Image and Vision Computing 30 (2012) 206–216

Contents lists available at SciVerse ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Using a Product Manifold distance for unsupervised action recognition[☆,★]

Stephen O'Hara[*], Yui Man Lui, Bruce A. Draper

Computer Science Department, Colorado State University, Fort Collins, CO 80523, United States

ARTICLE INFO

Article history:
Received 1 July 2011
Received in revised form 2 November 2011
Accepted 4 November 2011

Keywords:
Unsupervised learning
Product Manifold
Action recognition

ABSTRACT

This paper presents a method for unsupervised learning and recognition of human actions in video. Lacking any supervision, there is nothing except the inherent biases of a given representation to guide grouping of video clips along semantically meaningful partitions. Thus, in the first part of this paper, we compare two contemporary methods, Bag of Features (BOF) and Product Manifolds (PM), for clustering video clips of human facial expressions, hand gestures, and full-body actions, with the goal of better understanding how well these very different approaches to behavior recognition produce semantically relevant clustering of data.

We show that PM yields superior results when measuring the alignment between the generated clusters and the nominal class labeling of the data set. We found that while gross motions were easily clustered by both methods, the lack of preservation of structural information inherent to the BOF representation leads to limitations that are not easily overcome without supervised training. This was evidenced by the poor separation of shape labels in the hand gestures data by BOF, and the overall poor performance on full-body actions.

In the second part of this paper, we present an unsupervised mechanism for learning micro-actions in continuous video streams using the PM representation. Unlike other works, our method requires no prior knowledge of an expected number of labels/classes, requires no silhouette extraction, is tolerant to minor tracking errors and jitter, and can operate at near real-time speed. We show how to construct a set of training “tracklets,” how to cluster them using the Product Manifold distance measure, and how to perform detection using exemplars learned from the clusters. Further, we show that the system is amenable to incremental learning as anomalous activities are detected in the video stream. We demonstrate performance using the publicly-available ETHZ Livingroom data set.

1. Introduction

Although there is a great deal of research relating to the recognition of human behaviors and actions in video, much of the research to date has focused on the problem of classifying short video segments according to a small, fixed set of labels. Benchmark data sets typically include pre-segmented clips that show only a single behavior from less than a dozen possibilities. The performance task is to classify the clips. While the forced-choice paradigm has led to notable performance gains over the past five years, it leaves many questions unanswered regarding the larger challenge of detecting and recognizing human behaviors in less structured contexts and in continuous streams of input.

Action recognition is hard, and it is reasonable to attempt to simplify the problem using controlled data sets. However, in deference to the no-free-lunch theorem [3], the techniques used to push performance to the highest levels on classification benchmarks may not yield substantial gains in addressing the more general challenges relating to action recognition in less controlled, streaming data sources. Recent results from the Contest on Semantic Description of Human Activities (SDHA Challenge) [4] indicate that existing space-time feature-based approaches perform well on classification, yet detection in continuous videos remains difficult.

Additionally, it is desirable to develop learning methods that require minimal supervision because of the difficulty in curating and labeling large data sets and because of the difficulty in generalizing many forced-choice algorithms to uncontrolled environments. Human behavior recognition in streaming video, under real world conditions, is the challenge facing those trying to detect suspicious pedestrian behavior in subway stations, trying to automatically annotate a movie, trying to build household robotics to assist the elderly, and so on.

One aspect of the larger challenge is the unsupervised grouping of behaviors outside of closed-world assumptions. In the first part of this article, we seek to understand how contemporary action recognition techniques lend themselves to open-ended clustering, where even the number of clusters is unknown. Lacking any supervision, there is nothing except the inherent biases of a given technique to guide grouping of video clips. One might expect poor alignment between

[☆] Portions of this article previously appeared in conference proceedings: [1,2].

[★] Research partially supported by DARPA Contract W911NF-10-2-0066.

[*] Corresponding author.
E-mail addresses: svohara@cs.colostate.edu (S. O'Hara), lui@cs.colostate.edu (Y.M. Lui), draper@cs.colostate.edu (B.A. Draper).

0262-8856/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2011.11.001

the unsupervised clustering and the desired labels (classes) of a given data set. Perhaps surprisingly, this is not always so. In fact, a recent Product Manifold technique (from Lui et al. [5]) for measuring the distance of videos generates clusters on the KTH Actions benchmark that are over 90% aligned with the nominal class labels, as we show in Part 1. We also show that over three different data sets, the Product Manifold distance measure consistently clusters the data more accurately with respect to the nominal class labeling than a competing Bag of Features method. To understand how algorithm biases impact clustering, we explore alternative labelings of the data to measure how well they align with a given aspect of similarity among the video clips, as measured by either manifold-based or features-based representations.

A second aspect of the larger challenge is the ability to learn and recognize actions from continuous video streams. In realistic applications of human action recognition, the temporal and spatial localization of the actions will be unknown and the people being observed are likely to exhibit several different actions over time. In Part 2 of this article, we present a method of unsupervised learning of human micro-actions from long duration videos, based on computing distances between short track segments (called "tracklets") using a Product Manifold mapping. There is some variation in the use of the term *micro-action* in the literature. Here, we use it to mean a short-duration, single-entity action that can be recognized by a human observer with only a few seconds of video. Our method is efficient, can be trained relatively quickly, and can perform detections in near-real-time. We require that the entities of interest be detected and tracked over enough frames to observe any given micro-action, yet our tracklet extraction strategy mitigates minor tracking accuracy issues that are commonly encountered. We do not require any silhouette extraction or part detections of the subjects. We make no assumption on the number of micro-actions an entity may exhibit in any given length of time, and we allow for multiple labels to be applied simultaneously.

The rest of this paper is organized as follows. First we provide background material covering related work and the Product Manifold distance measure. Following the background, we divide the article into two parts, each addressing one of the two challenges outlined above. Part 1 presents our comparison of the Product Manifold distance to Bag of Features for unsupervised clustering of human actions, gestures, and expressions. Part 2 presents our approach to unsupervised recognition of human micro-actions in streaming video using the Product Manifold distance. We end with our conclusions and future work.

2. Background

This section provides relevant background on action recognition approaches and provides an overview of the Product Manifold distance measure. A recent survey on human action recognition [6] can provide additional background.

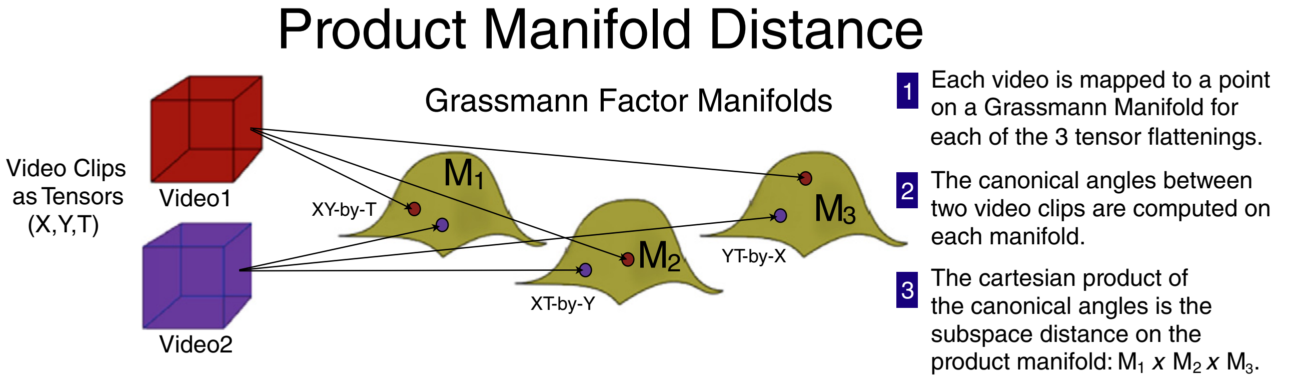
2.1. Related work

The Bag of Features approach has become one of the most popular methods for human action recognition in short video clips [7–14]. As adapted from similar methods of image classification and retrieval, Bag of Features approaches represent video clips as unordered sets of local space-time features. Features are quantized into discrete vocabularies, or codebooks. The space-time features in a video are assigned to their nearest neighbors in the codebook. The Bag of Features representation is typically a normalized histogram. Activity classification is often done by applying Support Vector Machines with appropriate kernels (χ^2 is common) to the Bag of Features representations.

There are many choices involved when implementing a Bag of Features approach. One must decide how to sample the video to extract localized features. Possible sampling strategies include space-time interest point operators, grids/pyramids, or random sampling. Each strategy comes with parameters including space and temporal scales, overlap, and other settings. From the sampled regions, an appropriate descriptor must be chosen to provide a balance between discrimination, robustness to small photometric and geometric perturbations, and compactness of representation. Wang et al. provide an evaluation of popular space-time interest point detectors and features [15], yet there is no conclusive result indicating which combination of detector and descriptor is best. The results are data-set dependent. Beyond feature detection and extraction, other design choices include codebook size, quantization method (e.g. K-Means), and distance function to be used in nearest-neighbor assignments.

Advantages of the Bag of Features approach include the relative simplicity of the representation compared to graphical or constellation models, and the lack of any requirement to pre-process the videos to perform segmentation, track moving objects, or any other image processing task beyond feature detection. As such, they are attractive for use in unsupervised systems that are designed to sample their environment and learn patterns without prior knowledge. The disadvantages include the difficulty in knowing precisely why two videos are considered similar, as there is little semantic meaning in the representation. For example, it is possible to correctly classify videos due to co-varying, but semantically irrelevant, background artifacts in the data set.

Departing from the bag-of-features works are silhouette motion methods such as those from Lin et al., Nater et al., and Guo et al. [16–18]. Lin employs joint likelihood maximization between the current observation and learned shape-motion prototypes. Nater clusters silhouettes and motion patterns and recognizes anomalous activities via outlier thresholding. Guo uses a sparse representation framework for action


Product Manifold Distance
Grassmann Factor Manifolds
Video Clips as Tensors (X,Y,T)
Video1
Video2
XY-by-T
XT-by-Y
YT-by-X
M_1
M_2
M_3

1 Each video is mapped to a point on a Grassmann Manifold for each of the 3 tensor flattenings.

2 The canonical angles between two video clips are computed on each manifold.

3 The cartesian product of the canonical angles is the subspace distance on the product manifold: $M_1 \times M_2 \times M_3$.

Fig. 1. Illustration of the Product Manifold Distance.

recognition. The sparse representations of actions are derived from covariance matrices of silhouette tunnel features.

Manifold-based methods [19,5] present an alternative approach to those based upon localized sampling or silhouette matching. Manifold-based approaches often attempt to map the high-dimensional video data into a lower dimensional space with some regular structure, such as a differentiable manifold. If a video can be represented as a point on a manifold, then the distance between two videos is the geodesic distance between the points. Assuming that the geodesic distance can be efficiently computed or approximated, it can be used to classify or cluster the corresponding videos. A state-of-the-art example of this approach is from Lui et al. [5], where the Product Manifold distance is introduced. Lui shows that the Product Manifold distance coupled with a simple nearest neighbor classifier outperforms competing methods on Cambridge Gestures and KTH Actions data sets. For an overview of the application of matrix manifolds to a variety of computer vision problems, we refer the reader to a contemporary survey by Lui [20].

There have been other investigations of unsupervised learning of actions. Niebles et al. [21] use probabilistic Latent Semantic Analysis (pLSA) to learn actions using a Bag of Features representation. Their method is not completely unsupervised because it takes advantage of a validation stage to select an optimal codebook and uses the number of classes to constrain an expectation maximization procedure. Nater et al. [17] employs unsupervised hierarchical learning of action silhouettes and motion patterns. Gilbert et al. [22] present a user-in-the-loop method of weakly supervised image and video clustering.

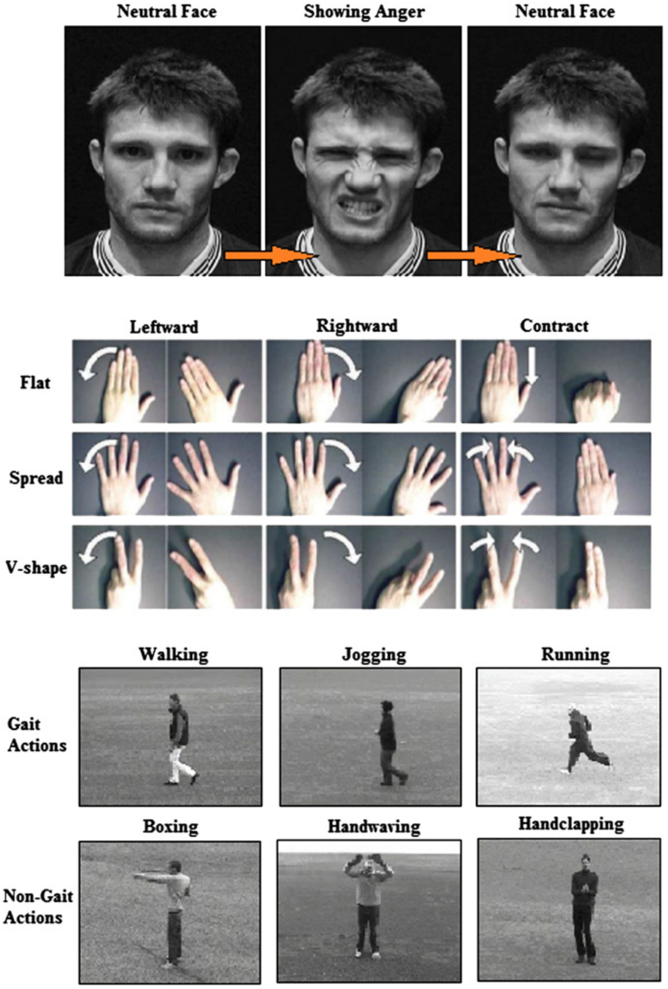


Fig. 2. From top to bottom: Expressions [8], Cambridge Gestures [23], and KTH Actions [7] data sets.

2.2. Product Manifold Distance

In this section, we provide an overview of Lui's Product Manifold Distance (PM Distance). Fig. 1 illustrates the process of computing the PM Distance between a pair of videos. A video can be represented as a stack of sequential images forming a data cube of dimension (x,y,t), where x and y are the width and height of the images and t is the number of frames. This data cube is a 3-mode tensor, which can be flattened from 3D to 2D along each of the dimensions: $(x,yt),(y,tx),(t,xy)$. Each tensor flattening is a matrix which can be factored using SVD to generate an orthonormal matrix, which defines a subspace represented by a point on a Grassmann manifold. Thus, the data cube of the video becomes three points, one on each of three separate Grassmann manifolds. The distance between two points on a single Grassmann manifold is the subspace distance represented by a set of canonical angles (also called principal angles) between the spaces. Computing the canonical angles requires some simple linear algebra. If A and B are orthogonal matrices, the singular values of $svd(A^TB)$ are the cosines of the canonical angles between the column spaces spanned by A and B, respectively.

There exists a Product Manifold which is the product of the three Grassmann manifolds. Each video is a point in the Product Manifold structure. The distance between video clips is the geodesic distance on the Product Manifold, which can be computed using the Cartesian product of the canonical angles between the points on the factor manifolds. Given canonical angles $\Theta=(\theta_1,\theta_2,\ldots,\theta_n)$, the chordal distance is the L2 norm of the component-wise sine function, $||sin\Theta||_2$. The chordal distance on the Cartesian product of the three sets of canonical angles is the PM Distance.

The advantages of the Product Manifold approach include the relatively small number of design choices, the lack of any training or lengthy codebook generation process, and its computational speed. The disadvantage of this method is the requirement to use fixed-size cubes in the representation. The video clips from the data sets must be cropped or scaled to a uniform-sized cube. The method works best when the activity in the videos is roughly aligned, although it is important to note that Lui's reported results on the KTH dataset includes classes where the actor is moving in different directions and undergoing scale changes, etc.

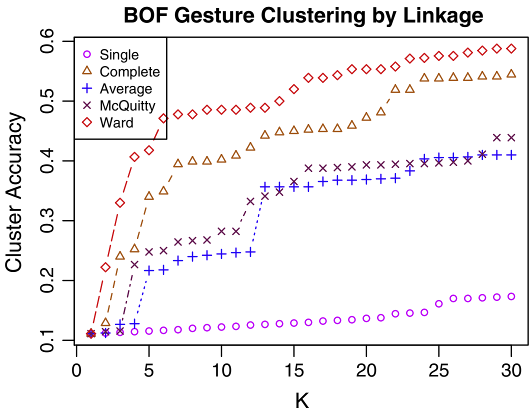


Fig. 3. Comparison of hierarchical clustering linkage methods. This graph is formed using the Bag of Features method on the Gestures data set, but performance was similar for both algorithms on all three data sets. All the other experiments in this paper employ Ward's linkage for hierarchical clustering.

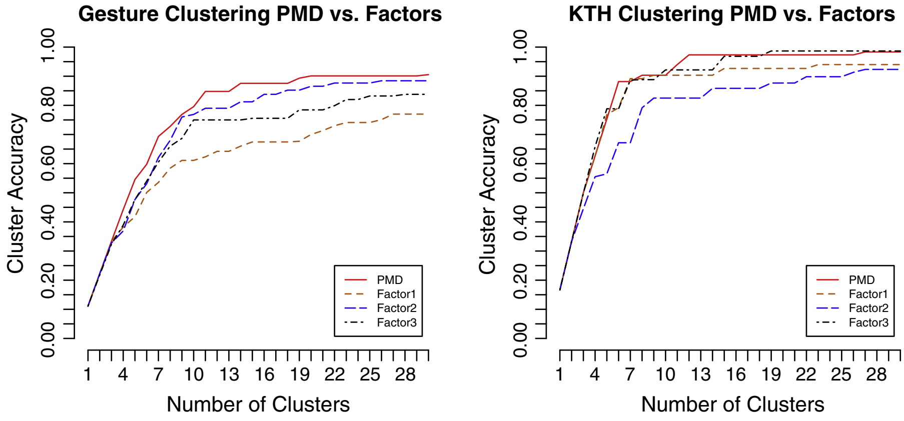


Fig. 4. The Product Manifold Distance performs better in terms of cluster accuracy than any single factor.

3. Part 1: unsupervised learning

In Part 1 of this article, we compare a mainstream Bag of Features approach to Lui's Product Manifold based method [5] in terms of how well they cluster data.

3.1. Method

At a high-level, our experimental method is to generate a pairwise distance matrix using both methods over three data sets relating to human expressions, gestures, and full-body actions. We apply a well-known hierarchical agglomerative clustering routine to the distance matrices to produce dendrograms of the similarity structure between the samples. The dendrogram can be cut at varying levels in the hierarchy to produce different numbers of clusters, from coarser to finer-grained grouping. We vary the number of clusters, *K*, over a range of values and observe how well the unsupervised grouping of the video clips compares to the desired labels. While we use labels to *evaluate* the clustering, the formation of the distance matrices and subsequent hierarchical clustering is entirely unsupervised. More details of each of these aspects can be found below.

Our intent with this study is to provide a comparison of the relative strengths and applicability of two popular approaches to unsupervised grouping of human behaviors. We selected Piotr Dollár's Bag of Features implementation [8], popularly known as the “Cuboids” algorithm, because the well-documented code is readily available upon request from the author, can be used to generate a number of feature descriptors, and generates competitive results. We used Lui's MATLAB implementation of the Product Manifold algorithm.

3.1.1. Data sets

We selected the following data sets for this study: Facial Expressions [8], Cambridge Gestures [23], and KTH Actions [7]. The samples in each data set are short video clips that exemplify a given expression, gesture, or action, respectively. Fig. 2 provides an illustration of each data set.

The Expressions data consists of 6 classes {anger, disgust, fear, joy, sadness, surprise}, repeated in 4 sets. The four sets are comprised of two subjects under two different lighting conditions performing 8 repetitions of all expressions, for a total of 192 videos. Each video clip starts with the subject in a neutral expression, then transitions into one of the expressions, and then back to neutral.

The Cambridge Gestures data consists of 9 classes, repeated in 5 sets of varying lighting, with 20 samples per class per set, for a total of 900 video clips. Each sample is a close-up of a single hand on a uniform background performing one gesture. The nine classes are divided into three shapes combined with three motions, as illustrated in Fig. 2.

The KTH Actions data consists of 6 classes {walking, jogging, running, boxing, handwaving, handclapping}, demonstrated by 25 subjects, each in 4 different scenes, for a total of 600 video clips. The first three scenes are taken outdoors, with a fairly uniform background. The fourth scene is taken indoors, also with a uniform background. Scene 2 varies the scale or angle from Scene 1. Scene 3 varies the clothing of the subject. Three of the classes involve a human gait, while the other three involve stationary actions. The subject varies direction of travel (for the gait classes), and is not always well-centered in the stationary actions.

All three data sets were designed to evaluate forced-choice classification algorithms. For the sake of familiarity within the action and gesture recognition community, we elected to use these same data sets, but in an evaluation scheme that measures unsupervised clustering and how the clusters align with different potential labelings

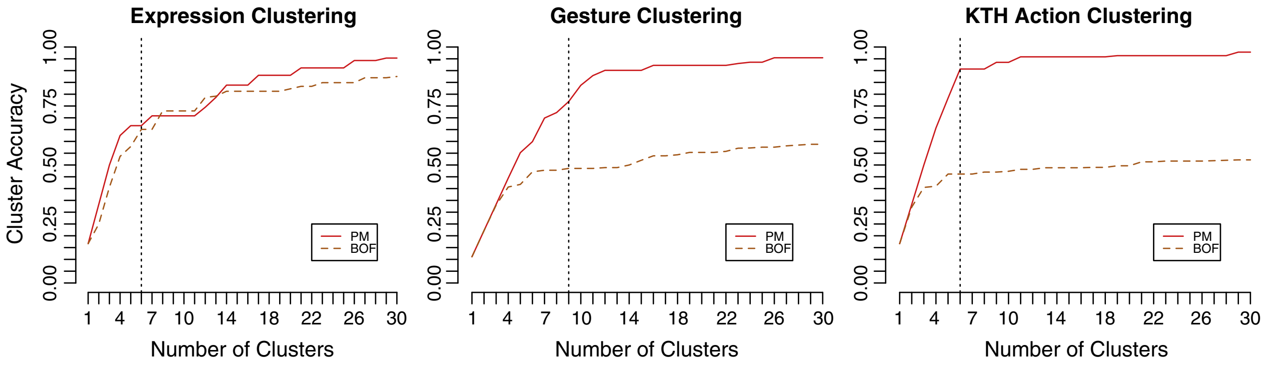


Fig. 5. PM vs BOF methods compared against the nominal class labels on all three data sets. Vertical line indicates number of nominal classes in the data set (6,9,6, respectively). PM generates much better clusters than BOF on the Gestures and KTH Actions data.

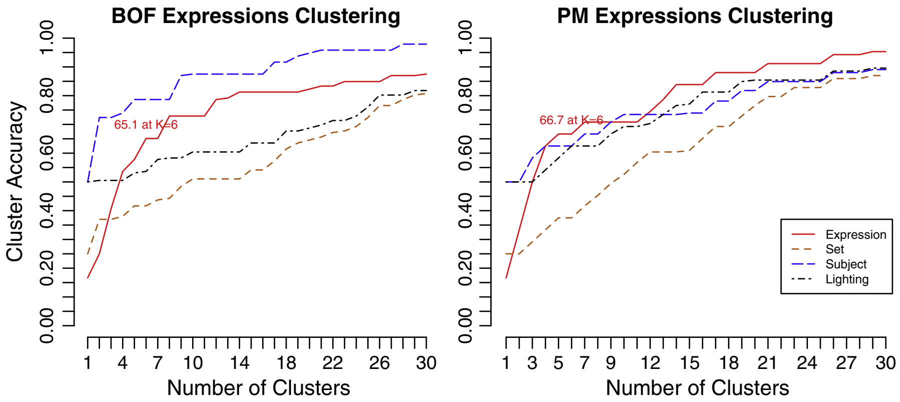


Fig. 6. Clustering of Expressions data. BOF clusters are more closely aligned to the subject labeling, but both methods perform similarly on other labels.

of the data. Video samples may be similar along different aspects than the externally applied class label, and our evaluation helps illustrate which of those aspects the algorithm is sensitive to.

3.1.2. Bag of Features

For the Expressions data, we used the code provided by Dollár, essentially unmodified, because it was developed in conjunction with this data set. Our minor changes were those required to use the Bag of Features representations to generate a distance matrix instead of as input to supervised classification. The code employs the Cuboids detector (separable linear filters, as described in [8]) coupled with the Cuboids descriptor, which is a flattened vector of gradients reduced via PCA to 100 dimensions.

For the Gestures and Actions data sets, we employ the Cuboids detector coupled with Histogram of Oriented Flow (HoF) features. We found this combination to generate the best performance in our tests, and it has been shown to generate good classification accuracy on KTH Actions, as demonstrated by Wang et al.'s evaluation of space-time features [15]. The HoF descriptor has 440 dimensions, which we employ with no dimensionality reduction. For the Cuboids detector, we set the spatial scale $\sigma = 2$ and the temporal scale $\tau = 3$ for Gestures and $\tau = 4$ for KTH Actions, which agree with the settings in Wang's evaluation.

We use a vocabulary of size 150 for all experiments, selected empirically among sizes ranging from 50 to 1000. The vocabulary was generated by K-Means over a random sample of 10% of all the features extracted from the data set. The Bag of Features representation was formed for each video and a pair-wise distance matrix generated using the χ^2 histogram distance function. Due to the randomness inherent in the vocabulary creation, we repeated the process 20 times and chose the vocabulary that generated the best results. For the remainder of this paper, this approach will be labeled "BOF."

3.1.3. Product Manifold

We used the code provided by Lui with no modifications beyond those required to generate pair-wise distance matrices on different data sets. Each video clip is rescaled to a $20 \times 20 \times 32$ tensor. Through the HOSVD, the tensors are projected onto the Product Manifold, and the pair-wise distances computed. For the remainder of this paper, this approach will be labeled "PM."

3.1.4. Cluster accuracy

We define *cluster accuracy* as the percentage of samples that were of the majority in their respective clusters. The minimum score always occurs when $K = 1$, in which case the cluster accuracy is the ratio of the number of samples in the largest class to the total number of samples in the data set, N. At the other extreme, when $K = N$, the cluster accuracy will be 1.0, as all samples will be assigned unique clusters and thus there will be no cluster "impurity." The computation is shown formally in Eq. (1), where C is the set of K clusters, x_i are the data points being clustered, and $|\cdot|$ indicates set cardinality.

$$
\begin{aligned}
C &= \left\{C^1, C^2, \ldots, C^K\right\} \\
X_L^K &= \left\{x_i | x_i \in C^K \wedge Label(x_i) = L\right\} \\
&\sum_{k=1}^{K} \max_{L \in Labels} \left|X_L^k\right| / \left|C^k\right|
\end{aligned}
\qquad (1)
$$

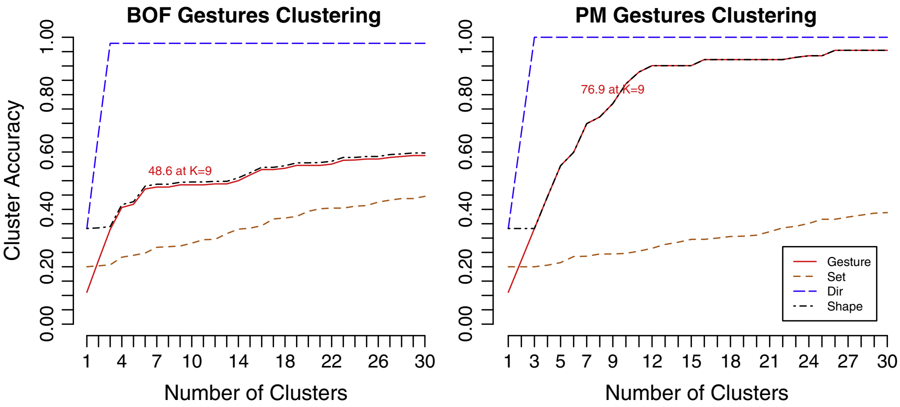


Fig. 7. Clustering of Gestures data. Direction of motion is nearly perfectly separated by both methods. Both methods show that Gesture class labeling is limited by the Shape.

3.1.5. Hierarchical clustering

We use an agglomerative hierarchical clustering method to group similar video clips. We tested several linkage methods and found that Ward's algorithm, which seeks to minimize the incremental increase in cluster variance, had superior clustering results. In addition to Ward's method, we evaluated Single linkage (nearest neighbor between cluster members), Complete linkage (furthest neighbor), Average linkage (average distance), and McQuitty's linkage (weighted average based on recursive agglomerations).

Fig. 3 shows a comparison of different linkage methods when employing the Bag of Features algorithm for measuring the similarity of Gestures. In this figure, we plot the cluster accuracy of the Gesture class label against the number of clusters, K, which was varied from 1 to 30. We perform a single hierarchical clustering per curve, and we vary K by selecting the cut in the hierarchy that yields the appropriate number of clusters. When K is unknown, the full curve may be more indicative than any single point in measuring performance. Regarding the selection of linkage method, Fig. 3 is representative of the results over all data sets and with both BOF and PM implementations — Ward's linkage is the best choice in all cases.

Hierarchical clustering is used because it produces deterministic results and it is easy to vary the number of clusters. In a completely unsupervised learning environment, the number of class labels is unknown, so the different levels of similarity/generalization provided by hierarchical clustering are appropriate.

3.1.6. PM Distance vs. Factors

One question that might arise regarding the PM Distance is whether it is any better than simply computing the subspace distance on a single unrolling of the tensor. To answer that question, we performed an experiment comparing the clustering performance of the Product Manifold distance to the performance of each factor considered separately. Recall that each factor manifold is a Grassmann manifold where the points are the subspaces spanned by a particular flattening of the video tensors (see Fig. 1).

For convenience, we label the Grassmann factor manifold arising from the (x,yt) flattening as “Factor 1”, and similarly, “Factor 2” from (y,xt), and “Factor 3” from (t,xy). In Fig. 4, we see that the PM Distance outperforms any single factor subspace distance in both KTH and Gestures data sets, which is an important validation for using all three unrollings in a single measure.

3.2. Results

We compared Bag of Features and Product Manifold methods for clustering facial expressions, hand gestures, and full-body actions. Each set of experiments is described below. A summary comparison of the relative performance of BOF and PM is illustrated in Fig. 5. This figure presents the performance curve when the generated clusters are compared against the nominal class labels provided by the data set. There are 6 classes in KTH Actions and Expressions, and 9 classes for Gestures, indicated by the vertical dotted black line. The solid red curve shows the cluster accuracy of PM over all K, the amber dotted curve shows BOF.

From this figure, PM strongly outperforms BOF on two of the three data sets, while Expressions yields comparable results. This may be in part because Dollár developed both the Expressions data set and the BOF implementation we adapted for this study, and thus the implementation may have a level of tuning for this data set not present in the others. However, we believe other factors are involved, which we present later.

A key result shown in Fig. 5 is the performance of PM on KTH Actions. At $K=6$, the cluster accuracy is 90.7%, suggesting that the KTH data set is intrinsically separable along the class labels using PM, and that one could *discover* the classes if they were not known a-priori. In [5], Lui reports nearest-neighbor classification results on KTH Actions using the Product Manifold representation, 96% using Schuldt's protocol ([7]) and 97% using leave-one-out. There are features-based approaches that have classification accuracies on KTH in the mid-to-upper 90's, but they employ strong supervised classifiers (for example [14] scores 94.5% using hierarchical features and multiple kernel learning). As with KTH Actions, the cluster accuracy on Gestures shows a big gap between PM and BOF.

Table 1
Gesture labels compared to 3 clusters. Gesture labels compared to X clusters.

	BOF cluster ID			PM cluster ID		
Label	1	2	3	1	2	3
F. Lft	99	0	1	100	0	0
S. Lft	100	0	0	100	0	0
V. Lft	100	0	0	100	0	0
F. Rgt	0	98	2	0	100	0
S. Rgt	1	99	0	0	100	0
V. Rgt	1	97	2	0	100	0
F. Cnt	0	4	96	0	0	100
S. Cnt	0	6	94	0	0	100
V. Cnt	0	2	98	0	0	100

In our work we desire representations for unsupervised learning, so the difference between the PM representation and the BOF representation, in terms of how much supervision is required to separate the data along semantically-meaningful partitions, is important. Our results suggest that BOF requires more supervision for high classification accuracy because clustering alone does not effectively separate the data. We explore these and other aspects in more detail, presented according to data set, below.

3.2.1. Expressions

We compared the clusters generated on the Expressions data to four labelings: the nominal Expression label from the data set (6 classes), the Set label (4 sets), and labels for Subject (2) and Lighting (2). Fig. 6 shows the results. Although the performance of the two methods is similar for Expression, Set, and Lighting labels, BOF clustering is much more closely aligned to subject identity than PM, as evidenced by the significantly higher curve.

With BOF, the Subject labeling generates less cluster impurity than Expressions. While the higher curve is indicative of the fact that there are only two subjects as opposed to six expressions, it is also true that with PM, the Subject labeling does not behave the same way. The subject identity is seemingly less useful to PM when grouping the

Table 2
Gesture labels compared to 9 clusters.

Label	1	2	3	4	5	6	7	8	9
Cluster ID — BOF representation									
F. Lft	95	4	0	0	0	0	0	1	0
S. Lft	46	54	0	0	0	0	0	0	0
V. Lft	26	74	0	0	0	0	0	0	0
F. Rgt	0	0	24	35	9	13	17	2	0
S. Rgt	0	1	30	25	20	6	18	0	0
V. Rgt	0	1	25	26	18	4	24	2	0
F. Cnt	0	0	1	2	0	0	1	92	4
S. Cnt	0	0	2	3	0	0	1	40	54
V. Cnt	0	0	1	0	1	0	0	52	46
Cluster ID — PM representation									
F. Lft	93	0	0	0	0	0	0	7	0
S. Lft	3	0	0	0	94	0	0	3	0
V. Lft	0	0	0	0	0	0	0	100	0
F. Rgt	0	43	57	0	0	0	0	0	0
S. Rgt	0	85	15	0	0	0	0	0	0
V. Rgt	0	62	38	0	0	0	0	0	0
F. Cnt	0	0	0	100	0	0	0	0	0
S. Cnt	0	0	0	0	0	80	20	0	0
V. Cnt	0	0	0	17	0	0	41	0	42

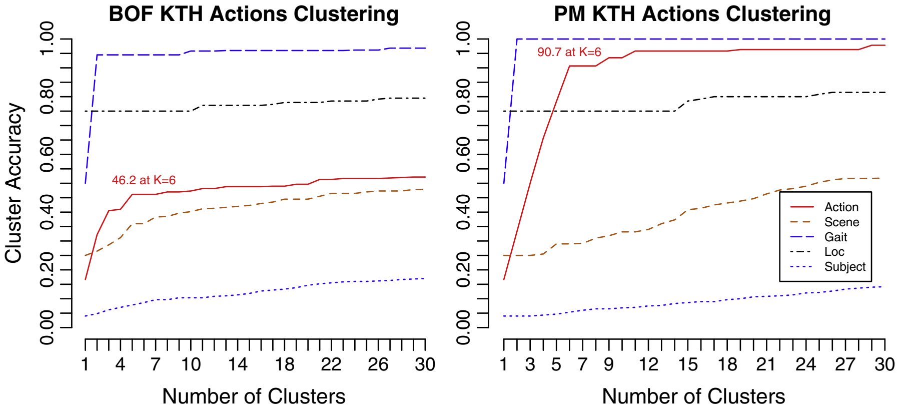


Fig. 8. Clustering of KTH Actions data. Both methods easily distinguish between Gait and Non-Gait actions. PM clustering performs excellently when judged against the nominal class labels.

expression video clips than it is with BOF. This leads to the speculation that if this small data set were expanded to include many more subjects, the sensitivity to subject identity evidenced by BOF may lead to decreased cluster accuracy when labeling by Expression, while PM performance might be less affected.

3.2.2. Gestures

We evaluated BOF and PM clustering against the following labels applied to the Cambridge Gestures data set: Gesture (the nominal class label, 9 classes), Set (5 sets with varying lighting conditions), Direction of motion (3 motions as per Fig. 2), and Shape (Flat, Spread, and V-Shape). Results are shown in Fig. 7.

One immediately obvious aspect of Fig. 7 is that both methods generate clusters that are nearly completely separable along direction of motion (98% accuracy at all K ranges for BOF and 100% for PM). At the same time, Gesture class labeling is nearly identical in performance to Shape labeling for both methods. We hypothesize that the hierarchical clustering groups the data first by motion direction, and later by shape. Further, because the overall performance of BOF is much lower than PM, it may be that PM is doing a much better job differentiating shape, while BOF struggles in this regard. This would not be surprising because BOF discards locations of features in the representation. As such, the histogram of space-time features located near the fingertips of the spread hand and flat hand may look very similar, and thus difficult to differentiate. The Product Manifold method, however, treats all pixels equally, preserving location information, and thus having less confusion between the hand shapes. To test this hypothesis, we further investigate the details of how clusters align to labels in the Gesture data set.

Given the strong affinity for both methods with the three gesture directions, we investigated the cluster accuracy when comparing the nominal class labels (Gesture) to clusters when $K=3$. The result in Table 1 shows that both methods nearly perfectly cluster along motion direction, as expected from Fig. 7.

When we raise K from 3 to 9, the number of nominal classes in the Gesture data set, we see that the two algorithms behave differently, as shown in Table 2. While PM begins to differentiate based on shape, BOF struggles to do so. BOF maintains its confusion between shapes within the same direction, while PM manages to cleanly separate Leftward motion into the three Shapes, and partially separate the Contraction motion as well. This evidence supports our hypothesis that shape is a secondary aspect of the clustering behind motion, and it proves to be the limiting factor on the overall agreement between the class labels and the clusters.

Restating an earlier point, with no supervision it is the inherent biases of the two methods that dictate which generates clusters that are better aligned with semantically-meaningful partitions. In this case, the bias of BOF to ignore relative spatio-temporal positions causes it to fail in many instances to match the nominal gesture label.

3.2.3. Actions

We chose the following labels to apply to the KTH Actions data set: Action (the nominal class label, 6 classes), Scene (4 scene types), Gait (2 types: gait or non-gait actions, as per Fig. 2), Location (2 types: indoors and outdoors, 75% are outdoors), and Subject (25 people). Results are shown in Fig. 8. We did not expect either method to align clusters against the Subject label, as the individuals can be hard to discern, and Scene 3 uses changes of clothing to further make identifying the subject difficult. Separating the actions based on Gait labeling proved easy for both methods. Although the performance curve for Location appears high, the base rate is 75% outdoors, and the results did not rise much above that minimum score. Clustering based on PM distances was very closely aligned to the nominal class labels, as shown by the 90.7% cluster accuracy at $K=6$.

Unlike with Gestures, we did not find a semantic labeling that best explains the performance of the nominal class labels. Given the high performance of PM clustering on the class labels, one is led to believe that the classes are inherently separable in most cases when using the PM representation, but not when using BOF.

Given that Support Vector Machines trained with similar BOF representations achieve classification accuracies in the upper 80's to lower 90's%, it is revealing that the clustering performance is comparatively poor on KTH Actions. Because of this, we believe that supervised training may be more important for achieving high accuracy with BOF representations of full-body actions than it is for PM representations.

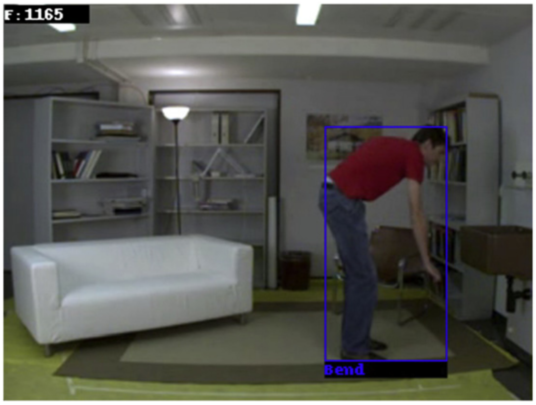


Fig. 9. Example of an activity detection from ETHZ Seq1.

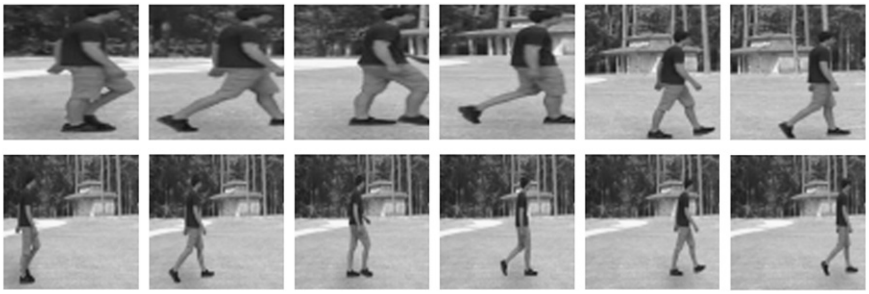


Fig. 10. Example tracklet created without (top) and with (bottom) stabilization strategy. Top tracklet uses the bounding rectangles from the track to clip tiles from the source. Bottom uses the full spatial extent of the track within the temporal window to define a single clipping region, and thus stabilizes the images and corrects for minor track drift. In both cases, the clipped tiles are rescaled to fit the fixed tracklet dimensions.

4. Part 2: streaming video

The results of Part 1 suggest that the Product Manifold representation may be better than Bag of Features for unsupervised action recognition. We now demonstrate an approach to applying the PM Distance to action detection in continuous video streams.

4.1. Method

We propose an unsupervised learning method for micro-action recognition based on clustering short duration video clips, called tracklets, that are extracted from entity tracks in longer duration videos. Each tracklet captures the appearance and motion of an entity for a second or two of time. We cluster the tracklets using the Product Manifold distance. In grouping similar tracklets, we find the repeated micro-actions in the video. We perform clustering with no foreknowledge of either the expected types or numbers of micro-actions present in the data. The idea is to discover the micro-actions, not to force-choice classify the activities into predefined classes.

The set of clusters may be given labels by the users of the system, a process we call “Selective Guidance.” It is important to note that the system would work with internally generated identifiers, although it would be challenging for the user to know the significance of a generic output like “action12” instead of “walking.” From each labeled cluster, we identify a small number of exemplar tracklets that best represent the group. Not all clusters are easily described with a concise label. For those that are easily described, we can apply that label to the cluster's exemplar(s). For those clusters that are semantically meaningless, we apply no label and extract no exemplar for run-time matching.

The set of exemplars is used in a nearest-neighbor matching strategy to detect and label micro-actions on previously unseen test video. We perform detection on streaming video without pre-segmenting the space-time regions of interest. As an entity being tracked changes behavior, the system will detect the change and apply a new label when appropriate.

At times, a tracklet from the test video may not be a good match to any of the exemplars. In such instances, the system will apply no label to the tracklet, and it will be remembered as a novel detection. The set of novel detections can be evaluated to produce additional exemplars, and thus the system can learn over time, boot-strapped from an initial training set. Further details on the various aspects of our approach are presented below.

4.1.1. Data

We use the publicly-available ETHZ Living Room data set for our evaluation [17]. We selected this data set because it represents the continuous surveillance problem better than many of the more popular action recognition benchmarks. Many action recognition data sets are designed to support forced-choice classification of segmented video clips. The ETHZ Living Room data, however, provides three video sequences. The first, over 7000 frames long, is a continuous recording of a person moving about a room and performing a few selected behaviors (walking, sitting, bending down). The first video (Seq1) is intended to allow an unsupervised system to learn the nominal behavior of the room's occupant. The second two videos (Seq2 and Seq3) are shorter, and are used to present novel behaviors, such as falling down or panicked gesticulations, to measure a system's ability to detect anomalous events. Fig. 9 shows a sample image from the first video of the data set. For brevity, in the remainder of this paper we refer to this data set as ETHZ.

4.1.2. Tracks and tracklets

To generate the tracks on ETHZ video sequences, we perform background subtraction using the median image of the first 2000 frames as the background model. We use the bounding box of the foreground mask to track the subject in the video. Processing is performed using grayscale imagery.

Action recognition approaches that rely on silhouette extraction [16,17] can be negatively impacted when the foreground mask is inaccurate. An important advantage to our method is that it processes all pixels within the bounding box, requiring no silhouette mask, and is therefore less sensitive to foreground/background segmentation challenges.

We define a tracklet to be a short contiguous section of a track that has been reshaped into a fixed-size data cube of dimension: (x,y,t), where the unit of time, t, is the frame number. The tracklet duration is chosen to be appropriate for capturing the motion of micro-actions, and thus is typically less than a few seconds long. A single track of a person over time will give rise to numerous tracklets, some of which may clearly contain a micro-action, and others may represent transitions between micro-actions and thus have no clear semantic label (see Fig. 12 for an example). The size of each frame in the tracklet is kept small in order to capture only large-scale

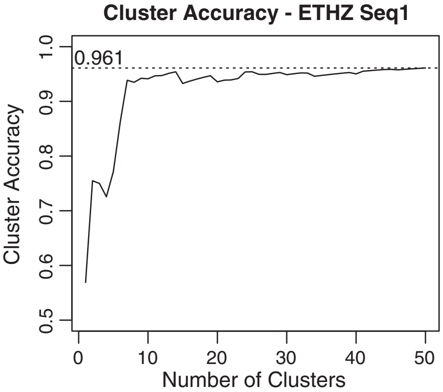


Fig. 11. Cluster Accuracy on ETHZ Seq1 tracklets. The sharp rise followed by a long plateau indicates many K values work well.

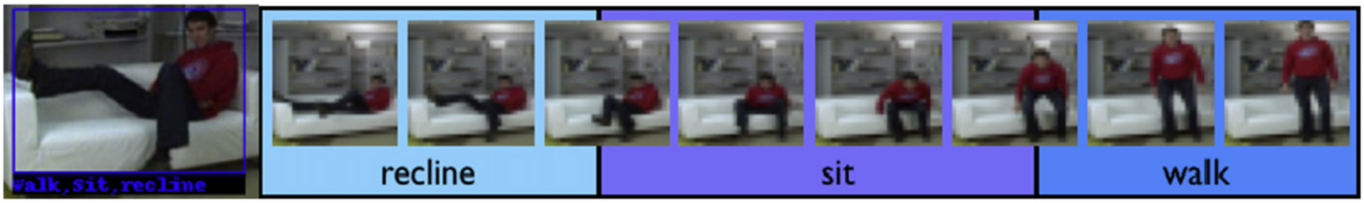


Fig. 12. Example of a tracklet labeled from multiple exemplars. This tracklet captures the transition between multiple states, and is thus correctly described by the unordered label set {walk, site, recline}. Sample frames on right are from the 48 frame, 32 × 32 pixel tracklet.

structure, eliminate high-frequency features, and de-emphasize individual appearance. In this investigation, we create tracklets of size (32 × 32 × 48).

We employ a sliding temporal window strategy, overlapping by 16 frames, for slicing tracks into tracklets. The bounding box of a track typically varies from frame-to-frame, and so the resulting tracklet can suffer from significant instability that negatively impacts the PM distance computation. To stabilize the tracks, we compute the bounding box that contains the spatial extent of the *entire* tracklet, and we use that box to clip tracklet tiles from corresponding frames in the video. The benefit of this simple stabilization strategy is illustrated in Fig. 10.

4.1.3. Clustering and exemplar selection

Given a set of tracklets extracted from the training video, we compute the pair-wise PM distances to form a distance matrix. As in Part 1, we use agglomerative hierarchical clustering with Ward's linkage to generate a cluster tree. The tree can be cut at a particular linkage threshold value to generate a set of clusters. With no prior knowledge of the expected number of clusters, it can be challenging to select the appropriate cut. It is an open question on how best to measure the clustering quality lacking any prior information. However, we have observed that the performance of our method rises quickly as K is increased, and then plateaus at a high level for K greater than approximately ten percent of the training sample size.

To convince ourselves that this is true, we measured the Cluster Accuracy against the choice of K, illustrated in Fig. 11. Generating this plot requires labeling the training data, but this is not an integral part of our method. Instead, for the experiments described later, we blindly chose values of K equal to 5, 10, 15, and 20% of the training set size.

The PM distance measure is non-Euclidean, and there is no closed-form computation for the mean value of the samples in a cluster. Instead, exemplars (medoids) can be selected from within each cluster that minimizes the sum of the distances to the other cluster members. More than one exemplar can be selected from within a cluster by removing the best medoid and repeating the process. Interestingly, we found that pulling two exemplars from clusters that represent "sitting" or "bending" resulted in one of the samples exhibiting the downward aspect of the motion and the other exemplar exhibiting the upward aspect (e.g., bending down vs. straightening back up.)

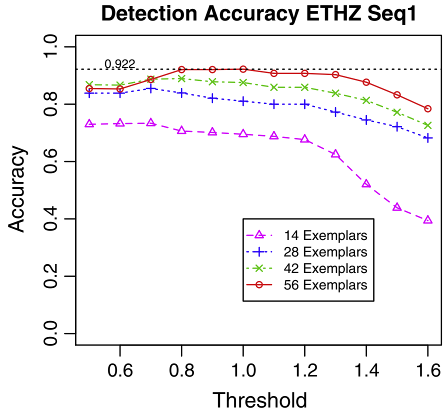


Fig. 13. Detection accuracy using different exemplar sets on Seq1. Accuracy is the average F1 score between predicted and ground truth label bit vectors. Threshold is the minimum sum of the weighted bits from the 3 nearest exemplars required to activate the corresponding label bit on the tracklet.

4.1.4. Detection

After exemplars have been trained, we use them to match against tracklets in the test videos. This process occurs in near-real-time on the streaming video. We compute the K-Nearest-Neighbors using the PM distance between each new tracklet and the exemplars. For the experiments reported herein, unless otherwise mentioned, we use 3 neighbors. Soft weighting is used so that selected exemplars contribute their labels to the new tracklet based on how close they are. A standard Gaussian decay is used with σ determined from the distribution of distances in the training samples.

We allow for multiple labels. Each tracklet maintains a bit vector of length equal to the cardinality of the label set. In the bit-vector, a 1 indicates the corresponding label applies to the tracklet, and a 0 means it does not. The weighted label vectors from the nearest exemplars are summed component-wise to produce the raw label vector of the new tracklet. A score threshold is applied to each component to generate the label bit vector. It is possible, even desirable, that the label vector will result in all zeros should none of the nearest exemplars be close enough to the sample.

Formally, the scoring computation is shown in Eq. (2), where ω_i is the weight based on the PM distance $d(i,x)$ between exemplar i and tracklet x, L_i is the label vector for exemplar i, P is the number of labels, s^p is the component score computed as the weighted sum of the corresponding components from the K nearest exemplars, and L_x is the computed label for tracklet x by comparing the component scores to a constant threshold t.

$$\begin{aligned} \omega_i &= e^{-d(i,x)^2/2\sigma^2} \\ L_i &= \left(l_i^1, l_i^2, \ldots, l_i^P\right) \\ s^p &= \sum_{i=1}^{K} \omega_i l_i^p \quad, \forall p \in \{1 \ldots P\} \\ L_x &= \left(s^1 \geq t, s^2 \geq t, \ldots, s^p \geq t\right) \end{aligned} \tag{2}$$

4.1.5. Anomalies and incremental learning

An anomaly is a tracklet that is too far from the exemplars to produce a non-zero label set. After the initial exemplars have been produced from the training data, we can run the system with a relatively high score threshold in order to generate a set of anomalous samples. We combine the anomalous tracklets with the current exemplar set, and then recompute the clustering over only the combined set (i.e. omitting all of the original training tracklets), yet keeping the K-value the same. In the resulting clusters, we look for any of the anomalous samples that are not grouped in the same clusters with current exemplars. This subset of the anomalous samples is selected to be added to the updated exemplar set, and selective guidance is used to generate labels where appropriate, or to assign the new exemplar to an existing label if it represents a novel aspect of a known micro-action.

4.2. Results

4.2.1. Experiment 1

The first experiment demonstrates the variation in performance when selecting different values for *K*, the number of clusters, and various score threshold values, as described previously. We chose four values of *K* to use in our initial clustering, where we blindly selected a number of clusters equal to 5%, 10%, 15%, and 20% of the number of training tracklets. Having extracted 283 tracklets from a sampling of video Seq1 for training, the values for *K* were 14, 28, 42, and 56, respectively. We selected one exemplar per cluster.

Fig. 13 shows the results. The accuracy is measured in terms of the average *F1* score between the predicted and ground truth label bit vectors. The *F1* score is the harmonic mean between precision and recall, which we apply to the multi-label bit-vectors to measure both the false positives (incorrect assignment of a 1) and false negatives (incorrect assignment of a 0). It is not surprising that having more exemplars leads to better overall performance, yet the performance drop when decreasing from 56 to 42 exemplars is not severe. When using the best score threshold of 0.8, the performance drops by 3% from 56 to 42, and 8% from 56 to 28. This adds support to our belief that performance is not sensitive to the choice of *K*, as long as *K* is beyond the steep rising curve, as described earlier (see Fig. 11).

Fig. 12 shows an example of a single tracklet that was given multiple labels. The detection was on a tracklet from Seq2 where the tracklet duration happened to contain the transition between three micro-actions. The advantage of allowing multiple labels is that such interstitial observations may be described as a set of appropriate labels. There are no exemplars that were learned that had more than two labels. This result required the contribution from two or more exemplars that, while different from each other, all had a similarity to the novel tracklet, as measured by the PM distance.

4.2.2. Experiment 2

The second experiment was performed to gauge how well the system can incrementally learn based on anomalous detections. We selected an exemplar set trained from Seq1, and used it to detect anomalous micro-actions from Seq2, which contains never-seen behaviors including falling down, jumping, reclining on the couch, and panicking. Fig. 14 shows a set of fourteen new exemplars identified from Seq2 using the procedure described in the Methods section.

After folding in the new exemplars with the original set, we performed detection on Seq3. We repeated this procedure, but reversed the roles of Seq2 and Seq3. The results are shown in Fig. 15. There is nearly a 10% performance improvement after incorporating the new exemplars.

5. Conclusion

5.1. Discussion

Lacking any supervision, and outside a forced-choice paradigm, it is important to design representations that are amenable to clustering human activities along semantically meaningful aspects. In Part 1 of this paper, we presented performance differences between Product Manifold and Bag of Features representations over three data sets representing, respectively, human expressions, hand gestures, and full-body actions. The pair-wise distance matrices generated by Product Manifold representations of the video clips led to superior clustering accuracy when compared with the nominal class labels of each data set.

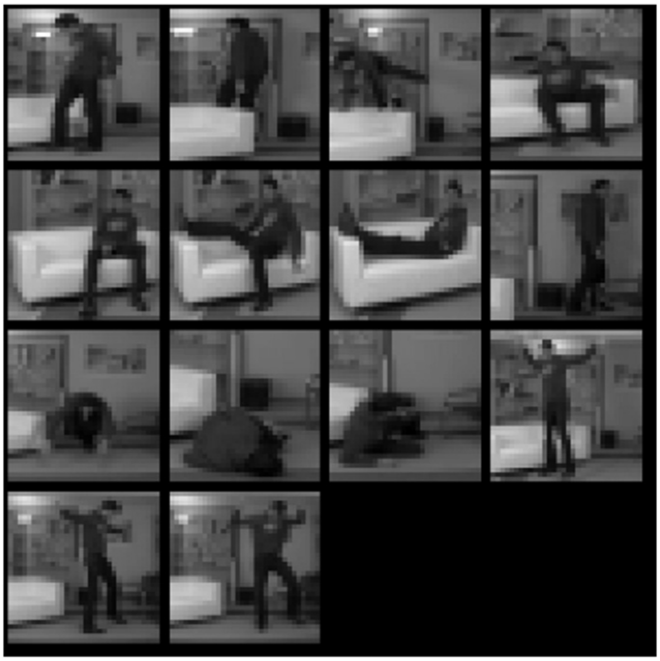


Fig. 14. Fourteen new exemplars were learned from the ETHZ Seq2 video, representing the novel micro-actions of jumping, reclining, falling, and panicking. Images are representative frames from the 48 frame, 32 × 32 pixel tracklets.

We also found that while gross motions were easily clustered by both methods, the lack of preservation of structural information inherent to the BOF representation leads to limitations that are not easily overcome without supervised training. This was evidenced by the poor separation of shapes in the hand gestures data by BOF, and the overall poor performance on full-body actions. There are BOF-based action recognition approaches that add additional spatial information to the representation, such as by using pyramid structures [14]. While it may be likely that clustering performance would improve with the added spatial information, the cost is in increased design and computational complexity. That said, we encourage other researchers to follow the protocol we presented in this article to facilitate comparative evaluations for unsupervised action learning.

We believe that to make progress on the fundamental challenge of human behavior recognition in continuous video, more research is required on open-world, incremental learning methods that require a minimum of supervision. In Part 2 of this paper, we presented a

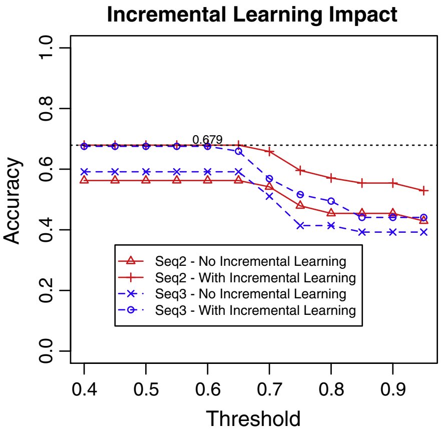


Fig. 15. Detection accuracy with and without incremental learning. With incremental learning, we detected anomalous micro-actions from Seq2, using them to update the base set of 56 exemplars for detection on Seq3 (and vice-versa for testing on Seq2 with new exemplars learned from Seq3).

step in this direction by showing how the Product Manifold representation for measuring similarity between video tensors can be applied to unsupervised, incremental learning of micro-actions.

In addition to those described earlier, our approach has the additional advantage of not requiring a large number of parameters and design choices. This is a clear improvement over many approaches that require parameter selection and design optimizations for feature detection, feature extraction, dimensionality reduction, codebook size, and so on. As an unsupervised method, we require no extensive training and validation stages. Offline training time required for computing the distance matrix is modest, because the PM distance computation between a pair of tracklets is fast (10's of milliseconds using unoptimized MATLAB code).

5.2. Future work

It is an open question on how best to select an appropriate number of exemplars (or clusters) without having prior knowledge of an expected number of behaviors to be observed. We presented a rudimentary method for incrementally updating the set of exemplars, yet more sophisticated methods may be required. Future work includes investigation of cluster quality, incremental learning strategies, and the application of micro-action detections to the recognition of longer term events and multi-entity interactions.

The clustering method may need to be adapted to an online method with outlier rejection for learning salient behaviors in continuous data streams, while avoiding clustering noise. By combining online unsupervised learning methods with Product Manifold distance measures, we hope to make significant advances towards our larger goal of developing behavior recognition capabilities in less controlled, non forced-choice scenarios operating on continuous data streams. This capability is what is ultimately required for the “persistent stare” needs of the video surveillance community and for advancing human–robot interactions.

Acknowledgments

We would like to thank Piotr Dollár for the Bag-of-Features code he has made available.

References

[1] S. O'Hara, Y.M. Lui, B.A. Draper, Unsupervised learning of human expressions, gestures, and actions, Proceedings of the IEEE Conference on Automatic Face and Gesture Recognition, 2011.

[2] S. O'Hara, B.A. Draper, Unsupervised learning of micro-action exemplars using a product manifold, Proceedings of the IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS), Klagenfurt, Austria, 2011, p. 6.

[3] D.H. Wolpert, The supervised learning no-free-lunch theorems, Proceedings of the Online World Conference on Soft Computing in Industrial Applications, 2001, pp. 25–42.

[4] M.S. Ryoo, C.C. Chen, J.K. Aggarwal, A. Roy-Chowdhury, An overview of contest on Semantic Description of Human Activities (SDHA) 2010, Proceedings of the International Conference on Pattern Recognition (ICPR), 2010.

[5] Y.M. Lui, J.R. Beveridge, M. Kirby, Action classification on Product Manifolds, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010.

[6] R. Poppe, A survey on vision-based human action recognition, Image Vis. Comput. 28 (6) (2010) 976–990.

[7] C. Schuldt, I. Laptev, B. Caputo, Recognizing human actions: a local SVM approach, Proceedings of the International Conference on Pattern Recognition (ICPR), 2004.

[8] P. Dollar, V. Rabaud, G. Cottrell, S. Belongie, Behavior recognition via sparse spatio-temporal features, Proceedings of the Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (PETS), 2005.

[9] Y. Ke, R. Sukthankar, M. Hebert, Efficient visual event detection using volumetric features, Proceedings of the International Conference on Computer Vision (ICCV), 2005.

[10] I. Laptev, On space-time interest points, Int. J. Comput. Vis. 64 (2) (2005) 107–123.

[11] L. Gorelick, M. Blank, E. Shechtman, M. Irani, R. Basri, Actions as space-time shapes, IEEE Trans. Pattern Anal. Mach. Intell. 29 (12) (2007) 2247–2253.

[12] P. Scovanner, S. Ali, M. Shah, A 3-dimensional SIFT descriptor and its application to action recognition, Proceedings of the 15th International Conference on Multimedia, ACM, 2007.

[13] K. Rapantzikos, Y. Avrithis, S. Kollias, Dense saliency-based spatiotemporal feature points for action recognition, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009.

[14] A. Kovashka, K. Grauman, Learning a hierarchy of discriminative space-time neighborhood features for human action recognition, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010.

[15] H. Wang, M.M. Ullah, A. Kläser, I. Laptev, C. Schmid, Evaluation of local spatio-temporal features for action recognition, Proceedings of the British Machine Vision Conference (BMVC), 2009.

[16] Z. Lin, Z. Jiang, L.S. Davis, Recognizing actions by shape-motion prototype trees, Proceedings of the International Conference on Computer Vision (ICCV), 2009.

[17] F. Nater, H. Grabner, L. Van Gool, Exploiting simple hierarchies for unsupervised human behavior analysis, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010.

[18] K. Guo, P. Ishwar, J. Konrad, Action recognition in video by sparse representation on covariance manifolds of silhouette tunnels, Proceedings of the ICPR Contest on Semantic Description of Human Activities (SDHA Challenge), Istanbul, Turkey, 2010, pp. 294–305.

[19] A. Elgammal, C.S. Lee, Tracking people on a torus, IEEE Trans. Pattern Anal. Mach. Intell. (2008) 520–538.

[20] Y. M. Lui, Advances in matrix manifolds for computer vision, Image and Vision Computing In Press (0) (2011) –, ISSN 0262–8856, doi:10.1016/j.imavis.2011.08.002, URL http://www.sciencedirect.com/science/article/pii/S0262885611000692.

[21] J.C. Niebles, H. Wang, L. Fei-Fei, Unsupervised learning of human action categories using spatial–temporal words, Int. J. Comput. Vis. 79 (3) (2008) 299–318.

[22] A. Gilbert, R. Bowden, Push and pull: iterative grouping of media, Proceedings of the British Machine Vision Conference (BMVC), 2011.

[23] T.K. Kim, S.F. Wong, R. Cipolla, Tensor canonical correlation analysis for action classification, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007.