21] apply the extended Kalman filter (EKF) as a sequential approximation to bundle adjustment in a special situation. They use all the frames in a video captured from a calibrated camera. However, their results could not compete with the ones of an offline method such as Ref. 1. It is also notable that the method of Brown and Lowe[1] does not need camera calibration. Morimoto and Chellappa[22] present one of the prior works in stabilization and mosaicing using EKF. Camera-motion parameters consist of the state vector of EKF and rotational camera moves with a constant focal length. There is no global optimization in their method, which causes the accumulation of misregistration errors. Kim and Hang[23] demonstrate a real-time mosaicing process using a sequential graph. In their approach, images must be mapped to a flat mosaic surface during image registration, which is not suitable for large fields of view.

Aligned images must be mapped onto a suitable compositing surface. The compositing surface, which depends on the camera motion and the application, can be flat, cylindrical, spherical, or any type of surface used for environment mapping.

After alignment, the value of each pixel in the overlap area must be determined. Weighted averaging can be used for this purpose. However, blurring can occur. For highest visual quality, some robust techniques first determine seams between images in the overlap area and then blend images. The Voroni algorithm is one way to select seams, but it ignores local image structures.[24] It is better to place the seams in the regions where the transition from one image to another is not visible. The graph-cuts method[25] has been used in Refs. 20, 26, and 27 to select optimal seams. Blending is applied to remove image edges that are still visible due to lens distortion, moving objects, misregistration errors, vignetting, and exposure differences. Multiband blending[28] is a traditional and robust algorithm that is used in many papers.[1,4,11] Blending in the gradient domain is another useful approach.[20,29] Position of moving objects in the final image must be determined before blending; otherwise, they cause visible artifacts (ghosts) in the final image.

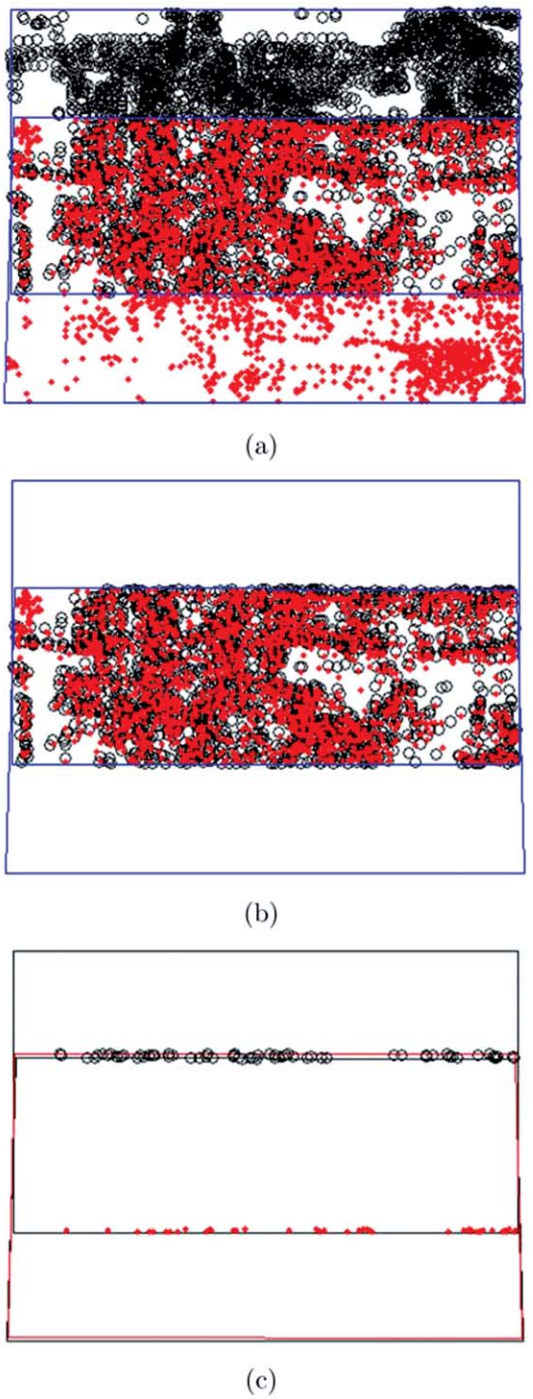

Fig. 4 Feature reduction: (a) two overlapped frames with features that are mapped to the common mosaic surface. The upper frame has 4658 features (circle elements), and the lower frame has 3477 features (point elements). (b) Removing features outside of the overlap area; the upper frame has 2659 features and the lower frame has 2268 features in the overlap area. (c) Features that are in the real overlap area but removed due to prediction errors; 73 and 53 features are incorrectly removed from the upper and the lower frames, respectively.

3 Camera Motion Model and Image Matching

Video sequences used in this paper have been taken by a handheld camera undergoing zooming, rotation, or translation. Video frames are aligned together with a 3×3 homography

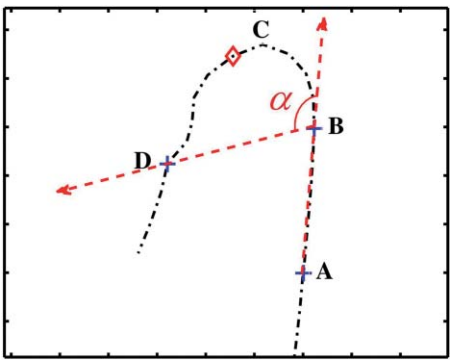

Fig. 5 Part of the camera-movement trajectory. the sign of the camera motion velocity is changed in position C. The ⋄ shows the new key frame place.

matrix. To estimate the homography matrix between two frames, SIFT features[18] are extracted from each frame. After finding corresponding features, the random sample consensus (RANSAC) algorithm is used to estimate the homography matrix.[30] In each iteration of this robust algorithm, four feature correspondences (three of them not collinear) are selected randomly to estimate the homography matrix **H** using the direct linear transformation (DLT) method[31] and the number of inliers (feature correspondences that are geometrically consistent with the estimated **H**) are computed. After N iterations, the sample set with the maximum number of inliers is selected as the final solution. A sufficient number of iterations must be performed to ensure that RANSAC has a good chance of finding a true set of inliers.[24] Because the perspective model is used for matching two images, there will be no limitations on camera movement. Therefore, both rotational and translational movements are permitted, but not both at the same time. The focal length can be variable in both cases. It is assumed that parallax does not occur in camera translation.

4 Algorithm Overview

As mentioned eariler, the prediction of the camera motion is the main purpose of this paper in order to select key frames from a video for panoramic view construction. One approach is to extract motion parameters and try to predict them. Because there is no limitation on the movement of an uncalibrated camera, the decomposition of the homography matrix is known to be very sensitive to image noise.[13] In order to solve this problem, the proposed method tracks the corners of the frames instead of estimating camera-motion parameters. All frames are transformed and projected onto the common mosaic surface. The first frame is considered as the reference frame, and all frames are warped into this reference coordinate system. In this surface, the corresponding corners of frames form four trajectories.

Table 1 Specifications of the four experimental videos.

Video No.	1	2	3	4
Total number of frames of the video shot	1336	880	1250	130
Number of aligned frames	115	49	72	19
Number of key frames	21	4	12	8
Number of overlap key frames (consecutive and non-consecutive)	53	3	21	10

Figure 1 shows some frames of three videos projected onto their reference plane. In Fig. 1, each frame is projected onto the previous frame using $\mathbf{H}_{i-1,i}$, which is computed by direct matching. $\mathbf{H}_{i,j}$ projects frame j to frame i. When i and j are nonconsecutive, homography matrices must be multiplied with each other Eq. (1). $\mathbf{H}_{1,j}$ projects frame j to the reference coordinate system,

$$\mathbf{H}_{i,j} = \mathbf{H}_{i,i+1}\mathbf{H}_{i+1,i+2}\ldots\mathbf{H}_{j-1,j} = \Pi_{m=i}^{j-1}\mathbf{H}_{m,m+1}. \quad (1)$$

Kalman filtering is used in this approach to avoid matching every frame to its previous frame. The flowchart in the Fig. 2 illustrates the process of the proposed method. Because of the different velocity of each corner on the common mosaic surface (Fig. 3), four distinct Kalman filters are used to predict the position of the next frame. Then, the overlap between the last key frame and the predicted frame is estimated. The predicted frame is marked as a new key frame if its overlap with the previous key frame is lower than a threshold. In this situation, this frame is aligned to the

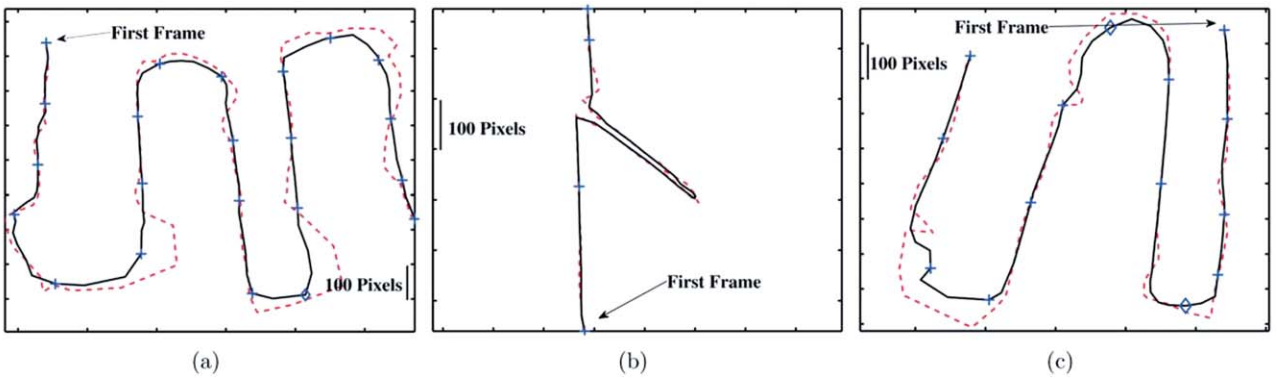

Fig. 6 Trajectories of the upper left corner of frames in videos: (a) 1, (b) 2, and (c) 3. The dotted lines are the predicted trajectories and the solid lines show the exact trajectory of corners which were obtained through alignment. Plus marks on solid lines indicate the upper left corners of the key frames. The ⋄ are key frames that are selected in the enhancement step.

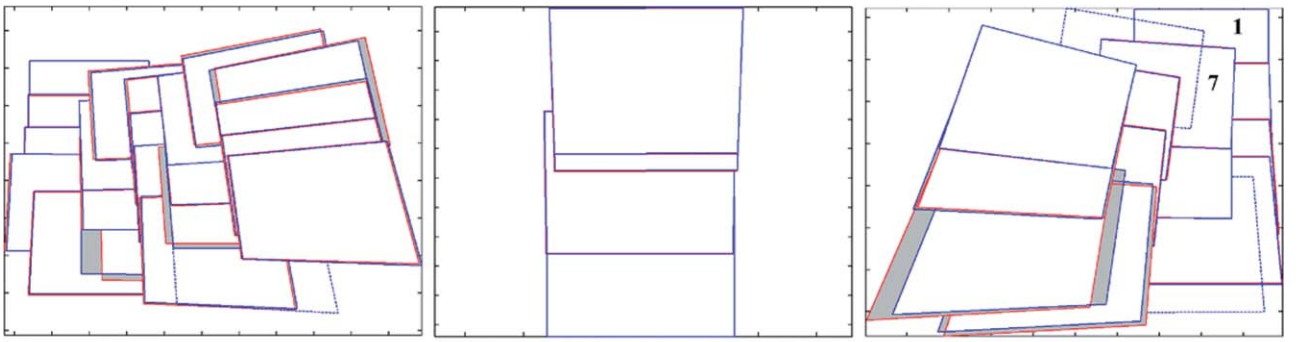

Fig. 7 Key frames of videos on the common mosaic surface: (a) 1, (b) 2, and (c) 3. Red lines show the edges of predicted frames and Blue quadrilaterals show real positions of them. The difference between each predicted key frame and corresponding aligned one is shown in gray. The dotted quadrilaterals are key frames that are selected in the enhancement step.

previous key frame. The corners of this new aligned key frame are used to update Kalman filters. When the overlap area is more than the overlap threshold (OT), the Kalman filters predict corners of the next frame. In this state, Kalman filters are not updated except for situations where the difference between the number of this frame and the number of previous aligned frame is larger than a specified threshold, referred to as distance threshold DT. The first five frames are aligned, and the Kalman filters are updated to acquire better prediction in the remainder of the process.

In each alignment step, the amount of OT and DT are updated, which are explained in Sec. 6. Feature reduction and the enhancement of key frame selection are the other steps of the algorithm and will be explained in Sec. 7 and 8.

5 Kalman Filter

The process of each corner movement over time, which forms the trajectory on the reference frame, can be considered as a dynamic system. If the camera is moved smoothly, then the corners trajectory can be approximated by a time-invariant linear system and, thus, a state-space approach can be employed to model it. Its dynamic equation can be expressed as

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{q}_{k-1}, \tag{2}$$

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{r}_k, \tag{3}$$

where $\mathbf{x}_k$ is the state vector of the system on the time step k and $\mathbf{y}_k$ stands for the observed corner position on the time step k. $\mathbf{A}$ and $\mathbf{H}$ are the state transition and the measurement matrices, respectively. Assume that $\mathbf{q}_{k-1} \sim N(0, \mathbf{Q}_{k-1})$ is the white Gaussian process noise and $\mathbf{r}_k \sim N(0, \mathbf{R}_k)$ is the white Gaussian measurement noise. The measurement noise is uncorrelated with the process noise.[32,33]

The Kalman filter provides a recursive solution for the least-squares estimation of a linear discrete-time dynamic system, which has equations that are similar to Eqs. (2) and (3).

We track the corners of frames in the reference coordinate plane (two-dimensional space). x and y, the position of each corner, are measured in Cartesian coordinates. The state vector in this problem can be expressed as

$$\mathbf{x} = (x, y, \dot{x}, \dot{y}, \ddot{x}, \ddot{y})^T, \tag{4}$$

where x and y are corner coordinates, $\dot{x}$ and $\dot{y}$ are velocities of corner motion, and finally, $\ddot{x}$ and $\ddot{y}$ are the corner accelerations. We use the Singer model for corners' motion modeling.[34,35] In the previous work,[36] we used a Wiener process acceleration model for this purpose. The Singer model assumes that the target acceleration is a zero-mean first-order stationary Markov process, whereas the Wiener model is referred to as a nearly constant acceleration model. They are compared in Sec. 9.

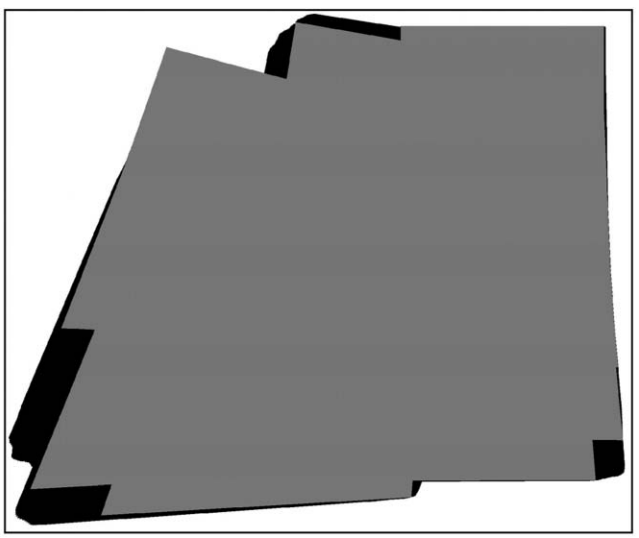

Fig. 8 The coverage rate of the proposed method for video 3. The black areas are parts of the scene that are not covered using the key frames.

Table 2 Comparisons between coverage rates (%) of the proposed method and the downsampling method.

Video	1	2	3	4
Down-sampling by a factor of 20	97.0	96.6	99.1	Fail
Proposed method	96.8	98.4	94.2	96.9

The transition matrix of the dynamic model is set to

$$\mathbf{A}=\begin{bmatrix} 1 & 0 & T & 0 & (\alpha T-1+e^{-\alpha T})/\alpha^2 & 0 \\ 0 & 1 & 0 & T & 0 & (\alpha T-1+e^{-\alpha T})/\alpha^2 \\ 0 & 0 & 1 & 0 & (1-e^{-\alpha T})/\alpha & 0 \\ 0 & 0 & 0 & 1 & 0 & (1-e^{-\alpha T})/\alpha \\ 0 & 0 & 0 & 0 & e^{-\alpha T} & 0 \\ 0 & 0 & 0 & 0 & 0 & e^{-\alpha T} \end{bmatrix}. \tag{5}$$

In transition matrix, [Eq. (5)], T is the step size and can be computed from the camera frame rate ($T = 1/\text{camera frame rate}$) and $\alpha = 1/\tau$. τ is the maneuver time constant and thus depends on how long the maneuver lasts.[34,35] In the correction step, we only measure the positions of each corner. Thus, the measurement matrix is set to

$$\mathbf{H}=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{6}$$

The misregistration error of SIFT features is considered as the measurement noise.

The initial position of each corner of the first frame is determined precisely, and their initial velocity and acceleration is assumed to be zero. The initial error covariance matrix is set to a diagonal matrix whose elements are equal.

6 Thresholds Updating

6.1 *Overlap Threshold*

The overlap area between two consecutive key frames must be balanced. A large overlap increases the computational cost without providing much data, and a small overlap results in less accurate image registration. To this end, a suitable overlap threshold is needed. The diversity of features in the overlap area also has an important role in the accuracy of alignment and depends on the context of images. In this work, it is assumed that a robust alignment can be achieved if the features' area is within 0.3 of the frame area (the area of the convex hull of features is considered as the features area). Equations (7) and (8) are used to determine the amount of the overlap threshold. It varies between 0.4 and 0.6 due to Eq. (8). In Eq. (7), OT′ will equal the previous threshold (pre − OT), if the diversity of features is good. In the initial step, the threshold is set to 0.6 and is updated after each alignment,

$$\text{OT}' = \text{preOT}\frac{0.3(\text{frame area})}{\text{convex hull area of features}}, \tag{7}$$

$$\text{OT} = \begin{cases} 0.4 & \text{OT}' < 0.4 \\ \text{OT}' & 0.4 \le \text{OT}' \le 0.6 \\ 0.6 & \text{OT}' > 0.6 \end{cases} \tag{8}$$

Furthermore, in situations where the number of matched features between two frames is < 200 or the absolute difference between the predicted overlap and the estimated overlap is > 0.15, OT is set to 0.6.

6.2 *Distance Threshold*

The distance between aligned frames is another important threshold. Until predicted overlap is less than OT, the Kalman filter is not updated. Thus, in addition to the overlap threshold, another threshold must be defined. If the distance between this predicted frame and previous key frame is larger than this threshold (DT), then the predicted frame is used for alignment and the Kalman filters are updated. The value of DT is computed as follows:

$$\text{DT} = \begin{cases} \max(10, \text{DT}'/2) & f < 200 \text{ or } |\text{pred.overlap} - \text{est.overlap}| > 0.15 \\ \min(20, \text{DT}' \times 1.5) & f > 200 \text{ and } |\text{pred.overlap} - \text{est.overlap}| < 0.15 \end{cases} \tag{9}$$

where DT′ is the previous value of DT and f is the number of matched features between two frames. With Eq. (9), the distance threshold (DT) will get three different values (10, 15, and 20). In the initial step, DT set to 10 and, during the process, it can increase in order to avoid extra Kalman filter updates. If the absolute difference between the predicted overlap and the estimated overlap is > 0.15, Kalman filters must be updated more frequently. Thus, the value of DT will be reset to 10.

7 Features Reduction

It is known that only features that are in the overlap area are suitable to align two frames. Removing features outside the overlap area improves the accuracy and speed of the alignment. All features of the predicted frame must be mapped to the common mosaic surface, and only those inside the predicted overlap area are selected for alignment. A homography matrix is needed to map them, and it is obtained from the four predicted corners of the frame.

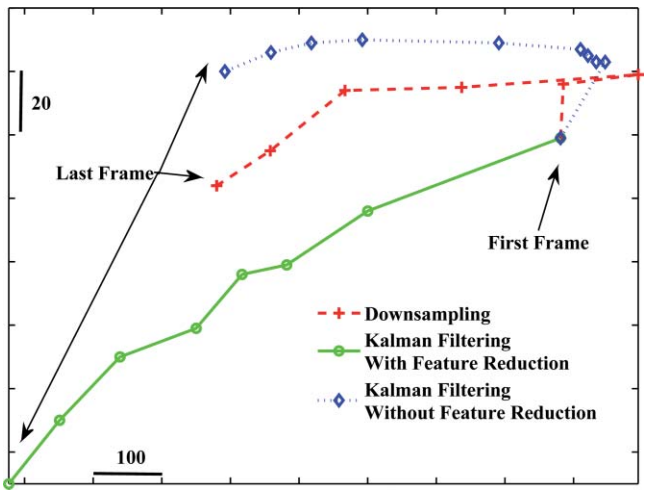

Fig. 9 Comparison between the estimated camera trajectories (coordinates of the upper left corner of the frames) on the common mosaic surface by three different approaches on video 4 (repeated structures). It can be seen that the two first methods produce a wrong camera-movement model at the start of their trajectories.

Figure 4 shows two aligned frames. In Fig. 4(a), the lower frame with point features is mapped onto the upper frame with circle features according to the prediction results and, in Fig. 4(b), the features outside the overlap area are removed. Some features may be removed incorrectly, but with respect to all features, they are not many [Fig. 4(c)]. In the case of images with repetitive patterns, this feature reduction plays an important role in the registration process and avoids misregistration. This is also shown in Sec. 9.

8 Enhancement of Key Frame Selection

Figure 5 shows part of the camera movement trajectory of one test video (video 3). The sign of camera motion velocity is changed in position C. The proposed method selects the frames in positions A and B as key frames and tracks the camera motion. The overlap area between the current frame and the last key frames (B) decreases steadily, but before the overlap area goes less than the threshold, the camera returns. Therefore, the frame in position D is selected as the next key frame (not any frames between B and D). As shown in Fig. 5, selecting a key frame between B and D improves the results and expands the area covered by key frames. An enhancement step is used to find such key frames. In this step, the degree between each of the three consecutive key frames (α) is calculated. If the degree is > 90 and the distance between B and D is large enough (for example, more than 50 frames), one aligned frame between these two key frames (for instance, $\Diamond$ in Fig. 5) is selected as a new key frame. This new key frame is selected in a way that covers the smallest overlap area.

Table 3 Run time of different methods (in 1000 s.)

Video	1	2	3	4
Aligning all frames for key frame selection	3.68	22.08	15.27	0.49
Proposed method without feature reduction	.27	1.30	0.77	.07
Proposed method with feature reduction	.25	.59	.45	.06

9 Results

In this section, implementation results of the proposed key-frame selection method are presented. The method was implemented in MATLAB, and several experiments were arranged. All video sequences were captured in a 360×640 resolution.

The total number of frames of the test video shots and the number of their aligned frames are shown in Table 1. The number of key frames extracted from the proposed method is specified in the forth row. In all video streams, the first frame is considered as the first key frame. The last frame is also selected as a key frame if the distance from its previous key frame is more than the distance threshold.

Figure 6 shows the trajectories of the upper left corner of frames in videos 1, 2, and 3 in the common mosaic surface. The dotted lines are the predicted trajectories, and the solid lines show the exact trajectory of corners that were obtained through alignment. Plus marks in Fig. 6 show the upper left corners of the key frames, and $\Diamond$ show key frames that are selected after enhancement.

A stigma is seen in the corner trajectory of video 2 [Fig. 6(b)]. This is because a zoom-in followed by a zoom-out exists in the camera motion of video 2. It shows that Kalman filtering of this algorithm can easily recognize zooming functions of the camera and can be used in multiresolution panorama.

Figure 7 shows all the key frames of videos 1–3 in the common mosaic surface. The existence of overlapping between nonconsecutive key frames can be determined from Fig. 7. Alignment of overlapped-nonconsecutive frames is also performed to increase overall accuracy. As an example in video 3, the frames numbered 1 and 7 are two nonconsecutive overlapped key frames. This alignment is only accomplished if the amount of overlapping is > 0.3. If the overlapped area is > 0.8, then one of the frames is removed. Consider a case where the camera has zigzag movements; in such cases, nonconsecutive alignments also appear. If no camera-movement estimation is done for a sequence of k different key frames, then a $k(k-3)/2$ overall extra alignment checking is needed to find the nonconsecutive alignments. However, with the proposed method, these overlaps are simply detected on a common mosaic surface without extra alignments. The numbers of overlapping key frames (consecutive or nonconsecutive) of videos 1–4 are given in Table 1.

As shown in Table 1, the proposed method is quite different from and more advantageous than a simple downsampling method. For example, 44 frames are selected with a downsampling rate of 20/1, while only four frames are needed

Table 4 Average of DT for test videos.

Video	1	2	3	4
Average of DT (frames)	11.9	19.1	19.4	9.3

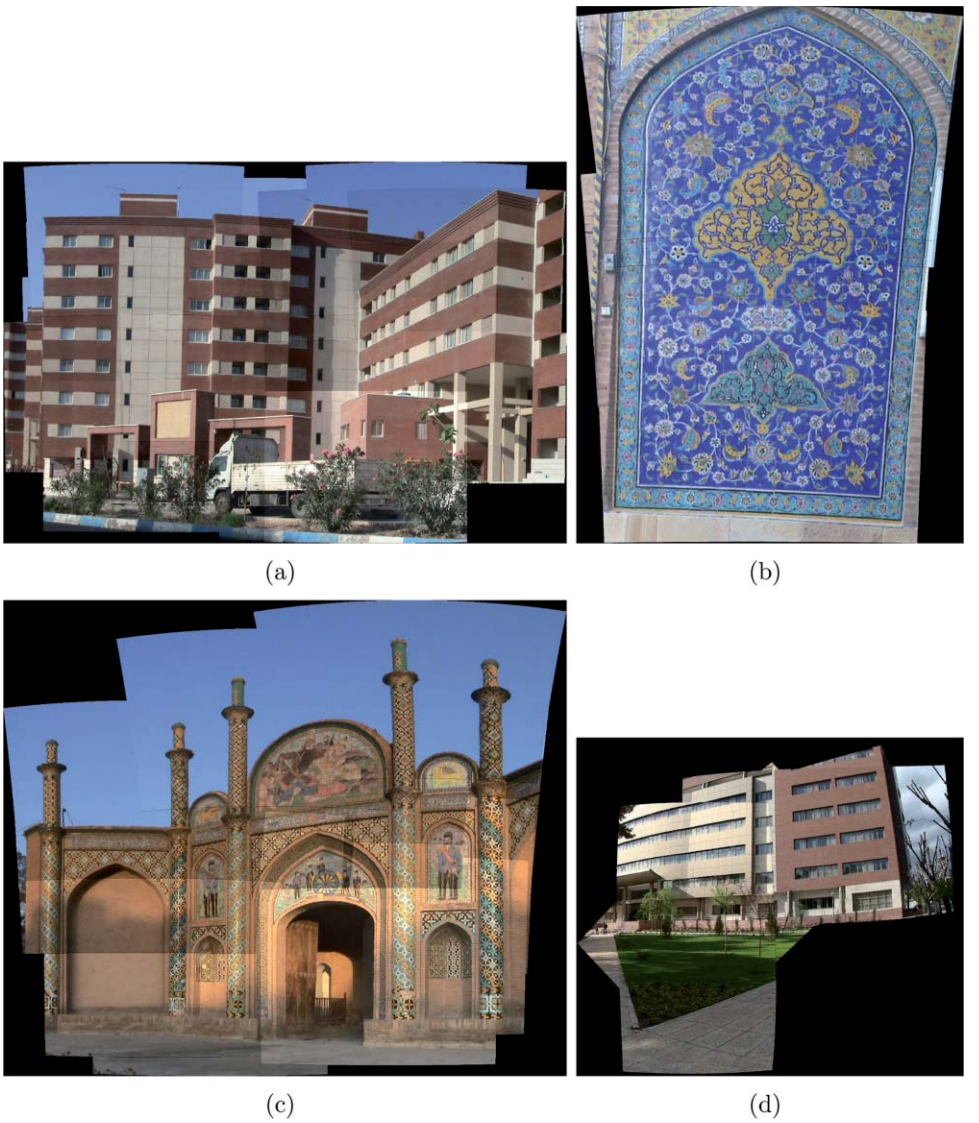

Fig. 10 Panoramic views constructed from four different videos with rotating cameras: (a) 1, (b) 2, (c) 3, (d) video from camera with roll motion.

for panorama creation of video 2. Furthermore, the precise camera behavior (zoom-in, zoom-out) is detected and, also, selection of the sampling rate is an important problem in the downsampling process. It is completely dependent on the camera-movement velocity and a constant number can never be suitable.

Some parts of the scene are inevitably lost if not all frames are used for panorama creation (Fig. 8). For the purpose of evaluating the proposed method, the whole scene was created using all frames, with no frames dropped. Coverage rates of the scenes for the proposed method and the downsampling method are shown in Table 2. The results obtained by the proposed method are very close to the results of the downsampling method and, in some cases, are better. whereas, the number of the key frames used in the proposed method is much less. It should be noted that the lost data are small parts of the surrounding area of the scene and not so important. On the other hand, sometimes the downsampling method fails to create panorama correctly from video (for instance video 4). This will be described in detail.

The method in Ref. 5 aligns all frames of a video to extract key frames. This is very time-consuming. Table 3 shows the required times to extract key frames from videos using the proposed method, with or without feature reduction, and the time to align all frames. The experiments were performed on a laptop with an Intel core 2 duo 2.5-GHz CPU.

The average number of SIFT features extracted from each frame of videos 1 and 2 is 1617 and 6378, respectively. This difference in the number of extracted fearures directly affects the computational time to select key frames.

Table 5 Comparisions Between the number of selected key frames in the method of Ref. 36 and the proposed method.

Video	1	2	3	4
Method of [36]	24	5	19	Fail
Proposed method	21	4	12	8

Table 6 Average alignment error for different methods (in pixel).

Video	1	2	3	4
Aligning all frames for key frame extraction	0.58	0.43	0.01	0.55
Proposed method without feature reduction	0.60	0.50	0.49	Fail
Proposed method with feature reduction	0.61	0.47	0.46	0.67

Table 4 shows the average of DT during execution of the algorithm. As can be seen, it varies for different videos. It directly depends on the speed and the type of camera movement and video content.

In Table 5, the number of selected key frames using the method of Ref. 36 and the proposed method are given for the four test videos. As can be seen, the number of key frames in the method of Ref. 36 are more than that of the proposed method, because Ref. 36 uses a fixed distance threshold (20) compared to a variable threshold in the proposed method. The method of Ref. 36 fails to extract key frames on video 4, because the feature-reduction step has not been used.

In the proposed method, feature mismatching of the repeated structures will not occur because it predicts the position of each frame using video-stream information. The camera moves on a straight line in video 4 and captures a repetitive structure. Figure 9 shows three movement trajectories with three different approaches on video 4. These approaches are (i) downsampling by a factor of 20, (ii) Kalman filtering with feature reduction and (iii) Kalman filtering without feature reduction. As shown in Fig. 9, the alignment methods without feature reduction fail to estimate the camera movement, whereas the Kalman filtering prediction with feature reduction estimates it correctly. The feature-reduction step removes unrelated features, which yields correct alignment.

In all experiments, a similar alignment algorithm between two frames is used; thus the only parameter that can affect the accuracy of matching is the amount of overlap area between two frames. The average alignment error of different methods between two frames is shown in Table 6. As can be seen, alignment accuracy is not changed much and still remains at < 1 pixel. This is due to proper distribution and good precision of SIFT features.

The Singer and Weiner models were used in the design of a Kalman filter to compare their results. In both cases, variable thresholds and feature reduction were applied. Both methods were run for 10 times, and the root-mean-square errors (the distance in units of pixels on a common mosaic surface) for the best, worst, and the mean cases are reported in Table 7. It can be seen that the Singer model yields better results.

Table 7 Rms errors of the Weiner and the Singer models.

	Weiner			Singer		
	Min	Max	Mean	Min	Max	Mean
Video 1	26.7	27.3	27	28	29.4	28.6
Video 2	16.2	17	16.9	14.2	14.8	14.6
Video 3	32.8	36.6	35.9	29.9	34.4	31.5
Video 4	59.7	88.2	67.1	55.7	68.9	61.6

Fig. 11 A panoramic view with repetitive structure constructed from a video with translating camera (Video 4).

The panoramic views of videos are shown in Figs. 10 and 11. To construct panoramas in Fig. 10, the bundle adjustment algorithm which has been developed in Ref. 1 is used. Using the method described in Ref. 24, frames were straightened and finally mapped onto a spherical surface.[2] The edges of some frames are visible in panoramic views of videos 1–3. A traditional and robust multiband blending algorithm[28] can be used to remove them. Figure 11 shows the panorama image that resulted from video 4 with a straight movement of the camera. This panoramic view is constructed without bundle adjustment; however, if we use bundle adjustment, then it will yield better results.

10 Conclusion

In this paper, a fast and accurate method for key-frames selection from a long video sequence to create panorama has been presented. In contrast to existing methods, this approach prevents the need of alignment of all frames of video stream by using the prediction of the camera motion. Because the thresholds vary depending on the video content and camera movement, a precise number of aligned frames could not be estimated. However, in the worst case, 0.1 of the frames are aligned to extract key frames. In addition, the nonoverlapping consecutive key frames can be simply detected on the camera mosaic surface without extra alignment. In this work, the homography matrix was used to map frames onto the common mosaic surface. The proposed method uses the information of video to reduce features and align frames with repeated structures more accurately. Any loop and zooming in the camera path can be detected easily as the key frames are projected onto the common mosaic surface. The only restriction to the camera motion is the degree of pan and tilt, which should be $<$ 180 deg, as long as the common mosaic surface is flat.

References

1. M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *Int. J. Comput. Vis.* **74**(1), 59–73 (2007).
2. R. Szeliski and H.-Y. Shum, "Creating full view panoramic image mosaics and environment maps," in *Computer Graphics (SIGGRAPH'97)*, p. 251–258 (1997).
3. H. S. Sawhney, R. Kumar, G. Gendel, J. Bergen, D. Dixon, and V. Paragano, "Videobrush: Experiences with consumer video mosaicing," in *Proc. of IEEE Workshop on Applications of Computer Vision*, p. 56–62 (1998).
4. H. S. Sawhney, S. Hsu, and R. Kumar, "Robust video mosaicing through topology inference and local to global alignment," in *Proc. of European Conf. on Computer Vision*, p. 103–119 (1998).

5. D. Steedly, C. Pal, and R. Szeliski, "Efficiently registering video into panoramic mosaics," in *Proc. of 10th Int. Conf. on Computer Vision*, p. 1300–1307 (2005).
6. R. Szeliski, "Video mosaics for virtual environments," *IEEE Computer Graphics Appl.* **16**(2), 22–30 (1996).
7. T. Mei, X.-S. Hua, H.-Q. Zhou, S. Li, and H.-J. Zhang, "Efficient video mosaicing based on motion analysis," in *Proc. of Int. Conf. on Image Processing*, p. 861–864 (2005).
8. R. Marzotto, A. Fusiello, and V. Murino, "High resolution video mosaicing with global alignment," in *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, p. I692–I698 (2004).
9. N. Gracias and J. Santos-Victor, "Underwater mosaicing and trajectory reconstruction using global alignment," in *Proc. of OCEANS Conf.*, p. 2557–2563 (2001).
10. S. Peleg and J. Herman, "Panoramic mosaics by manifold projection," in *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, p. 338–343, (1997).
11. S. Peleg, B. Rousso, A. Rav-Acha, and A. Zomet, "Mosaicing on adaptive manifolds," *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(10), 1144–1154 (2000).
12. Y. Wexler and D. Simakov, "Space-time scene manifolds," in *Proc. of 10th Int. Conf. of Computer Vision*, p. 858–863 (2005).
13. Z. Zhu, G. Xu, E. M. Riseman, and A. R. Hanson, "Fast construction of dynamic and multi-resolution 360° panoramas from video sequences," *Image Vis. Comput.* **24**(1), 13–26 (2006).
14. J.-W. Hsieh, "Fast stitching algorithm for moving object detection and mosaic construction," *Image Vis. Comput.* **22**, 291–306 (2004).
15. D.-H. Kim, Y.-I. Yoon, and J.-S. Choi, "An efficient method to build panoramic image mosaics," *Pattern Recogn. Lett.* **24**, 2421–2429 (2003).
16. H.-Y. Shum and R. Szeliski, "Construction of panoramic image mosaics with global and local alignment," *Int. J. of Comput. Visi.* **36**(2), 101–130 (2000).
17. P. F. Maclaunchlan and A. Jaenicke, "Image mosaicing using sequential bundle adjustment," *Image Vis. Comput.* **20** 751–759 (2002).
18. D. G. Lowe, "Distinctive image features from scale-invariant keypoints", *Int. J. Comput. Vis.* **60**(2), 91–110 (2004).
19. M. Brown and D. G. Lowe, "Recognising panoramas," in *Proc. of 9th IEEE Int. Conf. on Computer Vision*, p. 1218–1225 (2003).
20. A. Agarwala, K. C. Zheng, C. Pal, M. Agrawala, M. Cohen, B. Curless, D. Salesin, and R. Szeliski, "Panoramic video textures," *ACM Trans. Graphics* **24**(3), 821–827 (2005).
21. J. Civera, A. J. Davison, J. A. Magalln, and J.M.M. Montiel, "Drift-free real-time sequential mosaicing," *Int. J. Comput. Vis.* **81**(2), 128–137 (2009).
22. C. Morimoto and R. Chellappa, "Fast 3d stabilization and mosaic construction," in *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR'97)*, p. 660–665 (1997).
23. D. Kim and K. Hong, "Real-time mosaic using sequential graph," *J. Electron. Imaging* **15**(2), 023005 (2006).
24. R. Szeliski, "Image alignment and stitching: A tutorial," *Foundation and Trends in Computer Graphics and Vision* **2**(1), 1–104 (2006).
25. Y. Boykov, O. Vexler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.* **23**(11), 1222–1239 (2001).
26. N. Gracias, M. Mahoor, S. Negahdaripour, and A. Gleason, "Fast image blending using watersheds and graph cuts," *Image Vis. Comput.* **27**(5), 597–607 (2009).
27. V. Kwatra, A. Schodl, I. Essa, G. Turk, and A. Bobick, "Graphcut textures: Image and video synthesis using graph cuts," *ACM Trans. Graphics* **22**(3), 277–286 (2003).
28. P. J. Burt and E. H. Adelson, "A multiresolution spline with application to image mosaics," *ACM Trans. Graphics* **2**(4), 217–236 (1983).
29. A. Levin, A. Zomet, S. Peleg, and Y. Weiss, "Seamless image stitching in the gradient domain, " in *Proc. of 8th European Conf. on Computer Vision*, p. 377–389 (2004).
30. M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, " *Commun. ACM* **24**(6), 381–395 (1981).
31. R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed., Cambridge University Press, Cambridge, England (2003).
32. C. Wang, J.-H. Kim, K.-Y. Byun, J. Ni, and S.-J. Ko,"Robust digital image stabilization using the Kalman filter," *IEEE Trans. Consumer Electron.* **55**(1), 6–14 (2009).
33. G. Welch and G. Bishop, "An introduction to the kalman filter," Tech. Report No. 95-041, University of North Carolina at Chapel Hill, (1995).
34. R. A. Singer, "Estimating optimal tracking filter performance for manned maneuvering targets," *IEEE Trans. Aerospace Electron. Syst.* **AES-6**(4), 473–483 (1970).
35. X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. Part I: dynamic models," *IEEE Trans. Aerospace Electron. Syst.* **39**(4), 1333–1364 (2003).
36. M. J. Fadaeieslam, M. Fathy, and M. Soryani, "Key frames selection into panoramic mosaics," presented at *7th Int. Conf. on Information, Communications and Signal Processing* (2009).

Mohammad Javad Fadaeieslam recievied his BSc and MSc in computer engineering from Iran University of Science and Technology in 2003 and 2005. He is currently a PhD student in the field of image processing and computer vision. His research interests include extracting panorama image from video for traffic monitoring and other applications.

Mohsen Soryani recievied his BSc in electrical engineering from Iran University of Science and Technology (IUST) in 1980 and MSc in digital techniques and PhD in electronics (image processing) in 1986 and 1989, respectively, from Heriot-Watt University, Edinburgh, Scotland. He was with the Department of Electical Engineering, Mazandaran University from 1990 to 2002. Since 2002, he has been at the School of Computer Engineering of IUST as an assistant professor. His research interests include image processing, computer vision, and advanced computer architecture.

Mahmood Fathy recievied his BSc in electronics from Iran University of Science and Technology in 1985, MSc in computer architecture in 1987 from Bradford University, United Kingdom, and PhD in image processing computer architecture in 1991 from UMIST, United Kingdom. Since 1991, he has been an associate professor in the Computer Engineering School of IUST. His research interests include image and video processing, in particular, in traffic engineering and QoS in computer networks, including video and image transmission over the Internet.