Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2013, Article ID 605469, 9 pages
http://dx.doi.org/10.1155/2013/605469

Research Article

Action Recognition by Joint Spatial-Temporal Motion Feature

Weihua Zhang,[1] Yi Zhang,[1] Chaobang Gao,[2] and Jiliu Zhou[1]

[1] *School of Computer Science, Sichuan University, Chengdu 610065, China*
[2] *College of Information Science and Technology, Chengdu University, Chengdu 610106, China*

Correspondence should be addressed to Yi Zhang; yzhang@scu.edu.cn

Received 4 March 2013; Accepted 28 April 2013

Academic Editor: Xiaoyu Song

This paper introduces a method for human action recognition based on optical flow motion features extraction. Automatic spatial and temporal alignments are combined together in order to encourage the temporal consistence on each action by an enhanced dynamic time warping (DTW) algorithm. At the same time, a fast method based on coarse-to-fine DTW constraint to improve computational performance without reducing accuracy is induced. The main contributions of this study include (1) a joint spatial-temporal multiresolution optical flow computation method which can keep encoding more informative motion information than recent proposed methods, (2) an enhanced DTW method to improve temporal consistence of motion in action recognition, and (3) coarse-to-fine DTW constraint on motion features pyramids to speed up recognition performance. Using this method, high recognition accuracy is achieved on different action databases like Weizmann database and KTH database.

1. Introduction

Human action recognition remains a challenge in computer vision due to different appearances of people, unsteady background, moving camera, illumination changes, and so on. Although impressive recognition accuracy has been achieved recently, computational efficiency of action recognition is relatively ignored. Especially, while the sample size in action database increases, numbers of frames-per-action get larger, and/or resolution gets higher, while the computational complexity will explode in most systems. Therefore, it is desired to develop a framework to maximally accelerate action recognition performance without sacrificing recognition accuracy significantly.

In previous researches, two types of approaches have been proposed. One is to extract features from video sequence and compare with preclassified features [1–4]. This category uses some voting mechanism to obtain better recognition performance and can adapt more variance by using large amount of training data. Another approach builds up a class of models from training set and computes the recognition rate that testing data related to these models [5–10]. Because these models have unique features, it may lose some characteristics of the feature. This approach is computationally efficient but its accuracy is not as good as the first approach. In practical application, while recognizing actions, a large set of good training data is needed to obtain high recognition accuracy. As the result, we should balance the trade between accuracy and computational cost.

In this paper, we focus on achieving a higher computational performance without sacrificing accuracy significantly and recognizing actions in a real environment. Our approach is based on the observation of optical flow of human actions proposed by Efros et al. [2] who used optical flow as motion features of actions [11]. We extract shape information of training data for accuracy, and in the final stage, we present an enhanced dynamic time warping algorithm to calculate the similarity of two actions. The k-NN voting mechanism and motion sequence pyramid are combined to achieve a better computation performance. Finally, spatial enhancement on k-NN pyramid and a coarse-to-fine DTW constraint are combined to get a computational efficiency without sacrificing accuracy obviously. The main contributions of this paper are (1) a joint spatial-temporal multiresolution optical flow fetching method which can keep more motion information than [2], (2) an enhanced DTW method to improve temporal consistence of motion in action recognition, and (3) coarse-to-fine DTW constraint on motion features pyramids to speed up recognition performance.

The rest of this paper is organized as follows. The rest of this section reviews the related works. Section 2 introduces the framework of our joint spatial-temporal motion feature extracting and action recognizing algorithm. Sections 3 and 4 describe the approaches in detail, and Section 5 shows experiment results. Finally, Section 6 concludes the whole work.

Due to the difficulty of the problem, simplified approaches using low-level features are usually applied. A number of approaches using features which describe motion and/or shape information to recognize human action were proposed. Gorelick et al. [1] extracted features from human shapes which are represented as spatial-temporal cubes by solving Poisson Equation. Cutler and Davis [3] presented period action recognition. Bradski and Davis [7] developed a fast and simple action recognition method using a timed motion history image (tMHI) to represent motions. Efros et al. [2] developed a generic approach to recognize actions of small scale figures using features extracted from smoothed optical flow estimation. Schüldt et al. [12] used SVM classification schemes on local space-time features for action recognition.

Recently, Ke et al. [13] proposed a novel method to correlate spatial-temporal shapes to video clips. Thurau and Hlaváč [14] presented a method for recognition of human actions based on pose primitives for both video frames and still images. Fathi and Mori [15] developed a method constructing mid-level motion features which were built from low-level optical flow information and used ada-boost as classifier. Lazebnik et al. [16] gave a simple and computationally efficient "spatial pyramid" extension to represent images. Laptev et al. [17] presented a new method of local space-time features, multichannel nonlinear SVMs, extended space-time pyramids method, and finally got good result on KTH dataset. Shechtman and Irani [18] introduced a behavior-based similarity measurement which was also based on motion features to detect complex behaviors in video sequences. Rodriguez et al. [19] introduced a template-based method for recognizing human actions based on a Maximum Average Correlation Height (MACH) filter. This method successfully avoided the high computational cost commonly incurred in template-based approaches. There have been other interesting topics about action recognition. One is done by Schindler and Van Gool [20] who discussed how many frames action recognition requires. Their approach uses less frames or only one frame of a sequence to obtain good recognition accuracy. Jhuang et al. [21] presented a biologically motivated system for the recognition of actions from video sequences. Their system consists of a hierarchy of spatial-temporal feature detectors of increasing complexity to simulate the way human recognizes an action.

2. Framework

The framework for our action recognition algorithm is shown in Figure 1.

In Step 1, an input video is preprocessed to get central aligned space-time volume of each action by human detection and tracking. In Step 2, optical flow descriptors are calculated and formed into jointing multiresolution pyramid features which will be discussed in Section 4.2. In Step 3, the action to action similarity matrix of features from testing video motion feature database is computed. Enhanced CFC-DTW algorithm is applied to calculate the similarity of these two actions to reduce computation time. Finally, the testing input video is recognized as one of the actions in training dataset.

First of all, our method is operated on a figure centric spatial-temporal volume extracted from an input image sequence. This figure centric volume can be obtained by running a tracking or detecting algorithm over the input sequence. The input of our recognition algorithm should be stabilized to ensure the center is aligned in space. In the proposed study, background subtraction to Weizmann action dataset and object tracking to KTH dataset are used as preprocessing.

As shown in Figure 2, in order to reduce the influence of noise, background is subtracted from the original video sequence, and the frames from result sequence are sent to optical flow calculation. Generally, it is difficult to get foreground-background well-separated video. Therefore background subtraction is not needed on testing data. Only human tracking algorithm is performed to detect the center and scale of a person. For benchmarking, we test two preprocess methods on testing data.

Once the stabilized centric volume has been obtained, spatial-temporal motion descriptors are used to measure similarity between different motions.

Firstly, optical flow is employed for each frame $f(i,j,t)$ by Lucas and Kanade [22] algorithm. The horizontal and vertical components of optical flow are split into two vector fields, F_x and F_y, each of which is half-waved to four nonnegative channels, F_{x^+}, F_{x^-}, F_{y^+}, F_{y^-}. To deal with the inaccuracy of optical flow computation on coarse and noisy data, Efros et al. [2] smooth and normalize four channels to $\widehat{F}b_{x^+}$, $\widehat{F}b_{x^-}$, $\widehat{F}b_{y^+}$, $\widehat{F}b_{y^-}$. Results are shown in Figure 3.

A similarity measure is proposed to compare the sequences of action A and B, which is defined based upon four normalized motion channels, that is, $\widehat{F}b_{x^+}$, $\widehat{F}b_{x^-}$, $\widehat{F}b_{y^+}$, $\widehat{F}b_{y^-}$. Specially, the i frame of sequence A is represented by a_1^i, a_2^i, a_3^i and a_4^i respectively. Therefore frame to frame similarity of frame j of sequence B to frame i of sequence A is

$$S(i,j)=\sum_{c=1}^{4}\sum_{x,y\in I} a_c^i(x,y)\, b_c^j(x,y). \tag{1}$$

In order to get smoother results of similarity matrix calculated by (1), convolution is performed with identity matrix I, and T denotes how many frames to smooth, which could improve the accuracy in dynamic time warping. Consider

$$S^T(i,j)=S(i,j)\otimes I(T). \tag{2}$$

In order to get more accuracy with the continuity feature of action sequence, an enhanced dynamic time warping

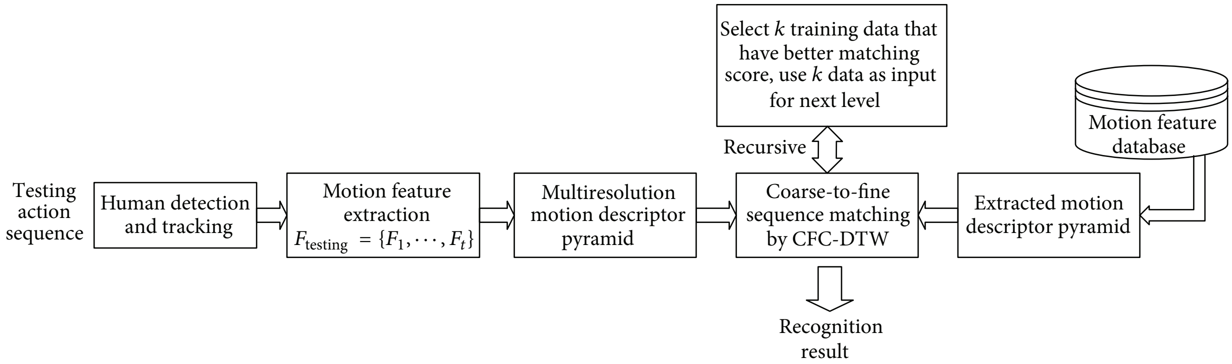

Select k training data that have better matching score, use k data as input for next level
Recursive
Motion feature database
Testing action sequence
Human detection and tracking
Motion feature extraction $F_{\text{testing}} = \{F_1, \cdots, F_t\}$
Multiresolution motion descriptor pyramid
Coarse-to-fine sequence matching by CFC-DTW
Extracted motion descriptor pyramid
Recognition result

Figure 1: Main process flow.

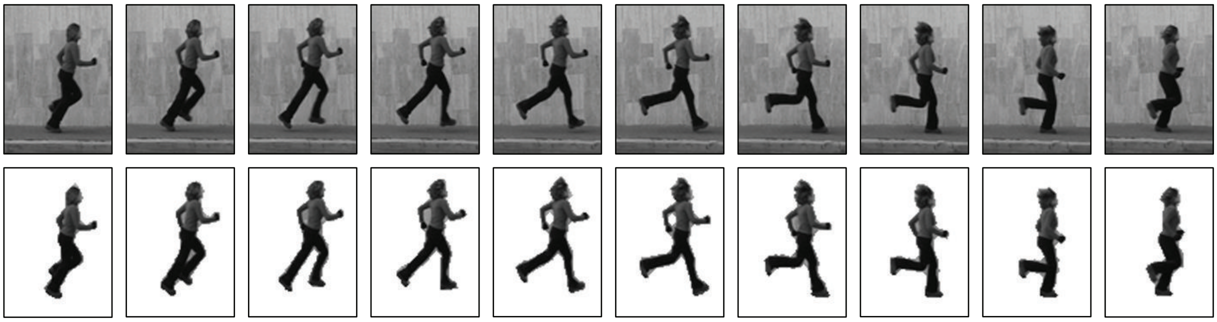


Figure 2: First row shows original video sequence for testing data (Lena_run1 from Weizmann dataset), and second row is background subtracted video sequence to calculate optical flow for training data.

algorithm is performed to find a matching path of two input actions in this paper, each point on this path represents a good matching pair, and all points are continuous in time domain. Similarity value on this path is summed up for similarity measurement.

When classifying an action, testing sequence is compared to preclassified sequence in lower resolution level of the feature pyramid. The best matching is chosen by action wide k nearest neighbor. And then this work is refined in a higher resolution level of the pyramid to choose the best matching by k nearest neighbor till the highest level of the pyramid has been compared.

Due to the complexity of action recognition problem, some actions are periodic while others are not. A single framework is developed to handle all these kinds of actions. A similarity measuring method based on DTW [23] and an enhanced DTW algorithm for action recognition is also introduced, which will be discussed in Section 3.

Finally, similarity between motion feature of testing action and preclassified database at the highest resolution level is calculated and the action with the best similarity score labeled the testing action.

3. Enhanced DTW for Action Recognition

While getting the similarity matrix of two actions, similarity matrix is generated from these data and the matrix can represent how much these two input actions are like each other. Previous research uses a frame to frame voting mechanism to get the similarity measure of two actions [2, 20]. For each frame in action A of testing data, k frames with the best matching score in all frames of training data are selected by voting. Although this simple selection of the best matching score in all of the frames should result in a better recognition rate, noise in some frames will cause a negative matching in action sequence. And due to the bad space alignment of action frames, same action gets a low similarity value but different actions higher. This k nearest neighbor algorithm has lack of a self-corrective mechanism which can keep the frame match continuity in time domain.

Differing from the frame to frame k nearest neighbor algorithm in [2], action to action similarity measurement is performed in our approach. This measurement calculates from frame to frame similarity matrix by summing up similarity values on the DTW path. This similarity measurement can be adaptive to speed variation of actions. Furthermore it keeps the continuity of frames in time domain. One frame can be correctly matched to another even if it does not have the highest matching score and it just lays on a DTW path, which will enhance the accuracy in action recognition. The demonstration of frame to frame similarity and action to action similarity is shown in Figure 4, and similarity measurement is defined according to

$$M_{\text{DTW}} = \sum_{[i,j]\in\text{path}} \frac{S^T(i,j)}{\text{length}}. \tag{3}$$

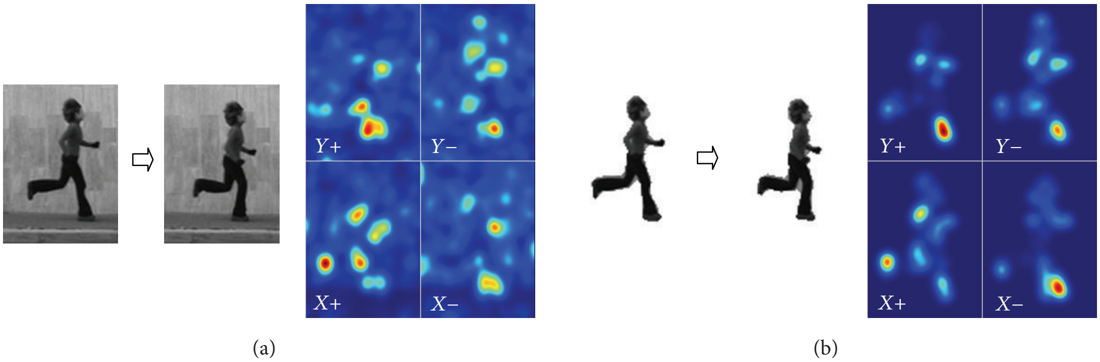


Figure 3: The optical flow descriptors of "Lena_run1." (a) Shows optical flow from original images, and (b) shows optical flow after background subtraction.

While using DTW in speech recognition area, constraints are added to the original algorithm for better recognition accuracy. Sakoe and Chiba gave their Sakoe-Chiba Band, and Itakura [24] shows the Itakura Parallelogram, and the latter is widely used in the speech recognition community. Since a spoken sentence always has a start position and an end position, applying these two constraints will get better alignment result and recognition result. Other approaches talk about processing of cyclic patterns on text matching or sequence matching [25]. The DTW algorithm is also widely used in signal processing area, such as finding waveform pattern of ECG [25, 26]. Recently, some researches in data mining use DTW as a sequence matching method and get inspiring achievement; they show their new boundary constraints and get good experiment result on video retrieval, image retrieval, handwriting retrieval, and text mining.

Previous work on DTW shows that better constraints on DTW get better recognition performance. Unlike general speech recognition, in action recognition, there are lots of periodical actions. Therefore, a new constraint should be found and be performed on the original DTW algorithm to adapt periodical actions and automatically align the start and end positions of actions.

While matching two actions, traditional DTW led a matching path on similarity matrix as shown in Figure 5(a). It looks like an actual path segment and two straight lines from the start point and to the end point. Since these two straight lines are not needed when calculating similarity value, a new method shown as Figure 6 was developed in the present study to get an accurate matching path as shown in Figure 5(b).

In our enhanced DTW algorithm, a constraint method called Coarse-to-Fine Constraint (CFC) is developed. This constraint can improve recognition speed of the $O(n^4)$ action recognition. Details about this algorithm will be discussed in the next section.

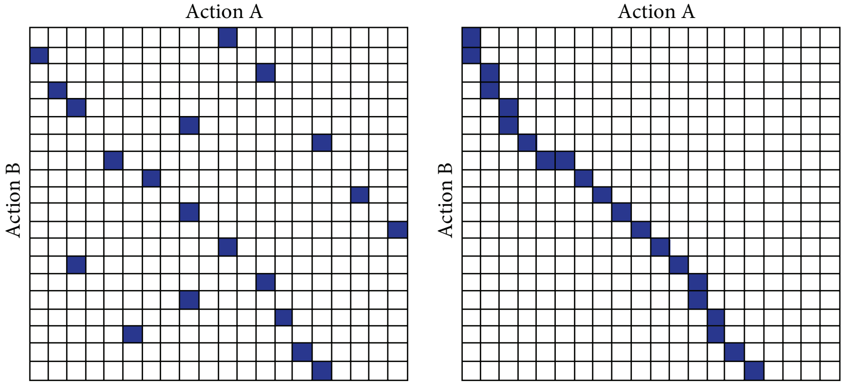
4. Multiresolution Coarse-to-Fine Matching

Similarity matrix calculation is a computationally expensive problem since only one element can be obtained in the action-action similarity matrix with multiplication frame by frame whose time complexity is $O(n^2)$. Therefore additional $O(n^2)$ calculating is needed to complete all of the elements in the similarity matrix. The matching method requires a total computations of $O(n^4)$ multiply operations. At the same time, when the training set gets bigger, more similarity calculations are needed. For example, processing all 93 videos in Weizmann dataset, using this similarity calculation will cost about 30 minutes in a 2.5 GHz Pentium E5200 Dual-Core computer. It is about 20 seconds average per recognition, which is an unacceptable performance while implementation. New methods should get better performance even if the training dataset have large amount of samples.

As mentioned in Figure 1, the main idea of this paper is comparing the similarity of two actions using multiresolution motion feature pyramid. Firstly, similarities were measured in low resolution level, then in a higher resolution level, till the highest level. In each of these Coarse-to-Fine comparison steps except for the highest level, actions that are selected by comparison only in lower resolution level are used as input of higher resolution level. That is, when comparing actions in low resolution level, actions that have the highest matching score in comparison results are selected by k nearest neighbor. These selected actions are used as the input for the higher resolution level in this multiresolution motion features pyramid.

At lowest resolution level, all of the actions in pre-classified were compared to the testing action, but the scales of these actions were very small. Therefore, the required computational effort is less than which compares the actions in their original scale. On the other hand, computation cost was higher in higher resolution level in the pyramid, but only a few actions in pre-classified database should be compared. The overall computational cost is decreased. This method achieved is more than 10 times faster than calculating the similarities in original resolution.

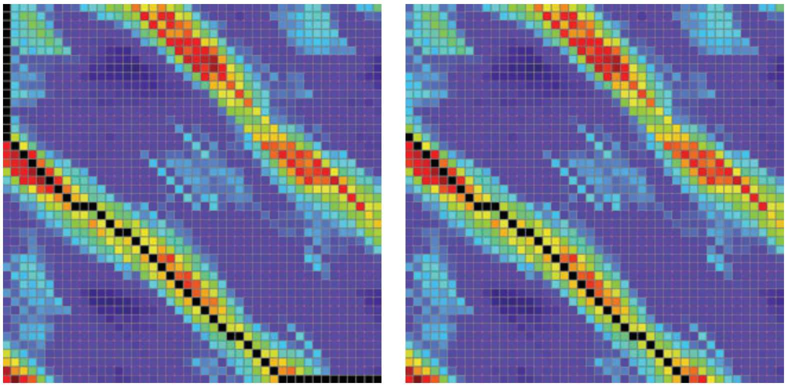
For performance purpose, a new DTW constraint is applied to the multiresolution motion feature pyramid. Each DTW matching path of similarity matrix is saved as a constraint for higher level. When calculating similarity matrix in



(a) Frame to frame nearest neighbor

(b) Action to action DTW path

Figure 4: Compare k-NN with DTW.



(a) DTW path

(b) autoaligned DTW path

Figure 5: DTW matching path of two actions. (a) Original DTW algorithm result. (b) Autoaligned DTW path shows matching start point and matching end point.

a higher resolution level, the saved path is convoluted with a kernel of 5 × 5 as

$$\begin{gathered} K = \left(k_{ij}\right)_{5\times 5}, \quad k_{ij} = 1, \\ i = -2, -1, \dots, 2, \;\; j = -2, -1, \dots, 2. \end{gathered} \tag{4}$$

The convoluted kernel will be used as a constraint in DTW algorithm. We name this constrained DTW as CFC-DTW.

4.1. Introduction of Gaussian Pyramid in Multiscale Images. In the field of image processing, Gaussian Pyramid [27] is widely used in image matching, image fusion, segmentation, and texture synthesis.

When obtaining multiscale images using the Gaussian Pyramid of each frame, low-pass filtering followed by subsampling for the images in previous levels can generate the Gaussian Pyramid shown in Figure 7. Each pixel value for the image at level L is computed as a weighted average of pixels in a 5 × 5 neighborhood at level $L-1$. Given the initial image $f_0(i,j)$ which has a size of $M \times N$ pixels, the level-to-level weighted average process is implemented by [27]

$$f_L(i,j) = \sum_{m=-2}^{2} \sum_{n=-2}^{2} r(m,n)\, f_L(2i+m, 2j+n), \tag{5}$$

where $r(m,n)$ is a separable 5×5 Gaussian low pass filter given by [23]

$$\begin{gathered} r(m,n) = r(m)\, r(n), \\ r(0) = a, \\ r(1) = r(-1) = \frac{1}{4}, \\ r(2) = r(-2) = \frac{1}{4} - \frac{2}{a}. \end{gathered} \tag{6}$$

Parameter a is set from 0.3 to 0.6 based on experiment results. The separation of $r(m,n)$ will reduce the computational complexity in generating multi-scale images.

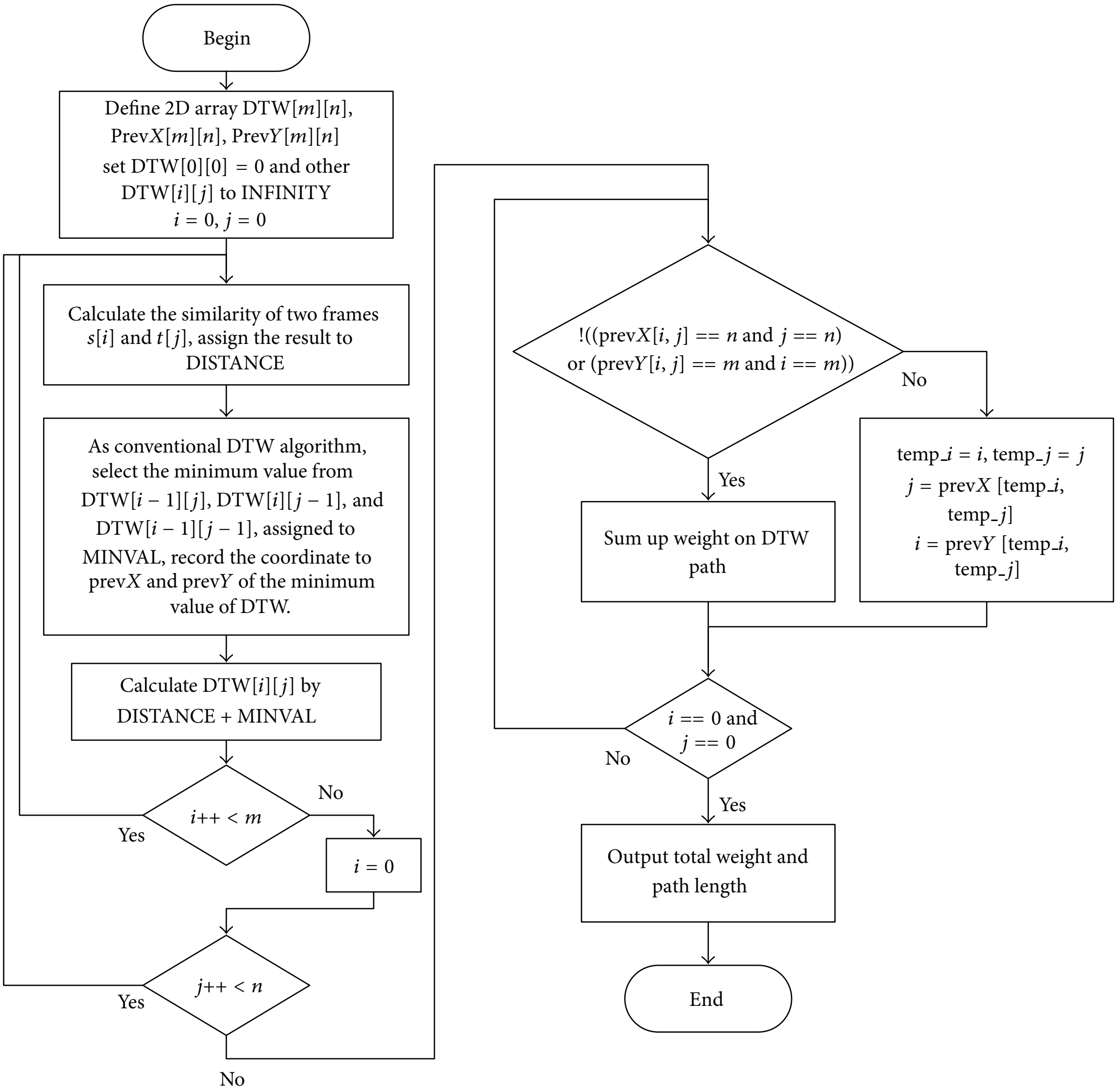

Begin
Define 2D array DTW[m][n], PrevX[m][n], PrevY[m][n]
set DTW[0][0] = 0 and other DTW[i][j] to INFINITY
i = 0, j = 0
Calculate the similarity of two frames s[i] and t[j], assign the result to DISTANCE
As conventional DTW algorithm, select the minimum value from DTW[i − 1][j], DTW[i][j − 1], and DTW[i − 1][j − 1], assigned to MINVAL, record the coordinate to prevX and prevY of the minimum value of DTW.
Calculate DTW[i][j] by DISTANCE + MINVAL
i++ < m
Yes
No
i = 0
j++ < n
Yes
No
!((prevX[i, j] == n and j == n) or (prevY[i, j] == m and i == m))
No
Yes
Sum up weight on DTW path
temp_i = i, temp_j = j
j = prevX [temp_i, temp_j]
i = prevY [temp_i, temp_j]
i == 0 and j == 0
No
Yes
Output total weight and path length
End

Figure 6: Flowchart of auto-aligned DTW.

4.2. Motion Sequence Pyramid. By extending Gaussian Pyramid, multiresolution coarse-to-fine similarity computing algorithm is introduced in the current study to reduce computation complexity.

Each pyramid has L level and every level relates to a scaling of original frame. The lowest level in this motion sequence pyramid has motion feature images with original size while the higher level images have smaller scales than that of originals.

In training, all L level pyramids of motion sequence descriptor in training database are stored as pre-classified actions database for similarity calculation. Consider

$$f_L(i,j,t) = \sum_{m=-2}^{2} \sum_{n=-2}^{2} r(m,n)\, f_{L-1}(2i+m, 2j+n, t), \quad (7)$$

where $f_L(i,j,t)$ denotes the image $f(i,j)$ on level L at frame t, as Figure 8.

It is obvious that computing similarity between two motion sequence pyramids at every level is not needed because L_0 has the most feature information. Performing equation (3) on level L_0 can get good recognition rate. But the frame size is so big that the computational cost is very high.

For performance purpose, multilayer classification starts from the lowest resolution level L_n. This resolution decreased motion sequence recognition can get acceptable classification result that actions with big difference are separated apart like walk and waving hand, run and bend, jogging, and boxing. This result can be used as the input for a higher resolution level. After getting the classification result in a lower resolution level L_s, select k actions with from high M_{DTW} to low and use this selected actions as the input of a higher resolution level L_{s-1} classification. This refinement is repeated until the highest resolution level L_0 is reached. Value of s can be chosen as $s = 1/(L+1)^2$; this value can also be found in a cross-validation.

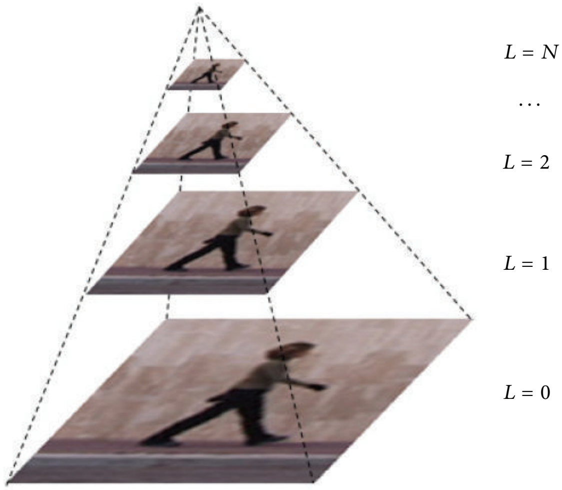


Figure 7: The Gaussian Pyramid.

Our results show that for the balance of computational performance and recognition accuracy, two levels of pyramid can get satisfied result, and the reason will be further discussed in experimental results section.

4.3. Coarse-to-Fine DTW. When searching for the best action matching in coarse-to-fine motion sequence pyramid, CFC-DTW is performed to accelerate calculation performance. First, in CFC-DTW, similarity matrix $S^T_{L_n}$ of two actions is calculated using (2) on lowest resolution level. When performing algorithm 1 on $S^T_{L_n}$, a 2D array denoting a DTW path with all elements on path equaling 1 is given as

$$T_{L_n}(i,j) = \begin{cases} 1 & S^T_{L_n}(i,j) \in \text{path}, \\ 0 & S^T_{L_n}(i,j) \notin \text{path}. \end{cases} \tag{8}$$

Secondly, as shown in Figure 9, convoluting $T_{L_n}(i,j)$ with kernel K leads to a coarse-to-fine constraint for the higher resolution level L_{n-1}, and a 5×5 convolution kernel of rectangle has been used in our work. Consider

$$\text{constraint}_{L_{n-1}}(i,j) = T_{L_n}(i,j) \otimes K. \tag{9}$$

Due to $\text{constraint}_{L_{n-1}}(i,j)$, the computation complexity of $S^T_{L_{n-1}}$ decreased. This coarse-to-fine constraint saves computation time from frames by frames multiplication.

5. Experimental Results

5.1. Dataset. We evaluated our approach on public benchmark datasets, Weizmann human action dataset [1], and KTH action dataset [12].

The Weizmann dataset contains 93 low-resolution (188 × 144 pixels, 25 fps) video sequences showing 9 persons performing a set of 10 different actions: bending down, jumping jack, waving one hand, waving two hands, in place jumping, jumping, siding, running, and walking. Background subtraction is used to get shape information and optical flow features of actions.

Table 1: Experiment on Weizmann dataset.

Method		Recognition rate	Recognition time
Enhanced DTW	Original image size	**100%**	5.33 s
CFC-DTW	Level 1: 30% size Level 2: 100% size $K = 3$	97.8%	1.17 s
	Level 1: 30% size Level 2: 100% size $K = 5$	98.9%	1.30 s
	Level 1: 20% size Level 2: 50% size $K = 10$	98.9%	**0.55 s**

The KTH dataset contains 25 persons acting 6 different actions: boxing, hand-clapping, jogging, running, walking, and hand-waving. These actions are recorded under 4 different scenarios: outdoors (s1), outdoors with scale-variations (s2), outdoors with different appearance (s3), and indoors (s4). Each video sequence can be divided into 3 or 4 subsequence for different direction of jogging, running, and walking. Human tracking method was used to get centric volume of these actions as pre-process. Background subtraction was not applied in this case. Results were compared with previous works.

Leave-one-out mechanism was used in the experiments. Each testing action had been compared to other 92 actions in the dataset. A total recognition rate and an average recognition time of each algorithm were evaluated. All methods mentioned in this paper were combined to the joint spatial-temporal feature to perform recognition.

The hardware environment is composed of a Windows 7 PC with 2.5 GHz Pentium E5200 Dual-Core CPU and 2 G Bytes system memory.

5.2. Results. Results of multiresolution coarse-to-fine pyramid method on Weizmann dataset are shown in Table 1. For computation efficiency, a two-level pyramid was built and different resolution reductions of each level were used. Recognition rate and average recognition time per action are shown in Table 1. K means numbers of input actions from lower resolution level to higher resolution level. CFC-DTW was used in this experiment at the same time.

Experiment result in Table 2 shows that our enhanced DTW algorithm can get 100% recognition rate with all frames calculated. By CFC-DTW acceleration, the recognition is 10 times faster than enhanced DTW and still gets acceptable recognition rate. In Table 1, 0.55-second period means that the CFC-DTW algorithm can be used in practical applications.

On KTH dataset, our approach obtained the best result in s2 comparing to [20, 21] (see Table 3). Videos in this scenario were captured from outdoors environments and with camera zoom-in and zoom-out, because the CFC-DTW kept the continuity of each frame in action sequence while matching. If one frame is not matched, it had always been corrected by nearly frames. The average recognition time was near 3 s

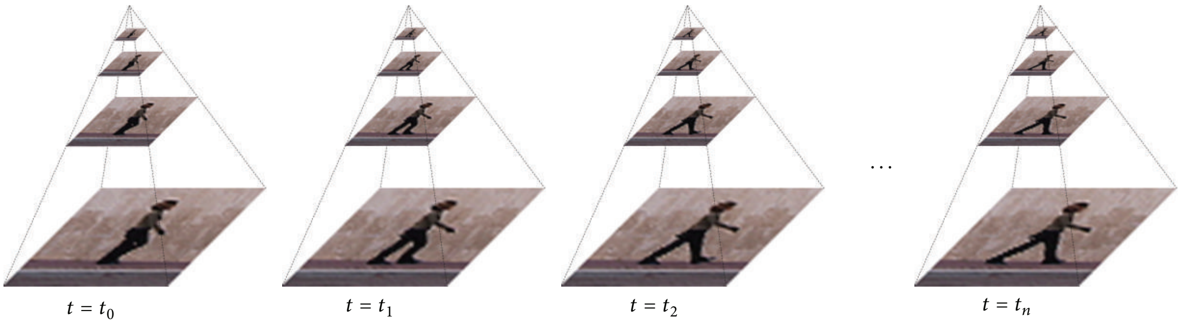
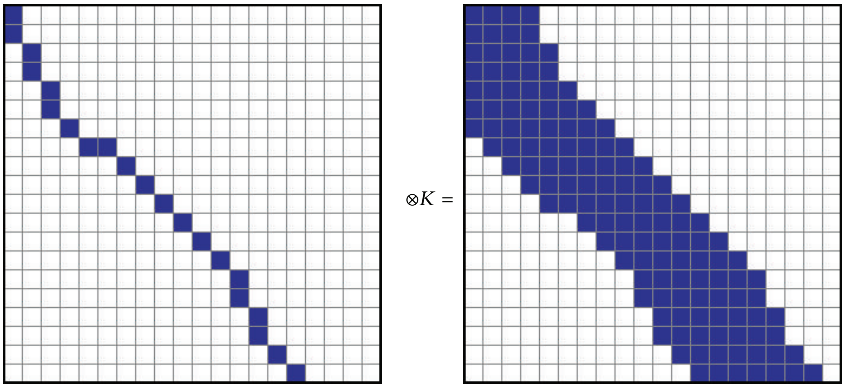


Figure 8: Motion sequence pyramid $f(i, j, t)$.



(a) DTW path of lower level

(b) DTW constraint on higher level

Figure 9: Coarse-to-fine DTW.

Table 2: Compared with other approaches on Weizmann dataset.

Method	Recognition rate	No. of frames
Gorelick et al. PAMI'07 [1]	93.5% 96.6% 99.6%	2 3 10
Schindler and Van Gool CVPR'08 [20]	93.5% 96.6% 99.6%	2 3 10
Jhuang et al. ICCV'07 [21]	97.8%	All
Fathi and Mori CVPR'08 [15]	100.0%	All
Our approach	100.0%	All

Table 3: Experiment on KTH dataset.

	Schindler and Van Gool CVPR'08 [20] SNIPPET 1/SNIPPET 7	Jhuang et al. ICCV'07 [21]	Our approach
s1	90.9%/93.0%	**96.0%**	94.0%
s2	78.1%/81.1%	86.1%	**88.3%**
s3	88.5%/**92.1%**	89.8%	91.4%
s4	92.2%/**96.7%**	94.8%	93.8%

in our testing platform. This performance can be improved by multicore technology and GPU computation for real-time purpose.

6. Conclusion

We present a fast action recognition algorithm with enhanced CFC-DTW. Although DTW is a time-consuming method, the proposed CFC-DTW and motion pyramid significantly speed up the traditional DTW method. Therefore, real-time recognition becomes possible in practice. Because the DTW can align the continuity of action, even low resolution videos can get an acceptable recognition rate. Furthermore, the algorithm developed in this study can be applied to thin client communication environment, since the coarse-to-fine feature of CFC-DTW can fit the requirement of action recognition in the environment [28–30], and modal data can be transferred in different level based on requirements.

Acknowledgments

This work is supported by the Grants from the Foundation of Sichuan University Early Career Researcher Award (2012SCU11036, and 2012SCU11070) and Plan for Science & Technology Support of Sichuan Province (2012RZ0005).

The authors are grateful to Dr. Linlin Zhou for her valuable suggestions on the writing of this paper.

References

[1] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, “Actions as space-time shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2247–2253, 2007.

[2] A. A. Efros, A. C. Berg, G. Mori, and J. Malik, “Recognizing action at a distance,” in *Proceedings of the 9th IEEE International Conference on Computer Vision*, pp. 726–733, October 2003.

[3] R. Cutler and L. S. Davis, “Robust real-time periodic motion detection, analysis, and applications,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 781–796, 2000.

[4] A. Yilmaz and M. Shah, “Actions sketch: a novel action representation,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR ’05)*, vol. 1, pp. 984–989, June 2005.

[5] S. Ali, A. Basharat, and M. Shah, “Chaotic invariants for human action recognition,” in *Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV ’07)*, pp. 1–8, October 2007.

[6] I. Laptev and P. Pérez, “Retrieving actions in movies,” in *Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV ’07)*, pp. 1–8, October 2007.

[7] G. R. Bradski and J. W. Davis, “Motion segmentation and pose recognition with motion history gradients,” *Machine Vision and Applications*, vol. 13, no. 3, pp. 174–184, 2002.

[8] A. Yao, J. Gall, and L. Van Gool, “Coupled action recognition and pose estimation from multiple views,” *International Journal of Computer Vision*, vol. 100, no. 1, pp. 16–37, 2012.

[9] Z. Jiang, Z. Lin, and L. S. Davis, “Recognizing human actions by learning and matching shape-motion prototype trees,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 533–547, 2012.

[10] A. Oikonomopoulos, I. Patras, and M. Pantic, “Spatiotemporal localization and categorization of human actions in unsegmented image sequences,” *IEEE Transactions on Image Processing*, vol. 20, no. 4, pp. 1126–1140, 2011.

[11] R. Poppe, “A survey on vision-based human action recognition,” *Image and Vision Computing*, vol. 28, no. 6, pp. 976–990, 2010.

[12] C. Schüldt, I. Laptev, and B. Caputo, “Recognizing human actions: a local SVM approach,” in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR ’04)*, vol. 3, pp. 32–36, August 2004.

[13] Y. Ke, R. Sukthankar, and M. Hebert, “Spatio-temporal shape and flow correlation for action recognition,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR ’07)*, pp. 1–8, June 2007.

[14] C. Thurau and V. Hlaváč, “Pose primitive based human action recognition in videos or still images,” in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR ’08)*, pp. 1–8, June 2008.

[15] A. Fathi and G. Mori, “Action recognition by learning mid-level motion features,” in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR ’08)*, pp. 1–8, June 2008.

[16] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: spatial pyramid matching for recognizing natural scene categories,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR ’06)*, vol. 2, pp. 2169–2178, June 2006.

[17] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, “Learning realistic human actions from movies,” in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR ’08)*, pp. 1–8, June 2008.

[18] E. Shechtman and M. Irani, “Space-time behavior-based correlation—OR—how to tell if two underlying motion fields are similar without computing them?” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 11, pp. 2045–2056, 2007.

[19] M. D. Rodriguez, J. Ahmed, and M. Shah, “Action MACH: a spatio-temporal maximum average correlation height filter for action recognition,” in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR ’08)*, pp. 1–8, June 2008.

[20] K. Schindler and L. Van Gool, “Action snippets: how many frames does human action recognition require?” in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR ’08)*, pp. 1–8, June 2008.

[21] H. Jhuang, T. Serre, L. Wolf, and T. Poggio, “A biologically inspired system for action recognition,” in *Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV ’07)*, pp. 1–8, October 2007.

[22] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI ’81)*, April 1981.

[23] H. Sakoe and S. Chiba, “Dynamic programming algorithm optimization for spoken word recognition,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.

[24] F. Itakura, “Minimum prediction residual principle applied to speech recognition,” *IEEE Transactions on Speech and Signal Processing*, vol. 23, no. 1, pp. 67–72, 1975.

[25] F. Wang, T. Syeda-Mahmood, and D. Beymer, “Finding disease similarity by combining ECG with heart auscultation sound,” in *Proceedings of the Computers in Cardiology (CAR ’07)*, pp. 261–264, October 2007.

[26] P. Laguna, R. Jane, and P. Caminal, “Automatic detection of wave boundaries in multilead ECG signals: validation with the CSE database,” *Computers and Biomedical Research*, vol. 27, no. 1, pp. 45–60, 1994.

[27] P. J. Burt and E. H. Adelson, “The Laplacian pyramid as a compact image code,” *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 532–540, 1983.

[28] J. B. Wang, M. Chen, X. Wan, and C. Wei, “Ant-colony-optimization-based scheduling algorithm for uplink CDMA nonreal-time data,” *IEEE Transactions on Vehicular Technology*, vol. 58, no. 1, pp. 231–241, 2009.

[29] J. Wang, Y. Jiao et al., “Optimal training sequences for indoor wireless optical communications,” *Journal of Optics*, vol. 14, no. 1, Article ID 015401, 2012.

[30] J. Wang, X. Xie, Y. Jiao et al., “Optimal odd-periodic complementary sequences for diffuse wireless optical communications,” *Optical Engineering*, vol. 51, no. 9, Article ID 095002, 6 pages, 2012.