15th IAPR International Conference on Machine Vision Applications (MVA)
Nagoya University, Nagoya, Japan, May 8-12, 2017.

A New Algorithm for Fast and Accurate Moving Object Detection Based on Motion Segmentation by Clustering

Yuchi Zhang[1] Guolin Li[1] Xiang Xie[2] Zhihua Wang[2]
Department of Electronic Engineering[1] Institute of Microelectronics[2]
Tsinghua University, Beijing, China
guolinli@tsinghua.edu.cn

Abstract

In this paper, we propose a training-free method for moving object detection in video sequences. Our method is mainly based on a novel clustering algorithm of accuracy and simplicity. For each frame, dense optical flow between its previous frame and itself is firstly measured. Then for each region whose optical flow is high, the clustering method is applied on the histogram of optical flow orientation to segment different moving objects which are close to each other. Lastly, the consistency of motion vectors of each moving object candidate is verified and the final detecting results are obtained. Experiments on videos in three public datasets show that our algorithm achieves a fast speed of at least 8.01 frames (compared to 1.25) per second and a high recall of at least 87.2% (compared to 83.5%) while the precision is 93.5% (compared to 89.8), which outperform the state-of-art algorithm.

1. Introduction

Moving object detection is a basic and widely used technique in many vision applications, such as pedestrian and vehicle detection in surveillance videos [1], outdoor navigation for robots [2], and simultaneous localization and mapping (SLAM) [3]. In general, existing methods for moving object detection can be broadly classified into two categories: one is based on machine learning, and the other makes use of difference between one frame and another frame or a modeled intensity distribution.

For the first kind of methods, trained classifiers are used as object detectors. For each frame of the video, a slide window is used to scan all over the image. The features in each window are extracted as the input of the classifier. For example, Viola and Jones [4] use cascaded classifiers based on simple features to detect object rapidly. This kind of method behaves well in many object detection tasks. However, there are two drawbacks of such method. One is that large datasets with labels of objects are required, and the training process might be time-consuming. The other is that a trained classifier could only detect one kind of interesting object. In a surveillance video for instance, a trained vehicle detector could only detect cars, buses or trucks on a road while walking pedestrians in the video are missed.

The second kind of methods could be further divided into two different kinds of algorithms. One is based on background subtraction [5]. By comparing to a background model, foreground objects are obtained. This algorithm is fast and does not require any prior information of moving objects, but is challenged by global illumination variation, relocation of background objects and complex background. The other is based on motion segmentation [6]. The previous frame of each frame is used to find out groups of moving pixels, thus each moving object is located. Such method bypasses the negative influence of complicated background and gradual illumination changing. However, it could still make mistakes due to noise and interference of moving background objects like waving tree branches.

To improve the motion based method, Bao *et al* [7] propose a two-stage training-free detecting system based on motion segmentation and motion saliency and consistency verification. Their framework, named as HiCoMo, could handle some challenging detection tasks such as various view point and occlusion. However, their method is not efficient, for example, it can usually process 2 or 3 frames per second when the resolution is 320×240. And it's hard to set an appropriate threshold for different pixel groups to merge according to their colors or optical flows.

In this paper, we propose a new algorithm based on a simple and reliable clustering method. Such method is very fast, and could accurately segment motions. We implement our algorithm along with that in [7] on several videos from three public datasets. The performance metrics including recall, precision and frame rate show that our algorithm performs better.

The rest of this paper is organized as follows. In Section 2, we introduce the clustering method and its application in our detecting system. The whole procedure of our algorithm is introduced in Section 3. In Section 4, we show our experimental results and compare this work to [7]. Finally, we conclude the paper in Section 5.

2. A Simple and Reliable Clustering Method

Clustering is a very effective algorithm to decompose different distributions in a dataset. For our detecting task, such method is used to segment objects which are close in distance but have different motions. There are various kinds of clustering algorithms. The most famous one is K-means [8]. This method is simple and accurate, thus is widely used in many clustering tasks. However, the K-means algorithm requires users to set the number of clusters K. And in our detecting system, we do not know how many objects there are in a high optical flow region before we find them out. Therefore, a new method is referred to by us to solve such problem.

Alessandro *et al* [9] propose a simple and accurate algorithm which does not require the prior information of the number of clusters. They regard centers of clusters as

peaks in the density of points. Moreover, these density peaks are far away from other points which are of more point density nearby. For a point i, its point density ρ_i is defined as follows:

$$\rho_i = \sum_j \chi(d_{ij} - d_c) \tag{1}$$

where χ is a kernel function which could be of Guassian or Cut-Off formula, and d_c is a cut-off distance as explained in detail in [9].

For each point i, another parameter δ_i is defined as:

$$\delta_i = min_{j:\rho_j > \rho_i}(d_{ij}) \tag{2}$$

which measures the distance of the closest point of higher density. For the point whose density is the largest, δ_i is set to be the distance from itself to the farthest point.

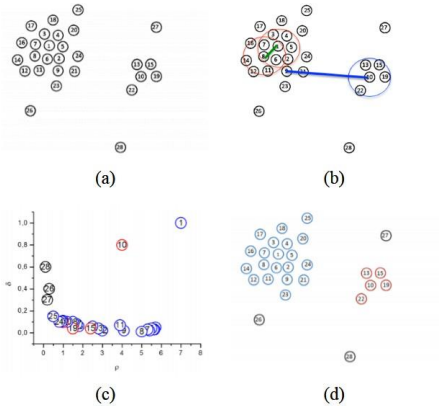

Figure 1. The approach of clustering method in [9]. (a) The original 28 points. (b) Calculation of the density of each point. (c) The decision graph. (d) The final results of the clustering algorithm.

Figure 1 shows how the clustering method in [9] works as an example. For each point, its ρ_i and δ_i are calculated using (1) and (2). Then we plot each δ_i versus its corresponding ρ_i in a same graph named as the decision graph. According to the decision graph, outliers whose ρ and δ are large at the same time are picked out as seeds. Thus these points could be clustering into categories centered by these seeds. Finally, we assign each point to the same cluster of its nearest neighbor of higher density. And points which are far away from any clustering centers are discarded as they might be noises.

In our detecting system, we refer to [9] and develop a reliable method to segment optical flows. We apply the clustering algorithm on the bins of histograms of optical flow directions instead of the coordinates of the points. The reason is that, optical flow orientation is a better metric for us to distinguish different moving objects than its amplitude. Figure 2 takes a scene of two people walking towards each other as example. The dense optical flow of the current frame (See Figure 2 (a)) is shown in Figure 2(b). We plot the optical flow vector of each pixel of the two close moving persons in Figure 2(c). We use the angle θ between the optical flow and the positive direction of the X-Axis to denote the moving direcdirection:

$$\theta = \begin{cases} arccos\dfrac{u}{\sqrt{u^2+v^2}} & v \geq 0 \\ 2\pi - arccos\dfrac{u}{\sqrt{u^2+v^2}} & otherwise \end{cases} \tag{3}$$

where u denotes the x component and v denotes the y component.

Figure 2 (d) is the histogram of the θ angles. Comparing Figure 2(c) and Figure 2(d), it is easier to find that there are two kinds of motions from Figure 2 (d) than from Figure 2(c), i.e. it is more appropriate to use the moving directions in optical flow segmentation rather than to use the amplitudes.

Instead of using the physical density in (1), we use the probability density. For a histogram H, the probability density of bin i is:

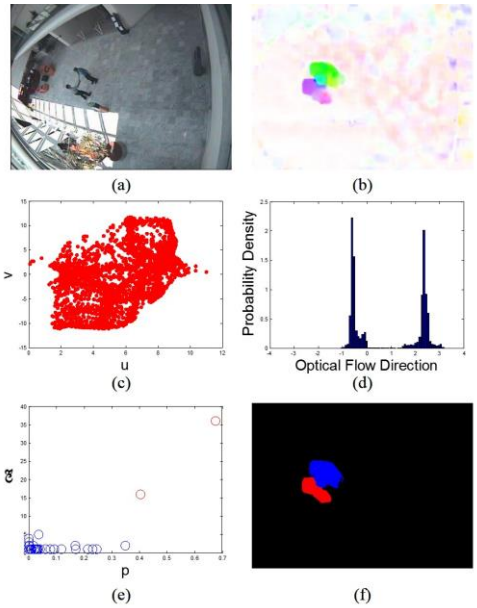

Figure 2. (a) The frame of the scene of two people meeting. (b) Dense optical flow graph of (a). (c) Motion vector distribution of the two persons. (d) Optical flow orientation distribution of the two persons. (e) The decision graph. (e) The final results of motion segmentation.

$$p_i = H(i) \tag{4}$$

where $i \in \{1,2,\dots,N\}$ and N is the number of bins. And its distance metric to the closest bin of higher probability is:

$$\delta_i = min_{j:p_j > p_i} D(i,j) \tag{5}$$

where D is the distance metric of two bins. Because θ and $\theta + 2n\pi$ denote the same direction, i.e. θ is cyclical, in H, the distance from the last bin to the first bin is 1. Thus D is expressed as:

$$D(i,j) = \min(|i-j|, N - |i-j|) \tag{6}$$

Moreover, for the bin i whose $H(i)$ is the largest, we set δ_i to be N.

We plot the δ versus p in the decision graph. For a clustering center, both p and δ are large. We use the product $\gamma = p\delta$ of each point and use the method in [9] to automatically pick out outliers. Moreover, we assume that the peaks are maxima point of probabilities, i.e. its value in the histogram is higher than its two neighbors. And this helps us to reject false peaks based on the results of the method in [9]. We mark the detected outliers using red circles and other points using blue in Figure 2(e). After assigning each bin the same label as the nearest bin of higher probability density, we obtain its own category. Finally, each kind of motion and different moving targets are segmented (See Figure 2(f)).

The proposed clustering method has two advantages: One is that the number of histogram bins is only dozens, thus it doesn't consume much time. The other is that the density and distance metrics are very good feature to distinguish outliers and which cluster a point belongs to.

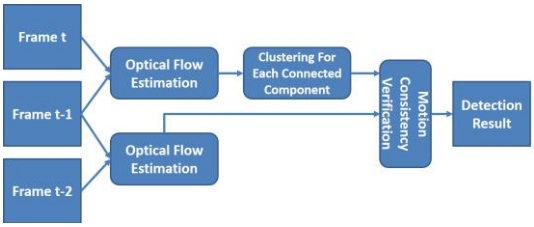

Figure 3. The block diagram of our detecting system.

3. The Architecture of the Proposed Algorithm

The block diagram of our detecting system is shown in Figure 3. Firstly, for each frame t, we measure the dense optical flow between frame t and $t-1$. High optical flow regions are extracted and the clustering method is applied for each independent region to separate different motions secondly. Thirdly, the motion consistency is verified for moving object candidates detected by the last two steps. Finally, we mark all detected moving objects using rectangular bounding boxes.

3.1. Extraction of High Optical Flow Regions

We use the Simple Flow Algorithm [10] to obtain the dense optical flow distribution of a frame. Then pixels of high optical flows are extracted using the criteria: $\sqrt{u^2+v^2} > V_{th}$, where u and v are optical components of x direction and y direction, and V_{th} is a threshold. We set a small $V_{th} = 1$ so that we reject most pixels belong to the background while keep all those belong to the moving objects in the frame. Small regions whose area are smaller than a threshold S_{th} are treated as noises and rejected. In this work, we set $S_{th} = 0.05S_{max}$ where S_{max} is the area of the largest connected component. Figure 4(a) is a frame in a video sequence for example. Figure 4(b) is the dense optical flow graph of such frame. And Figure 4(c) shows the results of the extraction of high optical flow region in a binary image.

3.2. Segment Different Motions

The clustering method we proposed in section II is applied to segment different components of motions, i.e. different probable moving objects. The bins of the histogram is set to be 36 in practice. The segmentation result is shown in Figure 4(d). This clustering process is very fast, as only 36 data points need to be processed.

3.3. Motion Consistency Verification

A moving object could not suddenly appear or disappear, thus the velocities of pixels of a moving object do not vary much between two successive frames. Thus the criteria of pixel-wise motion consistency test of an object at frame t could be written as:

$$\sqrt{\sum_{(x,y)\epsilon M} \frac{\left(u_t(x,y) - u_{t-1}(x',y')\right)^2}{+(v_t(x,y) - v_{t-1}(x',y'))^2}} > U_{th} \qquad (7)$$

where M is the set of pixel coordinates of the object, and (u_t, v_t) is the optical flow at frame t. Coordinates (x,y) and (x',y') are the position at the current frame and the last frame. We use the current position and optical flow to infer the coordinate at the last frame: $x' = x - u_t$ and $y' = y - v_t$. The threshold U_{th} depends on the area of the pixel group $|M|$. And in our algorithm, we set $U_{th} = 2.25|M|$ which means the optical flow could not change over 2.25 pixels of a true moving object between two continuous frames. The final result after the verification process is shown in Figure 4(e).

4. Experiments

We evaluate the performance of our algorithm using three public datasets: Caviar [11], AVSS-2007 [12] and PETS-2009 [13]. We pick out 16 long videos, of which 6 are from [11], 3 are from [12] and 7 are from [13]. These videos contain indoor and outdoor walking people or vehicles and pedestrians in a traffic scene. We carefully delete all positive labels of non-moving objects as some of the videos are for both moving and non-moving vehicle and pedestrian detection. And this is also the reason why we only use part of the videos of the three datasets, as most of the targets in these videos are moving and it is reasonable for us to remove labels of non-moving objects. These videos contain a variety of situations such as different view-points, close targets and occlusions.

We implement our algorithm and the method in [7] using C++ on the same computer with four Intel Core i5-4200U CPUs @ 1.6GHz and 8GB of RAM. We measure the recall and precision which represent the accuracy of detection and the average frame rate which is defined as the total processing time divided by the total frame number. We compare the performance of our algorithm with that in [7].

The comparison of how accurate these two methods are is shown in Table 1. We achieve a same or a little bit higher recall while our precision is higher than [7].

(a) (b) (c) (d) (e)

Figure 4. The whole procedure of our algorithm. (a) The raw frame. (b) Dense optical flow estimation. (c) High optical flow region extraction. (d) Motion segmentation by clustering. (e) The final results

Table 1. Recall (R) and Precision (P)

Dataset		HiCoMo [7]	This work
Caviar	R (%)	92.5	94.3
	P (%)	98.5	99.1
AVSS-2007	R (%)	83.5	87.2
	P (%)	89.8	93.5
PETS-2009	R (%)	88.9	89.6
	P (%)	94.7	95.9

Figure 5 shows the detecting results of the two algorithms on 3 typical frames. Both methods perform well and overcome some typical difficulties such as occlusion. However, the detecting system in [7] always fail to separate several objects which are close to each other.

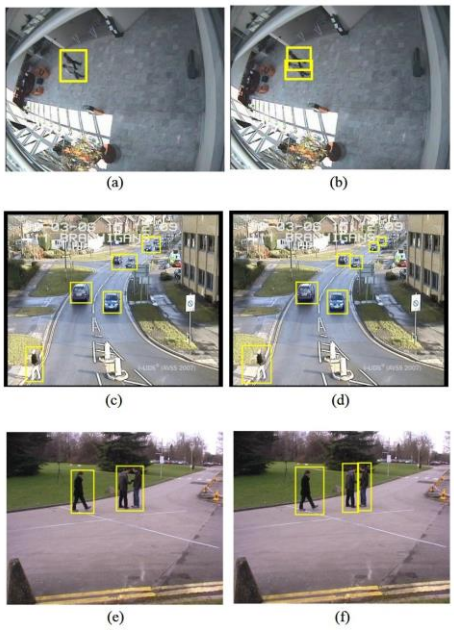

Figure 5. The comparison of the state-of-art algorithm with this work. (a), (c), (e) are the detecting results of the method in [7]. (b), (d), (f) are the detecting results of this work.

Table 2 shows the speed in average frame rate of each algorithm.

Table 2. Average Frame Rate

Dataset	Resolution	HiCoMo [7]	This work
Caviar	384 × 288	2.2	12.12
AVSS-2007	720 × 576	1.25	8.01
PETS-2009	720 × 576	1.57	8.40

From the above experimental results we find that our algorithm is much faster than [7]. The main reason is that the graph based image segmentation and the hierarchical merging process in [7] are very time consuming. We abandon such process and we adopt a very simple clustering method on a histogram with only scores of bins for each object. Thus our algorithm is more efficient.

5. Conclusion

In this paper we present a moving object detecting algorithm. We introduce a novel clustering method and apply it on the histogram of optical flow orientation. As the clustering method is fast and accurate and the number of bins of the histogram is small, we achieve high recall and precision detections while the speed of the algorithm is high. By comparing this work to a related work, we find our algorithm outperforms the state-of-art detecting systems.

Acknowledgement

This work was partly supported by National Natural Science Foundation of China (grant No. 61373073), partly by Shenzhen science and technology project (grants No. JCYJ20150331151358146)，and partly by Shenzhen science and technology project (JCYJ20160301151028370), partly by Independent Research Project of Tsinghua University (grants No.20131089223)

References

[1] Cucchiara R, Grana C, Piccardi M, et al. Statistic and knowledge-based moving object detection in traffic scenes[C]. Intelligent Transportation Systems, 2000. Proceedings. 2000 IEEE. IEEE, 2000: 27-32.

[2] Jung B, Sukhatme G S. Detecting moving objects using a single camera on a mobile robot in an outdoor environment[C]. International Conference on Intelligent Autonomous Systems. 2004: 980-987.

[3] Wang C C, Thorpe C, Thrun S. Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas[C]. Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on. IEEE, 2003, 1: 842-849.

[4] Viola P, Jones M J, Snow D. Detecting pedestrians using patterns of motion and appearance[J]. International Journal of Computer Vision, 2005, 63(2): 153-161.

[5] Heikkila M, Pietikainen M. A texture-based method for modeling the background and detecting moving objects[J]. IEEE transactions on pattern analysis and machine intelligence, 2006, 28(4): 657-662

[6] Brox T, Malik J. Object segmentation by long term analysis of point trajectories[C]. European conference on computer vision. Springer Berlin Heidelberg, 2010: 282-295.

[7] Bao X, Dubbelman G, Zinger S, et al. Training-free moving object detection system based on hierarchical color-guided motion segmentation[C]. Machine Vision Applications (MVA), 2015 14th IAPR International Conference on. IEEE, 2015: 154-157.

[8] MacQueen J. Some methods for classification and analysis of multivariate observations[C]. Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. 1967, 1(14): 281-297.Zhou X, Yang C, Yu W. Moving object detection by detecting contiguous outliers in the low-rank representation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(3): 597-610.

[9] Rodriguez A, Laio A. Clustering by fast search and find of density peaks[J]. Science, 2014, 344(6191): 1492-1496.

[10] Tao M, Bai J, Kohli P, et al. SimpleFlow: a non-iterative, sublinear optical flow algorithm[C]. Computer Graphics Forum. Blackwell Publishing Ltd, 2012, 31(2pt1): 345-353.

[11] http://groups.inf.ed.ac.uk/vision/CAVIAR

[12] http://www.elec.qmul.ac.uk/staffinfo/andrea/avss2007

[13] http://www.cvg.reading.ac.uk/PETS2007