Hindawi Publishing Corporation
EURASIP Journal on Image and Video Processing
Volume 2009, Article ID 738702, 13 pages
doi:10.1155/2009/738702

Research Article

Viewpoint Manifolds for Action Recognition

Richard Souvenir and Kyle Parrigan

Department of Computer Science, University of North Carolina at Charlotte, 9201 University City Boulevard, Charlotte, NC 28223, USA

Correspondence should be addressed to Richard Souvenir, souvenir@uncc.edu

Received 1 February 2009; Accepted 30 June 2009

Recommended by Yoichi Sato

Action recognition from video is a problem that has many important applications to human motion analysis. In real-world settings, the viewpoint of the camera cannot always be fixed relative to the subject, so view-invariant action recognition methods are needed. Previous view-invariant methods use multiple cameras in both the training and testing phases of action recognition or require storing many examples of a single action from multiple viewpoints. In this paper, we present a framework for learning a compact representation of primitive actions (e.g., walk, punch, kick, sit) that can be used for video obtained from a single camera for simultaneous action recognition and viewpoint estimation. Using our method, which models the low-dimensional structure of these actions relative to viewpoint, we show recognition rates on a publicly available dataset previously only achieved using multiple simultaneous views.

1. Introduction

Video-based human motion analysis and action recognition currently lags far behind the quality achieved using marker-based methods, which have shown to be very effective for obtaining accurate body models and pose estimates. However, marker-based studies can only be conducted reliably in a laboratory environment and, therefore, preclude in situ analysis. Comparable results derived from video would be useful in a multitude of practical applications. For instance, in the areas of athletics and physiotherapy, it is often necessary to recognize and accurately measure the actions of a human subject. Video-based solutions hold the promise for action recognition in more natural environments, for example, an athlete during a match or a patient at home.

Until recently, most of the research on action recognition focused on actions from a fixed, or canonical, viewpoint [1–4]. The general approach of these view-dependent methods relies on (1) a training phase, in which a model of an action primitive (a simple motion such as step, punch, or sit) is constructed, and (2) a testing phase, in which the constructed model is used to search the space-time volume of a video to find an instance (or close match) of the action. Because a robust human motion analysis system cannot rely on a subject performing an action in only a single, fixed view relative to the camera, viewpoint-invariant methods have been developed which use multiple cameras in both the training and testing phases of action recognition [5, 6]. These methods address the problem of view-dependence of the single camera systems, but generally require a multicamera laboratory setting similar in complexity to and equally as restrictive as marker-based solutions.

In this paper, we present a framework for learning a view-invariant representation of primitive actions (e.g., walk, punch, kick) that can be used for video obtained from a single camera, such as any one of the views in Figure 1. Each image in Figure 1 shows a keyframe from video of an actor performing an action captured from multiple viewpoints. In our framework, we model how the appearance of an action varies over multiple viewpoints by using manifold learning to discover the low-dimensional representation of action primitives. This new compact representation allows us to perform action recognition on single-view input, the type that can be easily recorded and collected outside of laboratory environments.

Section 2 opens with a review of related work in both view-dependent and view-invariant action recognition. In Section 3, we describe the two motion descriptors that we

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

Figure 1: These images show four keyframes from various viewpoints at the same time-point of an actor checking her watch. In this paper, we develop a framework to learn functions over classes of action for recognition from a continuous set of viewpoints.

will test in our framework; one is a well-known descriptor that we modify for our purposes and the second we developed for use in this framework. We continue in Section 4 by describing how we learn a low-dimensional representation of these descriptors. In Section 5 we put everything together to obtain a compact view-invariant action descriptor. In Section 6, we demonstrate that the viewpoint manifold representation provides a compact representation of actions across viewpoints and can be used for discriminative classification tasks. Finally, we conclude in Section 7 with some closing remarks.

2. Related Work

The literature on human motion analysis and action recognition is vast. A recent survey [7] provides a taxonomy of many techniques. In this section, we focus on a few existing methods which are most related to the work presented in this paper.

Early research on action recognition relied on single, fixed camera approaches. One of the most well-known approaches is temporal templates [1] which model actions as images that encode the spatial and temporal extent of visual flow in a scene. In Section 3 we describe temporal templates in more detail as we will apply this descriptor to our framework. Other view-dependent methods include extending 2D image correlation to 3D for space-time blocks [2]. In addition to developing novel motion descriptors, other recent work has focused on the additional difficulties in matching image-based time-series data, such as the intraclass variability in the duration of different people performing the same action [3, 4] or robust segmentation [8].

Over time, researchers have begun to focus on using multiple cameras to support view-invariant action recognition. One method extends temporal templates by constructing a 3D representation, known as a motion history volume [5]. This extension calculates the spatial and temporal extent of the visual hull, rather than the silhouette, of an action. In [6] the authors exploit properties of the epipolar geometry of a pair of independently moving cameras focused on a similar target to achieve view-invariance from a scene. In these view-invariant methods for action recognition, the models implicitly integrate over the viewpoint parameter by constructing 3D models.

In [9], the authors rely on the compression possible due to the similarity of various actions at particular poses to maintain a compact ($|actions| * |viewpoints|$) representation for single-view recognition. In [10], the authors use a set of linear basis functions to encode for the change in position of a set of feature points of an actor performing a set of actions. Our framework is most related to this approach. However, instead of learning an arbitrary set of linear basis functions, we model the change in appearance of an action due to viewpoint as a low-dimensional manifold parameterized by the primary transformation, in this case, viewpoint of the camera relative to the actor.

3. Describing Motion

In this paper, the goal is to model the appearance of an action from a single camera as a function of the viewpoint of the camera. There exist a number of motion descriptors, which are the fundamental component of all action recognition systems. In this paper, we will apply our framework to two action descriptors: the well-known motion history images (MHIs) of temporal templates [1] and our descriptor, the $\mathcal{R}$ transform surface ($\mathcal{R}$XS) [11], which extends a recently developed shape descriptor, the $\mathcal{R}$ transform [12], into a motion descriptor. In this section, we review the MHI motion descriptor and introduce the $\mathcal{R}$XS motion descriptor.

3.1. Motion History Images. Motion history images encode motion information in video using a human-readable representation whose values describe the history of motion at each pixel location in the field of view. The output descriptor is a false image of the same size in the x- and y- dimensions as frames from the input video. To create an MHI, H, using the video as input, construct a binary valued function $D(x, y, t)$, where $D(x, y, t) = 1$ if motion occurred at pixel location (x, y) in frame t. Then, the MHI is defined as

$$H_\tau(x, y, t) = \begin{cases} \tau & \text{if } D(x, y, t) = 1, \\ \max(0, H_\tau(x, y, t-1) - 1) & \text{otherwise,} \end{cases} \quad (1)$$

where τ is the duration of the motion or the length of the video clip if it has been preprocessed to contain a single

action. Intuitively, for pixel locations (x, y), $H_\tau(x, y, t)$ is the maximum value τ for motion occurring at time t and if there is no motion at (x, y), the previous intensity, $H_\tau(x, y, t-1)$ is at the pixel location which is carried over in a linearly decreasing fashion. Figure 2 shows two examples of the MHI constructed from an input video. Each row of the figure shows four keyframes from a video clip in which actors are performing an action (punching and kicking, resp.) and the associated motion history image representation. In our implementation, we replace the binary valued function, D, with the silhouette occupancy function, as described in [5]. The net effect of this change is that style of the actor (body shape, size, etc.) is encoded in the MHI, in addition to the motion. One advantage of the MHI is the human-readability of the descriptor.

3.2. $\mathcal{R}$ Transform Surface Motion Descriptor. In addition to testing our approach using an existing motion descriptor, we also develop the $\mathcal{R}$XS motion descriptor. The $\mathcal{R}$XS is based on the $\mathcal{R}$ transform which was developed as a shape descriptor to be used in object classification from images. Compared to competing representations, the $\mathcal{R}$ transform is computationally efficient and robust to many common image transformations. Here, we describe the $\mathcal{R}$ transform and our extension into a surface representation for use in action recognition.

The $\mathcal{R}$ transform converts a silhouette image to a compact 1D signal through the use of the two-dimensional Radon transform [13]. In image processing, the Radon transform is commonly used to find lines in images and for medical image reconstruction. For an image $I(x, y)$, the Radon transform, $g(\rho, \theta)$, using polar coordinates (ρ, θ), is defined as

$$g(\rho, \theta) = \sum_x \sum_y I(x, y)\, \delta(x \cos\theta + y \sin\theta - \rho), \tag{2}$$

where δ is the Dirac delta function which outputs 1 if the input is 0 and 0 otherwise. Intuitively, $g(\rho, \theta)$ is the line integral through image I of the line with parameters (ρ, θ).

The $\mathcal{R}$ transform extends the Radon transform by calculating the sum of the squared Radon transform values for all of the lines of the same angle, θ, in an image:

$$\mathcal{R}(\theta) = \sum_\rho g^2(\rho, \theta). \tag{3}$$

Figure 3 shows three examples of an image, the derived silhouette showing the segmentation between the actor and the background, the Radon transform of the silhouette and the $\mathcal{R}$ transform.

The $\mathcal{R}$ transform has several properties that make it particularly useful for representing image silhouettes and extensible into a motion descriptor. First, the transform is translation-invariant. Translations of the silhouette do not affect the value of the $\mathcal{R}$ transform, which allows us to match images of actors performing the same action regardless of their position in the image frame. Second, the $\mathcal{R}$ transform has been shown to be robust to noisy silhouettes (e.g., holes, disjoint silhouettes). This invariance to imperfect silhouettes is useful to our method in that extremely accurate segmentation of the actor from the background is not necessary, which can be difficult in certain environments. Third, when normalized, the $\mathcal{R}$ transform is scale-invariant. Scaling the silhouette image results in an amplitude scaling of the $\mathcal{R}$ transform, so for our work, we use the normalized transform:

$$\mathcal{R}'(\theta) = \frac{\mathcal{R}(\theta)}{\max_{\theta'}(\mathcal{R}(\theta'))}. \tag{4}$$

The $\mathcal{R}$ transform is not rotation-invariant. A rotation in the silhouette results in a phase shift in the $\mathcal{R}$ transform signal. For human action recognition, this is generally not an issue, as this effect would only be achieved by a camera rotation about its optical axis which is quite rare for natural video.

In previous work using the $\mathcal{R}$ transform for action recognition [14], the authors trained Hidden Markov Models to learn which sets of unordered $\mathcal{R}$ transforms corresponded to which action. In this paper, we extend the $\mathcal{R}$ transform to include the natural temporal component of actions. This generalizes the $\mathcal{R}$ transform curve to the $\mathcal{R}$ transform surface ($\mathcal{R}$XS), our representation of actions. We define this surface for a video of silhouette images $I(x, y, t)$ as:

$$S(\theta, t) = \mathcal{R}'_t(\theta), \tag{5}$$

where $\mathcal{R}'_t(\theta)$ is the normalized $\mathcal{R}$ transform for frame t in I. Each row of Figure 4 shows four silhouette keyframes, the associated $\mathcal{R}$ transform curves, and the $\mathcal{R}$ transform surface motion descriptor generated for the video. Each video contains roughly 70 frames, but we scaled the time axis from 0 to 1 so that our descriptor is invariant to the frame rate of the video and robust to the duration of an action.

The first row of Figure 4 depicts the visually-intuitive surface representation for the "sit down" action. The actor begins in the standing position, and his silhouette approximates a vertically-elongated rectangle. This results in relatively higher values for the vertical line scans (θ near 0 and π). As the action continues, and the actor takes the seated position, the silhouette approximates a circle. This results in roughly equal values for all of the line scans in the $\mathcal{R}$ transform and a flatter representation in the surface. Other motions, such as punching and kicking, have less dramatic, but similarly intuitive $\mathcal{R}$ transform surface representations. Figure 5 summarizes the process for creating a $\mathcal{R}$ transform surface motion descriptor from video.

In the following section, we describe our approach to view-invariant action recognition, which relies on applying manifold learning techniques to this particular action descriptor.

4. Viewpoint Manifold

Our goal is to provide a compact representation for view-invariant action recognition. Our approach is to learn a model which is a function of viewpoint. In this section, we describe methods for automatically learning a low-dimensional representation for high-dimensional data (e.g., $\mathcal{R}$ transform surfaces and motion history images), which lie

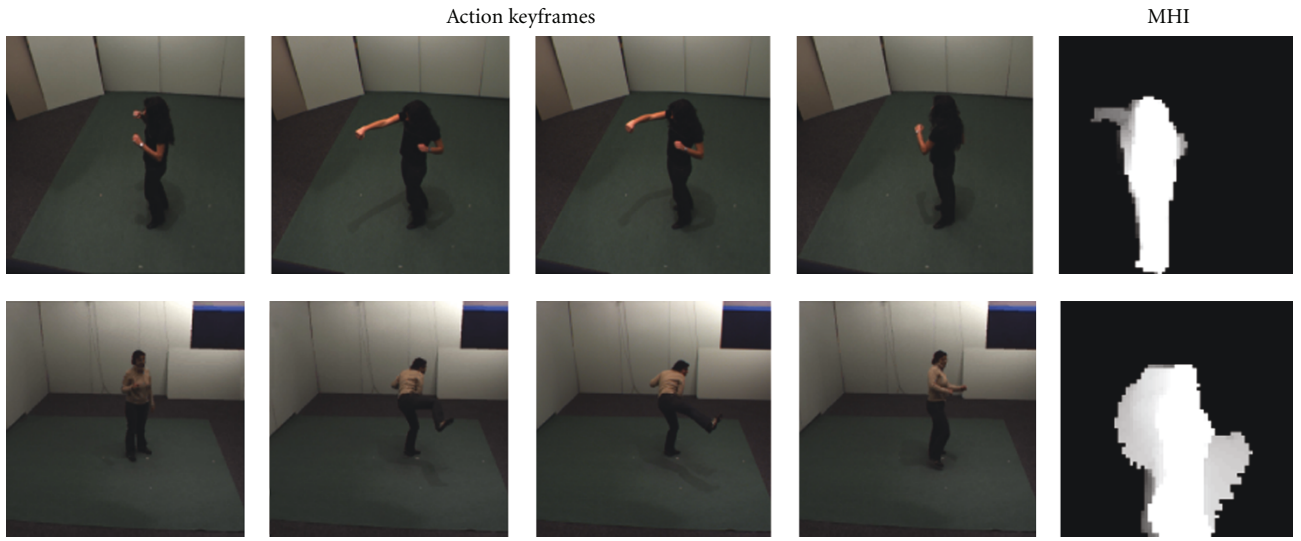


Figure 2: Each row shows four keyframes from different actions and the associated motion history image.

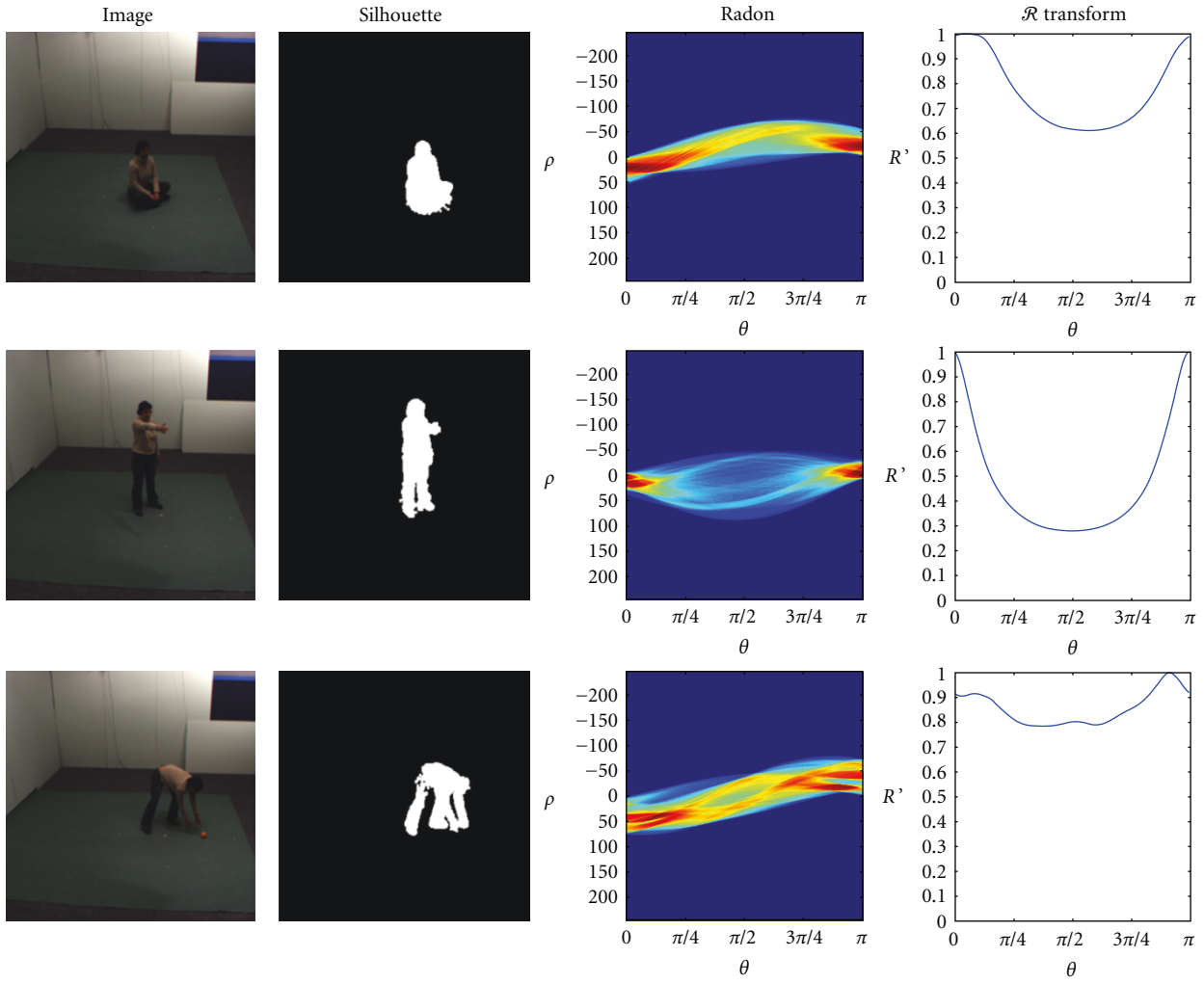


Figure 3: Each row shows the steps to apply the $\mathcal{R}$ transform to an image. The images (first column) are segmented to recover the silhouette (second column). The 2D radon (third column) is calculated and, using (3), the Radon transform is converted to the $\mathcal{R}$ transform (fourth column).

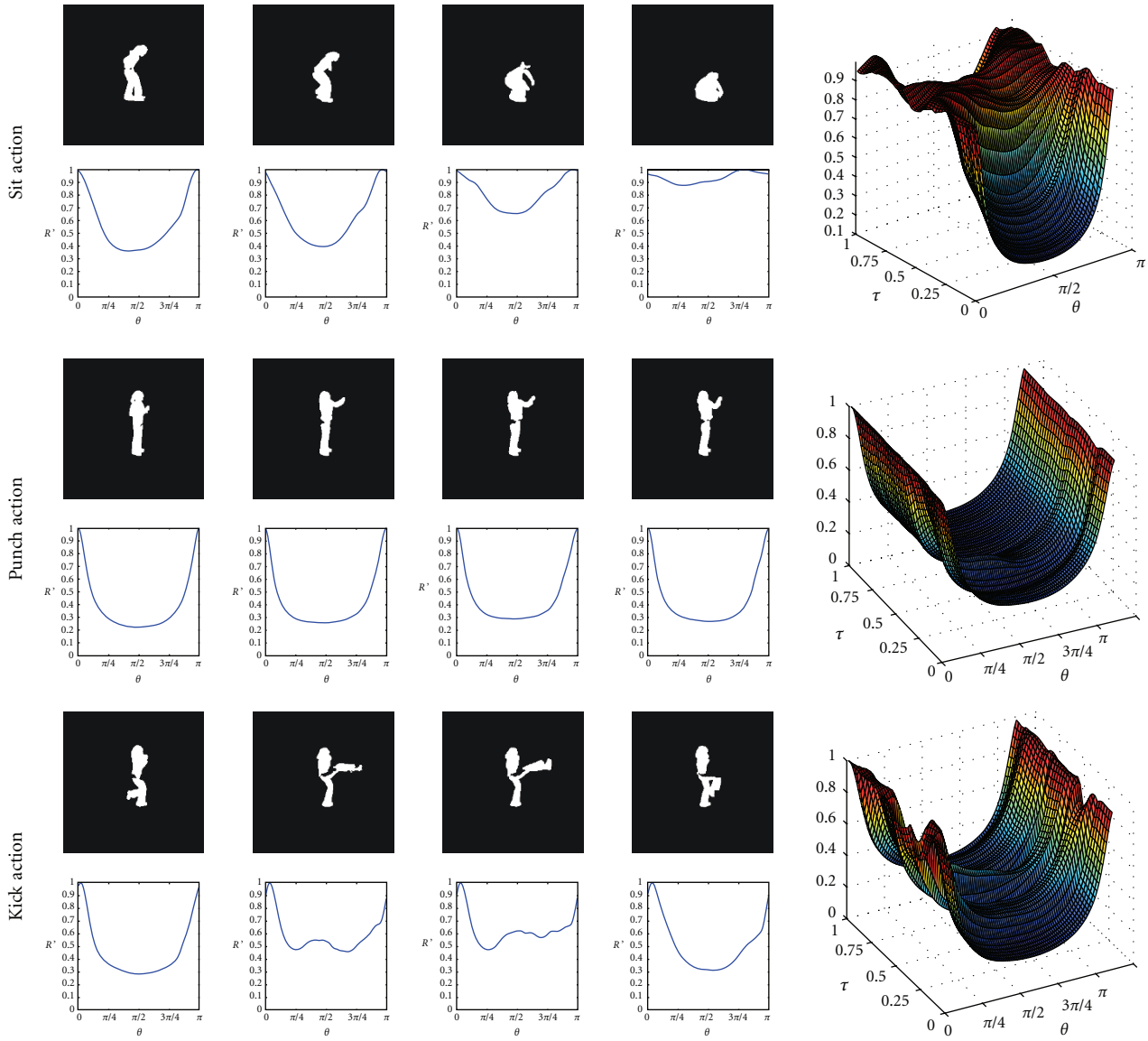


FIGURE 4: Each row shows a set of silhouette keyframes from videos of an actor performing sitting, punching, and kicking, respectively. The corresponding $\mathcal{R}$ transform curve is shown below each keyframe. The graph on the right shows the $\mathcal{R}$XS motion descriptor for the video clip.

on or near a low-dimensional manifold. By learning how the data varies as a function of the dominant cause of change (viewpoint, in our case), we can provide a representation which does not require storing examples of all possible viewpoints of the actions of interest.

4.1. Dimensionality Reduction. Owing to the curse of dimensionality, most data analysis techniques on high-dimensional points and point sets do not work well. One strategy to overcome this problem is to find an equivalent lower dimensional representation of the data. Dimensionality reduction is the technique of automatically learning a low-dimensional representation for data. Classical dimensionality reduction techniques rely on Principal Component Analysis (PCA) [15] and Independent Component Analysis (ICA) [16]. These methods seek to represent data as linear combinations of a small number of basis vectors. However, many datasets, including the action descriptors considered in this work, tend to vary in ways which are very poorly approximated by changes in linear basis functions.

Techniques in the field of manifold learning embed high-dimensional data points which lie on a *nonlinear* manifold onto a corresponding lower-dimensional space. There exists a number of automated techniques for learning these low-dimensional embeddings, such as Isomap [17], semidefinite embedding (SDE) [18], and LLE [19]. These methods have been used in computer vision and graphics for many applications, including medical image segmentation [20] and light parameter estimation from single images [21].

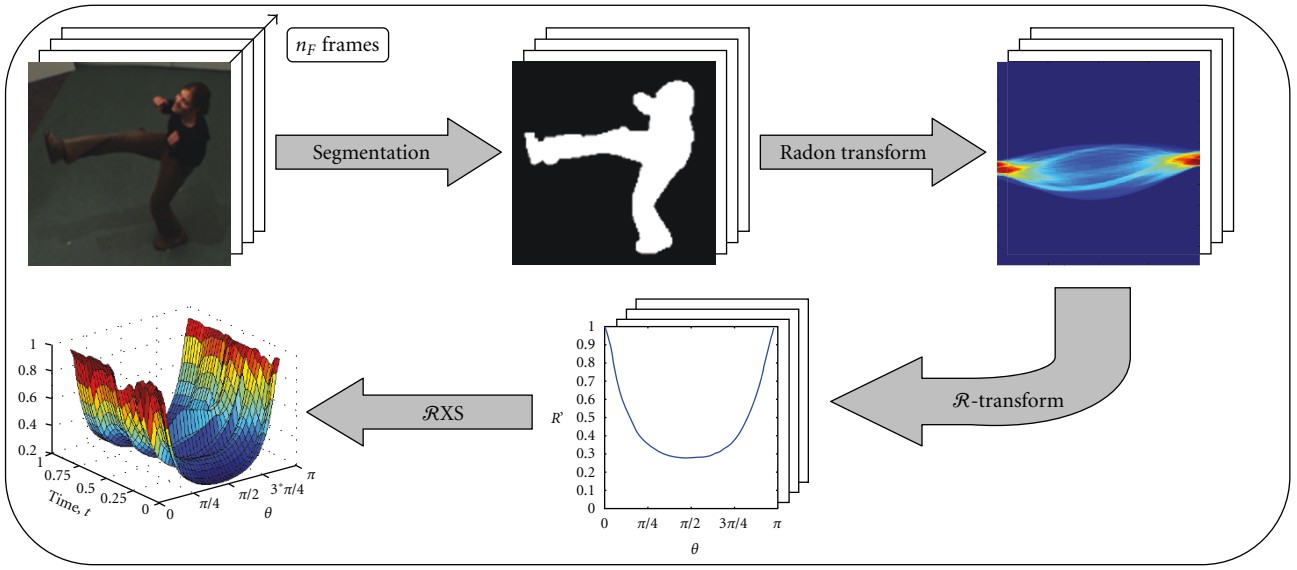


Figure 5: Diagram depicting the construction of the $\mathcal{R}$XS motion descriptor for a set of images.

In this paper, we use the Isomap algorithm, but the general approach could be applied with any of the other nonlinear dimensionality algorithms.

Isomap embeds points in a low-dimensional Euclidean space by preserving the geodesic pair-wise distances of the points in original space. In order to estimate the (unknown) geodesic distances, distances are calculated between points in a trusted neighborhood and generalized into geodesic distances using an all-pairs shortest-path algorithm. As is the case with many manifold learning algorithms, discovering which points belong in the trusted neighborhood is a fundamental operation. Typically, the Euclidean distance metric is used, but other distance measures have been shown to lead to a more accurate embedding of the original data. In the following section, we discuss how the choice of metrics to calculate the distance between motion descriptors affects learning a low-dimensional embedding, and present the distance metrics, both for MHI and $\mathcal{R}$XS, that we use to learn the viewpoint manifolds.

4.2. Distances on the Viewpoint Manifold. Recently, there has been some work [22, 23] on the analysis of formal relationships between the transformation group underlying the image variation and the learned manifold. In [23], a framework is presented for selecting image distance metrics for use with manifold learning. Leveraging ideas from the field of Pattern Theory, the authors propose a set of distance metrics which correspond to common image differences, such as non-rigid transformations and lighting changes. For the work of this paper, the data we seek to analyze differs from the natural images described in that work, in that they are compact representations of video and the differences between MHIs or $\mathcal{R}$XS which differ due to viewpoint or appearance are not accurately estimated using the metrics presented.

4.2.1. MHI Distance Metric. The most common distance metric used to classify MHIs is the Mahalanobis distance between feature vectors of the Hu moments [24]. It has been shown that this metric provides sufficient discriminative power for classification tasks. However, as we show in Figure 6, this metric does not adequately estimate the distances on the manifold of MHIs that vary only due to the viewpoint of the camera. For the *check-watch* action depicted in Figure 1, MHIs were calculated at 64 evenly-spaced camera positions related by a rotation around the vertical axis of the actor. Figure 6 shows the 3D Isomap embedding of this set of MHIs, where each point represents a single MHI embedded in the low-dimensional space, and the line connects adjacent positions. Given that these images are related by a single degree-of-freedom, one would expected the points to lie on some twisted 1-manifold in the feature space. In Figure 6(a), visual inspection shows that this structure is not recovered using Isomap and Hu moment-based metric.

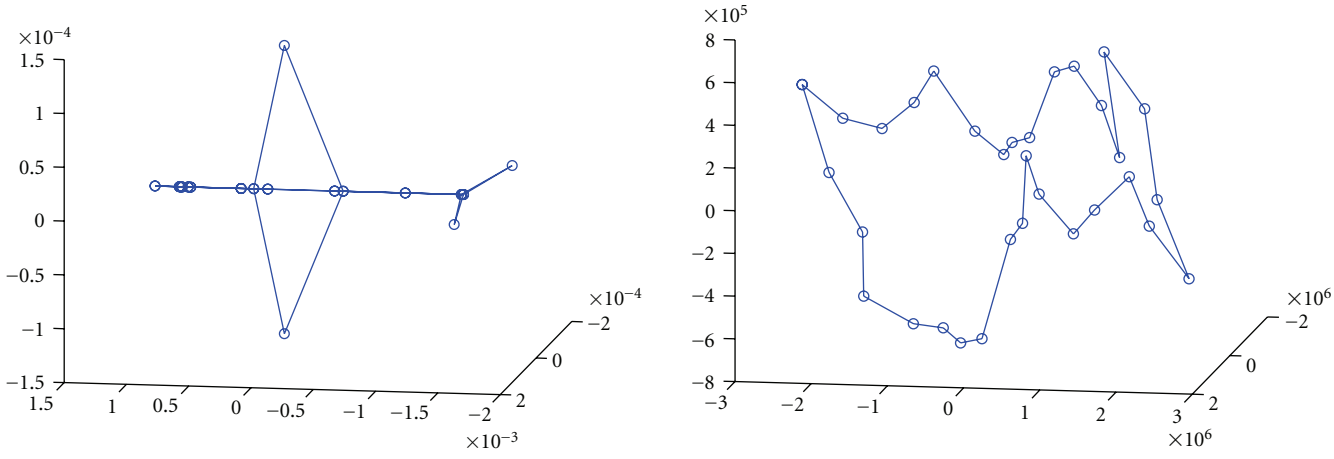
To address this problem, we propose using a rotation-, translation-, and scale-invariant Fourier-based transform [25]. We apply the following steps to calculate the feature vector, $H^F(r,\phi)$ for each MHI, $H(x,y)$. To achieve translation-invariance, we apply the 2D Fourier transform to the MHI:

$$F(u,v) = \mathcal{F}\left(H(x,y)\right), \tag{6}$$

F is then converted to a polar representation, $P(r,\theta)$, so that rotations in the original image correspond to translations in P along the θ axis. Then, to achieve rotation-invariance and output the Fourier-based feature vector, $H^F(r,\phi)$, the 1D Fourier transform is calculated along the axis of the polar angle, θ:

$$H^F(r,\phi) = \left|\mathcal{F}_\theta(P(r,\theta))\right|. \tag{7}$$

Then, the distance between two MHIs H_i and H_j is simply the L_2-norm of H_i^F and H_j^F. With this metric, we recover



(a) Hu moment feature vector

(b) Fourier-based feature vector

FIGURE 6: These graphs depict the 3-dimensional Isomap embedding of the motion history images of an actor performing the *check-watch* motion from camera viewpoints evenly spaced around the vertical axis. (a) shows the embedding using the Hu moment-based metric commonly used to classify motion history images and (b) shows the embedding using the Fourier-based metric described in Section 4.2.

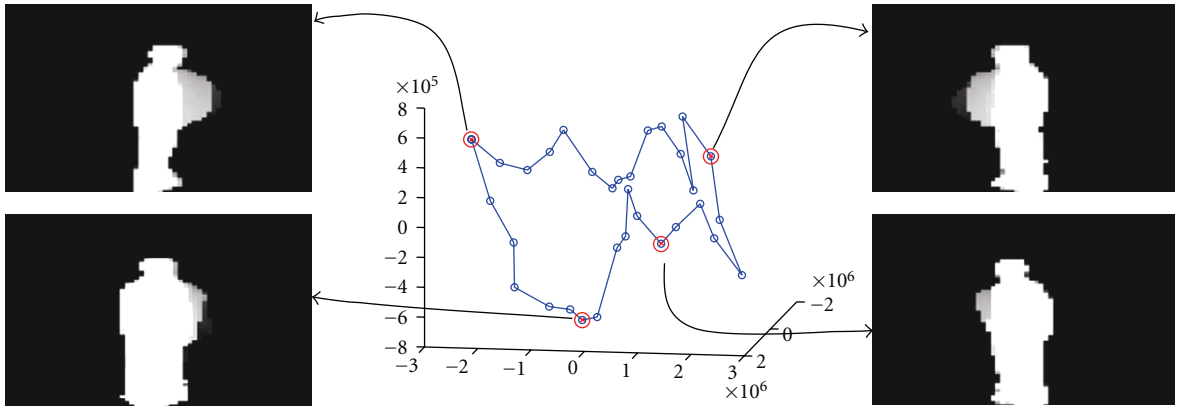


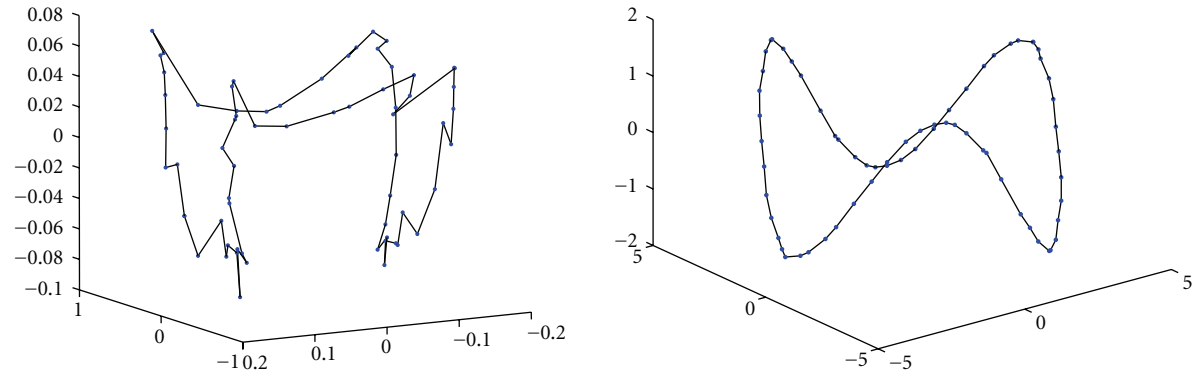
FIGURE 7: The graph in the center shows the 3D embedding of motion history images from various viewpoints of an actor performing the *check-watch* action. For four locations, the corresponding MHI motion descriptors are shown.

the embedding shown in Figure 6(b) which more accurately represents the change in this dataset. In Figure 7 we see the 3D Isomap embedding and the corresponding MHIs for four marked locations.

4.2.2. $\mathcal{R}$ Transform Surface Distance Metric. The $\mathcal{R}$ transform represents the distribution of pixels in the silhouette image. Therefore, to represent differences in the $\mathcal{R}$ transform, and similarly the $\mathcal{R}$XS, we select a metric for measuring differences in distributions. We use the 2D diffusion distance metric [26], which approximates the Earth Mover's Distance [27] between histograms. This computationally efficient metric formulates the problem as a heat diffusion process by estimating the amount of diffusion from one distribution to the other.

Figure 8 shows a comparison of the diffusion distance metric with the standard Euclidean metric. The graphs show the 3D Isomap embedding using the traditional Euclidean distance and the diffusion distance on a dataset containing $\mathcal{R}$ transform surfaces of 64 evenly-spaced views of an actor performing an action. As with the MHIs, these feature vectors are related by a smooth change in a single degree of freedom, and should lie on or near a 1-manifold embedded in the feature space. The embeddings using the diffusion distance metric appear to represent a more accurate measure of the change in the data due to viewpoint. Figure 9 shows the 3D Isomap embedding of 64 $\mathcal{R}$ transform surfaces from various viewpoints of an actor performing the punching action. For the four marked locations, the corresponding high-dimensional $\mathcal{R}$ transform surfaces are displayed.

For the examples in this paper, we use data obtained from viewpoints around the vertical axis of the actor. This data lies on a 1D cyclic manifold. Most manifold learning methods do not perform well on this type of data; however, we employ a common technique [28] and first embed this data into three dimensions, then to obtain the 1D embedding, we



(a) Euclidean distance (b) Diffusion distance

Figure 8: These graphs compare the embeddings using (a) the Euclidean distance and (b) the diffusion distance. Each point on the curve represents an $\mathcal{R}$XS and the curve connects neighboring viewpoints.

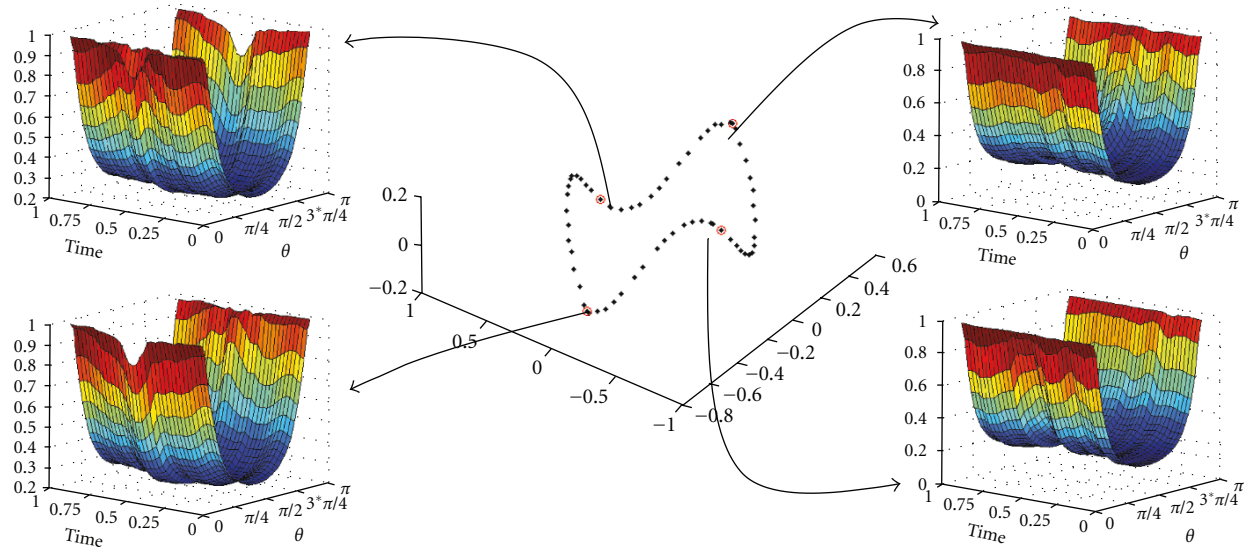


Figure 9: The graph in the center shows the 3D embedding of 64 $\mathcal{R}$ transform surfaces from various viewpoints of an actor punching. For four locations, the corresponding $\mathcal{R}$ transform surface motion descriptors are shown.

parameterize this closed curve using $\phi \in [0, 1]$ where the origin is an arbitrarily selected location on the curve.

It is worth noting that even though the input data was obtained from evenly-spaced viewing angles, the points in the embedding are not evenly spaced. The learned embedding, and thus the viewpoint parameter, ϕ, represents the manifold by the amount of change between surfaces and not necessarily the amount of change between the viewpoint. This is beneficial to us, as the learned parameter, ϕ, provides an action-invariant measure of the viewpoint, whereas a change in the $\mathcal{R}$ transform surfaces as a function of a change in viewing angle would be dependent on the specific action being performed. In the following section, we describe how we use this learned viewpoint parameter, ϕ, to construct a compact view-invariant representation of action.

5. Generating Action Functions

In this section, we leverage the power of learning the embedding for these motion descriptors. In Section 4, we showed how each action descriptor can vary smoothly as a function of viewpoint and how this parameter can be learned using manifold learning. Here, we develop a compact view-invariant action descriptor, using the learned parameterization, ϕ. So, for testing, instead of storing the entire training set of action descriptors, we can learn a compact function which generates a surface as a function of the viewpoint. To avoid redundancy, we will describe our function learning approach with the $\mathcal{R}$ transform surface, since the approach is identical using MHIs.

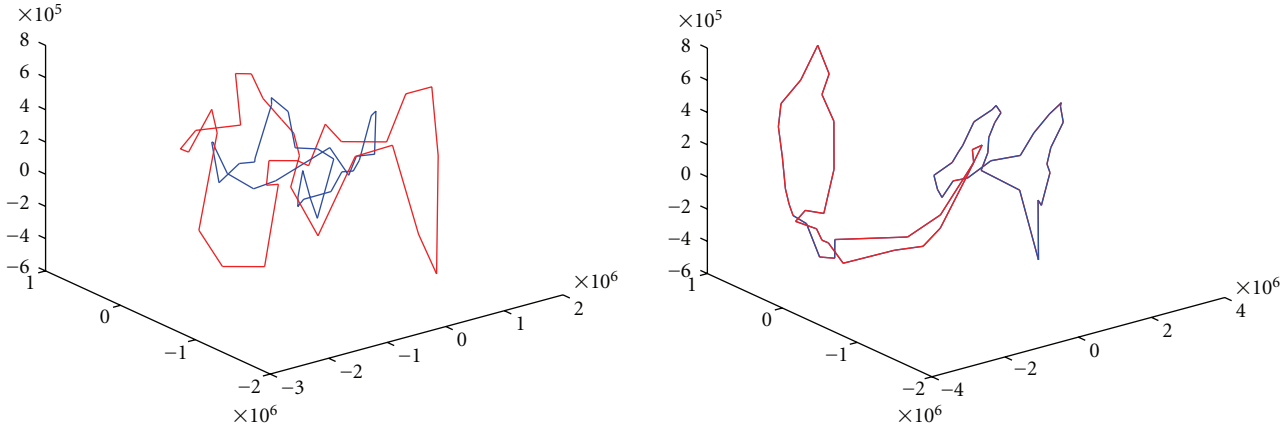


Figure 10: Two examples showing poor parameter estimates using manifold learning. In (a), two embeddings were computed separately and mapped to the same coordinate system and on (b), the mixed dataset was passed as input to Isomap. Neither approach recovers the shared manifold structure of both datasets. (Where color is available, one dataset is shown in blue and the other in red.)

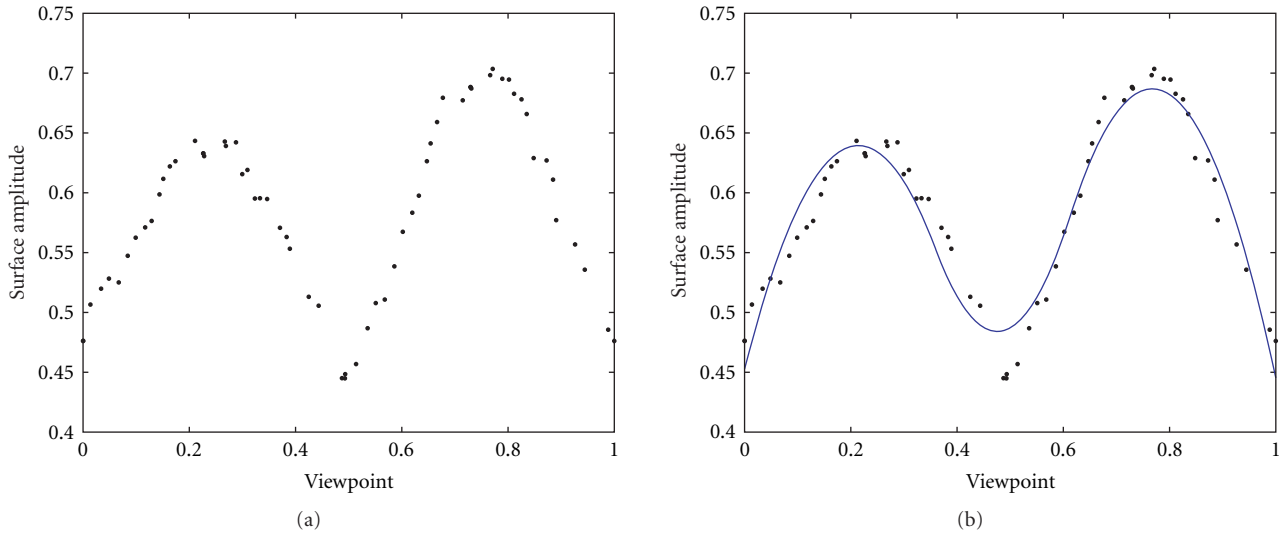


Figure 11: (a) The change in surface value of a specific location on an $\mathcal{R}$ transform surface as function of viewpoint and (b) a cubic B-spline approximation to learn the function, $f_{\langle\theta,t\rangle}(\phi)$, which represents the change in a surface at position $\langle\theta, t\rangle$ as a function of ϕ.

For a set of $\mathcal{R}$ transform surfaces related by a change in viewpoint, S_i, we learn the viewpoint parameter, ϕ_i, as described in Section 4. Then, for each location $\langle\theta, t\rangle$, we can plot the value of each descriptor $S(\theta, t)$ as function of ϕ_i. Figure 11 shows two such plots for the set of descriptors depicted in Figure 9. Each plot shows how the descriptor changes at a given location as a function of ϕ_i. Then, for each location, $\langle\theta, t\rangle \in \Theta$, we can approximate the function, $f_{\langle\theta,t\rangle}(\phi)$ using cubic B-splines, in a manner similar to [29]. Figure 11(b) shows an example of the fitted curve.

Constructing an arbitrary $\mathcal{R}$ transform surface, S_ϕ for a given ϕ is straightforward:

$$S_\phi(\theta, t) = f_{\langle\theta,t\rangle}(\phi) \tag{8}$$

For a new action to be tested, we construct an $\mathcal{R}$ transform surface S_q and use numerical optimization to estimate the viewpoint parameter, $\widetilde{\phi}_q$:

$$\widetilde{\phi}_q = \underset{\phi}{\operatorname{argmin}} \left\| f(\phi) - S_q \right\|. \tag{9}$$

The score for matching surface, S_q, to an action given $f(\phi)$ is simply $\|S_q - S_\phi\|$. In Section 6, to demonstrate action recognition results, we select the action which returns the lowest reconstruction error.

5.1. Individual Variations. The viewpoint manifolds, described so far, are constructed for a single actor performing a single action. We can extend this representation in a natural

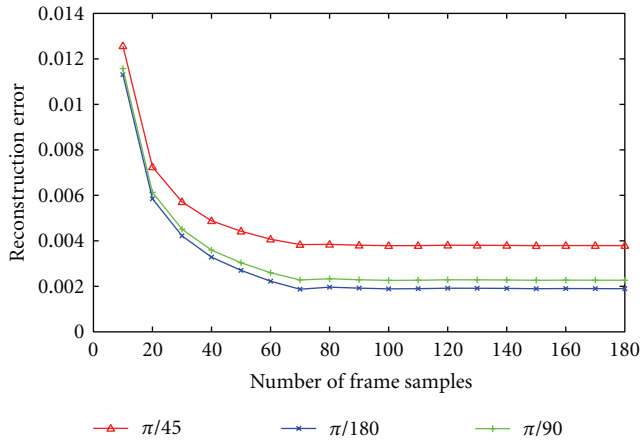


FIGURE 12: Mean reconstruction error for $\mathcal{R}$ transform surfaces as a function of the sampling size. The original size of the surface is 180 (degrees resolution in Radon transform) * number of frames in the video. The three curves represent 45, 90, and 180 (full) samples in the 1st dimension and the x-axis represents the sampling of the 2nd dimension. In our experiments we select a 90*40 representation.

way to account for individual variations in body shape and how the action is performed by learning the shared representation of a set of actors. This process requires first registering the action descriptors for all actors, or learning the space of combined manifolds.

Due to the variations in the way in which the same person performs the same action, or more importantly, how multiple people perform the same action, it is not the case that different motion descriptors are identical. In the case of people with significantly different body shapes, the respective descriptors may appear quite different.

As pointed out in [28], one of the well-known limitations of Isomap and other manifold learning algorithms is the inability to recover a meaningful low-dimensional embedding for a mixed dataset, such as a set consisting of viewpoint-varying motion descriptors obtained from multiple subjects. The two main reasons are that the interclass differences are generally not on the same scale as the intraclass differences and the high-dimensional structure of the dataset may vary greatly due to relatively small differences (visually) in the data points. Figure 10 shows an example where the embeddings are computed separately and mapped on the same coordinate frame and an example where the input consisted of the mixed data.

We address this problem in a manner similar to [28] by mapping all of the separately computed embeddings onto a unified coordinate frame to obtain a "mean" manifold. Using the Coherent Point Drift algorithm [30], we warp each manifold onto a set of reference points, or more specifically, one of the computed embeddings, selected arbitrarily. At this point, it is possible to separate the style variations from the content (viewpoint) variations, but for the work presented here, we proceed with the "mean" manifold.

For the reference manifold, $\overline{f}_{\langle\theta,t\rangle}(\phi)$, we calculate the mean value at each location $\langle\theta, t\rangle$ and for the set of manifolds, calculate the function variance:

$$\sigma^2_{\langle\theta,t\rangle} = \frac{1}{n}\sum_i \left(S_i(\theta,t) - \overline{f}_{\langle\theta,t\rangle}(\phi)\right) \tag{10}$$

where n is the number of $\mathcal{R}$ transform surfaces in the set. Intuitively, this is a measure of the inter-class variation of feature point $\langle\theta, t\rangle$. For action recognition, given a new example S_q, we modify (9) to include the function variances and calculate the normalized distance:

$$\widetilde{\phi}_q = \underset{\phi}{\operatorname{argmin}} \left\| \frac{\overline{f}(\phi) - S_q}{\sigma^2} \right\|. \tag{11}$$

In the following section, we show how this compact representation can be used to reconstruct motion descriptors from arbitrary viewpoints from the original input set, classify actions, and estimate the camera viewpoint of an action.

6. Results

For the results in this section, we used the Inria XMAS Motion Acquisition Sequences (IXMAS) dataset [5] of 29 actors performing 12 different actions. (The full dataset contains more actors and actions, but not all the actors performed all the actions. So, for the sake of bookkeeping, we only selected the subset of actors and actions for which each actor performed each action.) This data was collected by 5 calibrated, synchronized cameras. To obtain a larger set of action descriptors from various viewpoints for training, we animated the visual hull computed from the five cameras and projected the silhouette onto 64 evenly spaced virtual cameras located around the vertical axis of the subject. For each video of an actor performing an action from one of the 64 virtual viewpoints, we calculated the $\mathcal{R}$ transform surface as described in Section 3. For data storage reasons, we subsampled each $180 * n_f$ $\mathcal{R}$ transform surface (where n_f is the number of frames in the sequence) to 90*40. Figure 12 shows the plot of the mean reconstruction error as a function of the sampling size for 30 randomly selected actor/action pairs. In our testing, we found no improvement in action recognition or viewpoint estimation beyond a reconstruction error of .005, so we selected the size 90*40, which provides a reasonable trade-off between storage and fidelity to the original signal.

Following the description in Section 4, we embed the subsampled descriptors using Isomap (with $k = 7$ neighbors as the trusted neighborhood parameter) to learn the viewpoint parameter, ϕ_i and our set of reconstruction functions. In this section, we show results for discriminative action recognition and viewpoint estimation.

6.1. Action Recognition. We constructed $\mathcal{R}$ transform surfaces for each of the 12 actions for the 64 generated viewpoints. For each action, we learned the viewpoint manifold, and the action functions. To test the discriminative power of this method, we queried each of the 64*12 $\mathcal{R}$ transform surfaces with the 12 action classes for each actor. The graphs in Figure 13 show the results for these experiments. For

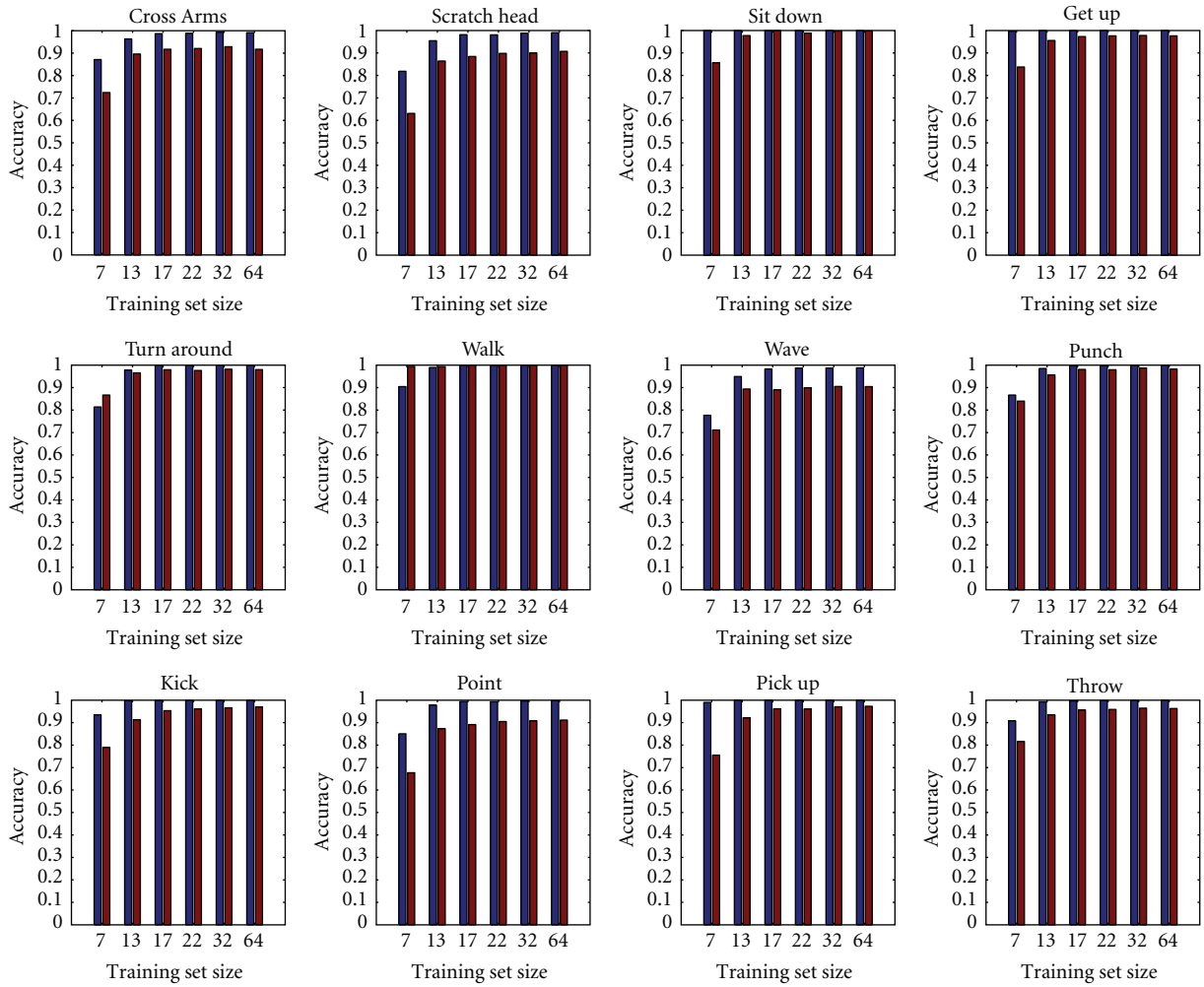


Figure 13: Each graph shows the average accuracy of action recognition experiments using both descriptors ($\mathcal{R}$ transform surfaces: left, blue and MHI: right, red) for 12 different actions using 29 different actors from the IXMAS dataset. For each method, we varied the number of viewpoints used in training.

each method, we trained the action function using evenly-spaced views around the vertical axis of the actor. However, we varied the number of viewpoints we used to construct the surface that represents the action. This is represented along the x-axis of each graph. For each graph, the bars show the average accuracy for each method. In general, the $\mathcal{R}$ transform surface motion descriptor outperformed the MHI-based approach, except for the walk action.

6.2. Viewpoint Estimation. To test the robustness of our compact model for viewpoint estimation, we augmented our action recognition experiments. We learned the viewpoint parameters for 64 evenly-spaced viewpoints about the vertical axis of the actor and constructed action functions using subsets of these descriptors for training. We calculated the difference between the estimated viewpoint ϕ_q (using (9)) and the known parameter ϕ_0. Figure 14 shows the mean error results for each action. Most of the results were very accurate as 1% error roughly corresponds to a rotation of 0.1 radians from a distance of 3 meters.

7. Summary and Conclusions

In this paper, we addressed the problem of view-invariant action recognition from a single camera by developing a manifold learning-based framework to develop a compact representation of action primitives from a continuous set of viewpoints. We demonstrated this approach using two motion descriptors: (1) the well-known motion history images of temporal templates and (2) one we extended from a shape descriptor for use in action recognition. Using this framework, we reported action recognition results that are comparable to methods which, unlike our method, require multiple cameras in the testing phase. In addition to action recognition, this approach also allows for simultaneous viewpoint estimation.

The work presented in this paper is an early step towards a learning system for viewpoint- and appearance-invariance in action recognition. The general direction of this work is to model how action representations change as a function of the variations common of video-based human motion

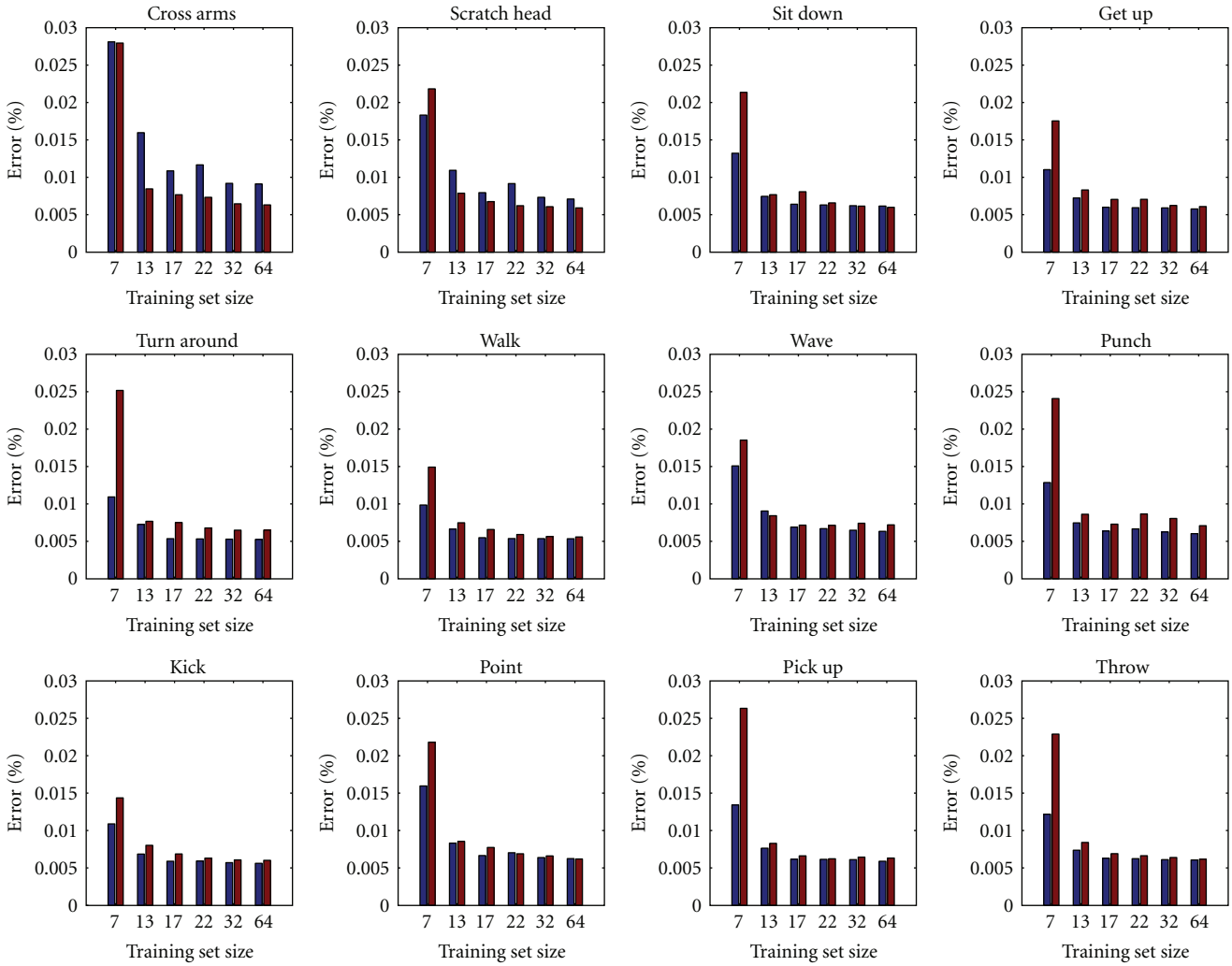


Figure 14: Each graph shows the average error in our viewpoint estimates when we tested using 64 evenly-spaced viewpoints from rotations about the vertical axis of an actor. We computed both descriptors ($\mathcal{R}$ transform surfaces: left, blue and MHI: right, red) for 12 different actions using 29 different actors from the IXMAS dataset and used (9) to estimate the viewpoint.. For each method, we varied the number of viewpoints used for training. (1% error roughly corresponds to a rotation of 0.1 radians from a distance of 3 meters.)

capture. We demonstrated results for the restricted case of 1D viewpoint changes, but believe that this general approach can be taken for other types of variations, including more general motion. In the future, we would like to extend this approach beyond silhouette-based motions and include appearance information to avoid the self-occlusion problem inherent to action silhouettes from certain viewpoints.

References

[1] J. W. Davis and A. F. Bobick, "The representation and recognition of human movement using temporal templates," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '97)*, pp. 928–934, 1997.

[2] E. Shechtman and M. Irani, "Space-time behavior based correlation," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 1, pp. 405–412, June 2005.

[3] N. V. Boulgouris, K. N. Plataniotis, and D. Hatzinakos, "Gait recognition using linear time normalization," *Pattern Recognition*, vol. 39, no. 5, pp. 969–979, 2006.

[4] A. Veeraraghavan, R. Chellappa, and A. K. Roy-Chowdhury, "The function space of an activity," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, vol. 1, pp. 959–968, 2006.

[5] D. Weinland, R. Ronfard, and E. Boyer, "Free viewpoint action recognition using motion history volumes," *Computer Vision and Image Understanding*, vol. 104, no. 2-3, pp. 249–257, 2006.

[6] A. Yilmaz and M. Shah, "Recognizing human actions in videos acquired by uncalibrated moving cameras," in *Proceedings of IEEE International Conference on Computer Vision*, vol. 1, pp. 150–157, 2005.

[7] L. Wang, W. Hu, and T. Tan, "Recent developments in human motion analysis," *Pattern Recognition*, vol. 36, no. 3, pp. 585–601, 2003.

[8] K. Kulkarni, S. Cherla, A. Kale, and V. Ramasubramanian, "A framework for indexing human actions in video," in

Proceedings of the 1st International Workshop on Machine Learning for Vision-Based Motion Analysis (MLVMA '08), 2008.

[9] F. Lv and R. Nevatia, "Single view human action recognition using key pose matching and viterbi path searching," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '07)*, pp. 1–8, June 2007.

[10] Y. Sheikh, M. Sheikh, and M. Shah, "Exploring the space of a human action," in *Proceedings of IEEE International Conference on Computer Vision*, vol. 1, pp. 144–149, 2005.

[11] R. Souvenir and J. Babbs, "Learning the viewpoint manifold for action recognition," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, 2008.

[12] S. Tabbone, L. Wendling, and J.-P. Salmon, "A new shape descriptor defined on the radon transform," *Computer Vision and Image Understanding*, vol. 102, no. 1, pp. 42–51, 2006.

[13] M. A. Fiddy, "The radon transform and some of its applications," *Journal of Modern Optics*, vol. 32, pp. 3–4, 1985.

[14] Y. Wang, K. Huang, and T. Tan, "Human activity recognition based on R transform," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR '07)*, pp. 1–8, June 2007.

[15] I. T. Jolliffe, *Principal Component Analysis*, Springer, New York, NY, USA, 1986.

[16] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, John Wiley & Sons, New York, NY, USA, 2001.

[17] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[18] K. Q. Weinberger and L. K. Saul, "Unsupervised learning of image manifolds by semidefinite programming," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, vol. 2, pp. 988–995, Washington, DC, USA, June 2004.

[19] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[20] Q. Zhang, R. Souvenir, and R. Pless, "On manifold structure of cardiac MRI data: application to segmentation," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, vol. 1, pp. 1092–1098, 2006.

[21] H. Winnemöeller, A. Mohan, J. Tumblin, and B. Gooch, "Light waving: estimating light positions from photographs alone," *Computer Graphics Forum*, vol. 24, no. 3, pp. 433–438, 2005.

[22] D. Donoho and C. Grimes, "When does isomap recover the natural parameterization of families of articulated images?" Tech. Rep., Stanford University, Palo Alto, Calif, USA, August 2002.

[23] R. Souvenir and R. Pless, "Image distance functions for manifold learning," *Image and Vision Computing*, vol. 25, no. 3, pp. 365–373, 2007.

[24] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 3, pp. 257–267, 2001.

[25] M. Zhang, "Feature extraction in character recognition with associative memory classifier," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 10, no. 4, pp. 325–348, 1996.

[26] H. Ling and K. Okada, "Diffusion distance for histogram comparison," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, vol. 1, pp. 246–253, 2006.

[27] Y. Rubner, C. Tomasi, and L. J. Guibas, "A metric for distributions with applications to image databases," in *Proceedings of IEEE International Conference on Computer Vision*, pp. 59–66, 1998.

[28] A. Elgammal and C.-S. Lee, "Separating style and content on a nonlinear manifold," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, vol. 1, pp. 478–485, Washington, DC, USA, June 2004.

[29] H. Murase and S. K. Nayar, "Illumination planning for object recognition using parametric eigenspaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 12, pp. 1219–1227, 1994.

[30] A. Myronenko, X. Song, and M. Carreira-Perpinan, "Non-rigid point set registration: coherent point drift," in *Advances in Neural Information Processing Systems*, B. Schölkopf, J. Platt, and T. Hoffman, Eds., vol. 19, pp. 1009–1016, MIT Press, Cambridge, Mass, USA, 2007.