A Sketch-Based Approach for Detecting Common Human Actions

Evan A. Suma, Christopher Walton Sinclair, Justin Babbs,
and Richard Souvenir

Department of Computer Science, University of North Carolina at Charlotte,
9201 University City Blvd, Charlotte, NC 28223, U.S.A
{easuma,cwsincla,jlbabbs,souvenir}@uncc.edu

Abstract. We present a method for detecting common human actions in video, common to athletics and surveillance, using intuitive sketches and motion cues. The framework presented in this paper is an automated end-to-end system which (1) interprets the sketch input, (2) generates a query video based on motion cues, and (3) incorporates a new content-based action descriptor for matching. We apply our method to a publicly-available video repository of many common human actions and show that a video matching the concept of the sketch is generally returned in one of the top three query results.

1 Introduction

Automated human activity detection from video could be used for searching archived athletic footage or detecting particular actions in a real-time security setting. However, this type of search is still an open, challenging problem. Commercial solutions (e.g., Google Video) typically employ search methods which do not operate on the content of the video; instead, a text query is matched to metadata of the video such as the title, description, or user comments. The possibility of incomplete or incorrect metadata is a well-known limitation to this approach. This leads in to a host of methods that fall under the umbrella of content-based video retrieval (CBVR).

The literature on CBVR is extensive; see [1] and [2] for surveys. Multiple taxonomies exist for the classification of CBVR approaches. Most relevant to our work are two broad classes of techniques characterized by the query method: (1) text-based (or concept-based) approaches and (2) example-based approaches. Text-based approaches, such as [3], typically rely on some (semi-supervised or unsupervised) step of grouping videos together based on some concept and refining the search within each cluster to obtain the desired result. Example-based approaches, as done in [4], typically match features of a query video against those in the database and return high-scoring matches.

Text-based approaches work when the content of the videos can be described succinctly. Also, a user can attempt to fine-tune search results simply by selecting new keywords to try. However, these methods fail when the query is ambiguous

G. Bebis et al. (Eds.): ISVC 2008, Part I, LNCS 5358, pp. 418–427, 2008.

(e.g., "driving" for cars versus swinging a golf club). Additionally, secondary objects may be overlooked if classification is based upon the primary object or action in the video. Example-based approaches can overcome the limitation of ambiguous searches because a query video is generally more informative than a text label. However, finding representative videos to use for querying other videos can be difficult. More specifically, if a video strongly matching a search concept were easily obtainable, it might not be necessary to perform the query in the first place.

In this paper, we present a method for querying databases of videos of human actions. Our method relies on using sketches of objects and motion cues as queries for video retrieval. We believe that allowing the user to provide the input in this manner combines the best features of both the text- and example-based approaches. The search query is more informative than a text-based approach and does not require that the user explicitly provide an example video. The framework presented in this paper is an automated end-to-end system which (1) interprets the sketch input, (2) generates a query video based on motion cues, and (3) incorporates a new content-based action descriptor for matching.

Sketch recognition is a related area where the goal is to infer the semantics of an input sketch. Unlike those methods, (e.g., [5]), we are not interested in classifying the action represented in the sketch, nor do we need to collect the gesture information associated with creating the sketch. Our goal is to search a video database or stream for a conceptual match. Searching image and video databases using sketches has previously been explored. In [6], the author presents a method using a sketch-based system to search a large static image database. In [7], the authors present a system which queries videos using sketches of motion cues and mainly provides for queries focusing on the translation of objects in a viewing frame and not the finer-grained articulated motions that our system is capable of matching.

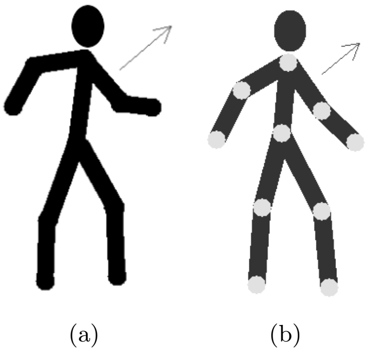


Fig. 1. Example sketches generated using (a) the freehand drawing interface and (b) human body model interface

The paper is organized as follows. In Section 2, we describe our approach for interpreting sketches. In Section 3, we explain our method for matching a query video against a database. In Section 4, we show the results of testing our method on a publicly-available human motion data set. Finally, in Section 5, we conclude with a discussion of limitations of our current approach and future directions for this work.

2 Interpreting Sketches

The query can be provided as a freehand sketch or generated from a human body model. While the former allows for the motion of arbitrary objects, the latter is more consistent and robust for human motion. In this section, we describe how we interpret the input sketches and infer the motion being described.

2.1 Input Methods

Figure 1(a) shows an example freehand sketch query. Our system includes a sketch drawing application which provides two drawing modes: (1) figure mode and (2) arrow mode. In figure mode, the user has access to various drawing tools, common to popular image editing applications, to specify the structure of the figure. In arrow mode, the user can click and drag on the image to add a motion arrow.

For representing human motion, we also a provide point-and-click interface to produce sketches by manipulating a human body model, as shown in Figure 1(b). Here, the user does not draw the figure. Instead, limbs on a human body model are positioned and resized by clicking and dragging the joint locations. For human motion queries, this method is a more robust than freehand sketching because the connectivity of the skeleton, as well as the locations of joints, is already known to the system.

2.2 Interpreting Motion Cues

While we are primarily concerned with videos involving human motion, a freehand sketch could potentially contain objects other than human figures. Also, an arrow in these images can either refer to a specific component of an object (e.g., a limb on a human) or the entire object moving as a whole. Thus, we examine the object's skeleton to determine the joint locations which separate the movable components of the figure. We employ a series of basic image processing techniques to generate a skeleton from a sketch. First, the sketch is blurred using a Gaussian kernel and thresholded to produce a binary image. Then, a medial axis transformation algorithm is applied to generate a skeleton of the image.

Figure 2 illustrates the process for interpreting the motion cues relative to the skeleton. First, the arrows are projected (in the reverse direction) onto the skeleton. We assume that the point of contact, or *origination point*, lies on the moving component being referenced by the arrow. To determine the component (which we assume to be a line segment) on which each origination point lies, the Hough transform [8] is used. In our implementation, we restrict this search to a local neighborhood that is defined by tracing n_l pixels along the skeleton from the origination point in the direction towards the center of mass. Empirically, we found that 20 pixels accurately sample the line segment in a typical 300 x 300 pixel sketch. The endpoints of the returned line segment are stored as the endpoints of the component on the skeleton. The endpoint that is closest along the skeleton towards the center of mass is the detected joint location for the

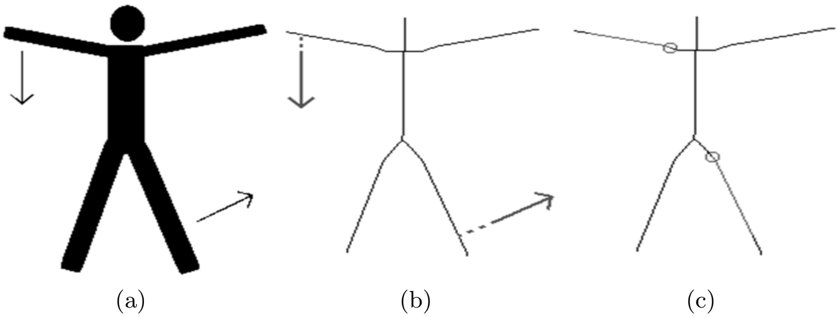


Fig. 2. (a) A freehand sketch with arrows indicating movement. (b) The image after skeletonization with arrows projected back onto the skeleton. (c) The detected line segments and joint locations.

component. If the line segment contains the center of mass, we interpret this to mean that the arrow motion corresponds to a movement of the entire object.

For sketches produced by manipulating the human body model, interpreting the arrows is simplified since the connectivity of the skeleton and locations of joints are known. The origination points are calculated in a similar manner by projecting each arrow in reverse until the skeleton is contacted.

2.3 Sketch Animation

The next step is to generate a video based on the sketch and motion cues to use as a search query. First, the image is segmented into individual components at the joint locations. This is done so that individual components can be translated and rotated according to the motion cues. At each joint, the image is "cut" along the normal to the skeleton, separating the component from the rest of the object. In complex skeletons such as human figures, components may be the children of other components (e.g., forearm and upper arm). To account for this possibility, the parent of each component, if any, is calculated by tracing from the joint location along the skeleton towards the center of mass to detect other joints. For images generated from a human body model, the parent-child relationship between joints is already known.

The angle of component rotation is determined by computing the angular difference between the component vector and the vector formed by the joint location and arrow end point. For each frame, each component is rotated by n_r degrees, where n_r is the total angular rotation divided by the (user-specified) number of frames. For complex motions, where both a child and parent component move, we chose to rotate the child component first, then add this to the rotation of the parent component. Figure 3 shows such an example. Though the motion sequence may not be visually pleasing, the imperfections in the generated video will not significantly affect the matching process.

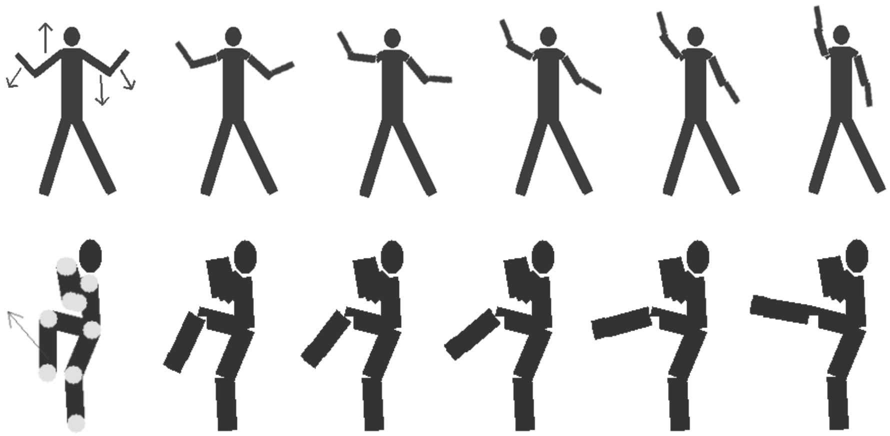


Fig. 3. (top) A freehand input sketch with multiple arrows and frames from the generated video sequence. (bottom) A sketch generated from a human body model and the corresponding video sequence.

3 Matching Videos

For this problem, it is important to model the content of the video rather than the appearance, since sketches do not share appearance characteristics with real video. We extend a recently developed shape descriptor, the $\mathcal{R}$ transform [9], into a motion descriptor. Compared to competing representations, the $\mathcal{R}$ transform is computationally efficient and robust to common image transformations. Here, we describe the $\mathcal{R}$ transform and our extension for matching video sequences.

3.1 $\mathcal{R}$ transform

The $\mathcal{R}$ transform was developed as a shape descriptor for object classification from images. The $\mathcal{R}$ transform converts a silhouette image to a compact 1D signal using the two-dimensional Radon transform. 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:

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

where δ is the Dirac delta function. 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 lines of the same angle, θ, in an image:

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

Figure 4 shows an example image, the derived silhouette showing the segmentation between the actor and the background, and the $\mathcal{R}$ transform.

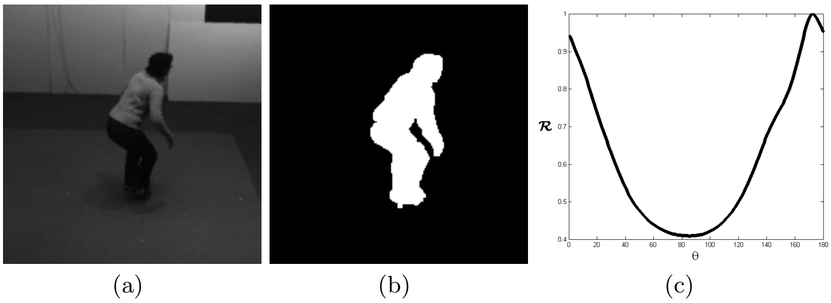


Fig. 4. An image (a) is converted into a silhouette (b) to which $\mathcal{R}'$ (c) is applied

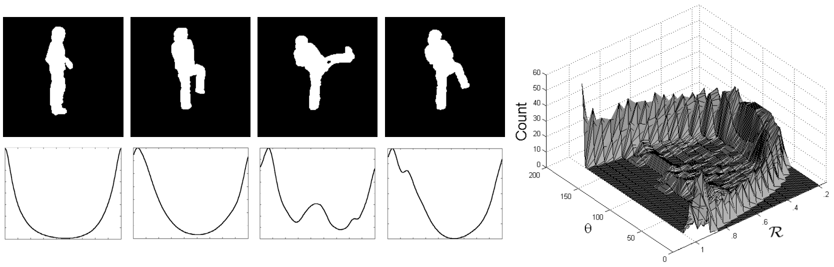


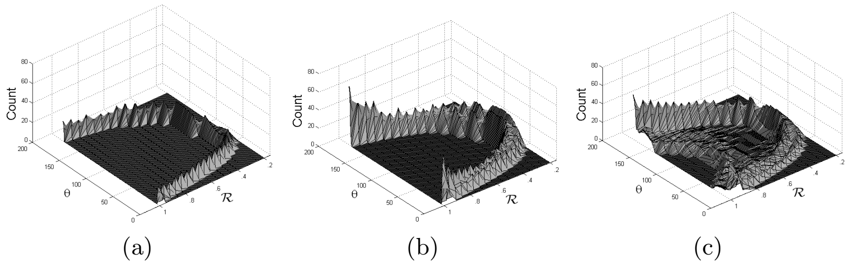
Fig. 5. A set of silhouette keyframes from a video of an actor performing a kick action. The corresponding $\mathcal{R}$ transform curve is shown below each keyframe. The graph on the right shows the $\mathcal{R}$ transform histogram motion descriptor for the video.

The $\mathcal{R}$ transform has several properties that make it particularly useful as an motion descriptor for 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 the same action regardless of the position of the actor in the frame. Second, the $\mathcal{R}$ transform has been shown to be robust to noisy silhouettes (e.g., holes, disjoint silhouettes). This invariance is useful to our method in that extremely accurate segmentation of the actor (in the real videos) 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 $\mathcal{R}$, so we use the normalized transform:

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

3.2 $\mathcal{R}$ transform Histograms

The $\mathcal{R}$ transform has been previously extended for use in action recognition. In [10], the authors trained Hidden Markov Models to learn which sets of unordered $\mathcal{R}$ transform corresponded to which action and in [11], the authors extend the $\mathcal{R}$ transform to include the natural temporal component of actions by concatenating sequential $\mathcal{R}$ transform curves into an $\mathcal{R}$ transform surface. The motion descriptor presented here combines ideas from these two approaches.



(a) (b) (c)

Fig. 6. (a) The $\mathcal{R}$ transform histogram from a sketch video of a pointing motion. (b) The $\mathcal{R}$ transform histogram from a real video of an actor performing a pointing motion. (c) The $\mathcal{R}$ transform histogram from a real video of an actor performing a standing motion.

The $\mathcal{R}$ transform can be applied to a single silhouette frame. A set of frames, therefore, can generate a set of $\mathcal{R}$ transform curves. The problem of matching action videos then becomes a problem of matching these sets of curves. For our representation, we maintain a 2D histogram of the $\mathcal{R}$ transform data. We discretize the 2D space of $\mathcal{R}$ transform curves into 180 (angles) * 20 ($\mathcal{R}'(\theta)$ values). Figure 5 shows the motion descriptor for a video of an actor kicking. The top row shows 4 silhouette keyframes and the bottom row shows the associated $\mathcal{R}$ transform for each frame. The graph on the right shows our $\mathcal{R}$ transform histogram motion descriptor for this video.

3.3 Matching $\mathcal{R}$ transform Histograms

Figure 6 shows (a) the $\mathcal{R}$ transform histograms for a generated sketch video of a pointing motion, (b) the $\mathcal{R}$ transform histogram from real video of the same action, and (c) a different $\mathcal{R}$ transform histogram from a real video of an actor performing a standing motion. On visual inspection, the histograms of the same motions, despite being from both sketches and real videos, appear more similar than the histograms from different motions. To quantify these differences, we employ a histogram-based distance metric. We use the 2D diffusion distance metric [12], which approximates the Earth Mover's Distance [13] 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. In Section 4, we demonstrate that this distance metric is robust to individual variations in action videos and can be used to as the basis of a simple nearest-neighbor classifier to discriminate between dissimilar actions.

4 Results

We used the Inria XMAS Motion Acquisition Sequences (IXMAS) dataset [14] to test our method. This data contains various actors performing multiple, different actions. We tested the system using 10 actions (check watch, cross arms, sit,

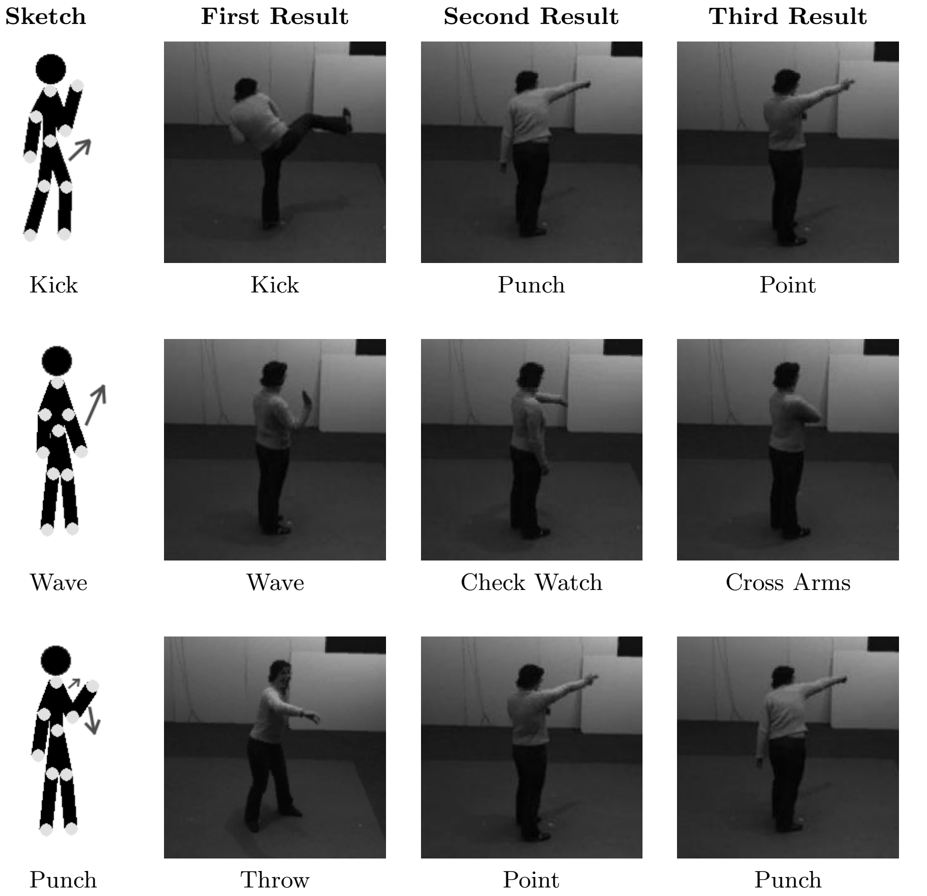


Fig. 7. Sample results from 3 queries on the IXMAS dataset. For each query, the input sketch and keyframes from the 3 top scoring matches are shown.

stand, wave, punch, kick, point, pick up, and throw) from the set. For each action, we generated a sketch and calculated the matching score to all of the action videos. Figure 7 shows sample results for 3 of these types of queries. Each row shows the input sketch and a keyframe from each of the top 3 closest matching videos from the database.

Table 1 summarizes the results. Each cell contains the distance between a user-generated sketch (rows) and a labeled video clip of an actor from the database (columns). (Lower values indicate a closer match.) For 9 out of 10 sketch queries, the intended video was one of the top three scoring matches to our input sketches. Some of the errors (e.g., punch-throw, cross arms-wave) are due simply to the similarity of these actions. For other actions (e.g., check watch), the self-occlusions inherent in the motion leads to ambiguity in the silhouette-based motion descriptor and unpredictable results.

Table 1. Query Results. Each cell contains the distance between a user-genereated sketch and a labeled video clip from the IXMAS data set.

	Video Sequence									
Sketch	**Watch**	**Cross**	**Sit**	**Stand**	**Wave**	**Punch**	**Kick**	**Point**	**Take**	**Throw**
Watch	3.17	3.20	3.35	3.62	2.58	2.51	3.00	3.02	3.45	3.82
Cross	1.79	1.73	2.83	3.00	1.71	1.91	2.02	2.10	3.18	3.52
Sit	3.14	3.17	1.68	1.88	2.80	2.14	1.91	2.41	1.72	2.30
Stand	3.38	3.43	1.84	1.96	3.09	2.48	2.18	2.62	1.86	2.26
Wave	1.39	1.53	2.57	2.49	1.32	2.18	1.71	1.82	2.94	3.16
Punch	3.06	2.98	2.51	2.22	2.80	2.08	2.30	2.00	2.25	1.94
Kick	2.59	2.64	2.02	2.03	2.10	1.73	1.67	1.91	2.12	2.44
Point	2.63	2.76	2.93	2.54	2.53	2.58	2.61	2.05	3.08	2.59
Take	3.22	3.23	2.81	3.19	3.21	3.32	3.06	3.38	2.71	3.64
Throw	3.88	3.90	2.80	2.41	3.63	2.98	3.02	3.06	2.44	2.06

5 Discussion and Future Work

We presented a method for the detection of human actions based on sketch input. Sketches are an intuitive input method, but can be flawed, or, worse, not representative of the user's intended query. This introduces an additional level of ambiguity compared to text-based approaches because it is based on the ability of the user. For our purposes, it makes the results somewhat harder to interpret as the residuals may be due to dissimilar actions or poorly sketched inputs.

The method introduced in this paper aims to represent 3D motions with 2D sketches. While a sketch can represent motion that varies with multiple degrees of freedom, this method is limited to succinct, atomic actions and restricted in the viewpoint. It may be possible to overcome some of these limitations by developing more sophisticated (but, perhaps less intuitive) representations for common human actions. However, we feel that our approach can still be useful to domains such as athletics and surveillance where the large quantities of video data contain examples which can be succinctly described by simple motion cues.

Finally, a technical limitation of our current approach is that it requires binary segmentation for the videos in the search database to generate the motion descriptor. We tried this approach using a different descriptor [15] and a matching method [16], which doesn't require segmentation. Currently, this approach is far more computationally-intensive than our current approach and we are exploring ways to optimize both methods for use on real-time video feeds.

References

1. Lew, M.S., Sebe, N., Djeraba, C., Jain, R.: Content-based multimedia information retrieval: State of the art and challenges. ACM Trans. Multimedia Comput. Commun. Appl. 2, 1–19 (2006)

2. Marchand-Maillet, S.: Content-based video retrieval: An overview. Technical Report 00.06, CUI - University of Geneva, Geneva (2000)
3. Naphade, M.R., Huang, T.S.: Semantic video indexing using a probabilistic framework. ICPR 03, 3083 (2000)
4. Taskiran, C., Chen, J.Y., Albiol, A., Torres, L., Bouman, C., Delp, E.: Vibe: a compressed video database structured for active browsing and search. IEEE Transactions on Multimedia 6, 103–118 (2004)
5. Paulson, B., Hammond, T.: Marqs: retrieving sketches learned from a single example using a dual-classifier. Journ. on Multimodal User Interfaces 2, 3–11 (2008)
6. Lew, M.: Next-generation web searches for visual content. Computer 33, 46–53 (2000)
7. Chang, S.F., Chen, W., Meng, H.J., Sundaram, H., Zhong, D.: Videoq: an automated content based video search system using visual cues. In: Fifth ACM Intnl conference on Multimedia, pp. 313–324. ACM, New York (1997)
8. Duda, R.O., Hart, P.E.: Use of the hough transformation to detect lines and curves in pictures. Commun. ACM 15, 11–15 (1972)
9. Tabbone, S., Wendling, L., Salmon, J.P.: A new shape descriptor defined on the radon transform. Comput. Vis. Image Underst. 102, 42–51 (2006)
10. Wang, Y., Huang, K., Tan, T.: Human activity recognition based on r transform. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
11. Souvenir, R., Babbs, J.: Learning the viewpoint manifold for action recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–7 (2008)
12. Ling, H., Okada, K.: Diffusion distance for histogram comparison. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 246–253 (2006)
13. Rubner, Y., Tomasi, C., Guibas, L.J.: A metric for distributions with applications to image databases. In: Proc. Intnl Conference on Computer Vision, pp. 59–66 (1998)
14. Weinland, D., Ronfard, R., Boyer, E.: Free viewpoint action recognition using motion history volumes. Comput. Vis. Image Underst. 104, 249–257 (2006)
15. Shechtman, E., Irani, M.: Matching local self-similarities across images and videos. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
16. Leibe, B., Leonardis, A., Schiele, B.: Combined object categorization and segmentation with an implicit shape model. In:IEEE Transactions onWorkshop on Statistical Learning in Computer Vision, Prague, Czech Republic, pp. 17–32 (2004)