Tracking and graph-cut based approach for panoramic background construction

Mohammad Javad Fadaeieslam
Mohsen Soryani
Mahmood Fathy

Tracking and graph-cut based approach for panoramic background construction

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. *An efficient method is presented for extracting motion behaviors and contours of moving objects in a wide view and for creating panoramic background. In the field of making panorama, the main goal of existing methods is to create a pleasing wide view. For this purpose, such methods do not track moving objects. They attempt to find optimal seams so that the result does not contain cut objects or blurring. Hence, moving objects are removed, repeated, or placed in an arbitrary location in the final panoramic image. We expand panorama applications from artistic views to surveillance usages. To investigate moving object behavior, the proposed method attempts to find correspondences between positions of a moving object in different selected frames by using SIFT features. It also presents a new approach to combine various types of information in order to extract the exact boundary of moving objects in moving cameras. The required information is obtained from the moving object's corresponding areas in other frames. Experiments were arranged to demonstrate the effectiveness and robustness of this method. The results show that this method, which uses fewer frames, is able to create better panoramic background compared with the existing methods. © 2013 SPIE and IS&T [DOI: 10.1117/1.JEI.22.4.041122]*

1 Introduction

Creating a panoramic view is an interesting field of research in computer vision, and a number of robust algorithms have been proposed.[1,2] Panorama synthesis has recently found commercial applications,[3,4] and attempts have also been made to create panorama on some mobile phone devices.[5] Creating a pleasing wide view is the main goal of many methods. To achieve this, researchers do not detect behavior of moving objects. They attempt to find optimal seams so that the result does not contain cut objects or heavy blurring. Hence, moving objects are removed, repeated, or placed in arbitrary selected locations in the final panoramic image.[3,5]

In some applications, such as surveillance and traffic monitoring, it is important to investigate the behavior of moving objects in wide areas. Extracting the behavior and contour of moving objects from a video in a wide area and creating a panoramic background view are the main purposes of this paper. In other words, we attempt to expand the panorama applications from artistic views to surveillance usages.

The main steps of the proposed method are (1) frame selection, (2) identification of the difference between each two consecutive selected frames, (3) finding correspondences between regions of difference to detect the behavior of moving objects, and (4) extracting the exact boundary of moving objects using graph-cut technique and panoramic background creation.

After frame selection, the proposed method calculates the differences of all consecutive selected frames. Then it uses areas with large differences to detect moving objects. We call these areas regions of difference (RoDs). In the next step, the correspondences between RoDs are detected. These correspondences are necessary for the algorithm to track moving objects in the scene.

Frame differencing is inadequate for exact boundary detection. Thus, this paper presents a new approach to extract the exact contours of moving objects in moving cameras. For this purpose, it segments the corresponding RoDs and then converts the moving objects boundary extraction problem to the binary labeling of these segments. It uses various types of information that are obtained from corresponding RoDs, which are the positions of a moving object in other frames, for labeling process.

The graph-cut algorithm[6–9] is widely used to solve labeling problems in computer vision. In labeling problems, a set of pixels or segments is labeled in such a way that an energy function in the standard form of Eq. (1) is minimized.

$$E(L) = \sum_{p \in P} D_p(L_p) + \sum_{(p,q) \in N} V_{p,q}(L_p, L_q). \qquad (1)$$

In this equation, P is the set of segments, L is the set of labels, and N is the set of all pairs of adjacent segments. $D_p(L_p)$ represents the cost of assigning label L_p to segment p and $V_{p,q}$ represents the cost of assigning different labels to the adjacent segments of p and q. Labeling RoDs in segment level is faster than labeling them in pixel level. It also reduces sensitivity to noise. After detection of moving objects' boundaries, background is constructed from key frames by the removal of moving objects.

The next parts of this paper are organized as follows. Section 2 reviews the related works. Section 3 describes

Paper 13197SS received Apr. 14, 2013; revised manuscript received Sep. 30, 2013; accepted for publication Oct. 18, 2013; published online Dec. 5, 2013.

the proposed method. Experimental results are shown in Sec. 4, and the conclusion is given in Sec. 5.

2 Related Works

The algorithms for generating panorama usually have two main steps in the literature: image alignment and blending. Image alignment is an important part of these algorithms, and its methods can be categorized in two groups:[1] direct (pixel-based) methods and feature-based methods. With innovation of robust features, like scale invariant feature transform (SIFT) (Refs. 2, 10, and 11), feature-based methods have become more popular particularly since they are robust in the existence of moving objects in the scene.

After matching the images that participate in a panoramic view, blending is a necessary step to overcome some artifacts made by misregistration errors, exposure differences, vignetting, parallax, lens distortion, and moving objects in the scene. Transition smoothing and optimal seam selection are two main approaches for this purpose. Multiband blending[12] is a robust transition smoothing technique that blends low-frequency bands of images over a large space and high-frequency bands of them over a short space. Brown and Lowe[2] have used this method for blending step in their efficient fully automatic image stitching method. Levin et al.[13] stitched images in the gradient-domain to avoid exposure differences. The transition approaches create ghosts in situations where large motions exist in the scene.

The optimal seam selection approaches[14–16] place a seam between two images in a region where transition from one image to another is not visible. In this way, a moving object is set to be remained in the image or removed entirely. Uyttendaele et al.[17] extract RoDs among aligned images by image subtraction and fill them from one image to prevent ghost creation. They do not detect moving objects in the RoDs. A fast moving object may cause two nonoverlapping RoDs, which are investigated separately. In each RoD, the corresponding object may be retained or removed. To produce a panoramic image, Zhu et al.[18] use all frames of a video. They detect initial positions of moving objects by image differencing and apply active contour model to refine moving objects' boundaries. They use a region grouping procedure after image differencing because the displacement of a moving object in consecutive frames is small and the moving object may correspond to several disconnected nearby regions. The global motion of the camera is panning and zooming in their method. Mills and Dudek[3] use both optimal seam selection and multiband blending methods to combine two images. To stitch more than two images, first the two best-matched images are selected to stitch and then, in an iterative process, the next best image that can be matched to one of them is selected and stitched to the mosaic.

The aim of almost all of the works mentioned above is to generate a pleasant and clear panoramic view from images. They do not extract moving objects, which may result in a moving object being removed or repeated in the final view. Objects should not be intersected by seams and ghosting and blurring should not occur.

In a pleasing panorama, seams between original frames are not detectable by viewers and the final image clarity is not lower than them. As mentioned before, several algorithms have been developed for this purpose. However, almost all of them have not evaluated their results quantitatively.[2,3,5] The proposed algorithms in the literature use different policies to synthesize the panoramic view, and as a result their products may be different in view point, scale, position of moving objects in the final view, etc. This is why it is difficult to evaluate them quantitatively.

In addition to creating pleasing panoramas, the proposed method finds the exact contours of vehicles, which are useful in traffic monitoring. In other words, our method attempts to extract foreground objects in the videos captured by moving cameras. For this purpose, some papers proposed methods to model the reference background image, but background modeling of outdoor scenes remains a very difficult problem.[19]

Lim et al.[20] developed an algorithm to extract the contours of nonrigid objects in videos captured by moving cameras. They did not use an exact registration method. But similar to our method, they use graph-cut to combine two types of information and extract exact contours. They use all frames and apply graph-cut in pixel level, which makes it more time consuming.

3 Algorithm Overview

3.1 *Frame Selection*

A video sequence consists of several hundred frames. Consecutive frames have large overlap areas, and as a result, creating a panoramic image from these frames, which do not provide much information, is a very time-consuming process. Moreover, subtracting consecutive frames causes fragmentation of moving objects that have large portions with smooth textures. Due to this fact, this algorithm uses the method proposed in Ref. 21 to extract suitable frames from a video for panorama generation. The camera motion is predicted in Ref. 21 and frames with a suitable overlap are selected as key frames. If the distance between the numbers of two key frames is high, this method also aligns one or more frames between these two key frames for correcting prediction of camera motion.

Our proposed algorithm considers all the aligned frames in Ref. 21 to investigate the behavior of moving objects. However, only key frames participate in the generation of the panoramic view. In other words, a high distance between two key frame numbers states that the time interval between them is large and the camera moves slowly. Therefore, we need some extra frames between them to detect moving objects' behavior in this interval. SIFT features and RANdom SAmple Consensus (RANSAC) methods are used in Ref. 21 to estimate the homography matrix between frames.

3.2 *Identification of Regions of Difference*

Determining the difference between frames is a common solution to find moving objects. For reaching this goal, each two consecutive selected frames are aligned and the pixels in the overlap area, which have large differences, are considered candidates for RoDs. The proposed method uses the information obtained from SIFT features to find suitable adaptive thresholds and refine candidate pixels for RoDs. SIFT is a local descriptor, and therefore, it can be assumed that there is no moving object around the corresponding SIFT features in two consecutive selected frames. The proposed

method uses these features to create a mask and searches to find the moving objects in the area out of the mask.

Camera motion may cause the variation of illumination in a video sequence. In this situation, adaptive difference threshold leads to better results. For this purpose, the mean difference of corresponding SIFT points is used as a criteria to determine the adaptive threshold. The steps of RoDs identification are as follows:

a. Corresponding SIFT features are extracted from two consecutive aligned frames.
b. To create a mask, a 10×10 window is centered on each SIFT feature at first. Then a morphological close operation is applied on the whole mask to produce a more smoothed mask.
c. The mean difference of corresponding SIFT points (MD) is determined as a criteria for threshold (T).
d. Pixels are selected if they meet the following three criteria: they are in the overlap area, they do not fall in the mask, and their difference is greater than $T = 2.5 \times \mathrm{MD}$.
e. The small connected components are removed and after that the morphological close operator is applied to the remaining blobs.
f. Blobs that are extracted from the i, $i-1$ and i, $i+1$ difference form the RoDs of frame i. If two blobs have a common area, the method merges them and produces a single RoD with a specified common area (CA). Figure 1 shows some of these steps, executed on two selected frames of a video sequence.

3.3 *Finding Corresponding Areas Between RoDs*

Since this method processes a few frames of a video, the relations between RoDs in two consecutive selected frames must be determined to detect the trajectories of the moving objects and extract background. It should be mentioned that the distance between the image positions of the same object in two consecutive selected frames may be high so that the detection of correspondences would not be done easily. This would be even harder in the presence of multiple moving objects. The proposed method finds correspondences between RoDs using SIFT features. SIFT features of each RoD in frame i are matched to SIFT features of each RoD in frame $i+1$ and $i-1$. Two RoDs are correspondent if RANSAC method finds a projective transform (H) between them. This concept is shown in Fig. 2, in which red and blue lines show matched features but only blue lines indicate geometrically consistent features (RANSAC inliers). If RANSAC fails to extract H or the matched SIFT features between two RoDs are <6, translation transform is used to detect corresponding RoDs.

As shown in Fig. 2, sometimes only a section of a moving object exists in the image (it occurs in conditions where the moving object enters into a frame or exits from it). There are other situations in which objects are changed in scale and/or rotation. SIFT features are robust in these situations.

It is assumed that moving objects move in such a way in the scene that there is up to one moving object in each RoD without any occlusion.

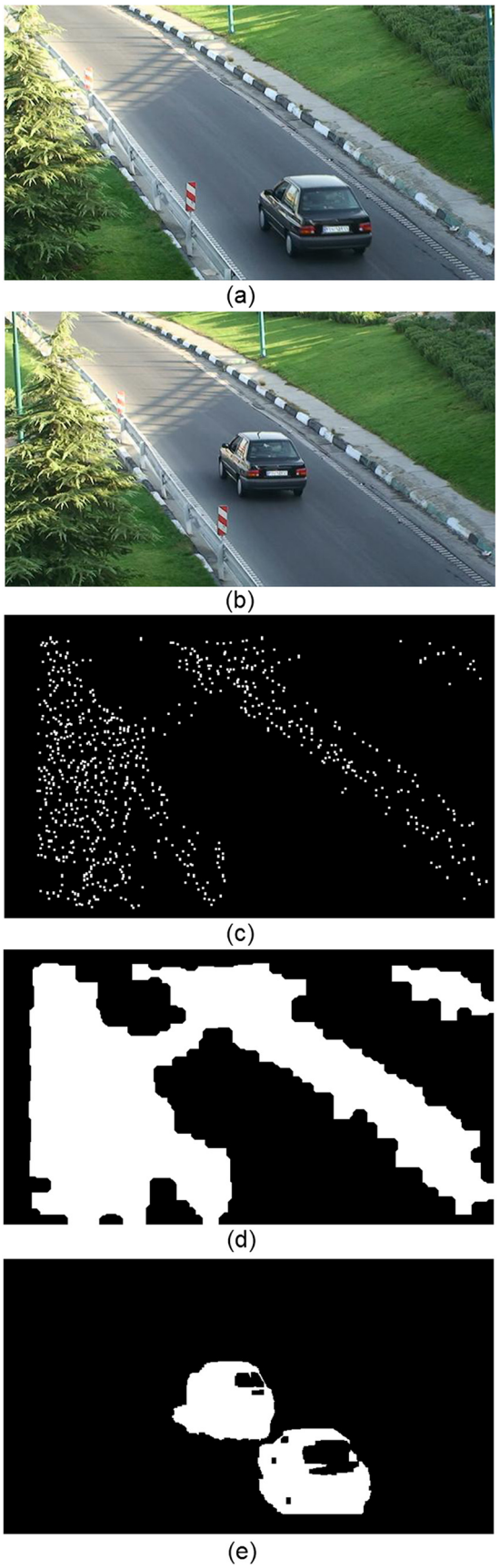

Fig. 1 (a) and (b) Two consecutive selected frames. (c) Corresponding SIFT features in one frame. (d) Mask obtained from SIFT features. (e) Regions of difference (RoDs) between them.

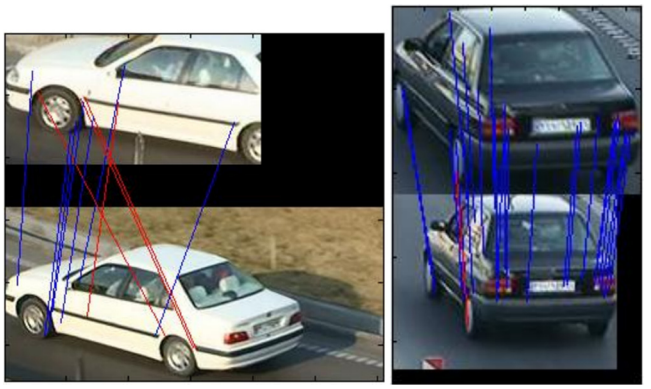

Fig. 2 Corresponding SIFT features. Blue lines indicate RANSAC inliers and red lines indicate RANSAC outliers. SIFT features are robust under rotation and scale variations.

An RoD in frame i that has no corresponding RoD is considered as background and is removed, if its overlap RoD in frame $i+1$ or $i-1$ has a corresponding RoD. In other words, a moving object is in frame $i-1$ or $i+1$ and its silhouette is in frame i.

3.4 *Extracting the Exact Boundary of Moving Objects Using Graph-Cut*

The difference between selected frames alone is not enough to separate moving objects. In some situations, RoDs consist of both moving object and parts of background. For example, if a moving object moves slowly, its position in frame i may overlap with its position in frame $i+1$ as shown in Fig. 3. Furthermore, it may be possible that multiple objects pass from one area of the scene as seen in Fig. 4. In this figure, three consecutive selected frames are shown in R, G, and B color components.

To determine the exact position of moving objects and construct a pleasing panoramic view, the exact boundary

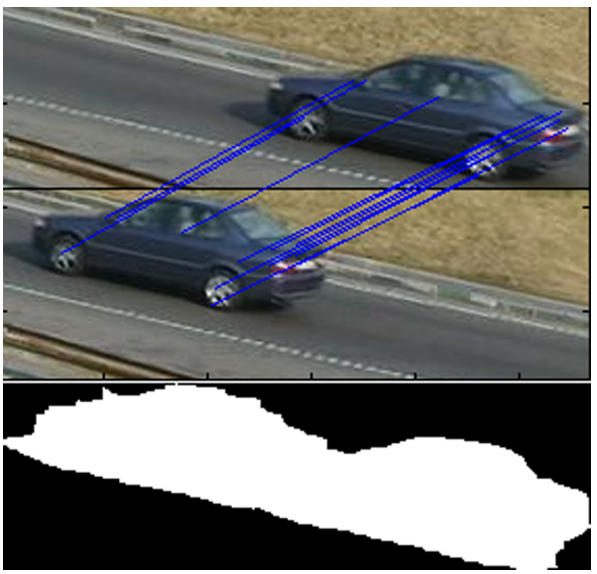

Fig. 3 A moving object and its RoD. Its position in selected frame i overlaps with its position in selected frame $i+1$.

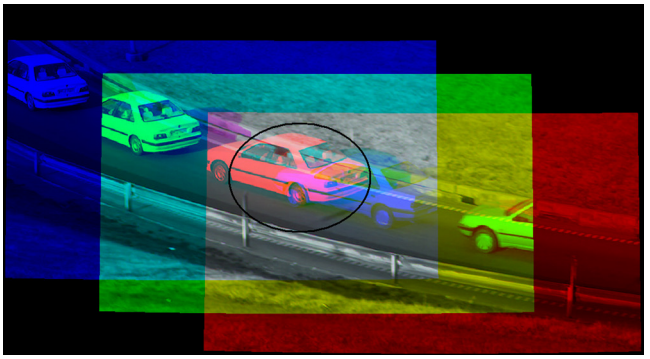

Fig. 4 Three consecutive selected frames are shown in this figure with different colors. Red figure is the first frame, green figure is the middle, and blue figure is the last one. There are two different cars inside the circle.

Algorithm 1 Steps to extract exact boundaries of moving objects.

Segment all RoD blocks

FBmat← Preprocessing

repeat

FBmat{i}← Graph cut(*FBmat*{i})

until all RoDs are processed

of them must be specified. Algorithm 1 shows the steps of the proposed method for exact boundary extraction.

In the first step, the algorithm obtains a surrounding box for each RoD and segments it. The surrounding box of each RoD is its bounding box that expands 15 pixels in each side. There are many algorithms for segmentation. Watershed[22] is a region-based segmentation method and has been extensively used in the literature. However, it may result in oversegmentation and needs a preprocessing or postprocessing step to overcome this drawback.[16,23] The robust region-based mean shift segmentation[24] algorithm is used to segment the RoDs. The mean shift algorithm preserves the desirable discontinuity characteristics of the image.

In the remaining steps, these segments are labeled as parts of the moving object or background.

The preprocessing step uses the information obtained from SIFT points (previously calculated in step 1 of Sec. 3.2), marginal area around bounding box, and CA (explained in step of Sec. 3.2) for the initial estimation of segment labels. A segment is considered as a part of moving object and labeled 1 in the following conditions: (1) a SIFT point falls in the segment or (2) at least half of its area is placed in CA. The segment is considered as background and labeled 0 in the following conditions: (1) some part of the segment is placed inside the marginal area (an area around the bounding box inside the surrounding area) or (2) more than half of the area is placed out of RoD. Otherwise, the segments are labeled -1. The label -1 expresses that the preprocessing step cannot determine an

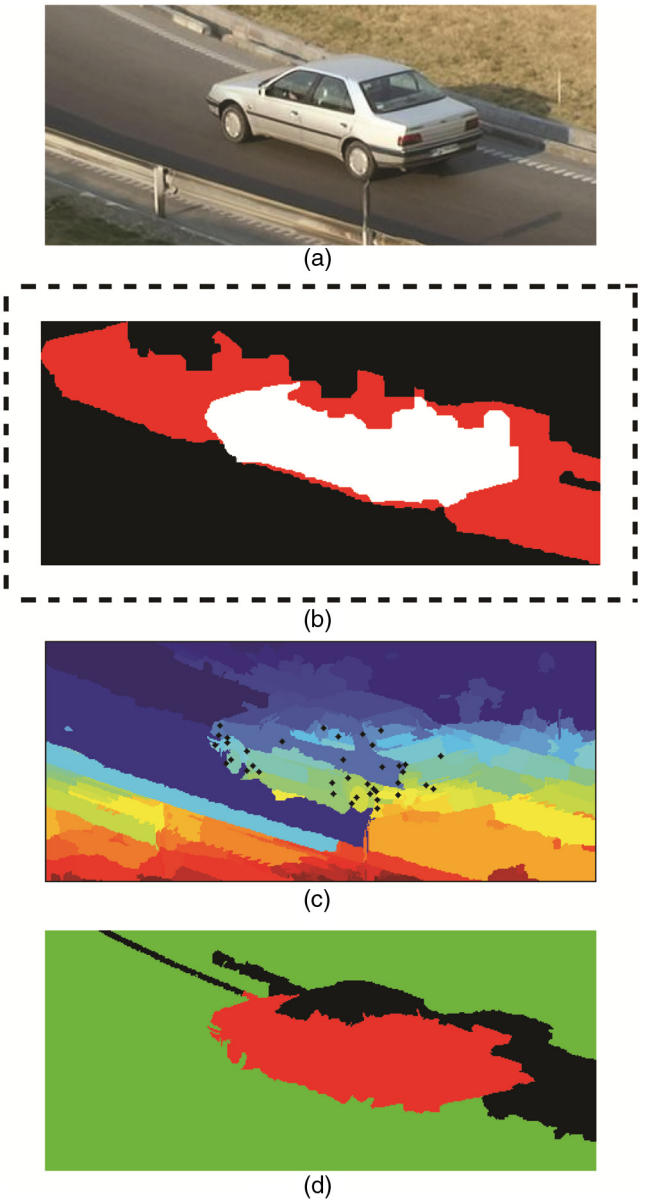

Fig. 5 (a) The moving object. (b) Its RoD is shown with red color and common area is shown with white color, the black box is bounding box, the dotted lines show the surrounding box. (c) The result of mean shift segmentation and SIFT feature. (d) The red area shows segments labeled 1, the green area shows segments labeled 0, and the black area shows unknown segments (label -1).

initial label for that segment. Figure 5 shows a moving object and the output of preprocessing function on it.

All RoDs are preprocessed and their labels are saved in Foreground Background matrix (*FBmat*). After the preprocessing step, in each iteration of the repeat loop, one RoD is selected and its labeling is optimized by graph-cut algorithm. Some papers run graph-cut method in pixel level.[14,25] Gracias et al. show in Ref. 16 that use of graph-cut in region level greatly reduces the search space for finding seams without affecting the quality, when compared to searching over all individual pixels in the overlap zone. So we use it in region level in this paper.

As seen in Eq. (1), we need D and V matrices to run graph-cut for labelling segments with minimum energy. $D_p(0)$ represents the cost of assigning label 0 to segment p, and $D_p(1)$ represents the cost of assigning label 1 to segment p.

This algorithm considers the dissimilarity between segment p and its similar area in corresponding RoD as the cost of assigning label 1 to segment p. An RoD in frame i can correspond to an RoD in the previous frame $(i-1)$, and we refer to it as *CrspPRoD* (corresponding previous RoD). Similarly, it can correspond to an RoD in the next frame $(i+1)$, referred to as *CrspNRoD* (corresponding next RoD) (Fig. 6). If p' and p'' are corresponding areas of p in *CrspPRoD* and *CrspNRoD*, $D_p(1)$ is obtained from Eq. (2):

$$D_p(1) = \begin{cases} min[dist(p,p'),dist(p,p'')] & \text{if } p \text{ labeled } -1, \\ 0 & \text{if } p \text{ labeled } 1, \\ maxdist & \text{if } p \text{ labeled } 0. \end{cases} \tag{2}$$

The *dist* function estimates the distance between two areas. It is explained in Sec. 3.5. The maximum dissimilarity obtained for segments with label -1 is considered as *maxdist*. It is possible that an RoD may have only one corresponding RoD, *CrspPRoD* or *CrspNRoD*, not both. In this situation, Eq. (2) becomes simpler and does not need to calculate the minimum.

Segment p has a corresponding area in frame $i-1$ and $i+1$, which is referred to as lp and np, respectively. The cost of assigning label 0 to segment p, $D_p(0)$, is obtained from Eq. (3). Some segments in RoD that are referred to as pb_x are labeled as background in the preprocessing step (x is the index of the background segments). In some situations, a background area divides into few segments by a moving object. It is possible that some of these segments are labeled as background in the preprocessing step. So the dissimilarity between pb_x and p are calculated in this equation to increase the precision of labeling.

$$D_p(0) = \begin{cases} min[dist(p,lp),dist(p,np),dist(p,pb_x)] & \text{if } p \text{ labeled } -1, \\ maxdist & \text{if } p \text{ labeled } 1, \\ 0 & \text{if } p \text{ labeled } 0. \end{cases} \tag{3}$$

$V_{p,q}$ represents the cost of assigning different labels to the adjacent segments of p and q. The proposed method uses the Euclidean distance (*Edist*) between the centers of two segments to estimate matrix V. Equation (4) shows this concept. α is a coefficient, which is determined relative to the size of moving object's segments.

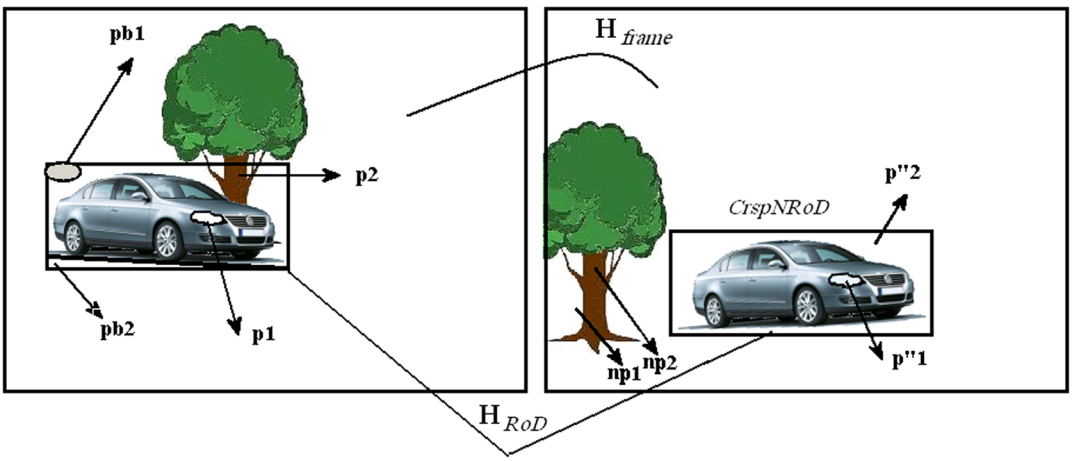

Fig. 6 Two consecutive selected frames. Left image is frame i and right image is $i+1$. H_{frame} maps two consecutive frames and H_{RoD} maps two corresponding RoDs. p_1 and p_2 are mapped onto np_1 and np_2 under H_{frame} and mapped onto p_1'' and p_2'' under H_{RoD}, respectively. p_1 is a segment on moving car and p_2 is a segment in background. So $D_1(1)$ is less than $D_1(0)$ and $D_2(0)$ is less than $D_2(1)$. pb_1 and pb_2 are two segments that are labeled as background in preprocessing step.

$$V_{p,q}(0,1) = V_{p,q}(1,0) = \frac{\alpha}{1 + Edist(p_{\text{center}}, q_{\text{center}})}. \quad (4)$$

After the above calculations, V and D are normalized (α is multiplied after normalization) and graph-cut algorithm is run. Graph-cut minimizes the labeling energy and extracts the exact boundary (labeling) of moving objects.

3.5 *Estimating the Distance Between Two Areas*

The proposed algorithm needs a region description method to describe similarity between segments. The covariance descriptor[26,27] is a good means for this purpose because it is robust against scale and large rotations. Porikli et al.[28] used this matrix as a region descriptor for the first time. Using this feature, we are able to fuse different types of low-level features into a small two-dimensional matrix efficiently. This matrix can be calculated fast based on integral images.

The pixels of a window that is centered at the center of a segment are used to construct the covariance descriptor. The width of a window varies between 10 and 30 pixels depending on the size of a segment. To extract this feature, each pixel of a window is converted to a nine-dimensional vector $F(x,y)$.

$$F(x,y) = [x, y, R, G, B, |I_x|, |I_y|, |I_{xx}|, |I_{yy}|]. \quad (5)$$

In this vector, x and y represent the location of the pixel, R, G, and B are color components, and I is pixel intensity. $I_x, I_{xx}, \ldots$ are intensity derivatives.The covariance of these vectors composes a 9×9 matrix to characterize the segment.[26]

The proposed method maps the center of each segment to its corresponding area (RoD and background) in the other frame using H_{RoD} and H_{frame}, respectively, and makes this point as the center of window. The size of this new window is equal to the size of the segment window. This window is considered as the corresponding area of the segment and its covariance descriptor is calculated for similarity measurement.

The covariance matrix space is not a vector space; therefore methods based on arithmetic differences cannot specify the difference between two covariance matrices. In this paper, the distance metric that is proposed by Foerstner and Moonen[29] [Eq. (6)] is used to calculate the dissimilarity between two covariance matrices (p_1 and p_2). In this equation, $\{\lambda_i(p_1,p_2)\}_{i=1\ldots n}$ are the generalized eigenvalues of p_1 and p_2.

$$dist(p_1,p_2) = \sqrt{\sum_{i=1}^{n} \ln^2 \lambda_i(p_1,p_2)}. \quad (6)$$

As shown above, for calculation of the covariance descriptor, a feature vector [such as Eq. (5)] must be specified. For this purpose, different papers use different feature

Table 1 Various types of feature vectors used in covariance descriptors.

27	$F(x,y) = [x, y, \lvert I_x\rvert, \lvert I_y\rvert, \sqrt{I_x^2 + I_y^2}, \lvert I_{xx}\rvert, \lvert I_{yy}\rvert, \arctan \frac{\lvert I_x\rvert}{\lvert I_y\rvert}]$
26	$F(x,y) = [x, y, R, G, B, \lvert I_x\rvert, \lvert I_y\rvert, \lvert I_{xx}\rvert, \lvert I_{yy}\rvert]$
28	$F(x,y) = [x, y, I, \lvert I_x\rvert, \lvert I_y\rvert]$
30	$F(x,y) = [R, G, B, I_x, I_y, I_{xx}, I_{yy}]$

Table 2 Specifications of the four experimental videos.

Video no.	1	2	3	4
Total number of frames of the video shot	226	432	405	151
Number of aligned frames	12	24	23	9
Number of key frames	4	10	8	8
Number of moving objects	1	2[a]	4	2

[a]One of these two objects was not detected; the reason is explained in Sec. 4.6.

vectors. Table 1 shows some of them. In the results section, their efficiencies are evaluated and compared.

4 Results

The proposed algorithm was implemented in MATLAB®, and some experiments were arranged. All video sequences were captured in a 360×640 resolution. The specifications of four videos are listed in Table 2. Key frames are extracted from these videos using the method described in Ref. 21. In the following subsections, we illustrate the results of our experiments and compare our results with existing methods.

4.1 *Variable Threshold for Frame Differencing*

As described in Sec. 3.2, camera motion may cause the variation of illumination in a video sequence. In this situation, the proposed method uses adaptive difference threshold to obtain better results (explained in step 4 of Sec. 3.2). Table 3 shows variations of threshold (T) for different videos to find RoDs.

4.2 *Selecting Suitable Feature Vector for Covariance Descriptor*

As mentioned in Sec. 3.5, different papers use different feature vectors to establish the covariance descriptor. Regarding this fact, an experiment has been set up to measure the efficiency of each vector in our application. Each segment of a moving object is considered, and its similarity to its corresponding regions in the background and the moving object are specified. The best feature vector produces a higher difference between these two similarities. The average of differences of these two similarities for segments of a moving object using various feature vectors can be found in Table 4. Details of each feature vector are given in Table 1.

Table 3 Variations of threshold (T) in different videos.

Video no.	1	2	3	4
Min	67	47	60	42
Max	125	137	90	200
Mean	90	77	73	84

Table 4 The efficiency of various feature vectors used in covariance descriptors.

Video no.	1	2	3			4	
Object no.	1	1	1	2	3	1	2
27	1.26	1.68	1.36	0.81	0.78	2.88	1.35
26	**10.98**	**3.23**	**2.53**	**1.65**	**4.24**	**7.66**	7.97
28	5.09	2.85	2.24	1.52	1.73	5.47	4.16
30	10.11	2.61	2.51	1.63	2.59	6.78	**8.74**

Note: The bold values are the maximum value of each column.

As is seen, color features, pixel locations, and gradients are good features for this purpose.

4.3 *Extracting Moving Objects' Contours*

As mentioned before, the proposed method uses graph-cut algorithm to optimize segmentation results. Matrix V, which is used in the energy function [Eq. (1)], shows the cost of choosing two different labels for two neighboring segments. As introduced in Eq. (4), this cost is dependent on the Euclidean distance of the two segments' centroids. Since the moving objects have different distances from the camera and their segments have different sizes, it seems that applying such cost for all moving objects does not produce the desired result. Therefore, a parameter α is

Table 5 Average size in pixels of moving objects' segments in different videos.

Video no.	1	2	3				4	
Object no.	1	1	1	2	3	4	1	2
Mean	98	125	140	65	91	106	175	155

Table 6 Accuracy of different methods in the extraction of moving objects' contours (%).

Video no.	1	2	3				4	
Object no.	1	1	1	2	3	4	1	2
Region of difference	74.2	71.6	81.5	86.5	91.8	75.2	84.4	79.1
Covariance descriptor	91.2	90.3	93.6	87.74	91.1	97.8	84.8	92.9
Graph cut with $\alpha = 1$	91.1	95.4	93.8	89.0	90.2	98.6	89.6	92.3
Graph cut with α relative to object size	92.3	95.7	93.1	89.2	88.5	98.6	89.9	92.5

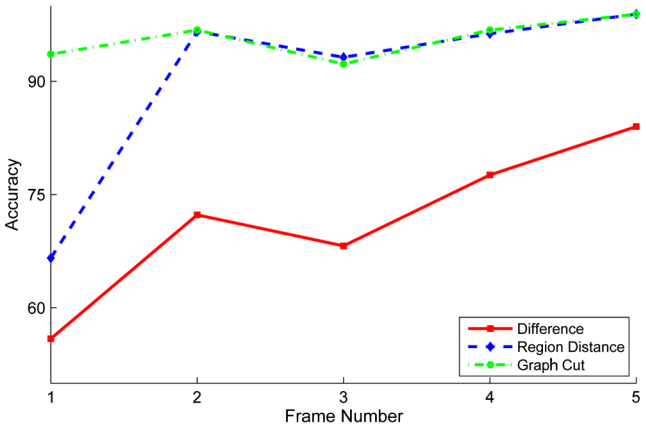

Fig. 7 Accuracy of the proposed method for extracting the moving object in video 2 in each selected frame.

considered and multiplied to the V value. This parameter is dependent on the size of the moving objects. The algorithm was run once with $\alpha = 1$ and once with $\alpha = (\text{Mean of segment pixels})/100$ to study the effect of this parameter. Table 5 shows the average segment sizes of different moving objects. Table 6 shows the average accuracy of different methods for extracting the moving objects' contours. To calculate the accuracy, we consider vehicles' contours, which are selected manually as ground truth, and then a bounding box that contains the RoD and the ground truth is specified. In this box, the contour accuracy is considered to be the ratio of the number of correctly labeled pixels to the total number of pixels.

The third row of Table 6 shows the accuracy of RoD that is acquired by differencing consecutive selected frames. In the fourth row, the labeling accuracy of covariance region

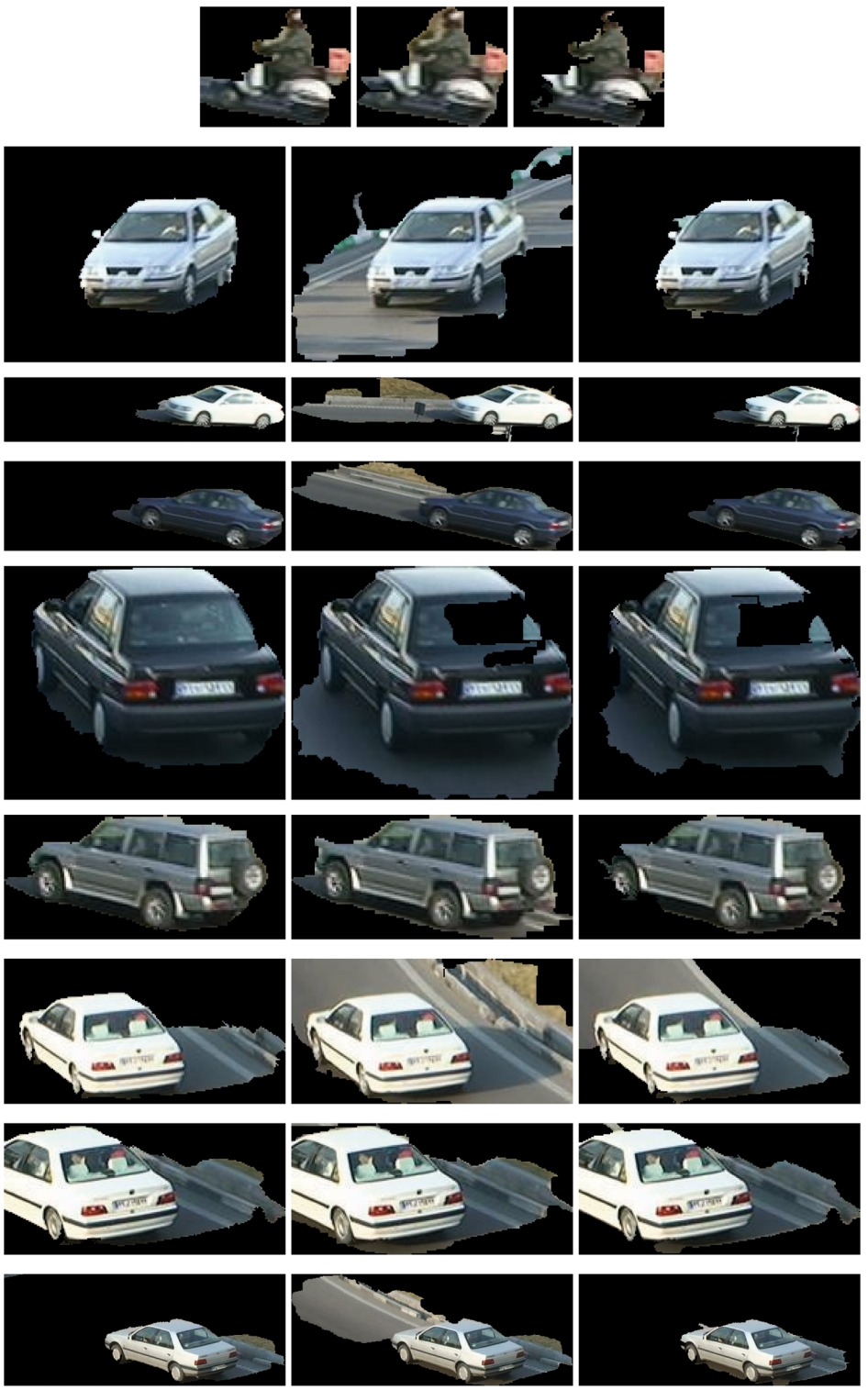

Fig. 8 Left column shows the foreground truth, middle column shows the RoDs, and right column shows the results of graph cut algorithm.

descriptor is given. The last two rows contains the results of the graph-cut method, with and without considering the object size. As shown in this table, the enhancement acquired by using parameter α is not considerable.

Figure 7 shows the accuracy of the proposed algorithm for extracting the moving object in each key frame of video 2. The red line illustrates the ability of frame differencing method in retrieving the contour of the moving object. The blue line gives the accuracy of extracting the moving object using covariance descriptor similarity, and the green line shows the results of the graph-cut method.

Figure 8 shows the results of the exact boundary detection for several moving objects. The left column shows the foreground truth, determined manually. The middle column shows the RoDs obtained from frame differencing. The results of graph-cut method are shown in the right column. As it can be seen, this method considers shadows as parts of moving objects. The directions of vehicles may change on the road, which makes the type of moving objects not fully rigid. But our method can handle such objects.

Active contour is one of the most effective methods for extracting moving objects' contours. There are different active contour models in the literature. In the first step of this method, a closed contour is created around the object. This contour is evolved step by step until it is completely aligned to the object edges.

Most active contour methods extract the object's contour without using the information about the positions of moving objects in the previous and the next frames. Therefore, when the moving object has several segments with different textures, active contour algorithms would have problems in finding region boundaries.

Chan and Vese[31] have introduced a level-set active contour method, which is noise tolerant. It works well in cases where images are blurred and object edges are not clear in the gradient image. We have used Chan and Vese's method for extracting object boundaries in moving camera and compared its results with our results.

To this purpose, for each moving object in each selected frame, a surrounding rectangle is considered. Then this rectangle is extended 5 pixels from each side and is used as the initial contour in Chan and Vese's algorithm. Performance of the algorithm for several objects is shown in Fig. 9. As can be seen, the efficiency of the algorithm in finding the contour of the moving object is clearly low.

4.4 *Creating Panoramic Background*

The panoramic views of all videos with the trajectories of moving objects obtained from the proposed method are shown in Fig. 10. We have used a simple bundle adjustment technique to produce the panoramic backgrounds. More powerful techniques can be used to create better results.

Background subtraction is one of the most common methods for moving object extraction. In this method, background is modeled using the information of all video frames, and the moving object is extracted by differencing each frame from the background model. There are many different methods for background modeling that are robust against noise and changes in illumination.[32] Colombari et al.[33] use this method. They assume camera motion and therefore align all frames for background modeling. The value of each pixel in the panoramic background is set to the median of the corresponding pixels in different frames.

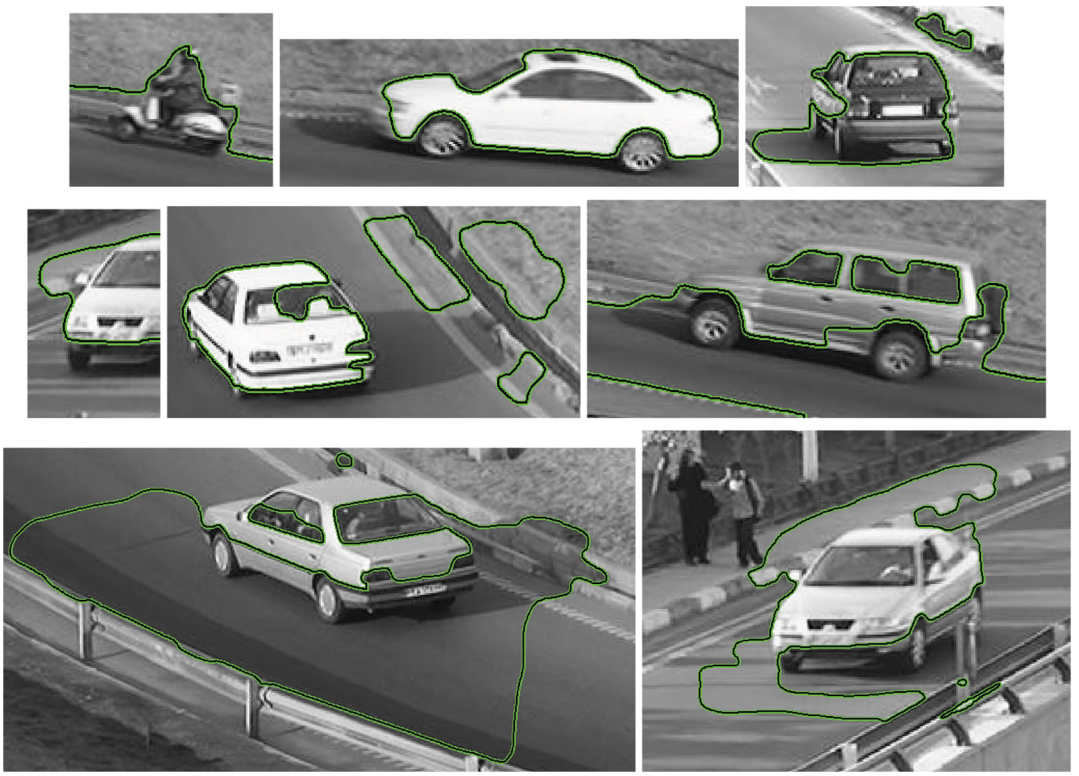

Fig. 9 Performance of the algorithm presented in Ref. 31 for extraction of some objects' contours.

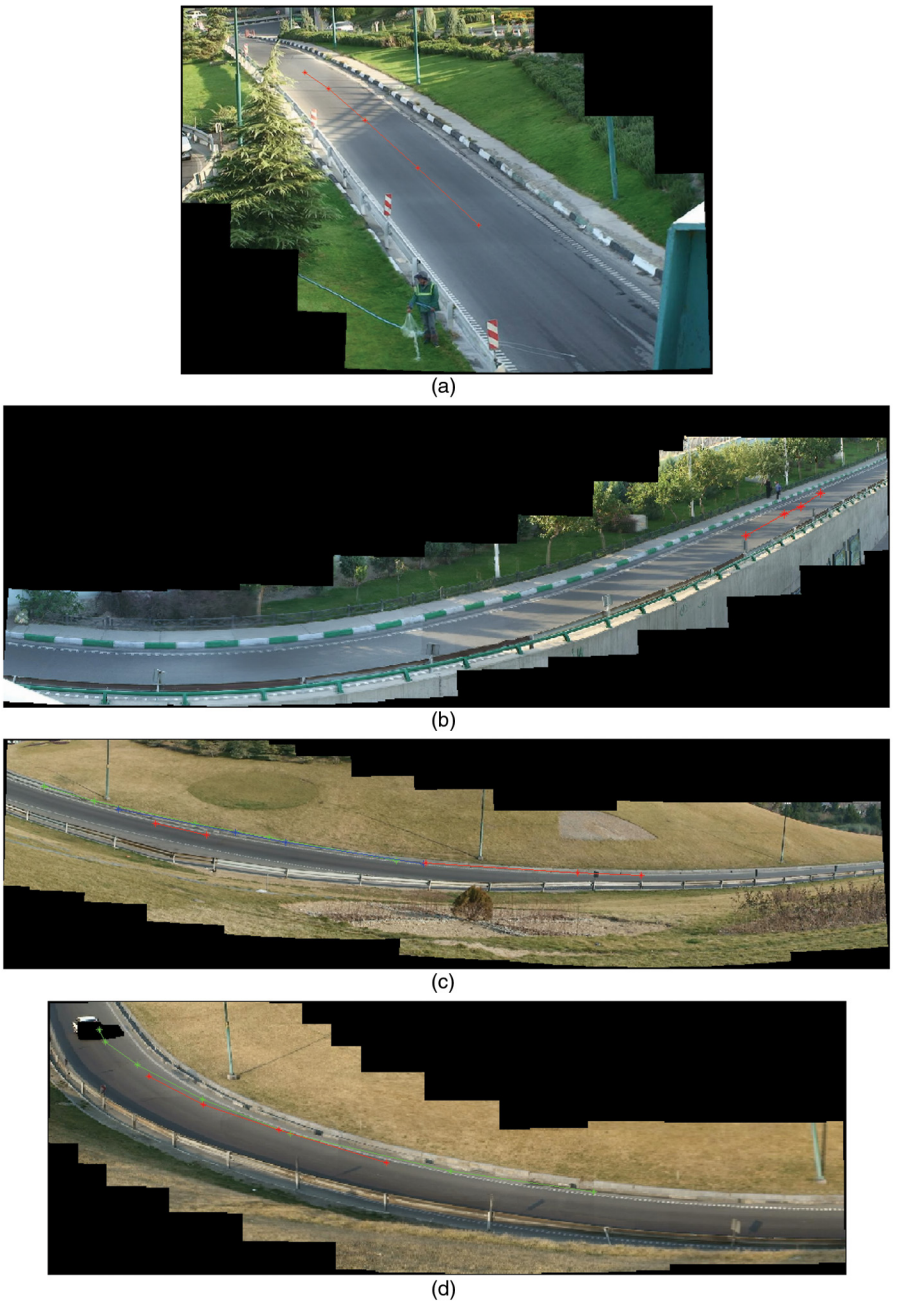

Fig. 10 Panoramic views and trajectories of moving objects for four videos produced by the proposed method.

In the presence of camera motion, background modeling is not a good method to construct panoramic backgrounds.[32,34] This is because misregistration errors cause the results to become blurred. Figure 11 shows the panoramic background of video 1, which is constructed using the Colombari's method. As can be seen, the final image is blurred and does not have a good quality because of alignment errors. To evaluate the sharpness, we used the no reference sharpness metric Q, which was presented in Ref. 35. Its values are 19.12 and 24.4 for Fig. 11 (Colombari) and Fig. 10(a) (our method), respectively. The greater Q shows the sharper image.

Figure 12 shows panoramic views of videos, which were created by AutoStitch software.[2] This software uses the multiband blending method[12] for panorama generation.

Fig. 11 Panoramic background of video 1, which is constructed using Colombari's method.[33]

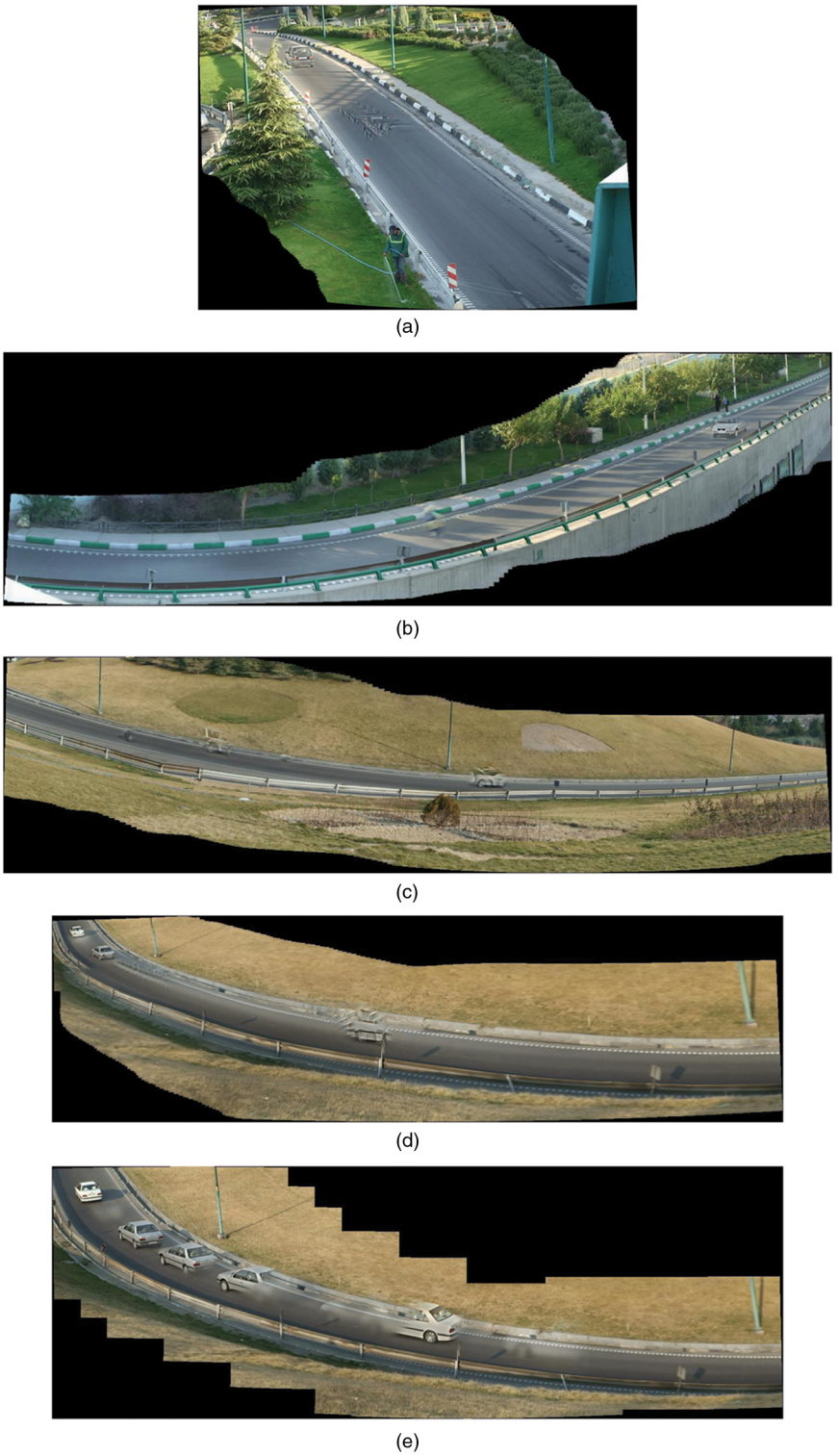

Fig. 12 Results of the AutoStitch software. (a) to (d) Panoramic views of four videos extracted from all frames of videos 1 to 4. (e) Panoramic view of video 4 extracted from key frames.

It is worth noting that AutoStitch has produced the best results among the algorithms evaluated by Refs. 36 and 37. In Fig. 12(d), AutoStitch extracts the panoramic view from all frames of video 4, but due to the large number of frames and memory limitations, only one third of the frames of videos 1 to 3 were used in (a) to (c). It can be said that AutoStitch uses multiband blending background modeling to extract panoramic background and find pixel values.

Table 7 Comparison between the runtime of the proposed method and AutoStitch (in seconds).

Video no.	1	2	3	4
Proposed method	36	70	68	53
AutoStitch	47	96	84	104

Fig. 13 Two consecutive selected frames of video 2. No relation between the images of the yellow car is detected.

As mentioned before and as can be seen in Fig. 12, this blending method can avoid the blur but still cannot completely eliminate the moving objects. Therefore, the results of our method (Fig. 10) are preferable. Only the selected key frames of video 4 are used to create the last panoramic view (e).

4.5 *Runtime*

Table 7 shows the required times for the proposed method and the AutoStitch software to create panoramic background for different videos (Figs. 10 and 12, respectively). Although the AutoStitch software was developed in C++, as seen in this table, the proposed method written in MATLAB® is faster. The experiments were performed on a laptop with an Intel core 2 duo 2.5-GHz CPU.

4.6 *Limitations*

Sometimes, it is possible that the corresponding SIFT features cannot be found between two corresponding RoDs. In this situation, RoDs are removed from frames to create a pleasing panoramic view, but the trajectory of the moving object cannot be recognized by the proposed method. Figure 13 shows this situation in video 2. The yellow car moves fast contrary to the camera motion. Although these two regions have similar color, there is no common area between them to establish a correspondence.

There is no limitation in the number of moving objects in the scene, but the proposed method does not handle the occlusion problem.

In Fig. 10(d), some parts of the moving object are seen in the panoramic background. As shown in Fig. 14, in the final frame (the green frame), some parts of the moving object are located outside the overlap area and, therefore, the algorithm is unable to remove them. Lack of information and error in the detection of the moving object's boundary make the black blob in the top-left corner of the final panoramic view of video 4 (no information for that area).

Fig. 14 Two final selected frames of video 4. Some parts of the moving object are located outside the overlap area.

5 Conclusion

We presented a method to extract motion characteristics and contours of moving objects in a wide view and to create panoramic background. In other words, we attempted to expand panorama applications from artistic views to surveillance usages. The proposed method detects positions of vehicles using RoD correspondence between each two consecutive selected frames. It uses SIFT features to find corresponding RoDs and discover positions of moving objects in other frames. These correspondences are necessary for the algorithm to track vehicles in the scene. Frame differencing is inadequate for moving objects separation. Thus, the proposed method combines various types of information to extract the exact boundary of moving objects in moving cameras. The contour extraction method is robust since it uses the information of moving objects in other frames. Finally, frame selection and running the graph-cut method in segment level make the proposed method faster.

References

1. R. Szeliski, "Image alignment and stitching: a tutorial," *Found. Trends Comput. Graph. Vis.* **2**(1), 1–104 (2006).
2. M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *Int. J. Comput. Vis.* **74**(1), 59–73 (2007).
3. A. Mills and G. Dudek, "Image stitching with dynamic elements," *Image Vis. Comput.* **27**(10), 1593–1602 (2009).
4. A. C. Murillo et al., "Localization in urban environments using a panoramic gist descriptor," *IEEE Trans. Rob.* **29**(1), 146–160 (2013).
5. Y. Xiong and K. Pulli, "Fast panorama stitching for high-quality panoramic images on mobile phones," *IEEE Trans. Consum. Electron.* **56**(2), 298–306 (2010).
6. Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.* **23**(11), 1222–1239 (2001).
7. Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(9), 1124–1137 (2004).
8. B. Fulkerson, A. Vedaldi, and S. Soatto, "Class segmentation and object localization with superpixel neighborhoods," in *Proc. of the Int. Conf. on Computer Vision*, pp. 670–677 (2009).
9. V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?," *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(2), 147–159 (2004).
10. D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.* **60**(2), 91–110 (2004).
11. K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(10), 1615–1630 (2005).
12. P. J. Burt and E. H. Adelson, "A multiresolution spline with application to image mosaics," *ACM Trans. Graph.* **2**(4), 217–236 (1983).

13. A. Levin et al., "Seamless image stitching in the gradient domain," in *Eighth European Conf. on Computer Vision*, pp. 377–389, Springer (2004).
14. A. Agarwala et al., "Interactive digital photomontage," *ACM Trans. Graph.* **23**(3), 294–302 (2004).
15. A. Agarwala et al., "Panoramic video textures," *ACM Trans. Graph.* **24**(3), 821–827 (2005).
16. N. Gracias et al., "Fast image blending using watersheds and graph cuts," *Image Vis. Comput.* **27**(5), 597–607 (2009).
17. M. Uyttendaele, A. Eden, and R. S. Szeliski, "Eliminating ghosting and exposure artifacts in image mosaics," in *Proc. of the 2001 Conf. on Computer Vision and Pattern Recognition*, pp. II:509–516, IEEE (2001).
18. Z. Zhu et al., "Fast construction of dynamic and multi-resolution 360° panoramas from video sequences," *Image Vis. Comput.* **24**(1), 13–26 (2006).
19. S. Sun et al., "Moving foreground object detection via robust SIFT trajectories," *J. Vis. Commun. Image Represent.* **24**(3), 232–243 (2013).
20. T. Lim, B. Han, and J. H. Han, "Modeling and segmentation of floating foreground and background in videos," *Pattern Recognit.* **45**(4), 1696–1706 (2012).
21. M. J. Fadaeieslam, M. Soryani, and M. Fathy, "Efficient key frames selection for panorama generation from video," *J. Electron. Imaging* **20**(2), 023015 (2011).
22. L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulation," *IEEE Trans. Pattern Anal. Mach. Intell.* **13**(6), 583–598 (1991).
23. W. Tao, H. Jin, and Y. Zhang, "Color image segmentation based on mean shift and normalized cuts," *IEEE Trans. Syst., Man, Cybern.* **37**(5), 1382–1389 (2007).
24. D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(5), 603–619 (2002).
25. C. Tao et al., "Efficient image stitching in the presence of dynamic objects and structure misalignment," *J. Signal Inf. Process.* **2**(3), 205–210 (2011).
26. O. Tuzel, F. Porikli, and P. Meer, "Region covariance: a fast descriptor for detection and classification," in *9th European Conf. on Computer Vision*, pp. 589–600, Springer (2006).
27. O. Tuzel, F. Porikli, and P. Meer, "Pedestrian detection via classification on Riemannian manifolds," *IEEE Trans. PAMI* **30**(10), 1713–1727 (2008).
28. F. Porikli, O. Tuzel, and P. Meer, "Covariance tracking using model update based on lie algebra," in *Conf. on Computer Vision and Pattern Recognition*, pp. 728–735, IEEE (2006).
29. W. Foerstner and B. Moonen, "A metric for covariance matrices," Technical Report, Department of Geodesy and Geoinformatics, Stuttgart University (1999).
30. M. Donoser and H. Bischof, "Using covariance matrices for unsupervised texture segmentation," in *19th Int. Conf. on Pattern Recognition*, pp. 1–4 (2008).
31. T. Chan and L. A. Vese, "Active contours without edges," *IEEE Trans. Image Process.* **10**(2), 266–277 (2001).
32. A. Yilmaz and M. Shah, "Object tracking: a survey," *ACM Comput. Surv.* **38**(4), 1–45 (2006).
33. A. Colombari, A. Fusiello, and V. Murino, "Segmentation and tracking of multiple video objects," *Pattern Recognit.* **40**(4), 1307–1317 (2007).
34. A. Yilmaz, X. Li, and M. Shah, "Contour-based object tracking with occlusion handling in video acquired using mobile cameras," *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(11), 1531–1536 (2004).
35. X. Zhu and P. Milanfar, "Automatic parameter selection for denoising algorithms using a no-reference measure of image content," *IEEE Trans. Image Process.* **19**(12), 3116–3132 (2010).
36. M. Brown, "Multi image matching using invariant features," PhD Thesis, University of British Columbia (2005).
37. J. Boutellier et al., "Objective evaluation of image mosaics," *Commun. Comput. Inf. Sci.* **21**, 107–117 (2009).

Mohammad Javad Fadaeieslam received his MSc and PhD degrees in artificial intelligence from Iran University of Science and Technology (IUST) in 2005 and 2013. He is currently an assistant professor in the Department of Electrical and Computer Engineering, Semnan University, Semnan, Iran. His research interests include extracting panorama image from video for traffic monitoring and other applications.

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

Mahmood Fathy received his BSc in electronics from IUST in 1985, MSc in computer architecture in 1987 from Bradford University, United Kingdom, and PhD in image processing computer architecture in 1991 from University of Manchester Institute of Science and Technology (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 Internet.