A Study on Interesting Points Extraction and Background Points Estimation for Action Recognition in Videos

向, 宇

https://doi.org/10.15017/1807060

A Study on Interesting Points Extraction and Background Points Estimation for Action Recognition in Videos

February 2017

Yu Xiang

Abstract

There are many videos in the world at present and we need any tool to retrieve our required videos accurately. So, this research started on action recognition for videos because actions of articulated figures in videos can be used as a query for the retrieval. According to camera holding methods, like hand-held, the using of camera rocker, the using of camera stand, the videos majorly captured by static camera and moving camera. Though to realize action recognition in both cases, common key issue is to extract feature points, problem statements are different from each other due to the camera conditions. There are two kinds of sampling methods for action recognition in videos. One is the sparse sampling and the other is the dense sampling. Space-time interest points are the representative for the sparse sampling. They mean an important pixel area for action recognition. Currently, more and more videos are recorded by a static 4K camera. Moving foreground could be extracted accurately in these videos. In these cases, not all of the space-time interest points are needed for action recognition. So, this research considered the end points and joint points of skeletons of articulated figures in static 4K videos are the most important for action recognition. To extract end points and joint points from each video frame, this research proposes an integrated process and especially proposes new template-based algorithm for finding end points and joint points from a skeleton.

Although this integrated process can extract end points and joint points accurately in 4K videos under a static camera, it cannot work in most practical videos because such videos frequently include camera motions, pan, zoom and shake. Compared to the space-time interest points, the dense sampling is more effective for dealing with these cases. However, camera motions are harmful to acquire correct and effective sampling feature so that the influence of camera motions should be eliminated. Recently, dense point trajectory was proposed as the state-of-the-art research of the dense sampling approaches for practical video datasets. If point trajectories on background can be removed, the accuracy would become improved. So, this research focused on the estimation of point trajectories on background. To use the point trajectory for the action

recognition, this research employed the point trajectory based segmentation method. For extracting a segment of background point trajectories, this research proposes selection and optimization principles for that. From the qualitative evaluation on practical videos, it can be found that the proposed selection and optimization principles work effectively. Furthermore, this research proposes a method for eliminating the influence of camera motions based on the selected background points. From the quantitative evaluation on practical video datasets, it can be found that the proposed method improves the action recognition accuracy.

Acknowledgements

I am using this opportunity to express my gratitude to my supervisor, Professor Yoshihiro Okada who supported me throughout my PhD. I am thankful for his trust. During the course of my research, he always trusted my selection, from the research direction to the research plan. No matter how many times I failed, he always trusted my new revised plan and allowed me to try. I am thankful for his aspiring guidance. Each time when I propose a new revised plan, he always gave me some invaluably constructive criticism to illustrate the shortage and friendly advice to make the plan more feasible and significant. I am thankful for his supports for my research. He provided me his own two server machines. I could not complete my experiment if there are not these two machines. I am thankful for his patience. Because my language is bad, he always explained his advice again and again to ensure that I really understand. For helping me publish papers, he found and collected suitable conference for me and carefully modified my papers. The most important thing I want to thanks is his encouragement. Once I thought that I could not finish my research and cried, he encouraged me and helped me analyze what I can do in the following days. I am sincerely grateful to him during these three years.

I would also like to thank the members of the Okada Laboratory who provided me help in my research and life. Especially Associate Professor Kosuke Kaneko, he helped me in constructing Linux server for my experiment.

Finally, I would like to thank my parents for their encouragements to pursue my studies abroad. When my body feels bad, my brain feels headache and my teeth feels pain I worried that the sickness will cause more serious problem in the future, they always encourage me and gave me some guidance for my life.

Contents

Chapter 1: Introduction

Recent advances of digital camera technologies allow us to make still images and videos easily. Therefore, currently there are many still images and videos in the world and we need any tools that help us to retrieve our required still images or videos accurately and quickly. According to camera holding methods, like hand-held, the using of camera rocker, the using of camera stand, the videos majorly captured by static camera and moving camera. Though to realize action recognition in both cases, common key issue is to extract feature points, the moving camera will destroy correct feature point sampling in foreground and bring in ineffective and harmful feature point sampling in background. So the problem statements are different from each other due to the camera conditions.

For extracting most important interest point from the video frames recorded by a static 4K camera, we focused on the space-time interest point, the representative technique in sparse sampling action recognition. We found that not all the interest points obtained from the current popular space-time interest point algorithms are the most important interest points. Moreover, we considered that accurate extraction of moving articulated figures is possible from the video frames recorded by a static 4K camera.

In our research, we considered that end points and joint points are the most important points as the space-time interest point. Then, we propose a whole integrated process for extracting such interesting points, i.e., the extraction of a silhouette image with the chromacity-based shadow removal [1] by the background removal of ViBe [2], two-scan labeling [3] and morphology-based optimization, thin algorithm [4] [5] [6] to generate a skeleton from the silhouette, end points and joint points extraction and redundant points removal.

The background removal of ViBe based on the moving foreground segmentation under a static camera is one of the state-of-the-art methods. According to the results of several experiments, chromacity-based shadow removal is the best method for our research. To improve shadow removal results, we also employ two-scan labeling [3] and morphology-based optimization. The

two-scan labeling used in [3] is a very popular connected component-labeling algorithm employed by most researches. Thin algorithm is the most popular algorithm for human skeleton extraction from a silhouette image and generates a skeleton as single pixel connected lines that is very important for our following end points and joint points extraction.

For extracting end points and joint points, we propose a template-based method. Furthermore, for removing redundant end points and joint points, we also propose an optimization process that removes short branch based on the human topology structure. For reducing the running time of optimization process, we propose using the eight directions recursion traverse for finding topology of end and joint points.

Although our proposed integrated process could accurately extract end points and joint points of video frames, it could not deal with practical videos including camera motions, i.e., pan, zoom and shake. According to Wang et al. [7], the dense sampling outperforms all the tested space-time interest point for human actions in realistic settings. So, we focused on the dense sampling for action recognition in more practical videos. Currently, the most representative research about the dense sampling is the dense trajectory proposed by Wang et al. [8] [9] [10]. They used a state-of-the-art human detector [11] to ensure their dense sampling belongs to the foreground. However, the human detector still could not ensure all the human with all kinds of gestures so that it could not be used for other articulated objects. Since the camera motion will cause point trajectories on background, so we focused on extracting the background points and using these points to estimate background motion pattern.

By thresholding the minimum motion of point trajectories to remove those point trajectories for subtracting the background motion pattern from all the point trajectories, we could eliminate the influence caused by camera motions. In addition, considered that we adopted the dense point trajectory for motion recognition, instead of using those traditional foreground/background segmentation methods, we used the dense point trajectory based segmentation (clustering). Compared to those traditional foreground background segmentation methods, the point trajectories in each segment (cluster) by the dense point trajectory are based on similar point trajectory measure completely, not other measure. This is helpful for background motion pattern estimation.

Currently, the latest and state-of-the-art dense point trajectory based segmentation (clustering) is the long-term point trajectory analysis proposed by Keuper et al. [12]. Unfortunately, their algorithm just segments the video frame into different clusters without any considering background clusters. So, in this research, we propose three principles and two parameters to recognize which cluster is the possible background cluster. From the qualitative evaluation on practical videos, it was found that the proposed selection principles work effectively. In addition, the long-term point trajectory analysis could not ensure that all the points in background clusters belong to real background, which is also the problem for all the foreground/background segmentation algorithms. Considered that our target is just to estimate the background motion pattern, so we do not need to extract accurately all the background points. Based on this, we also propose two optimization principles to remove the doubtful points in the background cluster.

We use the background points detected above to estimate the background motion pattern. We could rectify point trajectory descriptors by the background motion pattern and remove redundant trajectories by removing background motion pattern consistent trajectories. Moreover, we use the bag of visual words [13] technique to cluster the rectified point trajectory descriptors and to generate video descriptors as final training and test samples. The SVM [14] is used to train the classification model. Three challenging datasets, the Hollywood2 dataset [15], the UCF50 dataset [16], and the Olympic sports dataset [17] are used for quantitatively testing the performance of our video descriptors. From results of the test, it was found that the proposed video descriptors with background motion estimation improve the action recognition accuracy.

The remainder of this dissertation is organized as follows: Next Chapter 2 treats end point and joint point extraction of moving human in 2D videos. In the first section, related works to this topic are introduced. The system framework and its components under the sparse sampling are introduced in the following section. After that, we propose the entire integrated process for extracting interesting points such as end points and joint points. Among of them are the extraction of a silhouette image with the chromacity-based shadow removal by the background removal of ViBe, two-scan labeling and morphology-based optimization, thin algorithm to generate a skeleton from the silhouette, end points and joint points extraction and redundant points removal,

respectively. Finally, we show experimental results qualitatively and conclude Chapter 2. Next Chapter 3 treats background points estimation for action recognition in practical video datasets. In the first section, related works to this topic are introduced. The system framework and comparison to original trajectory descriptor under the dense sampling are introduced in the following section. We explain background points detection under the dense sampling approach in the third section and show its qualitative evaluation results in the fourth section. After that, we explain action recognition by our rectified video descriptor and show its quantitative evaluation results in the fifth and sixth sections, respectively. Then, we conclude Chapter 3. Finally, in Chapter 4, we conclude this dissertation and mention future works.

Chapter 2: End Point and Joint Point Extraction of Moving Human in 2D Videos

Space-time interest points are popular features for action recognition in 2D videos. Such methods for action recognition consider the point importance on space and time according to certain criterions. Fig. 2-1 shows the results of space-time interest points. Currently, more and more videos are recorded by a 4K camera. For example, Fig. 2-2 shows the scenes of sport videos recorded by a 4K camera. When the camera is static and the resolutions of the videos are high, it is possible to segment more accurate foreground silhouette. Obviously, the interest points on foreground silhouette are more effective than the interest points in Fig. 2-1 for action recognition.

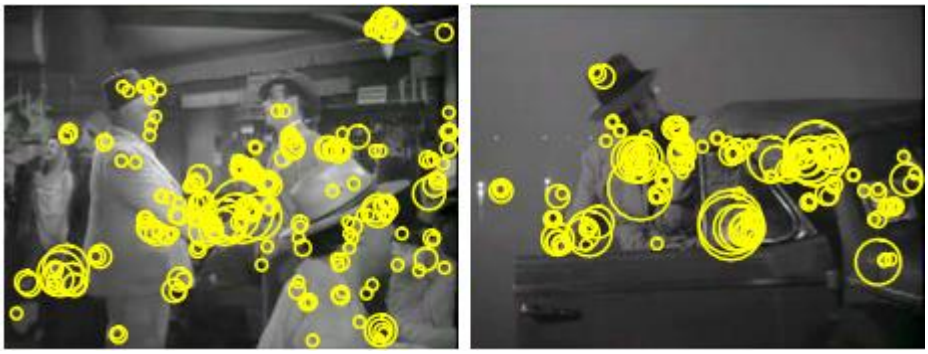


Figure 2-1. Interest points from [18].

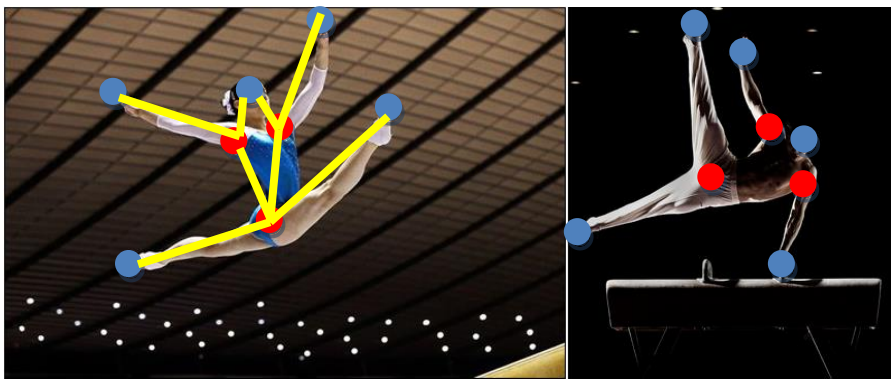


Figure 2-2. 4K video scenes from webpages [1] [2].

Moreover, we think that end points and joint points of the topology structure of a moving object are more effective interest points. It can be clearly said that the end points (blue points in Fig. 2-2) are the most important point for action recognition. We also think the joint points (red points in Fig. 2-2) are important points. First, the joint points could reflect the relative relationships for end points to joint points, joint points to joint points, as the yellow lines show. Intuitively these relationships are also useful for motion recognition. In addition, by analyzing the relative positions for joint points and the relative relationships, it is possible to recognize which end point is foot, which is hand and which is head. We thought that there are very important for action recognition. Finally, compared the joint points in Fig. 2-2 with the interest points in Fig. 2-1, it is obvious that joint points are more important than most other interest points. Therefore, we started on this research.

In this chapter, an effective interesting point extraction system is proposed. There are two primary contributions in the system:

1) We propose an entire integrated process for extracting such interesting points. Among of them are the extraction of a silhouette image with the chromacity-based shadow removal [1] by the background removal of ViBe [2] and morphology-based optimization, thin algorithm [3][4][5] to generate a skeleton from the silhouette, end points and joint points extraction and redundant points removal.

2) We also propose a template-based method for extracting end points and joint points.

This chapter is organized as follows. Section 2.1 introduces some related works and section 2.2 introduces the framework of our proposed system. Section 2.3 describes the segmentation of moving objects for background removal. Section 2.4 is the illustration of cast shadow, hole and noise dispose. Section 2.5 describes the skeleton extraction used in our research. We explain the details of end points and joint points extraction, and the redundant points removal in section 2.6 and section 2.7, respectively. Section 2.8 treats the experiment and analysis. Finally, in section 2.9, we conclude this chapter.

2.1 Related Work

The space-time interest points are the extension of space interest points [19] [20] and the scale invariant modification [21] for [20]. The concept of space-time interest points was first proposed by I. Laptev and T. Lindeberg [22] [23] that is derived from the space interest points [20]. Later, many kinds of space-time interest points have been proposed. The Cuboid interest point detector is based on temporal Gabor filters [24]. Wong and Cipolla [25] used global information from each video input to identify moving parts as selected relevant interest points. They illustrated the moving parts identified by their method are more important for action recognition. Hessian interest-point detector was proposed by Willems et al. [26] for blob detection in images. The detector measures the saliency with the determinant of the 3D Hessian matrix. The Chakraborty et al. [27] propose a novel approach for robust and selective STIP detection, by applying surround suppression combined with local and temporal constraints. As Fig. 2-1 shows, the extracted interest point from these methods still exist massive redundant useless interest points. The advantage of these methods is suitable for current most universal video. The disadvantage is that there are massive redundant points.

In this chapter, as mentioned above, we consider end points and joint points as the effective interest points for action recognition. There are several researches for extracting interesting points of moving objects from videos. Murat, et al. [28] introduces a silhouette based interesting point

Figure 2-3. Wrong end points.

Figure 2-4. Wrong skeleton with cast shadow.

extraction method. They segment a moving object as a silhouette based on a statistical background estimation that calculates the gravity of the silhouette and obtains a distance histogram from the center to each border point. They extract the local maximal points in the histogram and project them onto the silhouette. Their method is fast and effective if a human action comprises more stretch movements than bending movements. For example, as for the bending pose in Fig. 2-3, Murat's method will generate the wrong end points shown as the two red points.

In addition, since cast shadow [29] often appears in videos, this problem should be also considered. Otherwise, it will produce a wrong skeleton as shown in Fig. 2-4. For removing the bad influence of the shadow, we employ a chromaticity based shadow removal method [1], which will be introduced in section 2.4.1. Furthermore, instead of human template matching for the human shape detection, we use a perfect background subtraction algorithm ViBe [2] to generate the binary silhouette image of a human, and thin algorithm [4] [5] [6] to generate the human skeleton.

2.2 System Framework

Fig. 2-5 shows system components of our proposed method. The proposed system primarily consists of two parts; the morphology based pre-processing and interesting point extraction. The pre-processing part enhances the segmentation coming from the ViBe and chromaticity algorithms. This part is very important; otherwise, the skeleton extraction could not obtain an effective

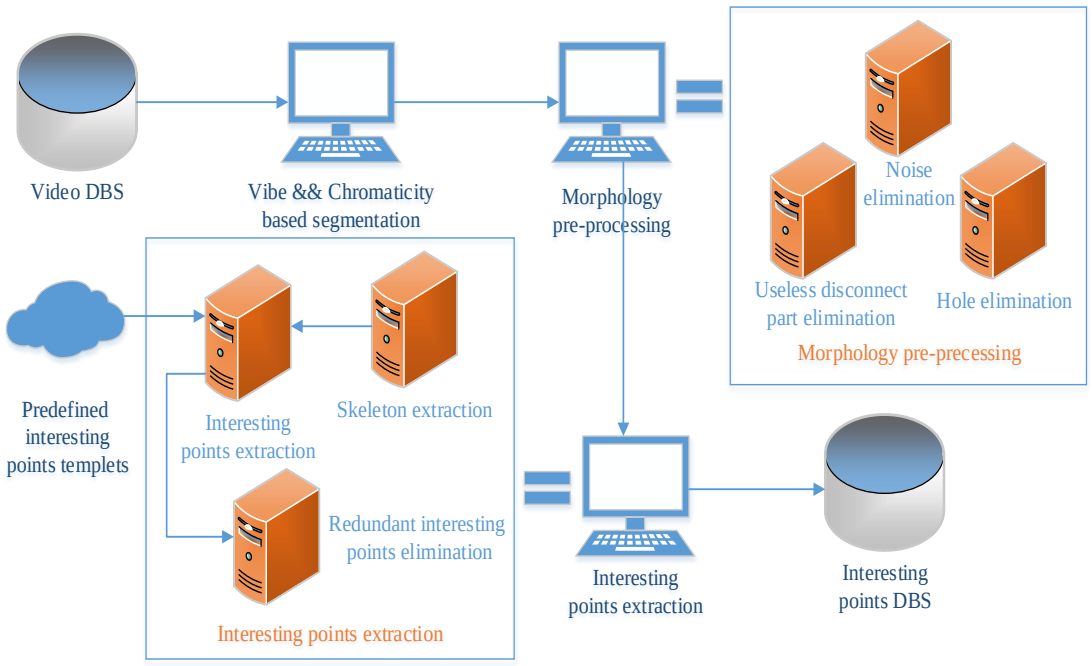


Figure 2-5. System framework diagrams.

skeleton for interesting points extraction. To the segmentation problems, three treatments have been considered. Noise elimination is carried out by removing the isolating points from the segmentation algorithm result. Disconnect parts elimination removes the segments and leaves the necessary segment. Holes elimination enhances the remaining segment. The interesting point extraction first obtains a skeleton from the remaining segment, and then extracts some interesting points from the skeleton according to the predefined templates.

2.3 Segmentation of Moving Objects for Background Removal

Because our research focused on the high-resolution video under a static camera for sparse sampling, we do not consider camera motions like pan, zoom and shake. The segmentation algorithms used widely are not suitable with us. These algorithms usually consider too many interference factors in motion foreground segmentation. Intuitively these algorithms usually run too slowly and are hard to achieve the segmentation result as good as background subtraction. As we knew, the best widely used motion segmentation algorithm [30] in 2014 could not obtain the

same good result as current popular background subtraction algorithms under the static camera and pure initial background. Although the background subtraction have many limitations, it could satisfy our basic requirement. In addition, the background subtraction usually runs quickly and needs little memory. As we have known so far, the best background subtraction algorithm is ViBe until 2014. The algorithm needs smaller memory and is more anti-noise than other background algorithms. The ViBe recognizes the problem of background subtraction as a classification problem. It wants to classify a new pixel value with respect to its immediate neighborhood in the chosen color space in order to avoid the effect of any outliers. For achieving this goal, it models each background pixel with a set of samples instead of with an explicit pixel model. The current value of the pixel is compared to its closest samples within the collection of samples. A new value will be compared to background samples and should be closer to some of the sample values rather than the majority of all values.

2.3.1 Pixel Model

As Fig. 2-6 shows, the algorithm denotes by $v(x)$ the value in an Euclidean color space taken by the pixel located at x in the image, and by v_i a background sample value with an index i. Each background pixel x is modeled by a collection of N background sample values taken in previous

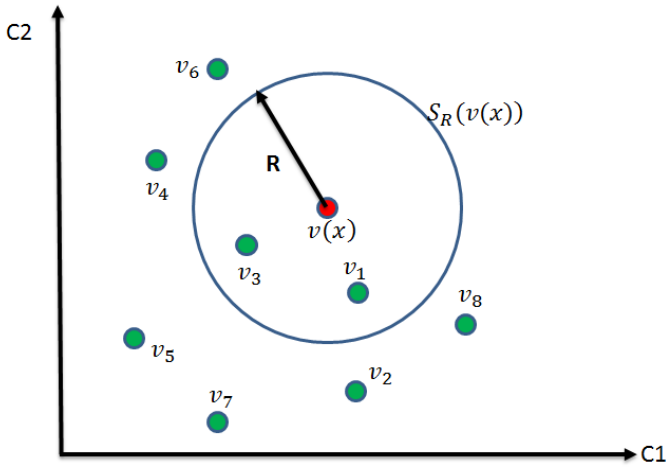


Figure 2-6. Comparison of a pixel value with a set of samples in two dimensions. Euclidean color space (C1, C2). To classify $v(x)$, count the number of samples of $M(x)$ intersecting the sphere of radius R centered on $v(x)$ from [2].

frames as Equation 2.1

$$M(x) = \{v_1, v_2, ..., v_N\}. \tag{2.1}$$

To classify a pixel value $v(x)$ according to its corresponding model $M(x)$, the algorithm compares it to the closest values within the set of samples by defining a sphere $S_R(v(x))$ of radius R centered on $v(x)$. The pixel value $v(x)$ is then classified as background if the cardinality, denoted by #, of the set intersection of this sphere and the collection of model samples $M(x)$ is larger than or equal to a given threshold $\#_{\min}$. More formally, it compare $\#_{\min}$ to

$$\#\{S_R(v(x)) \cap \{v_1, v_2, ..., v_N\}\}. \tag{2.2}$$

According to Equation 2.2, the classification of a pixel value $v(x)$ involves the computation of N distances between $v(x)$ and model samples, and of N comparison with a threshold Euclidean distance R.

2.3.2 Single Frame to Initialize Background Model

The algorithm selects samples randomly in the 8-connected neighborhood of each pixel. Formally, it assumes that $t = 0$ indexes the first frame and that $N_G(x)$ is a spatial neighborhood of a pixel location x, therefore

$$M^0(x) = \{v^0(y \mid y \in N_G(x))\}, \tag{2.3}$$

where locations y are chosen randomly according to a uniform law.

2.3.3 Background Model Update

A conservative update policy never includes a sample belonging to a foreground region in the background model. In practice, a pixel sample can be included in the background model only if it has been classified as a background sample. Such a policy seems, at first sight, to be the obvious choice. It actually guarantees a sharp detection of the moving objects gives that they do not share similar colors with the background. Unfortunately, it also leads to deadlock situations and

everlasting ghosts: a background sample incorrectly classified as foreground prevents its background pixel model from being updated. This can keep indefinitely the background pixel model from being updated and could cause a permanent misclassification. Unfortunately, many practical scenarios lead to such situations. For example, the location freed by a previously parked car cannot be included in the background model with a purely conservative update scheme, unless a dedicated update mechanism handles such situations.

Blind update is not sensitive to deadlocks: samples are added to the background model whether they have been classified as background or not. The principal drawback of this method is a poor detection of slow moving targets, which are progressively included in the background model. A possible solution consists of using pixel models of a large size, which cover long time windows. However, this comes at the price of both an increased memory usage and a higher computational cost. Furthermore, with a first-in first-out model update policy such as those employed in [31] or [32], 300 samples cover a time window of only 10 seconds (at 30 frames per second). A pixel covered by a slowly moving object for more than 10 seconds would still be included in the background model.

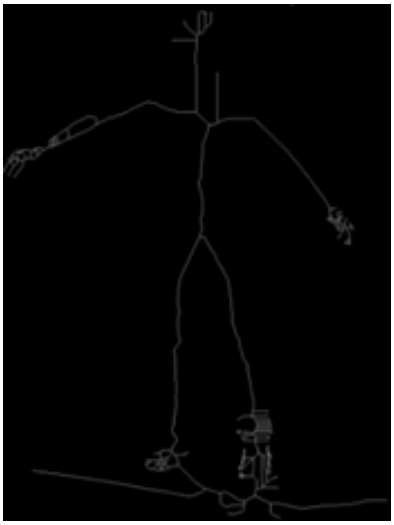
The ViBe algorithm adopts the update strategy combining the conservative update policy with what the authors of the W^4 algorithm [33] call a "detection support map". The "detection support map" counts the number of consecutive times that a pixel has been classified as foreground. If this number reaches a given threshold for a particular pixel location, the current pixel value at that location is inserted into the background model. So, it could overcome deadlock situations and everlasting ghosts occur in the traditional conservative update policy. Unlike many sample-based methods, which use first-in first-out policies to update their models, this algorithm employs a memoryless update policy, which ensures a smooth decaying lifespan for the samples stored in the background pixel models. It chooses the sample to be discarded randomly according to a uniform probability density function. The algorithm adopts a time subsampling strategy to update background model. It denotes a time subsampling factor ϕ. When a pixel is used to update the background model, the background model will be updated with a probability $1/\phi$. The strategy makes the background update less frequent, so that the slow moving targets are hardly included in

the background model. This solves the problem in Blind update. Furthermore, the ViBe algorithm not only updates the background model of a pixel x, it also updates the background model of a neighborhood of the pixel x. Formally, consider the 4- or 8-connected spatial neighborhood of a pixel x, that is $N_G(x)$, and assume that it has been decided to update the set of samples $M(x)$ by inserting $v(x)$. The algorithm also uses this value $v(x)$ to update the set of samples $M(y \in N_G(x))$ from one of the pixels in the neighborhood, chosen at random according to a uniform law. Through the background samples propagation, the algorithm makes spatial consistency. So, the background model could be spread outward ensuring the quick elimination of the ghost area.

2.4 Cast Shadow, Hole, and Noise

Although ViBe is very fast and generates good moving object segmentation, it still produces some harmful factors against the skeleton extraction. For example, these are holes in the right arm and the right foot, and rugged part in the left hand shown in Fig. 2-7 (b). In addition, the light disturbance will produce noise and the cast shadow. If we extract a skeleton from directly the segmentation result of ViBe, it will be not good as shown in Fig. 2-7 (c). Sanin et al. evaluate

(a) Video frame (b) Segmentation by ViBe



(c) Bad skeleton (d) Shadow removal

Figure 2-7. Initial segmentation and shadow removal.

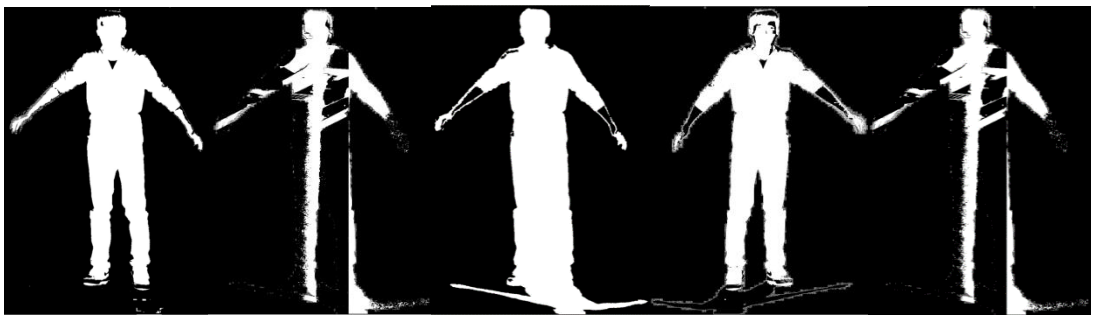


Figure 2-8. Results from left to right are chromacity-based method, the geometry-based method, the lr texture-based, the physical method, and the sr texture-based method shadow removal.

shadow detection methods in [1]. Primarily, there are five shadow detection methods, the chromacity-based method, the physical method, the geometry-based method, the SR texture-based method, and the LR texture-based method. Fig. 2-8 shows the results of these five shadow removal methods in our research. It is clarified that the chromacity-based method has the best result in our research, so we employ a chromaticity based shadow removal algorithm [1]. Fig. 2-7(d) shows the result of the algorithm after removing the cast shadow, but it maybe makes the hole problem more serious and causes disconnect parts. To solve this problem, we also apply a two-scan labeling [3] and morphology based algorithm as follows.

2.4.1 Chromaticity Based Shadow Removal

Several color spaces such as HSV [34], c1c2c3 [35] and normalized RGB [36] have proved to be robust for shadow detection [37]. The method used in our research choose the HSV proposed by Cucchiara et al. [34] because that color space provides a natural separation between chromaticity

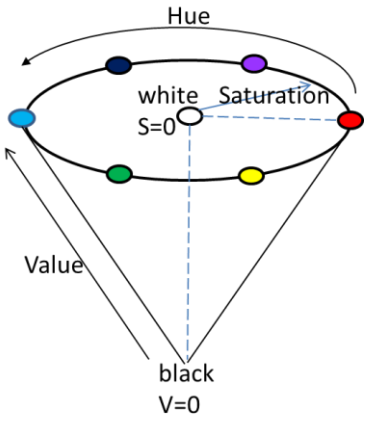


Figure 2-9. HSV color spaces.

and luminosity. The value (V) is the measure of intensity, so the values of pixels in the shadow should be lower than in the background. Considering the chromaticity, a cast shadow on background does not change its hue (H) and the authors of [34] noted that shadows often have lower saturation(s) than responding pixel. Formally, a pixel p is considered to belong to shadow part if it satisfies the following three conditions:

$$\begin{aligned} &\beta_1 \le (F_p^V / B_p^V) \le \beta_2 \\ &(F_p^S - B_p^S) \le \alpha_S \\ &| F_p^S - B_p^S | \le \alpha_H \end{aligned} \quad , \tag{2.4}$$

where F_p^C and B_p^C represent the component values, C, of HSV for the pixel position p in the frame (F) and in the background reference image (B), respectively. β_1, β_2, α_S and α_H represent thresholds that are optimized empirically. Working with alternative color spaces may produce different but not necessarily better results [37]. However, extending the pixel-level based analysis to an observation window improves results by countering pixel-level noise [35]. In our implementation of the HSV method, we used a 5-by-5 observation window rather than treating each pixel separately.

2.4.2 Two-Scan Labeling

As Fig. 2-10 shows, there are some redundant components existing in the segmentation result. We need to remove some small components and leave some large components. So, we need to label each connected components. Currently, the two-pass labeling algorithm is a very popular connected component-labeling algorithm. Most researches, which require connected components labeling employ this algorithm. Moreover, their results show that the two-scan labeling algorithm could label connected component very well. So in our research, we employ this algorithm [3] to complete the connected component labeling.

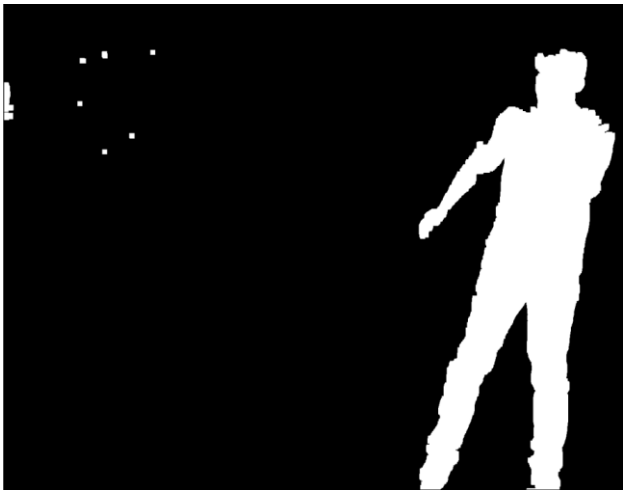


Figure 2-10. Redundant parts representation.

On the first scan:

1. Iterate through each element of the data by column, then by row (Raster Scanning)
2. If the element is not the background
    1. Get the neighboring elements of the current element
    2. If there are no neighbors, uniquely label the current element and continue
    3. Otherwise, find the neighbor with the smallest label and assign it to the current element
    4. Store the equivalence between neighboring labels

On the second scan:

1. Iterate through each element of the data by column, then by row
2. If the element is not the background
    1. Relabel the element with the lowest equivalent label

Here I use an example to illustrate the algorithm.

1. The array from which connected regions are to be extracted is given below (8-connectivity based). We first assign different binary values to elements in the array of Fig. 2-11 (a). Note that the "0~1" values written on the center of the elements in the array are elements' values. While, the "1,2,...,7" values in the next two arrays are the elements' labels. The two concepts should not be confused.

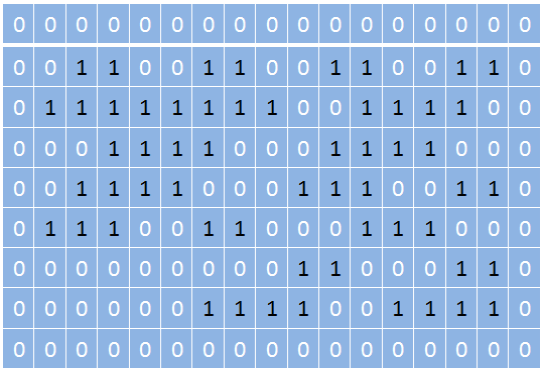


Figure 2-11 (a). Initial array.

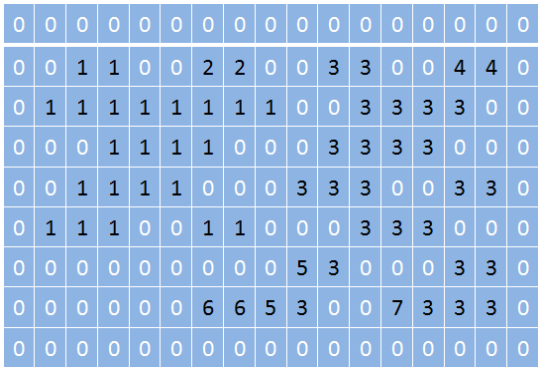


Figure 2-11 (b). The result after the first scan.

2. After the first scan, the following labels are generated. A total of 7 labels are generated in accordance with the conditions highlighted above. Fig. 2-11 (b) shows the result. Now the label equivalence relationships are generated as table 2-1.

Table 2-1 Equivalence relationships.

Set ID	1	2	3	4	5	6	7
Equivalent Labels	1,2	1,2	3,4,5,6,7	3,4,5,6,7	3,4,5,6,7	3,4,5,6,7	3,4,5,6,7

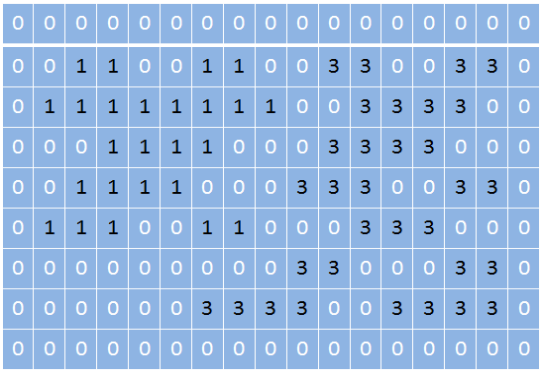


Figure 2-11 (c). The result after the second scan.

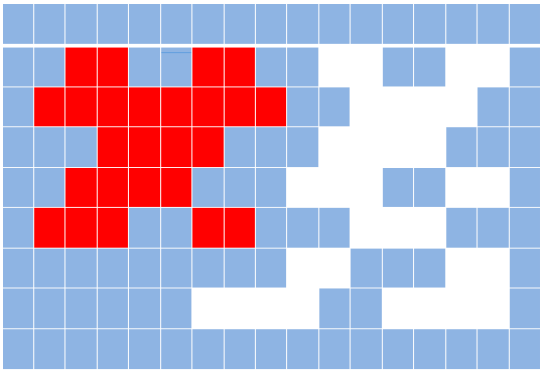


Figure 2-11 (d). Color result after second scan.

3. Array generated after the merging of labels is carried out. Here, the label value that was the smallest for a given region "floods" throughout the connected region and gives two distinct labels, and hence two distinct labels exist. Fig. 2-11 (c) shows the result.

4. Final result of Fig. 2-11 (d) shows two different regions in different color that have been found in the array.

2.4.3 Morphology

In Fig. 2-12, finally, we will obtain the cyan part as the moving object silhouette. There are some problems in the cyan part, such as holes, rough borders and so on. If we do not deal with these problems, the skeleton to be extracted and the extraction result of interesting points will become as shown in Fig. 2-13. It is obvious that these problems are the morphology problems and there is no special algorithm for treating these problems. However, we could solve these problems by flexibly applying morphology methods, such as dilation, erosion and flood. In the following, we give our morphology procedure for solving these problems.

Input: binary segmentation image I_{in}, 2 by 2 kernel M.

Output: improved segmentation image I_{out}.

$I_{dilate}=I \otimes M$, $I_{erode}=I_{dilate} \Theta M$.

Execute two-scan labeling algorithm to label each connection area with C_i,

$I_C=\{C_1, C_2...C_n\}$, C_n is the collection area with the pixels having the same label.

$I_{pmo} = \max(I_c)$, I_{flood} = do the flood on I_{pmo}, $I_{hole} = I_{flood}$ xor 1, $I_{out} = I_{hole}$ or I_{pmo}.

In the algorithm, $\otimes$ and Θ represent dilation and erosion, respectively. Pmo means the remaining segment. Max function represents the largest segment we just reserve. We execute the flood algorithm on the remaining segment. The two-scan labeling result is shown in Fig. 2-12, where cyan is the pmo. Flood finds the hole in pmo. Bitwise operations xor and or fill out the hole in pmo, or the hole will make the extracted skeleton worse, as shown in Fig. 2-13. Based on the

above algorithm, in most cases, we can eliminate most harmful elements like noise, error motion segment and so on, and improve the connectivity of the primary motion object.

Figure 2-12. Two-scan labeling representation.

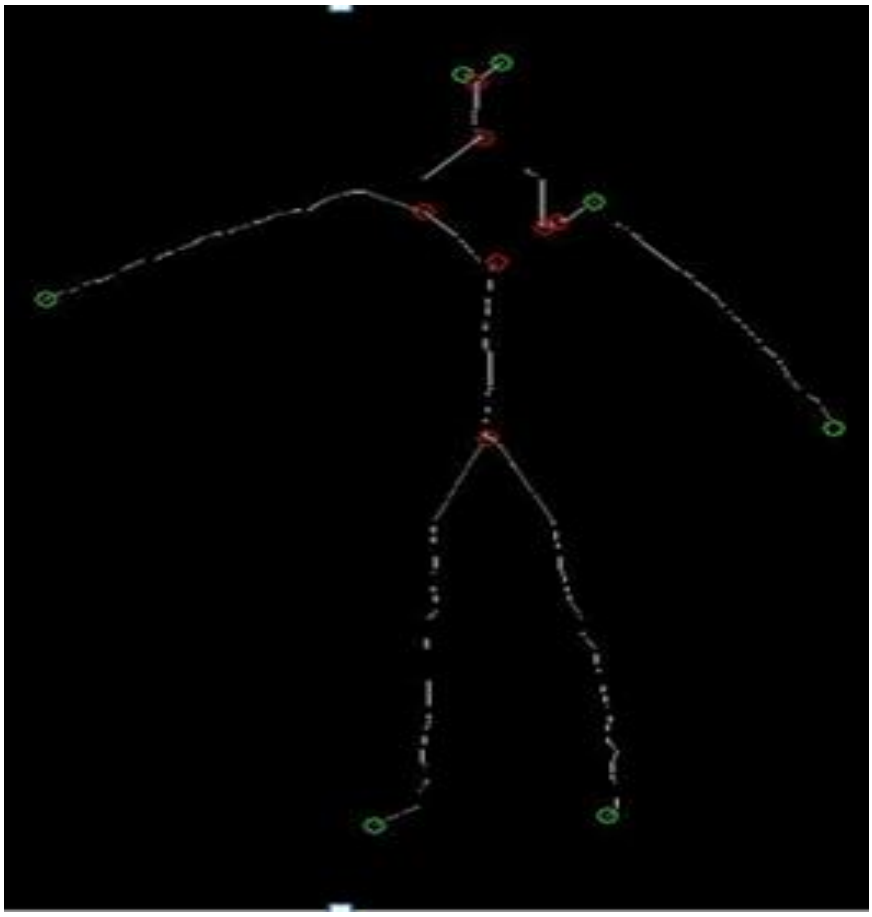
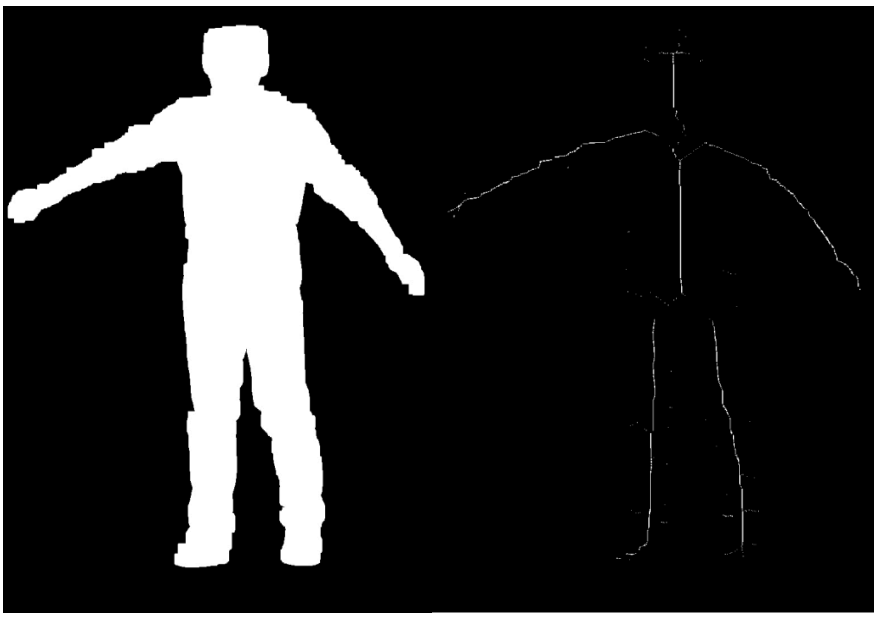


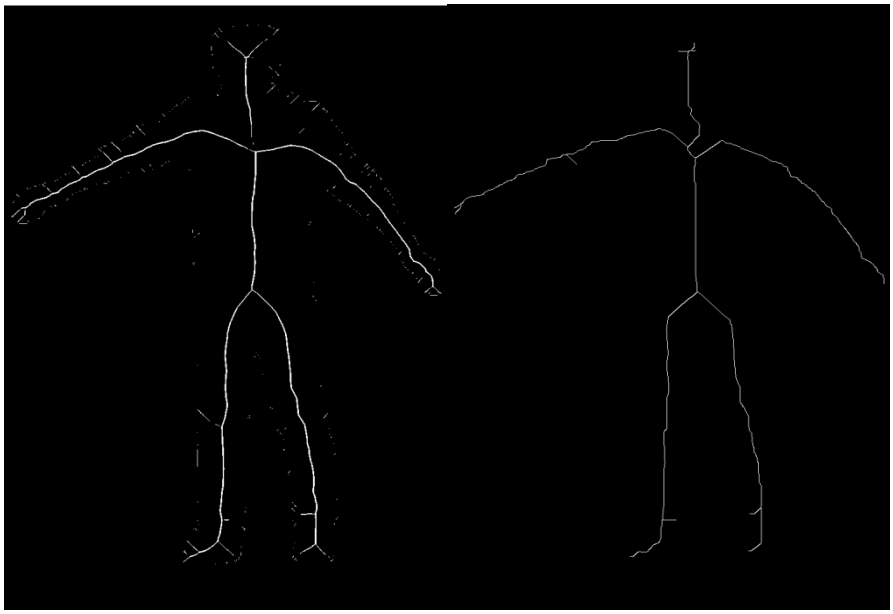
Figure 2-13. Wrong skeleton with hole.

2.5 Skeleton Extraction

There are many skeleton extraction methods for 2D videos. The classical skeleton extraction methods have achieved good result. The main problem for all the algorithms is the redundant branch. The Fig. 2-14 shows the results for some classical skeleton extraction methods. Sub image a is the improved segmentation, sub image b is the morphology based skeleton, sub image c is the



(a) Improved segmentation result. (b) Morphology based skeleton.



(c) Distance transform based skeleton. (d) Thin based skeleton.

Figure 2-14. Skeleton comparison.

distance transform based skeleton, sub image d is the thin based skeleton. Obviously, the morphology and distance transform based skeletons could not ensure the single pixel eight direction connectivity, the thin based skeleton could ensure this and this is necessary for the following end point and joint point extraction. In addition, the branches and the noises are fewest in the thin based skeleton methods. Considering the single pixel connectivity, the branches and the noises, thin algorithm is the best choice for us. In addition, for skeleton extraction in others' research, thin algorithm is also the most popular choice. In our research, we use the thin algorithm [4] to generate the skeleton for the segmented foreground above. Compared to the other classical skeleton extraction methods, the algorithm could ensure single-connected skeleton in eight directions. There are two sub-iterations in the algorithm for deciding whether the contour point P_1 should be deleted. Fig. 2-15 shows the two sub-iterations process, if the point P_1 satisfies any sub-iteration, it should be deleted. The $B(P_1) = P_2 + P_3 + P_4 + ... + P_8 + P_9$. $A(P_1)$ is the number of 01 patterns in the ordered set $P_2, P_3, P_4, ..., P_9$. Fig. 2-16 is an example of point P_1 and its neighbor and Fig. 2-17 shows the point set location illustration. Obviously, for the example, $A(P_1) = 2$, therefore P_1 is not deleted from the picture. Compared to the sub-iteration 1, in the sub-iteration 2, only the conditions 3 and 4 are changed.

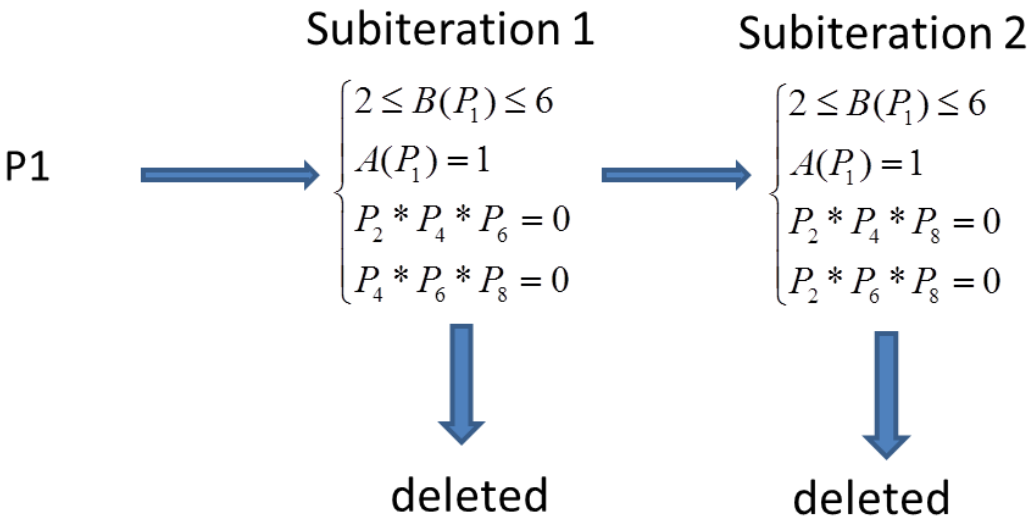


Figure 2-15. Two subiterations process.

0	0	1
1	P1	0
1	0	0

Figure 2-16. Example for point P_1.

P9	P2	P3
P8	P1	P4
P7	P6	P5

Figure 2-17. Point locations.

2.6 End Points and Joint Points Extraction

Our research extracts the end points and joint points from a skeleton. Geetha, et al. [38] introduce an end points extraction method. They design an adaptive template for detecting the human in an image. However, their template could not ensure that the bounding box contain an entire human when it is like a stretch poses. As well, they ignore joint points, which are also very important for the human action recognition. Most researches [39] [40] [41] [42] [43] extracting the end points and joint points from a skeleton do not introduce details for extracting process. They usually do not consider the redundant end point and joint point removal. The [42] just gave the end point detection. Our research gives the details for end point and joint point extraction, and redundant point removal. Furthermore, we propose an effective traverse method to speed up the redundant point removal. For finding interesting points from a skeleton, we use template definitions for end point, joint point on the 8 neighbors. Fig. 2-18 shows two examples of templates for end point definitions and four examples of templates for joint point definitions. Because we want to distinguish end points and joint points, so we need to recognize all the possible end point and joint point templates. In our research, we define 16 templates for end point and 159 templates for joint point. For compute could understand these templates, we use the binary system to indicate these

templates. First, we define the weight values of neighbors of P1 clockwise as shown in Fig. 2-17. $P_2, P_3, P_4, ..., P_9 = 2^0, 2^1, 2^2, ..., 2^7$. Fig. 2-19 shows their unique identifier. We classify the unique identifiers into end point set: {1, 3, …..} and joint point set {21, 37, 41, 168, ……}. To each pixel, by calculating the identifier value, we can label it as an end point or a joint point.

Figure 2-18 (a). End point.

Figure 2-18 (b). Joint point.

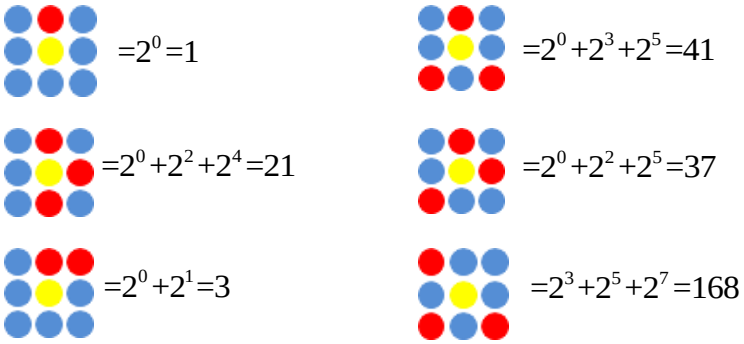


Figure 2-19. Unique identifier calculations.

2.7 Redundant Points Removal

Although the above interesting points extraction algorithm could obtain the basic end point and joint point sets as shown in Fig. 2-20, the two sets both include some redundant points. For example, end point 5 painted by a green circle is useless as an end point. For end point 3 and 4, any one is enough. In addition, after removing above end points, naturally joint points 1 and 5 painted by red circles are also useless. We observe that these invalid end points are usually the shortest branches among all the joint-end point branches, so sequentially removing the shortest joint-end point branches enables effectively eliminate the redundant points. For most human skeleton models, joint points 2, 3 and 4 painted by red circles, indicating the left, right shoulders

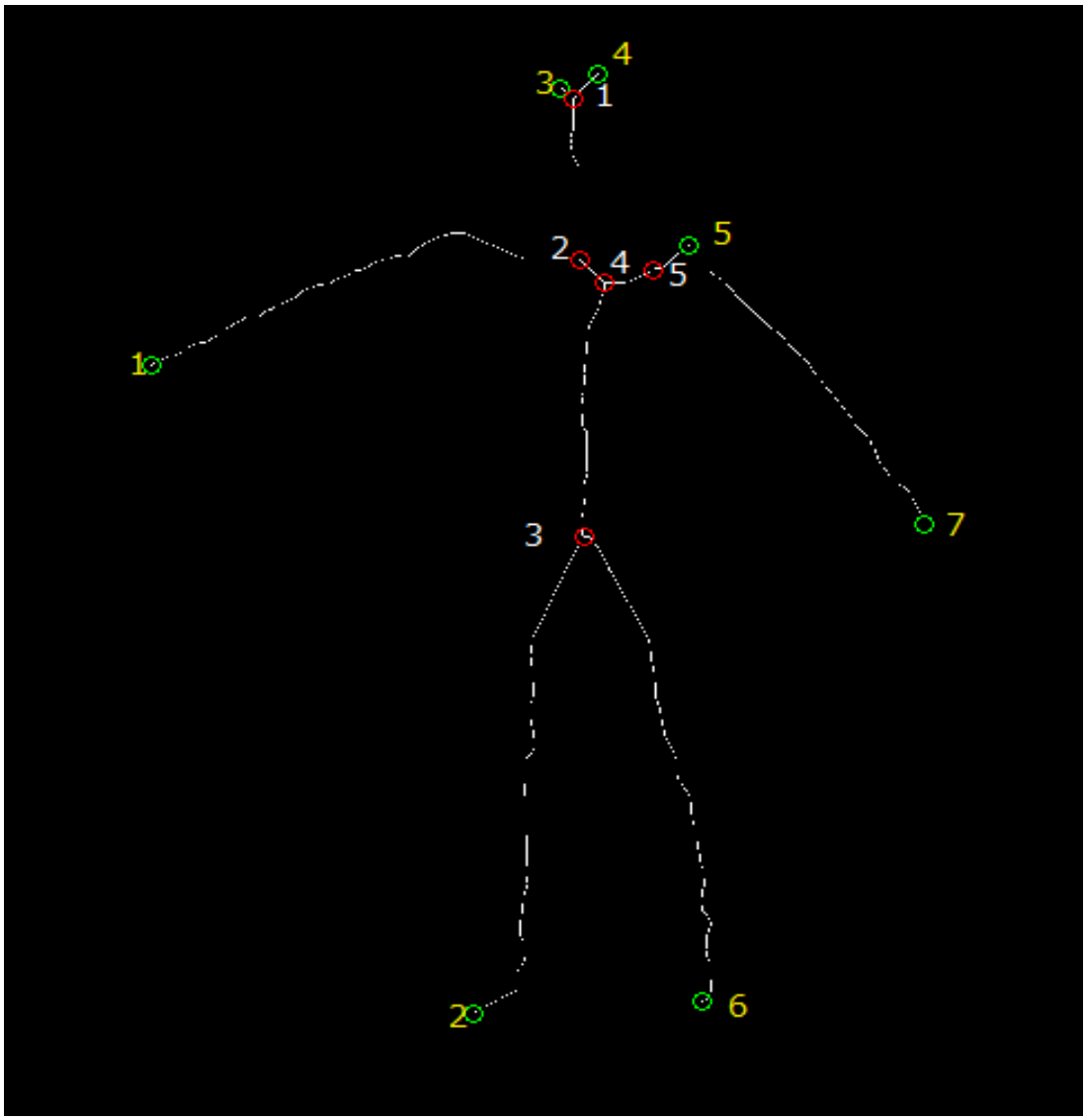


Figure 2-20. Interesting points result.

and hip, are key joints. Considering the principle for the termination condition, we propose a skeleton optimization algorithm as follows.

Input: binary skeleton image S, joint point set JP and end point set EP .

Output: the new JP and EP after pruning redundant points.

For each point i in JP do

Execute eight directions recursion traverse (EDRT) algorithm to find its directly connecting end point set $SE_i = \{(endpoint_1, len_1), (endpoint_2, len_2) ... (endpoint_n, len_n)\}$ and the joint point set $SJ_i = \{(jointpoint_1, len_1), (jointpoint_2, len_2) ... (jointpoint_n, len_n)\}$.

End

Define total end point set $TE = \{\cup SE_i, i \in JP\}$ and joint point set $TJ = \{\cup SJ_i, i \in JP\}$.

For num(JP)>3 && num(EP)>5 do

In TE, to the points having some same joint points, select the shortest branch and prune.

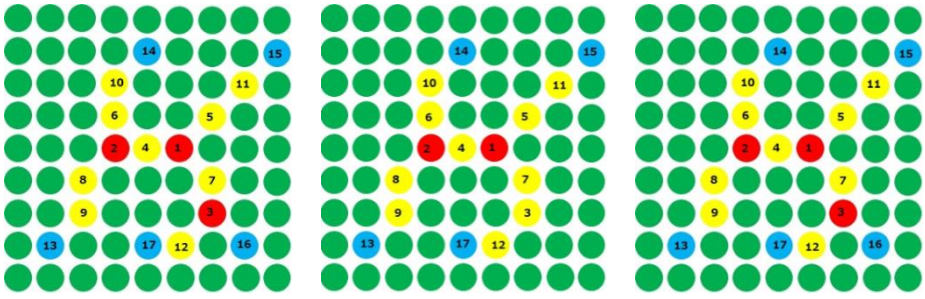
Remove the corresponding end point and joint point to the branch.

Update TE, TJ, JP and EP.

End

Return JP and EP.

In the algorithm, subscript i is the index of a point in joint point set JP. The subscript 1 to n indicate some directly connected points. They also come from the end point set and joint point set, respectively. Len is the length from joint point i to some end point or joint point. For example, in Fig. 2-21 (a), input sets are $JP = \{1,2,3\}$ and $EP = \{13,14,15,16,17\}$. For the points in JP, $SE_1 = \{(1,15,3)\}$, $SE_2 = \{(2,14,3),(2,13,3)\}$, $SE_3 = \{(3,16,1),(3,17,2)\}$, and $SJ_1 = \{(1,2,2),(1,3,2)\}$, $SJ_2 = \{(2,1,2)\}$, $SJ_3 = \{(3,1,2)\}$, $TE=\{SE_i, i \in JP\}$, $TJ=\{SJ_i, i \in JP\}$. In the TE, obviously the shortest length is 1, so point 16 should be removed. After removing the point 16, point 3 is not joint point now. Fig. 2-21 (b) shows the result of current $TE = \{(1,15,3),(1,17,4),(2,14,3),(2,13,3)\}$ and $TJ = \{(1,2,2),(2,1,2)\}$. Now, there are three branches owning the same shortest length 3, randomly remove one and update the TE, TJ, JP and EP, until satisfying the termination condition.



(a) (b) (c)

Figure 2-21. (a) Pruning example. (b) One step pruning result. (c) EDRT example.

In the optimization algorithm, search one direction at each time until its point becomes 0 value, or reaches an end point or a joint point, then clockwise search the other directions. The speed is

very low. So, we use an effective EDRT algorithm to make the whole optimization procedure faster. The algorithm is a parallel algorithm described as follows.

Input: joint point set JP , end point set EP, binary skeleton image S, binary visited image F.

Output: TE and JE .

For each point p_j in JP do

Push p_j into FIFO queue A

For each point p_k in A do

According to the binary visited image F, record the visited values of

eight neighbor points of p_k as B.

To the point with value 1 in S and value 0 in F, set its visited value in F as 1.

If finding an end point or a joint point among the eight neighbor points.

Recover the visited value of eight neighbor points of p_k in

F from B.

Else

Push the eight neighbor points into A.

End

End

Here, we show an example in Fig. 2-22 to explain this algorithm. For joint point 2, we hope to find the end points and joint points directly connecting to it. Fig. 2-22 shows FIFO queue A. In the first loop, we push the points satisfying conditions, i.e. point 4, 6, and 8 into the FIFO queue

4	6	8	10	9	13(end,3)
6	8	10	9	14(end,3)	
8	1(jun,2)				

Figure 2-22. EDRT FIFO example.

shown in the most left part and set them visited value as 1 in the image F. Then, pop 4 as a new tested point, we find that only point 1 satisfies conditions, but point 1 is a joint point, so we need not execute the EDRT on it. The distance between point 2 and point 1 on eight neighbors is 2 shown in the second most left part. Now we should process the second point in queue A, the point 6, obviously only point 10 satisfying conditions, so we add 10 into queue, then we process point 8, only point 9 satisfying conditions, so we add 9 into queue. Finally, we will find the direct connection end point 13 and joint point 1 for tested point 2.

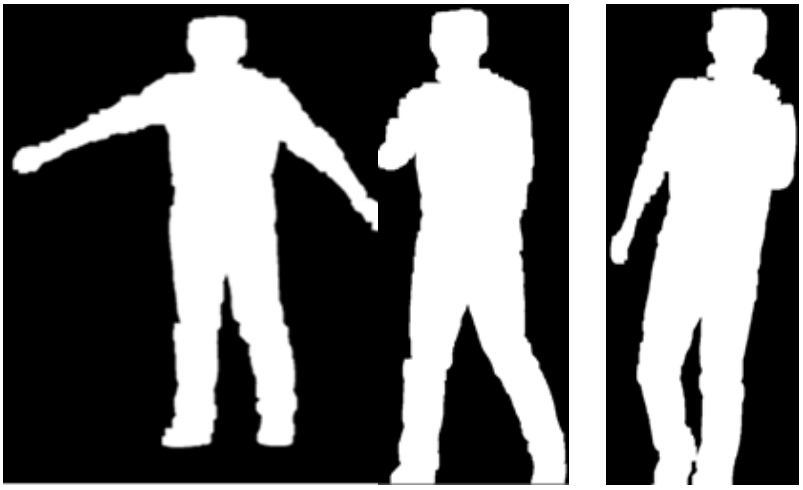
2.8 Experiment and Analysis

Fig. 2-23 shows some interesting points extraction results of video frames by our system. The video was taken by a canon digital video camera. The first row is the frames from the video. The shadow, motion blur and noise will influence the ViBe segmentation result, but through our enhancement, the segmentation will be improved greatly. Row 2 and 6 show the enhanced segmentation results. Row 3 and 7 are the interesting points extraction results according to the interesting point templates. Because the contours of segmentation results are not smooth, so there are redundant branches that cause meaningless interesting points when extracting the skeleton. Row 4 and 8 are the results after pruning. By comparing row 3, 7 and row 4, 8, we could clearly find that in most cases, the redundant points will be eliminated effectively, particularly to the end points. Because our pruning algorithm uses the key joint points and human topology as termination conditions, so the algorithm performance will not be influenced by the number of branches. The optimization results of column 3 prove it clearly. In most cases, the limbs of the human will be the longest branches, so eliminating a shortest branch each time is reasonable. From column 4, 5, and 6, we could find that although the skeleton branches connecting to the end points are not correct as a human topology, even in column 4, the joint point connecting to the left hand is not in a good position; the extracted end points are accurate. So even if an error skeleton has been generated due to the small angle at elbow, but it is still a high probability that the length from some human joints to the limb end points is longer than to a wrong branch end point. So the

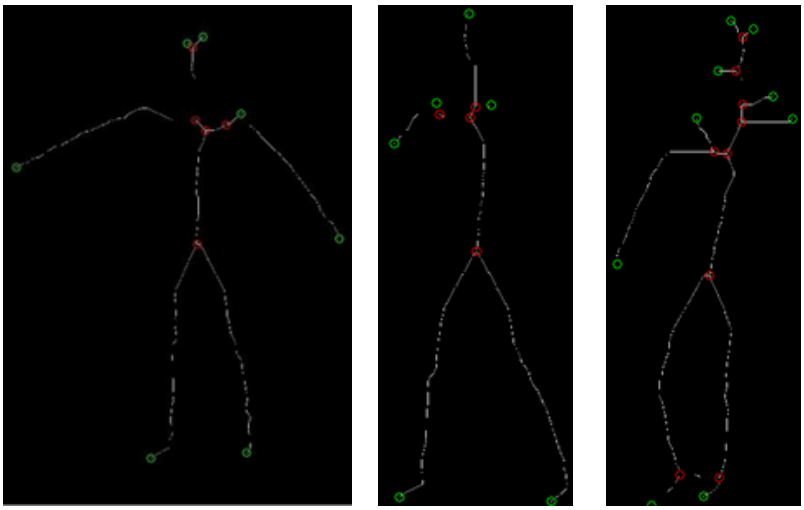
errors in Fig. 2-23 should be avoided greatly. For example, to an outstretched human body, as column 1 shows, the human joints connecting hands are the shoulder joint points. In this case, the end point at the hand could be recognized as a meaningful end point easily. But as for the examples in column 5 and 6, because the wrong skeleton for judging whether the end point at the

row1

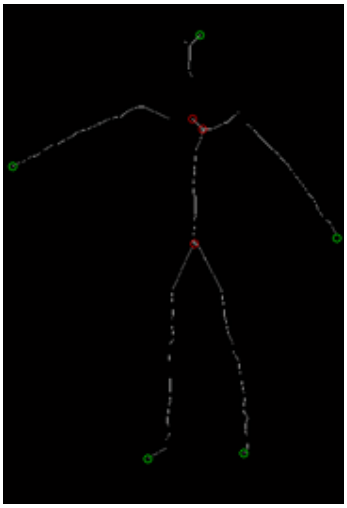
row2

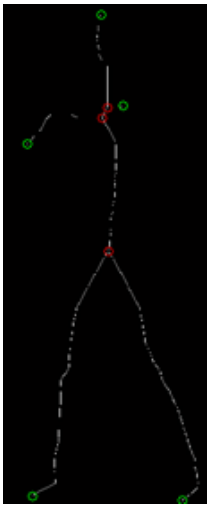


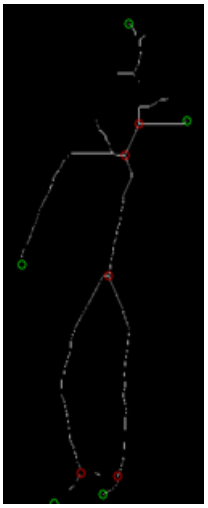
row3



row4



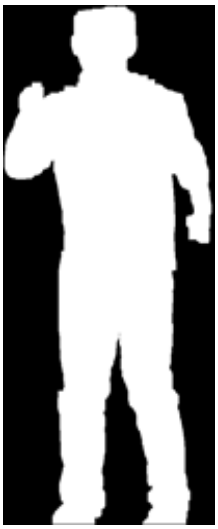




column 1 column 2 column 3

row5

row6



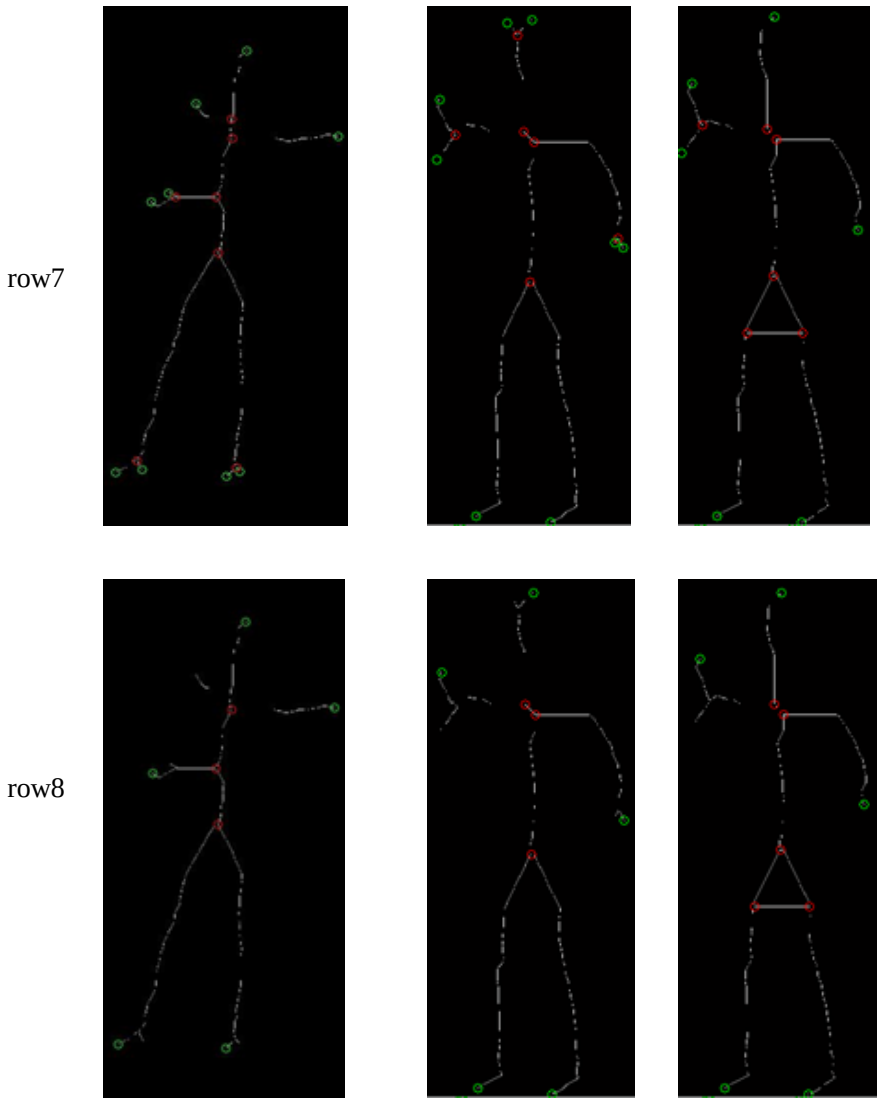


Figure. 2-23 video segmentations, interested points, and interested points optimization results. Row 1 to 4 is a group. Row 5 to 8 is a group.

hand is a meaningful end point, we need to compare with a wrong branch end point. But relying on the advantage in nature, the end point at the hand still has a high probability to beat wrong branch end point. In column 6 and column 3, there are some other wrong skeleton results, the closed loops between joint points, but the finally end point results show that our algorithm could overcome this error and extract the correct end points. This is because that instead of selecting the shortest joint-end point branch to prune each time, our algorithm will only select the branches with twin branch sharing the same joint. This simple but effective process frees us from the complex closed loops calculation. If you need, you can also easily find the closed loops based on the end

points result. Of course, there is some shortage in our algorithm, for example, there are some errors in the hands of column 2 and left hand of column 3. The errors are caused by fore and back occlusion of the body. Differently with 3D videos, there is no depth data in 2D videos, so some tracking method is necessary for solving this problem.

2.9 Conclusion

In this chapter, we proposed a human topology based interesting points extraction system for 2D video frames. Although the state-of-the-art background subtraction algorithm ViBe has an excellent foreground segmentation performance, it could not still satisfy the requirement of skeleton extraction. However, after our enhancement process, the segmentation is more suitable for skeleton extraction. We use two template types for extracting interesting points. However, due to some errors in extracted skeleton, redundant interesting points exist. We also propose an excellent human topology based pruning algorithm. The excellent EDRT algorithm used in the pruning algorithm realizes a semi-parallel traverse, not obtaining the correct relationships between a joint point and an end point, but having a high speed. The reasonable pruning theory and termination conditions ensure that even under some error skeleton cases, the algorithm could still avoid the errors arising in Fig. 2-3 easily and finally generate correct end points.

Of course, there are still some shortages of the proposed system, as it is incapable of dealing with the occlusion situation. Maybe adding some tracking module could overcome this shortage.

Chapter 3: Background Points Estimation in Practical Video Datasets for Action Recognition

Although we consider that the extraction of end points and joint points is possible by applying our approach in a high-resolution video under a static camera, it maybe does not work in most practical videos because such videos usually include camera motions like pan, zoom and shake. The space-time interest points sampling described in previous chapter could deal with the problem rather than our proposed end point and joint point extraction algorithm. However, according to Wang et al. [7], the dense sampling outperforms all the tested space-time interest point for human actions in realistic settings. This is the reason why we started on this dense sampling research. Camera motions like shake certainly generate ineffective motion trajectory on background. These motion trajectories will influence the sampling for correct motion trajectory. So, extracting background pixels and estimating the background motion pattern is necessary. Because we use the motion trajectory for motion recognition, we chose a point trajectory based segmentation proposed by Keuper et al. [12]. The method generates cluster for video frame. Only one cluster is background cluster, so we still need some principles to select the possible background cluster. In addition, not all the pixels in selected background cluster really belong to the background. So, we propose optimization principles to remove these pixels.

We use the extracted background points above to improve the dense trajectory proposed by Wang et al. [8] [9] [10] for action recognition. Combined with the extracted background points above, we could estimate the motion pattern caused by camera motions. According to the motion pattern, we can rectify the dense trajectory and obtain rectified trajectory descriptors. The short trajectories will be recognized as background point trajectories so removed. Then, we will also use the bag of visual words to generate video descriptors for action recognition in videos from the rectified trajectory descriptors. Finally, we test the performance of our improved video descriptors

on three challenging datasets, the Olympic sports dataset [17], the UCF50 [16], and the Hollywood2 [15]. The SVM [14] is used to train and test the action samples.

This chapter is organized as follows. Section 3.1 introduces some related works. Section 3.2 introduce system framework. Section 3.3 explains details of our accurate background points detection and section 3.4 shows its some experiments and results. In section 3.5, we explain our improved video descriptors for action recognition. Section 3.6 shows experimental results of the action recognition. Finally, we conclude this chapter in section 3.7.

3.1 Related Work

Most efforts have been devoted to the design of good feature representations in the literature for human action recognition. We mainly focus on the feature trajectory for motion recognition, which are more related to our work. In [44], Uemura et al. extracted trajectories of based on KLT tracker. The work of [45] computes the velocity histories of tracked keypoints by Birchfield's implementation of the KLT tracker for action recognition. Gaidon et al [46] and, Raptis and Soatto [47] proposed tracklet, which emphasizes more on the short trajectories. Later, Raptis et al. [48] extended [47] to a mid-level representation by grouping trajectories based on appearance and motion information, leading to a set of discriminative action parts, which are identified by the found trajectory clusters. Wang et al. [49] performed trajectory-based modeling using a nonparametric Bayesian models for action recognition. Wu et al. [50] proposed to decompose Lagrangian particle trajectories into their camera-include and object-include clusters for solving the moving camera problem in action recognition. Some other works also explored object-level trajectories [51] [52] for video content recognition. Currently, the most popular and effective trajectory-based motion modeling is proposed by Wang et al. [8] called the dense trajectory. They generated trajectories based on dense local patches and showed that the dense trajectories significantly outperform KLT tracking of sparse local features e.g., the SIFT patches. The state-of-the-art results have been seen on four datasets of varying difficulty. To remove the global motion, Wang et al. [9] used a state-of-the-art human detector [11] as a mask to remove feature matches

inside the bounding boxes when estimating the homography by RANSAC [11]. Even if they use the state-of-the-art human detector, some errors in the human detection still exist. Furthermore, obviously, although using the human detector could remove the trajectories on background, it also restricts the scope of application. So, we considered eliminating the influence of background motion trajectory caused by camera moving.

Currently, researches aiming at single background extraction are not so many for video datasets. Adeel Mumtaz et al. [53] propose a joint foreground-background mixture model (FBM) that simultaneously performs background estimation and motion segmentation in complex dynamic scenes. Berger et al. proposes an online method for background modeling of dynamic point trajectories via tracking of a linear subspace describing the background motion. To cope with variability in trajectory durations, we cast subspace tracking as an instance of subspace estimation under missing data, using a least-absolute deviation formulation to robustly estimate the background in the presence of arbitrary foreground motion. Matthew et al. [54] [55] relies on Bayesian selection of the transformation that best describes the geometric relation between consecutive frames. Based on the selected transformation, they propagate a set of learned background and foreground appearance models using a single or a series of homography transforms. Jongwoo Lim and Bohyung Han [56] propose an online background subtraction algorithm with superpixel-based density estimation for videos captured by a moving camera. The algorithm maintains appearance and motion models of foreground and background for each superpixel, computes foreground and background likelihoods for each pixel based on the models, and determines pixelwise labels using binary belief propagation. Mingliang Chen et al. [57] use the efficient minimum spanning tree to improve the robust estimators like M-smoother, and use the improved M-smoother to update the Gaussian mixture models at each pixel location. Additionally, they use the optical flow to track the foreground pixels. These tracked data are used to ensure temporally consistent background subtraction. Most methods update their background model by the information observed in past frames, but Shimada et al. [58] creatively proposed a bidirectional background modeling which the information used to update modeling not only came from past frames but from futures. Besides the background extraction in videos, most researches focus on the moving object extraction. There are also some researches for image dataset concern

the background detection. Some background detection algorithms are based on saliency. Zhu et al. [59] propose a robust background measure, called boundary connectivity. The conceptual definition is that object and background regions in natural images are quite different in their spatial layout. i.e., object regions are much less connected to image boundaries than background ones. Although the above methods focus on background subtraction, their goal is the segmentation for background and foreground semantic as much as possible. So, they do not guarantee all the pixel in their extracted background belongs to the real background. In addition, not all the above methods are based on motion trajectory. We consider that if the background pixels are extracted mainly based on motion trajectory, not semantics, they will be more helpful to estimate the background motion pattern. This is the main reason for us to start on the research of motion trajectory based background pixels extraction.

3.2 System Framework

Fig. 3-1 illustrates the process of our improved video descriptor generation system and represent the conventional video descriptor as comparison. Obviously, the big difference is that we provide a process to remove camera motion influence. The camera motion is the main problem causing the correct video descriptor estimation. In our research, we tried to eliminate the influence by estimating the background motion pattern. Considering that we used the trajectories as the motion feature for action recognition, although there are some other state-of-the-art approaches for video frame segmentation, such as [60], we used a trajectory based segmentation (clustering) algorithm, the long-term point trajectories analysis [12] based segmentation, to segment each video frame into different clusters. Since this segmentation algorithm is based on the motion trajectory similarity measure, not other measure, so all the pixels in the same cluster have similar motion pattern. This is helpful for background motion-pattern estimation. The [12] did not specify which cluster is the background cluster, so we propose a background cluster selection and optimization algorithm for obtaining accurate background pixels. These pixels were used for estimating background motion pattern. We rectified the trajectory descriptor and remove camera moving

consistent motion trajectories based on background motion pattern, and used the bag of visual words technique for generating our improved video descriptors. Finally, we use the SVM and three challenging datasets to test the performance of our improved video descriptors.

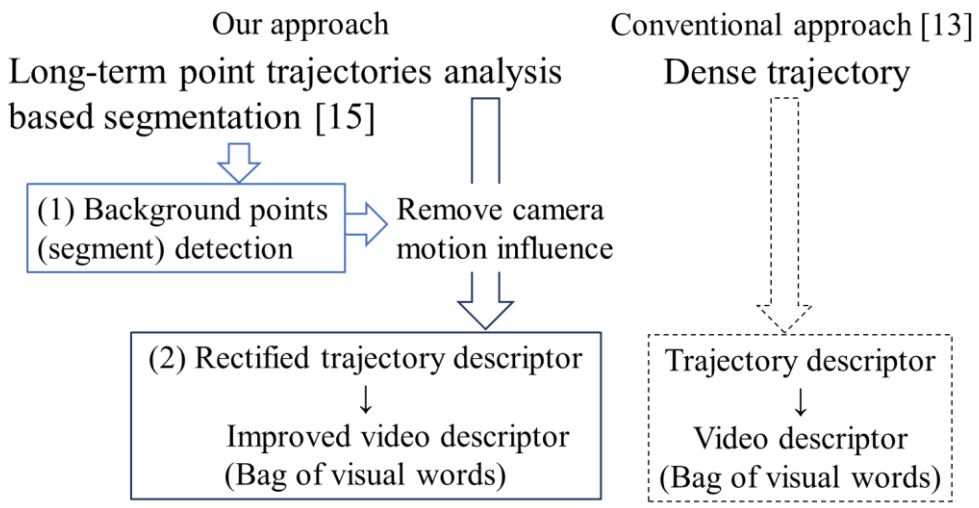


Figure 3-1. System framework for action recognition with background removal and comparison to conventional approach.

3.3 Background Points Detection

Motion segmentation is a very important part for analysis of digital videos. This is because the most important information used in analysis processing is related with foreground object. In some videos, although we could extract the foreground exactly, such videos are recorded by a static camera and in the high-resolution as what we treated in Chapter 2. Most practical videos are recorded by a moving camera and in the low-resolution. When we analyze the later videos, it is difficult to extract the foreground object exactly. So, directly analyzing the topology and motion pattern for foreground object is almost impossible. As mentioned above, in this situation, dense sampling has been proved an effective method. Nevertheless, background motion caused by camera motions still influences effective motion trajectory extraction. So, eliminating this influence is very important. Since video frames have the time continuity, the background points detection from a single frame is unreasonable. So, in our research, we consider completing the

segmentation according to the time continuity of videos. There are some spatio-temporal based segmentation methods, mainly classified into the superpixel based one or non-superpixel based one. The representative of super pixel based method is the "Spatio-Temporal Object Detection Proposals" proposed by Oneata et al. [60]. Although this segmentation could achieve a very good segmentation result, by considering our recognition is based on motion trajectory, we chose a motion trajectory based segmentation proposed by Keuper et al. . The motion trajectory based method could still achieve very good segmentation and this algorithm is newer than [60]. The most important thing is that since Keuper's segmentation is according to the similarity measure of motion trajectory so the pixels in each cluster in a final result have similar motion trajectory. This will be helpful to estimate consistent background motion.

Keuper's algorithm only segments each video frame into some clusters. So, we still need to recognize which cluster is the background. For this goal, we propose three principles and two important parameters for selecting the background cluster from the result clusters obtained by Keuper's algorithm. Be different with the background extraction algorithm mentioned in the related work, we optimize the selected background cluster for ensuring all the pixels in the background cluster is the real background pixel.

3.3.1 Point Trajectory Segmentation

The algorithm formulates the segmentation of a video sequence based on point trajectories as a minimum cost multi-cut problem. Unlike the commonly used spectral clustering formulation, the minimum cost multi-cut formulation gives natural rise to optimize not only for a cluster assignment but also for the number of clusters while allowing for varying cluster sizes. There are two important points for this segmentation. One is the large displacement optical flow [61] and its GPU-based parallel version [62]. Based on the large displacement optical flow tracking, the algorithm generates the long-term motion trajectory, the trajectory will be used to complete the motion segmentation by the minimum cost multi-cut method [12] [63] [64]. The other important point is the use of minimum cost multi-cut.

A. Large Displacement Optical Flow

Based on large displacement optical flow, all points showing some underlying structure are tracked until they are occluded. The occlusion reasoning is done based on the comparison of forward and backward optical flow. When trajectories get lost due to occlusions such that the sampling is sparser than the desired rate, new trajectories are started. The result is a set of reliable trajectories that start in some frame of the sequence and end in another. Depending on the data, many trajectories do not have any frames in common. However, the longer they are, the more valuable motion information they are expected to carry. The author of Horn-Schunck dense optical flow [65] creatively proposed connecting the two dimensional velocity fields with the gray. They bring in the optical flow constraints equation, which generates the basic algorithm for estimating optical flow. Since that paper, all kinds of optical flow is based on their model. The large displacement optical flow is also based on the model.

1. Horn-Schunck Model

Let $I_1, I_2 : (\Omega \subset R^2) \to R^d$ be the first and the second frame to be aligned, and So the image brightness at the point (x, y) in the image plane at time t be denoted by $I(x, y, t)$. When the pattern moves, there are two constraints about optical flow:

(i) the corresponding points should have the same gray value or color, in other word, the brightness of a particular point in the pattern is constant. Consider a patch of the brightness pattern that is moved a distance dx in x-direction and dy in y-direction in time dt. According to the constant principle:

$$I(x, y, t) = I(x + dx, y + dy, t + dt). \tag{3.1}$$

Expanding the $I(x + dx, y + dy, t + dt)$ about the point (x, y, t), the formula 3.1 is changed into formula 3.2:

$$I(x, y, t) = I(x, y, t) + \frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt + \alpha, \tag{3.2}$$

where the α contains second and higher order terms in dx, dy, and dt. So subtracting the $I(x, y, t)$ from both side of formula 3.2 and dividing by dt we could get the formula 3.3:

$$\frac{\partial I}{\partial x}\frac{dx}{dt}+\frac{\partial I}{\partial y}\frac{dy}{dt}+\frac{\partial I}{\partial t}+O(dt)=0\,, \tag{3.3}$$

where $O(dt)$ is a term of order dt. So in the limit as $O(dt)\to 0$ the formula 3.3 is changed into

$$\frac{\partial I}{\partial x}\frac{dx}{dt}+\frac{\partial I}{\partial y}\frac{dy}{dt}+\frac{\partial I}{\partial t}=0\,. \tag{3.4}$$

If we let $u=\frac{dx}{dt}$ and $v=\frac{dy}{dt}$, then we could see a single linear equation with two unknowns u and v:

$$I_x u+I_y v+I_t=0\,, \tag{3.5}$$

where the additional abbreviations I_x, I_y, and I_t represent the partial derivatives of image brightness according to x, y and t, respectively. Obviously, one equation with two unknowns could not obtain a certain solution. So, we only get the value of the movement in the direction of the brightness gradient (I_x, I_y), i.e. the "Normal flow". Fig. 3-2 illustrates the "Normal flow":

$$\frac{I_t}{\sqrt{I_x^2+I_y^2}}\,. \tag{3.6}$$

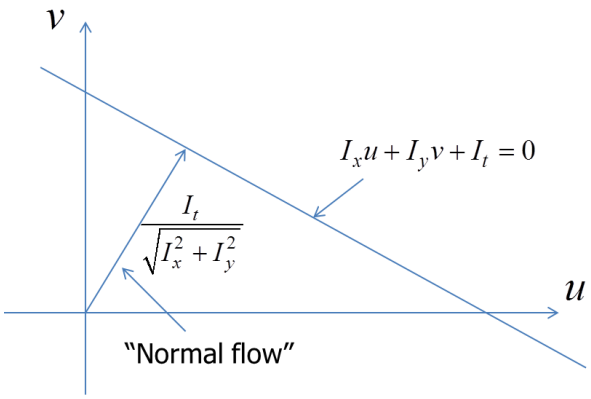


Figure 3-2. Normal flow.

So, we should add another constraint. Here, if every point of the brightness pattern can move independently, there is little hope of recovering the velocities. However, in practice, opaque objects of finite size undergoing rigid motion or deformation. In this case, neighboring points on the objects have similar velocities and the velocity field of the brightness patterns in the image varies smoothly almost everywhere. This constraint is called smoothness constraint or regularity

constraints. The constraints could be expressed by penalizing the square of the magnitude of the gradient of the optical flow velocity:

$$\begin{aligned} |\nabla u|^2 &= \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 \\ |\nabla v|^2 &= \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 \end{aligned} . \tag{3.7}$$

Images may be sampled on a square grid at regular intervals over frames. So, the measurement should be an average over the area of a picture cell and over the length of the time interval. In addition to quantize the sample in space and time, the measurement also quantized the brightness. This quantization will destroy the formula (3.5). i.e. the formula could not be zero. So, Instead of considering $I_x u + I_y v + I_t = 0$, minimizing its left part is a good choice. Let $E_b = \int_\Omega (I_x u + I_y v + I_t) d(x,y)^T$ and $E_{smooth} = \int_\Omega (|\nabla u|^2 + |\nabla v|^2) d(x,y)^T$, now the problem has been changed into minimizing the total error:

$$E(u,v) = E_b + E_{smooth} . \tag{3.8}$$

Until now, this is the Horn-Schunck model; we could solve the optical flow fields (u,v) by minimizing $E(u,v)$.

2. Model Improvement in Large Displacement Optical Flow

The Horn-Schunck model could not deal with the fast moving points very well, i.e. large displacement. Although the large displacement optical flow (LDOF) is based on the constraints proposed in the Horn-Schunck model, they improve the Horn-Schunck model so that the dense optical flow estimation has a good effective on fast moving videos. Let $\mathbf{x} := (\mathrm{x}, \mathrm{y})^T$ and $\mathbf{w} := (\mathrm{u}, \mathrm{v})^T$. Considering the occlusions and other non-Gaussian deviations of the matching criterion, the LDOF add a robust function $\psi(\mathrm{s}^2) = \sqrt{s^2 + \varepsilon^2}$, $\varepsilon = 0.001$. Now, E_b and E_{smooth} has been changed into:

$$\begin{aligned} E_b(\mathbf{w}) &= \int_\Omega \psi((I_2(\mathbf{x} + w(\mathbf{x})) - I_1(\mathbf{x}))d\mathbf{x} \\ E_{smooth}(\mathbf{w}) &= \int_\Omega \psi((|\nabla u(\mathbf{x})|^2 + |\nabla v(\mathbf{x})|^2))d\mathbf{x} \end{aligned} . \tag{3.9}$$

Considering the illumination effects, matching the color or gray value is not always reliable. So, they add a constraint on the gradient, which is invariant to additive brightness changes [66]:

$$E_{grad}(\mathbf{w}) = \int_{\Omega} \psi(|\nabla I_2(\mathbf{x}+\mathbf{w}(\mathbf{x}))^T - \nabla I_1(\mathbf{x})^T|^2) d\mathbf{x}. \quad (3.10)$$

Considering the drawbacks of descriptor matching (it is a discrete method that does not provide subpixel accuracy and the fixed spatial extent of richa descriptors is responsible for inaccuracies at motion discontinuities and in case of all non-translational motions.), they add another term which expresses the point correspondences from descriptor matching into the variation approach:

$$E_{match}(\mathbf{w}) = \int \delta(\mathbf{x})\rho(\mathbf{x})\psi(|\mathbf{w}(\mathbf{x}) - \mathbf{w}_1(\mathbf{x})|) d\mathbf{x}, \quad (3.11)$$

where $\mathbf{w}_1(\mathbf{x})$ is the correspondence vectors obtained by descriptor matching at some points $\mathbf{x}$. $\delta_i(\mathbf{x})$ is 1 if there is a descriptor available in frame 1 at point $\mathbf{x}$; otherwise, it is 0. The equation 3.11 assumes that the descriptors are already matched. The algorithm formulates the matching task by minimized another energy term:

$$E_{desc}(\mathbf{w}_1) = \int \delta(\mathbf{x}) |\mathbf{f}_2(\mathbf{x}+\mathbf{w}_1(\mathbf{x})) - \mathrm{f}_1(\mathbf{x})|^2 d\mathbf{x}, \quad (3.12)$$

where $\mathrm{f}_1(\mathbf{x})$ and $\mathrm{f}_2(\mathbf{x})$ denote the fields of feature vectors in frame 1 and frame2, respectively. So, now the problem is a single optimization problem of the whole model with all terms together:

$$E(\mathbf{w}) = E_b(\mathbf{w}) + \gamma E_{grad}(\mathbf{w}) + \alpha E_{smooth}(\mathbf{w}) + \beta E_{match}(\mathbf{w}, \mathbf{w}_1) + E_{desc}(\mathbf{w}_1), \quad (3.13)$$

where α, β and γ are tuning parameters which can be determined manually according to qualitative evidence on a large variety of videos, or be estimated automatically from ground truth data [67].

B. Minimum Cost Multi-Cut

The input consists of a weighted, undirected graph $G = (V, E)$ with a non-negative weight c_e for every edge $e \in E$, and a set of terminal pairs $\{(\mathrm{s}_1, \mathrm{t}_1), (\mathrm{s}_2, \mathrm{t}_2), ..., (\mathrm{s}_k, \mathrm{t}_k)\}$. A multi-cut is a set of edges that are disconnected to each of the terminal pairs. Formally, a set $E' \subseteq E$ is a multi-cut if for all $i = 1, 2, ..., k$, there is no path between s_i and t_i in the graph $(V, E \setminus E')$. The cost of a multi-

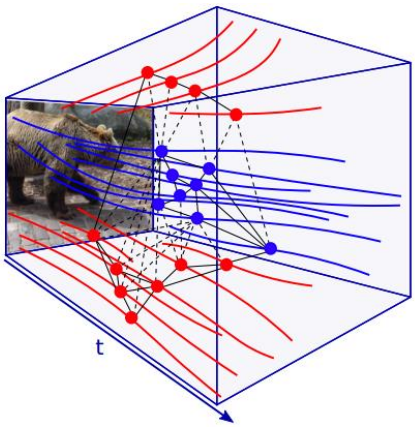


Figure 3-3. Long-term point trajectories represented by nodes in the graph $G = (V, E)$ those are connected with edges $e \in E$ to trajectories with some temporal overlap but can also be connected over time. Here, a segment can be represented as ether a node labeling (displayed in colors) or a consistent edge labeling (solid line $\cong 0$, dashed line $\cong 1$) from [12].

cut E' is given by $\cos t(E') = \sum_{e \in E'} c_e$. Thus, the minimum multi-cut problem (or multi-cut for short) is to find a multi-cut E' minimizes $\cos t(E')$. To formulate grouping as minimum cost multi-cut problem, Keuper et al. build a graph G such that every point trajectory is represented by a vertex $v \in V$ as shown in Fig. 3-3. If every vertex is connected by an edge $e \in E$ to its nearest neighbors, all solutions to the MC problem yield a segment into connected components. The weights of the edges $e \in E$ define how two trajectories are similar or dissimilar.

3.3.2 Background Recognition

The motion trajectory based segmentation clusters those pixels into different components. Now, what we need to do is to distinguish which component is the background. As mentioned above, we select the background based on the nature of background in real videos. We introduce the details of the selection procedure here.

Fig. 3-4 depicts the framework of our proposed whole procedure, which mainly includes two parts, i.e., background selection and background optimization. As mentioned above, The algorithm could not ensure all the pixels in the selected background component belong to the real background. So, after selecting the background component, optimization is necessary. The

optimization could remove some points that not belong to the real background. In the following, we will introduce the details of background extraction and background optimization.

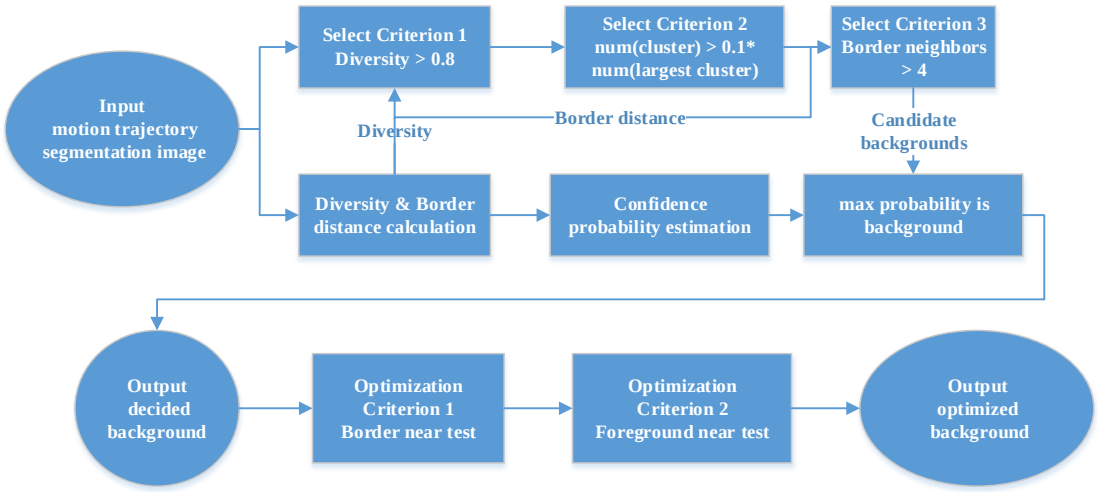


Figure 3-4. Framework of background extraction and optimization.

A. Background Extraction

The background extraction mainly includes two judgement parameters and three candidate background-judgement criterions. One parameter is diversity. This parameter is used to evaluate the concentrative degree of some components. Evidently, the background component is discrete and the foreground component is concentrative. In this research, we propose a method quantify the degree. The large number means discrete and small number means concentrative. The diversity will be used in criterion 1 and in background confidence-probability estimation. Another parameter is border distance. This parameter evaluates far and near degree from components to borders. This parameter is also used in confidence probability estimation, and some middle results will be used in criterion 3. We will introduce the details of diversity and border distance, and those criterions relative to them.

1. Diversity

The idea of diversity comes from the definition of color spatial distribution (CSD) [68]. However, there are some differences between diversity and CSD due to the different research target and objective. Firstly, although their research target is a single image, our research processes video frames. Secondly, although their research objective is to find salient parts, which are usually foreground in videos, our research objective is to select the background component. Thirdly,

before the processing, each pixel in a video frame has been classified into an explicit component indicated by one human-defined color, but in their research, these pixels just have probability calculated by Gaussian Mixture Models (GMM) to each color really existing in an image. These differences make the different calculation procedure. We define the diversity as:

$$Div(C) = Div_h(C) + Div_v(C), \tag{3.14}$$

where $Div_h(C)$ and $Div_v(C)$ are respectively the diversity of pixel x in horizontal direction and vertical direction. The definition of horizontal direction is:

$$Div_h(C) = \frac{1}{|X|_C} \sum_x p(C \mid I_x) \mid x_h - M_h(C) \mid^2, \tag{3.15}$$

$$M_h(C) = \frac{1}{|X|_C} \sum_x p(C \mid I_x) x_h. \tag{3.16}$$

Here, x_h is the x-coordinate of the pixel x, and $|X|_C = \sum_x p(C \mid I_x)$. $p(C \mid I_x)$ is the probability of each pixel x belonging to a certain component C. Instead of using Gaussian Mixture Models (GMM) to estimate the probability of each pixel belonging to the component, we use the motion trajectory analysis described in section 3.3.1 to cluster pixels into some components for a video frame and indicate each component with one color. So, for each pixel, the probability belonging to a certain component is just 0 or 1. In addition, as mentioned above, we focus on the background selection, not foreground. So, the existence of pixels far from the component center is important evidence for the background. However, due to the error during the motion trajectory segmentation, there will be some pixels in the component are outliers, that is, although some pixels are far from the component center, they do not belong to this component. So, instead of perceptual judgement, we need a quantization for the diversity. Because the existence of outliers, instead of using all the pixels in the components, we just use the partial pixels with the highest diversity. This partial processing is proved important; otherwise, some foreground components will have higher diversity than background because those pixels in the background near from the component center will decrease the diversity. According to these analyses, the diversity of a certain component is:

$$Div_{w\%}(C) = w\%(Div(C)), \tag{3.17}$$

where w% represents the pixels having w% largest diversity in component C. Finally, the diversity of C is:

$$Div_{normal}(C) = \frac{Div_{w\%}(C) - \min(Div)}{\max(Div) - \min(Div)}, \quad (3.18)$$

where Div_{normal} means the diversity set of all the components. Diversities are normalized into [0, 1] for all the components before combining them in Equation (3.21). As mentioned above, the parameter diversity will be used in the candidate background-selection criterion. Obviously, in most cases the pixels in the background component distribute in the whole frame and have the large diversity. So, we use the parameter to do the initial selection. If the diversity of some components is larger than some thresholds, we classify the components into background candidates. This is the criterion 1.

2. Border Distance

Generally, only the background has the large diversity. In these cases, the diversity could be enough to distinguish the background component from all the components. However, we could see that not only the background has the large diversity in the motion trajectory segmentation [12] for real videos having complicated scenes and sometimes there are another one or two components have the large diversity. So besides the diversity, another parameter is necessary. Compared to the foreground, background usually has more pixels near to the frame borders. So, we consider the use of average distances from these pixels to borders as another judgement parameter. We respectively define distances from a point to a top border, a bottom border, a left border and a right border as P_t, P_d, P_l, P_r. The border distance of a point is the minimum distance among these four distances, $P_{dis} = \min\{P_t, P_d, P_l, P_r\}$. Similarly to the diversity estimation, not all the points are effective and only pixels away from the gravity of some component are effective. So, we estimate the variance for these points' border distance as the border distance of the component. The equation of border distance is:

$$Dis(C) = \underset{w\%C}{Var}(P_{dis}), \quad (3.19)$$

where w% represents the farthest points what we extract away from the gravity of a component. Similarly to the parameter diversity, border distances are also normalized into [0, 1] for all the components before combining them in Equation (3.20):

$$Dis(C)_{normal} = (Dis(C) - Dis_{\min}) / (Dis_{\max} - Dis_{\min}). \quad (3.20)$$

The disadvantage of this procedure is that in some situations, some foreground components also have very small border distances, for example, shots of an object moves into or moves out. However, in these situations, the regions they occupy are small. Besides the reason caused by these actual situations, over-segmentation also happens in the motion trajectory segmentation. The over-segmentation often happens in foreground components, and even if sometimes happens in background components, it generally segments into very small parts from whole background components. It barely influences the recognition to the rest background component. For treating these actual cases, we consider that the very small component is not the background component. Considering the videos with different resolution, we could not use an absolute quantity as the definition of a small component. For each video frame, we find the largest component, and consider those components smaller than the P% of largest component not to be background. This is the criterion 2. Another easily thought thing is that at that moment, the moving objects are just near to minor borders. We divide four borders into eight parts from the middle of horizontal and vertical direction.

As Fig. 3-5 shows, the four borders have been divided into eight parts, a, b, c, d, e, f, g, h. Through the experiments, we find that the eight parts are enough, and more parts cannot improve the accuracy of background points extracting greatly. When we calculate the distance P_{dis} for every pixel, we also record which part is nearest to the pixel. So, to any component we could

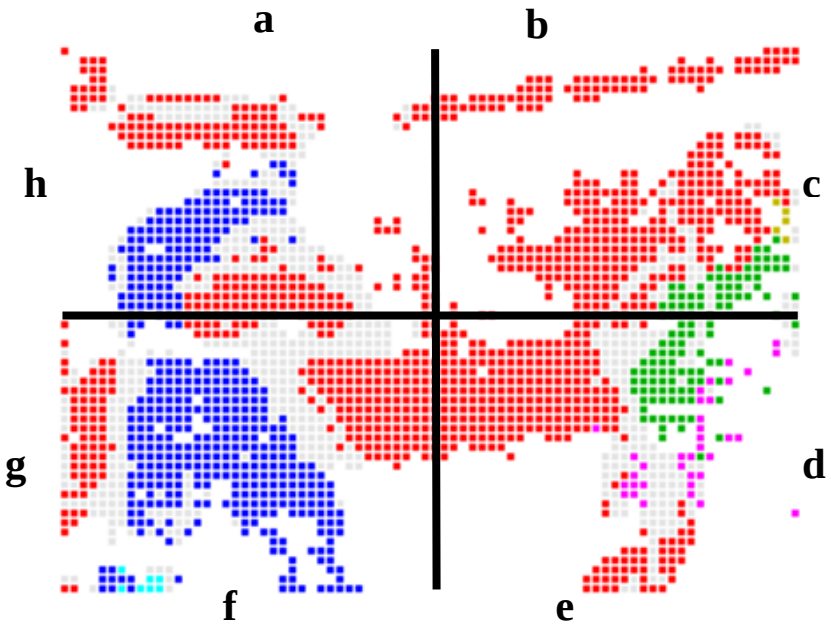


Figure 3-5. Eight border representation.

know how many parts occur in our record. If the number of recorded parts is larger than Q, we consider that this component is background. This is the criterion 3.

3. Selection

Until now, we have introduced two selection parameters and the relevant principles to these two parameters for selecting the candidate backgrounds. Now, we need to use these two parameters to determine finally background from candidate backgrounds. We combine two parameters as Equation (3.21):

$$P_{select} = Weight_{dis} * (1 - Dis(C)_{normal}) + Weight_{div} * Div(C)_{normal}, \quad (3.21)$$

where the $Weight_{dis}$ is the weight of a border distance and the $Weight_{div}$ is the weight of diversity. Because large diversity value could indicate background in the most situations, the border distance only is the auxiliary judgement parameter. In addition, the large diversity and a small border distance mean high probability being background, so we need to subtract the border distance by 1. The finally selected background is the candidate background with the max P_{select}.

B. Background Optimization

As mentioned above, our research objective is to select the background as accurate as possible not as most as possible. In other word, we hope the points in selected background components really belong to this component. Fig. 3-6 shows segmentation results. The image in right bottom is result image. Each color indicates different cluster. Based on above principles, we could know the red cluster is the background cluster. Obviously, the cluster occupies some foreground parts, so an optimization is necessary. Considering our research target, we introduce a simple and effective algorithm. There are two primary optimization principles:

1: The points near to border have high probability being background points.

2: The points near to foreground pixels have high probability being foreground points.

According to the two principles, we propose the algorithm as follows:

Input: point p in the selected background component, a frame width and a frame height.

(1) Generate the Border Select Box with horizontal gap as Hgap, vertical gap as Vgap as Fig. 3-7 shows. For point p,

If point p is out of Border Select Box

point p is the background point.

else

(2) Generate the Point Judgement Box centered on p, with width as Rwidth, height as Rheight.

if there are no foreground points in Point Judgement Box of p

point p belongs to the background component

else

p probably does not belong to the background component. So, remove this point from the selected background component.

Fig. 3-7 illustrates the optimization processing. In this frame, the point C is out of the border select box, so it could be held in the background component, the red component. There is no foreground point in Point Judgement Box of point A, so point A could be held in the background component. There are pink foreground points in Point Judgement Box of point B, so point B should be removed from the background component.

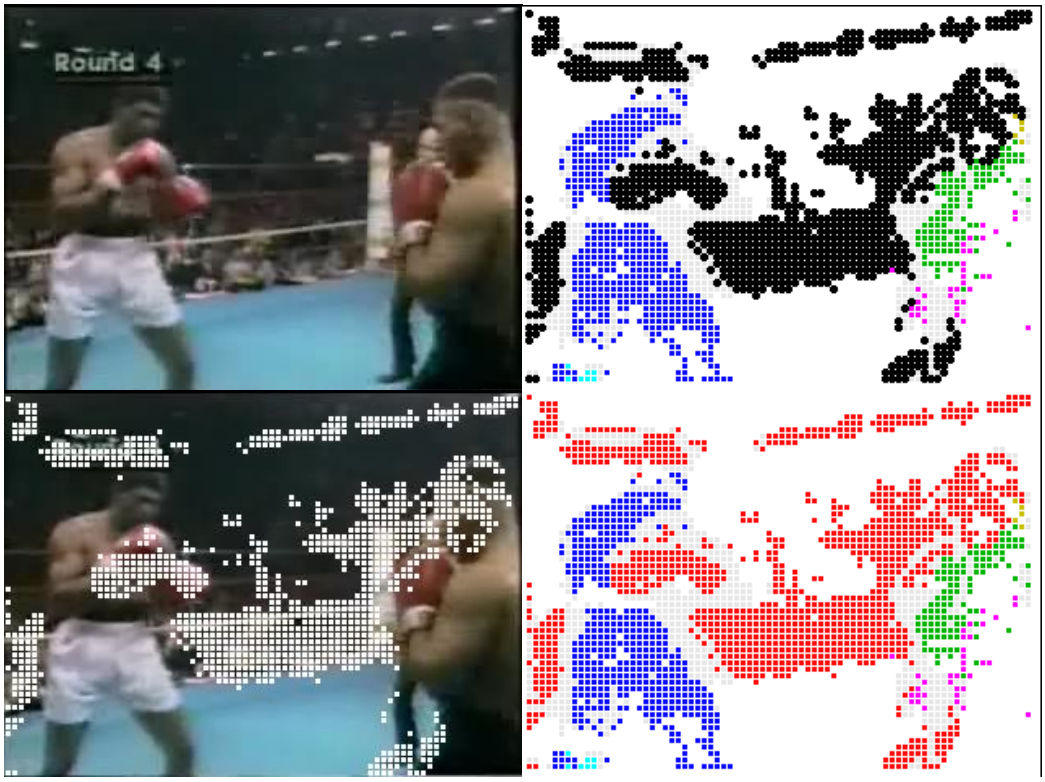


Figure 3-6. The fourth image is the motion trajectory segmentation result. Each color indicates a different cluster. Obviously, the red component occupies the human part. So, if we do not employ another optimization, the background we extract will cover the human part.

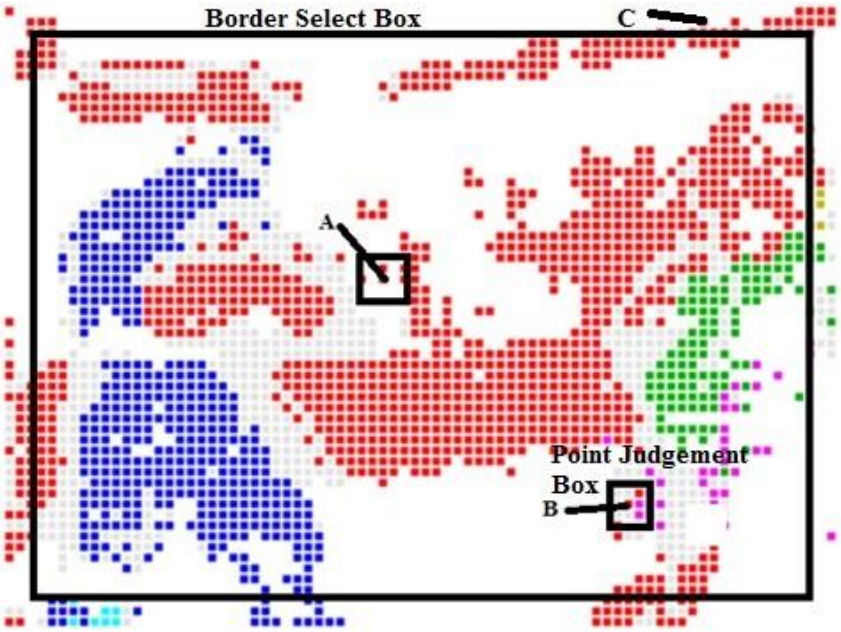


Figure 3-7. Optimization illustration.

3.4 Experiments of Background Points Detection

To the best of our knowledge, we are the first one only concern the accuracy of extracting background point. Being different with other background segmentation researches, they concern both the accuracy and integrity of extracted background point. It means that we will remove candidate background points easier, not worry if delete real background points. However, this is considered in traditional background segmentation. So, it is unfair to compare the accuracy of extracting background point between our research and traditional background segmentation. Of course, it is also inappropriate to compare the integrity point between our research and traditional background segmentation.

As mentioned above, we test our algorithm using some challenging datasets of real videos: Hollywood2 dataset [15], Stanford Olympic Sports dataset [17], and UCF50 dataset [16]. We used the train samples from Olympic Sports Dataset to adjust the specific values above and use the test samples from Olympic Sports dataset, Hollywood2 dataset, and UCF50 dataset to test these values. According the experiment results of train samples, for the parameters in background selection we set w% of largest diversity in component as 10%, set threshold H in criterion 1 as 0.8, set the E%

in criterion 2 as 10%, set the threshold R in criterion 3 as 4, set the $Weight_{div}$ as 0.8, set the $Weight_{dis}$ as 0.2. The second column of Fig. 3-8 (a) shows some background points estimation results of train samples. The second column of Fig. 3-8 (b) shows some background points estimation result of test samples based on these parameters' value. It is obvious that the parameter values we choose in background selection let the algorithm select the right background cluster for the test samples. For the parameters in background optimization, we set Hgap as image height of 1/40, Vgap as image width of 1/40, Rheight as image height of 1/20, Rwidth as image width of 1/20. The third column of Fig. 3-8 (a) shows some background points estimation results of train samples. The third column of Fig. 3-8 (b) shows some background points estimation result of test samples based on these parameters' value. It is obvious that the parameter values we choose in background optimization let the algorithm do the correct optimization.

Many videos in these datasets contain camera motions, viewpoint transformation and zoom. These influence the extraction result of background pixels. For example, the videos in Fig. 3-8 (a) contain complicated hinge motions. Some motions are fast like the punch and the jump. Some motions are very slow like Tai Chi video. The background of Tai Chi video is static. The others have the camera moving. The videos of climbing and jump have the viewpoint translation. In addition, the video of climbing has the viewpoint zoom. Although there are so many different situations in our test videos, the proposed algorithm could finally extract accurate background pixels.

As mentioned above, the whole background extraction is not necessary. What we need is the accuracy of background pixels. It can be clearly said that the background pixels in the left column nearly do not overlay the foreground, and the background pixels in the right column often overlay foreground. The reason causing this case is that the segmentation could not segment different object accurately. After all, the motion trajectory analysis based segmentation is to distinguish different motion not object and just the intermediate result. Since we give up the super pixel based segmentation optimization, the optimization in this research is necessary. However, from the biking example we find that sometimes the optimization is unnecessary. In these cases, the motion trajectory analysis based segmentation is consistent with object segmentation. This may be the

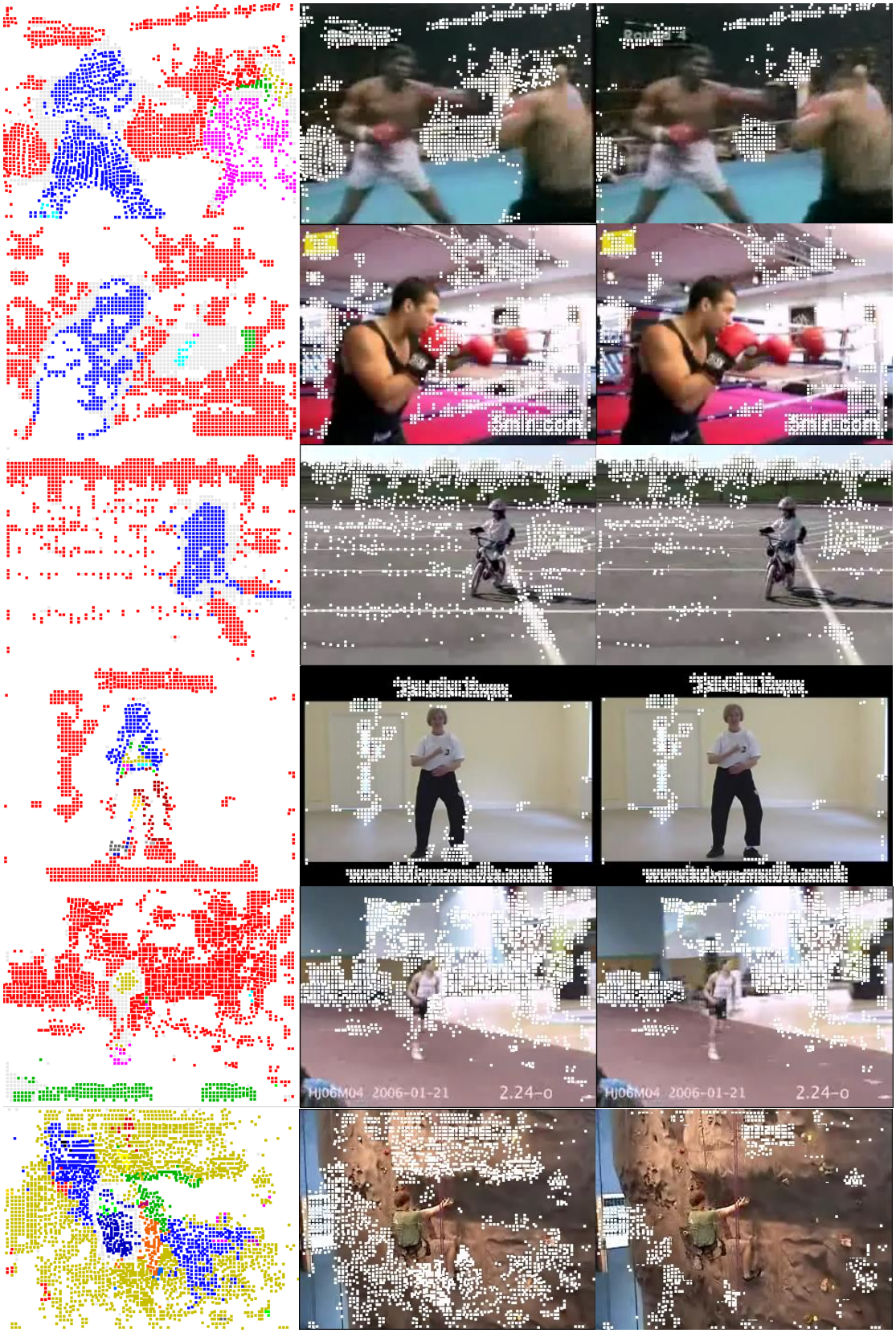


Figure 3-8 (a). Experimental results of train samples from Olympic Sports dataset.

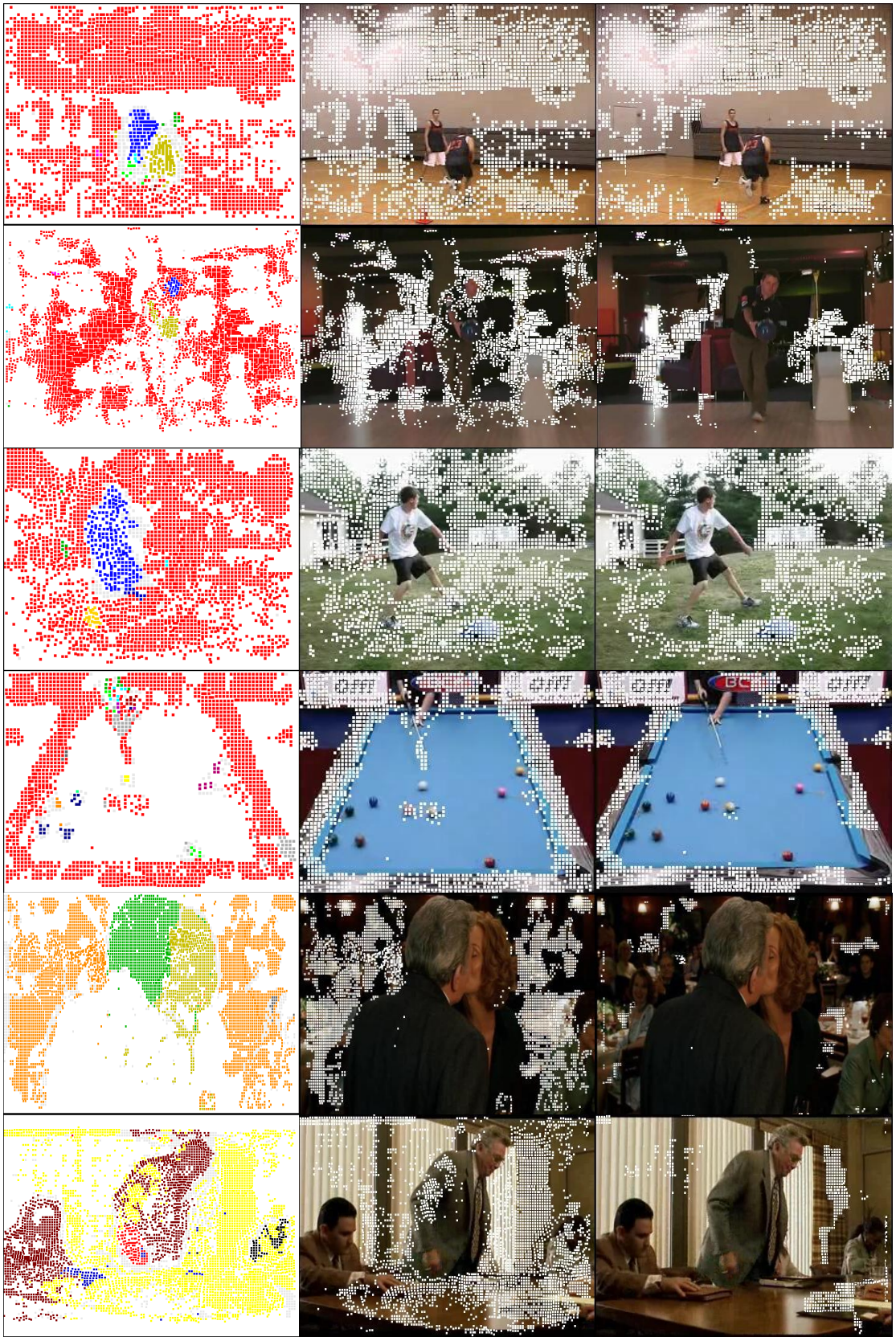


Figure 3-8 (b). Experimental results of test samples from Olympic Sports dataset, Hollywood2 dataset, and UCF50 dataset.

fault of our algorithm, but in most cases, optimization is necessary. The table 3-1 shows the processing time for different videos and average time for each frame of these videos.

Table 3-1. Video segmentation time, background point extraction time for above six videos and the average time for each frame of every videos.

	Punch race 121 frames	Punch Training 231 frames	Biking 150 frames	Tai Chi 170 frames	High Jump 74 frames	Climbing 300 frames
Segmentation time	8min 2s	26min 3s	9min 30s	8min 10s	4min 20s	16min 40s
Video processing time	1.909s	4.864s	1.608s	1.953s	2.112s	5.574s
Average time per each frame	15.77ms	21.05ms	10.72ms	11.48ms	28.54ms	18.58ms

3.5 Improved Video Descriptor for Action Recognition

In this chapter, we propose video motion representation called improved video descriptor. This representation is generated based on local patch trajectories. In our research, we adopt the dense trajectory approach proposed by Wang et al. [8] [9] [10], because it has been shown effective on several benchmarks.

We estimate the background motion pattern by the background points obtained from section 3.3. Then we rectify point trajectory descriptors by the background motion pattern and remove redundant trajectories by removing background motion pattern consistent trajectories. After that, we use bag of visual words and k-means to generate improved video descriptors. Finally, the SVM is used to do the classification.

3.5.1 Dense Trajectory

The dense trajectory proposed by Wang et al. [8] [9] [10] is extracted for multiple spatial scales as Fig. 3-9 shows. Feature points are sampled on a grid spaced by W pixels and tracked in each scale separately.

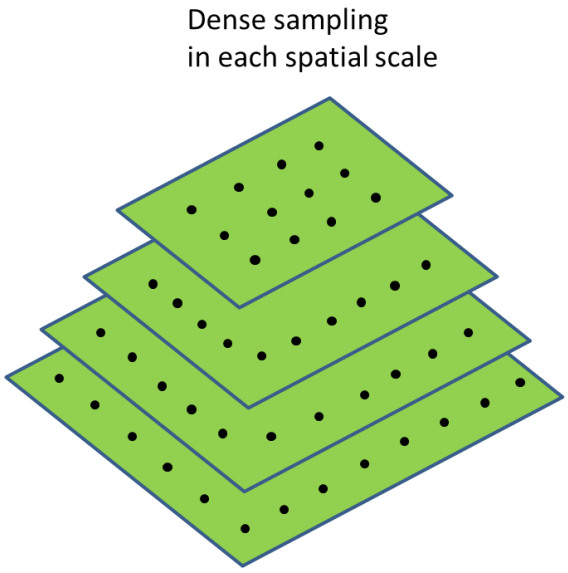


Figure 3-9. Dense sampling in spatial scale.

According to Wang's suggestion, the sampling step size of $W = 5$ is enough as dense to acquire good results. The 8 spatial scales spaced by a factor of $1/\sqrt{2}$. Each point $P_t = (x_t, y_t)$ at frame t is tracked to the next frame $t+1$ by median filtering in a dense optical flow fields $w = (u_t, v_t)$.

$$P_{t+1} = (\mathrm{x}_{t+1}, \mathrm{y}_{t+1}) = (\mathrm{x}_t, \mathrm{y}_t) + (\mathrm{M} * \mathrm{w})|_{(\overline{x_t}, \overline{y_t})}, \tag{3.22}$$

where M is the median filtering kernel, and $(\overline{x_t}, \overline{y_t})$ is the rounded position of (x_t, y_t). This is more robust than bilinear interpolation used in [62], especially for points near motion boundaries. Once the dense optical flow field is computed, points can be tracked very densely without additional cost. Points of subsequent frames are concatenated to form trajectory: $(P_t, P_{t+1}, P_{t+2}, \ldots)$.

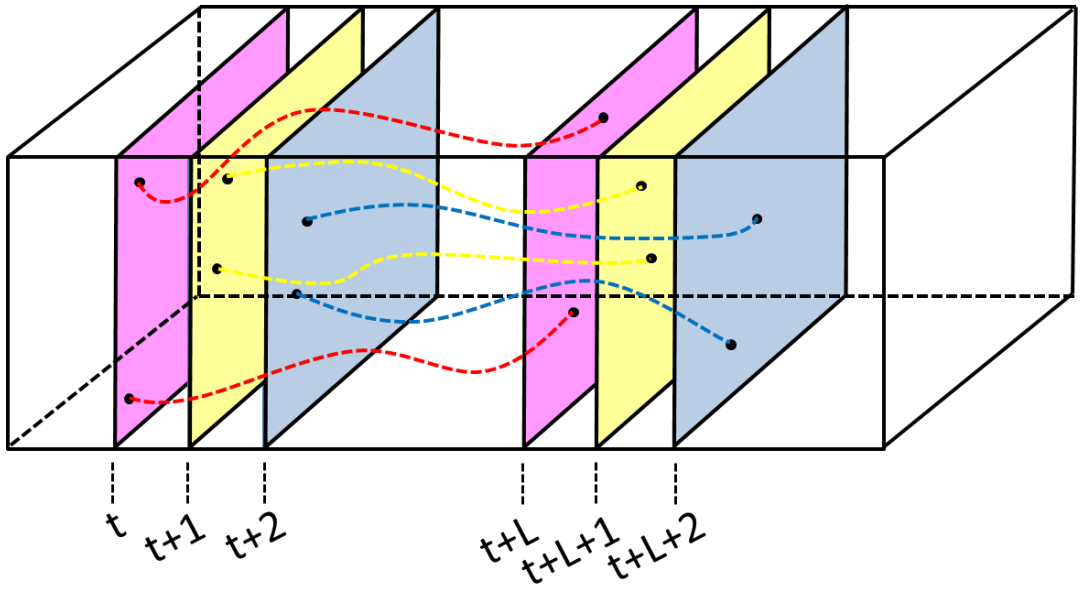


Figure 3-10. Illustration of trajectory from [5].

Fig. 3-10 illustrates point trajectories. A trajectory is a series of tracking for one point. A common problem in tracking is drifting. Trajectories tend to drift from their initial location during tracking. To avoid this problem, we limit the length of a trajectory to L. So, if the size of tracking frames is less than L, we give up this tracking. As soon as a trajectory exceeds length L, it is removed from the tracking process, see Fig. 3-10, the red curve, yellow curve, and blue curve express different initiative tracking frames. Since any tracking is ended after some frames, in our experiment, we chose a trajectory length of L = 15 frames as what Wang et al. do. If no tracked point is found in a W × W neighborhood, this feature point is sampled and added to the tracking process.

In homogeneous image areas without any structure, it is impossible to track points. Here, we use the criterion proposed by Shi and Tomasi algorithm [70]. The algorithm is one of the corner detections. The corner is the point which can be robustly detected even in a noise pattern. For example,

(1) line endings; (2) the intersection of two edges; (3) an isolated point of local intensity maximum or minimum; (4) a point for which there are two dominant and different edge directions in a local neighborhood of the point; (5) a point on a curve where the curvature is locally maximal.

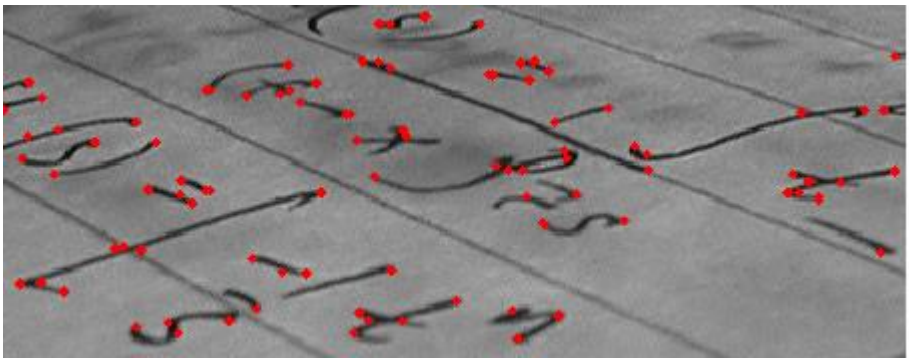


Figure 3-11. Illustration of different corners from webpage [3].

The corner examples are shown in Fig. 3-11. The red point represents the corner. These red point nearly cover all the types described above. When a feature point is sampled, we check the smaller eigenvalue of its autocorrelation matrix. If it is below a threshold, this point will not be included in the tracking process. Since for action recognition we are mainly interested in dynamic information, static trajectories are pruned in a pre-processing stage. Trajectories with sudden large displacements, most likely to be erroneous, are also removed.

The shape of a trajectory encodes local motion patterns. Given a trajectory of length L, we describe its shape by a sequence of displacement vectors $\Delta P_t = (P_{t+1} - P_t) = (x_{t+1} - x_t, y_{t+1} - y_t)$. The resulting vector is normalized by the sum of the magnitudes of the displacement vectors:

$$S_{normal} = \frac{(\Delta P_t, ..., \Delta P_{t+L-1})}{\sum_{j=t}^{t+L-1} \left\| \Delta P_j \right\|}. \tag{3.23}$$

This vector becomes a trajectory descriptor. As widely known, taking into account background motion caused by camera motions could be helpful for more effective motion trajectories extraction. These motion trajectories could improve the action recognition performance. Since one camera motion produce the same motion pattern to all the background points, using the average velocity of part background points to estimate the background motion pattern is reasonable. So, the background points from section 3.3 could be used to estimate background motion pattern for frame t as $B_t = \frac{\sum_C (u_t, v_t)}{sum(C)}$, where the C is the background cluster, the (u_t, v_t) is the optical flow for some point in C. We rectify the displacement vector in frame t for a trajectory by $\Delta P_t^{'} = (\Delta P_t - B_t)$. We rectify the trajectory descriptor by

$$S' = \frac{(\Delta P'_t, ..., \Delta P'_{t+L-1})}{\sum_{j=t}^{t+L-1} \left\| \Delta P'_j \right\|}. \quad (3.24)$$

Obviously, after eliminating the background motion, the points from background area should be relatively static. Beside the background points, some foreground points still maybe static if these points keep the same motion pattern to background motion pattern. We think that only motion trajectories from dynamic points could contribute to action recognition, the static point is no useful. For each trajectory, we compute the magnitude of vector $(\Delta P'_t, ..., \Delta P'_{t+L-1})$. If the magnitude is lower than a threshold (i.e., 1 pixel), the trajectory is considered to be consistent with the camera motion, and thus removed. The remaining trajectories are called as effective trajectory and the corresponding descriptors are called as effective trajectory descriptors.

3.5.2 Bag of Visual Words for Improved Video Descriptor

Features based on the bag of visual words have been proved effective for image classification and action recognition. We have introduced our effective trajectory features above. Now, we will introduce how to combine the bag of visual words with our effective trajectory descriptors to generate improved video descriptors. Fig. 3-12 illustrates relationships between videos in a dataset and their rectified trajectories. The parallelogram with different color refers to the different action category in the dataset, such as the blue refers to the action snatch, the gray refers to the action bowling, the yellow refers to the action tennis and the pink refers to the action vault. For each action category, we generate effective trajectory descriptors according to the formula 3.24. These trajectory descriptors will be used to generate a codebook. Fig. 3-13 illustrates the codebook construction. The first row is all the videos in a dataset. These parallelograms with different color refer to different action groups. We use the formula (3.24) to calculate effective trajectory descriptors for all videos. Then, we use the k-means clustering method to cluster these descriptors into n clusters. After that, we need to compute centers (visual words) for all the clusters. We could obtain the following vector: $(Center_1, Center_2, ..., Center_{n-1}, Center_n)$. This vector becomes our codebook. Using this codebook, improved video descriptors can be calculated as follows.

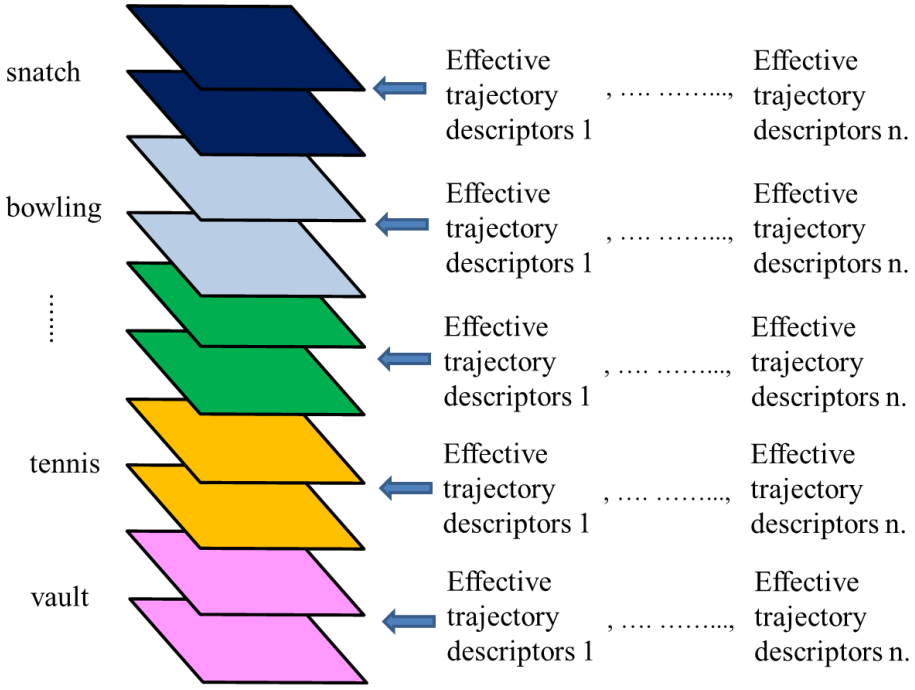


Figure 3-12. Action categories and corresponding effective trajectory descriptors.

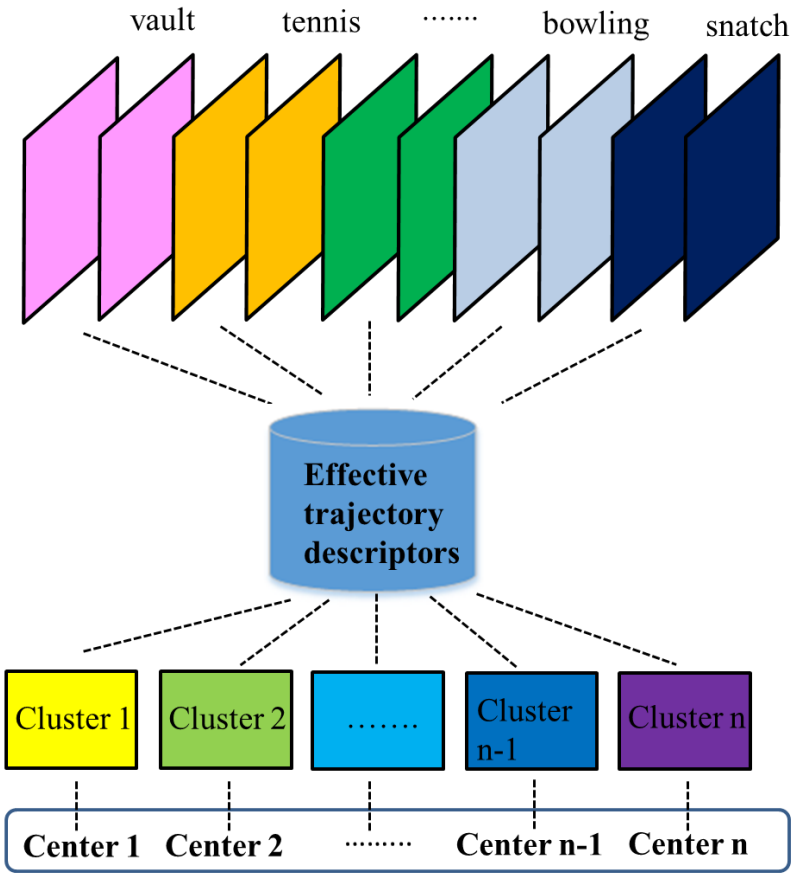


Figure 3-13. Codebook construction.

Fig. 3-14 illustrates the calculation of improved video descriptors. For example, a ‘vault’ video has many effective trajectory descriptors calculated by formula (3.24). For each effective trajectory descriptor, we calculate the Euclidean distances between the descriptor and all centers in the codebook. The center having nearest distance is the bag, to which the descriptor belongs. The corresponding visual word in the improved video descriptor will be added 1. We apply this process to all the effective trajectory descriptors and then obtain the resulting histogram of word occurrences used as the improved video descriptor to the ‘vault’ video. In the action recognition experiment, we fix the number of codebook words to 4,000, which has shown empirically good results for a wide range of video datasets.

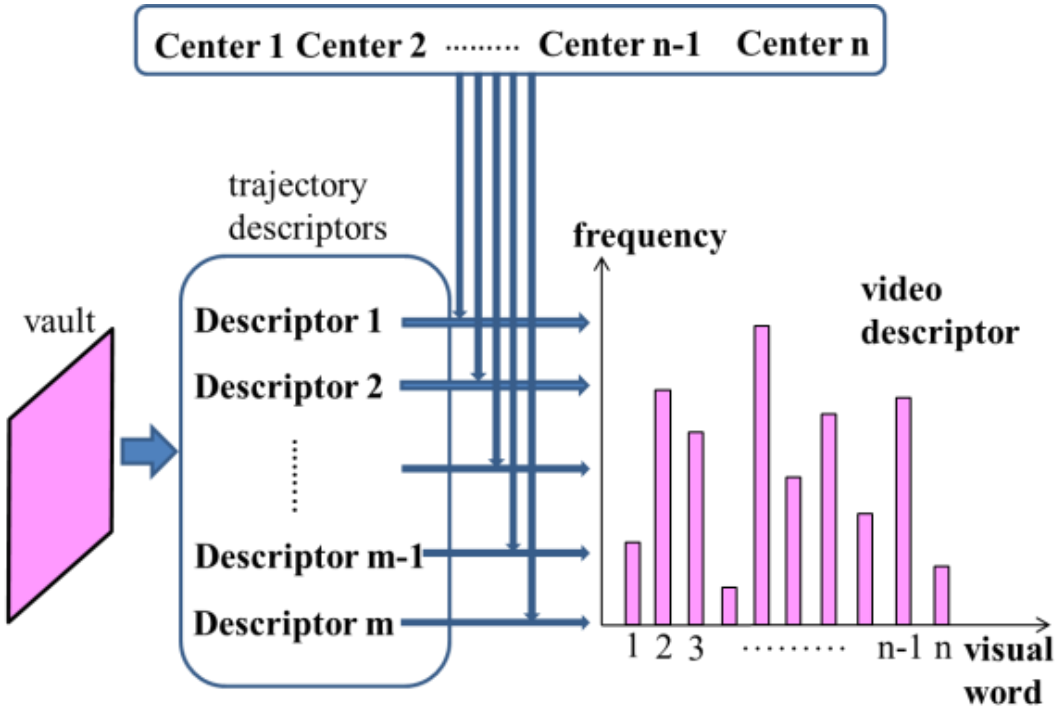


Figure 3-14. Improved video descriptor construction.

3.6 Experiments of Improved Video Descriptor for Action Recognition

3.6.1 Datasets

First of all, we introduce the datasets used in our experiments. In action recognition field, there are many excellent datasets, such as KTH Action Recognition Datasets [22], Hollywood2 [15], UCF50 [16], Olympic Sports Dataset [17], UCF YouTube Action Dataset [71], HMDB : A Large Video Database for Human Motion Recognition [69] and so on. These datasets have different characteristics. KTH is an early Action Recognition Datasets. It is more like a laboratory's datasets. (1) It is generated in some special scenes. These scenes are usually simple, for example, a barren lawn or a room with white wall and pure color floor. (2) The actions in the dataset are simple and not very much. There are only eight actions, walking, jogging, running, boxing, hand waving and hand clapping. (3) The camera is static. (4) There is no sudden light and shot change. (5) There is no other disturbance in background. In this way, this dataset is simple. Currently, those popular and state-of-the-art algorithms usually could achieve very high accuracy for action recognition on this dataset. So, it is difficult to judge which algorithm is better. In addition, videos in the dataset are black and white. So, the algorithm considering color factor could not carry on their experiment on this dataset. On the contrary, some datasets collect the video samplings entirely from the universal living scenes. The Hollywood2, UCF YouTube and HMDB are this kind of video datasets. The Hollywood2 dataset collects the video samplings from 32 movies. The UCF YouTube dataset collects the video samples from the famous video website YouTube. Most videos in YouTube come from all kinds of living scenes. The videos in HMDB dataset are similar to the videos in YouTube. These videos come from all kinds of living scenes every day. Another same characteristic for these three datasets is large. For one action category, there are usually many different scenes, backgrounds, shot changes and motion interferences. Although all these factors make the action recognition difficult, more and more researchers adopt these datasets in their experiments. Each promotion of recognition accuracy in these datasets represents that action

recognition based on computer could be used more widely in our life. Besides the three datasets above, there is another kind of dataset. The videos of this kind dataset still come from real living scene, but be different from the former three datasets, this kind dataset do not collect the video sampling from all kinds of living scenes, they usually focus on some special fields, such as sports, aerial road and so on. The UCF50 dataset and the Olympic Sports dataset stand for this kind dataset. All the actions in these two datasets are related with sport, but the action type in UCF50 dataset is more than in Olympic Sports. In our research, considering recognition challenge we gave up the KTH dataset and selected the Hollywood2 datasets from the three universal living scenes datasets. In addition, considering the comparison between the universal living scenes dataset and the special field dataset we also selected two sport datasets, the UCF50 and the Olympic Sports. Those are three datasets (Hollywood2, Olympic Sports and UCF50) that are among the most challenging datasets in the literature.

A. Hollywood2

Hollywood2 is the widely adopted dataset. The dataset has been collected from 69 different Hollywood movies and includes 12 action classes. It contains 1,707 videos divided into a training set of 823 samples and a test set of 884 samples. Training and test videos come from different movies. The performance is measured by mean average precision (mAP) over all classes, as described in [15]. Fig. 3-15 shows the video frame examples from the action type ‘answer telephone’ in Hollywood2 dataset. Although all the actions are ‘answer telephone’, there are many differences among them.

Figure 3-15. Action “answer telephone” examples in Hollywood2 dataset.

B. Olympic Sports Dataset

The Olympic Sports Dataset contains complex motions that go beyond simple punctual or repetitive actions. The dataset has 783 clips and 16 sports actions class, which are collected from YouTube sequences. The classes are basketball lay-up, bowling, clean_and_jerk, discus_throw, hammer_throw, javelin_throw, diving_platform_10m, diving_springboard_3m, high_jump, long_jump, pole_vault, shot_put, snatch, tennis_serve, triple_jump, and vault. We adopt the provided train/test split by [17] and report mAP over all classes. Fig. 3-16 shows some examples of the dataset.

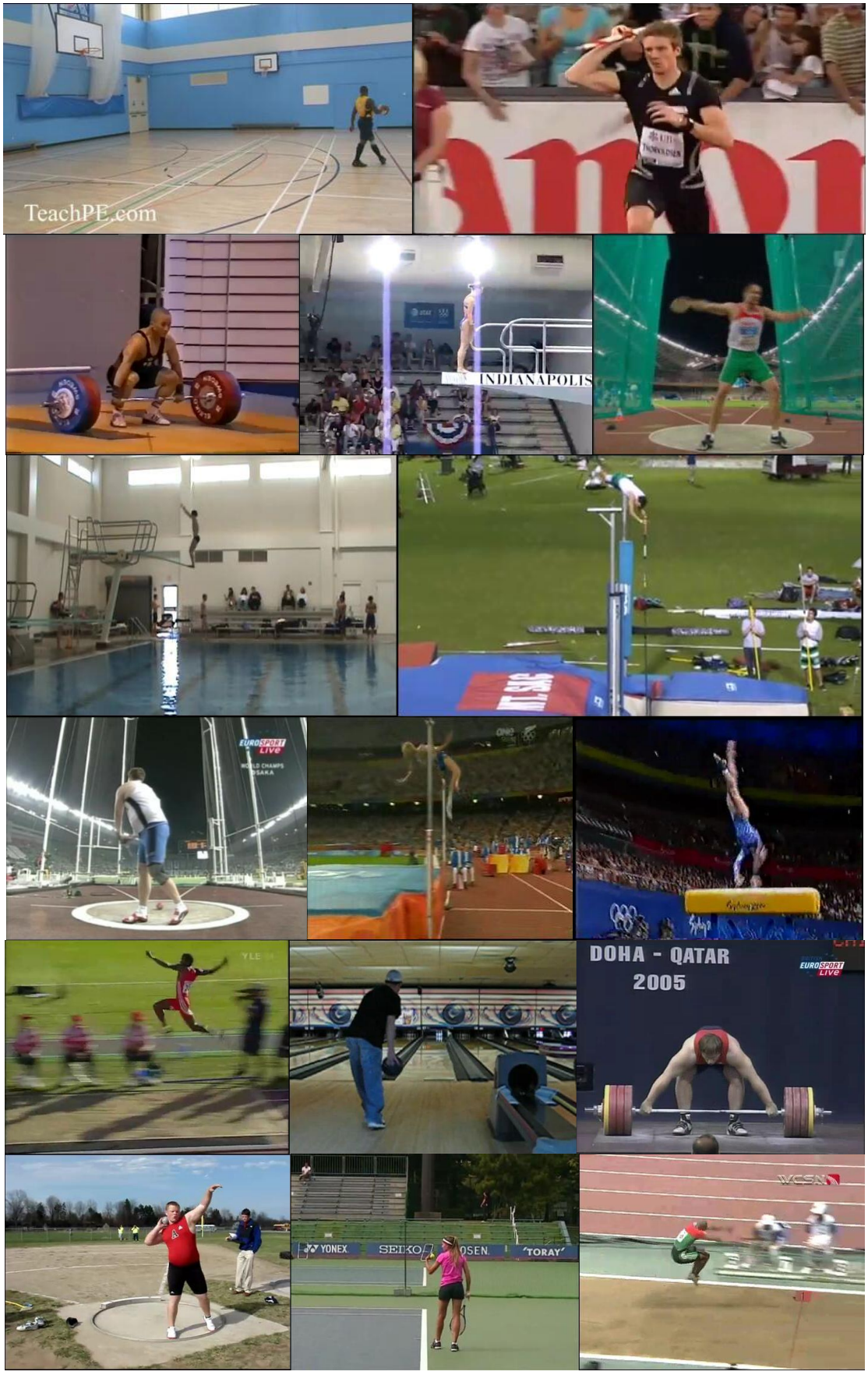


Figure 3-16. Action examples in Olympic Sports Dataset.

C. UCF50 dataset

The UCF50 dataset [16] is currently one of the popular publicly available datasets for action recognition. Be different with the Olympic Sports Dataset, this Dataset contains not only the race sports but also more sports in life. The UCF50 dataset has 50 action categories, consisting of real-world videos taken from YouTube. Each category is grouped into 25 groups. For each group, there are at least 4 action clips. In total, there are 6,618 video clips. The video clips in the same group may share some common features, such as the same person, similar background or similar viewpoint. We apply the leave-one-group-out cross-validation and report average accuracy over all classes like the way of [16]. Fig. 3-17 shows some examples of the dataset.

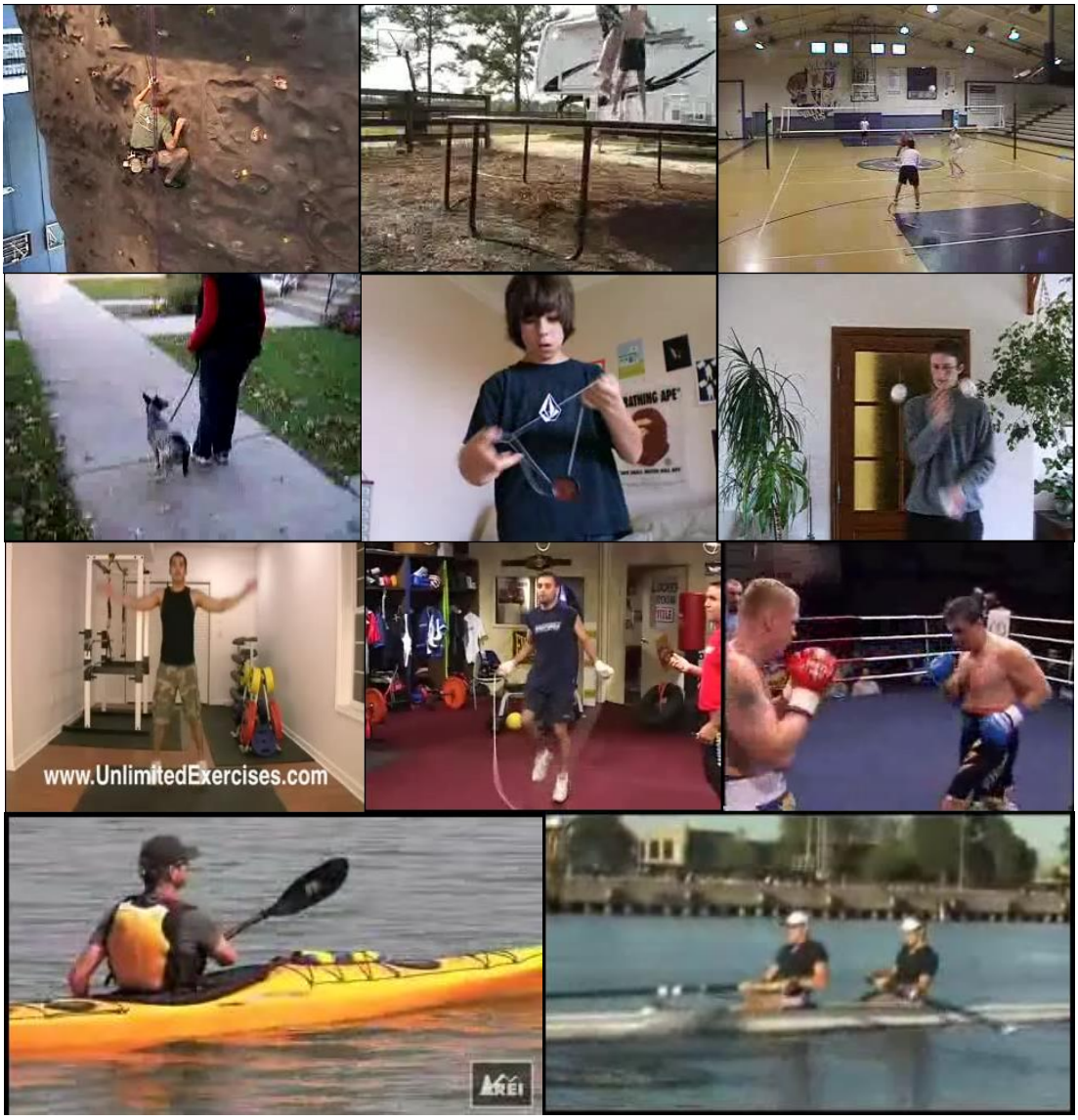


Figure 3-17. Action examples in UCF50 dataset.

Obviously, the actions in sport are usually with more complex than in daily life. To some action recognition techniques, the action complexity could be benefit with improvement of recognition accuracy. For example, if the action recognition is just based on motion trajectory, instinctively, the recognition accuracy for complex motion will be higher than for simple motion. This is because the complex motion trajectory could enhance the uniqueness of motion pattern. Fig. 3-18 (a) and Fig. 3-18 (b) illustrate this viewpoint. Fig. 3-18 (a) is action stand up and Fig. 3-18 (b) is action handshake.

Figure 3-18 (a). Actions stand up.

Suppose that you only see the first three video frames of Fig. 3-18 (b), maybe you will also recognize the action as stand up. Only after seeing the fourth video frame, you could understand that the action is handshake. See, even for human, it is difficult if there is not enough information. How can you expect the computer is cleverer than you?

Figure 3-18 (b). Action hand shake.

The reason is that motion trajectory is an important sign for motion recognition. Fig. 3-19 clearly shows that the motion trajectory of stand up is included in the motion trajectory of handshake. Maybe for us, we could still distinguish the two actions from their motion trajectories, but for computer it is difficult unless we use some special principle. It is impossible that you could not develop a special principle for some action category, which could distinguish this action category from other all action categories. In this research, we use the unanimous principle for all actions' recognition. If the action is complex, the recognition accuracy raises. This could be seen in the experimental result described in section 3.6.2.

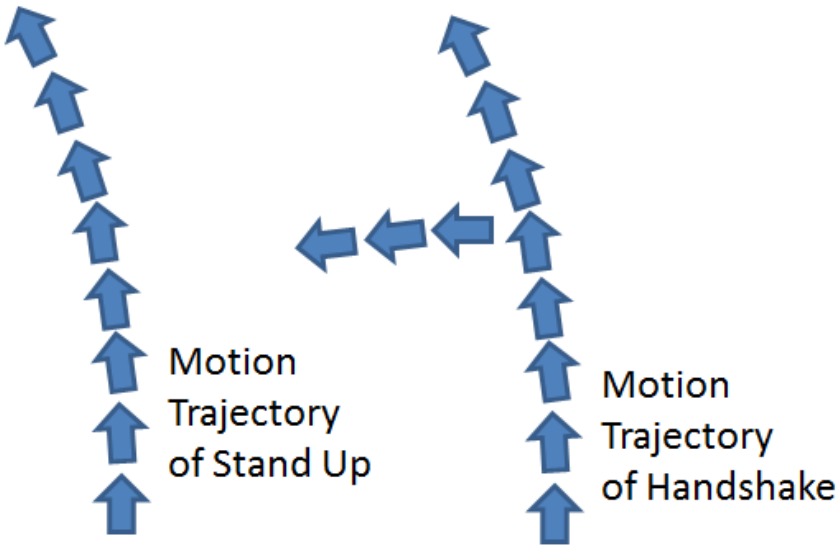


Figure 3-19. Motion trajectory for stand up and hand shake.

3.6.2 Results

Fig. 3-20 shows the obvious difference between the original trajectory and our rectified trajectory considering the background motion. The video examples in Fig. 3-20 come from the Olympic Sports dataset. Due to the camera motion, there are many redundant trajectories in the original trajectories. It is obvious that so many trajectories are located on the background, which will influence the accuracy of action recognition.

In following each figure, column 1 is original input videos, column 2 is original point trajectories (dense trajectory) proposed by Wang et al. and column is our rectified point trajectories considering the background motion.

In the diving springboard 3m, shot put and so on, camera shake exists. So, you could see many small motion trajectories. By our background motion pattern removal, massive meaningless trajectories have been removed. The small motion trajectories about water wave are still removed. These small motion trajectories about water wave still maybe the error or meaningless tracking.

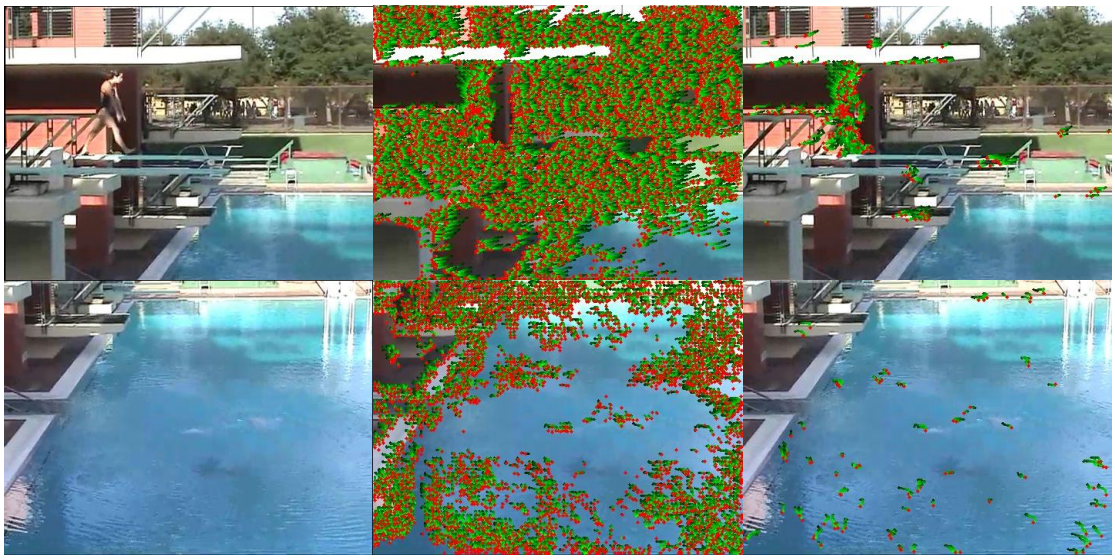


Figure 3-20-1. Diving springboard 3m video (195 frames): column 1 is input video frames (top: frame No. 45, bottom: frame No. 188), column 2 is original point trajectories, and column 3 is our rectified point trajectories considering background motion.

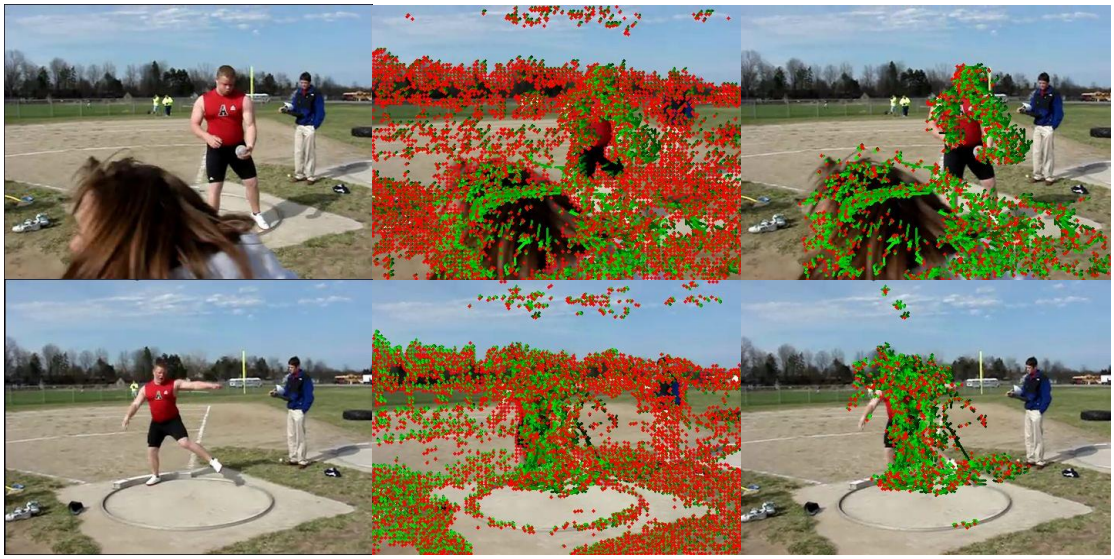


Figure 3-20-2. Shot put video (276 frames): column 1 is input video frames (top: frame No. 23, bottom: frame No. 233), column 2 is original point trajectories, and column 3 is our rectified point trajectories considering background motion.

Although a static camera is common in the bowling and tennis serve videos, there are still many sampling points due to the sampling strategy of dense trajectory that samples by a fixed step

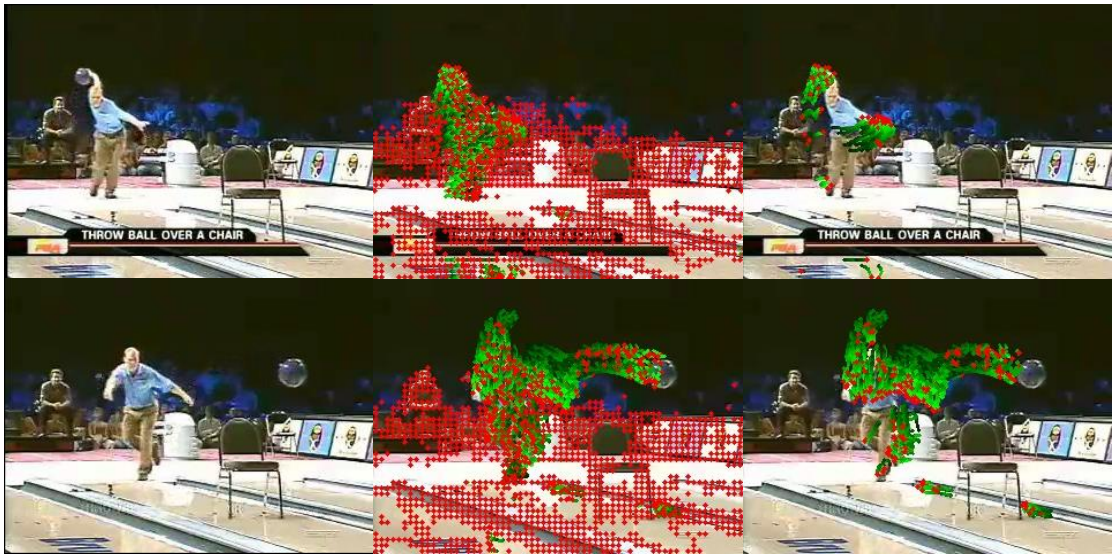


Figure 3-20-3. Bowling video (102 frames): column 1 is input video frames (top: frame No. 56, bottom: frame No. 96), column 2 is original point trajectories, and column 3 is our rectified point trajectories considering background motion.

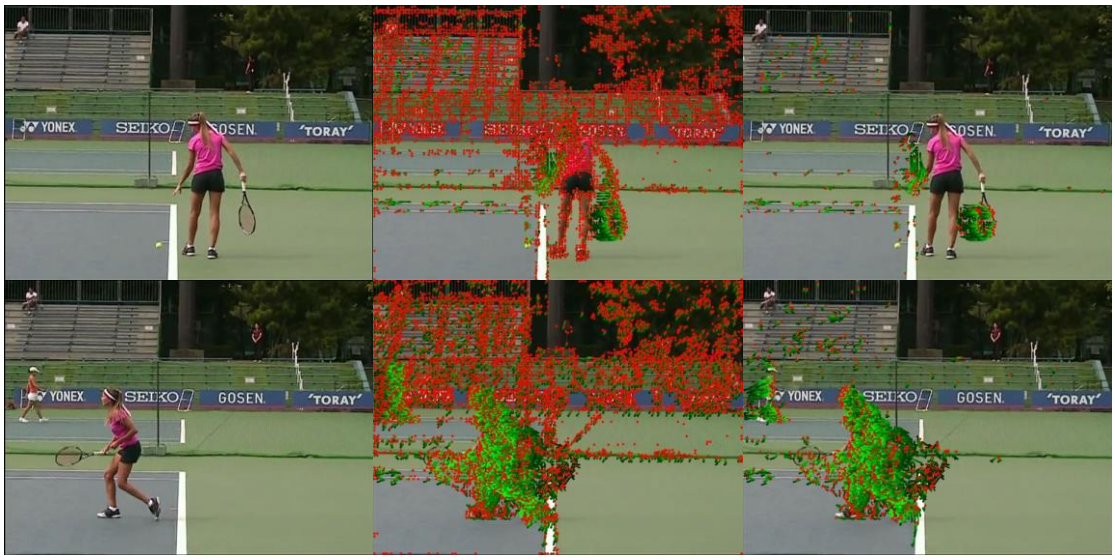


Figure 3-20-4. Tennis serve video (202 frames): column 1 is input video frames (top: frame No. 47, bottom: frame No. 193), column 2 is original point trajectories, and column 3 is our rectified point trajectories considering background motion.

W in row and column. If we do not deal with these sampling points specially, it will influence the accuracy of classification. Adding the special strategy is not a good choice because it is difficult to predicate how many action categories exist, and there is some hardly imaginary motion trajectory noise. So, it can be said that our algorithm deal with this situation very well. Furthermore, as mentioned above, our method eliminates small motion trajectories. Although the trajectory amount is sparser than the original dense trajectory, there is no problem. In section 3.6.1, we have

explained that based on the features motion trajectories, the complex motion video is easier to be recognized than the simple motion video. The one important reason is that there are more long motion trajectories with uniqueness in the complex motion video than in simple motion video. So it indicates that the long motion trajectory is useful than short motion trajectory for action recognition. These short motion trajectories occurring in the homogeneous area and the corner area cause error tracking.

Although the clean and jerk video contains the camera shake and zooming, if the zooming velocity is slow, our algorithm is still effective to this situation. However, if the zooming velocity is quick, it is difficult to unify the background motion pattern just by the operation introduced in section 3.5.2.

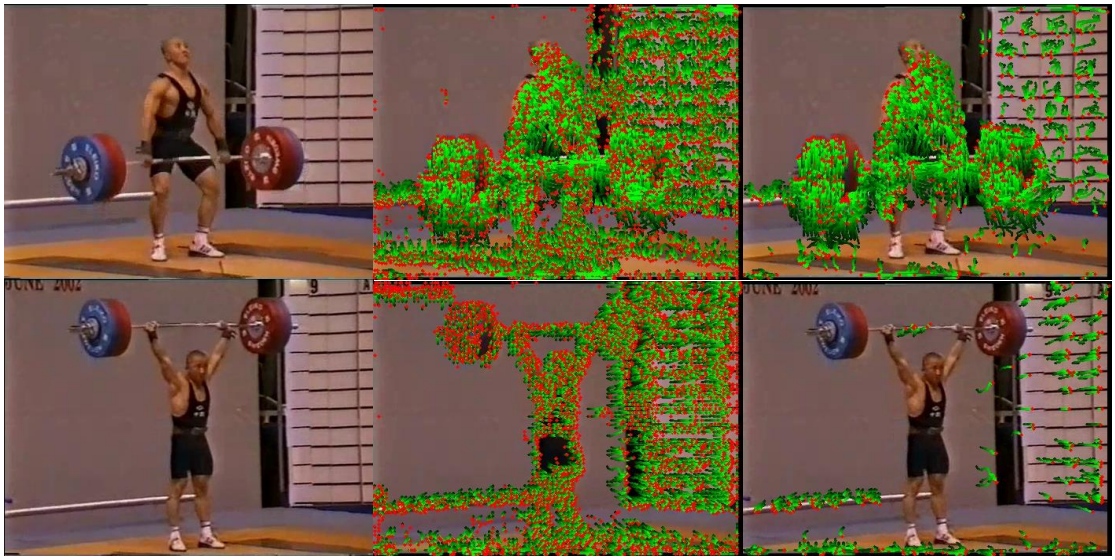


Figure 3-20-5. Clean and jerk video (461 frames): column 1 is input video frames (top: frame No. 207, bottom: frame No. 408), column 2 is original point trajectories, and column 3 is our rectified point trajectories considering background motion.

In the discus throw video, the net between the camera and the object influences the motion trajectory extraction. Compared to the motion trajectory extraction results, it can be said that our algorithm is useful for this situation.

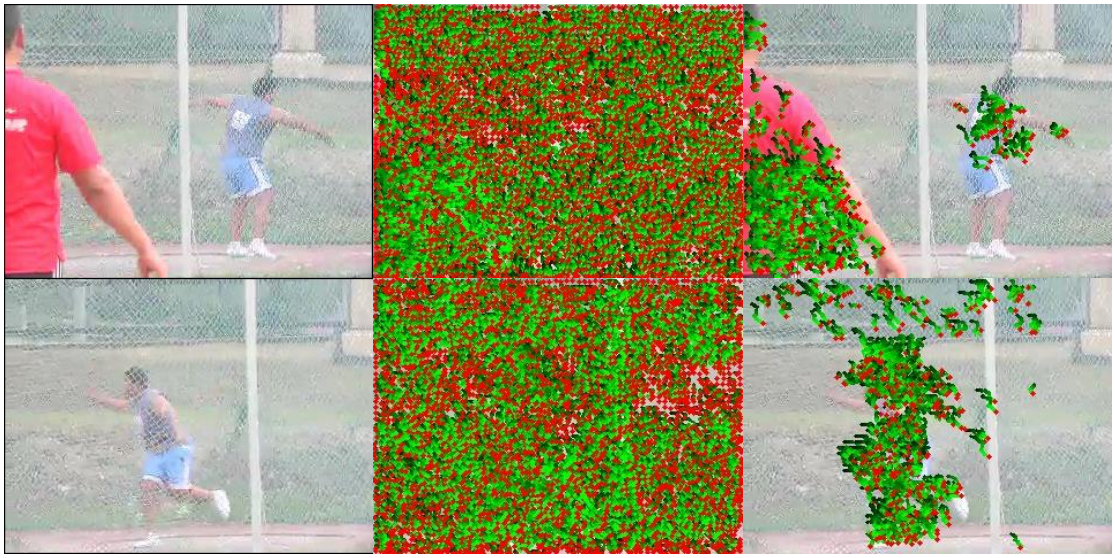


Figure 3-20-6. Discus throw video (121 frames): column 1 is input video frames (top: frame No. 43, bottom: frame No. 100), column 2 is original point trajectories, and column 3 is our rectified point trajectories considering background motion.

The quick moving camera usually exists in the basketball, high jump and so on. It is obvious that short motion trajectories are removed but not long motion trajectories. There is difference between a camera motion and a sportsman motion. In most situations, the relative camera motions in the action basketball, high jump, and vault are irregular. Our method is powerless in these situations. In the triple jump, pole vault videos, the camera is regularly located in the back on to the sportsman. So, our method is not very effective in these cases.

Figure 3-20-7. Basketball layup video (152 frames): column 1 is input video frames (top: frame No. 19, bottom: frame No. 125), column 2 is original point trajectories, and column 3 is our rectified point trajectories considering background motion.

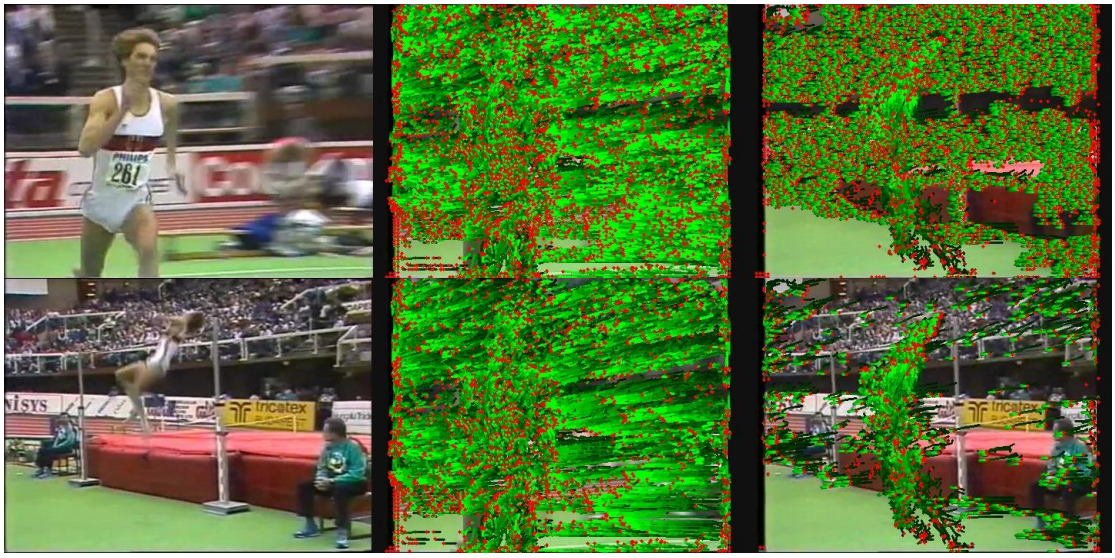


Figure 3-20-8. High jump video (161 frames): column 1 is input video frames (top: frame No. 21, bottom: frame No. 78), column 2 is original point trajectories, and column 3 is our rectified point trajectories considering background motion.

Figure 3-20. Original point trajectories and rectified point trajectories with background removal.

When generate the improved video descriptors for action recognition, we fix the number of codebook words to 4,000, which has shown empirically good results for a wide range of datasets. An SVM is used for the classification. We execute the classification under the original video descriptor and our improved video descriptor. Table 3-2 gives the recognition accuracy for each category of Olympic Sports Dataset.

From the mAP of Olympic Sports Dataset, you could find that our method is really help to improve the recognition accuracy. However, seeing the accuracy of each category, you could find that algorithm is not useful for all the action categories. The effective categories are diving platform 10m, diving springboard 3m, discus throw, Hammer throw, Shot put, and tennis serve. The ineffective categories are basketball layup, high jump, pole vault, bowling, long jump, snatch, and vault. There are other special categories, the clean and jerk, javelin throw, and triple jump. In these special categories, the recognition accuracy of clean and jerk is 100%, it could not be improved by our algorithm. The javelin throw and triple jump are 0%. At first, we thought that our algorithm could break the 0%, but it was failed for these two action categories. There are two possible primary reasons; one is that the relative moving between camera shot and human is small or static; another is the quick moving to camera shot. Due to the two reasons, the large motion

trajectories produced by the moving camera shot under static background become the main part for video trajectory set. Now, we will discuss the accuracy for all the remaining actions.

Table 3-2. Recognition accuracy for each category.

Motion Category	**Accuracy of Original Video Descriptor**	**Accuracy of Improved Video Descriptor**
basketball layup	70.0%	60.0%
bowling	88.9%	88.9%
clean and jerk	100%	100%
discus throw	81.8%	100%
diving platform 10m	88.9%	100%
diving springboard 3m	50.0%	87.5%
hammer throw	37.5%	75.0%
high jump	72.7%	63.6%
javelin throw	0%	0%
long jump	83.3%	83.3%
pole vault	87.5%	57.1%
shot put	70.0%	90.0%
snatch	88.9%	88.9%
tennis serve	71.4%	85.7%
triple jump	0%	0%
vault	80.0%	80.0%
Olympic Sports (mAP)	66.9%	72.5%

First, we discuss the success action categories. For these actions, the camera shot usually move slowly or is static, or the moving distance is small each time. In these situations, the sampling points on background generate small motion trajectories or static samplings. In addition, for these six actions, the shot view angle is usually very large, so it could contain more human and objects. Note that here the object not only indicate the object under common concept in life but anything

could produce the massive corners, like spitted water or wave. These human and objects are usually help to generate the samplings. Especially, for the actions hammer throw, discus throw and so on, the protection net shown in the Fig. 3-20-6 make great contribution for the samplings generation. Obviously, the massive samplings on background will generate massive small motion trajectories or static samplings. These static samplings and small motion trajectories will reduce the motion accuracy. When we show the motion trajectory examples in Fig. 3-20, we explain that our algorithm is useful for removing small motion trajectory and static sampling. So, we could eliminate the bad influence from these disturbing factors.

As the slow moving camera shot is the key for the success action categories. The quick moving camera shot is the main reason for the failed action categories. For the action categories basketball layup, high jump, pole vault, long jump, vault, camera shot moves quickly. As we discuss with showing the motion trajectory examples in Fig. 3-20, our algorithm could not deal with the large motion trajectory very well. These large motion trajectories make the bad influence to recognition accuracy. In addition, the recognition accuracy for long jump and vault is very high. When the test samples are not so many, it is difficult to improve the recognition accuracy. Except these two action categories, the accuracy of the other three actions, basketball layup, high jump and pole vault decrease. It is odd. Intuitively even if the accuracy does not rise, it should not decrease. We note that in the action basketball layup, there are usually some players. Since our algorithm is based on unified background motion pattern, obviously, the algorithm could not eliminate the motion trajectory provided by these players. Although all the players come here for playing basketball, only the player having a basketball takes the similar action to the player having basketball in other basketball layup video. The other players usually act randomly which is harmful to recognition accuracy. In addition, since our algorithm could remove static samplings and small motion trajectories, it not only removes these on background, but on human. So maybe, the remaining effective motion trajectories (provided by a player having a basketball) and the ineffective motion trajectories will cause the lower identification degree video descriptors. The SVM classifier trained by these descriptors could not ensure better performance. The same problem occurs in the action high jump. The competition area for high jump usually is outdoors. So, besides the sportsman, many other people and objects exist. When some effective small

motion trajectories have been removed, the effect of ineffective motion trajectories for the video descriptor will be magnified. Furthermore, the negative sample video come from the action javelin throw. The motion trajectory of javelin throw is similar to high jump and pole vault. So, these will influence the recognition accuracy. If we try other action category videos, which are unlike with the high jump and pole vault, the accuracy of high jump and pole vault become 72.7% and 100% respectively. Until now, there are only two action categories left, the action bowling and snatch. In the two actions, the camera shot is static and slowly, so there must be some static samplings and small motion trajectories. According to what we say above, our algorithm is useful for removing these disturbing factors. We naturally consider that the algorithm should raise the recognition accuracy for the two actions. It is weird that there is no effect for the two actions. Of course, the high accuracy on the original motion trajectory and small-scale test samples are one possible reason. In addition, we note that the video time of blowing and snatch is long and there are usually long periods that the human is static. In this period, although our algorithm removes some disturbing factors, the long shooting time make dense trajectory system accumulate massive disturbing factors. Moreover, our algorithm still removes some effective motion trajectories on human. In this case, we could not ensure the remaining motion trajectories could express a better performance.

Table 3-3 gives the results as the mAP of the original video descriptor based on the bag-of-features in the three datasets using the original trajectory descriptors. In addition, for discussing our improved video descriptor, we also present the results using the effective trajectory descriptor considering background motion on the same datasets.

Table 3-3. Performance of the original video descriptor and our improved video descriptor on Hollywood2, Olympic Sports and UCF50.

	Hollywood2	Olympic Sports	UCF50
Original Video Descriptor	40.6%	66.9%	63.5%
Improved Video Descriptor	45.7%	72.5%	68.4%

From Table 3-3, it can be clarified that the improved video descriptors based on our background motion estimation is help to improve the performance for the action recognition on these challenging datasets. The idea of combining saliency and background nature could be suitable to select background cluster from the clusters by long-term point trajectory analysis [12]. The

optimization makes the remaining background cluster be more accurate for estimating background motion pattern. The improved video descriptors based on accurate background motion estimation are the primary reason to accuracy improvement of action recognition. Furthermore, we note that the recognition accuracy of Olympic Sports is highest and that of Hollywood2 is lowest. During the introduction of datasets in section 3.6.1, we guess that the datasets with complex actions have higher recognition accuracy than the datasets with simple actions. Obviously, the action in Olympic Sports is more complex than the action in Hollywood2. Most actions in Hollywood2 are simple actions, which have a low identification degree. Here, the experimental result supports our judgement. In addition, this research could widely be used in all kinds of videos, slow action video, fast action video, dominant foreground video and so on. This universality is another major reason to accuracy improvement.

3.7 Conclusion

We have introduced an approach that generate improved video descriptor with background points estimation for action recognition in practical videos, where extensive camera motion (Pan-Tilt-Zoom) exists [72], which affects the performance to many existing features. Our proposed solution explicitly extracts motion information in videos. To the best of our knowledge, we are the first one propose an idea called accurate partial background extraction according to the characteristic of background motion pattern estimation. This idea allows us not to consider the extraction integrity for real background, but the extraction precision for real background. We combine long-term point trajectory analysis with saliency and background nature to obtain background cluster. The use of saliency and background nature makes us not limit the video subject to some specific one, such as human [8], animal and so on. The researches [73] [74] are similar to our research. However, the most important difference is that besides possible background cluster selection, we propose background cluster optimization based on the accurate partial background extraction in section 3.3.2. From Fig. 3-6, we could see that the optimization is necessary, especially to our improved video descriptor generation for action recognition.

The experimental results are enough to prove the effectiveness of our method, but the warp optical flow mentioned in [8] [9], and the interaction of moving objects [73] have been proved important and effective for action recognition. So, one possible promising future work is to consider combining the idea of warp optical flow and the interaction of moving objects with our research. Another possible meaningful work is to add our research into the technique for event detection [75] [76] since we all focus on the motion trajectory and exploit statistics representation for the analysis of the activities.

Chapter 4: Concluding Remarks

In this dissertation, to the key of sparse sampling and dense sampling for action recognition we propose solutions respectively. The sparse sampling hope the sampling points are the most meaningful points to motion of foreground object. We consider the end point and joint point are the most meaningful points, and propose an accurate end point and joint point extraction process. However, this process requires the static camera video and not so much occlusions. In the future, maybe adding some tracking and predicting module could overcome the occlusion problem. In the dense sampling for action recognition, camera motion is primary reason for affecting the correct video descriptor generation. Our research detects the background points accurately, estimates the background motion pattern, and updates the trajectory descriptor by removing the background motion. In addition, through the magnitude of the rectified trajectory descriptor vector, we eliminate the useless trajectories. The remaining rectified trajectory descriptors (the effective trajectory descriptors) could be used to generate the improved video descriptors. Using these improved video descriptors could recognize the action more accurately. However, still there are shortages about this research. If the camera moves quickly, our method could not deal with the useless trajectories very well. In the future, for solving this problem, we could (1) consider the appearance model and use more mathematical method to estimate the background cluster like [74]; (2) use the extracted background points in a more effective way.

Published Papers

[1] Yu Xiang, Yoshihiro Okada, Kosuke Kaneko : Accurate Background Points Detection for Action Recognition in Practical Video Datasets, Proc. of the Int. Conf. on Computer Graphics, Visualization, Computer Vision and Image Processing 2016, pp. 195-205, July 2016.

[2] Yu Xiang, Yoshihiro Okada : Robust Approach for Interesting Points Extraction of Moving Human from 2D Videos, ISIP2015 Post Proceedings, Springer series "Communications in Computer and Information Science" (CCIS), pp. 88-104, August 2016.

[3] Yu Xiang, Yoshihiro Okada, Kosuke Kaneko : Action Recognition for Videos by Long-term Point Trajectory Analysis with Background Removal, The 12th International Conference on Signal Image Technology & Internet based Systems (SITIS), IEEE CS Press, pp. 23-30, November 2016.

[4] Yu Xiang, Shohei Nakamura, Hiroki Tamari, Shigeru Takano, Yoshihiro Okada : 3D Model Generation of Cattle by Shape-from-Silhouette Method for ICT Agriculture, VENOA 2016, IEEE CS Press, pp. 611-616, July 2016.

References

[1] Andres Sanin, Conrad Sanderson, and Brian C. Lovell, “Shadow Detection: A Survey and Comparative Evaluation of Recent Methods,” Pattern Recognition, Vol. 45, No. 4, Apr. 2012, pp. 1684-1695.

[2] Olivier Barnich and Marc Van Droogenbroeck, “ViBe: A Universal Background Subtraction Algorithm for Video Sequences,” IEEE Transactions on Image Processing, Vol. 20, No. 6, Jun. 2011, pp. 1709-1724.

[3] Li Feng He, Yu Yan Chao and Kenji Suzuki, “A Run-Based Two-Scan Labeling Algorithm. IEEE Transactions on Image Processing,” Vol. 17, No. 5, May. 2008, pp. 749-756.

[4] T. Y. Zhang and C. Y. Suen, “A Fast Parallel Algorithm for Thinning Digital Patterns,” Communications of the ACM, Vol. 27, No. 3, Mar. 1984, pp. 236-239.

[5] Louisa Lam, Seong-Whan Lee and Ching Y. Suen, “Thinning Methodologies-A Comprehensive Survey,” IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 14, No. 9, Sep. 1992, pp. 869-885.

[6] D. C. Costa and C. A. B. Mello, “Topological Stacking Grayscale Thinning for Edge Detection and Real-time Applications,” Proc. IEEE International Conference on Image, Oct. 2014, pp. 4717-4721.

[7] Heng Wang, Muhammad Muneeb Ullah, Alexander Klaser, Ivan Laptev and Cordelia Schmid, “Evaluation of local spatio-temporal features for action recognition,” Proc. British Machine Vision Conference, Sep. 2009, pp. 124.1-124.11.

[8] Heng Wang, Alexander Klaser, Cordelia Schmid, and Liu Cheng-Lin, “Action Recognition by Dense Trajectories,” IEEE Conference on Computer Vision and Pattern Recognition, Jun. 2011, pp. 3169-3176.

[9] Heng Wang, Alexander Kl ̈aser, Cordelia Schmid, and Cheng-Lin Liu, “Dense trajectories and motion boundary descriptors for action recognition,” International Journal of Computer Vision, Vol. 103, No. 1, Jan. 2013, pp. 60-79.

[10] Heng Wang and Cordelia Schmid, “Action Recognition with Improved Trajectories,” Proc. IEEE International Conference on Computer Vision,” Dec. 2013, pp. 3551-3558.

[11] Alessandro Prest, Cordelia Schmid, and Vittorio Ferrari. “Weakly supervised learning of interactions between humans and objects,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 34, No. 3, Nov. 2012, pp. 601-614.

[12] Margret Keuper, Bjoern Andres, and Thomas Brow, “Motion Trajectory Segmentation via Minimum Cost Multicuts,” Proc. 2015 IEEE International Conference on Computer Vision. Dec. 2015.

[13] Fei-Fei Li and Pietro Perona, “A Bayesian Hierarchical Model for Learning Natural Scene Categories,” Proc. IEEE Conference on Computer Vision and Pattern Recognition, Jun. 2005, pp. 524-531.

[14] Corinna Cortes and Vladimir Vapnik, “Support-Vector Networks,” Machine Learning, Vol. 20, No. 3, Sep. 1995, pp. 273-297.

[15] Marcin Marszałek, Ivan Laptev, and Cordelia Schmid, “Actions in context,” Proc. IEEE Conference on Computer Vision and Pattern Recognition, Jun. 2009, pp. 2929-2936.

[16] Kishore K. Reddy and Mubarak Shah. “Recognizing 50 human action categories of web videos,” Machine Vision and Applications, Vol. 24, No. 5, Jul. 2013, pp. 971-981.

[17] Juan Carlos Niebles, Chih-Wei Chen, and Fei-Fei Li, “Modeling temporal structure of decomposable motion segments for activity classification,” Proc. European conference on Computer vision, Sep. 2010, pp. 392-405.

[18] Ivan Laptev, Marcin Marszałek, Cordelia Schmid, and Benjamin Rozenfeld, “Learning realistic human actions from movies,” Proc. IEEE Conference on Computer Vision and Pattern Recognition, Jun. 2008, pp. 1-8.

[19] J. Alison Noble, “Finding corners,” Image and Vision Computing, Vol. 6, No. 2, May 1988, pp. 121-128.

[20] Chris Harris and Mike Stephens, “A combined corner and edge detector,” Proc. The Fourth Alvey Vision Conference, 1988, pp. 147-151.

[21] Krystian Mikolajczyk and Cordelia Schmid, “Indexing based on scale invariant interest points,” Proc. The Eighth IEEE International Conference on Computer Vision, Feb. 2001, pp. 525-531.

[22] Ivan Laptev and Tony Lindeberg, “Space-time Interest Points,” Proc. The Ninth IEEE International Conference on Computer Vision, Oct. 2003, pp. 432-449.

[23] Ivan Laptev, “On Space-time Interest Points,” International Journal of Computer Vision, Vol. 64, No. 2/3, Sep. 2005, pp. 107-123.

[24] Piotr Dollar, Vincent Rabaud, Garrison Cottrell, and Serge Belongie, “Behavior Recognition via Sparse Spatio-Temporal Features,” 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, Oct. 2005.

[25] Shu-Fai Wong and Roberto Cipolla, “Extracting Spatiotemporal Interest Points using Global Information,” Proc. IEEE International Conference on Computer Vision, Nov. 2007, pp. 3455-3460.

[26] Geert Willems, Tinne Tuytelaars, and Luc Van Gool, “An Efficient Dense and Scale-Invariant Spatio-Temporal Interest Point Detector,” Proc. European Conference on Computer Vision, Oct. 2008, pp. 650-663.

[27] Bhaskar Chakraborty, Michael B. Holte, Thomas B. Moeslund, and Jordi Gonz ŀez, “Selective spatio-temporal interest points,” Computer Vision and Image Understanding, Vol. 116, No. 3, Mar. 2012, pp. 396-410.

[28] Murat Ekinci and Eyup Gedikli, “Silhouette Based Human Motion Detection and Analysis for Real-Time Automated Video Surveillance,” Turkish Journal of Electrical Engineering and Computer Sciences, Vol. 13, No. 2, Jan. 2005, pp.199-229.

[29] Li Qun Xu, Jose Luis Landabaso, and Montse Pardas, “Shadow Removal with Blob-based Morphological Reconstruction for Error Correction,” Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, Mar. 2005, pp. 729-732.

[30] Michael Donoser and Dieter Schmalstieg, “Discrete-Continuous Gradient Orientation Estimation for Faster Image Segmentation,” Proc. IEEE Conference on Computer Vision and Pattern Recognition, Jun. 2014, pp. 3158-3165.

[31] Ahmed Elgammal, David Harwood, and Larry Davis, "Non-parametric Model for Background Subtraction," Proc. the European Conference on Computer Vision, Jun. 2000, pp. 751–767.

[32] Hanzi Wang and David Suter, "A Consensus Based Method for Tracking : Modelling Background Scenario and Foreground Appearance," Vol. 40, No. 3, Pattern Recognition, Mar. 2007, pp. 1091–1105.

[33] Ismail Haritaoglu, David Harwood, and Larry S. Davis, "W 4: Real-time Surveillance of People and Their Activities," IEEE Transcations on Pattern Analysis and Machine Intelligence, Vol. 22, No. 8, Aug. 2000, pp. 809-830.

[34] Rita Cucchiara, Costantino Grana, Massimo Piccardi, and Andrea Prati, "Detecting Moving Objects, Ghosts, and Shadows in Video Streams," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, No. 10, Oct. 2003, pp. 1337–1342.

[35] Elena Salvador, Andrea Cavallaro, and Touradj Ebrahimi, "Cast Shadow Segmentation using Invariant Color Features," Computer Vision and Image Understanding, Vol. 95, No. 2, Aug. 2004, pp.238–259.

[36] A. Cavallaro, E. Salvador, and T. Ebrahimi, "Shadow-Aware Object-Based Video Processing. IEEE Proceedings," Vision, Image and Signal Processing, Vol. 152, No. 4, Aug. 2005, pp.398–406.

[37] Yong Shan, Fan Yang, and Runsheng Wang, "Color Space Selection for Moving Shadow Elimination," International Conference on Image and Graphics, Aug. 2007, pp. 496–501.

[38] M Kalaiselvi Geetha, Anandsankar B, Lakshmi S Nair, Amrutha T, and Amith Rajeev, "An improved Human Action Recognition system using RSD Code generation," Proc. International Conference on Interdisciplinary Advances in Applied Computing, Oct. 2014.

[39] Wei Shen, Xiang Bai, Xingwei Yang, and Longin Jan Latecki, "Skeleton Pruning as Trade-off Between Skeleton Simplicity and Reconstruction Error," Science China Information Sciences, Vol. 56, No. 4, Apr. 2013, pp. 1-14.

[40] Xiang Bai, Wenyu Liu, and Zhuowen Tu, "Integrating contour and skeleton for shape classification," Proc. IEEE 12th International Conference on Computer Vision Workshops, Oct. 2009, pp. 360-367.

[41] Christian Lovato, Umberto Castellani, and Andrea Giachetti, "Automatic Segmentation of Scanned Human Body Using Curve Skeleton Analysis," International Conference on Computer Vision / Computer Graphics Collaboration Techniques and Applications, 2009, pp. 34-45.

[42] Sadia Gul and Muhammad Faisal Khan, "Automatic Extraction of Contour Lines from Topographic Maps," International Conference on Digital Image Computing: Techniques and Applications, Dec. 2010, pp. 593-598.

[43] Porawat Visutsak and Korakot Prachumrak, "Geodesic-based Skeleton Smoothing," International Journal of Mathematical Models and Methods in Applied Sciences, Vol. 5, No. 4. 2011, pp. 713-721.

[44] H. Uemura, S. Ishikawa, and K. Mikolajczyk, "Feature tracking and motion compensation for action recognition," Proc. British Machine Vision Conference, Sep. 2008, pp. 30.1-30.10.

[45] Ross Messing, Chris Pal, and Henry Kautz, "Activity recognition using the velocity histories of tracked keypoints," Proc. IEEE International Conference on Computer Vision, Sep./Oct. 2009, pp. 104–111.

[46] Adrien Gaidon, Zaid Harchaoui, and Cordelia Schmid, "Recognizing activities with cluster-trees of tracklets," Proc. British Machine Vision Conference, Sep. 2012, pp. 30.1–30.13.

[47] Michalis Raptis and Stefano Soatto, "Tracklet descriptors for action modeling and video analysis," Proc. European Conference on Computer Vision, Sep. 2010, pp. 577–590.

[48] Michalis Raptis and Stefano Soatto, "Discovering discriminative action parts from mid-level video representations," Proc. IEEE Conference on Computer Vision and Pattern Recognition, Jun. 2012, pp. 1242–1249.

[49] Xiao Gang Wang, Keng Teck Ma, Gee-Wah Ng, and W. Eric L. Grimson, "Trajectory analysis and semantic region modeling using nonparametric hierarchical Bayesian models," International Journal of Computer Vision, Vol. 95, No. 3, Dec. 2011, pp. 287-312.

[50] Shan Dong Wu, Omar Oreifej, and Mubarak Shah, "Action recognition in videos acquired by a moving camera using motion decomposition of Lagrangian particle trajectories," Proc. International Conference on Computer Vision, Nov. 2011, pp. 1419-1426.

[51] Faisal I. Bashir, Ashfaq A. Khokhar, and Dan Schonfeld, “Object trajectory-based activity classification and recognition using hidden Markov models,” IEEE Transactions on Image Processing, Vol. 16, No. 7, Jul. 2007, pp. 1912–1919.

[52] Alexandre Hervieu, Patrick Bouthemy, and Jean-Pierre Le Cadre, “A statistical video content recognition method using invariant features on object trajectories,” IEEE Transactions on Circuits and System for Video Technology, Vol. 18, No. 11, Nov. 2008, pp. 1533-1543.

[53] Adeel Mumtaz, Weichen Zhang, and Antoni B. Chan, “Joint Motion Segmentation and Background Estimation in Dynamic Scenes,” Proc. IEEE Conference on Computer Vision and Pattern Recognition, Jun. 2014.

[54] Matthew Berger and Lee M. Seversk. “Subspace Tracking Under Dynamic Dimensionality for Online Background Subtraction,” Proc. IEEE Conference on Computer Vision and Pattern Recognition, Jun. 2014.

[55] Daniya Zamalieva, Alper Yilmaz, and James W. Davis, “A Multi-Transformational Model for Background Subtraction with Moving Cameras,” Proc. European Conference on Computer Vision, Sep. 2014.

[56] Jongwoo Lim and Bohyung Han, “Generalized Background Subtraction using Superpixels with Label Integrated Motion Estimation,” Proc. European Conference on Computer Vision, Sep. 2014, pp. 173-187.

[57] Ming Liang Chen, Qing Xiong Yang, Qing Li and Ming-Hsuan Yang, “Spatiotemporal Background Subtraction Using Minimum Spanning Tree and Optical Flow”, Proc. European Conference on Computer Vision, Sep. 2014, pp. 521-534.

[58] Atsushi Shimada, Hajime Nagahara, and Rin-ichiro Taniguchi, “Background Modeling Based on Bidirectional Analysis,” Proc. IEEE Conference on Computer Vision and Pattern Recognition. Jun. 2013.

[59] Wang Jiang Zhu, Shuang Liang, Yi Chen Wei, and Jian Sun. “Saliency Optimization from Robust Background Detection,” Proc. IEEE Conference on Computer Vision and Pattern Recognition, Jun. 2014, pp. 2814-2821.

[60] Dan Oneata, Jérôme Revaud, Jakob Verbeek, and Cordelia Schmid, "Spatio-Temporal Object Detection Proposals," Proc. European Conference on Computer Vision, Sep. 2014, pp. 737-752.

[61] Thomas Brox and Jitendra Malik, "Large Displacement Optical Flow: Descriptor Matching in Variational Motion Estimation," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 33, No. 3, Mar. 2011, pp. 500-513.

[62] Narayanan Sundaram, Thomas Brox and Kurt Keutzer, "Dense Point Trajectories by GPU-Accelerated Large Displacement Optical Flow," Proc. European Conference on Computer Vision, Jul. 2010.

[63] Sunil Chopra and M. R. Rao. "The Partition Problem," Mathematical Programming, Vol. 59, No. 1, Mar. 1993, pp. 87-115.

[64] Michel Marie Deza and Monique Laurent, "Geometry of Cuts and Metrics," Springer, 1997.

[65] Berthold K.P. Horn and Brian G. Rhunck, "Determining Optical Flow," Artificial Intelligence, Vol. 17, No. 1-3, Aug. 1981, pp. 185-203.

[66] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert, "High Accuracy Optical Flow Estimation Based on A Theory for Warping," Proc. European Conference on Computer Vision, May 2004, pp. 25–36.

[67] De Qing Sun, Stefan Roth, J. P. Lewis, and M. J. Black, "Learning Optical Flow," Proc. European Conference on Computer Vision, 2008, pp. 83–87.

[68] Ming Ming Cheng, Jonathan Warrell, Wen-Yan Lin, Shuai Zheng, Vibhav Vineet, and Nigel Crook, "Efficient Salient Region Detection with Soft Image Abstraction," Proc. IEEE International Conference on Computer Vision, Aug. 2013, pp. 1529-1536.

[69] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A Large Video Database for Human Motion Recognition," Proc. International Conference on Computer Vision, Nov. 2011.

[70] Jianbo Shi and Carlo Tomasi, "Good features to track," Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Jun. 1994.

[71] Jingen Liu, Yang Yang, and Mubarak Shah, “Learning Semantic Visual Vocabularies using Diffusion Distance,” Proc. IEEE International Conference on Computer Vision and Pattern Recognition, Jun. 2009.

[72] Christian Micheloni, Bernhard Rinner, and Gian Luca Foresti, “Video analysis in pan-tilt-zoom camera networks”, IEEE Signal Processing Magazine, Vol. 27, No. 5, Sep. 2010, pp. 78-90.

[73] Yu Gang Jiang, Qi Dai, Xiang Yang Xue, Wei Liu, and Chong-Wah. Ngo, “Human Action Recognition in Unconstrained Videos by Explicit Motion Modeling,” IEEE Transactions on Image Processing, Vol. 24, No. 11, Nov. 2015, pp. 3781-3795.

[74] Ali Elqursh and Ahmed Elgammal, “Online Moving Camera Background Subtraction,” Proc. European Conference on Computer Vision, Oct. 2012, pp. 228-241.

[75] Christian Micheloni, Lauro Snidaro, and Gian Luca Foresti, “Exploiting Temporal Statistics for Events Analysis and Understanding”, Image and Vision Computing, Vol. 27, No. 10, Sep. 2009, pp. 1459-1469.

[76] Claudio Piciarelli, Christian Micheloni, and Gian Luca Foresti, “Trajectory-based anomalous event detection”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 18, No. 11, Nov. 2008, pp. 1544-1554.

References of Webpages

[1] http://dcdv.zol.com.cn/422/4223196.html

[2] http://dcdv.zol.com.cn/426/4265347.html?from=bdshare

[3] https://zh.wikipedia.org/wiki/%E8%A7%92%E6%A3%80%E6%B5%8B#/me/media/File:

Corner.png