IEICE TRANS. INF. & SYST., VOL.E96–D, NO.5 MAY 2013

LETTER

Self-Similarities in Difference Images: A New Cue for Single-Person Oriented Action Recognition

Guoliang LU[†a)], *Nonmember and* **Mineichi KUDO**[†], *Member*

SUMMARY Temporal Self-Similarity Matrix (SSM) based action recognition is one of the important approaches of single-person oriented action analysis in computer vision. In this study, we propose a new kind of SSM and a fast computation method. The computation method does not require time-consuming pre-processing to find bounding boxes of the human body, instead it processes difference images to obtain action patterns which can be done very quickly. The proposed SSM is experimentally confirmed to have high power/capacity to achieve a better classification performance than four typical kinds of SSMs.

key words: *action recognition, self-similarity matrix, difference images*

1. Introduction

In this study, we concentrate on recognizing actions performed by single users driven by natural demands for it in single-person oriented applications, e.g., single-person computer interaction, action monitoring of people living alone for home-aid assistance.

In this field, significant research efforts have been made to extract effective action patterns for recognition. Among them, temporal self-similarity matrix (SSM) based approach is an important branch [1]–[7]. The training phase is as follows. First, in every frame of a given video sequence, bounding boxes of human body are detected/located by background subtraction or by a human detection/tracking algorithm, and then they are normalized to be the same size and the same orientation. Second, frame features are computed from such normalized boxes by using original color description [1]–[3], [6], human silhouettes [2], [3], histogram of gradient (HoG) [4], [5], optical flow [4], [5], [7], or body-trajectory [4], [5]. Last, the temporal SSM is obtained by the distances computed between all video frame pairs. In the recognition phase, the SSM of a newly observed sequence is firstly generated in the same way, and then it is compared with the learned template SSMs of typical actions to assign the most probable action to the given sequence by using sequences alignment, the *bag-of-words* framework and so on. Such conventional SSM computation methods were demonstrated to be effective on action recognition in some databases, e.g., Weizmann, IXMAS databases. On the other hand, it is clear that the recognition performance strongly depends on the quality of bounding boxes and finding such boxes also costs extra time.

In this study, we propose to learn a new kind of SSM directly from difference images of a given video sequence. The procedure is outlined as follows. First, difference images are generated by subtracting every frame from its preceding frame in the video. Next, two histograms are obtained by projecting the difference values onto X- and Y-axis, and then an SSM is calculated on the basis of these two histograms. Our contributions are described as follows:

- The proposed SSM does not rely on bounding boxes that are usually detected in pre-processing, thus, it is not affected much by the detection error of bounding boxes. Indeed, this is vertified in the experimental results with a better performance than the other four typical kinds of SSMs in the same classification procedure.
- The proposed SSM can be calculated very fast because it only needs computations of frame difference and two projections. It does not require a high-cost process of finding bounding boxes. Therefore it is more suitable for real-time applications compared with conventional SSM based action recognition.

This manuscript is structured as follows. The proposed method of SSM is given in Sect. 2. Section 3 describes the employed classification technique. In Sect. 4, experimental results are presented and discussed, followed by conclusions in Sect. 5.

2. Learning SSM from Difference Images

We impose the following assumptions: (1) the sampling rate (25 fps in experiments) is sufficiently high for generating difference images, (2) the camera-view is stationary or could be estimated, and (3) a video sequence includes a single action only, that is, a segmentation has been made already.

2.1 Difference Image

A difference image expresses the change of the current frame relative to its pre-recorded frame, and has been used for action detection [8], [9]. Usually it does not require background subtraction nor object detection/tracking and thus it is obtained very fast. The gray-level difference image $\Delta I(x, y, t)$ between two concecutive frames $I(t)$ and $I(t-1)$ is computed as:

$$\Delta I(x, y, t) = |I(x, y, t) - I(x, y, t-1)|, \quad (1)$$

Manuscript received September 7, 2012.
Manuscript revised December 19, 2012.

[†]The authors are with Laboratory of Pattern Recognition and Machine Learning, Graduate School of Information Science and Technology, Hokkaido University, Sapporo-shi, 060–0814 Japan.
a) E-mail: luguoliang@main.ist.hokudai.ac.jp

DOI: 10.1587/transinf.E96.D.1238

where $I(x, y, t) \in [0, 255]$ is the pixel value at (x, y) of tth frame of a given frame sequence.

As a pre-processing, we remove small differences by a threshold value λ: $\Delta I(x, y, t) = 0$ if $\Delta I(x, y, t) < \lambda$, where λ is set to 5 in the experiments. In addition, we remove isolated spatial noise prior to (1) by applying a smoothing filter: $I(t) \leftarrow I(t) * G$, where G is a filter (a *mean* filter of 3×3 pixels, in the experiments) and '$*$' is the convolution operation.

2.2 Frame Representation by X- and Y-Axis Projections and the Spatio-Temporal Volumes

For one difference image $\Delta I(x, y, t)$ of size $n \times m$, we project the difference pixel values onto X- and Y-axis, respectively, for image representation (Fig. 1). Then, normalize these histograms to the maximum value of one in H_t^X and H_t^Y, respectively, where H_t^X is the histogram of X-axis projection in the tth frame and H_t^Y is that of Y-axis.

Next, we combine the X- and Y-axis histograms, i.e. H_t^X and H_t^Y, into the spatio-temporal volumes $V^X = [H_1^X, H_2^X, \ldots]$ and $V^Y = [H_1^Y, H_2^Y, \ldots]$, respectively. One notes that these volumes have a high potential to distinguish performed actions, e.g., as seen in Fig. 2, we can see in the V^X and V^Y the periodic movement of action **jack** and in the V^X the forward movement of **walk**.

2.3 Computation of Self-Similarity Matrix (SSM)

The differences among actions seen in V^X and V^Y are more clearly and appropriately summarized in their temporal self-similarity matrices (SSMs). Let Z denotes X or Y. We suppose that a frame sequence F of $T + 1$ frames has been described by the spatio-temporal volume $V^Z = [H_1^Z, H_2^Z, \ldots, H_T^Z]$ (H_1^Z corresponds to the 2nd frame since it is computed from a difference image). We measure the anti-similarity (distance) between every ith and jth frames to construct the self-similarity matrix as:

$$D^Z = [d_{i,j}^Z] = \begin{pmatrix} d_{1,1}^Z & \cdots & d_{1,T}^Z \\ \vdots & \ddots & \vdots \\ d_{T,1}^Z & \cdots & d_{T,T}^Z \end{pmatrix}, \; d_{i,j}^Z = \|H_i^Z - H_j^Z\| \tag{2}$$

where $\|\cdot\|$ is the Euclidean distance between two histograms of ith and jth frames. Obviously, D^Z is symmetrical and its diagonal elements are always zeros. The differences among actions are visually clear as different patterns in their SSMs. For example, in Fig. 2, repeated patterns in D^X and D^Y show the periodical nature of the action of **jack**.

2.4 Video Representation

The patterns/structures in SSMs for comparing human actions could be captured by one of local descriptors [4], [5], time-frequency analysis [1]–[3], and *frame-to-frame convolution* [7]. The local descriptor is employed here due to its largest applicability. The procedure of obtaining the local descriptors $p(D)$ of an SSM, D, is explained in Fig. 3 with some illustrative images. We make ready C local descriptor sequences $\mathcal{P}^k = \{p_1^k, p_2^k, \ldots, p_C^k\}$ for every action k, where the lengths $|p_c^k|$ can differ. It is noted that we have two local descriptor sequences to represent a video, say p^X and p^Y, from D^X and D^Y, respectively.

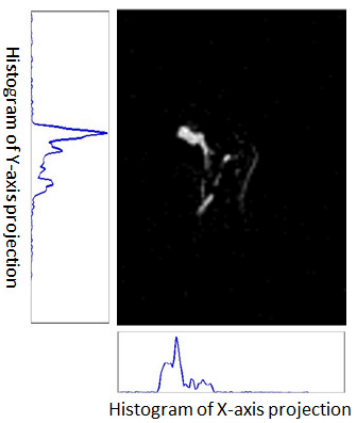

Fig. 1 A difference image and its X- and Y-axis projection histograms.

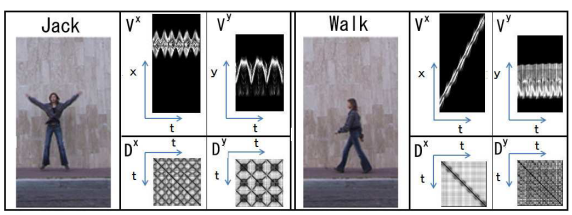

Fig. 2 The spatio-temporal volumes V^X and V^Y and their SSMs D^X and D^Y in two actions of **jack** and **walk**.

3. Sequences Alignment Based Action Recognition

For a newly observed sequence F, its SSM $D(F)$ and its local descriptor sequence p are calculated. Then, to recognize the performed action in F, we employ *sequences alignment*

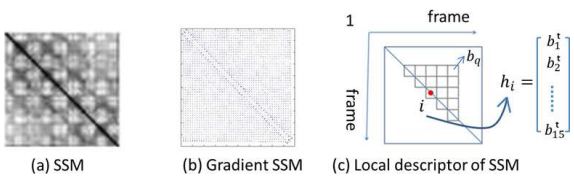

Fig. 3 Given a SSM (a), the gradient (b) is first computed. Then, an upper-right triangle mask consisting of 15 blocks (c) (each of which was set to be a square [2] of 3×3 frames) is located at each diagonal element i. Within every block $q \in \{1, 2, \ldots, 15\}$, we compute 8-bin histogram b_q of gradient directions [4], [5]. The local descriptor h_i of the diagonal element i is obtained by concatenating all 15 b's as a vector of length $8 \times 15 = 120$D: $h_i = [b_1^t, \ldots, b_{15}^t]^t$, where t denotes transpose. We thus have a vector set of length T as $p(D) = [h_1, \ldots, h_T]$ for one SSM D with T diagonal elements.

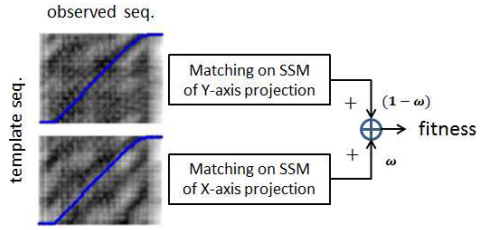

Fig. 4 The minimum cost path is searched by dynamic programming; Then, fitness is learned as a weighted scheme on X- and Y-axis projections.

that assigns F to the action of sequence with the maximum matching fitness as follows[†].

3.1 Dynamic Sequence Matching

Let us assume $p = [h_1, h_2, \ldots, h_T]$ to be the local descriptors of the sequence F. Dynamic programming is used to find the optimal alignment between p and p_c^k. With a cost function $dist(h, h')$, typically the Euclidean distance, between two local descriptors h and h', we find the smallest cost path σ^* connecting $p = \{h_1, h_2, \ldots, h_T\}$ and $p' = \{h'_1, h'_2, \ldots, h'_{T'}\}$ as

$$\sigma^*(p, p') = \arg \min_{\sigma=(\sigma_1,\sigma_2)} \sum_{l=1}^{|\sigma|} dist(h_{\sigma_1(l)}, h'_{\sigma_2(l)}), \quad (3)$$

where $\sigma_1(l) \in \{h_{l-1}, h_l, h_{l+1}\}$ and $\sigma_2(l) \in \{h'_{l-1}, h'_l, h'_{l+1}\}$ for $l \in \{2, 3, \ldots, |\sigma|-1\}$ with the terminal conditions $\sigma_1(1) = h_1$, $\sigma_2(1) = h'_1$, $\sigma_1(|\sigma|) = h_T$ and $\sigma_2(|\sigma|) = h'_{T'}$. Using (3), we measure the matching score of p to p_c^k, and then that to action k by $s(p, k) = \min_c \sigma^*(p, p_c^k)$.

3.2 Action Assignment

Since actions are represented separately in X- and Y-axis projections, there are two matching scores, $s^X(p, k)$ and $s^Y(p, k)$. Taking into account the difference of relative importance of X- and Y-axis projections, as shown in Fig. 4, we define the fitness of p to action k by

$$fitness(p, k) = -(\omega \cdot s^X(p, k) + (1 - \omega) \cdot s^Y(p, k)), \quad (4)$$

with a weight $\omega \in [0, 1]$, where ω is estimated using cross validation over all training sequences[††].

Finally, the newly observed sequence F with local descriptors p is assigned to k^*, that is, $F \rightarrow k^*$, where $k^* = \arg\max_k(fitness(p, k))$.

4. Experiment

The experiments were made on Weizmann database, where human bounding boxes are given. We regarded two actions of **wave1** and **wave2** as one action **wave** as made in [4], [5], since these two actions are generally confused to be each other with flow-based descriptions [12].

In the following two subsessions, we first compared the performance in recognition rate[†††] of proposed computation method with those of other conventional methods in the framework described in Sects. 2 and 3. Then, we compared the performance with *state-of-the-art* works in another framework. In both cases, the recognition rate was estimated by leave-one-out cross-validation.

4.1 Comparison with Conventional SSMs

We compared the proposed SSM computation method with four popular competitors with: (1) original color description, (2) human silhouettes, (3) histogram of gradient (HoG) and (4) optical flow. We excluded the method with body-trajectory, because it is complicated to label/track the human body parts, e.g., head and arm, in a video, and no general database has provided such trajectory information. The same procedures were applied in common to the obtained SSM computed with five compared methods in the later processes of extraction of the local descriptor and of recognition by using dynamic sequences matching.

4.1.1 Implementation

First, we normalized each of given bounding boxes to be a size 90×60 pixels; Second, SSMs were computed with the compared methods as belows.

SSM computed with original color description or human silhouettes: SSM is generated by simple *sum of square difference (SSD)* [6] between two bounding boxes expressed by the original gray-level values or by binary silhouette associated with the human body. In order to account for tracking errors, each bounding box B of size $L \times L$ is described by block-based representation as follows. First, B is divided into non-overlapping blocks b's of size $\ell \times \ell$. The *mean* value $v(b)$ in every block b is then computed as $v(b) = \sum_{(x,y)\in b} I(x, y)/(\ell \times \ell)$. Finally, we combine $\lceil L/\ell \rceil \times \lceil L/\ell \rceil$ $v(b)$'s into a single vector $v = [v(1), v(2), \ldots]^t$. The size of b was set to 3×3 pixels in the experiment.

SSM computed with HoG: SSM is generated with the Euclidean distance between histograms of oriented gradient (HoG) of two bounding boxes [4], [5]. This descriptor characterizes the local shape of human body by capturing the gradient structure. In our implementation, we used a 3×3 spatial bin and nine gradient orientations for calculating HoG followed by [10]. The feature dimension is 81.

SSM computed with optical flow: SSM is computed by optical flow. The optical flow is calculated using the Lucas and Kanade algorithm [13] on every bounding box

[†]It is noted that we used *sequences alignment* as a typical classification technique to evaluate the potential of the proposed SSM, although many alternatives exist.

[††]The value of ω was estimated as 0.6 using leave-one-out cross-validation on the Weizmann database (ω was investigated from 0.1 to 1 at step of 0.1 in this estimation). That means the X-axis projection is more informative than Y-axis projection. Indeed, there is a larger variety in V^X than that in V^Y as seen in Fig. 2. We then simply set $\omega = 0.6$ in the experiment.

[†††]The recognition rate is defined as a ratio of the number of correctly recognized sequences over the total sequences number.

with three consecutive frames. As same as [4], [5], three different SSMs are computed based on X-direction (denoted by of_x in Table 1), Y-direction (of_y) and both (of_xy). Every bounding box B is described by block-based representation for absorbing tracking errors as described above, although *median* is used instead of *mean* [11].

4.1.2 Performance in Action Recognition

Figure 5 shows the confusion matrix of action recognition by the proposed SSM. It is observed that among 9 action classes, actions of **bend** and **run** are perfectly correctly recognized. It implies that the SSMs computed from these two actions are very discriminating to the others. It is also observed that the recognition rate of **jump** is very low, 22.2%. This is probably because the style of this action strongly depends on the performers, as reported in [4], [5].

From Table 1, we see that the SSM computed with the proposed method outperformed the four SSMs computed with compared methods in recognition rate. That implies that SSMs of the competetors tend to be affected much by the quality of pre-detected human bounding boxes. Indeed the extraction quality are sometimes not satisfactory (see Fig. 6). In this sense, our SSM derived directly from difference images are more robust than those.

4.2 Comparison with *State-of-the-Art* [4], [5]

The recognition rate of 77.8% has been obtained in the above subsession, it is much lower than 94.6% in [4], [5]. However, it should be noted that our rate was achived in a straightforward framework that is different from the framework in [4], [5] where SVM classifier was used with *bag-of-words (BoW)* features. For fair comparison, in this subsession, we used the local descriptors extracted from the SSMs of all training video sequences to generate *codebook* as well and recognized actions with the linear-SVM using BoW as the same as [4], [5].

Figure 7 shows the recognition rates by the proposed SSM using BoW features at size of codebook from 60 to 120 (with an interval of 5). The best performance of 95.6% was obtained at size 105 of codebook. It is better than 94.6% in [4], [5], although their result was obtained with multiple SSMs, i.e., *original, of_x, of_y* and *HoG* were combined together. This reveals again that our proposed SSM has a better power/capacity of absorbing detection error (as previously illustrated in Fig. 6). In Fig. 8, we also show the confusion matrix of action recognition corresponding to the best recognition rate of 95.6%. It is seen that the recognition rates of almost all actions have been improved compared with Fig. 5. Especially, the recognition rate of **jump** is increased to 89% which is far better than 77.8% reported in [4], [5].

4.3 Discussion

Since the goal of this study is to propose a more informative and more efficient construction method of SSM, we measured the computation complexities (*space* complexity, O_{space}, and *time* computation, O_{time}) in three phases. The proposed SSM computation method and four conventional ones are compared in Table 2. For the compared methods, detection of bounding boxes is conformed by a typical processing, i.e., the bounding box is firstly detected by background subtraction at the first frame and then followed by tracking frame by frame, e.g. using mean-shift [14]. Since the proposed computation method does not need background modeling, and thus can bypass a time-consuming process of detection of bounding boxes.

As an unavoidable nature of flow descriptor, the proposed SSM also has difficulty in discriminating some actions with *similar flow*, e.g., **wave1** and **wave2** in the Weizmann database, even if they look different visually. To cope with this problem, combining other structure descriptors could be effective.

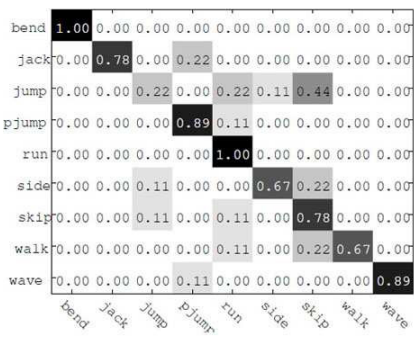

Fig. 5 Confusion matrix of action recognition by the proposed method.

Table 1 Performances on recognition rate (%) by compared SSMs.

SSM	**Ours**	original	silhouette	HoG	of_x	of_y	of_xy
Rec.	**77.8**	48.9	61.1	46.7	54.4	57.8	68.9

Fig. 6 Some examples of human bounding boxes with incorrect/lost pixels, provided in Weizmann database.

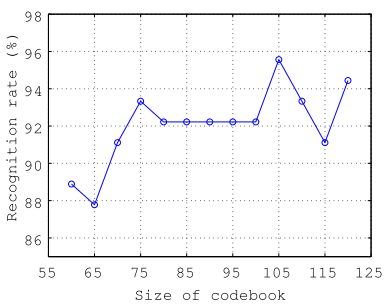

Fig. 7 Performance in recognition rate by the proposed SSM with a linear-SVM using BoW features.

Table 2 Complexities of the proposed computation method and conventional ones. Here, p is the number of pixels in an image and p' is the number of pixels in a detected bounding box; N is the frame number of a newly observed video; N' is the frame number of training sequence(s) for background modeling.

Method	Background Modeling	Detection of bounding boxes	Frame Representation
Original	For typical pixel-based modeling: $O_{space}(p), O_{time}(N')$	Background subtraction: $O_{space}(p)$; + Subject tracking: $O_{space}(p'), O_{time}(N)$.	-
Silhouette			-
HoG			$O_{space}(p'), O_{time}(N)$
Optical flow [15]			$O_{space}(p'^2), O_{time}(N)$
Ours	-	-	$O_{space}(p), O_{time}(N)$

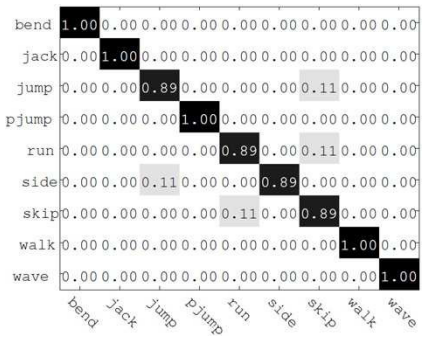

Fig. 8 Confusion matrix of action recognition corresponding to the best recognition rate shown in Fig. 7.

In addition, we have assumed a single performer in this study, the proposed method, however, can be utilized even for the scenes where multiple performers are doing actions simultaneously. We can (1) adopt a sliding-window approach over the video, by which we first assume only one performer in each searching 3D subvolume and then compute the SSM by the proposed method, and finally the action in this subvolume is recognized; (2) reserve only the target performer by excluding other unintereseted performers in the observing scene, by which the action of this performer is recognized as stated already. For this case, the target performer has to be located prior to action recognition, but we only require an approximate position of this performer so as to exclude other performer(s), which can be obtained by a naive detection algorithm. This relaxes the requirement of detection algorithm.

5. Conclusion

In this study, we have proposed a new kind of SSM and its fast computation method for action recognition; it seeks action patterns directly from difference images in a video. Experiments on the Weizmann database showed that the proposed SSM outperformed other four popular competitors in recognition rate. Although it was hard to compare their real computation costs in real-life database, due mainly to implementation difficulties, the SSM can be calculated very fast since it has bypassed a high-cost process of finding bounding boxes, as shown in Table 2. In the future, we will (1) combine other structure descriptor [16] to further improve the recognition performance of proposed SSM; (2) apply the proposed method into some practices, e.g., action monitoring of people living alone for home assistance.

References

[1] R. Cutler and L.S. Davis, "Robust real-time periodic motion detection, analysis and applications," IEEE Trans. Pattern Anal. Mach. Intell., vol.22, no.8, pp.781–796, 2000.
[2] C. BenAbdelkader, R. Cutler, and L.S. Davis, "Gait recognition using image self-similarity," EURASP J. Appl. Signal Process., vol.4, pp.572–585, 2004.
[3] C. BenAbdelkader, R. Cutler, and L.S. Davis, "Motion-based recognition of people in eigengait space," Proc. Int'l Conf. Automatic Face and Gesture Recognition, pp.267–274, 2002.
[4] I.N. Junejo, E. Dexter, I. Laptev, and P. Pérez, "View-independent action recognition from temporal self-similarities," IEEE Trans. Pattern Anal. Mach. Intell., vol.33, no.1, pp.172–185, 2011.
[5] I.N. Junejo, E. Dexter, I. Laptev, and P. Pérez, "Cross-view action recognition from temporal self-similarities," Proc. ECCV, vol.2, pp.293–306, 2008.
[6] E. Shechtman and M. Irani, "Matching local self-similarities across images and videos," Proc. CVPR, pp.1–8, 2007.
[7] A.A. Efros, A.C. Berg, G. Mori, and J. Malik, "Recognizing action at a distance," Proc. ICCV, pp.726–733, 2003.
[8] M.B. Holte, T.B. Moeslund, and P. Fihl, "View invariant gesture recognition using the CSEM SwissRanger SR-2 camera," International Journal of Intelligent Systems Technologies and Applications, vol.5, no.3, pp.295–303, 2008.
[9] M. Yang, F. Lv, W. Xu, K. Yu, and Y. Gong, "Human action detection by boosting efficient motion features," Proc. ICCV Workshops, pp.522–529, 2009.
[10] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," Proc. CVPR, vol.1, pp.886–893, 2005.
[11] A. Fathi and G. Mori, "Action recognition by learning mid-level motion features," Proc. CVPR, pp.1–8, 2008.
[12] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," Proc. Int'l Conf. on Multimedia, pp.357–360, 2007.
[13] B.D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," Proc. Int'l Conf. Artificial Intelligence, pp.674–679, 1981.
[14] P. Meer, "Kernel-based object tracking," IEEE Trans. Pattern Anal. Mach. Intell., vol.25, no.5, pp.564–577, 2003.
[15] A. Bab-Hadiashar and D. Suter, "Robust optic flow computation," Int. J. Comput. Vis., vol.29, no.1, pp.59–77, 1998.
[16] P. Natarajan and R. Nevatia, "View and scale invariant action recognition using multiview shape-flow models," Proc. CVPR, pp.1–8, 2008.