Copyright © 2015 Tech Science Press CMES, vol.107, no.2, pp.103-124, 2015

Real-Time Moving Targets Detection in Dynamic Scenes

Fan Li[1], **Yang Yang**

Abstract: The shift of the camera leads to unsteadiness of backgrounds in video sequences. The motion of camera will results in mixture of backgrounds and foregrounds motion. So it is a challenge for targets detection in dynamic scenes. A real-time moving target detection algorithm with low complexity in dynamic scenes is proposed in this paper. Sub-block based image registration is applied to remove the global motion of the video frame. Considering the blocks in one frame have different motion vectors, the global motion of each block is separately estimated. Then, a neighbor-based background modeling is applied to extract the moving objects. Moreover, combination of image registration and neighbor-based background modeling can precisely divided foregrounds from backgrounds. At last, a method, based on feature point motions, is adopted to track the foregrounds in time. The experimental results demonstrate that our method can process videos in real-time, without the effect of time delay. What is more, comparative results by quantitative evaluations manifest that the proposed approach can achieve the best classification accuracy.

Keywords: Background model; Image registration; Moving target detection; Segment; Tracking.

1 Introduction

Moving target detection is widely used in military reconnaissance and commercial tracking, which will have the widespread application prospect and research value in the future. It is the core technology in surveillance systems. In general, surveillance systems can be divided in two categories, one is static surveillance platform and the other is dynamic surveillance platform. In static platform, the background is static and only targets moves in a frame. So it is easy to extract moving targets in static platform. However, in dynamic platform, both targets and camera are moving, the

[1] The Ministry of Education Key Laboratory for Intelligent Networks and Network Security, School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, P.R. China, Email: lifan@mail.xjtu.edu.cn

motion of targets and camera are mixed together. It is indeed a challenge to extract targets in dynamic platform. Attention has been paid to moving targets detection in unmanned aerial vehicle (UAV) videos in recent years [Teutsch and Kruger (2012); Yang et al. (2012)].

Existed algorithms for moving targets detection in dynamic scenes can be categorized into two groups: optical flow approach [Patel and Parmar (2014); Frakes et al. (2013)] and modeling-based approach [Varma and Sreeraj (2014); Zhang and Zhou (2010); Barnich et al. (2011); Kim et al. (2010)]. Optical flow approach computes the optical flow between two adjacent frames in order to get motion information for each pixel. Motion vectors of moving objects are different to the vectors of neighbor backgrounds. It is superior in accuracy. However it suffers from the drawback that large quantities of calculations are required. As a result, it cannot satisfy those systems with the requirement of real-time. Modeling-based approach performs one or more models to represent the background for each pixel. S. Varma and M. Sreeraj (2014) proposed a codebook incorporating spatial-temporal context of each pixel for modeling. Each pixel is represented by a codebook and each codebook is composed of code words. Codebooks are able to capture background motion over a long period time. So it should learn from a long training sequence at first. Gaussian Mixed Model (GMM) [Zhang and Zhou (2010)] works better at solving the problems of adaptability and computing expensiveness. However, it always leads to misclassification when the camera moves seriously. It can apply ideal results to those videos captured by slow moving cameras. In general, the test videos are all in superior conditions. Those cameras are always moving gently (translational movement with an overwhelmingly low speed). So those test videos, which are captured by dramatically shaking camera with irregular movements, challenge the robustness of existed algorithms.

Considering the cameras with violent motions influence detection results seriously, an ocean of researchers engaged in solving the problem. Image registration [Tavares (2014); Alves and Tavares (2015); Bastos and Tavares (2010); Oliveira and Tavares (2014)], setting two frames with different backgrounds in a same coordinate, is performed before detection. The global-motion of the background should be eliminated so that the remaining motions, which are due to moving objects, can be detected. Ibrahim et al. (2010) use Scale-invariant feature transform features and RANSAC to gain the points in backgrounds. The transform model is generated using the extracted match points. T Moving objects are detected by dynamic background noise removal technique. Cheraghi and Sheikh (2012) using the Shi & Tomasi corner detector, the corners are detected and the camera motion has to be estimated. After camera motion compensation, adaptive background subtraction is applied for detecting and extracting the moving objects. Walha et al. (2014) extract the local features (Scale Invariant Feature Transform) in two frames to estimate global motion and construct a reliable background before segmentation. Then they detect moving objects by Kalman filtering. The result shows that noises are decreased dramatically and foregrounds are separated accurately. However, it cannot balance the contradiction between processing speed and robustness well and also the foreground extraction is strictly sensitive to motion compensation.

The image registration is the core technology of the moving object detection. The prevailing image registration methods assume that all the features they have extracted are coplanar and then they build a transform matrix to perform registration. The assumption is not in reality in certain situations. There are always large depth variations exist in 3D. When they are projected to the 2D space, high-rise objects (the street lamps or tall buildings) always move faster than other objects in image planar, as we show in Fig. 1. Therefore, it is a challenge to find a unified transform matrix for the whole image to describe the global motion of the video frame. Moreover the detection accuracy depends strongly on the fidelity of image registration. Although subsequent processes have adopted for better detection results, those processes cannot decrease yet the influence of position errors caused by image registration.

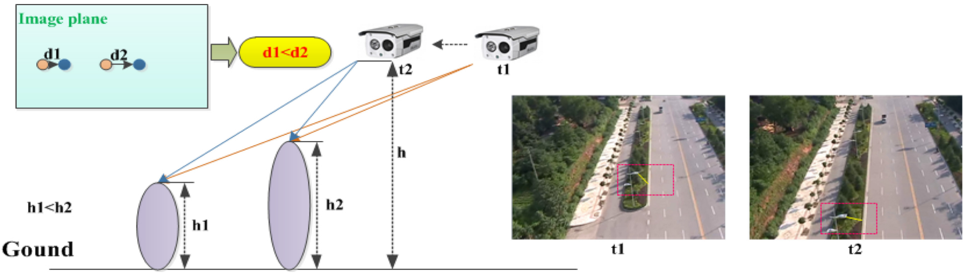


Figure 1: Two objects with different heights move in different speed in the image planes.

In our work, we propose a real-time moving target detection algorithm with low complexity in dynamic scenes. The main contribution of this paper can be summarized as follows:

1.1 Sub-block based image registration

There are two important factors which influence the accuracy of image registration. Firstly, the background in various regions in the video frame will be different which is caused by the changing of UAV flight positions and shoot angles. Secondly, when backgrounds with different heights in 3D project to 2D planar scene, it also causes different background motions in the various regions. Therefore, it is not accurate to represent global motion of a video frame by only one transform model. We propose a method to segment the video frame into several blocks and each block obtains a transform model independently. Block segmentation decreases the number of different height objects in each block, therefore decrease the effect of heights while image registration. Hence, different blocks with different transform models can represent motion of each block more precise.

1.2 Combination of image registration and neighbor-based background modeling

A transform model in image registration cannot exactly estimate motion for each pixel. It causes that there are certain deviations between models transformed pixel locations and the real locations. We propose a neighbor-based background modeling after image registration to recognize moving targets. For a certain pixel, we put its pixel value and the neighborhood pixel values as background samples in the reference frame. When a certain pixel after image registration compared with its background samples, the pixel also can be matched to a sample which is formed by neighbor information. So we propose a method by combining global motion and neighbor-based background model to overcome the error effect of transform models and precisely divide foregrounds from backgrounds.

1.3 Low computational complexity

Moving object detection is mainly applied to surveillances which have critical real-time requirement. So moving targets detection systems only with real-time property can satisfy our demands in reality. Our proposed algorithm has the low computational complexity throughout the processes. In image registration, we adopt pyramid optical flow to gain the exact positions of matching feature points while image registration. Pyramid optical flow estimates the rough matched point positions in the top level of pyramid (smallest resolution) image. Then it searches the exact point positions within the rough position-centered windows from up to down in the pyramid until the bottom level. Pyramid optical flow helps a lot in narrowing the searching field while refining the exact feature point positions and it reduces the searching times even better. In overall system, once optimal foregrounds are recognized by registration and background modeling, the two steps are no longer needed for targets detection. We just need to extract the feature points in the foregrounds and then track the foregrounds by the motion of related points. By this way, we shorten detection time a lot.

The rest of this paper is structured as follows. In Section II, we provide an overview of our system architecture. Efficient implementations of our approach are described by numerical and comparison results in Section III. This paper concludes with Section IV.

2 Our proposed method

In this work, we proposed a real-time moving object detection method in dynamic scenes. The system design is illustrated in Fig. 2 as with three steps: segmentation-based image registration, background modeling and tracking. We segment each video frame into several blocks. Each block is registered independently. For each block, we extract the matched feature points in two consecutive frames and the points are used to estimate the transform model for the block. When the global motion of each block is compensated by the transform model, the frames can be registered in the same background. Then background modeling is adopted to set foregrounds and backgrounds apart. Once the optimal result is realized after detection for several frames, a method of tracking, according to the motion of related points, is employed to track and show the foregrounds.

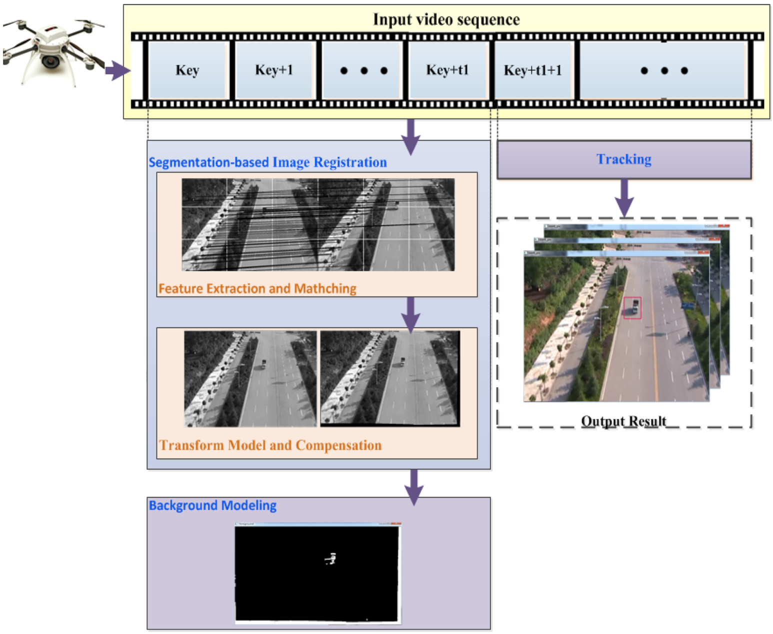

Input video sequence
Key
Key+1
• • •
Key+t1
Key+t1+1
• • •
Segmentation-based Image Registration
Feature Extraction and Mathching
Transform Model and Compensation
Background Modeling
Tracking
Output Result

Figure 2: Module diagram for system.

2.1 Blocks segmentation-based image registration

In order to achieve the best result with low error, the camera motion has to be estimated firstly. Classical methods of image registration are always with the steps:

Points Extraction and Matching and *Transform Model and Compensation*. We use segmentation to decrease the error effect of traditional method and increase the accuracy.

2.1.1 Blocks segmentation

As we have mentioned before, the changing of UAV flight positions and different heights in 3D project to 2D planar scene will both cause different background motions of the whole image. Knowledge of 3D geometric segmentation [Han et al. (2013); Jia et al. (2012)] can estimate the height of each point, but it is not worth to do so. Firstly, if UAV flights in a relative high altitude, high-rises would have a little impact and the effects caused by those high-rises can be ignored even. Secondly these methods always use some complicated geometric information and they consume lots of time (about two frames in a second) which makes it impossible for real-time detection. Here we will show that the segmentation can reduce the height effect. Even though we cannot avoid the height effect completely, but we can decrease it greatly.

The justification of segmentation incorporates two important components: a) processing in each block ensures there are fewer objects of different heights in a block. b) if there are noises or only one high-rise object somewhere, segmentation avoids the noises (or high-rise objects) to affect the whole image.

2.1.1.1 Ensure less objects of different heights

The traditional image registration algorithms suffer from the height problem due to their underlying assumption that the scene can be regarded approximately planar. Under the precondition of this assumption, the law of each feature point motion is familiar. Then a uniform transform can express their motion commendably and the motion of the camera can be shown at the same time.

As camera moves, high-rise objects move faster than planar objects in image plane. High-rises have the same properties with moving objects, moving faster than planer objects. So high-rises are easy to be detected as foregrounds. If there are various heights of objects in a scene, it is impossible to find a transform model for various speeds and that will make a large overall error.

We propose a method to decrease the height effect. The segmentation is demonstrated in Fig. 3. As we can see in the certain frame, there are about three different heights in the frame totally, the heights of trees, lamps and a road. So there are three motion rules actually. If a video is captured in urban areas, the background is more complicated. Therefore there will be more motion rules in an image.

In order to make sure that there is only one height object in each block, video frame

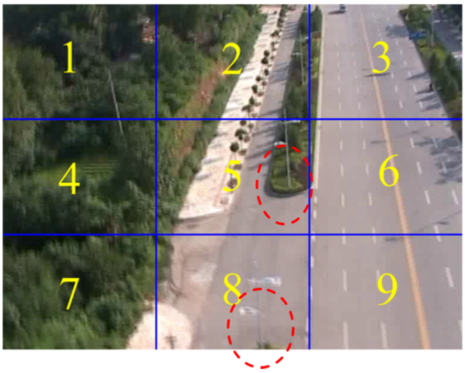


Figure 3: Blocks segmentation

is segmented as blocks with the same and suitable size. When UAVs flight is in low altitudes, there will be not too many objects in an image essentially. So after blocks segmentation, it can basically guarantee there is only one height in each block. And there will be no more than two kinds of heights in each block on the worst. Although there might be more than one object of different heights, objects in the same height may take over the most portions. So most pixels have the same motion property, we can represent the primary motion.

The segmentation helps a lot to decrease the probability of including various heights of objects in each block. The transform model independent for each block can represent motion more precise.

2.1.1.2 Avoiding Impact on the Whole Image

We have shown that segmentation can decrease the diversity of heights in pre-subsection. But what if there is just few distinct high-rise? As we can see in Fig 3, there is about just one height in each block. But in blocks 5 and 8, there are two kinds of heights, a lamp and a road. Without blocks segmentation, those feature points on the lamp ought to be used for the global motion. We will get a global motion that smoothes the different motion rules. And it leads to that global motion error for the whole image is rather large. Those points of high-rise influence the whole image seriously. After blocks segmentation, blocks without the high-rise conform to the assumption of planar. And other areas are never affected by the high-rises. Transform models of most blocks are performed by the feature points in approximately planar scenes. So taken as a whole, the performance will improves largely.

We cannot eliminate the effect of high-rise objects completely through segmentation. And we cannot eliminate the effect of plane oscillation as well. But the segmentation reduces the influences without consuming too much time.

2.1.2 Feature points extraction and matching

Feature point extraction is devoted to identifying characteristic points or interest points. The *Harris* et al. (1998), as the most widespread feature point, relies on a central principle: at a corner, the image intensity will change largely in both horizontal and vertical directions. The Harris detector uses the second moment matrix as the basis of its corner decisions; approximating the eigenvalues of the matrix and comparing it to a predefined threshold to determine the existence of corners.

Feature point matching is adopted to find the corresponding corners between images. The matched corners are tracked by *pyramid optical flow* method in Bouguet (2000). The algorithm makes use of spatial intensity information to search for the position that yields the best match.

The basic idea of *optical flow* is to set:

$$\sum_{x=p_x-w_x}^{p_x+w_x} \sum_{y=p_y-w_y}^{p_y+w_y} I_{t-1}(x,y) = \sum_{x=p_x-w_x}^{p_x+w_x} \sum_{y=p_y-w_y}^{p_y+w_y} I_t(x+d_x, y+d_y). \tag{1}$$

where $I_{t-1}(p_x, p_y)$ means the pixel value of point (p_x, p_y) at frame $t-1$. $d = (dx, dy)$ means the displacement of matched points between frame $t-1$ and frame t. w_x and w_y are the window size. The function is measured on an image neighborhood of size $(2w_x+1, 2w_y+1)$.

In *pyramid optical flow*, a group of pyramid images have to be built at first by down sampling; resolution increases from the highest to the lowest level. *Pyramid optical flow* estimates the rough location of matched point in the highest level (L) by (1). Then the location delivers to the next layer as:

$$p_x^{L-1} = 2\left(p_x^L + d_x^L\right). \tag{2}$$

$$p_y^{L-1} = 2\left(p_y^L + d_y^L\right). \tag{3}$$

The next level optical flow residual vector d^{L-1} is then computed through (1). And a more precise location is estimated by optical flow. Continue this process until figure out the location at the lowest level.

A pyramid implementation makes it possible to estimate much larger image velocities than a one-level implementation and it is faster than traditional techniques for examining potential matches.

2.1.3 Transform model and compensation

There are 2 most-used transform models: affine model (with 6 parameters) and perspective model (with 8 parameters). In general, perspective model is a more advanced model. But time spending on perspective model computation is about two times longer than affine model computation. So considering the computational intensive, perspective model is not suitable for real-time applications and we used affine model here.

Affine model helps to map the coordinate relationship between two consecutive images. An affine model transformation, with 6 unknown parameters (a,b,c,d,e,f), is represented as:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & e \\ c & d & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{4}$$

where $[x,y]$ is the coordinate of a corner point at frame $t-1$ and $[x',y']$ is the coordinate of a matched corner point at frame t. The matched points have been obtained in *points extraction* and *matching*.

We use RANSAC [Mikolajczyk and Schmid (2002)], an iterative method, to filtering outlier sets (feature point sets in foregrounds) and retain the inlier sets (feature point sets in backgrounds). All of the inlier sets are adopted to compute parameters of the affine model through (4).

The background motion between two consecutive frames (frame $t-1$ and frame t) is modeled by the affine model. So once parameters of the affine matrix are obtained, we can estimate the apparent global motion between the consecutive images.

Image registration above is dealt between two consecutive frames. Current frame should be registered with the reference frame. We express the global motion from frame p to frame q as H_{pq}. The transformation from frame k to n is defined by:

$$H_{kn} = \prod_{i=k}^{n-1} H_{i(i+1)} \tag{5}$$

So the affine matrix of the current frame should be obtained by multiplying the affine matrixes from the reference frame to the current frame.

Up to now, a current frame can be compensated to a frame that has the same background in reference frame by the affine matrixes.

But there will cause a problem. There are only 6 parameters in the affine matrix, but there are much more than 6 feature point sets for affine matrix computation. So what we solve by (4) is an overdetermined equation. As we all know that overdetermined equation is an inconsistent equation and we cannot find an exact solution to the equation. But we can find a proximate solution. So while solving the overdetermined linear equations, there must be some deviation in the affine matrix. Also we cannot ensure that each region is planar. So there is no doubt that all the points are not in a completely same motion rule. A transform model in image registration cannot exactly estimate motion for each pixel. Background modeling will solve the problem. So the deviation does not matter here and we will discuss it in details later.

2.2 Background modeling

It is able to recognize foregrounds by frame difference if the condition of image registration is extremely ideal. But that was not the case. Firstly, it is impossible to eliminate those errors that caused by heights in 3D space completely as we have mentioned in Sec A. Secondly, there are just few parameters in the transform model, but there are lots of feature points for transform matrix computation and the transform model is expected for satisfying all of the feature points as we have shown in the prior sub-section. So there must be some inevitable deviations after image registration. A host of general background models have proposed to solve the problem. Most background models assume that pixel value of a certain pixel is subject to some distributions in time domain. The assumption only applies if image registration is further ideal. But it is not necessarily true. Pixel values of a certain position cannot obey distributions strictly. We will demonstrate the deviations in image registration in quantity now.

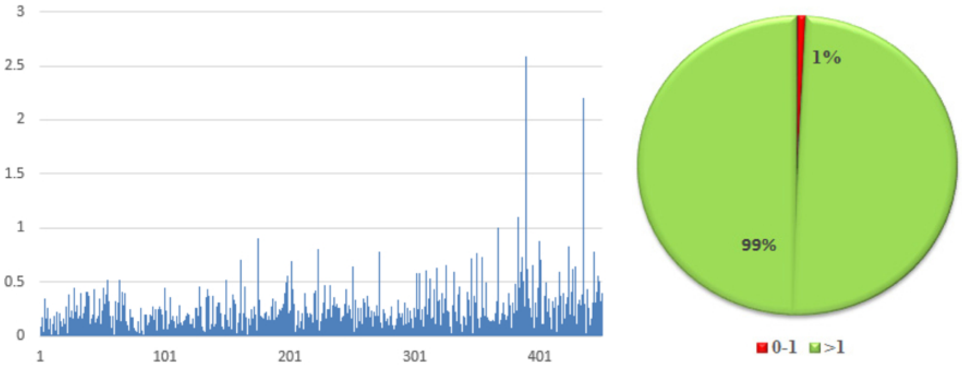


Figure 4: the left figure shows distance between sets *TransP*$_t$ and P_t. The right figure shows value distribution of distance between sets *TransP*$_t$ and P_t.

We represent a feature point extracted by *Harris* at frame $t-1$ as P_{t-1} and represent the corresponding feature point extracted by *pyramid optical flow* at frame t as P_t. The point P_{t-1} can be transformed to a position ($TransP_t$) by the transform model. As we know, distance between P_t and $TransP_t$ should be 0 in ideal conditions. We compare the distance between $TransP_t$ and P_t. We show the results in Fig. 4. We can realize that there are only 1% pixels exceed one pixel deviation by the transform model. That is to say, almost all (nearly 99%) transformed pixels are closed to their real positions.

We know that the transform relation between the current frame and the reference frame is the same as relation between the current image and its steady image:

$$I_{Ref}(p_x, p_y) * H_{Reft} = I_t(p_x, p_y). \tag{6}$$

$$I_{std}(x, y) * H_{Reft} = I_t(x, y). \tag{7}$$

So that is to say, when current frame is registered with the reference frame, almost all pixels can be transformed to the positions within one pixel deviation in steady image, except 1% pixels are not transformed to their right positions. Thus this motivates us to adopt the background model with neighbor information for separating foregrounds and backgrounds. Therefore the true background information will be included in the background models.

We describe the main idea of the neighbor-based background modeling [Barnich et al. (2011)] briefly:

2.2.1 *Initialization of model*

At the reference frame, we populate the pixel models $(M(p_x, p_y))$ with values (v_i) found in the spatial neighborhood of each pixel (p_x, p_y):

$$M(p_x, p_y) = \{v_1, v_2 \dots v_N\} \tag{8}$$

The size of the neighborhood needs to be chosen so that it is large enough to comprise the pixel value of real background.

2.2.2 *Classifying*

When a new frame comes, kind of each pixel can be estimated. We compare the current pixel value $(I_t(p_x, p_y))$ to the background samples $(M(p_x, p_y))$. Classifying the current pixel as a background when at least there are # pixel values in the background samples are close to the corresponding pixel value in the current frame. Otherwise we classify the current pixel as a foreground.

2.2.3 *Updating of model*

If the pixel is found to be as a background, we randomly replace one of the samples by the current pixel value. If the pixel is found to be as a foreground, there is nothing has to be changed for the background samples. So that foreground information will be never included in the background samples.

2.3 Tracking

Once the optimal targets have been located for several frames, the object tracking component will help to keep the target's position in time.

Kalman filtering is widely used in tracking. The position of a moving target in the future can be estimated by the Kalman filtering. Sometimes targets may be temporarily occluded, but sometimes targets may even be disappeared definitively. In [Pinho et al. (2005); Pinho and Tavares (2009); Tavares and Padilha (1995)], management model-based methods combined Kalman filtering and other ways, using a confidence value to each tracked feature, to handle the occlusion. The Kalman filtering estimated the position and the confidence value determine whether the target should be tracked or not. In [Pinho et al. (2007); Pinho et al., (2006)], the proposed Net Present Value model, based on the economic Theory of Capital, has been applied. The occluded feature may be kept on tracking or it may be excluded of the tracking process depending on its historical behavior.

The former tracking algorithms we have talked before are always focused on the occlusion problem in long image sequences. But it is not necessary to focus on the problem in our method. Firstly, the UAV flies rather high, the odd of occlusion is quite low in those video sequences essentially. What is more, tracking is just processing in a few frames in our method (we just choose 3 frames for tracking). So the odds of occlusion are also rather low during just few frames. Secondly, we order 25 frames for a period (we will show it in the second paragraph of Sec. III). That is to say, we detect the moving targets in each 25 frames. So if a target is occluded or a new target appears in the scene, we can recognize it by the background model. Hence we do not need to consider much about occlusion in tracking.

As we mentioned above, feature point is less on quantity, while more on information. It is able to enhance the rate of computation and makes real-time schemes possible. We have represented the whole image by feature points while image registration for time saving. So now we represented the foregrounds motion by feature points too.

For each foreground in previous frame, we extract its corresponding feature points using *Harris* extractor. Then PLK is applied to find the matched feature points in the current frame. We can obtain the motion vector (Mv_{xi}, Mv_{yi}) of point i. Assume that there are K corners in each foreground. Motion of these feature points are almost the same as the foreground. But feature points extraction and matching cannot be entirely accurate. So we represent the motion of a foreground by the average motion of its corresponding points. Foreground motion (FMv_x, FMv_y) will be estimated by:

$$FMv_x = \sum_{i=0}^{K} Mv_{xi} \Big/ K. \tag{9}$$

$$FMv_y = \sum_{i=0}^{K} Mv_{yi} \Big/ K. \tag{10}$$

It is fast to calculate the average movements of all K corners in each direction (x or y direction). In contrast with image registration and background modeling, the tracking cancels some time-consuming processes and it can save much time.

3 Experimental results and analysis

We have complemented the algorithm in C++ with the open-source OpenCV. All the experiments work on PC, 3.40 GHz CPU and 4.0 GB RAM, with Windows 7 as an operating system. In this section, the effectiveness of the proposed method for separating foreground from the background is demonstrated for a variety of videos, *highwayI*, *highwayII* and *urban*, all of them are acquired with freely moving cameras. All of the experiments are tested using the same parameters.

For our test videos, there are 25 frames per second. So we order 25 frames for a period. We just deal with few frames in a period. The same processes we did in each period. We call the first frame in a period as the reference frame. The reference frame should be altered periodically. Image registration and background modeling are processed for *DetNum* frames to get the optimal foregrounds. Then tracking is adopted only for $TraNum$ frames.

3.1 Metric of performance

Precision, recall and *accuracy* (*acc*) are applied for the proposed algorithm. They are defined as follows:

$$precision = N_{FF} / \left(N_{FF} + N_{BF}\right). \tag{11}$$

$$recall = N_{FF} / \left(N_{FF} + N_{FB}\right). \tag{12}$$

$$acc = 0.5 * precision + 0.5 * recall. \tag{13}$$

where F and B denote the foreground and background respectively. N_{FF} is the number of marks that circle the real moving targets. N_{BF} is the number of marks that circle background as moving targets by mistake. N_{FB} is the number of real moving targets that have been detected as background wrongly. In order to minimize errors, the *precision*, *racall* and *acc* percentages needs to be as high as possible.

3.2 Determination of our parameters

From previous discussions, it appears that our method has the following parameters:

- Size of background samples for each pixel (S)
- Frame number for image registration and background modeling ($DetNum$)
- Frame number for tracking ($TraNum$)

In our experience, the values of background model size (S) and frame numbers ($DetNum$ and $TraNum$) factors are important to excellent results.

To determine an optimal value for S, we compute the distance and distribution of $TransP_t$ and P_t, that we have done and shown in Fig. 4. There are about 99% pixels are within one pixel deviation to the real location. So in order to eliminate the one pixel deviation, we select samples in the 4-connected neighborhood ($S = 5$) of each pixel, which can make sure that the real pixel value of background is in the background samples. The deviation of some pixels' location is larger than one pixel and the 4-connected samples cannot include the real background information. It may motivate us to selected samples for each pixel in a larger neighborhood. But it is not an option. Although a larger neighborhood do helps us to set background samples with its real information, there will be a great increase of samples for the whole image if the number of samples for each pixel increased. For example, if we select the 8-connected pixels as the samples of a pixel ($S = 9$), the time of background modeling for each pixel will be twice as big as the 4-connected strategy. The time spends on background modeling for a video frame doubled as well. So for a video frame, the time will increase a lot and it is quite negative for real-time systems. In addition, as we can see in Fig. 4, there are just 1% pixel background models are not exact, and it is not desirable to trade plenty of time for the accuracy of just 1% pixels. So $S = 5$ has proved to be satisfactory.

In order to confirm an optimal value for *DetNum*, we compute the evolution of *precision*, *recall* and *acc* with all the test videos and show the percentage of *precision*, *recall* and *acc* in Fig. 5. In "*highwayI*", the superior *precision* first appears at frame 3. However the superior *recall* appears at frame 4. But when we consider *precision* and *recall*, the superior *acc* first appears at frame 4 and it tends to saturate for values higher than 4. Large frame number leads to higher computation cost. We can see the same condition in other test videos. Although the superior *precision*s or *recall*s may not be first appears at frame 4, but the third row in Fig. 5 shows us that the superior *acc* always first appear at frame 4. So we have reason to select *DetNum* as 4.

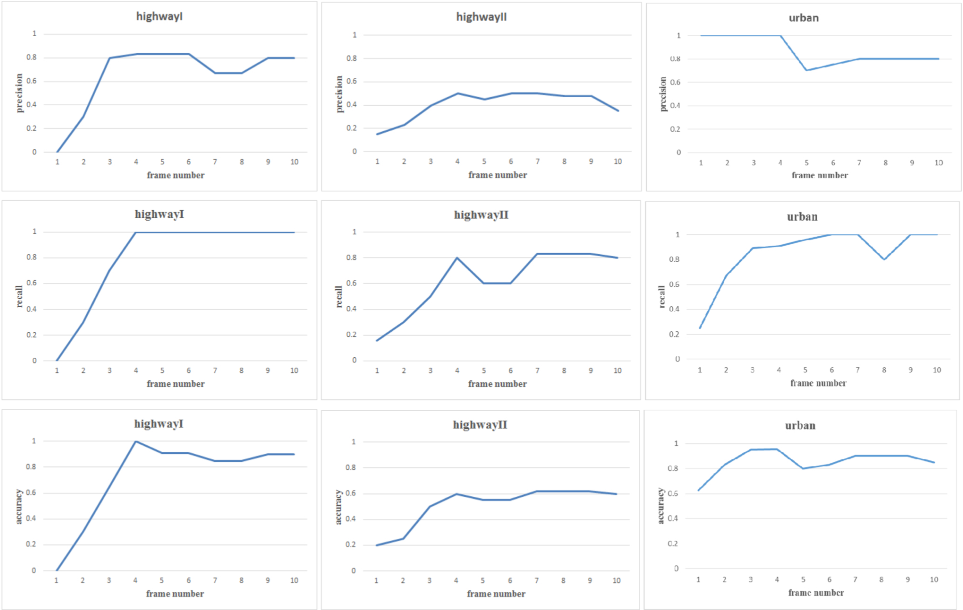


Figure 5: First row shows precisions of three test videos; second row shows recalls of three test videos; third row shows accurucys of three test videos.

Frame number $(TraNum)$ for tracking is based on the response time of human eyes and frame rate. Response time of human eyes is 0.1s and the time of showing one frame is 0.04s (25 fps). If mark of a target is shown in one frame only, the mark will not be seen by our eyes and we consider the mark never appeared any more. When the mark shows for two successive times, there are just 0.08s, we cannot realize the targets too. Only the mark is show in three or more successive times, marking of moving objects can be realized since the show time is longer than eye response time. So in order to realize marks of targets with less time, we just need to track the mark of foreground for 3 frames, that is $TraNum = 3$.

3.3 Performance evaluation

3.3.1 Evaluation of blocks segmentation

In order to achieve the objective analysis of segmentation in image registration, we use Peak Signal to Noise Ratio (*PSNR)* to evaluate the quality. *PSNR* is an objective criteria to evaluate image quality. A higher value is better. Since there are various heights in sequence "*highwayII*" and the camera moves drastically and irregularly, so the evaluation of blocks segmentation is tested on the video. We show the *PSNR*s in two cases, registration with and without blocks segmentation as in Table 1. We record *PSNR*s in each block and the whole image for three certain frames of the sequence. Here an image with 768*576 is segmented into 9 blocks of same size.

With segmentation, we can see that the *PSNR* is different in each block. On average, the values in blocks 5 and 8 are always smaller than others. Because there are high-rise lamps in the blocks, as the frame we see in Fig. 3. The lamps impact registration of the related block. However *PSNR*s of other blocks are rather high, that is to say the high-rises never impact blocks without them. Also, for the purpose of comparing with the method without segmentation, we compute the *P-SNR* of whole image. As we can see *PSNR* with segmentation increases 0.1316dB, 0.6399dB, 0.2348dB independently compared with method without segmentation. This means for the whole, blocks segmentation helps a lot to image registration from an objective perspective aspect.

Table 1: *PSNR* of each block and the whole image with and without segmentation

PSNR \ Situations Frame time	Blocks segmentation (dB)										Without Segmentation (dB)
	1	2	3	4	5	6	7	8	9	Whole Image	Whole Image
t	33.9801	27.4119	29.8701	31.2288	**28.0024**	36.9913	28.7356	**28.7356**	36.6695	**30.0017**	**29.8701**
$t+1$	33.8171	29.0459	34.3286	30.4965	**27.4862**	37.3389	30.0690	**28.6369**	36.3698	**30.1374**	**29.4975**
$t+2$	27.9604	31.3183	32.6901	30.8068	**28.8880**	36.6695	28.4459	**29.4384**	34.3286	**30.8068**	**30.5720**

3.3.2 Evaluation of neighbor-based background modeling

In this section, we will testify the effectiveness for the combination of the background modeling and image registration. In the same condition of image registration, we compare our neighbor-based model with other classical methods as Background Difference method (BD), Single-Gaussian background model (SG) and Gaussian Mixure Models (GMM) [Zhang and Zhou (2010)]. Fig. 6 demonstrates example of foreground detection for one typical frame of three sequences. Foreground and background pixels are shown in white and black respectively. The moving object are quite small as we shown, in order to show them obviously we show them in the original frames with boxes and show the true foregrounds detection with boxes as well.

Obviously, the BD method has the worst detection results. It recognizes a plenty of backgrounds as foregrounds falsely. Meanwhile, moving objects may be lost if they move so slowly as show in sequence "*urban*". Combination with SG and GMM models are better than BD. In *highwayI*, SG and GMM can recognize the moving car with few errors. But in sequence *highwayII* and *urban*, the background is more complicated, there are still lots of false detections. GMM can realize all the moving objects, this can have a high value of *precision*, but the *recall* should be

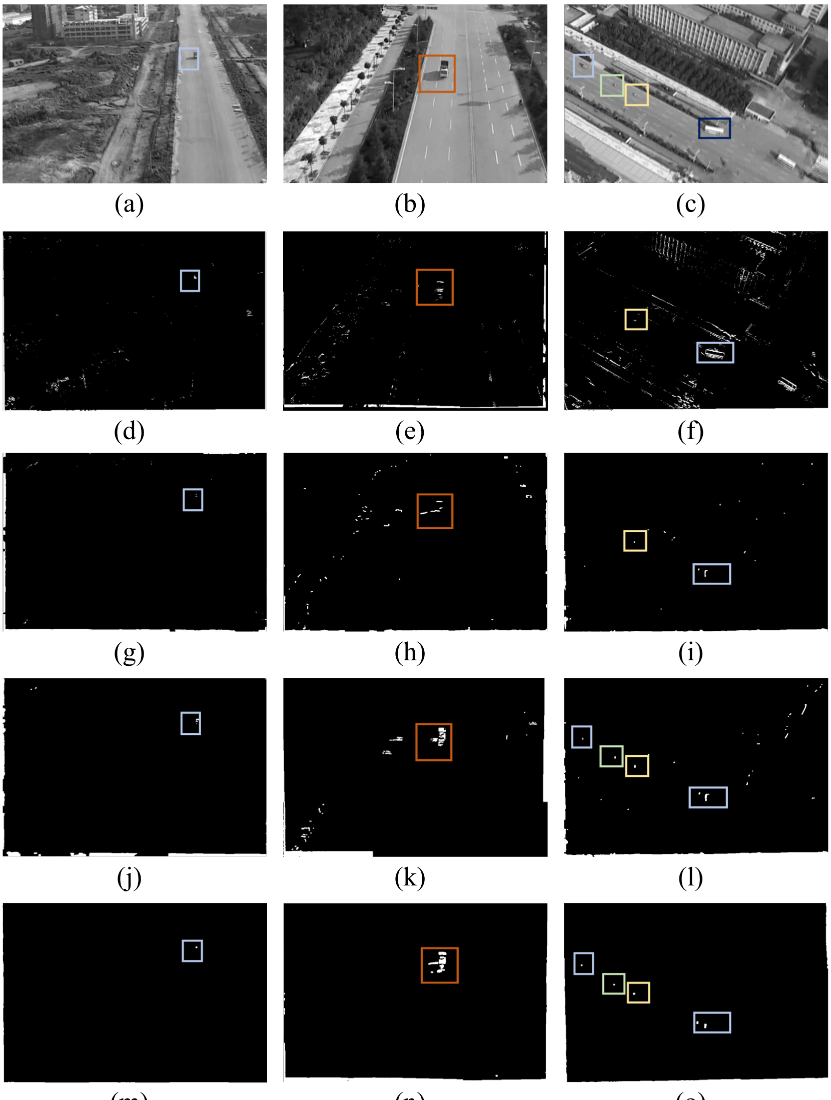
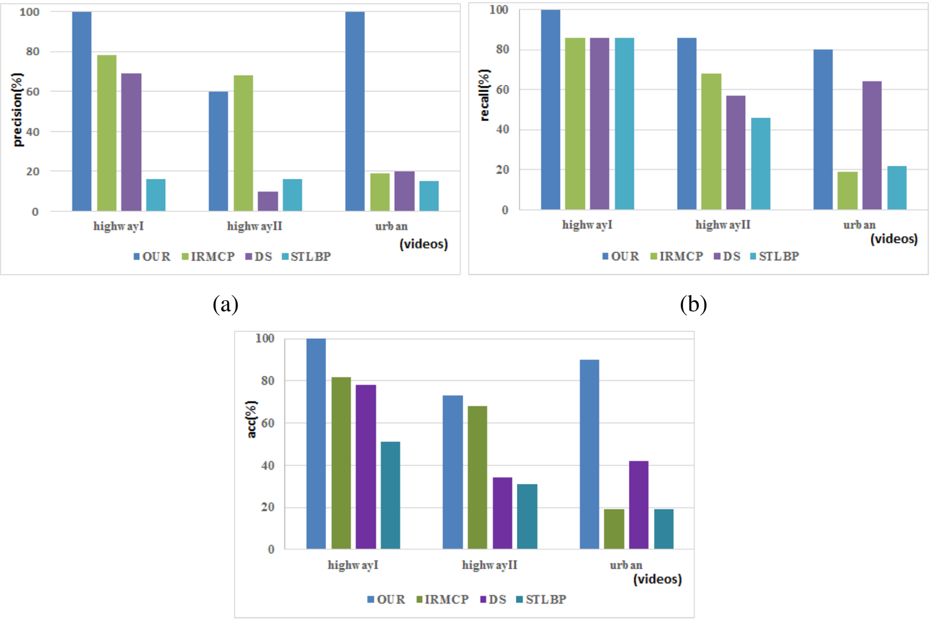


Figure 6: Backgrounds and foregrounds segmentation of test videos. **a,b** and **c** are original frames of highwayI, highwayII and urban with real moving objects representation, **d,e** and **f** are foregrounds of highwayI, highwayII and urban by **BD, g,h** and **i** are foregrounds of highwayI, highwayII and urban by **SG**. **j,k** and **l** are foregrounds of highwayI, highwayII and urban by **GMM**. **m,n** and **o** are foregrounds of highwayI, highwayII and urban by **our** method.

rather low because it recognizes lots of background areas as foregrounds, especially somewhere at the buildings or trees. But our method can recognize the buildings or trees as backgrounds. Meanwhile, in terms of detection potion of a target, our combination can detection the largest portion of the moving object; more portion than other methods, except BD. However, it is so low in all the evaluation criterions.

Sequence "*highwayII*" and "*urban*" with more complicated conditions, violent vibration and various heights, proves our method can increases the detection results a lot. The moving objects can be separated from backgrounds with far less wrong detections, even if those targets are very small. Our combination can increase the detection results a lot.



(a)

(b)

(c)

Figure 7: Comparative results of *precision, reacall and acc*.

3.3.3 *Evaluation of overall system*

We will demonstrate the evaluation of the overall system with objective criterion in Fig. 8.

Four methods of detection in dynamic scenes are adopted to compare with our method. 1) Detection with Image Registration for a Moving Camera Platform [Cheraghi and Sheikh (2012)] (referred as IRMCP), 2) Detection with Scatterness

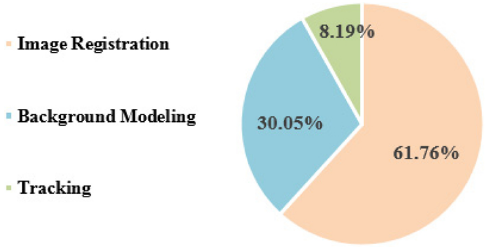


Figure 8: Time distribution for image registration, background modeling and tracking.

[Kim et al. (2010)] (referred as DS), 3) detection with Spatial-Temporal Local Binary Patterns [zhang et al. (2008)] (referred as STLBP).

The *precision*, *recall* and *acc* are employed to verify the performance of the proposed method. For "*highwayI*", all of the methods can achieve better results in both *precision* and *recall*, except STLBP. Because of the simple highway background, low resolution and small size of objects, the LBP feature is not obvious. So many backgrounds are recognized as foregrounds, that leads to the precision is rather low. In "*urban*", with the background environment more complex, we see that our method clearly outperforms the other techniques: its *precision*s and *recall*s are the highest, and the *acc*s are about at least more than 2 times higher than other methods. For the most high-rises influenced sequence "*highwayII*", *acc* of our method decreases; comparing with other two test videos. Even so, our method is greater than others.

Through test sequences with different characteristics, the results demonstrate our method is the most robust in both precision and recall and our methods to be superior.

Note that, if those moving targets move less than 0.5 pixels between two continuous frames, we cannot recognize it by our eyes, so we will regard them as backgrounds.

3.4 Computation complexity

Fig. 8 demonstrates the relative computation time for each main component of the proposed moving object detection system. As we can see, the image registration takes 61.76% of the overall computation time in the architecture as we compute a transform model for each block. Totally, there are about 91.81% time for registration and background modeling and there are just 8.19% time spending on tracking. That is a great improvement to real-time detection. So once we recognize the best detection results, tracking is applied. The system with tracking helps a lot for decrease of computation complexity.

Table 2: Computation complexity

Operation / Resolution	Image Registration (s)	Background Model (s)	Tracking (s)	Total (s)
320*192	0.105	0.026	0.009	0.140
640*480	0.207	0.103	0.026	0.336
768*576	0.294	0.143	0.039	0.476

Table 2 demonstrates the total time of each main step with different resolutions in a period. For image 768*576, time spends on tracking is just about 0.039s. But the time spends on image registration and background modeling is about 0.437s. The tracking economizes about 0.398s for detection, decreasing about 91%. With the increase of resolution, time spends on detection increases. But the total time is less than 1s in all of the test videos, which is the strongest evidence for our real-time process.

4 Conclusion

In this paper, we proposed a novel algorithm for real-time moving targets detection in dynamic scenes, especially for UAV. The algorithm adopts image registration at first. While image registration, we segment the whole image into several blocks, that decreases the effect of depth and increases the accuracy of registration by allocating each block with an independent transform model. After image registration, a neighbor-background model is adopted to set background and foreground apart. The combination of image registration and neighbor-based background models can reduce the error influence caused by image registration and increase the detection results a lot. Moreover, we also introduce a simple tracking method to reduce the detection complexity.

References

Teutsch, M.; Kruger, W. (2012): Detection, Segmentation, and Tracking of Moving Objects in UAV Videos. *2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, pp. 313-318.

Yang, Y.; Liu, F., Wang, P.; Luo, P.; Liu, X. (2012): Vehicle detection methods from an unmanned aerial vehicle platform. *2012 IEEE International Conference on Vehicular Electronics and Safety*, pp. 411-415.

Patel, M. P.; Parmar, S.K. (2014): Moving Object Detection with Moving Background using Optic Flow. *IEEE International Conference on Recent Advances and Innovations in Engineering*, pp. 1-6.

Frakes, D.; Zwart, C.; Singhose, W. (2013): Extracting motion data from video using optical flow with physically-based constraints. *International Journal of Control, Automation and Systems*. vol. 11, no. 1, pp. 48-57.

Varma. S.; Sreeraj, M. (2014): Hybrid Background Subtraction in video using Bi-level CodeBook model. *2014 Fifth International Conference on the Applications of Digital Information and Web Technologies*. pp. 124-130.

Zhang, X.; Zhou, J. (2010): Moving Target Detection in Complex Scenes Based on Spatial-Temporal Domain Analysis. *2010 3rd International Congress on Image and Signal Processing*, pp. 1520-523.

Barnich, O.; Van Droogenbroeck, M. (2011): ViBe: a universal background subtraction algorithm for video sequences. *IEEE Transaction on Image Process*. vol. 20, no. 6, pp. 1709-1724.

Kim, J.; Ye,G.; Kim, D. (2010): Moving Object Detection under Free-Moving Camera. *2010 IEEE International Conference on Image Processing*. pp. 4669-4672.

Ibrahim, A.W.N.; Ching, P.W.; Seet, G.L.G..; Lau, W.S.M.; Czajewski, W. (2010): Moving Objects Detection and Tracking Framework for UAV-based Surveillance. 2010 Fourth Pacific-Rim Symposium on *Image and Video Technology (PSIVT)*, pp. 456-461.

Cheraghi, S.A.; Sheikh, U.U. (2012): Moving Object Detection Using Image Registration for a Moving Camera Platform. *2012 IEEE International Conference on Control System, Computing and Engineering*, pp. 355-359.

Walha, A.; Wali, A.; Alimi, A.M. (2014): Vedio stabilization with moving object detecting and tracking for aerial video surveillance. *Multimedia Tools Application*, published online.

Han, B.; Paulson, C.; Wu, D. (2012): Depth-based image registration via three-dimensional geometric segmentation. *Computer Vision, IET,* vol. 6, no. 5, pp. 397-406.

Jia, Z.Y.; Gallagher, A.; Chang, Y.J.; Chen, T. (2012): A Learning-Based Framework for Depth Ordering. *2012 IEEE Conference on Computer Vision and Pattern Recognition (Cvpr)*, pp. 294-301.

Harris, C.; Stephens, M. (1988): A COMBINED CORNER AND EDGE DETECTOR. *Proc of 4th AlVey Vision Conference*, pp. 147-152.

Bouguet, J.Y. (2000): Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm. *Intel Corporation Microprocessor Research Labs*, pp. 1-9.

Mikolajczyk, K.; Schmid, C. (2002): An affine invariant interest point detector. *European Conference on Computer Vision*. pp. 128-142.

Cheraghi, S.A.; Sheikh, U.U. (2012): Moving Object Detection Using Image Registration for a Moving Camera Platform. *2012 IEEE International Conference on Control System, Computing and Engineering*, pp. 355-359.

Zhang, S.P.; Yao, H.X.; Liu, S.H. (2008): Dynamic Background Modeling and Subtraction Using Spatial-Temporal Local Binary Patterns. *2008 15th IEEE International Conference on Image Processing*, pp. 1556 - 1559.

Tavares, J.M.R. (2014): Analysis of Biomedical Images based on Automated Methods of Image Registration. *Advances in Visual Computing, Lecture Notes in Computer Science*, vol. 8887, pp. 21-30.

Alves, R.S.; Tavares, J.M.R. (2015): Computer Image Registration Techniques Applied to Nuclear Medicine Images. *Computational and Experimental Biomedical Sciences: Methods and Applications*, vol. 21, pp. 173-191, 2015.

Bastos, L.F.; Tavares, J.M.R.S. (2010): Improvement of modal matching image objects in dynamic pedobarography using optimization techniques. *Progress in Computer Vision and Image Analysis*, vol. 73, pp. 339-368.

Oliveira, F.P.M.; Tavares, J.M.R.S. (2014): Medical image registration: a review. *Computer Methods in Biomechanics and Biomedical Engineering 17*, pp. 73-93.

Pinho, R.R.; Tavares, J.M.R.S.; Correia, M.V. (2005): A Movement Tracking Management Model with Kalman Filtering, Global Optimization Techniques and Mahalanobis Distance. Lecture Series on Computer and Computational Sciences, vol.1, pp. 1-3.

Pinho, R.R.; Tavares, J.M.R.S. (2009): Tracking Features in Image Sequences with Kalman Filtering, Global Optimization, Mahalanobis Distance and a Management Model. Computer Modeling in Engineering & Sciences, pp. 51-75.

Tavares, J.M.R.S.; Padilha, A. (1995): Matching Lines in Image Sequences with Geometric Constraints. RecPad'95 - 7th Portuguese Conference on Pattern Recognition.

Pinho, R.R.; Tavares, J.M.R.S.; Correia, M.V. (2007): An Improved Management Model for Tracking Missing Features in Computer Vision Long Image Sequences. WSEAS Transactions on Information Science and Applications, pp. 196-203.

Pinho, R.R.; Tavares, J.M.R.S.; Correia, M.V. (2006): An Improved Management Model for Tracking Multiple Features in Long Image Sequences. WSEAS Transactions on Information Science and Applications, pp. 2165-2171.