A reduced LTP coding on spatial interest points for Human Action Recognition

Chunkeng Dong
School of Communication & Information Engineering
UESTC
Chengdu, China
dongchunkeng126@126.com

Hao Song
School of Electronic Engineering
UESTC
Chengdu, China
a6802739@163.com

***Abstract*—We present a novel action recognition method which is based on combining detecting 2D informative feature points and coding their local changes in motion directions by Local Trinary Patterns. The resulting method is extremely efficient, and thus is suitable for real-time uses of simultaneous recovery of human action of several lengths and starting points. Tested on some publicity available datasets in the literature known to us, our method can make good performance both on accuracy and efficiency for recognition.**

Keywords- reduced LTP; interest points ;action recognition;

I. INTRODUCTION

Human action recognition from video is an area of immense importance to visual surveillance, video indexing, and several other computer-vision domains. Despite of extensive research, fueled by the ongoing advancements in object recognition, the gap between the current capabilities and the applications' needs remains large.

Subjects under observation can vary in posture, appearance and size. Occlusions and complex backgrounds can impede observation, and variations in the environment, such as in illumination, can further make observations difficult. Moreover, there are variations in the behaviors themselves. Many of the problems described above have counterparts in object recognition. The inspiration for our approach comes from approaches to object recognition that rely on sparsely detected features in a particular arrangement to characterize an object, e.g. [6, 1, 18]. Such approaches tend to be robust to pose, image clutter, occlusion, object variation, and the imprecise nature of the feature detectors. In short they can provide a robust descriptor for objects without relying on too many assumptions.

Beyond recognition accuracy, there are other constraints on the design of action recognition methods. Ideally for several applications, such methods would work efficiently in an online manner, and require simultaneous detection of action at several possible time scales (different action lengths) and for every possible starting point.

LTP[4] method for action recognition is a efficient method. It can capture time-varying "dynamic textures" of motion at every pixel at every frame. But most region of an image does not have motion information. This has directly led two bad results: (a) large amount of operations in no motion regions; (b) informative cues for recognition drowned in the whole coding information.

We proposed a method combining detecting 2D interest points and coding them by a reduced Local Binary Patterns. It only encodes LTP at some small patches centered at interest points. This method highlights the useful motion information in the feature of the whole video. It makes more distinguishing between different samples.

What's more, The LTP code itself has redundant information. So a reduced LTP is proposed in this paper.

II. RELATED WORK

Two popular trends can be identified in the action recognition literature, obtaining top-level performance on existing benchmarks. First, there are contributions which compute representations for nearby frames (either motion centric or to the entire frame) [4, 5]. Such approaches usually rely on optical-flow, on appearance, or on a combination of the two. Second, there are contributions which focus on identifying space-time interest points and on representing those local entities [6, 7].

The Local Binary Patterns (LBP) [8], for example, use short binary strings to encode the micro-texture centered around each pixel. A whole 2D image is represented by the frequencies of these binary strings. In [9] the LBP descriptor was extended to 3D video data and successfully applied to facial expression recognition tasks. These methods evolved from techniques originally designed for recognizing textures in 2D images, by extending them to time-varying "dynamic textures" (e.g., [9]). Another LBP extension to videos, related to our own work here, is the Local Trinary Patterns (LTP) descriptor of [3].

Tracking and behavior recognition are closely related problems, and in fact many traditional approaches to behavior recognition are based on tracking models of varying sophistication, from paradigms that use explicit shape models in either 2D or 3D to those that rely on tracked features; for a broad overview see [9]. The basic idea is that given a tracked feature or object, its time series provides a descriptor that can be used in a general recognition framework.

III. PROPOSED ALGORITHM

We proposed a method combining detecting 2D interest points and coding them by a reduced Local Binary Patterns. In the following sections we describe our algorithm in detail. In Section A we talk about detection of spatial interest points. We describe a reduced LTP in Section B. The effect of coding using our method can be seen in Figure1.

Published by Atlantis Press, Paris, France.

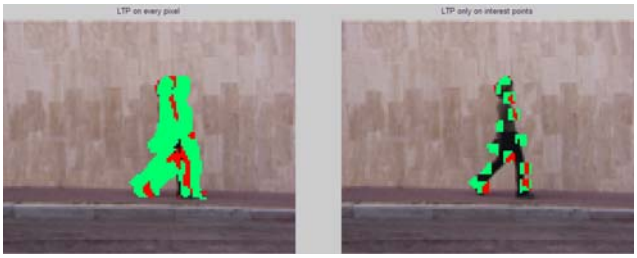

Figure 1. a comparison of traditional LTP coding method (the left part) and LTP coding only on interest points (the right part) . In the figure above, green pixels and red pixels represent the motion information.

A. *Feature Detection*

This step was to extract the interest points from 2D images. There are contributions which focus on Feature detection, such as [10, 12].

We choose Harris corner detector to extract the interest points. The Harris corner detector is a popular interest point detector due to its strong invariance to [11]: rotation, scale, illumination variation and image noise. The Harris corner detector is based on the local auto-correlation function of a signal. The effect of Harris corner detecting is shown in Figure 2.

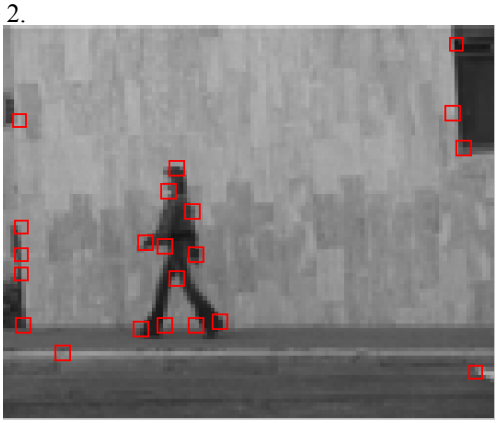

Figure 2. The effect of Harris corner detecting

Harris corner detection process is as follows:

*1).*Calculate the first gradients of image $I(x,y)$ along directions x and y respectively, and then construct a matrix H

$$H=\begin{pmatrix} \hat{I_x^2} & \hat{I_x I_y} \\ \hat{I_x I_y} & \hat{I_y^2} \end{pmatrix} \quad (1)$$

Where $\hat{}$ denotes smooth performed by Gaussian function

$$W=\exp-(x^2+y^2)/2\sigma^2 \quad (2)$$

*2).*Then define the corner response to be

$$R=|H|-k(traceH)^2 \quad (3)$$

B. *A reduced LTP*

The LTP [3] encoding is described in Figure 3. Patches at eight shifted locations at times $t-\Delta t$ and $t+\Delta t$ are compared to a central patch at time t to produce 16 similarities. Consider 3*3 pixel patch, where the location of the center of the patch in the current frame (at time t) is denoted (0; 0). The locations of center pixels of eight 3*3 patches in the previous frame and the next frame are denoted (4; 0), (3; 3), (0; 4), (3; -3), (-4; 0), (-3; -3), (0; -4), and (-3; 3). For each of 8 different locations in the previous frame (at time $t-\Delta t$) and the same locations in the next frame (at time $t+\Delta t$) SSD distances of 3 × 3 patches to a central patch in current frame (at time t) are computed. $SSD1$ and $SSD2$ are computed patch distances at one of the eight locations.

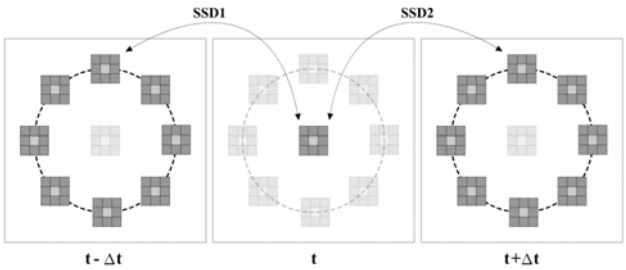

Figure 3. An illustration of the encoding process of LTP.

In the system, due to its computational simplicity, the SSD (sum of square differences) score as the basic distance between the patches. The lower the SSD score, the larger the similarity. For each pixel $p(x,y)$, the code can be decided as follows:

$$value(p)=\begin{cases} 1 & if \quad SSD1-TH>SSD2 \\ 0 & if \quad |SSD1-SSD2|<TH \\ -1 & if \quad SSD2-TH>SSD1 \end{cases} \quad (4)$$

Where TH is a threshold set to 1000. A value of 1 indicates that the former motion is more likely, 1 indicates that the latter is more likely. A value of 0 indicates that both are compatible in approximately the same degree.

In this way, the walk motion is coded in Figure 4. The 8 LTP figure coded on (-3; 3), (0; 4), (3; 3), (-4; 0), (4; 0), (-3; -3), (0; -4), (3; -3) can be divided them into 4 groups： a((-3; 3), (3; -3)）;b（(0; 4), (0; -4)）;c（(3; 3), (-3; -3),）;d（(-4; 0), (4; 0)）.

It is easy to find out that the two LTP figures in each group are complementary. Further to say, the red pixels in one LTP figure roughly correspond to the green pixels in the other LTP figure in each group. In each group, the code of one LTP figure is redundant. Therefore, a reduced LTP coding method is proposed. We just do LTP code on 4 locations of (-3; 3), (0; 4), (3; 3), (-4; 0). The computational complexity is reduced by half when the improved coding mode is adopted. A more intuitive description shows in

figure4. We just do LTP code on location of （(-3; 3), (0; 4), (3; 3), (-4; 0)）shown in Figure 5.

Then one trit is assigned for each of four comparisons that are made between pairs of similarities that share the same spatial shifts. Thus 4 trits are used to represent each pixel in the video. The entire frame is coded by this reduced LTP, and the histograms of the 4-digit trinary strings are measured in each. The string of all zeros indicate that there is no motion and is disregarded (not counted in any bin). The rest of the strings are mapped to bins in an unconventional manner, which reduces the number of possible bins from $3^4-1=80$ to $2\times(2^4)=32$, where each string is counted twice. First the positive part of the string is extracted. In this part every −1 digit is converted to the 0 digit. The positive part is then distributed between all possible binary bins. The same process repeats with the negative part which is accumulated in a separate set of 16 bins.

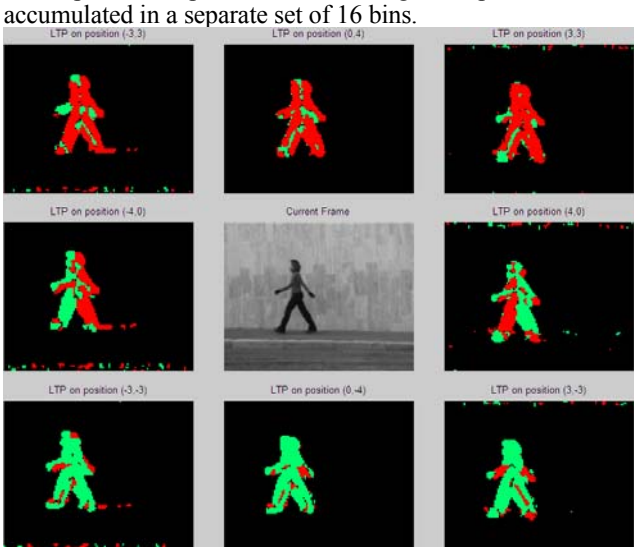

Figure 4. LTP coding of a frame in a walk video. Every pixel in current frame is coded with 8 different patches (center location of (4; 0), (3; 3), (0; 4), (-3; 3), (-4; 0), (-3; -3), (0; -4), (3; -3).) in the previous frame and the next frame respectively to form 8 LTP figures above. Green pixels correspond to LTP code value -1. Red pixels correspond to LTP code value 1. Black pixels correspond to LTP code value 0.

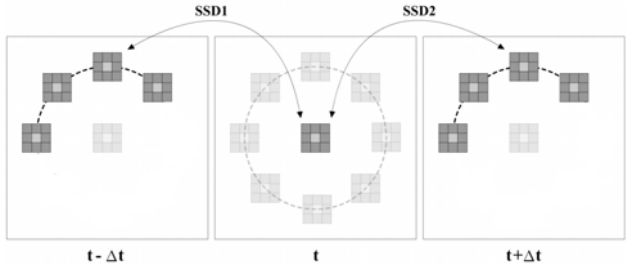

Figure 5. An illustration of the encoding process of reduced LTP.

IV. Recognizing Actions

If the boundaries of the video are given, we divide the video to k equal time slices, and compute the accumulated histograms for each region among the frames of each time-slice. In practice we find it accurate enough to skip frames, and compute the histograms of no more than 10 frames per time slice. The histograms for the k time slices are accumulated to one vector of length $32k$ which is used to represent the entire video. In order to recognize an action, we apply a classifier to those vectors. Specifically we use linear SVM, on the square-root values of the vectors. The square root operation to the values of histogram is meant to approximate the Bhattacharyya coefficients between probability distributions.

A crucial question in motion recognition is the detection of the starting point and length of motion in video. In most existing benchmarks, the part of the video where the motion to be recognized reside is given. This is unrealistic for most applications, and results in an optimistic performance expectation.

We suggest to tackle the two additional unknowns (time shift and scale) by running several detectors in parallel, each observing different starting points and different scales of action length. This system can be built efficiently by reusing previous computations. Since our encoding is very efficient, and since we employ straight-forward linear classification, multiple detections can be achieved at better than real-time rates, our system applied to the recognition of 4 actions at 3 time scales runs at 25 frames-per-second on a modest dual core Intel processor 2.80 GH, 2 GB RAM desktop computer， without using the GPU or other special purpose hardware.

V. Experiments

We ran our system on some existing benchmarks. Our method can make good performance both on accuracy and efficiency for recognition.

A. *Parameters*

There are few parameters for our system. Gauss spatial scale of Harris detector is set to 0.5 when extract the interest points. Centered of the interest points, all patches are chosen to have a size of 3 × 3, and are spread around the central patch at the integer approximations of a circle of radius 4. Also fixed are the value of Δt which is 3, and the threshold on the difference between the two SSD scores, which is set to 1, 000. The number of time slices for each dataset is $2 \leqslant k \leqslant 4$.

B. *Weizmann*

The Weizmann action recognition dataset consists of nine subjects performing nine different actions: bending down, jumping jack, jumping, jumping in place, galloping sideways, running, walking, waving one hand, and waving both hands. The evaluation is done in a leave-one-person-out manner: 8 subjects are used for training, and the remaining one for testing. The experiment is repeated for all 9 persons, and the results are averaged. As Table 1 indicates, our method achieves maximal performance on this dataset.

TABLE I. COMPARISON TO PREVIOUS RESULTS ON THE WEIZMANN DATASET

Name	Percentile	Ref
Our method	100%	
LTP	100%	[4]
Jhuang	98.8%	[13]
Dollar	86.7%	[6]

C. The UCF sports dataset

The UCF sports dataset have collected a large set of action clips from various broadcast sport videos. The actions in this dataset include diving, golf swinging, kicking, lifting, horseback riding, running, skating, swinging a baseball bat, and pole vaulting. The full dataset contains over 200 video sequences. The actions are featured in a wide range of scenes and view points. Testing for this dataset is performed using the leave-one-out framework. The confusion matrix we obtain for this set of experiments is depicted in Table 2. The overall mean accuracy we obtain for this dataset is 82.2%, compared to 79.2% reported in [3].

TABLE II. CONFUSION MATRIX FOR THE UCF SPORTS DATABASE

	Diving	Golf Swing	Ride Horses	Swing	Walk
Diving	**0.82**	0	0.18	0.00	0.00
Golf Swing	0.07	**0.93**	0.00	0.00	0.00
Ride Horses	0.00	0.00	**1.00**	0.00	0.00
Swing	0.00	0.00	0.00	**0.90**	0.10
Walk	0.22	0.00	0.16	0.00	**0.62**

VI. CONCLUSION

We present an effective real-time system for action recognition. The system is based on combining detecting 2D informative feature points and coding their local changes in motion directions by Local Trinary Patterns. LTP is also improved in this paper. In this way, encoding computation is reduced by a certain extent compared with the original LTP. The dimension of the extracted feature vector is greatly reduced. The resulting method is more efficient than traditional LTP. Tested on some publicity available datasets for human action recognition, our method can make good performance both on accuracy and efficiency for recognition.

REFERENCES

[1] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In CVPR, 2003.

[2] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. PAMI, 26(11):1475–1490, Nov 2004.

[3] Yeet, L., Wolf, L.: Local trinary patterns for human action recognition. In: ICCV.2009

[4] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In ICCV, pages 1–8, 2007.

[5] A. Fathi and G. Mori. Action recognition by learning mid-level motion features. CVPR, 2008.

[6] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In VS-PETS, October 2005.

[7] A. Klaeser, M. Marszalek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In BMVC, 2008.

[8] Ojala, T., Pietik ainen, M., Maenp a a, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. TPAMI 24 (2002)

[9] Kellokumpu, V., Zhao, G., Pietik ainen, M.: Human activity recognition using a dynamic texture based method. In: BMVC.

[10] A. Frome, D. Huber, R. Kolluri, T. Bulow, and J. Malik. Recognizing objects in range data using regional point descriptors. In ECCV, 2004.

[11] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. International Journal of Computer Vision, 37(2):151–172, June 2000

[12] C. Harris and M. Stephens. A combined corner and edge detector. In Proc. Alvey Conf., pages 189–192, 1988.

[13] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In ICCV, pages 1–8, 2007.