MOTION DETECTION WITH SPATIOTEMPORAL SEQUENCES

Tong Zhang，*Haixian Wang*[*]

Research Center for Learning Science, Southeast University, Nanjing, Jiangsu 210096, PR China
{220112995, hxwang}@seu.edu.cn

ABSTRACT

In this paper we propose a new method to detect motion in a greyscale video. In our algorithm, several spatiotemporal sequences with different lengths are used to filter the frames in the video. Then these filtered images are combined together to get the real motion. The performance of our algorithm is tested with several human action datasets in which different actions are performed. The detected results of our algorithm are compared with previous works and the targets we extract manually. The experimental results show that the responses of our filter are close to the real action of the human in the original video.

Index Terms—motion detection, video surveillance, background subtraction, video processing, spatiotemporal sequences.

1. INTRODUCTION

Human activity recognition has become an active research topic in recent years and drawn a lot of attention due to its important applications, such as video surveillance, video indexing and browsing, and analysis of sports. Background subtraction is usually used to extract the motion in human activity recognition. In the past decades, numerous algorithms of background subtraction have been proposed, in which modeling the pixel color and intensities is a usual method [1]. Another algorithm, Gaussian Mixture Model (GMM) proposed in [2], is also a popular method for background subtraction. Based on a weighted mixture of Gaussians, GMM models the distribution of the values observed over time at each pixel. In [3], a texture-based method using Local Binary Pattern (LBP) histograms is proposed. It shows promising performance in dynamic scenes. The method of Robust Principal Component Analysis (RPCA), is proposed in [4] to recovery a low rank matrix from corrupted observations which can be used to rebuild the background.

Another method of detecting motion information is to apply spatiotemporal filters to the video. Spatiotemporal filters are commonly used for different purposes in video processing. Dollar proposes the interest point detector to detect the points that contain important motion information [5]. It is based on a quadrature pair of 1D Gabor filters applied temporally and a 2D Gaussian filter applied spatially. This interest point detector has been widely used in human action and expression recognition. Action spotting in [6] considers an action as a conglomeration of motion energies in different spatiotemporal orientations. So it performs spatiotemporal energy decomposition to the motion at a point by using broadly tuned 3D Gaussian third derivative filters [7] and then uses these decomposed energies as a low-level action representation. Despite the various purposes of using the temporal sequences, the common process of these methods above is to apply several 1D temporal filters to the video and then sum these filtered results together with different weights. The results of this process contain the motion of the human in both the current frame and the neighbor frames in a temporal window. For motion detection, the purpose is just to detect the motion in the current frame. So if the motion belonging to the neighbor frames could be eliminated from the results, this process could be used to achieve the goal of detecting the real motion.

Inspired by the previous works, in this paper, we introduce a new filter. In our method, a group of temporal sequences with different lengths are firstly applied to the video to get corresponding response. We thus obtain the rough spatiotemporal motion of each sequence by utilizing the temporal information of its response. Then by eliminating the error parts of the rough motion, we can finally obtain the real motion of the video.

2. THE RESPONSE FUNCTION OF INTEREST POINT DETECTOR

The response function of interest point detector proposed in [5] is defined as:

$$R = R_1^2 + R_2^2 \tag{1}$$

$$R_1 = I * g * h_{ev}, \ R_2 = I * g * h_{od} \tag{2}$$

$$h_{ev}(t;\tau,\omega) = -\cos(2\pi t\omega)e^{-t^2/\tau^2}, \ h_{od}(t;\tau,\omega) = -\sin(2\pi t\omega)e^{-t^2/\tau^2}, -r \le t \le r \tag{3}$$

[*] Corresponding author.

where I denotes the original greyscale video and $g(x,y,\sigma)$ is the 2D Gaussian smoothing kernel. The two parameters σ,τ correspond to the spatial and temporal scales of the detector and the parameter r corresponds to the length of the temporal filter. h_{ev} and h_{od} are a quadrature pair of 1D Gabor filters applied temporally. Motions in the video will evoke strong responses when this interest point detector is applied while the static background and slight noise will be suppressed at the same time. Motions belonging to other frames in a temporal window will appear in the current frame which locates in the center of the temporal window at the same time. And the length of the temporal window corresponds to the length of the temporal sequence we have used. Moreover, the motions belonging to the temporal window are weighted according to the value of each point in the temporal sequence.

3. PROPOSED ALGORITHM

In the following sections, the details of our algorithm are described, including the two critical steps which are rough motion detection in Subsection 3.1 and error motion elimination in Subsection 3.2.

3.1. Rough motion detection

Rough motion detection of our algorithm is achieved by convolving the video with spatiotemporal sequences which is inspired by the interest point detector proposed in [5]. In our algorithm, the values of the temporal sequences above are modified in order to preserve the necessary motion while eliminating the most error parts in the temporal window. Our rough motion detector is defined as:

$$Y_p(l_1,l_2,\cdots,l_k)=\max(R_p(l_1),R_p(l_2),\cdots,R_p(l_k)) \tag{4}$$

$$Y_n(l_1,l_2,\cdots,l_k)=\max(R_n(l_1),R_n(l_2),\cdots,R_n(l_k)) \tag{5}$$

$$R_p(u)=\begin{cases}M_1(I,u), & if\,(M_1(I,u)>0)\,\&\,(|M_1(I,u)|-|M_2(I,u)|)>th_1\\ 0, & others\end{cases} \tag{6}$$

$$R_n(u)=\begin{cases}-M_1(I,u), & if\,(M_1(I,u)<0)\,\&\,(|M_1(I,u)|-|M_2(I,u)|)>th_1\\ 0, & others\end{cases} \tag{7}$$

$$M_1(I,u)=I*g*h_1(u),\ M_2(I,u)=I*g*h_2(u) \tag{8}$$

$$h_1(u,t)=\begin{cases}1, & if\ |t|=u\\ -2, & if\ \ t=0\\ 0, & others\end{cases},\quad h_2(u,t)=\begin{cases}u/|u|, & if\ |t|=u\\ 0, & others\end{cases},\quad -u\le t\le u \tag{9}$$

where the values of the parameters $l_1,l_2,\cdots,l_k$ determine the lengths of the temporal sequences. h_1 and h_2 represent two temporal sequences used to detect the motion in the video and u limits the lengths of these sequences. M_1,M_2 are the convolving results corresponding to h_1,h_2. $R_p(l_1),R_p(l_2),\cdots,R_p(l_k)$ make up the positive part of rough motion $Y_p(l_1,l_2,\cdots,l_k)$, which captures the motions of those moving parts whose gray values are higher than the background. Similarly, $R_n(l_1),R_n(l_2),\cdots,R_n(l_k)$ and $Y_n(l_1,l_2,\cdots,l_k)$ capture the motions of the moving parts which has lower gray values than the background. $Y_p(l_1,l_2,\cdots,l_k)$ and $Y_n(l_1,l_2,\cdots,l_k)$ compose the total rough motion.

For a given length l, the motion detector $R_p(l)$ and $R_n(l)$ may miss detecting the necessary motion when dealing with a cyclic action with a period whose value happening to be the same with l. So multiply motion detectors with different lengths $l_1,l_2,\cdots,l_k$ are used to solve this problem, which is shown in (4) and (5). Thus the rough motion detector can capture almost all the necessary motion in the video. Another problem is that an acyclic action may evoke error response if its trajectory overlaps at a specific interval determined by l. The real motion of the video can be gotten after this kind of error information being eliminated.

3.2. Error motion elimination

As Y_p and Y_n capture the motions of the gray values which are higher and lower than the background respectively, the algorithm of eliminating the error motion from the background is defined as:

$$Y=Y_p+Y_n \tag{10}$$

$$Y_p=Y_p-F_p(Y_p),\quad Y_n=Y_n-F_n(Y_n) \tag{11}$$

$$F_p(Z)=\begin{cases}(I-Z/2)*(-h_d), & if\ (I-Z/2)*(-h_d)>th_1\\ 0, & else\end{cases} \tag{12}$$

$$F_n(Z)=\begin{cases}(I+Z/2)*(h_d), & if\ (I+Z/2)*(h_d)>th_2\\ 0, & else\end{cases} \tag{13}$$

$$h_d(N,t)=\begin{cases}2\cdot N, & if\ t=0\\ -1, & others\end{cases},-N\le t\le N \tag{14}$$

where h_d is a temporal sequence used to detect the error motion and I represents the original video. F_p and F_n represent two temporal filters which are able to detect the error motion in Y_p and Y_n respectively. Y is the final motion extracted from the video. N limits the length of the temporal sequence h_d.

The error blobs in the rough motion always occupy the points which belong to the background. We detect these error blobs with a temporal sequence h_d. Assuming Y_p is detected correctly, when Y_p is subtracted from the original video I in (12), the higher gray values of the moving targets in I may be decreased as low as the background. Thus, in the modified video, this part of motion is eliminated and cannot be detected by h_d. However if Y_p contains mistakes, the gray values of the background corresponding to the error blobs are decreased during the process of subtraction. Thus new motion is created. When h_d is applied to the modified video, the new created motion can be detected. The detected error motion will be removed from Y_p as shown in (11). So

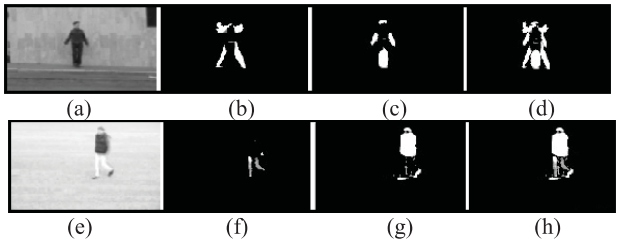

Fig. 1. Rough motion detected from the actions jack and walking. The left column contains the original actions and from the second to the fourth column are the corresponding positive rough motion, negative rough motion and total rough motion.

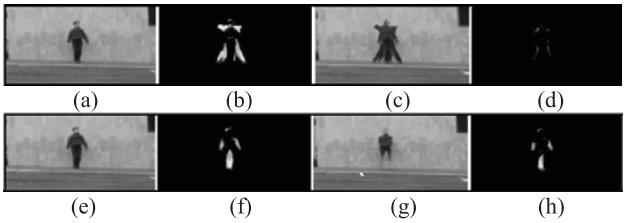

Fig. 2. The process of error motion elimination in (11) and (12). (a) the original action; (b) the positive rough motion; (c) the subtraction of (a) and (b); (d) the final positive motion; (e) the original action; (f) the negative rough motion; (g) the sum of (e) and (f); (h) the final negative rough motion.

by this process, the error motion of Y_p is removed while the correct motion being preserved. The process of removing error motion from Y_n is similar to Y_p and the process is shown in (11) and (13). Thus the mistakes can be eliminated from the rough motion.

4. EXPERIMENTAL RESULTS

In this section, we first describe how we set the values of the parameters. And then we illustrate the process of our method in detail by showing the results of each step. We show the rough motion gotten from several videos and the process of eliminating the error motion. After that, we demonstrate the performance of our method by applying it to several human action datasets, including the KTH dataset, the Weizmann dataset and the Ballet dataset. Finally the results of our algorithm, GMM and RPCA are compared.

The main parameters in these equations are the lengths of the temporal sequences used in our rough motion detector, which are the parameters $l_1, l_2, \cdots, l_k$ in (4) and (5). As we put above, these values are set in case that the rough motion detector may miss detecting necessary motion when dealing with cyclic actions. So $l_1, l_2, \cdots, l_k$ should be set without a common factor. In our experiments, we set the number k to be 4 and l_1, l_2, l_3, l_4 to be 3, 8, 13 and 19.

Fig. 1 shows the results of rough motion detector in (4) and (5). The positive motion mainly contains the moving parts which possess higher gray values than the background,

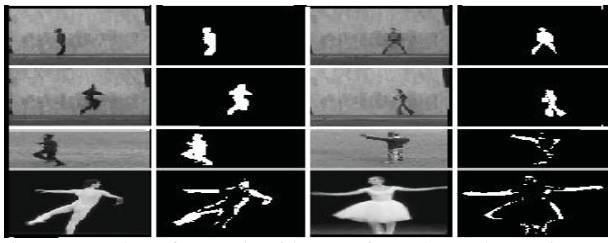
Fig. 3. Results of our algorithm performed on the Weizmann dataset, the KTH dataset and the Ballet dataset, including the actions jump, side, running, boxing, turning and left-to-right hand opening.

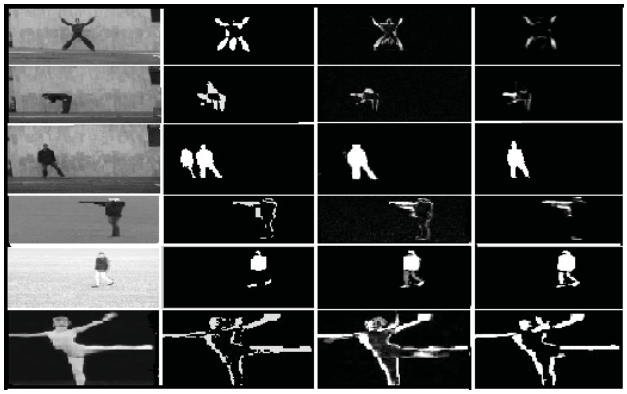
Fig. 4. Comparison of the results between GMM, RPCA and our algorithm. From the left to the right columns are the original action, the ground truth, the result of GMM, the result of RPCA, and the result of our algorithm respectively.

and the negative motion is on the contrary. These results may also contain unexpected motion belonging to other frames in a temporal window. What's more, the action which has a more complex trajectory may evoke much more error responses, just like the action jack in Fig. 1(b).

Fig. 2 shows the process of the error motion elimination with an example of action jack. The positive rough motion shown in Fig. 2(b) contains mistakes. When the error motion is subtracted from the original action, the gray values of the background will be decreased, as shown in Fig. 2(c). And during this process, new motion which doesn't belong to the original video is created. The new created motion in Fig. 2(c) can be detected by h_d and then removed from the positive rough motion. The result is shown in Fig. 2(d). By doing this, mistakes of the rough motion can be eliminated. However, correctly detected motion may change the moving areas into background and thus eliminate the corresponding motion in the original video, as shown in Fig. 2(g). These correct parts of the rough motion cannot be detected from the video by h_d in (12) and (13) and subtracted from the rough motion in (11).

We then apply our algorithm to the KTH dataset, the Weizmann dataset and the Ballet dataset. Some results are represented in Fig. 3. The results show that our algorithm is

Table 1. Comparison between GMM, RPCA and our algorithm

Method	Sensitivity	Specificity	Accuracy
GMM	0.864	0.971	0.969
RPCA	0.971	0.982	0.982
Our method	0.963	0.978	0.978

able to detect the necessary motion from the videos which contain various types of action.

We compare the performances of our algorithm, GMM and RPCA. We choose videos of different actions performed by various people from the three action datasets. Then we apply the three algorithms to these videos and compare the results. Some of them are shown in Fig. 4. The results show that RPCA and our algorithm perform relatively better than GMM because GMM may model the people in the former frames as background. For example, in the process of the action jack, GMM models the first several frames as background, including the actor. When the actor starts acting, the area he previously occupied is wrongly detected as a foreground blob, which is commonly referred as a ghost. This ghost will not disappear until the background model adapts to the newly exposed background. RPCA and our algorithm alleviate this problem, as shown in the first and second row of Fig. 4. Another problem of GMM is that if the gray values of the moving targets are close to the background, then GMM may miss detecting them, just as shown in the action walking in Fig. 4.

To get a quantitative evaluation, we manually segment the moving parts of the human from the videos. Then the performances of the three algorithms are evaluated by comparing the detected motion with the manual results. We use the sensitivity, the specificity and the accuracy in [8] as the statistical measures. These measures are calculated by the formulas in (15), (16) and (17).

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{15}$$

$$\text{Specificity} = \frac{TN}{FP + TN} \tag{16}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{17}$$

In these equations, true positive (TP) represents the number of the correctly classified foreground pixels, false positive (FP) represents the number of the background pixels that are incorrectly classified as foreground, true negative (TN) represents the number of the correctly classified background pixels, and false negative (FN) represents the number of foreground pixels that are incorrectly classified as background. Sensitivity measures the proportion of the actual positives which are correctly identified. Specificity measures the proportion of negatives which are correctly identified.

The quantitative results of GMM, RPCA and the proposed algorithm are shown in Table 1. The results show that the sensitivity, specificity and accuracy of our algorithm are better than those of GMM and approximate the algorithm RPCA. This result is consistent with the conclusion we made according to Fig. 4.

5. CONCLUSION

In this paper, we propose a new method to detect the motion in a video. We describe the algorithm in details and illustrate the process by showing the results of each step. We then apply our algorithm to three action datasets which contain various actions performed by different people. We also compare our algorithm with GMM and RPCA. The results show that in most situations our algorithm outperforms GMM and approximate the performance of PCA.

6. RELATION TO PRIOR WORK

The work presented here has focused on detecting motion in a given video by using spatiotemporal sequences. Although processing videos with spatiotemporal sequences is not new, we expand the use of it to achieve a new goal. We change the formats of those temporal sequences to make them fit the purposes of both rough motion detection and error motion elimination. To compare our algorithm with other methods, we choose action datasets which contain various actions. These actions can evaluate the algorithms comprehensively. We also segment these moving targets from the videos to get a quantitative evaluation.

7. ACKNOWLEDGEMENT

This work was supported in part by the National Natural Science Foundation of China under Grant 61375118, the Natural Science Foundation of Jiangsu Province under Grant BK2011595, and the Program for New Century Excellent Talents in University of China under Grant NCET-12-0115.

8. REFERENCES

[1] K. Kim, T.H. Chalidabhongse, D. Harwood, and L. Davis, "Background modeling and subtraction by codebook construction," in *Proc. IEEE Conf. Image Processing*, vol. 5, pp. 3061-3064, 2004.

[2] C. Stauffer and W.E.L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2, pp. 246-252, 1999.

[3] M. Heikkilä and M. Pietikäinen, "A texture-based method for modeling the background and detecting moving objects," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 657-662, 2006.

[4] J. Wright, A. Ganesh, S. Rao, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *Proc. Neural Information Processing Systems*, vol. 3, 2009.

[5] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pp. 65-72, 2005.

[6] K. G. Derpanis, M. Sizintsev, K. Cannons, and R. P. Wildes, "Efficient action spotting based on a spacetime oriented structure representation," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1990-1997, 2010.

[7] K. Derpanis and J. Gryn, "Three-dimensional nth derivative of Gaussian separable steerable filters," in *Proc. IEEE Conf. International Conference on Image Processing*, vol. 3, 2005.

[8] A. N. Kumar and C. Sureshkumar, "Background subtraction based on threshold detection using modified K-means algorithm," in *Proc. IEEE Conf. Pattern Recognition, Informatics and Medical Engineering*, pp. 378-382, 2013.

[9] D. Parks and S. Fels, "Evaluation of background subtraction algorithms with post-processing," in *Proc. IEEE Conf. Advanced Video and Signal Based Surveillance*, pp. 192-199, 2008.

[10] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognition Letters*, vol. 27, no. 7, pp. 773-780, 2006.

[11] P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in *European Workshop on Advanced Video Based Surveillance Systems*, 2001.

[12] M. F. Abdelkader, R. Chellappa, and Q. Zheng, "Integrated motion detection and tracking for visual surveillance," in *Proc. IEEE Conf. Computer Vision Systems*, pp. 28-34, 2006.

[13] S. Denman, C. Fookes, and S. Sridharan, "Improved simultaneous computation of motion detection and optical flow for object tracking," in *Proc. International Conf. Digital Image Computing: Techniques and Applications*, pp. 175-182, 2009.

[14] G. Jing, D. Rajan, and C. E. Siong, "Motion detection with adaptive background and dynamic thresholds," in *Proc. International Conf. Information, Communications and Signal Processing*, pp. 41-45, 2005.

[15] F. C. Chen and S. J. Ruan, "Accurate motion detection using a self-adaptive background matching framework," *IEEE Trans. Intelligent Transportation Systems*, vol. 13, no. 2, pp. 671-679, 2012.

[16] S. C. Huang, "An advanced motion detection algorithm with video quality analysis for video surveillance systems," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 21, no. 1, pp. 1-14, 2011.