

Background Subtraction for Moving Cameras based on trajectory-controlled segmentation and Label Inference

Xiaoqing Yin¹, Bin Wang², Weili Li¹, Yu Liu¹, Maojun Zhang¹

¹ College of Information System and Management, National University of Defense Technology
Changsha, Hunan, 410073 – China

² Facility Design and Instrumentation Institute, China Aerodynamics Research and Development Center
Mianyang, Sichuan, 621000 - China
[e-mail: yinxiaoqing89@gmail.com]

*Corresponding author: Xiaoqing Yin

*Received February 8, 2015; revised July 17, 2015; accepted August 11, 2015;
published October 31, 2015*

Abstract

We propose a background subtraction method for moving cameras based on trajectory classification, image segmentation and label inference. In the trajectory classification process, PCA-based outlier detection strategy is used to remove the outliers in the foreground trajectories. Combining optical flow trajectory with watershed algorithm, we propose a trajectory-controlled watershed segmentation algorithm which effectively improves the edge-preserving performance and prevents the over-smooth problem. Finally, label inference based on Markov Random field is conducted for labeling the unlabeled pixels. Experimental results on the motionseg database demonstrate the promising performance of the proposed approach compared with other competing methods.

Keywords: Trajectory classification, trajectory-controlled watershed segmentation, label inference.

1. Introduction

Background subtraction algorithms aim to identify moving objects in the scene for subsequent analysis. Effective methods of isolating these objects such as background modeling approach have been achieved using stationary cameras. These methods have promoted the development of the visual surveillance technology on object detection, recognition and tracking [1]. However, the assumption that the camera keeps stationary limits the application of the traditional background subtraction algorithms in moving camera platforms such as mobile phones and robots. Larger percentage of video content obtained by moving cameras leads to the increasing need for effective algorithms that can isolate moving objects in such video sequences [2, 3].

Effective methods are proposed to handle this problem, which separate the foreground and background using feature point trajectory [1-3]. Long term trajectories are obtained based on the analysis of dense points in long sequences of frames and used to accumulate motion information over frames, which prevents merging of objects that move in different patterns. Afterwards, the foreground and background trajectories are distinguished by applying the trajectory classification approach. Trajectory points are labeled as foreground/background points according to the results of trajectory classification. Thereafter, the labeling approach for pixels is performed to obtain foreground segmentation.

Background subtraction algorithms for moving cameras can be divided into two categories: point trajectory based clustering [2, 4-6] and spatio-temporal segmentation [7, 8]. Point trajectory based methods track points to extract trajectories and cluster trajectories according to motion similarity. Dense point representation of moving objects is obtained and then turned into dense regions [5]. Although the effectiveness of trajectory-based methods has been proven by experiments on various datasets, certain problems remain including the inaccurate trajectory classification and edge-preserving performance of moving objects, which affect video segmentation accuracy. Spatial-temporal segmentation methods extend image segmentation to spatial-temporal domain. Although the segmentation results keep temporal consistent, these methods often encounter the over-smooth problems. Without the guidance of foreground/background feature points, imprecise boundary location for moving objects is caused by inaccurate region classification and merging results.

Our goal is to combine the point trajectory clustering with image segmentation and label inference to develop a new form of background subtraction algorithm for video sequences captured by freely moving cameras. The new method can improve the accuracy of foreground segmentation. Videos captured by moving cameras are taken as input and a foreground/background binary mask for each frame in the video sequence is generated by the proposed algorithm.

The rest of the paper is organized as follows. In Section 2, we review previous research on trajectory classification and image segmentation. In Section 3, we describe our point trajectory classification method. PCA algorithm is used to remove the outliers for the foreground trajectories. In Section 4, we present the trajectory-controlled watershed segmentation algorithm which effectively improves the edge-preserving performance and prevents the over-smooth problem. In Section 5, we propose a Markov Random field based label inference algorithm for labeling the unlabeled pixels. In Section 6, we present the experimental results of the proposed method and compare this method with conventional ones. In Section 7, we summarize the main conclusions and discuss directions for future research.

2. Related Work

Foreground segmentation from video sequences for moving cameras has drawn attention to the field of computer vision and video processing over past decades [1-7]. The motion analysis of point trajectories is an effective tool for foreground segmentation.

To address the problem of trajectory classification, Sheikh et al. [3] used RANSAC to estimate the basis of 3D trajectory subspace with the inliers and outliers of trajectories corresponding to the background and foreground points, respectively. However, a time-consuming implementation is required to achieve accurate estimations and all trajectories should be obtained accurately, thus limiting the applicability of RANSAC. Spectral clustering-based methods [4] use information around each point to build a similarity matrix between pairs of points and implement segmentation by applying spectral clustering. These methods often face difficulties in addressing points near the intersection of two subspaces.

To preserve the boundaries of moving objects, the image is segmented into local regions and region classification is conducted according to the types of points. Regions containing the foreground or background points are labeled as corresponding regions, and the regions without detected points are labeled by comparing the region trajectories with the point trajectories. Various methods [10-13, 19, 20] have been applied to video segmentation. Zhang et al. [13] proposed a video object segmentation method based on the watershed algorithm and clustering region trajectories. Jiman Kim [19] and Yanli Wan [20] proposed new types of moving object segmentation methods for moving cameras.

However, the conventional watershed algorithm fails to explicitly preserve the boundary fragments, thus leading to the deformation of the shape and contour of moving objects. As a result, regions on the contour of the moving objects are likely to encounter inaccurate labeling. Marker-controlled watershed segmentation [16, 17] is an efficient unsupervised segmentation algorithm which is able to suppress the over-smooth problem and obtain more accurate regions boundaries. However, the marker-extraction step of this algorithm is a hard task due to the difficulty of extracting markers exactly. As foreground/background trajectory points can be regarded as markers, the combination of dense trajectories and watershed algorithm provides new direction for the improvement of marker-controlled watershed segmentation.

To achieve unsupervised labeling, existing labeled pixels are used to conduct inference for the unlabeled pixels. Based on the inference algorithm proposed by Sheikh et al. [3], we implement a MRF based inference algorithm for labeling foreground/background pixels.

On the basis of the previous research, we present a new method for background subtraction based on trajectory classification, image segmentation and label inference. The proposed approach overcomes the limitations of traditional methods and provides pixel-wise foreground/background labeling on video sequences captured by hand-held cameras.

3. Point Trajectory classification

We define a novel low-dimensional descriptor for describing trajectory shape, including the displacement vector [14]. If a trajectory is given with the start position and length, the motion displacement vector for the trajectory is expressed as follows:

$$\Delta T = [P_{s+l-1} - P_s] = [x_{s+l-1} - x_s, y_{s+l-1} - y_s] \quad (1)$$

The overall displacement is used to describe the motion of the trajectories. The motion similarity of the trajectories i, j is measured by $S(T_i, T_k) = \|\Delta T_i - \Delta T_k\|$, which is applied in the following clustering approach. Given the samples in the motionseg database [1], the trajectories of the foreground and background are clustered (Fig. 2).

However, the trajectory set still includes some trajectories with missing parts or large errors due to the errors of optical flow algorithm. A few outliers of trajectories with false length and shape may have the same displacement as the correct foreground trajectories. Therefore, after the classification based on displacement vector, it is needed to remove the outliers of the trajectories. Few researchers conducted outlier detection for foreground trajectories in their research. In this study, we apply an outlier detection strategy based on PCA to remove the false trajectories.

The observation matrix D is constructed with each column denoting one of the dense trajectories:

$$D = [T_1, T_2, \dots, T_{N_T}] = \begin{bmatrix} \Delta x_1^{(1)} & \Delta x_1^{(2)} & \dots & \Delta x_1^{(N_T)} \\ \Delta y_1^{(1)} & \Delta y_1^{(2)} & \dots & \Delta y_1^{(N_T)} \\ \vdots & \vdots & & \vdots \\ \Delta x_{N_F-1}^{(1)} & \Delta x_{N_F-1}^{(2)} & \dots & \Delta x_{N_F-1}^{(N_T)} \\ \Delta y_{N_F-1}^{(1)} & \Delta y_{N_F-1}^{(2)} & \dots & \Delta y_{N_F-1}^{(N_T)} \end{bmatrix} \quad (2)$$

where $\Delta x_i^{(j)} = x_{i+1}^{(j)} - x_i^{(j)}$, $\Delta y_i^{(j)} = y_{i+1}^{(j)} - y_i^{(j)}$ are the variations for the coordinates of corresponding dense points, $T_j = [\Delta x_1^{(j)}, \Delta y_1^{(j)}, \dots, \Delta x_{N_F}^{(j)}, \Delta y_{N_F}^{(j)}]^T$ is the j^{th} trajectory and N_T, N_F denote the number of trajectories and frames, respectively.

As principle component analysis performs well on reducing the dimensionality of a data set consisting of large numbers of variables, it is suitable for reducing the dimension of the trajectory vector and finding the inner structure of the data. There are two advantages for using PCA:

(1) As outliers are detected based on the low-dimension data produced by PCA, errors of sampling (or missing) trajectory points will not significantly impair the outlier detection result;

(2) The complexity of outlier detection is decreased.

The outliers are further detected by measuring the difference between the trajectory and its nearest neighbors. The outlier detection strategy is conducted as follows:

Step 1. Extract the principle components of the trajectory vectors to reduce the dimension of data.

Step 2. For the principal component score of each trajectory, seek the k nearest neighbors.

Step 3. Detect the outliers based on the sum of Euclidean distance to the k nearest neighbors. Trajectories with the $lp\%$ largest distance are detected as outliers.

In our experiment, we set $k = 10$, $lp = 5$. The representative trajectory classification results of Cars 1, Cars 4 and Marple 7 sequences in the motionseg dataset [1] are illustrated in Fig. 1. The points in red and those in blue represent the starting points of background and foreground trajectories, respectively. The background points in red circles are falsely detected as part of the foreground. These false trajectories are removed using PCA-based outlier detection strategy.



Fig. 1. Classification result using motion displacement for cars1 in the motionseg dataset [2]. The left column represents the initial classification result. The points in red and those in other colors represent the starting points of background and foreground trajectories, respectively. In the right column, some background points are mistakenly detected as part of the foreground, which are labelled by red circle. These noisy points are removed by using PCA method.

Based on the point trajectories which are divided into foreground and background trajectories, the movement of the objects is accurately represented, which will contribute to the foreground construction in the following.

4. Trajectory-controlled Watershed Segmentation

Watershed segmentation algorithm is an effective segmentation algorithm with relatively low computational complexity, which generates watershed regions with boundaries closely related to edges of objects and reveals structure information in images. However, the conventional watershed algorithm suffers imprecise location of region boundaries. Structural

information of contours and details in the video sequences are affected by the inaccurate location of region boundaries.

To overcome the problems discussed above, we propose a trajectory-controlled watershed segmentation algorithm. We combine optical flow trajectory with marker-controlled watershed algorithm and present a trajectory-controlled watershed segmentation algorithm to address the background subtraction problem for moving cameras. The procedure of the proposed segmentation algorithm is illustrated in **Fig. 2**.

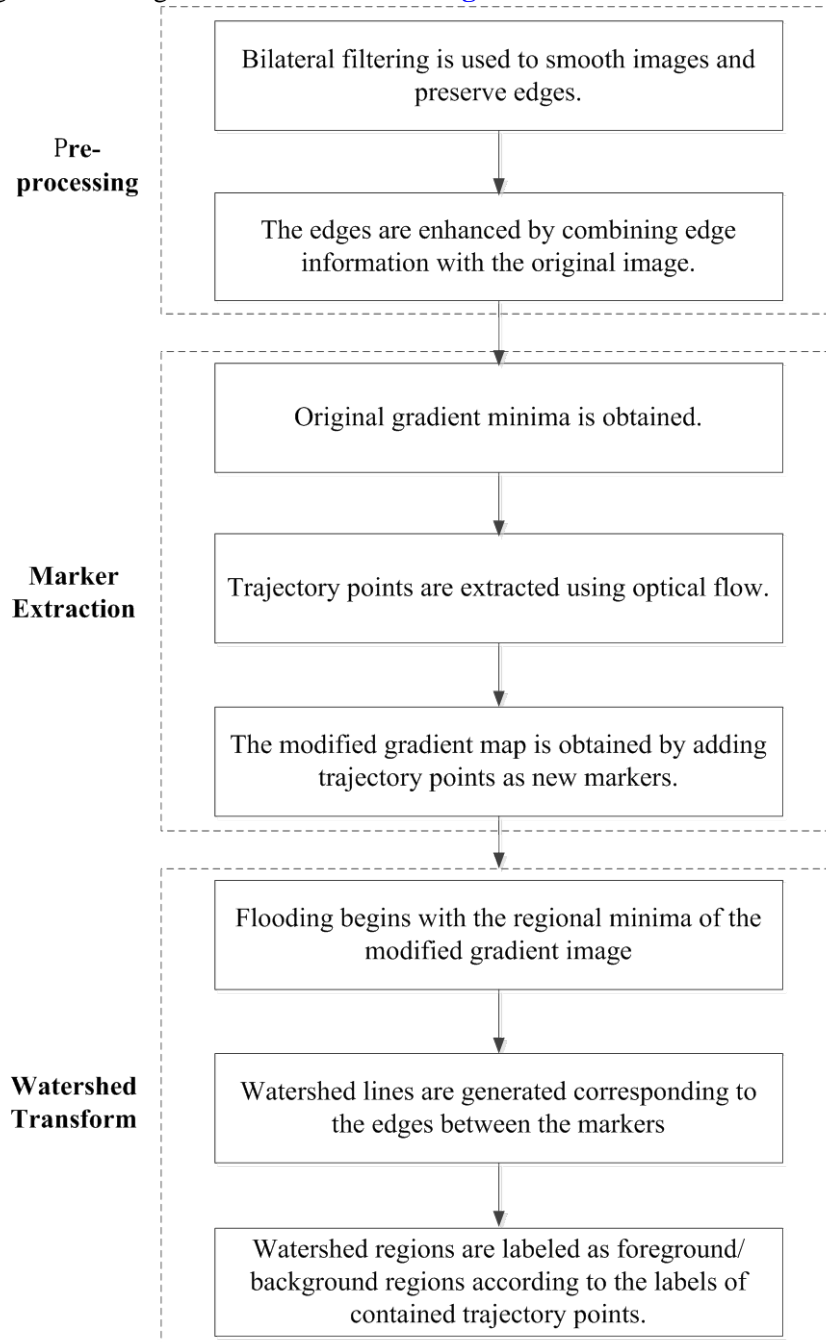


Fig. 2. The procedure of the proposed segmentation algorithm

4.1 Pre-processing

To reduce noise and enhance the target boundaries, a pre-processing step is needed. Bilateral filtering [15] is a good solution to smoothing images effectively and preserving edges. To further improve the performance of the segmentation problem, bilateral filtering is applied to remove small image details and noise while prevent the blurring of boundary structures.

The edges are enhanced by combining edge information with the original image:

$$I = k_e I_e + k_o I_o \quad (3)$$

where I_o and I_e are the original image and the gradient magnitude, respectively, and k_e, k_o are the coefficients. By calculating the weighted sum of edges and the original image and increasing the value of $\frac{k_e}{k_o}$, the intensity of edge information is increased and the edges are enhanced compared with smooth regions. With the enhancement of edges, watershed transform generates more continuous and accurate watershed lines along the edges, and foreground/background boundaries are then precisely constructed based on the watershed lines. In the experiments we tested a series of $\frac{k_e}{k_o}$ value to obtain the proper ratio and found that $\frac{k_e}{k_o} = 2.3$ generates the optimal results on the testing video sequences according to the measure metrics. With the limitation of $k_e + k_o = 1$, the parameters are set as $k_e = 0.3, k_o = 0.7$.

4.2 Marker Extraction

The original gradient minima of background/foreground parts, which are homogenous regions with similar gray values, are chosen as the initial markers.

As useful prior knowledge for identifying the foreground and the background, trajectory points are regarded as components that mark the smooth regions of an image. Approximate shape and contour of the moving objects are represented by trajectory points. First estimation of the segmentation is therefore obtained. The sparsity of trajectory points also suppresses the over-segmentation problem. Trajectory points are selected as new markers to guide the generation of watershed regions like seeds in the region growing process. Therefore the problem of extracting markers is solved.

For an example frame in the Marple 7 video sequence, the original gradient minima is firstly obtained as shown in Fig. 3(a). Trajectory points (Fig. 3(b)) based on optical flow are added to original gradient minima and imposed as minima of the gradient function. The combined gradient minima is taken as markers that guides watershed segmentation. The input marker image for watershed segmentation is a binary image that consists of marker points, where each marker corresponds to a specific watershed region.

4.3 Watershed Transform

After the modified gradient image is obtained, watershed transform [16] is applied to find the accurate contour of the moving objects. The detailed strategy is as follows:

1. Flooding begins from regional minima of the modified gradient image and generates connected regions. The flooding step is achieved by erosion and dilatation

operations. Each connected region contains one local minimum in the gradient image. As trajectory points are imposed as region minima in the gradient image, they act as seeds in the flooding process and guide the formation of watershed regions.

2. Regions without markers are allowed to be merged.
3. Based on morphological dilation in step 1, dam construction is performed to generate the watershed lines, which correspond to significant edges between the regions. As each trajectory points generates a watershed region in step 1, the watershed lines construct the contours of these regions, which are also the boundaries of the foreground and background regions.

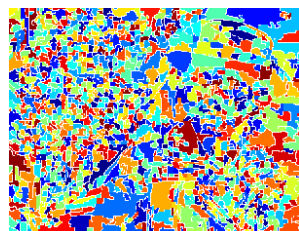
Segmentation result of watershed transform is shown in **Fig. 3(c)**. Based on the segmentation result, foreground/background labeling is performed for all the regions. Watershed regions containing trajectory points are labeled as foreground/background regions according to the labels of corresponding trajectory points. As shown in **Fig. 3(d)**, watershed regions with foreground and background trajectory points are indicated in white and black, respectively, and the gray parts indicate regions without trajectory points.



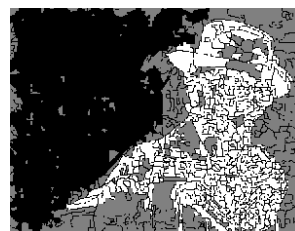
(a) Original gradient minima



(b) Trajectory point image



(c) Segmentation result



(d) Binary labeling based on segmentation

Fig. 3. Illustration of the proposed segmentation algorithm. (a) Original gradient minima image. White regions represent the regional minima. (b) Trajectory point image. White and grey points indicates the locations of foreground/background trajectory points, respectively. (c) Segmentation result. All the regions are painted as different colors. (d) Foreground/background labeling based on segmentation. The regions containing the foreground trajectory points are labeled as foreground regions (white parts). Background regions and unlabeled regions are represented as black and grey parts, respectively.

As shown in **Fig. 3(d)**, although most of the regions have been identified as foreground/background parts, the segmentation result for each frame still contains a few unlabeled regions which impairs the completeness of moving objects. In order to improve the accuracy of background subtraction, a label inference procedure conducts binary labeling

for each unlabeled pixel according to the probability belonging to the foreground/background, which is described in Section 5.

5. Label Inference for the Unlabeled Pixels

In this section, existing labeled regions are used to conduct inference and the unlabeled pixels are identified as foreground/background pixels.

We define the feature for each region R using a color-location vector:

$$v(R) = [\bar{r}, \bar{g}, \bar{b}, x_c, y_c] \quad (4)$$

where $\bar{r}, \bar{g}, \bar{b}$ are the mean values of color vector (r, g, b) for the region and (x_c, y_c) denotes the coordinates of the region center: $x_c = \frac{1}{n_p} \sum_{i=1}^{n_p} x_i, y_c = \frac{1}{n_p} \sum_{i=1}^{n_p} y_i$, where n_p is the number of pixels in the region. For an unlabeled pixel u , the color-location vector is denoted as $v(u) = [r, g, b, x, y]$ where $[r, g, b]$, $[x, y]$ are the color vector and coordinates of pixel u . Based on the labeled regions, the probability of belonging to the foreground/background $p_f(u), p_b(u)$ can be calculated as follows:

$$\begin{cases} p_f(u) = \frac{\sum_{i=1}^{N_f} n_{fi} \tau[v(u) - v(r_{fi})]}{\sum_{i=1}^{N_f} n_{fi}} \\ p_b(u) = \frac{\sum_{j=1}^{N_b} n_{bj} \tau[v(u) - v(r_{bj})]}{\sum_{j=1}^{N_b} n_{bj}} \end{cases} \quad (5)$$

In **Equation (5)**, $\tau(x) = \|\Delta V_c\|^\alpha \|\Delta X\|^\beta$ measures color and location difference. $\Delta V_c = [r - \bar{r}, g - \bar{g}, b - \bar{b}]$ and $\Delta X = [x - x_c, y - y_c]$. The combination $\alpha = -1, \beta = -1$ generates the optimal results in our experiments. N_f, N_b denote the number of regions which have been identified as the foreground/background regions. n_{fi}, n_{bj} denote the number of pixels for foreground/background regions r_{fi}, r_{bj} ($i = 1 \dots N_f, j = 1 \dots N_b$).

The objective of this method is to solve the solution L^* that satisfies the following equation:

$$L^* = \arg \max_L [p(L) \prod_{j=1}^N p(u | l_j)] \quad (6)$$

where $p(u | l_j) = \begin{cases} [p_b(u)]^{-1}, & l_j = 0 \\ p_f(u), & l_j = 1 \end{cases}$ and the prior $p(L)$ is computed by constructing a pairwise

Markov Random Field:

$$p(L) = \exp \left(\lambda \sum_{j=1}^N \sum_{k=1}^N (l_j l_k + \eta(1-l_j)(1-l_k)) \right) \quad (7)$$

Where λ sets the degree of smoothness that is imposed by $p(L)$. In our experiment, the value of λ is calculated as follows:

$$\lambda = \frac{\sum_{i=1, l \in S_l}^N \log \left(\prod_{j=1}^N p(u | l_j) \right)}{\sum_{j=1, k=1, j \in S_l, k \in S_l}^N (l_j l_k + \eta(1-l_j)(1-l_k))}$$

where S_l is the set of pixels which have been labeled using the trajectory-controlled watershed segmentation algorithm. As most of the pixels have been labeled as foreground/background in Section 4, the value of λ can be approximately estimated without considering the small number of unlabeled pixels.

The optimizing problem can be converted to an energy minimization problem with the energy function consisting of one data term and one smooth term, which is effectively solved using Graph Cuts [18-19]. The data term in the energy function is the log form of labeling probability:

$$T_D = -\log \left(\prod_{j=1}^N p(u | l_j) \right)$$

which represents the sum of cost for assigning labels in L to pixels in P. And the smoothness term is expressed as follows:

$$T_s = -\lambda \sum_{j=1}^N \sum_{k=1}^N (l_j l_k + \eta(1-l_j)(1-l_k))$$

which is a sum cost over all pairs of neighboring pixels $\{p, q\}$.

The Energy Functions of Graph Cuts is constructed as the combination of the data term and the smoothness term:

$$E = T_D + T_s = -\log \left(\prod_{j=1}^N p(u | l_j) \right) - \lambda \sum_{j=1}^N \sum_{k=1}^N (l_j l_k + \eta(1-l_j)(1-l_k))$$

Neighborhood system is constructed in the form of 4 pixel-connected grid graph. Given the initial labeling of all 0's to the pixels, the energy of the optimized labeling is calculated via expansion moves using min-cut and max-flow algorithm [18]. The Segmentation result obtained by the energy minimization process is shown in Fig. 4 (b).

Unlike the similar MRF described in [3], we introduce a balance parameter $\eta = \frac{N_b}{N_f}$ to

set the degree of label influence, which is related to the number of foreground/background trajectory points. According to our experiments, over-smooth problem of moving objects is significantly suppressed by introducing the label influence term η as shown in Fig. 4.

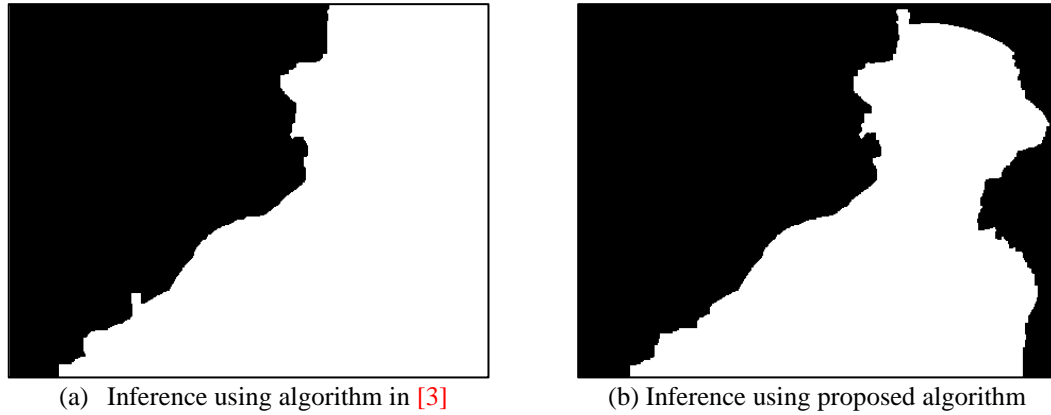


Fig. 4. Comparison of MRF based label inference using Graph Cuts.

6. Experiments

We validate the performance of our method on the motion segmentation dataset provided by Brox et al [2]. The moving objects in this dataset are mainly people and cars. The PC for conducting the experiments has 2 GB memory and 1.60 GHz CPU.

For evaluation purpose, we take Precision and Recall as metrics [6]. Let TP, FP, FN denote the numbers of true foreground pixels, false foreground pixels, and false background pixels, respectively. Then the Precision and Recall metrics can be obtained:

$$Prec = \frac{TP}{TP + FP}, Rec = \frac{TP}{TP + FN}$$

The comparison of average Precision and Recall metrics is shown in **Table 1**. Compared with the algorithms [3, 6, 13, 19, 20], the proposed algorithm achieves the highest Precision and significant Recall, which indicates less false foreground/background pixels and more accurate segmentation result.

Table 1. Comparison on Average Precision and Recall

	Average Precision	Average Recall
Proposed method	0.8372	0.8524
[13]	0.8191	0.8270
[3]	0.6957	0.8903
[6]	0.6135	0.8058
[19]	0.8122	0.8263
[20]	0.8050	0.8465

Fig.5 shows some representative results of our method (video sequences cars 1, cars 4 and marple 7 provided by the dataset). As shown in **Fig. 5**, although the contours of the foreground objects are slightly affected by blurred boundaries and errors of location of trajectory points, our method can generate satisfying segmentation results.



Fig. 5. Representative results on the sequences in motionseg dataset [2] using our method. The three columns correspond to the results of cars1, cars4 and marple 7. From left to right: results of cars1, cars4 and marple 7.

The visual effects of foreground segmentation for [3] [6] [13] [19] [20] and the proposed algorithm are shown in Fig. 6. Compared with algorithms [3, 6, 13, 19, 20], the proposed method generates better visual results of foreground segmentation. Moving objects obtained using the proposed method show more accurate contours and boundaries of the objects.



Fig. 6. Comparison on visual effects of foreground segmentation. The four columns correspond to the results of [3], [13], [6], [19], [20] and the proposed algorithm. From left to right: [3], [13], [6], [19], [20] and the proposed algorithm.

The performance comparison with [3, 6, 13, 19, 20] on precision and recall is shown in Table 2 and Table 3.

Table 2. Comparison of Precision results with [3, 6, 13, 19, 20]

Precision	Cars 1	Cars 4	Marple 7
proposed method	0.8602	0.8945	0.9413
[13]	0.8275	0.8069	0.8518
[6]	0.6358	0.6770	0.7840
[3]	0.7634	0.7803	0.6487
[19]	0.8024	0.8566	0.8092
[20]	0.7963	0.8149	0.8622

Table 3. Comparison of Recall results with [3, 6, 13, 19, 20]

Recall	Cars 1	Cars 4	Marple 7
proposed method	0.9017	0.9591	0.9240
[13]	0.9438	0.9187	0.8911
[6]	0.9643	0.9843	0.8058
[3]	0.9904	0.9853	0.8372
[19]	0.8121	0.9056	0.8235
[20]	0.8347	0.9218	0.8502

In our experiment, processing speed on the dataset is measured by the number of frames processed per second (fps). The comparison results are shown in Table 4. Although the proposed method performs slower than the algorithm in [6, 19], the Precision and Recall results are better. The proposed method outperforms algorithms [3, 13, 20] in terms of both processing speed and segmentation results.

Table 4. Comparison on average processing speed (fps) on the motionseg dataset

Algorithm	Average fps
proposed method	0.356
[3]	0.096
[6]	1.042
[13]	0.263
[19]	0.533
[20]	0.316

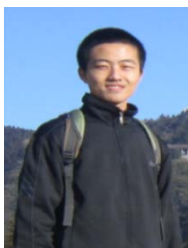
7. Conclusions

In order to cope with the problems of video segmentation on moving cameras, we present a new background subtraction method. Our major work includes trajectory classification, image segmentation and label inference. Satisfactory performance of the proposed approach is shown in the comparison experiments. The consistency of segmentation across the video sequence and more efficient algorithms for hardware implementation are considered as the future research direction of this work. We also find it possible and valuable to introduce interactive segmentation in our method as Grab Cut [21] does.

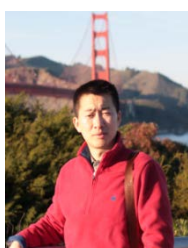
References

- [1] Ali Elqursh and Ahmed Elgammal, "Online Moving Camera Background Subtraction," in *Proc. of 12th European Conference on Computer Vision*, Part VI, 228-241, 2012. [Article\(CrossRef Link\)](#)
- [2] T. Brox and J. Malik. "Object Segmentation by Long Term Analysis of Point Trajectories," *Perspectives in Neural Computing*, 179-193, 2010. [Article\(CrossRef Link\)](#)
- [3] Yaser Sheikh, Omar Javed and Takeo Kanade, "Background Subtraction for Freely Moving Cameras," in *Proc. of International Conference on Computer Vision (ICCV)*, 90-102, 2009. [Article\(CrossRef Link\)](#)
- [4] P. Ochs and T. Brox. "Higher Order Motion Models and Spectral Clustering," in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 614-621, 2012. [Article\(CrossRef Link\)](#)
- [5] Ochs, P. and Brox, T., "Object Segmentation in Video: A Hierarchical Variational Approach for Turning Point Trajectories into Dense Regions," in *Proc. of International Conference on Computer Vision (ICCV)*, 323-330, 2011. [Article\(CrossRef Link\)](#)
- [6] Yosuke NONAKA, Atsushi SHIMADA, Hajime NAGAHARA and Rin-ichiro TANIGUCHI, "Real-Time Foreground Segmentation from Moving Camera based on Case-based Trajectory Classification," in *Proc. of IAPR*, 808-812, 2013. [Article\(CrossRef Link\)](#)
- [7] J. Lezama, K. Alahari, J. Sivic, and I. Lapte, "Track to the Future: Spatio-temporal Video Segmentation with Long-range Motion Cues," in *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition*, 670-678, 2011. [Article\(CrossRef Link\)](#)
- [8] F. Galasso, M. Iwasaki, K. Nobori, and R. Cipolla. "Spatio-temporal Clustering of Probabilistic Region Trajectories," in *Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2011. [Article\(CrossRef Link\)](#)
- [9] John Wright, Yigang Peng and Yi Ma, "Robust Principal Component Analysis: Exact Recovery of Corrupted Low-Rank Matrices by Convex Optimization," vol. 3, no. 10, 45-62, 2008. [Article\(CrossRef Link\)](#)
- [10] Young-Seob Jeong, Chae-Gyun Lim, Byeong-Soo Jeong and Ho-Jin Choi, "Topic Masks for Image Segmentation," *KSII Transactions on Internet and Information Systems (TIIS)*, vol.7, no.12, 3274-3292, 2013. [Article\(CrossRef Link\)](#)
- [11] Jingbo Zhou, Shangbing Gao and Zhong Jin, "A New Connected Coherence Tree Algorithm For Image Segmentation," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 6, no. 4, 547-565, 2012. [Article\(CrossRef Link\)](#)
- [12] L. Grady, "Random Walks for Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 28, no. 11, 1768-1783, 2006. [Article\(CrossRef Link\)](#)
- [13] Geng Zhang, Zejian Yuan, Dapeng Chen, Yuehu Liu and Nanning Zheng, "Video Object Segmentation by Clustering Region Trajectories," in *Proc. of International Conference on Pattern Recognition (ICPR)*, 2598-2601, 2012. [Article\(CrossRef Link\)](#)
- [14] Yu-Gang Jiang, Qi Dai, Xiangyang Xue, Wei Liu and Chong-Wah Ngo, "Trajectory-based Modeling of Human Actions with Motion Reference Points," in *Proc. of European Conference on Computer Vision*, Part V, 425-438, 2012. [Article\(CrossRef Link\)](#)

- [15] S. Kiranyaz, M. Ferreira and M. Gabbouj, "Automatic Object Extraction over Multiscale Edge Field for Multimedia Retrieval," *IEEE Transactions on Image Processing (TIP)*, vol. 15, no. 12, pp. 3759-3772, 2006. [Article\(CrossRef Link\)](#)
- [16] J. M. Gauch, "Image segmentation and analysis via multiscale gradient watershed hierarchies," *IEEE Transactions on Image Processing*, 8(1): 69-79, 1999. [Article\(CrossRef Link\)](#)
- [17] Qingbing Zeng, Yubin Miao, Chengliang Liu and Shiping Wang, "Algorithm based on Marker-controlled Watershed Transform for Overlapping Plant Fruit Segmentation," *Optical Engineering*, vol. 48, no. 2, 21-37, 2009. [Article\(CrossRef Link\)](#)
- [18] Yuri Boykov, Olga Veksler, and Ramin Zabih. "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001. [Article\(CrossRef Link\)](#)
- [19] Jiman Kim, Xiaofei Wang, Hai Wang, Chunsheng Zhu and Daijin Kim, "Fast Moving Object Segmentation for Freely Moving Cameras," *Multimedia Tools and Applications*, 67(1), 311-335, 2013. [Article\(CrossRef Link\)](#)
- [20] Yanli Wan, Xifu Wang and Hongpu Hu, "Automatic Moving Object Segmentation for Freely Moving Cameras," *Mathematical Problems in Engineering*, Accepted, 2014. [Article\(CrossRef Link\)](#)
- [21] Carsten R., Vladimir K. and Andrew B., "*GrabCut: Interactive Foreground Extraction Using Iterated Graph Cuts*," 23(3), 309-314, 2004. [Article\(CrossRef Link\)](#)



Xiaoqing Yin received the B.S., and M.S. degrees in Multimedia Information System and Virtual Reality from National University of Defense Technology, Changsha, Hunan, China, in 2011 and 2013, respectively. Currently, he is working toward the Ph.D. degree in the College of Information System and Management, National University of Defense Technology. His current research interests include image and video stitching, object detection and recognition.



Bin Wang received the B.S., M.S. and Ph.D. degrees in Control Science and Engineering from College of Information System and Management, National University of Defense Technology, Changsha, Hunan, China, in 2006, 2008 and 2013, respectively. He is currently working in the Facility Design and Instrumentation Institute, China Aerodynamics Research and Development Center, Mianyang, Sichuan. His current research interests include machine learning, object detection and recognition, image and video denoising.



Weili Li received the B.S. degrees in System Engineering from National University of Defense Technology, Changsha, Hunan, China, in 2012. Currently, she is working toward the Ph.D. degree in the College of Information System and Management, National University of Defense Technology. Her current research interests include image and video stitching.



Yu Liu received the B.S. degree from NorthWestern Polytechnical University, Xi'an, China in 2005 and the M.S. degree on image processing and Ph.D. degree on computer graphics from University of East Anglia, Norwich, UK, in 2007 and 2011, respectively. He is currently a lecturer in the Department of System Engineering, College of Information System and Management, National University of Defense Technology, Changsha, China. His current research interests include image and video processing, computer graphics, and visual-haptic technology.



Maojun Zhang received the B.S. and Ph.D. degrees in System Engineering from National University of Defense Technology, Changsha, China, in 1992, and 1997 respectively. He is currently a professor in the Department of System Engineering, College of Information System and Management, National University of Defense Technology. In 2001, he became a visiting professor at the University of Ottawa, Canada. His research interests include computer vision, image and video stitching, information system engineering, system simulation and virtual reality technology.