IEICE TRANS. INF. & SYST., VOL.E100–D, NO.8 AUGUST 2017

LETTER

Trajectory-Set Feature for Action Recognition

Kenji MATSUI[†], *Nonmember*, **Toru TAMAKI**[†a)], *Member*, **Bisser RAYTCHEV**[†], *Nonmember*, *and* **Kazufumi KANEDA**[†], *Member*

SUMMARY We propose a feature for action recognition called Trajectory-Set (TS), on top of the improved Dense Trajectory (iDT). The TS feature encodes only trajectories around densely sampled interest points, without any appearance features. Experimental results on the UCF50 action dataset demonstrates that TS is comparable to state-of-the-arts, and outperforms iDT; the accuracy of 95.0%, compared to 91.7% by iDT.

key words: *action recognition, trajectory, improved Dense Trajectory*

1. Introduction

Action recognition has been well studied in the computer vision literature [1] because it is an important and challenging task. Deep learning approaches have been proposed recently [2]–[4], however still a hand-crafted feature, improved Dense Trajectory (iDT) [5], [6], is comparable in performance. Moreover, top performances of deep learning approaches are obtained by combining the iDT feature [3], [7], [8].

In this paper, we propose a novel hand-crafted feature for action recognition, called Trajectory-Set (TS), that encodes trajectories in a local region of a video. The contribution of this paper is summarized as follows. We propose another hand-crafted feature that can be combined with deep learning approaches. Hand-crafted features are complement to deep learning approaches, however a little effort has been done in this direction after iDT. Second, the proposed TS feature focuses on the better handling of motions in the scene. The iDT feature uses trajectories of densely samples interest points in a simple way, while we explore here the way to extract a rich information from trajectories. The proposed TS feature is complement to appearance information such as HOG and objects in the scene, which can be computed separately and combined afterward in a late fusion fashion.

There are two relate works relevant to our work. One is trajectons [9] that uses a global dictionary of trajectories in a video to cluster representative trajectories as snippets. Our TS feature is computed locally, not globally, inspired by the success of local image descriptors [10]. The other is the two-stream CNN [2] that uses a single frame and a optical flow stack. In their paper stacking trajectories was also reported but did not perform well, probably the sparseness of trajectories does not fit to CNN architectures. In contrast, we take a hand-crafted approach that can be fused later with CNN outputs.

Manuscript received March 2, 2017.
Manuscript revised April 27, 2017.
Manuscript publicized May 10, 2017.

[†]The authors are with Hiroshima University, Higashihiroshima-shi, 739–8527 Japan.
a) E-mail: tamaki@hiroshima-u.ac.jp

DOI: 10.1587/transinf.2017EDL8049

2. Dense Trajectory

Here we briefly summarize the improved dense trajectory (iDT) [6] on which we base for the proposed method. First, the image pyramid for a particular frame at time t in a video is constructed, and interest points are densely sampled at each level of the pyramid. Next, interest points are tracked in the following L frames ($L = 15$ by default). Then, the iDT is computed by using local features such as HOG (Histogram of Oriented Gradient) [10], HOF (Histogram of Optical Flow), and MBH (Motion Boundary Histograms) [11] along the trajectory tube; a stack of patches centered at the trajectory in the frames.

For example, between two points in time t_0 and t_L, a trajectory T_{t_0,t_L} has points $p_{t_0}, p_{t_1}, \ldots, p_{t_L}$ in frames $\{t_0, t_1, \ldots, t_L\}$. In fact, T_{t_0,t_L} is a vector of displacement between frames rather than point coordinates, that is, $T_{t_0,t_L} = (v_0, v_1, \ldots, v_{L-1})$ where $v_i = p_{i-1} - p_i$. Local features such as HOG_{t_i} are computed with a patch centered at p_{t_i} in frame at time t_i.

To improve the performance, the global motion is removed by computing homography, and background trajectories are removed by using a people detector. The Fisher vector encoding [12] is used to compute an iDT feature of a video.

3. Proposed Trajectory-Set Feature

We think that extracted trajectories might have rich information discriminative enough for classifying different actions, even although trajectories have no appearance information. As shown in Fig. 1, different actions are expected to have different trajectories, regardless of appearance, texture, or shape of the video frame contents. However a single trajectory T_{t_0,t_L} may be severely affected by inaccurate tracking results and an irregular motion in the frame.

We instead propose to aggregate nearby trajectories to form a Trajectory-Set (TS) feature. First, a frame is divided into non-overlapping cells of $M \times M$ pixels as shown in

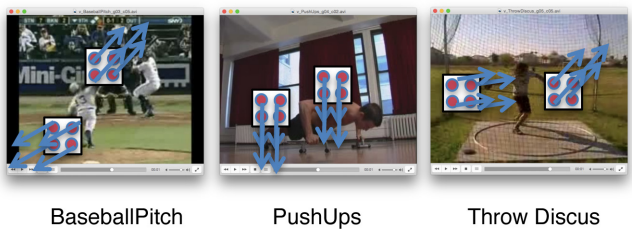

Fig. 1 Different actions in UCF50 [13] have different trajectory information.

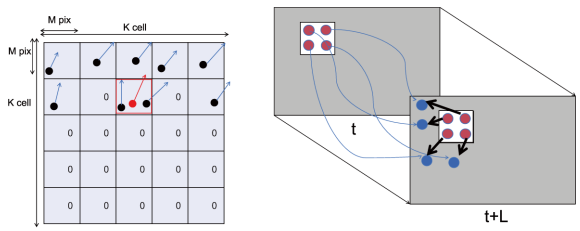

Fig. 2 (a) A block and cells in the starting frame. Starting points of trajectories in each cell are shown in black circles with motion vector arrows. Cells with no starting points are filled with 0. If there are multiple trajectories starting from the same cell, the average trajectory is used for the cell (the averaged starting point is shown in red in this figure). (b) A TrajectorySet feature consists of K^2 trajectories (shown as blue curves) starting from the same block in the starting frame t and wander across the successive L frames. Magenta circles are the starting points of trajectories, and blue circles are corresponding end points. The displacement vectors between starting and end points are shown as black arrows.

Fig. 2 (a). Next, $K \times K$ cells form a block*. This results in overlapping blocks of $MK \times MK$ pixels with spacing of M pixels.

The key concept of the TS feature is to collect trajectories that start in a local region (or block) in the starting frame (see Fig. 2 (a)). In each cell of a block in the starting frame, we find a trajectory starting from the cell. (If there are multiple trajectories starting from the cell, the average trajectory is used. If no trajectory starts from the cell, we use a zero vector as the trajectory of the cell.) By repeating this procedure for all $K \times K$ cells in the block, we have a set of trajectories starting from the block. We concatenate the trajectories to form a TS feature of dimension $2LK^2$ for the block. As shown in Fig. 2 (b), the TS feature consists of trajectories that start in the same block in the starting frame and wander across frames. Note that the end points of the trajectories are not necessary close to each other. This implies that we enforce the locality of trajectories only in the starting frame.

In our default setting, $L = 15$, $M = 10$, and $K = 5$, then the TS feature is a 750 dimensional vector. Figure 3 shows examples of TS features for different categories. We can see different motion patterns appear in each of TS features.

Here we can propose some variations. Instead of using a trajectory as a series of displacements $T_{t_0,t_L} = (v_0, v_1, \ldots, v_{L-1})$, we can simply a series of coordinates like as $T_{t_0,t_L} = (p_0, p_1, \ldots, p_L)$, but in local coordinate systems

*Note that we borrow the terms from HOG [10].

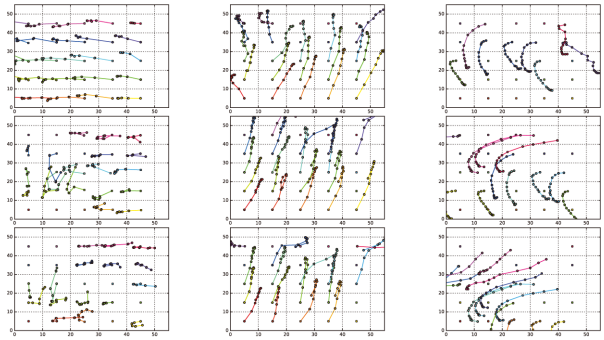

Fig. 3 Examples of TS features of (a) BaseballPich, (b) PushUps, and (c) ThrowDiscus in the UCF50. Each row shows different TS features obtained from different blocks and different sets of 15 frames. Each plot shows 25 trajectories (in different colors) starting from each of cells in a block. Trajectories are shown with 16 points (some points are overlapped) connected with lines. The block and cell sizes are 50×50 and 10×10 pixels, respectively.

instead of the global coordinate system. For further reducing computation cost, we can skip every two frames by summing successive two displacement vectors (that is, by skipping one frame in $(v_0, v_1, \ldots, v_{L-1})$ to generate $(v_0 + v_1, v_2 + v_3, \ldots)$), resulting in feature vectors of dimension 400. We call these processes "skip2" in the results.

4. Experimental Results and Discussion

Here we describe experimental results of the proposed method. We used UCF50 [13]. It has 50 action categories. Videos in each category are divided into 25 groups, and we evaluate the accuracy with the leave-one-group-out cross validation. The resolution of videos are 320×240 @ 30fps, and the durations are between 1 and 6 seconds. For TS feature construction, we use $M = 10$ pixels, $K = 5$, and $L = 15$, and randomly sample 1% of TS features for encoding with the Fisher vector with 64 Gaussians. A multi-layer perceptron (MLP) of three layers, with a middle hidden layer of 100 nodes, was are trained.

Results are shown in Table 1. We compare the proposed TS feature with the original iDT feature and other recent methods. Skip 2 version of TS feature doesn't perform well, showing that we need to take care about parameter tuning for a better performance. Exploring the effects of parameters (skipping, M, K, and L) is an important part of our future work.

By comparing with other recent methods, our TS feature outperforms the original iDT, and is better than most of other methods, even without any appearance information of the scene. We are now planning to validate how the proposed TS feature can be combined with other methods, including deep learning approaches, for improving the performance.

Recent work of action recognition uses more larger datasets, such as UCF101 [24] and HMDB51 [25]. Our preliminary results on UCF101 with the proposed TS feature is about 30%, which is awful compared to the current baseline (87% [2]) and the state-of-the-arts (94% [3], [4]). The

Table 1 Comparison of results on UFC50.

	accuracy
Wang+2013 (DT) [14]	83.6
Kataoka+2015 [15]	84.5
Beaudry+2016 [16]	88.3
TS skip2 (ours)	**89.4**
Li+2016 [17]	90.3
Wang&Schmid 2013 (iDT) [6]	91.7
Peng+2016 [18]	92.3
Yang+2017 [19]	92.4
Lan+2015 [20]	93.8
Lan+2015 [21]	94.4
Xu+2017 [22]	94.8
TS (ours)	**95.0**
Duta+2016 [23]	97.8

reason is that many categories in UCF101 have videos of almost static scenes. For example, in "playing piano" category, a person plays a piano indoor and only small portion of the hands move very small amount. This is the limitation of approaches focusing on motion only, therefore motion and appearance cues are used to help each other [26], [27]. This is the direction we will explore in the near future.

Acknowledgments

This work was supported in part by JSPS KAKENHI grant number JP16H06540.

References

[1] S. Herath, M. Harandi, and F. Porikli, "Going Deeper into Action Recognition: A Survey," Image and Vision Computing, vol.60, pp.4–21, 2017.

[2] K. Simonyan and A. Zisserman, "Two-Stream Convolutional Networks for Action Recognition in Videos," in Advances in Neural Information Processing Systems 27, ed. Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger, pp.568–576, Curran Associates, Inc., 2014.

[3] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional Two-Stream Network Fusion for Video Action Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.1933–1941, IEEE, June 2016.

[4] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. van Gool, "Temporal segment networks: Towards good practices for deep action recognition," European Conference on Computer Vision ECCV 2016, vol.9912, pp.20–36, Springer, Cham, 2016.

[5] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '11, Washington, DC, USA, pp.3169–3176, IEEE Computer Society, 2011.

[6] H. Wang and C. Schmid, "Action recognition with improved trajectories," Proceedings of the 2013 IEEE International Conference on Computer Vision, ICCV '13, Washington, DC, USA, pp.3551–3558, IEEE Computer Society, 2013.

[7] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning Spatiotemporal Features with 3D Convolutional Networks," 2015 IEEE International Conference on Computer Vision (ICCV), pp.4489–4497, IEEE, Dec. 2015.

[8] L. Wang, Y. Qiao, and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.4305–4314, IEEE, June 2015.

[9] P. Matikainen, M. Hebert, and R. Sukthankar, "Trajectons: Action recognition through the motion analysis of tracked features," 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops 2009, pp.514–521, IEEE, Sept. 2009.

[10] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01, CVPR '05, Washington, DC, USA, pp.886–893, IEEE Computer Society, 2005.

[11] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," Proceedings of the 9th European Conference on Computer Vision - Volume Part II, ECCV'06, Berlin, Heidelberg, vol.3952, pp.428–441, Springer-Verlag, 2006.

[12] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the fisher vector: Theory and practice," Int. J. Comput. Vision, vol.105, no.3, pp.222–245, Dec. 2013.

[13] K.K. Reddy and M. Shah, "Recognizing 50 human action categories of web videos," Mach. Vision Appl., vol.24, no.5, pp.971–981, July 2013.

[14] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," International Journal of Computer Vision, vol.103, no.1, pp.60–79, 2013.

[15] H. Kataoka, Y. Aoki, K. Iwata, and Y. Satoh, "Evaluation of Vision-Based Human Activity Recognition in Dense Trajectory Framework," International Symposium on Visual Computing, Advances in Visual Computing, vol.9474, pp.634–646, Springer, Cham, 2015.

[16] C. Beaudry, R. Péteri, and L. Mascarilla, "An efficient and sparse approach for large scale human action recognition in videos," Machine Vision and Applications, vol.27, no.4, pp.529–543, May 2016.

[17] Q. Li, H. Cheng, Y. Zhou, and G. Huo, "Human Action Recognition Using Improved Salient Dense Trajectories," Computational Intelligence and Neuroscience, vol.2016, pp.1–11, 2016.

[18] X. Peng, L. Wang, X. Wang, and Y. Qiao, "Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice," Computer Vision and Image Understanding, vol.150, pp.109–125, 2016.

[19] Y. Yang, D.C. Zhan, Y. Fan, Y. Jiang, and Z.H. Zhou, "Deep Learning for Fixed Model Reuse," Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI17), San Francisco, pp.2831–2837, 2017.

[20] Z. Lan, X. Li, M. Lin, and A.G. Hauptmann, "Long-short Term Motion Feature for Action Classification and Retrieval," tech. rep., CoRR abs/1502.04132, 2015.

[21] Z. Lan, M. Lin, X. Li, A.G. Hauptmann, and B. Raj, "Beyond Gaussian Pyramid: Multi-skip Feature Stacking for action recognition," Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol.07-12-June-2015, pp.204–212, 2015.

[22] Z. Xu, R. Hu, J. Chen, C. Chen, H. Chen, H. Li, and Q. Sun, "Action recognition by saliency-based dense sampling," Neurocomputing, vol.236, pp.82–92, 2017.

[23] I.C. Duta, B. Ionescu, K. Aizawa, and N. Sebe, "Spatio-Temporal VLAD Encoding for Human Action Recognition in Videos," International Conference on Multimedia Modeling MMM 2017, vol.10132, pp.365–378, Springer, Cham, 2017.

[24] K. Soomro, A.R. Zamir, and M. Shah, "UCF101: A Dataset of 101 human actions classes from videos in the wild," Tech. Rep. November, CRCV-TR-12-01, 2012.

[25] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," Proceedings of the IEEE International Conference on Computer Vision, pp.2556–2563, IEEE, Nov. 2011.

[26] Y. Wang, J. Song, L. Wang, L. Van Gool, and O. Hilliges, "Two-Stream SR-CNNs for Action Recognition in Videos," BMVC, pp.1–12, 2016.

[27] M. Jain, J.C. Van Gemert, and C.G.M. Snoek, "What do 15,000 object categories tell us about classifying and localizing actions?," Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp.46–55, IEEE, June 2015.