Movement Pattern Histogram for Action Recognition and Retrieval

Arridhana Ciptadi[1], Matthew S. Goodwin[2] and James M. Rehg[1]
[arridhana,rehg]@gatech.edu, m.goodwin@neu.edu

[1]College of Computing, Georgia Institute of Technology
[2]Department of Health Sciences, Northeastern University

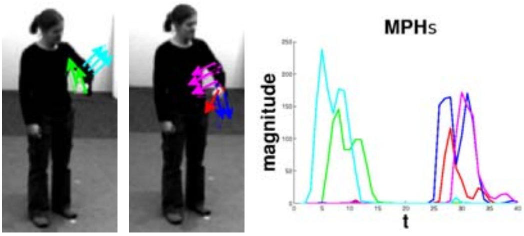
Abstract. We present a novel action representation based on encoding the global temporal movement of an action. We represent an action as a set of movement pattern histograms that encode the global temporal dynamics of an action. Our key observation is that temporal dynamics of an action are robust to variations in appearance and viewpoint changes, making it useful for action recognition and retrieval. We pose the problem of computing similarity between action representations as a maximum matching problem in a bipartite graph. We demonstrate the effectiveness of our method for cross-view action recognition on the IXMAS dataset. We also show how our representation complements existing bag-of-features representations on the UCF50 dataset. Finally we show the power of our representation for action retrieval on a new real-world dataset containing repetitive motor movements emitted by children with autism in an unconstrained classroom setting.

1 Introduction

The recognition and retrieval of actions in videos is challenging due to the need to handle many sources of variations: viewpoint, size and appearance of actors, scene lighting and video quality, etc. In this paper we introduce a novel action representation based on motion dynamics that is robust to such variations.

Currently, state-of-the-art performance in action classification is achieved by extracting dense local features (HOG, HOF, MBH) and grouping them in a bag-of-features (BOF) framework [26]. The basic BOF representation ignores information about the spatial and temporal arrangement of the local features by pooling them over the entire video volume. More recently, it has been shown that considering the spatial and temporal arrangements (dynamics) of an action (eg. extracting separate BOF model for each subvolume of a video [14,26] or modelling the spatio-temporal arrangements of the interest points [29]) adds more discriminative power to the representation.

Our approach is based on the observation that the dynamics of an action provide a powerful cue for discrimination. In Johansson's moving light display experiment, it was shown that humans perceive actions by abstracting a coherent structure from the spatio-temporal pattern of local movements [9]. While humans respond to both spatial and temporal information, the spatial configuration of movements that comprise an action is strongly affected by changes in viewpoint. This suggests that representing the temporal structure of an action could be valuable for reducing the effect of viewpoint. Motivated by this observation, we define human actions as a composition of temporal patterns of movements.



(a) Frame 10 (b) Frame 31 (c) MPHs

Fig. 1. Movement Pattern Histogram for $checkwatch$ action. (a)-(b): Arrows indicate optical flow direction and are color coded according to the flow words (flows are subsampled for presentation). (c): MPH set for $checkwatch$. (Best viewed in color)

Our key hypothesis is that the temporal dynamics of an action are similar across views. For example, the timing pattern of acceleration and deceleration of the limbs is largely preserved under viewpoint changes. In our representation, an action is decomposed into movement primitives (corresponding roughly to body parts). We encode the fine-grained temporal dynamics of each movement primitive using a representation that we call the *movement pattern histogram* (MPH). We describe an action as a collection of MPHs (see Fig. 1).

An advantage of video-level pooling methods such as BOF is that computing similarity between representations can be done reliably using $L2$ or χ^2 distance function. In part this is because these representations discard the temporal structure of an action, obviating the need for temporal alignment as a part of the matching process. In contrast, computing similarity between two sets of MPH requires alignment and we describe an novel method to do so using a simultaneous alignment and bipartite matching formulation. Such formulation allows for matching across viewpoints and we present an efficient algorithm to solve it.

Our MPH representation can be used in two ways: 1) as a stand-alone action representation for action recognition/retrieval across multiple viewpoints; and 2) to complement existing BOF representations for action recognition. We demonstrate that our approach outperforms standard representations for cross-view recognition tasks in the IXMAS dataset [27]. We also show that our representation complements existing representations for the classification task in the UCF50 dataset [21]. Finally, we show that our representation yields state-of-the-art results for the task of action retrieval in the novel Stereotypy dataset that we introduce (stereotypies are repetitive body movement patterns frequently associated with autism and are often the target of behavioral therapy). In summary, this paper makes three contributions:

- We introduce the *movement pattern histogram*, a novel representation of actions as a multi-channel temporal distribution of movement primitives.
- We present a novel optimization approach to matching movement pattern histograms across videos based on maximum bipartite graph matching.
- We introduce the Stereotypy dataset, a new annotated video corpus obtained by recording children with autism in a classroom setting[1]. We will make this dataset publicly available.

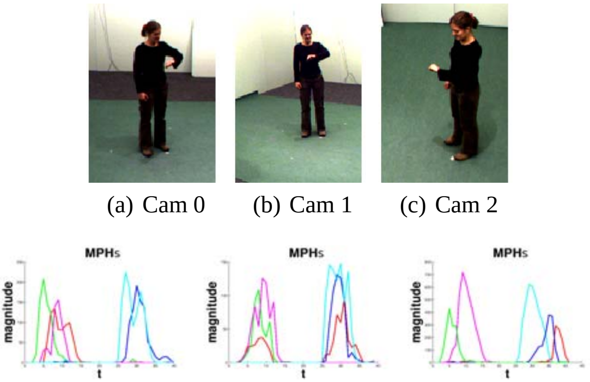
2 Related Work

There is a vast literature on action/activity representation (a recent survey can be found in [20]). A classic representation of action in videos is based on space-time templates [6,11]. While this approach captures the fine-grained detail of an action, it is challenging to achieve robustness to variations. A popular framework used by many authors is the bag-of-features model, with varying local features: interest points [14,5], tracks of points [18,10,26] or frame based descriptors [25]. Many of these methods use descriptors such as HOG/HOF [14,5,26], MBH [26], MIP [13] or shape-flow [25] that are not robust to variations in viewpoint and thus may not support accurate matching of actions across views. Moreover, the BOF framework typically only have a very coarse model of action dynamics (eg. by dividing a video into several subvolumes). *In contrast, our representation captures the fine-grained dynamics of an action while being robust to variations in viewpoint.*

Recently, interesting work has been done to address the challenge of viewpoint variation in action recognition. Liu et.al. [17] tackle the viewpoint problem through transfer learning by building a mapping between codebooks from different viewpoints. However, their framework requires knowledge of the camera viewpoint associated with each action (in testing and training). In a similar spirit, [16,30] learn a series of linear transformations of the feature vector extracted from a video to make it invariant to viewpoint changes. However, a linear transformation is not guaranteed to accurately model view-invariant mapping. Also, performance of their method drops significantly in the absence of multi-view observations of actions in training examples. In addition, [17,16,30] use the shape-flow descriptor that requires extraction of a bounding box and silhouette of an action, which can be challenging in real-world videos. Note that these methods assume a discrete number of pre-defined camera positions, which limits applicability of the methods since the need to collect examples across viewpoints can be burdensome.

Junejo et.al. [10] propose the self similarity matrix (SSM) which exhibits invariance to viewpoint changes. They compute SSM by either point tracking or pairwise frame similarity. However, point tracking is not always accurate and computing pairwise frame similarity means the feature will not be robust to slowly changing background. Another representation robust to changes in viewpoints is the hankelet ([15]). Hankelet is a hankel matrix representation of a tracklet that is invariant to affine transformations. Results in [10] and [15] show that SSM and hankelet are susceptible to large viewpoint changes.

[1] Note that the Stereotypy dataset was collected under an IRB-approved protocol, following best-practices for research with vulnerable subject populations. Consent to publish has been obtained for all images and results.



(a) Cam 0 (b) Cam 1 (c) Cam 2

(d) Cam 0-MPH (e) Cam 1-MPH (f) Cam 2-MPH

Fig. 2. (a)-(c): Three different views of the checkwatch action. (d)-(f): MPH representations of checkwatch for each view. Note the structural similarity of the MPH curves despite huge changes in viewpoint.

3 Action Representation

In this section we describe our action representation, the *movement pattern histogram* (MPH). MPH encodes the global temporal pattern of an action without requiring explicit tracking of features over time. In Sec. 4 we present an iterative method for matching two sets of MPHs.

3.1 The Movement Pattern Histogram

To illustrate our approach, consider the action of a person checking a watch seen from frontal view (Fig. 1). This action can be characterized by the upward movement of the hand and upper arm during the early part of the action (to bring the watch to a readable distance) and the downward movement of the same body parts at the end of the action. We can imagine encoding these body part movements with a cluster of flow vectors, where each cluster explains some portion of the total flow across the video. We denote these clusters as *flow words*. In the check-watch example, the upward hand movement might be mapped to a single flow word. That word would be present in the first half of the frames and absent in the other half (when the hand moves downward).

Given a set of extracted flow words, our goal is to represent an action by encoding the pattern of temporal occurrence of the flow words. In the example of Fig. 1, the green and cyan words occur early in the action (when the hand and upper arm are raised) while the blue and magenta words occur later in the action. We construct an MPH for each flow word which encodes its dynamics.

We now describe the process of constructing the MPH representation. We assume that the video is captured using a static camera (we relax this assumption in Section 3.2). First we compute dense optical flow over the video clip. Then, we use EM to cluster together the flow vectors from all frames based only on the flow direction (we only consider flow vectors whose magnitudes are above a certain threshold). Each flow cluster defines a single flow word. In Figure 1(a)-1(b) we can see the flows color-coded according to the five flow words. We then generate an MPH for each of the flow clusters by binning the flow vectors. Each bin t in the MPH h_c corresponds to frame number t, and contains the sum of flow magnitudes for all pixel flows f that corresponds to cluster c in that frame. Let m_c denote the set of flow vectors that map to cluster c:

$$h_c(t) = \sum_{f(t) \in m_c} \|f(t)\| \tag{1}$$

In Fig. 1(c) we can see that the green MPH corresponding to upward hand movement is active at the beginning of the action and the blue MPH that corresponds to downward hand movement is active at the end. Note that MPH is quite different from other flow-based models such as the histograms of oriented optical flow (HOOF) [4]. HOOF models the distribution of optical flow direction in each frame, making it a viewpoint-dependent representation, while MPH models the temporal distribution of the *magnitudes* of the different flow clusters.

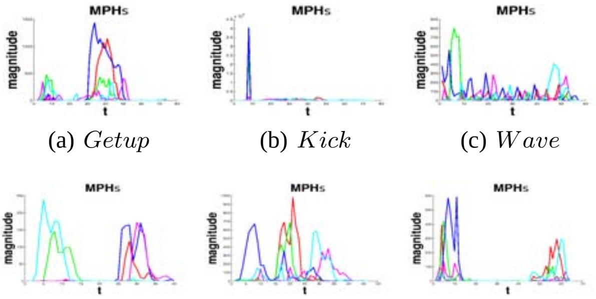
MPH differs in two ways from the standard histogram representations of visual words which are used in action recognition. First, each MPH corresponds to a single flow word and describes the variation in its magnitude over time. In contrast, BOF uses a single fixed histogram describing the co-occurrence of all visual words. Second, the MPH provides a very fine-grained temporal description (one bin per frame) but a very coarse spatial description (all occurrences of a word in a frame are binned together), in order to gain robustness to viewpoint variations.

Figure 2 illustrates the robustness of the MPH representation to viewpoint variation. We can see that the shapes of the MPH sets are quite similar in spite of substantial changes in viewpoint.

Figure 3 shows MPHs for different actions. The MPH representation achieves a certain invariance property under viewpoint changes because it marginalizes out information about appearance, spatial configuration, and flow direction of an action. While spatial configuration and appearance can be important for discriminating certain actions (eg. high punch vs low punch), Fig. 3 demonstrates that the temporal nature of an action can also be very discriminative. Note how MPH captures the dynamics of the different actions: wave (Fig. 3(c)) consists of hand moving left and right and this periodicity is reflected in the MPH. Even in cases where the mechanics of two actions are similar (checkwatch and scratchhead both involve upward and downward movement of the hand), the dynamics of the actions make the MPH sets distinct (Fig. 3(d) vs 3(f)).

3.2 Compensating for Camera Motion

Sometimes action in the real-world is captured using a moving camera. This can cause problems for our representations if we assume that all flows in the video are relevant to the action. To minimize the effect of camera motion we can apply a video stabilization technique such as [7] before computing dense optical flow. However, since we only need to remove the background motion between two consecutive frames (i.e. we don’t need to produce smooth camera trajectory for the whole video), we can apply a simpler solution. We estimate the background motion by computing homography between frames from the optical flow motion vectors (this is similar to [8] but instead of assuming affine motion between frames we use homography). Using the dense optical flow computed, we select a subset of flow vectors located in textured regions (using criteria in [23]) and

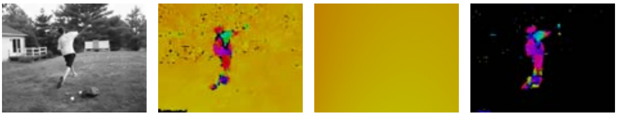


(a) $Getup$ (b) $Kick$ (c) $Wave$

(d) $Checkwatch$ (e) $Punch$ (f) $Scratchhead$

Fig. 3. MPHs of different actions.

perform homography estimation with RANSAC. From the estimated homography, we compute the camera-induced background motion for every pixel in that frame and then subtract the background motion from the computed flow vectors. We do this background motion estimation for every frame in the video and use the corrected flow vectors to compute MPH. Figure 4 shows the result of our motion compensation.



(a) (b) (c) (d)

Fig. 4. Motion compensation results from UCF50: b) Original flow, c) Estimated background motion, d) Motion compensated flow. Flows are color coded following the Middlebury convention.

4 Computing Similarity

Given our new MPH representation, how can we compute similarity between two videos – $target$ and $source$? Accurate similarity measure is important for action recognition and retrieval. Our assumption is that if the two videos correspond to the same action, we can find matching in which the MPH pairs are highly correlated. Let $h_i^t \in \mathbb{R}^{l_t}$ and $h_j^s \in \mathbb{R}^{l_s}$ be the movement pattern histogram for primitives (clusters) i and j in the $target$ and $source$ videos, respectively (l_s and l_t are the number of frames of the videos). Note that since each video is clustered independently, there is no a priori relationship between MPHs from separate videos. Let $T = \{h_1^t, h_2^t, ...h_K^t\}$ and $S = \{h_1^s, h_2^s, ...h_K^s\}$, where K is the total number of flow words in the target and source video. We can construct an undirected bipartite graph $G = (V, E)$ where every single element of T is connected to every single element of S, the vertex set $V = T \cup S$, and $e_{ij} \in E$ is the edge between h_i^t and h_j^s. The weight of edge e_{ij} is the similarity measure between two signals h_i^t and h_j^s. We use the Pearson correlation coefficient (PCC) to compute e_{ij} due to its invariance to scaling: $e_{ij} = PCC(h_i^t, h_j^s) = \frac{cov(h_i^t, h_j^s)}{\sigma_{h_i^t}\sigma_{h_j^s}}$

The similarity between the target and source video is the maximum weighted bipartite matching score of graph G.

Simultaneous alignment and matching

Since an action can be performed at different speeds, the two sets of histograms S and T might not be temporally aligned. This negatively impacts our correlation measure. In order to overcome this problem, we propose a simultaneous alignment and matching method where we iteratively perform alignment and matching of S and T.

Let $\boldsymbol{H_s} = [h^s_1, h^s_2, ...h^s_K]$ and $\boldsymbol{H_t} = [h^t_1, h^t_2, ...h^t_K]$ be the matrices that we construct from S and T. Without loss of generality, let us assume that we normalize the MPH in S and T so that they all have zero mean and unit standard deviation. Also, we zero-pad each vectors h^s_j and h^t_i such that $l_s = l_t = l$. Under this condition, finding the maximum weighted bipartite matching of graph G is equivalent to computing a $K \times K$ binary matrix $\boldsymbol{M}$ that minimizes $C_m = \|\boldsymbol{H_s M} - \boldsymbol{H_t}\|^2_F$, where $\Sigma_i \boldsymbol{M}(i,j) = 1, \Sigma_j \boldsymbol{M}(i,j) = 1$.

To align $\boldsymbol{H_s}$ and $\boldsymbol{H_t}$, we can use dynamic time warping (applying DTW or its variants eg. [31] on a time series data is a common approach for doing activity alignment) to compute binary matrices $(\boldsymbol{D_s}, \boldsymbol{D_t})$ that minimize $C_{dtw} = \|\boldsymbol{D_s H_s} - \boldsymbol{D_t H_t}\|^2_F$, where $\Sigma_j \boldsymbol{D_s}(i,j) = 1$ and $\Sigma_j \boldsymbol{D_t}(i,j) = 1$. Note that DTW optimization infers $\boldsymbol{D_s}$ and $\boldsymbol{D_t}$ using dynamic programming such that the temporal ordering of the rows in $\boldsymbol{H}_{\boldsymbol{s}}$ and $\boldsymbol{H_t}$ is preserved. The DTW solution $(\boldsymbol{D_s}, \boldsymbol{D_t})$ are binary matrices of size $l' \times l$ where l' is the length of the alignment path between $\boldsymbol{H_s}$ and $\boldsymbol{H_t}$. Putting the previous two steps together, we get the final cost function that we want to minimize:

$$\begin{aligned} & C_{mdtw} = \|\boldsymbol{D_s H_s M} - \boldsymbol{D_t H_t}\|^2_F \\ \text{where } & \Sigma_i \boldsymbol{M}_{ij} = 1,\ \Sigma_j \boldsymbol{M}_{ij} = 1 \\ & \Sigma_j \boldsymbol{D_s}(i,j) = 1 \\ & \Sigma_j \boldsymbol{D_t}(i,j) = 1 \end{aligned} \qquad (2)$$

Optimizing C_{mdtw} is a non-convex optimization problem with respect to the matching matrix $\boldsymbol{M}$ and alignment matrices $\boldsymbol{D_s}$ and $\boldsymbol{D_t}$. We can perform iterative optimization by alternating between computing $(\boldsymbol{D}_{\boldsymbol{s}}, \boldsymbol{D_t})$ and $\boldsymbol{M}$:

1. Set $\boldsymbol{M}$ as $K \times K$ identity matrix
2. Fix $\boldsymbol{M}$ and minimize $C_{dtw} = \left\|\boldsymbol{D_s H^m_s} - \boldsymbol{D_t H_t}\right\|^2_F$, where $\boldsymbol{H^m_s} = \boldsymbol{H_s M}$, to optimize for $(\boldsymbol{D_s}, \boldsymbol{D_t})$
3. Fix $(\boldsymbol{D_s}, \boldsymbol{D_t})$ and minimize $C_m = \left\|\boldsymbol{H^{dtw}_s M} - \boldsymbol{H^{dtw}_t}\right\|^2_F$, where $\boldsymbol{H^{dtw}_s} = \boldsymbol{D_s H_s}$ and $\boldsymbol{H^{dtw}_t} = \boldsymbol{D_t H_t}$, to optimize for $\boldsymbol{M}$
4. Iterate 2-3 until convergence

Both step 2 and 3 monotonically decrease/non-increase C_{mdtw}. Since C_{mdtw} has a lower bound of 0 this optimization will converge. DTW can be solved in $O(l^2)$ and minimizing C_m using Hungarian algorithm takes $O(K^3)$. Hence the complexity of this algorithm is $O(l^2) + O(K^3)$ and since l and K are typically small this is efficient to compute (l is typically between 60-150 depending on how long the action is. K depends on the number of peaks of the flow distribution, typically between 4-6). Empirically we observe that this optimization converges after 2-3 iterations.

To optimize for $\boldsymbol{M}$, the task is to find the set of edges $e_{ij} \in E$ that defines a perfect matching in G such that the sum of the edges in the matching is maximum. We solve this using the Hungarian algorithm to compute a set of λ for the following problem:

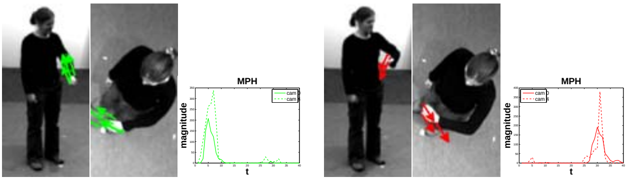
$$\begin{aligned} \max_{\lambda} \quad & \sum_{(i,j)\in E} \lambda_{ij} e_{ij} \\ \text{s.t.} \quad & \sum_{j\in N(i)} \lambda_{ij} = 1 \; \forall i \in source \\ & \sum_{i\in N(j)} \lambda_{ij} = 1 \; \forall j \in target \\ & \lambda_{ij} \in \{0,1\} \end{aligned} \tag{3}$$

where e_{ij} is the correlation between the i-th column of $\boldsymbol{H_s}$ and j-th column of $\boldsymbol{H_t}$, and $N(i)$ is the set of vertices that are adjacent to vertex i.

After obtaining the λ for maximum matching, we define the similarity score between two videos as the maximum weighted bipartite matching score of graph G:

$$score = \sum_{(i,j)\in E} \lambda_{ij} e_{ij} \tag{4}$$

Figure 5 illustrates an example of the matching result. Note that while the two actions are captured from widely different viewpoints, our matching algorithm is able to establish the correspondence between flow clusters by exploiting the temporal property of MPH. For instance, the matched MPH pair 1 (Fig. 5(c)) corresponds to flow words that belong to the hand while it is moving up at the beginning of the action (Fig. 5(a)-5(b)) and the matched pair 3 (Fig. 5(f)) corresponds to flow words of the hand while it is moving down at the end of the action (Fig. 5(d)-5(e)). This intuitive interpretation of the matching result is possible since our flow words map to well-defined spatial regions in the video. We believe this interpretability is a highly desirable property for real-world applications.



(a) C0 (b) C4 (c) Pair 1 (d) C0 (e) C4 (f) Pair 3

Fig. 5. Matching of cam 0 and cam 4 for checkwatch. Note how the matched MPH pair correspond to the same body part movements ((a)-(b): Hand moving up, (d)-(e): Hand moving down).

5 Experimental Results

To evaluate the performance of our method we performed experiments on the IXMAS dataset [27], UCF50 dataset [21] and a new real-world Stereotypy dataset that consists of a collection of videos ranging from 10 to 20 minutes each (with total length of 2 hours). We consider three tasks. First, to show robustness of our representation to variations in viewpoint, we perform cross-view action recognition experiments on the

IXMAS dataset. Second, to show how our feature complements BOF representation, we perform action recognition on the UCF50 dataset. Finally, we demonstrate the power of our approach on a real-world problem by doing action retrieval on the Stereotypy dataset.

For the action retrieval task, we compare retrieval results against several BOF representations: cuboid [5], self-similarity matrix (SSM) [10], shapeflow [25], dense trajectories [26], and cuboid+shapeflow (used in [17,16,30]). Note that BOF representation has been previously used for action retrieval (eg. [3]).

To compute MPH we used GPU-based dense optical flow [28]. To select the K for MPH we examined the number of peaks in the distribution of flow directions in sample videos. We chose $K = 5$ (5 MPH per video) for all experiments. The details for competing methods are in the supplemental material.

5.1 Results on IXMAS Dataset

The IXMAS dataset contains videos of 11 types of actions captured from 5 viewpoints. There are 30 examples per action performed by several actors. We perform the standard cross-view classification task on the IXMAS dataset and compare it against methods described in [10,16,15,30]. It is important to note that in this particular experiment we are not assuming any view-correspondence in the training data. For this experiment we use 1NN classifier and a 6-fold cross validation procedure (identical cross-validation procedure as in [16,15,30]).

Table 1. Classification results by using a single view for training on IXMAS. Each row is a training view, and column a test view.

	Test View					
	c0	c1	c2	c3	c4	Avg.
	Ours, [16], [15], [10]	Ours, [16], [15], [10]	Ours, [16], [15], [10]	Ours, [16], [15], [10]	Ours, [16], [15], [10]	Ours, [16], [15], [10]
c0		80.3, 63.6, **83.7**, 75.2	63.6, 60.6, 59.2, **69.7**	68.5, 61.2, 57.4, **71.8**	**56.4**, 52.6, 33.6, 49.4	**67.2**, 59.5, 58.5, 66.5
c1	80.0, 61.0, **84.3**, 78.5		62.1, 62.1, 61.6, **67.9**	59.7, 65.1, 62.8, **71.5**	47.9, **54.2**, 27.0, 48.0	62.4, 60.6, 58.9, **66.5**
c2	63.6, 63.2, 62.5, **70.0**	62.1, 62.4, 65.2, **73.0**		**79.7**, 71.7, 72.0, 68.5	**75.5**, 58.2, 60.1, 55.2	**70.2**, 63.9, 64.9, 66.7
c3	67.0, 64.2, 57.1, **73.6**	65.8, 71.0, 61.5, **72.4**	**83.6**, 64.3, 71.0, 67.3		46.4, **56.6**, 31.2, 45.9	**65.7**, 64.0, 55.2, 64.8
c4	**54.5**, 50.0, 39.6, 44.5	49.4, **59.7**, 32.8, 41.5	**72.1**, 60.7, 68.1, 55.2	50.0, **61.1**, 37.4, 37.9		56.5, **57.9**, 44.5, 44.8
Avg.	66.3, 59.6, 60.9, **66.7**	64.4, 64.2, 60.8, **65.5**	**70.4**, 61.9, 65.0, 65.0	64.5, **64.8**, 57.4, 62.4	**56.5**, 55.4, 38.0, 49.6	**64.4**, 61.2, 56.4, 61.9

We focus on two recognition tasks: 1) classifying videos captured from the *test* view using training data captured from the *train* view, and 2) classifying videos captured from the *test* view using training data from all of the other views. We compare against results in [16,15,30] using the non-correspondence mode, since in many applications the need to have multi-view correspondence in training data can be burdensome.

The results for the first recognition task (classifying videos from the test view using training from the train view) can be seen in Table 5.1. For this task, our method improves the average recognition accuracy by 2.5% compared to the next best approach (see the highlighted cell in Table 5.1). Hankelet ([15]) is only robust to affine transformation and thus achieves low accuracy when classifying videos trained from very different viewpoints (eg. c0 vs c4).

The results for classifying videos from the test view by using training from all of the other views can be seen in Table 2. Note that even though the result of [10] is achieved by including videos from all the views (including the test view) for training, our approach still yields the best result. *Our method can use the additional training views more effectively due to its ability to generalize across viewpoints.*

Table 2. Cross-view recognition accuracy on IXMAS (trained on videos captured from all views except the $test$ view). Note how our representation gives a significantly more accurate result.

Method	**Test View**					
	c0	*c1*	*c2*	*c3*	*c4*	*Avg.*
Ours	**83.9**	**81.8**	**87.6**	**83.0**	**73.6**	**82.0**
[30] (test view used for transfer learning)	66.4	73.5	71.0	75.4	66.4	70.5
[16] (test view used for transfer learning)	62.0	65.5	64.5	69.5	57.9	63.9
[10] (trained on all cameras)	77.0	78.8	80.0	73.9	63.6	74.6

5.2 Results on UCF50 Dataset

The UCF50 dataset contains 6618 videos of 50 types of actions. For this experiment we use the leave-one-group-out (LoGo) cross validation as suggested in [21].

Many videos in UCF50 were captured using low-res handheld cameras with various motion artifacts due to camera shake and rolling shutter. Clearly, the fine-grained motion features that our method exploits are difficult to extract in this case. However we still believe that it is valuable to characterize the limitations of our approach by analyzing the UCF50. Another important characteristic of this dataset is that the scene context gives a significant amount of information about the type of action in the video. For example, many of the actions are performed using a specific set of instruments (eg. barbell in bench press) and representing those cues can help immensely for classification. This suggests the need to combine our representation (which only models the dynamics of an action) with a complementary appearance-based representation.

We combine our representation with Fisher Vector (FV) encoding [19] (which can be seen as an extension to BOF) of the dense trajectory descriptor described in [26]. To convert our pairwise action similarity measure to a feature vector we use a method similar to ActionBank [22]. In ActionBank, the videos in the training set function as the bases of a high-dimensional action-space. For example, if we have N videos in the training set, the feature vector for video v is a vector of length N where the value of $N(i)$ is our similarity measure between video v and the i-th video in the training set. The full feature vector for each video is then simply a concatenation of the FV representation of dense trajectory and our ActionBank-like representation. For this experiment we use 1-vs-all linear SVM (with $C = 0.1$) for training and classification.

Classification results on this dataset can be seen in Table 3. The accuracy improvement obtained by adding our representation suggests that MPH encodes information that is complementary to HOG, HOF and MBH.

Comparing results of MPH + FV of dense trajectory against only FV of dense trajectory, the most significant improvement in accuracy comes from the class PizzaTossing (an improvement of 10.5% from 65.8% to 76.3%). A large part of this improvement comes from a better discrimination between PizzaTossing and Nunchucks classes. Many of the videos of these these two classes share a significant similarity in appearance: a person performing an action in a small room captured from close to frontal view. Thus, MPH (which models the dynamics of the action) increases discrimination between these two classes. Another notable improvement comes from the class RockClimbing (an improvement of 6.9% from 85.4% to 92.3%). About half of the improvement for this class comes from a better discrimination against RopeClimbing. While the actual movement of climbing a rope vs climbing a wall with a rope is different, the context of these two classes are very similar since wall and rope tend to be the prominent features in the video. Thus, MPH provides a powerful cue to help discriminate between these two classes. On the other hand, MPH can also increase confusion between classes. We observe the biggest drop in accuracy in the class HorseRace (a decrease of 3.1% from 98.4% to 95.3%) partly due to increased confusion with Biking. This is likely due to the fact that from a distance, the movement dynamics of HorseRace and Biking look similar: people moving on a trajectory with their body moving slightly up-and-down with a particular frequency. Human action is a complex concept defined by the interplay of a number of elements: movements, human pose, instruments used, and surrounding background context. A better approach to modellng any of these elements is a step towards a better action representation.

Table 3. Classification results on UCF50.

Method	**Accuracy (LoGo)**
Ours (MPH) + FV of [26]	**90.5**
Dense trajectories [26] w/ FV encoding	88.9
MBH + scene context[21]	76.9
GIST3D + STIP [24]	73.7
MIP [13]	72.7

5.3 Results on Stereotypy Dataset

We also address the problem of *action retrieval*: Given a single example video clip containing an action of interest, the task is to retrieve all matching instances of that action from an unstructured video collection. The strength of our bottom-up matching approach is that it can compute a similarity measure between activities without learning. It can therefore be used in situations where the space of possible activities is very large and difficult to define a priori and when it is difficult to find an extensive amount of training examples across different views.

In the domain of behavioral psychology, there is currently great interest in studying the effectiveness of behavioral therapy for children with autism [2]. These children frequently exhibit repetitive motor movements, known as *stereotypies*. In comparison to more traditional functional activities, stereotypies are often unique expressions of individual behavior and highly person-dependent, making it challenging to construct a general model of such behaviors [1]. At the same time, it would be very useful to be able

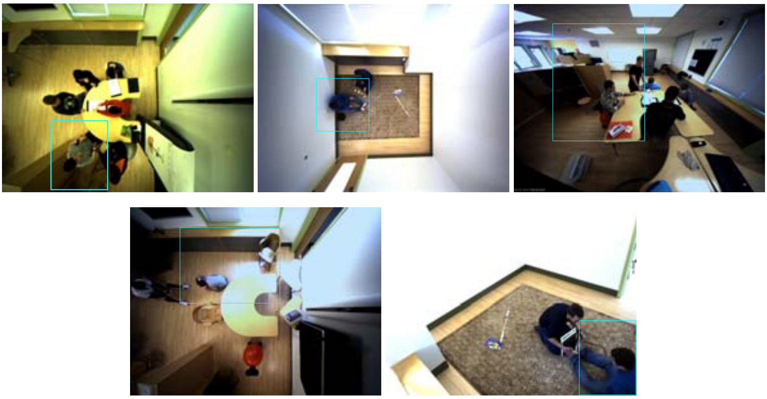


Fig. 6. The Stereotypy dataset. Region of interest is indicated by the bounding box.

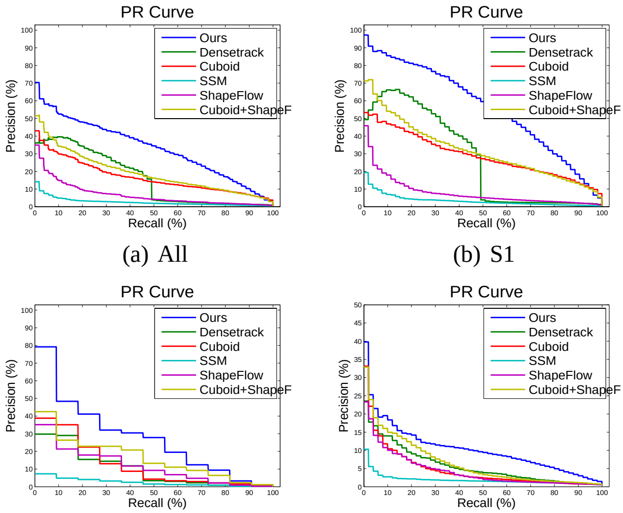
to retrieve all instances of a particular stereotypy exhibited by a child across multiple therapy sessions given only a single example. We conducted an experiment to evaluate the effectiveness of our algorithm in this context.

We collaborated with experimental and educational psychologists on analyzing videos obtatined of children with autism who engage in stereotypies in a classroom setting. The dataset consists of videos captured from various viewing locations. Representative frames are shown in Figure 6 (note the variations in viewpoints and appearance of the videos). We are interested in three types of stereotypies exhibited by the children: jumping up from chair (S1), jumping on the floor (S2), and paddling movement of the hands (S3). A psychologist with autism expertise and familiarity of the children provided ground truth labels for the stereotypies. The dataset contains 54 instances of S1 behavior, 12 instances of S2 behavior and 51 instances of S3 behavior. For each video in the dataset we manually identified a bounding box of the region of interest, which defines the input for the retrieval task.

We used sliding window to split the videos into a series of 60-frame clips where the window slides 15 frames at a time. We extracted 12410 clips from all of the videos in the dataset.

Action retrieval task: We identified all clips containing stereotypies, and used eac of those clips as the $target$ input for retrieval. Given a $target$ clip, we computed the similarity of the clip against the rest of the clips in the dataset. The similarity score for our algorithm is described in Section 4. For BOF, we found that $L2$ distance between the normalized feature vectors yielded the best results. We then ranked the videos according to the similarity score and measured performance by using precision-recall (PR) curve, a common metric for retrieval. We counted a clip as a hit if it overlapped with at least 50% of the groundtruth annotation.

The PR curves for retrieving stereotypies can be seen in Figure 7(a). Note that our method performed significantly better than various BOF representations. This is likely because a therapist sometimes came to interact with a child during the course of the video (Fig. 8(b)) and the child often moved, changing his relative angle to the camera. These variations will affect any appearance-based representation. The accuracy of dense trajectories drops significantly after around 50% recall. While dense trajectories representation is good at capturing the discriminative aspect of the behavior, it is very viewpoint dependent and thus can only retrieve instances of the behavior captured from the same viewpoint (the videos in the dataset are captured either from overhead view or side view). Note that shapeflow and SSM perform especially poorly in this setting due to the therapist sometimes appearing in the background. Shapeflow relies on successful extraction of silhouette of the foreground actor (we use [12] for computing silhouette), which is challenging in videos. SSM uses pairwise frame similarity, thus a slowly changing background (eg. due to the therapist moving) has a huge effect on the representation.



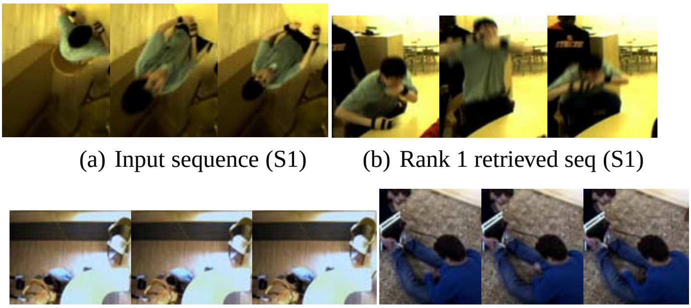
(a) All (b) S1

(c) S2 (d) S3

Fig. 7. Precision-recall for curves for all behaviors.

To better illustrate our retrieval results, we performed the retrieval task using a single example of S1 that can be seen in Figure 8(a). In this particular example, we were able to retrieve a clip containing another S1 behavior ranked 1 in the retrieval results. Note that since our representation is agnostic to camera viewpoint, the retrieved results can contain clips captured from viewpoints that are different from the input (Fig. 8(b)). Behavior S1 has very distinct dynamics and as a result our approach performed very well, often able to retrieve the top 3 results with 100% accuracy. The PR curves for the S1 behavior can be seen in Figure 7(b).

Another visual example of our retrieval results can be seen in Figure 8(c)-8(d). Note how our method is able to retrieve the same behavior under massive variations in appearance (different room, clothing, viewpoint, lighting condition and scaling). Indeed in real-world videos, it is often the case that we can not control elements of the scene



(a) Input sequence (S1) (b) Rank 1 retrieved seq (S1)

(c) Input sequence (S3) (d) Rank 1 retrieved seq (S3)

Fig. 8. Retrieval results for behavior S1 (a-b) and S3 (c-d).

that have a large effect on the appearance of the subject such as clothing worn, subject's orientation with respect to the camera and lighting conditions. In the absence of training data it is difficult to learn how to discount these variations. Our motion-based matching approach provides a powerful tool in this setting.

Behavior S3 contains a lot of instances where the hands are occluded by the child's own torso or objects such as a chair. Due to occlusion, there will be some MPHs that are observable from one view, but not the other. Note that this occlusion problem affects all methods that rely on seeing movements to extract a representation (such as all of the interest-points-based methods). In our representation, the number of MPHs not occluded often will be sufficient for computing similarity between activities. The quantitative result for the paddling behavior can be seen in Figure 7(d). Given the difficulty of the task, our method was able to produce reasonable results even though there were a significant number of occlusions and pose changes in the videos.

6 Conclusions

We present a novel action representation that encodes the fine-grained dynamics of an action and is robust to variations in appearance. Our simultaneous matching and alignment formulation explicitly handles variations in the dynamics of an activity and allows matching of features extracted from different viewpoints. Our representation naturally complements existing BOF representations and performs well on traditional action recognition datasets as well as on a new real-world stereoypy dataset.

7 Acknowledgment

This research was supported in part by NSF Awards 0960618 and 1029679.

References

1. Albinali, F., Goodwin, M.S., Intille, S.S.: Recognizing stereotypical motor movements in the laboratory and classroom: A case study with children on the aautism spectrum. In: Proceedings of the 11th international conference on Ubiquitous computing. pp. 71–80. ACM (2009)
2. Association, A.P.: Diagnostic and Statistical Manual of Mental Disorders DSM-IV-TR, Fourth Edition, Text Revision. American Psychiatric Pub. (2000)
3. Cao, L., Ji, R., Gao, Y., Liu, W., Tian, Q.: Mining spatiotemporal video patterns towards robust action retrieval. Neurocomputing (2012)
4. Chaudhry, R., Ravichandran, A., Hager, G., Vidal, R.: Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. CVPR (2009)
5. Dollar, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior Recognition via Sparse Spatio-Temporal Features. ICCV-VS PETS (2005)
6. Gorelick, L., Blank, M., Shechtman, E., Irani, M., Basri, R.: Actions as Space-Time Shapes. PAMI (2007)
7. Grundmann, M., Kwatra, V., Essa, I.: Auto-directed video stabilization with robust l1 optimal camera paths. In: CVPR (2011)
8. Jain, M., Jégou, H., Bouthemy, P., et al.: Better exploiting motion for better action recognition. CVPR (2013)
9. Johansson, G.: Visual Perception of Biological Motion and a Model for Its Analysis. Attention, Perception, & Psychophysics 14(2), 201–211 (1973)
10. Junejo, I., Dexter, E., Laptev, I., Pérez, P.: View-Independent Action Recognition from Temporal Self-Similarities. PAMI (2010)
11. Ke, Y., Sukthankar, R., Hebert, M.: Volumetric features for video event detection. IJCV (2010)
12. Kim, K., Chalidabhongse, T.H., Harwood, D., Davis, L.: Real-time foreground–background segmentation using codebook model. Real-time imaging 11(3), 172–185 (2005)
13. Kliper-Gross, O., Gurovich, Y., Hassner, T., Wolf, L.: Motion interchange patterns for action recognition in unconstrained videos. ECCV (2012)
14. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning Realistic Human Actions from Movies. In: CVPR (2008)
15. Li, B., Camps, O.I., Sznaier, M.: Cross-view activity recognition using hankelets. CVPR (2012)
16. Li, R., Zickler, T.: Discriminative Virtual Views for Cross-View Action Recognition. CVPR (2012)
17. Liu, J., Shah, M., Kuipers, B., Savarese, S.: Cross-View Action Recognition via View Knowledge Transfer. In: CVPR (2011)
18. Messing, R., Pal, C., Kautz, H.: Activity Recognition Using the Velocity Histories of Tracked Keypoints. In: ICCV (2009)
19. Perronnin, F., Sánchez, J., Mensink, T.: Improving the fisher kernel for large-scale image classification. ECCV pp. 143–156 (2010)
20. Poppe, R.: A survey on vision-based human action recognition. Image and vision computing 28(6), 976–990 (2010)
21. Reddy, K.K., Shah, M.: Recognizing 50 human action categories of web videos. Machine Vision and Applications (2012)
22. Sadanand, S., Corso, J.J.: Action bank: A high-level representation of activity in video. In: CVPR (2012)
23. Shi, J., Tomasi, C.: Good features to track. CVPR (1994)

24. Solmaz, B., Assari, S.M., Shah, M.: Classifying web videos using a global video descriptor. Machine Vision and Applications (2012)
25. Tran, D., Sorokin, A.: Human Activity Recognition with Metric Learning. ECCV (2008)
26. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Dense trajectories and motion boundary descriptors for action recognition. IJCV (2013)
27. Weinland, D., Boyer, E., Ronfard, R.: Action Recognition from Arbitrary Views using 3D Exemplars. IJCV (2007)
28. Werlberger, M., Trobin, W., Pock, T., Wedel, A., Cremers, D., Bischof, H.: Anisotropic Huber-L1 optical flow. BMVC (2009)
29. Yuan, C., Li, X., Hu, W., Ling, H., Maybank, S.: 3d r transform on spatio-temporal interest points for action recognition. CVPR (2013)
30. Zhang, Z., Wang, C., Xiao, B., Zhou, W., Liu, S., Shi, C.: Cross-view action recognition via a continuous virtual path. CVPR (2013)
31. Zhou, F., de la Torre, F.: Canonical Time Warping for Alignment of Human Behavior. NIPS (2009)