TELKOMNIKA, Vol. 11, No. 4, April 2013, pp. 1871~1876
ISSN: 2302-4046

Boost Action Recognition through Computed Volume

Li Wang*[1], Ting Yun[2] , Haifeng Lin[3]
[1,2,3]Nanjing Forest University
No. 159 Longpan Road, Nanjing, 210037, Jiangsu, China
*Corresponding author, e-mail: wang.li.njfu@gmail.com

Abstract

We detect interest points in temporal-spacial space and use the local feature plus their positions to recognize action in a video. Although some previous methods take advantage of the position of each interest point besides the local feature of them, and achieve good performance, it consumes much time to position these points due to the complexity of an action. We propose two simple methods to position each interest point, and design a new feature for action recognition. Evaluation of the approach on two sets of videos suggests its effectiveness.

Keywords*: computer vision, action recognition, interest point*

1. Introduction

Action recognition has attract much interest due to its' potential application. There have existed some kinds of features created from 3D dimension space including temporal and spacial space, such as Cuboids [1], HOG/HOF [2], HOG3D [3], and Extended SURF [4]. These features are composed of local features extracted from some temporal-spacial positions. A simple way using them is to cast them into codewords and create a statistical histogram vector as their feature. The so called bag-of-words (BoW) method has achieved good performance on some data set such as KTH [5]. From that, integrating BoW and temporal-spacial layout of interest point make some progress, such as Spatial Pyramid Matching [6], Implicit Shape Model [7, 8]. We focus on finding temporal-spacial layout of interest points in this paper.

In action recognition, we want to create an axis or a volume in spacial-temporal space to get each local features. The center of the space is limited inside of a human body. We can detect human body or manually annotate them using a bounding box whose center can be treated as the center of the space. Detecting people is not easy, but still have some efficient methods such as [8-10]. After composing all detected bounding boxes and aligning them, a volume containing a human body will be created. Although it seems obvious to use the method for creating volume, some problems still existed, such as unefficient detection, object missed and varied scale of detected objects. So we propose a simple method to get the volume from the interest points themselves, and create a feature combining BoW and positions of them. We found it can improve the recognition accuracy without spending much time on detection or annotation.

2. Creating Volume

When an action is recorded in a video, there always exist some variations in the raw data of video along the temporal or spacial axis. After detecting these variations in spacial-temporal space, we can get their positions and extract their local features around them. There have some features recently, such as Cuboids, HOG/HOF and so on. Although they can work well on some data such as KTH, they ignored temporal-spacial layout information. To take advantage of the information, we first create a volume containing these BoW, and then create a histogram vector with BoW based on their spacial-temporal layout.

In the next, we show how we create the volume using two different regression methods.

Received January 3, 2013; Revised February 12, 2013; Accepted February 24, 2013

2.1. Fitting Straight Line

The spacial-temporal volume can be described in 3D space. Each point's position in the space can be represented as (x, y, t). At each time t which can also be identified as frame index, there is a frame which may contain a point at (x, y) position. When detecting these points caused by a movement of a person, we got their traces which are very complex. People not only perform global movements but also local movements. For simple actions, we can model global movements of a point as:

$$\begin{cases} x = v_x t + x_0 \\ y = v_y t + y_0 \\ z = v_z t + z_0 \end{cases} \quad (1)$$

(x_0, y_0, z_0) is the initial position of a point. (v_x, v_y, v_z) is the speed of a point. z is actually the frame index of the point. We choose the average point of all points as the center of volume, and subtract that from each pint position x, y, z. The equation 1 can be converted to:

$$\begin{cases} x = v_x t \\ y = v_y t \\ z = v_z t \end{cases} \quad (2)$$

In fact, we can only describe the position of human body in 2D space through an image. So we only get $v_x$and v_y . When their value is 0, the people usually stand still and the detected points have local movements without global movements. We also know that a point will move along temporal space, even though it doesn't move in spacial space. So it still got some different positions along the temporal space, and we still need to assign a constant speed v_z to a point. This simple model can be used well in KTH. There are 6 type of actions which are partly displayed in Figure 1.

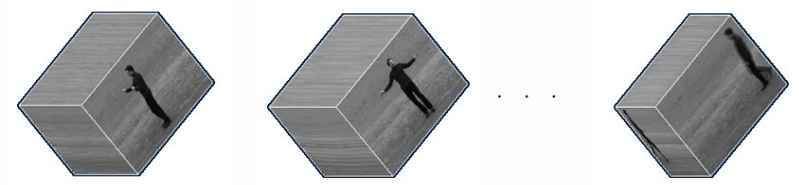


Figure 1. Some Sample Actions of KTH

The people in KTH usually moves in constant speed or stand still. Because all points have the same global movements, we can use linear regression to compute (v_x, v_y) which can be used as orientation of volume containing an action. A problem is that we need to find two points which belong to the same position of a body. So we can compute the speed of the point and divide the speed to two different kinds of speed: one is global speed, and the other is local speed. We assume speed for each point as $v = v_{global} + v_{local}$, and there are n positions which have a detected speed at an instant time t. Because the sum of all local speed will be zero due to their movement around body, we can compute the global speed at the time t as $v^t_{global} = \frac{1}{n}\sum_{i=1}^{n} v^t_i$. So the global speed can be computed as follow:

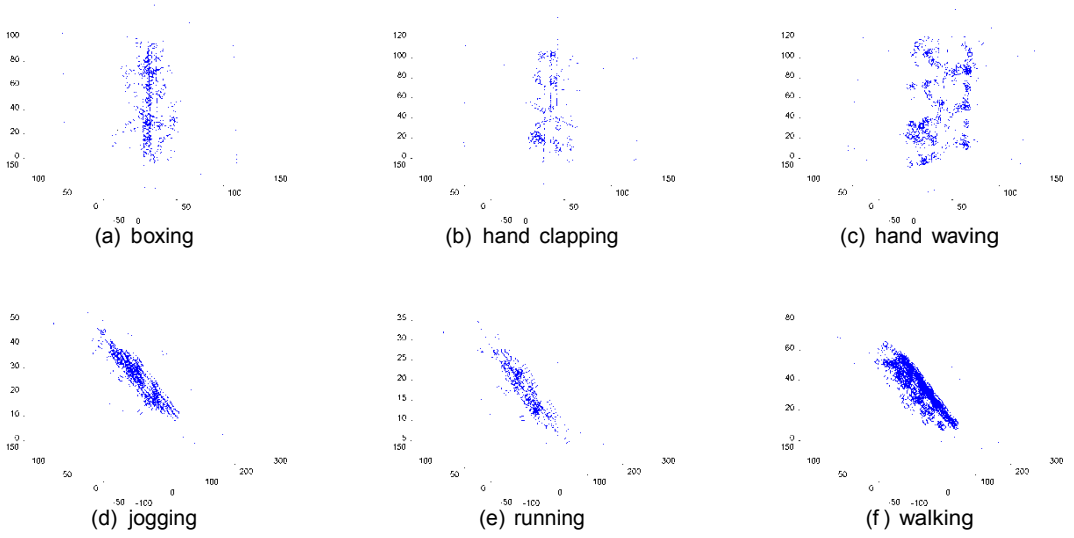
$$\begin{aligned} v^t_{global} &= \frac{1}{n}\sum_{i=1}^{n} v^t_i \\ &= \frac{1}{n}\sum_{i=1}^{n}(p^{t+1}_i - p^t_{map(i)}) \\ &= \frac{1}{n}\sum_{i=1}^{n} p^{t+1}_i - \frac{1}{n}\sum_{i=1}^{n} p^t_{map(i)} \end{aligned} \quad (3)$$

p_i^{t+1} is the i point that we detected at $t+1$ time; $map(i)$ is a point that map from i, and has similar local feature with i point; moreover, we assume they belong to the same part of a body. So we convert a difficult problem to a simple problem. We only need to compute the speed of a center point which is the center of all points during a fixed time, and get the trace of the body, and create a volume based on the trace. The center point is computed as the average locations of all points as Figure 2 displayed.

(a) Boxing (b) Gaussian Distribution

Figure 2. Each Center is Averaged by All Points on Several Sequential Frames. All Center Points Composed the Curve Displayed in the Right

Next, we use linear regression to get the speed vector of the center point. Although not all center points can fit to a line, their total error is minimized through linear regression. We use singular value decomposition(SVD) to get the speed vector. If there are m points in a clip, we can combine them into a m × 3 matrix in which each row is the position x, y, z of a point. We decompose the matrix to M = U ΣV through SVD and use the first column of V as the speed (v_x, v_y, v_z) which can be used to compute the center of each frame. The volume's size can be fixed size or equal to the size of a hull which contains all points. We use fixed size in KTH and get the volumes of different actions, and display them in Figure 3.



(a) boxing (b) hand clapping (c) hand waving

(d) jogging (e) running (f) walking

Figure 3. Use Linear Regression to Create Volumes of Different Types of Actions.

2.2. Fitting Parabola

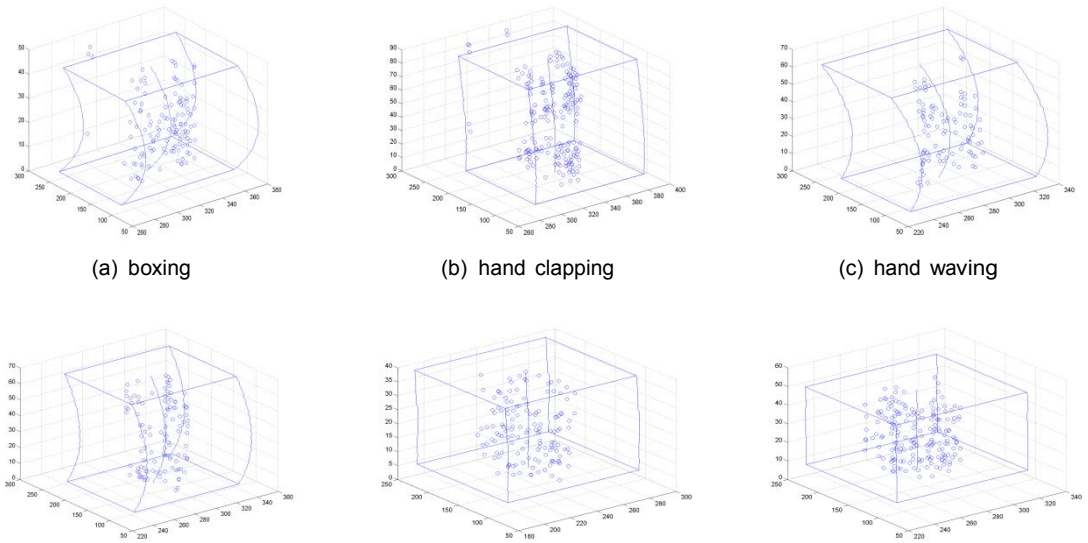
We assume the speed of a person in KTH is constant. But it's not true in any time. We have created a data for detection of sneeze and cough actions. For the data, we need to

recognize the sneeze or cough actions. When a people want to sneeze or cough, he usually stop before doing that.

So we use parabola regression to get the track of the person considering the acceleration in an action. The parabola equation can be wrote as :

$$\begin{cases} x = a_x t^2 + v_x t + x_0 \\ y = a_y t^2 + v_y t + y_0 \end{cases} \quad (4)$$

Because we add acceleration and use parabola regression, the accuracy of created tracks should be more accuracy than the ones from linear regression. Through least squares, we can get $\hat{x} = (a_x, v_x, x_0)$ and $\hat{y} = (a_y, v_y, y_0)$ which can be used to compute parabola volume which is also the track of a person. Because each frame is captured in a constant duration, we use interval between two sequential frames as fundamental time unit. If the time of first frame is $t = 1$, we can treat the time of the ith frame as $t = i$. If there are m frames, we can create a $m \times 3$ matrix A whose ithrow is $[1, t_i, t_i^2]$. To get the best parabola function, We should choose $\hat{x}$ to satisfy $A^T A\hat{x} = A^T b$, and so do $\hat{y}$. After getting $\hat{x}$ and $\hat{y}$, we can get the track of the virtual center point. Then we create a fixed size bounding box based on each center point, and get the volumes like Figure 4 displayed.



(a) boxing (b) hand clapping (c) hand waving

(d) jogging (e) running (f) walking

Figure 4. Use Parabola Regression to Create Volumes of Different Types of Actions

3. Feature with Space-time Layout

As recent works [2, 5] show that local feature can be used efficiently in describing human action. A simple way to use local feature is to classify them as different codewords and map them to a history vector. Although their spacial-temporal locations are ignored, Laptev [2] still got impressive performance on image and action analysis using BoW method. We think that the location of local feature is also an important description of actions. Integrating BoW and spacial-temporal location of local features should achieve better results than only using BoW.

If the number of codewords is$K$, we can create a histogram vector by assigning each local feature a codeword. To take advantage these features' position, we compute distance between each pair of local features in a volume. The distance will be quantized to S unit. Each pair of codewords with a specified distance will be counted and a vector of length $K \times K \times S$ will be created at last whose value is the accumulated number. In practice, we further

summarized it into a $2K \times S$ vector. A K-dim vector is extracted from the diagonal element, and another K-dim is obtained by summoning over all the off-diagonal elements row-wise.

4. Experiments

We use support vector machine (SVM) for action recognition through LibSVM. The local feature used in the following experiments is all from Cuboid feature. Then we did some experiments on KTH and sneeze-cough (SC) data separately to prove the effectiveness of our method.

4.1. Accuracy Measures

There is a standard accuracy measure which is the agreement between a measured quantity and the true value of that quantity, and can be computed by confusion matrix. For binary classification, this becomes (TP+TN)/ALL, where TP, TN, FP, and FN are True Positive, True Negative, False Positive and False Negative respectively, and ALL=(TP+TN+FP+FN). Nevertheless, they can cause problem for datasets with imbalanced distributions among different categories. For Sneeze-Cough dataset, 3/4 of the Sneeze-Cough belongs to background actions category. If we use the standard accuracy measure, a rate of 75% is reached when a classifier treats each sample to background action. This leads to the utilization of precision and recall, which are computed by TP/ (TP+FP) and TP/ (TP+FN) respectively. In this paper, we adopt a different accuracy measure using TP/ (TP+FP+FN) for this binary classification task, which can be regarded as a lower-bounding summary of the (precision, recall) pair.

4.2. KTH

KTH data contains 2392 video clips in which 25 actors perform 6 different type actions under 4 different contexts. These actions include boxing, hand clapping, hand waving, jogging, running, and walking. We use the same split scheme of [5] and get results as Table 1 shows. We see that the result of BoW plus parabola is the best, because using parabola regression achieves better accuracy volume.

Table 1. Combine BoW with Volume or No Volume to get the Results on KTH.

Feature	BoW	BoW+straight line regression	BoW+parabola regression
Accuracy	89.0%	90.3%	90.7%

4.3. Sneeze-Cough

We create a new Sneeze-Cough video dataset that tailors to the specific challenges and characteristics of recognizing flu-like behavior symptoms in public areas. This dataset contains 960 color video clips collected from 20 human subjects. The data acquisition process is carried out in an indoor environment with semi-controlled lighting condition. Each clip contains one specific action performed by one subject in a particular view and pose. Video shots are normalized to 480x290 resolutions, with stream rate of 5 frames per second, and each lasts for around 15 seconds. In addition to the two flu-like behaviors, namely sneeze and cough, six common background action types are also included: drinking, phone calling, scratching head, stretching arms, wiping glasses and waving hands. Each person performs each type of action six times under 2 different poses: standing and walking, and 3 different views: roughly frontal/left/right. We also perform horizontal flip on each video to produce an additional video set with reflective views. So we get a set of 1920 videos finally.

We use 15 persons for training, and 5 persons left for testing. So we have 480 clips for testing and the other clips for training. Although there're 8 types of action in all, our objective is to differentiate the flu-like behaviors from the other actions. So it becomes a binary recognition problem. For the imbalance data problem, we use TP/ (TP+FP+FN) to compute accuracy as we mentioned above.

From the results showed in Table 2, we achieve the worst results using BoW directly. Using BoW plus volume created by parabola regression get better results. To prove our

algorithm effectiveness, we also annotate each bounding box around the objective in each frame and combine them into a volume. It gets the best results, but it consumes much time on annotation.

Table 2. Combine BoW with Volume or No Volume to get the Results on Sneeze-Cough Data.

Feature	BoW	BoW+parabola regression	manually annotated volume
Accuracy	26.1%	27.4%	29.6%

5. Conclusion

To take advantage of the positions of local features, we need to get a volume to compute the coordinate of the local features. Although the volume can be created by each bounding box in each frame, it will consume much time and lead to difficulity of experiment if we manually annotated these bounding boxes. So we created volume automatically based on these local features directly. The created volume not only improve the accuracy of recognition, but also can be applied in application easily.

Acknowledgements

This work was financially supported by the Jiangsu Natural Science Foundation (BK2012418).

References

[1] I Laptev, T Lindeberg. *Space-time interest points. In Computer Vision.* Proceedings. Ninth IEEE International Conference on. 2003; 1: 432-439.
[2] I Laptev, M Marszalek, C Schmid, B Rozenfeld. *Learning realistic human actions from movies*. In CVPR. 2008.
[3] Alexander Klaser, Marcin Marszalek, Cordelia Schmid*. A spatio-temporal descriptor based on 3d-gradients*. In BMVC08.
[4] Geert Willems, Tinne Tuytelaars, Luc Gool. *An efficient dense and scale-invariant spatio-temporal interest point detector*. In Proceedings of the 10th European Conference on Computer Vision: Part II. 2008; 650–663.
[5] C Schuldt, I Laptev, B Caputo. *Recognizing human actions: A local svm approach*. In ICPR, 2004.
[6] S Lazebnik, C Schmid, J Ponce. *Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories*. In CVPR, 2006.
[7] Bastian Leibe, Ales Leonardis, and Bernt Schiele. *Combined object categorization and segmentation with an implicit shape model*. In In ECCV workshop on statistical learning in computer vision. 2004; 17–32.
[8] TH Thi, L Cheng, J Zhang, L Wang, S Satoh. Integrating local action elements for action analysis*. Computer Vision and Image Understanding.* 2011.
[9] N Dalal, B Triggs. *Histograms of oriented gradients for human detection. In Computer Vision and Pattern Recognition*, CVPR 2005. IEEE Computer Society Conference. 2005; 1: 886–893.
[10] Pedro Felzenszwalb, David McAllester, Deva Ramanan. *A discriminatively trained, multiscale, deformable part model*. IEEE Conference on Computer Vision and Pattern Recognition. 2008; 9: 1-8.