Learning the Viewpoint Manifold for Action Recognition

Richard Souvenir and Justin Babbs
Department of Computer Science
University of North Carolina at Charlotte
{souvenir, jlbabbs}@uncc.edu

Abstract

Researchers are increasingly interested in providing video-based, view-invariant action recognition for human motion. Addressing this problem will lead to more accurate modeling and analysis of the type of unconstrained video commonly collected in the areas of athletics and medicine. Previous viewpoint-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 data set previously only acheieved using multiple simultaneous views.

1. Introduction

The analysis of human motion from video is an important problem in computer vision with many 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. State of the art methods rely on marker-based motion capture, which has shown to be very effective for obtaining accurate body models and pose estimates. However, these studies are generally conducted in a laboratory environment and, therefore, preclude *in situ* analysis. Video-based solutions hold the promise for action recognition in more natural environments, e.g., 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. 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. These methods address the problem of view-dependence of the single camera systems, but generally require a multi-camera laboratory setting similar to the marker-based solutions.

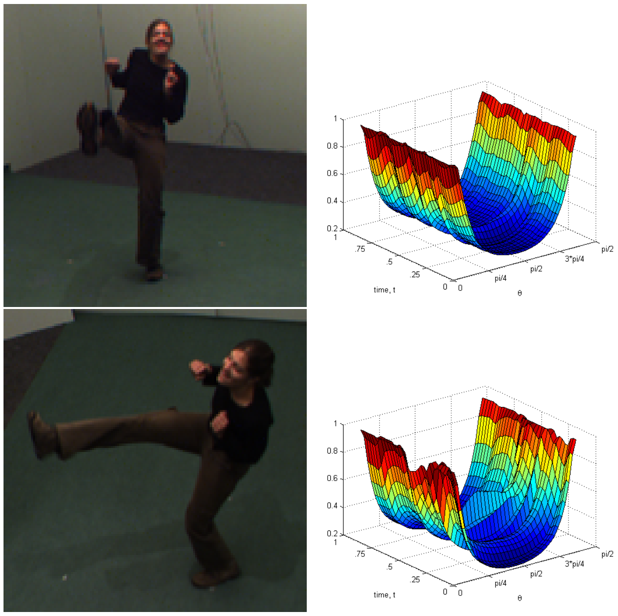


Figure 1. These images show two viewpoints, at the same time-point, of an actor kicking. The surfaces on the right are the motion descriptors we describe in this paper which were derived from each video. (This data was obtained from the IXMAS data set.)

The work presented in this paper presents a framework for learning a viewpoint-invariant representation of primitive actions (e.g., walk, punch, kick, sit) that can be used for video obtained from a single camera, such as any one of the views in Figure 1, which shows an example of an actor performing an action captured from multiple viewpoints.

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

Our framework supports learning how the appearance of an action varies as the viewpoint changes by learning a low dimensional representation of action primitives using manifold learning.

We review recent work in action recognition in Section 2. In Section 3, we describe the motion descriptor we use in our framework and continue in Section 4 to describe 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 remarks about future directions of this project.

2. Related Work

The literature on human motion analysis and action recognition is vast (see [12] for a taxonomy of recent techniques). In this section, we focus on a few existing methods which are most similar 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. Other view-dependent methods include extending 2D image correlation to 3D for space-time blocks [7].

Over time, researchers have begun to focus on using multiple cameras to support viewpoint-invariant action recognition. One method [14] extends temporal templates by constructing a 3D representation, known as a motion history volume. This extension calculates the spatial and temporal extent of the visual hull, rather than the silhouette, of an action. In [15] the authors exploit properties of the epipolar geometry of a pair 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 [5], the authors rely on the compression possible due to the similarility of various actions at particular poses to maintain a compact $(|actions| * |viewpoints|)$ representation for single-view recognition. In [8], 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.

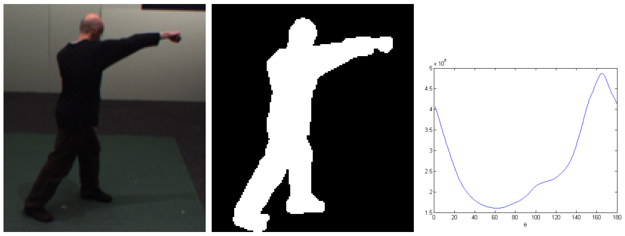


Figure 2. This example image (left) is converted into a silhouette (middle) to which the $\mathcal{R}$ transform (right) can be applied.

3. Representing Motion

For 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 form the basis of most action recognition systems. For this project, we extend a recently developed shape descriptor, the $\mathcal{R}$ transform [10], into an action descriptor. 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 for use in action recognition.

3.1. $\mathcal{R}$ transform

The $\mathcal{R}$ transform was developed as a shape descriptor to be used in object classification from images. The $\mathcal{R}$ transform converts a silhouette image to a compact 1D signal through the use of the two-dimensional Radon transform. In image processing, the Radon transform, like the Hough transform, is commonly used to find lines in images. For an image $I(x, y)$, the Radon transform, $g(\rho, \theta)$, using polar coordinate (ρ, θ), is defined as:

$$g(\rho, \theta) = \sum_x \sum_y I(x, y)\delta(x \cos\theta + y \sin\theta - \rho), \quad (1)$$

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:

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

Figure 2 shows an example image, the derived silhouette showing the segmentation between the actor and the background, and the $\mathcal{R}$ transform .

The $\mathcal{R}$ transform has several properties that make it particularly useful for action recognition from a sequence of silhouettes. First, the transform is translation-invariant. Translations of the silhouette do not affect the value of $\mathcal{R}$ transform , which allows us to match images of actors

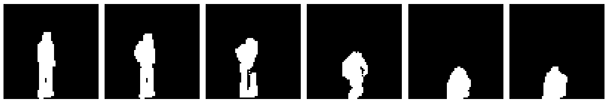


Figure 3. Six silhouette keyframe images of an actor sitting down. The images are low resolution and noisy. The $\mathcal{R}$ transform surface for this action is shown in Figure 4.

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. 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:

$$\mathcal{R}'(\theta) = \frac{\mathcal{R}(\theta)}{max_{\theta'}(\mathcal{R}(\theta'))} \tag{3}$$

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.

3.2. $\mathcal{R}$ transform Surface

In previous work using the $\mathcal{R}$ transform for action recognition [13], the authors trained Hidden Markov Models to learn which sets of unordered $\mathcal{R}$ transform 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, our representation of actions. We define this surface for a video of silhouette images $I(x, y, t)$ as:

$$S(\theta, t) = \mathcal{R}'_t(\theta) \tag{4}$$

where $\mathcal{R}'_t(\theta)$ is the normalized $\mathcal{R}$ transform for frame t in I. Figure 3 shows six silhouette keyframe images of an actor sitting down. The video of this action contained 70 frames and Figure 4 shows the $\mathcal{R}$ transform surface generated from this sequence. For each action, 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.

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 the 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 have less dramatic, but similarly intuitive $\mathcal{R}$ transform surface representations.

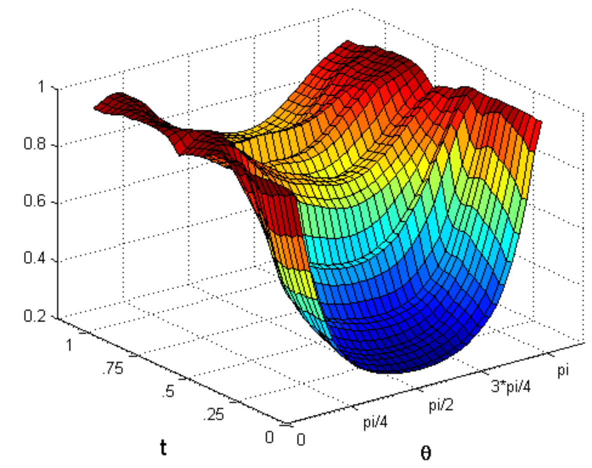


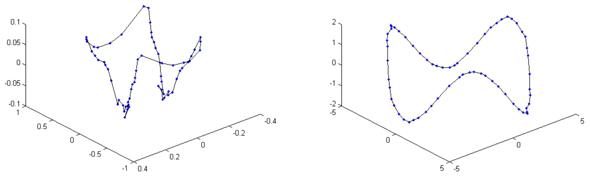
Figure 4. An $\mathcal{R}$ transform surface of the sit-down action. This surface models a changing $\mathcal{R}$ transform of silhouette images from time t=0 to time t=1. Key frames from this action are shown in Figure 3.

Section 2 highlighted several existing descriptors used for action recognition. The benefit of using our surface-based representation of the $\mathcal{R}$ transform relates to its use in providing a compact representation for view-invariant action recognition. In the next 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), which lie 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 set of actions of interest.

Dimensionality reduction is the technique of automatically learning a low-dimensional representation for data. The most commonly used dimensionality reduction technique in computer vision applications is Principal Component Analysis (PCA) [3], which seeks to represent data as linear combinations of a small number of basis vectors. However, many data sets, specifically $\mathcal{R}$ transform surfaces related by a change in viewpoint, tend to vary in ways which are very poorly approximated by changes in linear basis



(a) Euclidean Distance (b) Diffusion Distance

Figure 5. These graphs compare the embeddings using (a) the Euclidean distance and (b) the diffusion distance. Each point on the curve represents an $\mathcal{R}$ transform surface and the curve connects neighboring viewpoints.

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 exist a number of automated techniques for learning these low-dimensional embeddings. These methods have been used in computer vision for many applications, including medical image segmentation [16]. To learn the low-dimensional embedding of $\mathcal{R}$ transform surfaces, we choose to use the Isomap [11] algorithm.

Isomap embeds points in a low-dimensional Euclidean space by preserving the geodesic pair-wise distances of the points in the 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 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, for certain classes of problems, other distance measures have been shown to lead to a more faithful embedding of the original data [9].

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 $\mathcal{R}$ transform surface, we select a metric for measuring differences in distributions. We use the 2D diffusion distance metric [4], which approximates the Earth Mover's Distance [6] 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 5 shows a comparison of the distance metrics. 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. Empirically, we determined that the embeddings using the diffusion distance metric represented an accurate measure of the change in the data due to viewpoint. Figure 6(a) 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 surfaces are depicted in Figure 6(b).

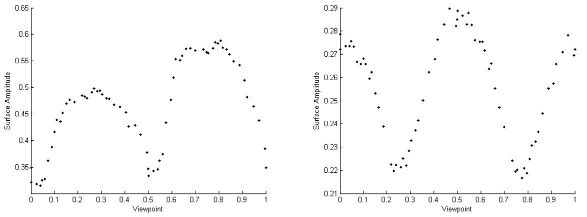


Figure 7. Each plot shows the change in surface value of a specific location on an $\mathcal{R}$ transform surface as function of viewpoint.

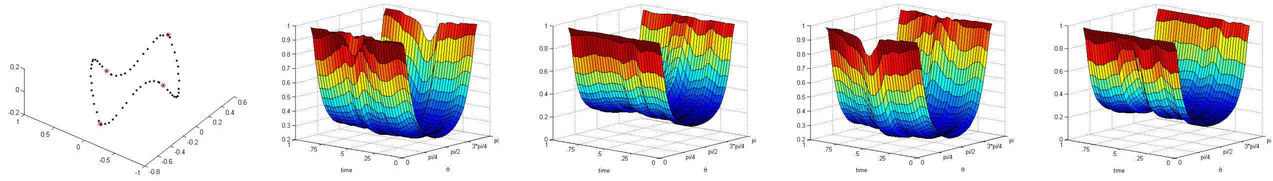
For the examples in this paper, we use data obtained from viewpoints around the vertical axis of the actor. This data lies on a 1D cyclic manifold. Most manifold learning methods do not perform well on this type of data, however, we employ a common technique [2] and first embed this data into three dimensions, then to obtain the 1D embedding, we parameterize this closed curve using $\phi \in [0, 1]$ where the origin is 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 next section, we describe how we use this learned viewpoint parameter, ϕ, to construct a compact view-invariant representation of action.

5. Functional Representation of $\mathcal{R}$ Transform Surface Manifold

In this section, we leverage one of the most useful properties of our $\mathcal{R}$ transform surface representation. In Section 4, we showed how $\mathcal{R}$ transform surfaces 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 training set of action descriptors, we learn a function which generates a surface as a function of the viewpoint.

For a set of $\mathcal{R}$ transform surfaces related by a change in viewpoint, S_i, we learn the viewpoint parameter, ϕ_i. Then, for each location $\langle \theta, t \rangle$, we can plot the value of each surface $S(\theta, t)$ as function of ϕ_i. Figure 7 shows two such plots for the set of descriptors depicted in Figure 6. Each plot shows how the surface changes at a given location as a function of



(a) 3D Embedding

(b) $\mathcal{R}$ transform surfaces at selected points

Figure 6. The graph in (a) 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 surfaces are depicted in (b).

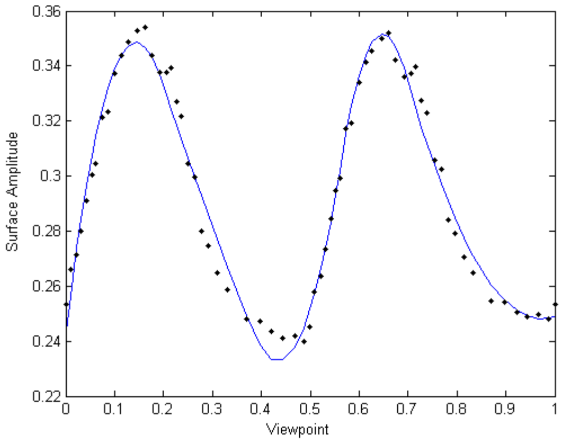


Figure 8. A cubic B-spline approximation to learn the function, $f_{\langle\theta,t\rangle}(\phi)$, which represents the change in a surface at position $\langle\theta, t\rangle$ as a function of ϕ.

ϕ_i. Then, for each location, $\langle\theta, t\rangle \in \Theta$, we approximate the function, $f_{\langle\theta,t\rangle}(\phi)$ using cubic B-splines. Figure 8 shows an example of the fitted curve.

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

$$S_\phi(\theta, t) = f_{\langle\theta,t\rangle}(\phi) \tag{5}$$

For a query action, we construct an $\mathcal{R}$ transform surface S_q and use numerical optimization to estimate the viewpoint parameter, $\tilde{\phi_q}$:

$$\tilde{\phi_q} = argmin_\phi ||f(\phi) - S_q||. \tag{6}$$

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

This viewpoint manifold of $\mathcal{R}$ transform surfaces is constructed from a single actor for a single action. We can extend this representation in a natural 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 of action descriptors for all actors. In many data sets used in dimensionality reduction, it is usually the case the intra-class variation is much smaller than the inter-class variation. Because of this, most embedding techniques rarely learn a unified embedding for mixed data. To overcome this limitation, we take an approach, similar to [2], and embed each data set individually followed by a non-rigid registration to a reference embedding. For the reference manifold, we calculate $f_{\langle\theta,t\rangle}(\phi)$ (as previously described) and for the set of manifolds, calculate the function variance:

$$\sigma^2_{\langle\theta,t\rangle} = \frac{1}{n}\sum_i S_i(\theta, t) - f_{\langle\theta,t\rangle}(\phi) \tag{7}$$

where n is the number of $\mathcal{R}$ transform surfaces in the set. Intuitively, this is a measure of the inter-class variation of surface point $\langle\theta, t\rangle$. For action recognition, given a new example S_q, we modify Equation 6 to include the function variances and calculate the normalized distance:

$$\tilde{\phi_q} = argmin_\phi || \frac{f(\phi) - S_q}{\sigma^2} ||. \tag{8}$$

In the next section, we show how this compact representation can be used to reconstruct $\mathcal{R}$ transform surfaces from the original input set, classify actions, and estimate the viewpoint of an action given the $\mathcal{R}$ transform surface.

6. Results

For the results in this section, we used the Inria XMAS Motion Acquisition Sequences (IXMAS) dataset [14] of various actors performing 13 different actions. This data was collected by 5 calibrated, synchronized cameras. To obtain a larger set of $\mathcal{R}$ transform surfaces 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 storage and efficiency reasons, we sub-sampled each $180 * n_f$

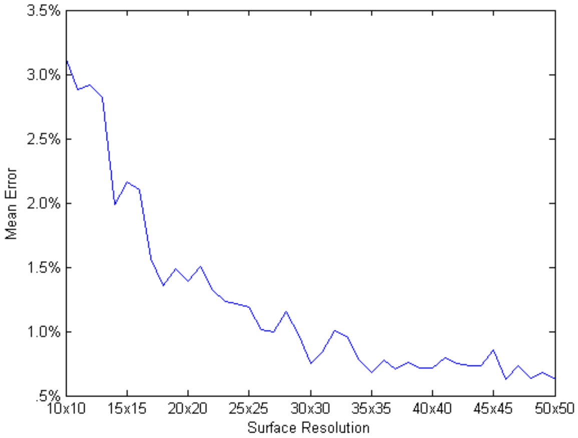


Figure 9. Mean reconstruction error for $\mathcal{R}$ transform surfaces as a function of the sampling size. In our experiments we select a 35*35 representation.

surface (where n_f is the number of frames in the sequence) to 35*35. Figure 9 shows the plot of the mean reconstruction error as a function of the sampling size. The selected size, 35*35, provides a reasonable trade-off between storage and fidelity to the original signal efficiency.

Following the description in Section 4, we embed the sub-sampled $\mathcal{R}$ transform surfaces 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 13 actions for the 64 generated viewpoints. For each action, we learned the viewpoint manifold, and the surface representation functions. To test the discriminative power of this method, we queried each of the 64*13 $\mathcal{R}$ transform surfaces with the 13 action classes. Figure 10 shows the confusion matrix for this experiment. Each column of the matrix depicts the prediction for each of the 64 instances of that action. Brighter colors represent higher values. For each query action, the target action provided the best match.

To take into account individual variation in actions, we selected one of the actors from the dataset as a reference and calculated the function variances (Equation 7) using four other actors from the data set for training. For testing, we used an actor not in the training set and calculated the matching score for each of the 13 actions from multiple viewpoints. For each action, on average, the highest match was to the correct query action. This compares favorably to the results published in [14]. However, it is worth noting that they used a multi-camera query and constructed a 3D volume, while our method relies on traditional 2D input.

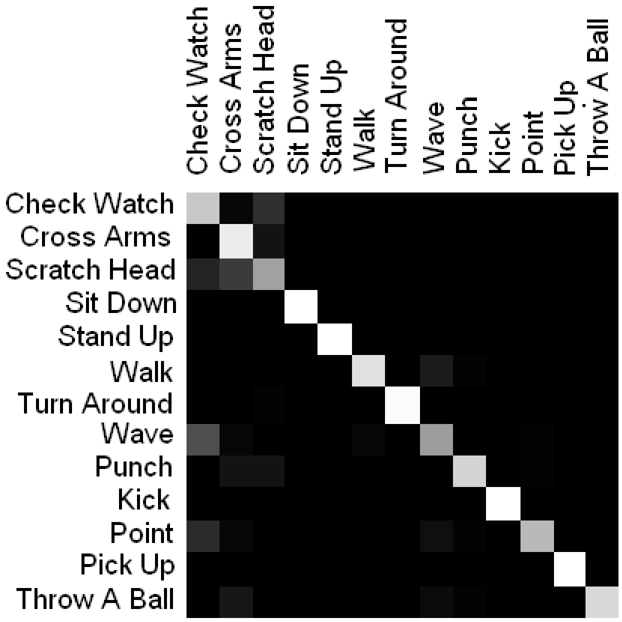


Figure 10. Confusion matrix showing the results of a classification experiment using 64 viewpoints of an actor performing each action in the IXMAS data set. Each column of the matrix depicts the prediction for each of the 64 instances of that action. Brighter colors represent higher values.

6.2. Viewpoint Estimation

To test the robustness of our compact model for viewpoint estimation, we performed leave-one-out (LOO) experiments for a single actor where the manifold was generated using all but one, S_0, of the surfaces. Using the known viewpoint parameter of S_0, ϕ_0, we calculated the difference between the estimated viewpoint ϕ_q (using Equation 6) and the known parameter ϕ_0. Table 1 shows the mean results for all 64 views for each action. Most of the results were very accurate. In general, the errors occurred in situations where the bulk of the action was occluded (e.g., viewing a punch from behind the actor) and accurate estimates of the angle are impossible.

In a similar experiment using a viewpoint manifold constructed from multiple actors, we noticed that the accumulation of the individual variation had a significant impact on the results. Figure 11 shows the results. We estimated the viewpoint parameter for a test actor from multiple viewpoints using a viewpoint manifold constructed from 5 different actors. A majority of the results were accurate ($< 5\%$), however, a noticeable fraction of the results were incorrect by an amount ($\sim 50\%$) which indicated the best match was to the diametrically opposite viewpoint. We believe that this is due to ambiguity inherent in using a silhouette-based descriptor and that these results can be im-

Action	% Error	Action	% Error
watch	0.98% ± 1.1%	arm cross	1.7% ± 0.79%
scratch	1.4% ± 1.1%	sit down	0.85% ± 0.58%
stand up	0.87% ± 0.66%	walk	1.0% ± 0.59%
turn	1.2% ± 0.72%	wave	0.96% ± 0.62%
punch	0.94% ± 0.68%	kick	1.1% ± 0.47%
point	0.68% ± 0.41%	pick up	1.3% ± 1.0%
throw	0.73% ± 0.64%		

Table 1. Using 64 evenly-spaced viewpoints from rotations about the vertical axis of an actor, for each action, we calculated the $\mathcal{R}$ transform surface and used Equation 6 to estimate the viewpoint. This table shows the mean error (in percent) and standard deviation from the known locations. (1% error roughly corresponds to a rotation of 0.1 radians from a distance of 3 meters.)

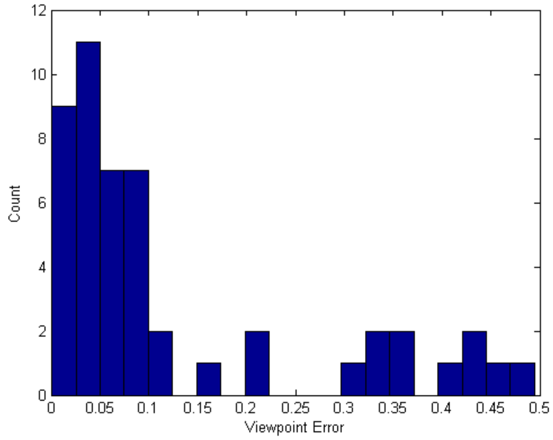


Figure 11. Histogram showing viewpoint error estimates. We compared actions performed from multiple viewpoints to a $\mathcal{R}$ transform surface manifold created from the combined model of 4 other actors.

proved with the addition of an appearance model.

7. Summary and Future Directions

In this paper, we addressed the problem of view-invariant action recognition. We extended a shape descriptor for use in action recognition and learned the viewpoint manifold to provide a compact representation. This allows us to perform simultaneous action recognition and 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 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 *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 928–934, 1997.

[2] A. Elgammal and C.-S. Lee. Separating style and content on a nonlinear manifold. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 478–485, June 2004.

[3] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, 1986.

[4] H. Ling and K. Okada. Diffusion distance for histogram comparison. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 246–253, 2006.

[5] F. Lv and R. Nevatia. Single view human action recognition using key pose matching and viterbi path searching. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007.

[6] Y. Rubner, C. Tomasi, and L. J. Guibas. A metric for distributions with applications to image databases. In *Proc. International Conference on Computer Vision*, pages 59–66, 1998.

[7] E. Shechtman and M. Irani. Space-time behavior based correlation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 405–412, June 2005.

[8] Y. Sheikh, M. Sheikh, and M. Shah. Exploring the space of a human action. *Proc. International Conference on Computer Vision*, 1:144–149, 2005.

[9] R. Souvenir and R. Pless. Image distance functions for manifold learning. *Image Vision Comput.*, 25(3):365–373, 2007.

[10] S. Tabbone, L. Wendling, and J.-P. Salmon. A new shape descriptor defined on the radon transform. *Comput. Vis. Image Underst.*, 102(1):42–51, 2006.

[11] J. Tenebaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, December 2000.

[12] L. Wang, W. Hu, and T. Tan. Recent developments in human motion analysis. *Pattern Recognition*, 36(3):585–601, March 2003.

[13] Y. Wang, K. Huang, and T. Tan. Human activity recognition based on r transform. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007.

[14] D. Weinland, R. Ronfard, and E. Boyer. Free viewpoint action recognition using motion history volumes. *Comput. Vis. Image Underst.*, 104(2):249–257, 2006.

[15] A. Yilmaz and M. Shah. Recognizing human actions in videos acquired by uncalibrated moving cameras. In *Proc. International Conference on Computer Vision*, pages 15–21, 2005.

[16] Q. Zhang, R. Souvenir, and R. Pless. On manifold structure of cardiac mri data: Application to segmentation. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 1:1092–1098, 2006.