Contents lists available at ScienceDirect

Computer Vision and Image Understanding

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

A new pose-based representation for recognizing actions from multiple cameras

Selen Pehlivan *, Pınar Duygulu

Department of Computer Engineering, Bilkent University, Ankara 06800, Turkey

ARTICLE INFO

Article history:
Received 20 June 2009
Accepted 4 November 2010
Available online 20 November 2010

Keywords:
Pose representation
Action recognition
Arbitrary view
Multi-camera

ABSTRACT

We address the problem of recognizing actions from arbitrary views for a multi-camera system. We argue that poses are important for understanding human actions and the strength of the pose representation affects the overall performance of the action recognition system. Based on this idea, we present a new view-independent representation for human poses. Assuming that the data is initially provided in the form of volumetric data, the volume of the human body is first divided into a sequence of horizontal layers, and then the intersections of the body segments with each layer are coded with enclosing circles. The circular features in all layers (i) the number of circles, (ii) the area of the outer circle, and (iii) the area of the inner circle are then used to generate a pose descriptor. The pose descriptors of all frames in an action sequence are further combined to generate corresponding motion descriptors. Action recognition is then performed with a simple nearest neighbor classifier. Experiments performed on the benchmark IXMAS multi-view dataset demonstrate that the performance of our method is comparable to the other methods in the literature.

1. Introduction

Understanding human actions by video analysis is one of the challenging problems of computer vision [1–4,6,8,12,13]. The majority of work on this subject focuses on understanding actions from videos captured by a single camera [7,9–11]. However, single-camera systems have drawbacks degrading the recognition performance, such as dependency on viewpoint and problems due to self-occlusion. Multi-camera systems have emerged as a solution [5,14,15,23,28], and now more affordable, they are preferred nowadays against single camera systems for many applications.

While methods that integrate a set of 2D views are also gaining popularity, in the majority of the works a single frame is modeled as a 3D volume. An argument against volumes is about the efficiency of algorithms used for reconstruction, but with recent real-time algorithms [33,34] this problem can be overcome. In volume based systems it is difficult to use an articulated body model reliably [14,15], thus most studies consider actions as 3D shapes that change over time.

The representation should be view independent and should not restrict the camera configuration, that is the action should be recognized for any orientation of the person and for arbitrary positions of the cameras. While there are available 3D shape descriptors that provide rotational invariance for shape matching and retrieval [19–21], for human action recognition it is sufficient to consider variations around the vertical axis of the human body (e.g. the action of an upside-down person should be considered different).

In this study, focusing on the importance of human poses in understanding actions [18,16,17], we present a new representation for encoding the shape of the pose initially provided as a volume. We consider body parts as cylindrical shapes with various sizes and rotations. However, representing volumetric data as cylinders is error-prone to noise and difficult to model. Instead, we use their projections onto horizontal planes intersecting the human body in every layer. We assume that the intersection can be generalized as circles. Furthermore, changes in the number, area and relative position of these circles over the body and over time provide important cues. The proposed representation does not depend on the view and therefore does not require alignment of the poses, which is difficult to do in the case of noisy data, to perform an orientation equalization. We focus on representing the poses to explore to what extent a human pose can help in understanding human actions. We therefore keep the classification part simple and observe that better results can be obtained with our proposed method compared to complex representations.

Before describing the details of our method, we briefly discuss previous work on multi-camera action recognition in Section 2. We then present the view-independent pose representation in Section 3, the motion representation for actions using proposed pose features in Section 4 and our method for action recognition in Section 5. Experimental results on a benchmark dataset are

* Corresponding author.
E-mail addresses: pselen@cs.bilkent.edu.tr (S. Pehlivan), duygulu@cs.bilkent.edu.tr (P. Duygulu).

1077-3142/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2010.11.004

provided in Section 6 and compared with other studies in Section 7. Finally, we summarize our main contributions and present future plans in Section 8.

2. Related work

The approaches proposed for multi-camera systems can be grouped based on how they use images from multiple cameras and how they ensure view independence. One group of approaches use 3D representations of poses constructed from multiple camera images to further model the actions.

In Ref. [15], a 3D cylindrical shape context is presented for multi-camera gesture analysis over volumetric data. For capturing the human body configuration, voxels falling into different bins of a multilayered cylindrical histogram are accumulated. Next, the temporal information of an action is modeled using Hidden Markov Model (HMM). This study does not address view independence in 3D, instead, subjects are asked to rotate while training the system.

A similar work on volumetric data is by Cohen and Li [14] where view-independent pose identification is provided. Reference points placed on a cylindrical shape are used to encode voxel distributions of a pose that result in a shape representation invariant to rotation, scale and translation.

Two parallel studies based on view-invariant features over 3D representation are performed by Canton-Ferrer et al. [22] and Weinland et al. [23]. These studies extend the motion templates of Bobick and Davis [7] to three dimensions, calling them Motion History Volume (MHV). MHV represents the entire action as a single volumetric data, functioning as the temporal memory of the action. In Ref. [23], the authors provide view invariance for action recognition by transforming MHV into cylindrical coordinates and using Fourier analysis. Unlike [23], Canton-Ferrer et al. [22] ensure view invariance by using 3D invariant statistical moments.

One recent work proposes a 4D action feature model (4D-AFM) to build spatial–temporal information [27]. It creates a map from 3D spatial–temporal volume (STV) [11] of individual videos to 4D action shape of ordered 3D visual hulls. Recognition is performed by matching STV of observed video with 4D-AFM.

Stressing the importance of the pose information, in recent studies action recognition is performed over particular poses. Weinland et al. [24] present a probabilistic method based on exemplars using HMM. Exemplars are volumetric key-poses extracted by reconstruction from action sequences. Actions are recognized by an exhaustive search over parameters to match the 2D projections of exemplars with 2D observations. A similar work is that of Lv and Nevatia [25], called *Action Net*. It is a graph-based approach modeling 3D poses as transitions of 2D views rendered from motion capture sequences. Each 2D view is represented as a shape context descriptor in each node of the graph. For recognition, the most probable path on *Action Net* returns the matched sequence with the observed action.

In [26], activities are modeled as a combination of various HMMs trained per body part over motion capture data. Then, 2D tracks over videos and their corresponding matches in 3D help to solve viewpoint variations. Through this method, the study provides a more generic system for unseen and composite activities.

In Ref. [28], Souvenir and Babbs extend shape descriptor based on radon transform and generate 64 silhouettes taken from different views of a visual hull. Action recognition is performed by estimating the optimum viewpoint parameter that would result in the lowest reconstruction error.

In another group of studies, the image features extracted from multiple camera images are fused to understand actions. One such work is presented in [29], where bag-of-video-words approach is applied to a multi-view dataset. The method detects interest points and extracts spatial–temporal information by quantizing them. However, it is hard to infer the poses by the orderless features. Moreover, extracted features like interest points are highly influenced by illumination affects and the actors' clothing, relative to the reconstructed volumes.

In this study, we focus on view-independent features extracted from volumetric poses. We test our approach on IXMAS multi-camera action dataset [23] that was also used by Refs. [23–25,27–29]. Detailed discussion and comparison with other studies can be found in Section 7.

3. Pose representation

In our study, actions are considered as 3D poses that change over time. We emphasize the importance of pose information since in some cases a key pose can reveal enough information for classifying actions [18,16,17]. In the next section, we introduce motion features as changes in consecutive poses, necessary for learning action dynamics.

Volumetric data reveal important cues for understanding the shape of a human pose, but representing an action based on volumetric data is costly and can easily be affected by noise. Our proposed pose representation is effective in keeping the important characteristics of the poses while being efficient with the encodings used. The representation is independent of the view and translation and does not require the poses to be aligned or the actions to be performed in specified orientations and positions.

In the following, we first describe our method to encode the volumetric data, then present the features used for representing the pose information.

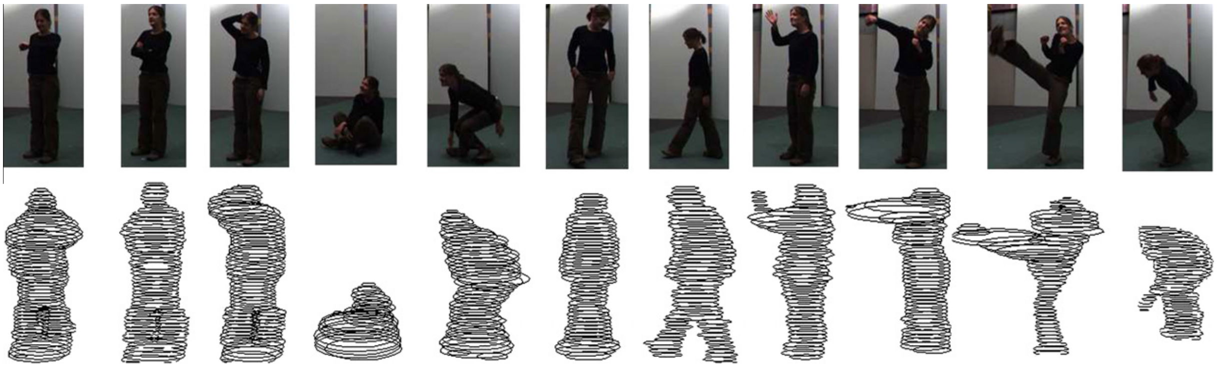

Fig. 1. Poses from some actions: check watch, cross arms, scratch head, sit down, get up, turn around, walk, wave, punch, kick and pick up. We divide the volumetric representation of a pose into layers. The intersection of body segments with each layer are then coded with enclosing circles.

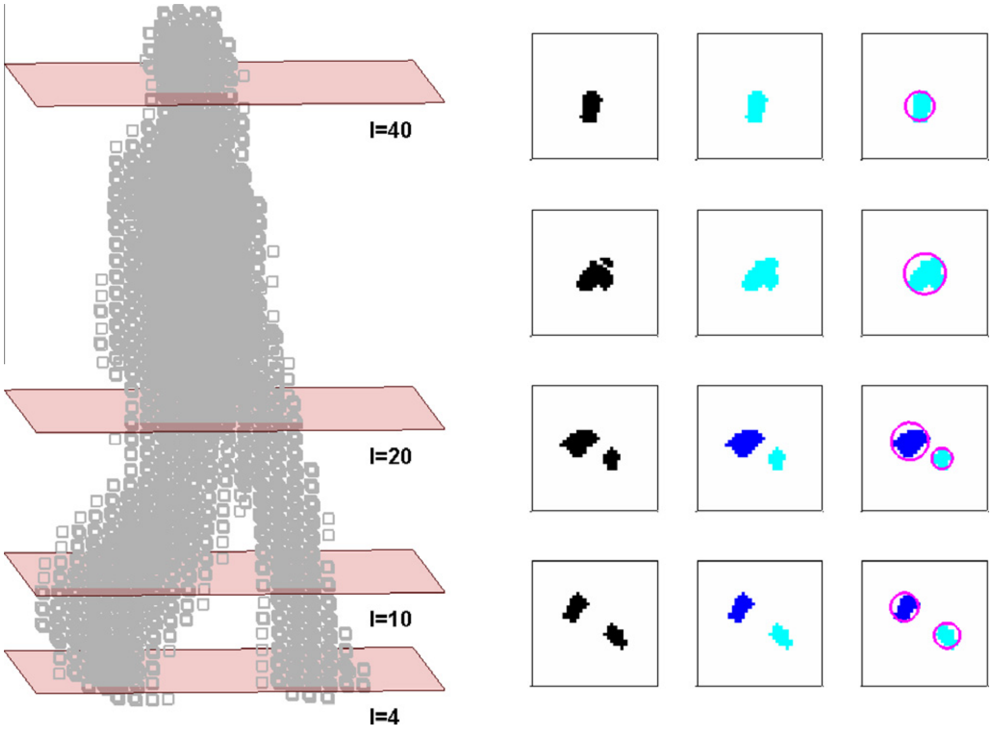

Fig. 2. Representation of volumetric data as a collection of circles in each layer. From left to right: voxel grid of a walking pose and four sample layers (at $l = 4$, $l = 10$, $l = 20$, $l = 40$), image in each layer instance, connected components (CCs) over the enhanced image, and fitted circles.

3.1. Encoding the volumetric data using layers of circles

Human body parts have cylindrical structures, and therefore the intersection of a pose with a horizontal plane can be modeled as a set of ellipses that can be further simplified by circles. We base our approach on this observation, and describe 3D poses using a set of features extracted from circles fitted to projections of the volumetric data on horizontal planes. As shown in Fig. 1, circular representation preserves the pose information while significantly reducing the data to be processed. In Section 6, we compare the performances when ellipses are used for representation rather than circles.

We assume that 3D poses are provided as visual hulls in the form of voxel grids constructed by multi-camera acquisition. For our representation, a voxel grid is divided into horizontal layers perpendicular to the vertical axis and turns into a collection of layers $l = 1, \ldots, n$, where n is the number of layers on the vertical axis. For instance, for a 64 × 64 × 64 voxel grid we obtain 64 layers.

Initially, each layer contains a set of pixels that is the projection of voxels on it. Since visual hull reconstruction may result in noise and defects that should be eliminated prior to feature extraction, we first evaluate each layer as a binary image consisting of pixels. Then we apply morphological closing using a disk structural element with a radius of 2 to close up internal holes and reduce noise in the data. We find the connected components (CCs) in each layer by looking at the 8-neighbors. Finally, a minimum bounding circle is fitted to each CC. Fig. 2 illustrates the entire process on a sample pose.

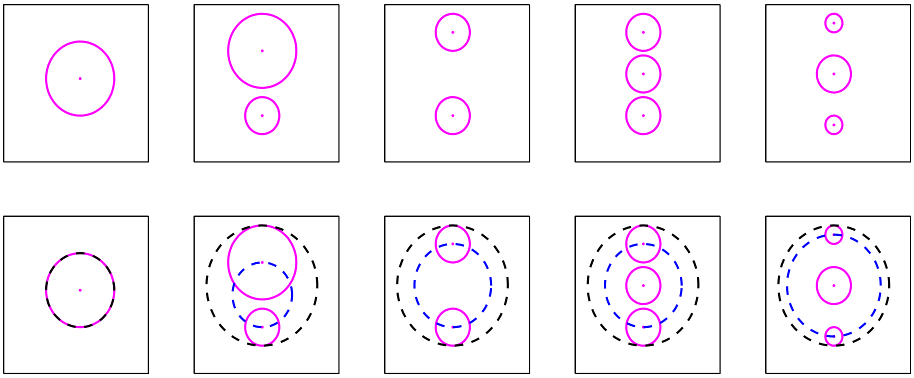

Fig. 3. Proposed circular features on example cases (best viewed in color): Given the fitted circles in the top row, we show the proposed circular features in the bottom row. For all examples, the number of fitted circles in each layer corresponds to the number of circles feature. The outer circle (black) is the minimum bounding circle, which includes all circles, and the inner circle (blue) is the circle bounding the centers of all the circles. In the first example, there is only one fitted circle, therefore the area of the outer circle is equal to the fitted circle, and there is no inner circle. In the second and third examples, the number of circles and the areas of the outer circles are the same. However, the areas of the inner circles are different because of the distance between the fitted circles. In the fourth and fifth examples, we present cases that include three body parts (number of circles). If we compare the third and fourth cases, we observe that the areas of the outer and inner circles are the same, while the number of circles are different. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

3.2. Circle-based pose representation

The proposed circular representation provides important local cues about the body configuration: the number of body parts passing through that layer, how much they spread over that layer and how far they are from each other. We utilize the following circular features to model these cues (see Fig. 3). The proposed features allow a simple and easy way to provide a view-independent representation.

3.2.1. Number of circles

The articulated structure of the body creates a different number of intersections in each layer. For example, the layer corresponding to the head is likely to have a single circle, and a layer corresponding to the legs is likely to have two circles. Therefore, as our first feature, we examine the number of circles to find the number of body parts intersecting with a layer.

In this representation, a pose is described by a vector **c**,

$$\mathbf{c} = [c_1, \ldots, c_n]^T \tag{1}$$

where each c_l, $1 \leqslant l \leqslant n$, is the number of circles extracted from layer l.

3.2.2. Area of outer circle

The maximum area covering all body parts passing through a layer is another important local cue to understand the body configuration in this layer, even in the case of noise. For example, the maximum area at the level of the legs can provide information such as whether the legs are open or closed.

To include all circles in that layer, the maximum area that covers all body parts in a layer is found by fitting a minimum bounding circle. We refer to this minimum bounding circle as the *outer circle*.

In this representation, a pose is described by a vector **o**,

$$\mathbf{o} = [o_1, \ldots, o_n]^T \tag{2}$$

where each o_l, $1 \leqslant l \leqslant n$, is the area of the outer circle in layer l.

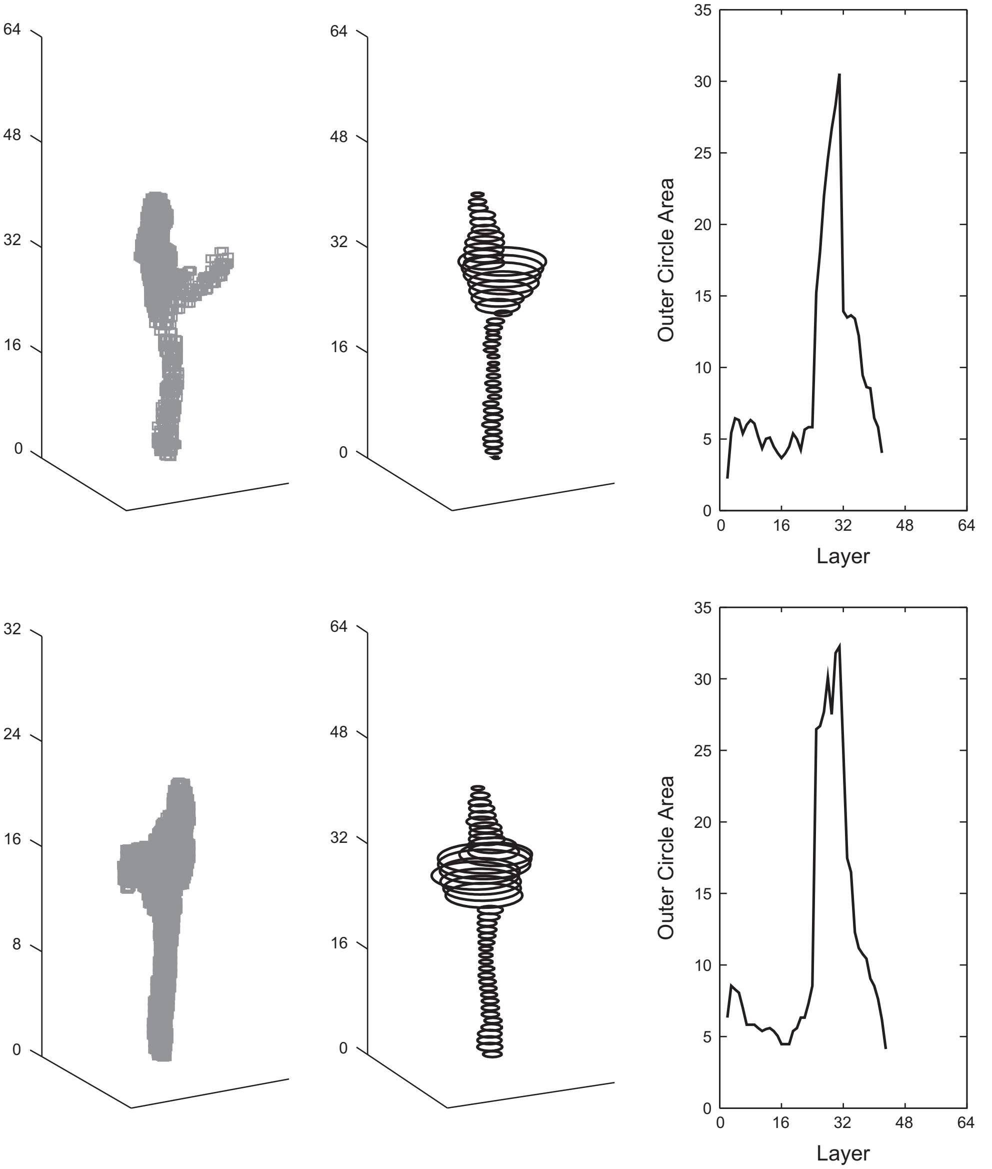

Fig. 4. View invariance: Two kick poses performed by two different actors. The left column is the volumetric data for a pose from the kick action (64 × 64 × 64 voxel grid). The middle column is the representation of the pose as a collection of bounding circles. The right column is the plot of the area of the outer circle's feature vector showing the bounding circle's area per layer. Note that the feature vectors of the same pose from different viewpoints are very similar.

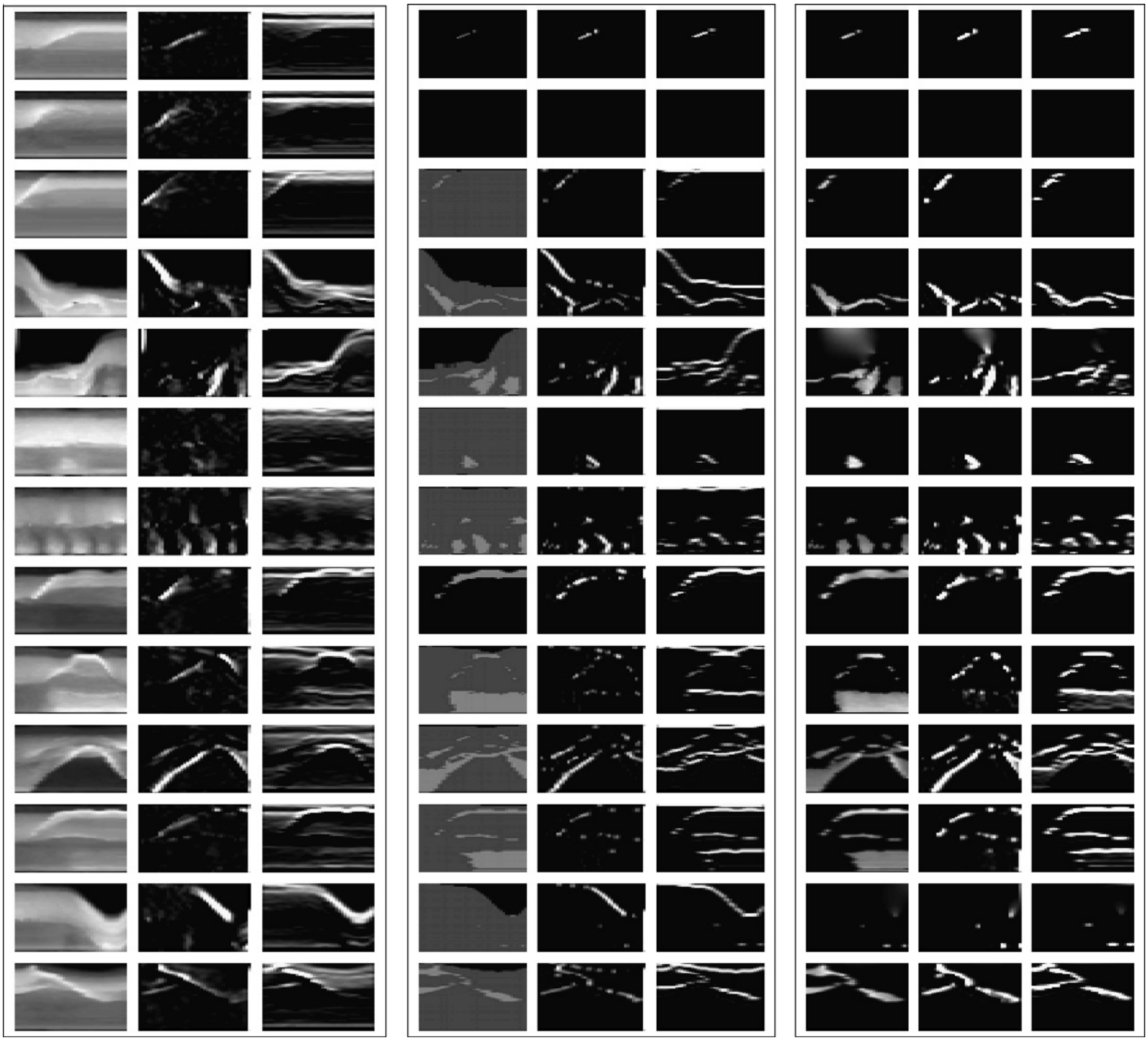

Fig. 5. *Motion Set* for 13 actions. From top to bottom: check watch, cross arms, scratch head, sit down, get up, turn around, walk, wave, punch, kick, point, pick up and throw. From left to right: $\mathbf{O}$, $\mathbf{O}_t$, $\mathbf{O}_b$, $\mathbf{C}$, $\mathbf{C}_t$, $\mathbf{C}_b$, $\mathbf{I}$, $\mathbf{I}_t$, $\mathbf{I}_b$.

3.2.3. Area of inner circle

The third feature encodes spaces in a pose volume and is represented by the relative distances between body parts. For this purpose, an inner circle bounding the centers of all circles is found. In this representation, a pose is described by a vector $\mathbf{i}$,

$$\mathbf{i} = [i_1, \ldots, i_n]^T \tag{3}$$

where each i_l, $1 \leqslant l \leqslant n$, is the area of the inner circle in layer l.

Note that rather than using the area of outer and inner circles, the radii could be used. The choice for area is made in order to amplify the differences. It is also empirically observed that area is better than radius.

For all feature vectors $\mathbf{c}$, $\mathbf{o}$, and $\mathbf{i}$, encoding starts in layer $l = 1$ corresponding to the bottom of the voxel grid, and a value of zero is stored for any layer with no pixels. This kind of encoding retains the order of the extracted features and the body location with respect to the floor (bottom of voxel grid).

3.3. Discussion on the proposed pose representation

The proposed circular features have several advantages. First, these features store discriminative information about the characteristics of an actual human pose. In most cases, poses are identified by the maximum extension of the body parts corresponding to the silhouette contours in the 2D scenarios. However, this information is usually lost in single-camera systems depending on relative actor orientation. When we have volumetric data, we can easily approximate this information by the bounding circles. Moreover, fitting free form circles helps us to solve pose

Table 1
Motion Set: the motion descriptor as a set of motion matrices.

Number of circles		
$\mathbf{C} = [\mathbf{c}^1, \ldots, \mathbf{c}^m]$	$\mathbf{C}_t = G_t \circ \mathbf{C}$	Variation of circle count in a fixed layer through time
	$\mathbf{C}_b = G_b \circ \mathbf{C}$	Variation of circle count through body
Area of outer circle		
$\mathbf{O} = [\mathbf{o}^1, \ldots, \mathbf{o}^m]$	$\mathbf{O}_t = G_t \circ \mathbf{O}$	Variation of outer circle in a fixed layer through time
	$\mathbf{O}_b = G_b \circ \mathbf{O}$	Variation of outer circle through body
Area of inner circle		
$\mathbf{I} = [\mathbf{i}^1, \ldots, \mathbf{i}^m]$	$\mathbf{I}_t = G_t \circ \mathbf{I}$	Variation of inner circle in a fixed layer through time
	$\mathbf{I}_b = G_b \circ \mathbf{I}$	Variation of inner circle through body

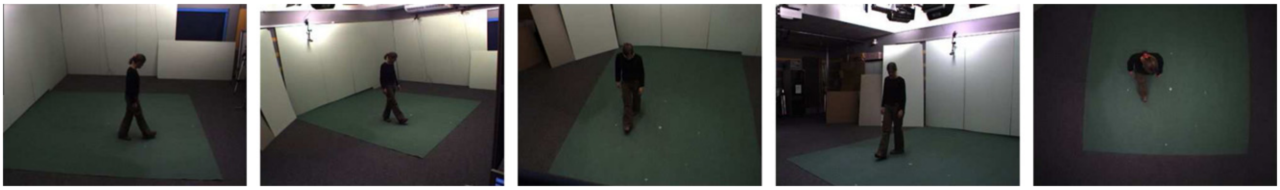

Fig. 6. Example views from the IXMAS dataset captured by five different synchronized cameras [23].

ambiguities related to the actor's style. It is robust for handling the variations that can occur in the performance of the same pose by different actors.

Second, our pose representation significantly reduces the encoding of 3D data while increasing the efficiency and preserving the effectiveness of the system. A circle can be represented with only two parameters, i.e. center point and radius, and the average number of circles per layer is approximately 2. This significant reduction in the number of voxels is further improved with the introduction of the vectors **c**, **o**, and **i**. Although the encoding is lossy, it is enough to identify poses and further actions when combined with the temporal variations.

Most importantly, the proposed features are robust for viewing changes. In a multi-camera system, an actor should be free to perform an action in any orientation and position. The system should thus be able to identify similar poses even in the case of viewpoint variations, that is, in the case of rotating a pose through the vertical axis. As the proposed circular features are extracted in each layer perpendicular to the vertical axis, any change in the orientation or position of the pose will result in the translation of the extracted circles but will not affect their areas (also radius), counts or relative positions. Therefore, the introduced three feature vectors **c**, **o**, and **i** hold pose information independently from the view and are robust to small changes in the poses. Fig. 4 shows an example in which the descriptors are very similar while the action is performed by two different actors and from two different viewpoints.

4. Motion representation

In the previous section, we present our representation to encode a 3D pose in a single action frame using feature vectors **c**, **o**, and **i**. In the following, we use the proposed pose vectors to encode human actions as motion matrices formed by concatenating pose descriptors in all action frames. Then we introduce additional motion features to measure variations in the body and temporal domains.

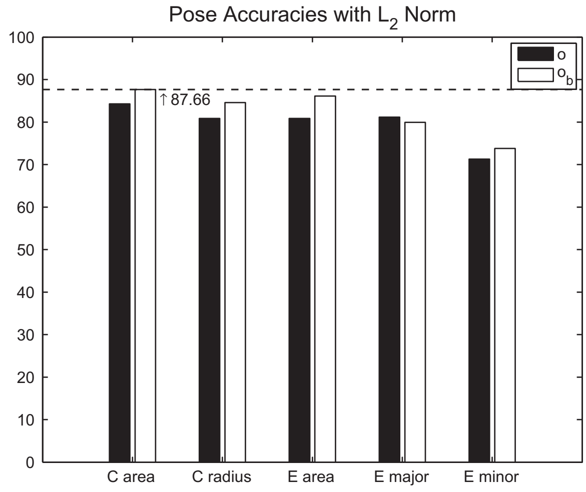

Fig. 7. Leave-one-out classification results for pose samples. Poses are classified using the L_2 norm with features **o** and $\mathbf{o_b}$. Results show performances of the **o** feature when computed using *Circle Area*, *Circle Radius*, *Ellipse Area*, *Ellipse Major Axis and Ellipse Minor Axis*, respectively.

4.1. Motion matrices

Let $\mathbf{p^t} = [p_1^t, \ldots, p_n^t]^T$ be the pose vector at frame t, where n is the number of layers, and p_l^t is any representation of the fitted circles in layer l. That is, $\mathbf{p^t}$ is one of the vectors **c**, **o**, or **i**. We define a matrix **P** as a collection of vectors $\mathbf{p^t}$, $\mathbf{P} = [\mathbf{p^1}, \ldots, \mathbf{p^m}]$, where m is the number of frames in an action period. Matrix **P** holds all poses during an action and can be visualized as an image of size $n \times m$, where n is the number of rows, and m is the number of columns (see Fig. 5).

A generic motion descriptor should be scale invariant in order to be independent from actor, gender, etc. In our study, rather than normalizing and scaling at the pose level, we apply them to motion matrices containing consecutive frames. In this way, we obtain a better understanding of body variation in terms of width and height through time.

Let **P** be any of the motion matrices described above with entries p_l^t, $l = 1, \ldots, n$ and $t = 1, \ldots, m$. First we obtain a normalized matrix entry pn_l^t as follows:

$$pn_l^t = \frac{p_l^t - min(\mathbf{P})}{max(\mathbf{P}) - min(\mathbf{P})} \tag{4}$$

where $min(\mathbf{P})$ and $max(\mathbf{P})$ are minimum and maximum values of matrix **P** respectively.

Then, we resize the **P** matrix by bilinear interpolation to an average size trained over samples. Resizing eliminates frame count and layer count differences among matrices, while preserving motion patterns.

Since we work on voxel data, the valuable information will be the voxel occupancy in a layer and its variation in amount and direction over time (corresponding to the velocity of the parts). Using matrix **P**, a specific action can be characterized with the changes in the circles over the body or over time. A single column stores information about the amount of change in the shape of the pose (body) at a given time t. Similarly, a single row stores information about the amount of change through the entire action period (time) for a given layer l.

In order to extract this information, we evaluate the differences on the features of the circles in consecutive layers and in consecutive frames. We apply a vertical differentiation operator G_b, as an approximation to vertical derivatives, over matrix **P** to find the change over the body, and refer to the resulting matrix as $\mathbf{P_b}$.

Table 2
Minimum, maximum and average sizes of all primitive actions after pruning.

	Layer count (n)	Frame count (m)
min	41	13
max	58	99
avg	≈48	≈40

Similarly, we convolve matrix $\mathbf{P}$ with a horizontal differentiation operator, G_t, to find the change over time, and refer to the resulting matrix as $\mathbf{P}_t$.

The two new matrices $\mathbf{P}_b$ and $\mathbf{P}_t$, together with the original matrix $\mathbf{P}$, are then used as the motion descriptors that encode variations of the circles over the body and time to be used in recognizing actions.

Remember that $\mathbf{P}$ can be created using any of the pose descriptors $\mathbf{c}$, $\mathbf{o}$, or $\mathbf{i}$. Therefore, a motion descriptor consists of the following set of matrices, which we refer to as *Motion Set*:

$$\mathcal{M} = \{\mathbf{C}, \mathbf{C}_t, \mathbf{C}_b, \mathbf{O}, \mathbf{O}_t, \mathbf{O}_b, \mathbf{I}, \mathbf{I}_t, \mathbf{I}_b\} \qquad (5)$$

where $\mathbf{C} = [\mathbf{c^1}, \ldots, \mathbf{c^m}]$ is the matrix generated using the number of circles feature, $\mathbf{O} = [\mathbf{o^1}, \ldots, \mathbf{o^m}]$ is the matrix generated using the area of the outer circle feature, $\mathbf{I} = [\mathbf{i^1}, \ldots, \mathbf{i^m}]$ is the matrix generated using the area of the inner circle feature, and the others are the matrices obtained by applying vertical and horizontal differentiation operators over these matrices. The summary of the motion descriptors is given in Table 1.

4.2. Implementation details for motion matrices

We perform some operations over matrix $\mathbf{P}$ for pruning and noise removal prior to the normalization and formation of new matrices $\mathbf{P}_b$ and $\mathbf{P}_t$.

After pose vector concatenation, matrix $\mathbf{P}$ is in the size of $64 \times m$, where 64 is the height of the voxel grid and m is the action period. However, some layers (rows) are not occupied by the body for the entire action. Therefore, first operation is to clean up the motion matrices from rows that are not used throughout the action period. After pruning, we obtain a $n \times m$ matrix, where n depends on the maximum height of the actor over all action frames. Similarly, action periods may vary from one actor to another, resulting in a different number of frames.

The provided volumetric data is obtained by shape from the silhouette technique. However, extracted silhouettes have some defects, affecting the volumes and our circular representation. After pruning, we perform interpolation over motion matrices $\mathbf{P}$ to enhance the circular representation and to fill gaps corresponding to missing circles. In some layers, missing voxels result in smaller circle sizes. We apply a spatio-temporal smoothing over the motion matrices to tackle this problem. If a circle in a layer has a smaller value than the average of its 4-neighbors then it is replaced with the average value.

5. Action recognition

With the introduction of the motion descriptors in the previous section, action recognition is turned into a problem of matching the *Motion Set* ($\mathcal{M}$) of the actions.

We present a two-level action classification scheme to improve the classification accuracy. In the first level, actions are classified in terms of global motion information by a simple approach that divides actions into two groups: upper-body actions such as punch or wave and full-body actions such as walk or kick. Then, in the second level, actions are classified into basic classes based on motion matrix similarities measured by Earth Mover's distance (EMD) or Euclidean distance.

5.1. Full-body vs. upper-body classification

A simple way to classify actions is to evaluate global motion information such as detecting whether the motion occurs in the upper or lower part of the human body. We observe that lower-body motions cause changes in the entire body configuration. We

Table 3
Leave-one-out nearest neighbor classification results for pose representation evaluated with bi o_b. Vectors are compared using the L_2 norm.

	Acc. (%)	Check watch	Cross arms	Scratch head	Sit	Walk	Wave	Point punch	Kick	Pick
C_{area}	87.66	80.56	88.89	94.44	100.00	100.00	77.78	80.56	77.78	88.89
E_{area}	86.2	86.11	100.00	77.78	100.00	97.22	66.67	63.89	83.33	100.00
E_{major}	79.94	61.11	63.89	77.78	100.00	100.00	75.00	83.33	72.22	86.11

Table 4
Stand-alone recognition performances of each matrix in *Motion Set* for 13 actions and 12 actors. The motion matrices are scaled to 48 × 40. Results denote that **O** outperforms others with Euclidean Distance and $\mathbf{O}_t$ is the best one for Earth Mover's Distance.

Acc. (%)	O	O_t	O_b	I	I_t	I_b	C	C_t	C_b
Euclidean	85.26	66.00	81.00	49.40	31.00	37.80	57.10	32.50	37.60
Earth Mover's	72.90	80.34	75.40	53.00	48.30	45.50	64.70	61.80	55.60

propose an action classification method that decides whether a test instance is a member of the full or upper body *Motion Sets*.

We represent each action with a feature that reveals the amount of motion in each layer during an action period. For this purpose, we calculate v_l, the variation of the outer circle area, for all layers of $n \times m$ matrix **O**, where n is the number of body layers, m is the length of the action period. Then, we train a binary Support Vector Machine (SVM) classifier using the defined feature vector, $\mathbf{v} = v_1, \ldots, v_n$, to learn upper-body and lower-body actions. SVM classifier is formed using RBF kernel.

As will be discussed later, the proposed two-level classification method increases the recognition performance. Moreover, the total running time decreases a considerable amount, parallel to the descending pairs to be compared.

5.2. Matching actions

After the first-level classification as an upper-body or full-body action, we perform a second, level classification. We calculate a *Motion Set*, $\mathcal{M}$ for each action and use it for comparison with other actions. For this purpose, we use the nearest neighbor classification.

Let us assume $\mathcal{M}^i$ and $\mathcal{M}^j$ are *Motion Sets* containing motion matrices for two action instances. Let $\mathbf{P}^i$ be one $n \times m$ matrix from set $\mathcal{M}^i$, where n is the layer count and m is the time. During the comparison of the two actions, we compute a global dissimilarity $D(\mathbf{P}^i, \mathbf{P}^j)$. We use the same methodology to compare all the matrices in the *Motion Sets*. Note that if a motion belongs to the upper-body set, we perform a comparison only for the upper body layers, $\frac{n}{2}, \ldots, n$.

5.3. Distance metrics

There are many distance metrics, such as L_p norms, Earth Mover's Distance [30], Diffusion Distance [31] and χ^2. L_p norms and χ^2 are affected by small changes in bin locations, whereas, EMD and Diffusion Distance alleviate shift effects in bin locations and in some cases outperform other distance metrics [31,32].

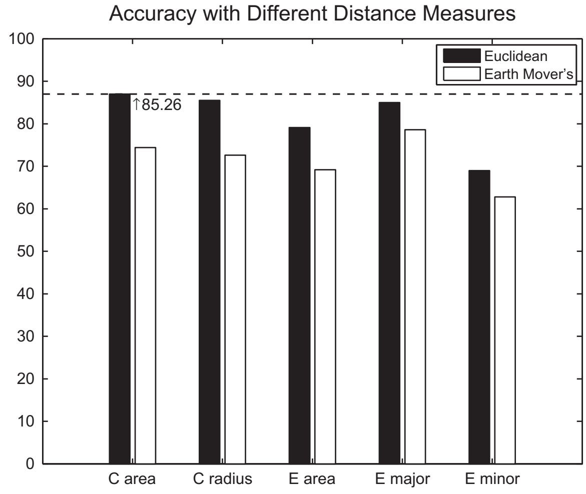

Fig. 8. Leave-one-out classification results for action samples. Actions are classified with the **O** feature using L_2 and *EMD*. We show performances of the **O** feature when computed using *Circle Area*, *Circle Radius*, *Ellipse Area*, *Ellipse Major Axis* and *Ellipse Minor Axis*, respectively.

We test two different distance measures. One is the well-known L_2 norm and the other is the shift-invariant EMD. Although EMD is robust for shift effects, we find that L_2 gives better performance than EMD in some cases.

Classical EMD algorithms have high $O(N^3 \log N)$ computational complexity. In this study, we apply EMD-L_1, which is a $O(N^2)$ EMD algorithm based on L_1 ground distance [32]. Furthermore, EMD-L_1 algorithm does not require normalization to the unit mass over the arrays. This increase its feasibility over our motion matrices, which are only scale normalized.

6. Experimental results

6.1. Dataset

We test our method on the publicly available IXMAS dataset [23], which is a benchmark dataset of various actions taken with multiple cameras. There are 13 actions in this dataset: check watch, cross arms, scratch head, sit down, get up, turn around, walk, wave, punch, kick, point, pick up and throw. Each action is performed three times with free orientation and position by 12 different actors. Multi-view videos are recorded by five synchronized and calibrated cameras. Example views from the camera setup are shown in Fig. 6. Since our focus is on pose and motion representation, we do not deal with reconstruction and use the available volumetric data in the form of a 64 × 64 × 64 voxel grid. For motion segmentation, we use the segmentations provided by Weinland et al. [23].

6.2. Evaluation of pose representation

In the first part of the experiments, we evaluate the performance of proposed pose representation. For this purpose, we

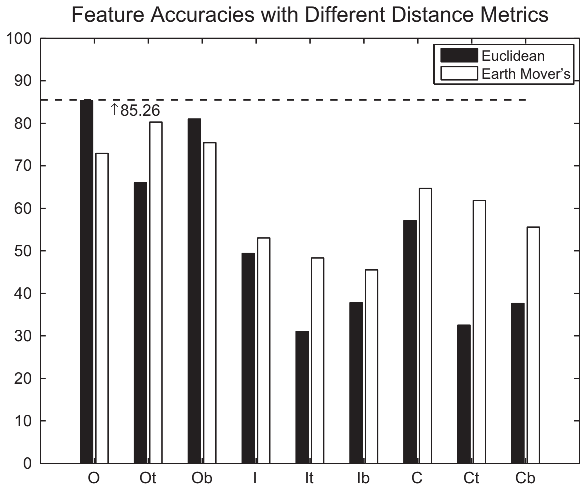

Fig. 9. Leave-one-out classification results for action samples. Actions are classified with various features using L_2 and *EMD*. Results denote that **O** outperforms others with Euclidean Distance and $\mathbf{O_t}$ has the highest accuracy for Earth Mover's Distance.

construct a key-pose dataset obtained from IXMAS. Among the dataset actions, sit down, stand up and pick up have identical key-poses, but in different order. Similarly, punch, point and throw actions can be grouped as identical in terms of key-poses. We construct a dataset of nine classes consisting of 324 key-poses in total; key-pose is selected from each action video of the dataset (see Table 3).

We measure pose retrieval performance by a simple nearest neighbor classifier using L_2 norm. We use $\mathbf{o}$ and $\mathbf{o_b}$ pose vectors, with variation through the body, as our feature. Experiments show that the $\mathbf{o_b}$ feature outperforms $\mathbf{o}$. Accuracies and class performances for the $\mathbf{o_b}$ feature are shown in Table 3.

While computing the $\mathbf{o}$ and $\mathbf{o_b}$ features, we follow a similar procedure as for motion matrices. First we obtain a pruned circular representation to remove unoccupied layers at the top and bottom of the visual hulls, then normalize it with the maximum value. Then, we resize all pose vectors to a fixed size of 48 using bilinear interpolation. Finally, we compute the $\mathbf{o_b}$ feature by applying a $[-1\,0\,1]$ filter over $\mathbf{o}$.

In addition to circle fitting, we perform experiments for a representation based on ellipses. We compute $\mathbf{o}$ vectors by ellipses rather than circles. For ellipses, we define three features that can be used instead of the circle area: ellipse area, ellipse major axis and ellipse minor axis. The results are illustrated in Fig. 7.

Similarly, we test the performance of the circle radius for pose retrieval. Although performances of all features are very close to each other, circle area outperforms others. Moreover, using circle area rather than circle radius provides a slightly better performance. Here, the scale factor reveals the differences among circles.

6.3. Evaluation of motion representation and action recognition

Each action is represented with a *Motion Set*, $\mathcal{M}$, where motion descriptors $\mathbf{C}$, $\mathbf{O}$, $\mathbf{I}$ are obtained by concatenation of pose descriptors. After enhancement, all $n \times m$ main motion matrices $\mathbf{C}$, $\mathbf{O}$, and $\mathbf{I}$ are resized to a fixed size matrix by bilinear interpolation. For the IXMAS dataset, considering the average, minimum and maximum layer and frame sizes for actions (shown in Table 2), matrices are scaled into a fixed 48×40 size matrix. Other values are also experimented with, but only slight changes are observed.

Other matrices in $\mathcal{M}$, i.e. $\mathbf{C}_b$, $\mathbf{C}_t$, $\mathbf{O}_b$, $\mathbf{O}_t$, $\mathbf{I}_b$, and $\mathbf{I}_t$, are obtained by applying the *Sobel operator* on matrices $\mathbf{C}$, $\mathbf{O}$, $\mathbf{I}$ to measure the circular variations through the body and time.

We perform a two-level action classification. In the first level, a simple binary SVM classifier is trained for full-body vs. upper-body classification. On the IXMAS dataset, we label check watch, cross arms, scratch head, wave, punch, point and throw actions as upper-body actions, and sit down, get up, turn around, walk, kick and pick up as full-body actions. We selected three actors for each action for training, and use the rest of the actors for testing. We obtain a classification accuracy of 99.22%, which is almost perfect.

In the second level, actions are compared within their corresponding upper-body or full-body sets using the nearest neighbor method with different distance metrics, L_2 norm and EMD-L_1. We use the leave-one-out strategy. All samples belonging to an actor are excluded, to be used as test samples, and the samples remaining are used as training samples. The accuracy results are obtained by averaging over all the actors.

In the first part of the experiments, we evaluate stand-alone recognition performances for each motion representation (see Fig. 9). The results indicate that the accuracy of matrix $\mathbf{O}_t$ is higher than the accuracies of all other motion matrices when EMD is used as the distance measure. In EMD experiments, motion matrices $\mathbf{O}_t$ and $\mathbf{O}_b$ outperform the original matrix $\mathbf{O}$. However, this is not true for matrices $\mathbf{C}$ and $\mathbf{I}$. For the Euclidean case, $\mathbf{O}$ outperforms the rest, with the highest 85.26% accuracy. But in this case, $\mathbf{O}_t$ has a lower performance than $\mathbf{O}$ and $\mathbf{O}_b$. Detailed accuracies are given in Table 4.

Similar to the pose experiments, we also evaluate the behavior of the most discriminative feature in the cases of different structures fitted to the voxels. For this purpose, we show performances of the $\mathbf{O}$ feature in Fig. 8 when computed using *Circle Area*, *Circle Radius*, *Ellipse Area*, *Ellipse Major Axis* and *Ellipse Minor Axis*, respectively, with different distance metrics.

In the second part of the experiments, we evaluate the recognition performance as a mixture of motion matrices to utilize different motion features at the same time. For this purpose, we combine two motion matrices from *Motion Set* using a linear weighting scheme. Stand-alone evaluation indicates that matrices $\mathbf{O}$ and $\mathbf{O}_t$ have the best performances for Euclidean and EMD, respectively. Therefore we choose them as the main features and add other features as pairwise combinations. We calculate the weighted sum of distances for two action sequences i and j as:

$$D(i,j) = \alpha D(\mathbf{O}^i, \mathbf{O}^j) + (1-\alpha) D(\mathbf{P}^i, \mathbf{P}^j) \qquad (6)$$

where $\mathbf{P} \in \mathcal{M}$ and $\mathbf{P} \neq \mathbf{O}$ ($\mathbf{P} \neq \mathbf{O_t}$ for EMD).

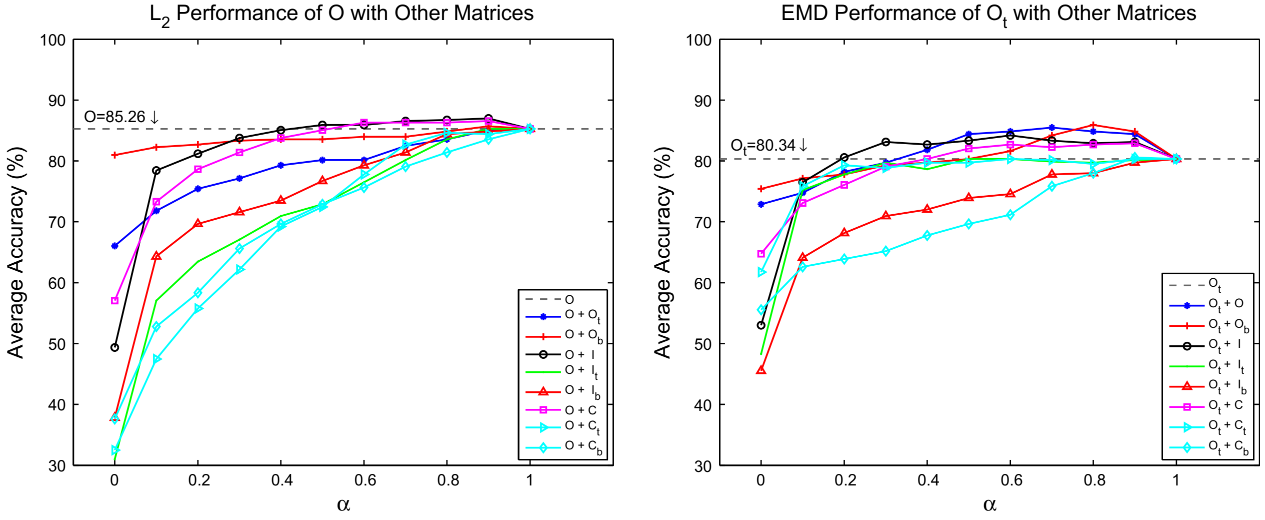

Fig. 10. α search for two-level classification: the combination of $\mathbf{O}$ matrix with others using L_2 and the combination of $\mathbf{O}_t$ matrix using *EMD*. $\mathbf{O}$ and $\mathbf{I}$ pair gives the highest accuracy (**86.97%**) with Euclidean Distance and $\alpha = 0.9$ value. $\mathbf{O}_t$ and $\mathbf{O}_b$ give the best performance, with $\alpha = 0.8$, resulting in an accuracy of **85.90%** for EMD measurement.

As shown in Fig. 10, $\mathbf{O}_t$ and $\mathbf{O}_b$ give the best performances with $\alpha = 0.8$, resulting in an accuracy of **85.90%** for EMD measurement. On the other hand, **O** and **I** pair gives the highest accuracy; **86.97%** with Euclidean Distance and $\alpha = 0.9$ value.

To further incorporate all features, we select one feature matrix per feature type that has the highest accuracy among three. We pick either **P**, $\mathbf{P}_t$ or $\mathbf{P}_b$ from each type. Based on the stand-alone performances, one matrix outperform others in the same feature type. This time, we calculate the weighted sum of distances with *alpha* and *beta* parameters. For Euclidean distance, we report a maximum accuracy of 88.63% with weights 0.7, 0.1 and 0.2 for the **O**, **C** and **I** combination, respectively. For the EMD experiment, we report 85.26% with same weights for $\mathbf{O_t}$, **C** and **I**, respectively.

Fig. 11 shows the confusion matrices for the best configurations. In the experiments, we observe no important difference in the performance with the addition of new features, and for simplicity we prefer to combine only three features.

The recognitions of three actions, namely, wave, point and throw, have lower performances with respect to other action categories. The main problem with the wave action is confusion with the scratch head action. In many wave sequences, even in the volumetric data, the periodicity of the wave action is rarely or never recognized. Some actors just barely move hand, while others move arms. From this observation, a low recognition rate is to be expected.

Similarly, the point and punch actions are confused with each other. Both actions contain similar poses, but a difference between punch and point is in the duration of the actions. In motion patterns, we observe high peak for punch actions. This peak can be a high value occurring in the same layer but with a short duration, or it can be an increase through the z axis (corresponding y axis in the motion matrix). After scaling, the distinctive pattern is preserved and peak variations among these two actions can be clearly observed. But again, depending on the actor's style, the duration is not consistent; some actors perform a punch action very slowly, and it looks like the point action. In such cases, it is likely that the scaling of the motion matrices results in the similar representation of these two action categories.

Another low recognition performance is observed for the throw action. The dataset contains variants of the throw action, performed by different actors, as throw from above and throw from below. Even in this situation, we achieve 80.56% accuracy for the

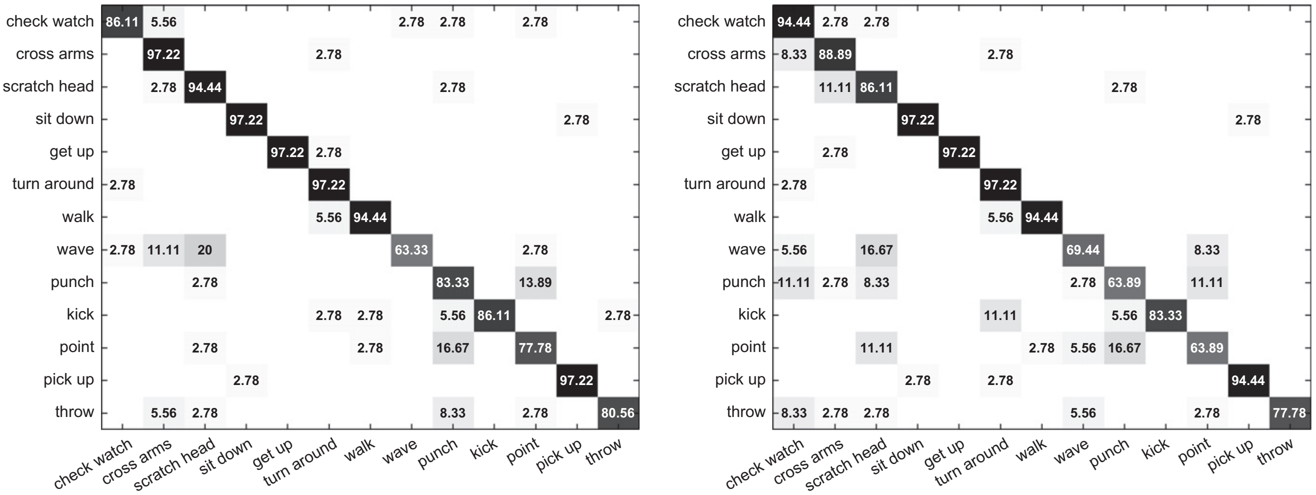

Fig. 11. Confusion matrices for two-level classification on 13 actions and 12 actors. The left confusion matrix is for Eucledian Distance, with an accuracy 88.63%, while the right one is the result of the EMD, with an accuracy of 85.26%. Results are computed by the sum of weighted distances over three feature matrices. Matrix size is 48 × 40.

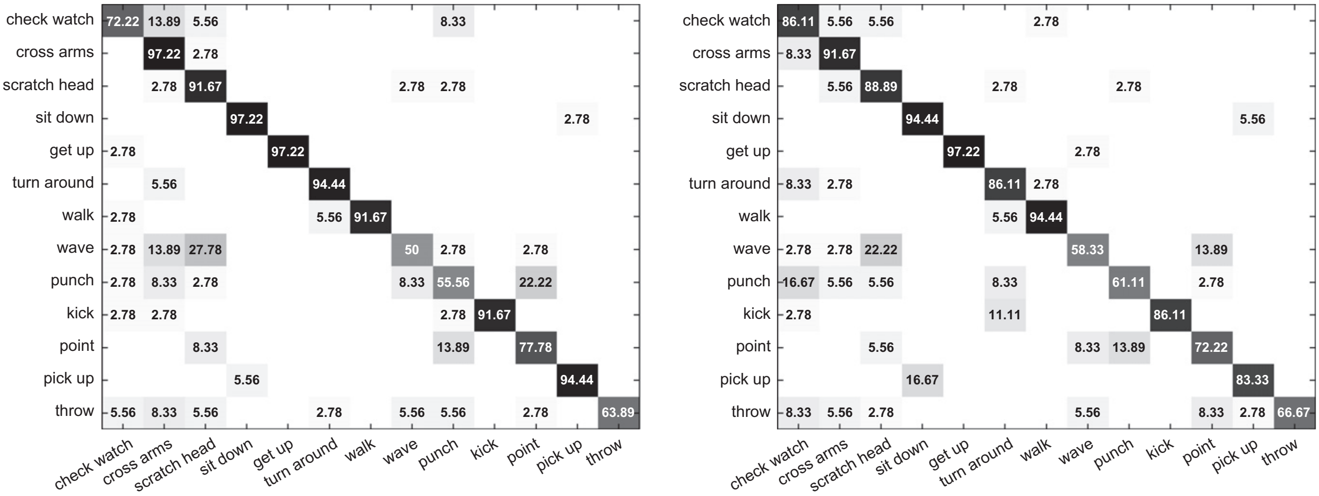

Fig. 12. Confusion matrices for single-level classification on 13 actions and 12 actors. The left confusion matrix is for Eucledian computation giving 82.69% accuracy by combining **O** and **I** with weights 0.8, 0.2. The right confusion matrix is the EMD computation reported 82.05% accuracy by combining **O** and **I** with weights 0.7, 0.3. Results are computed by the sum of weighted distances over three feature matrices. Matrix size is 48 × 40.

throw action, with fewer training samples in the same category. Aside from this, throw is mainly confused with punch action, because of similar peak patterns related to actor style.

In the third part of the experiments, we evaluate the system's performance on single-level classification (i.e. classifying a full set of actions without classifying them into full-body vs. upper-body actions). As in the two-level case, **O** and $\mathbf{O}_t$ outperform others when we compare their stand-alone performances. Similarly, we compute the distances among action samples by first selecting one feature from each feature type, and then combining the weighted sum of three distances. For the Euclidean experiment, the **O** feature again gives the highest accuracy 82.69% when combined with **I** using weights 0.8, 0.2. For EMD, we report an accuracy of 82.05% by combining **O** and **I** with weights 0.7, 0.3. Please note that both experiments are done by combining three feature types. However, **C** does not increase the results. The confusion matrices of the best reported results are given in Fig. 12.

We observe that adding a two-level classification improves the overall accuracy from 82.69% to 88.69%. Moreover, if execution time is crucial, this method decreases the running time by eliminating half of the comparisons between pairs for matching.

7. Comparisons with other studies

The IXMAS dataset is introduced in [23] and used in several other studies. In this section, we compare our results with the other studies in the literature that use the same dataset. Remember that we use the full dataset, with 13 actions and 12 actors, and obtain a **88.63%** recognition performance.

In some studies [23,24,27], not the full set, but a subset consisting of 11 actions and 10 actors is used. In order to compare our method with those studies, we also test our method over 11 actions and 10 actors. With a similar evaluation of different configurations as in the full case, we obtain a 98.7% performance for the upper-body vs. full-body classification, and for action recognition in a two-level classification scheme the best reported performance, **90.91%**, is obtained for the **O**, **I** and **C** combination with weights 0.7, 0.1, 0.2, respectively using Euclidean Distance. Similarly, we report a 90.30% recognition accuracy for the EMD experiment, for an $\mathbf{O}_t$, **I** and **C** combination with weights 0.7, 0.2, 0.1 respectively. The confusion matrix for 11 actions, and 10 actors is shown in Fig. 13.

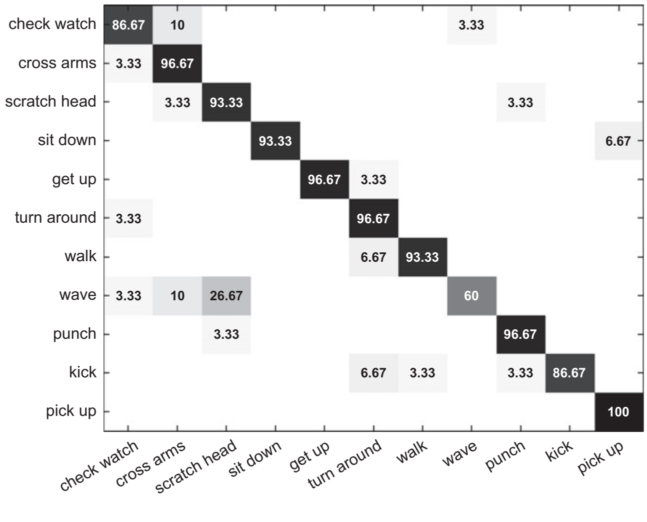

Fig. 13. Confusion matrix for two-level classification on 11 actions and 10 actors, with an accuracy of 90.91%. This is the result of **O**, **I** and **C** combination weights 0.7, 0.1, 0.2. Matrix size is 48 × 40.

Table 5
Comparison of our method with others tested on the IXMAS dataset.

Method	Accuracy (over 11 actions) (%)	Accuracy (over 13 actions) (%)
Weinland et al. [23]	93.33	–
Our method	*90.91*	*88.63*
Liu and Shah [29]	–	82.8
Weinland et al. [24]	81.27	–
Lv and Nevatia [25]	–	80.6
Yan et al. [27]	78.0	–

Although a direct comparison is difficult, we arrange studies in terms of performances on the multi-view IXMAS dataset in Table 5. As can be seen from the results, our performance is comparable to the best result in the literature and superior to the other studies in terms of the recognition rate. This is very promising considering the efficiency of the proposed encoding scheme.

Weinland et al. [23] report an accuracy of 93.33% over 11 actions. However, in their later work [24], they obtain 81.27% accuracy on the same dataset. This shows that 3D-based recognition is more robust than 2D to 3D matching. Lv and Nevatia [25] use 2D matching. Note that they test on the whole dataset, however they further add standing action, and divide throw action into two action categories as throw above and throw below. Liu and Shah [29] report the second-highest score over 13 actions. Their approach is based on multiple images features rather than 3D representations. However, it proposes an orderless feature set, which is difficult to use for pose estimation.

8. Summary and future directions

The principal contributions of this paper can be summarized as follows:

- First, we introduce a new layer-based pose representation using circular features. For this purpose, three view-invariant features are introduced to encode the circles in a layer, which are then combined over all layers to describe a pose on a single frame: the number of circles, the area of the outer circle that covers all body parts passing through a layer, and the area of the inner circle that encodes the distance between body parts in a layer. The study demonstrates that this circular model is effective for providing view invariance. Moreover, representation does not require alignment.
- Second, we introduce a new motion representation for describing actions based on our pose representation. Motion features are extracted from matrices constructed as a combination of pose features over all frames. The extracted motion features encode spatial–temporal neighborhood information in twofold: variations of circular features in consecutive layers of a pose and variations of circular features in consecutive poses through time.
- Finally, we use our motion representation to propose a two stage action recognition algorithm using two well-known distance measures: L_2 norm and EMD-L_1 "cross-bin" distance [32]. Experiments show that the performance of our method is comparable to the other methods in the literature.

Although we study volumetric data in the form of a voxel grid throughout the paper, the idea of circular features can be easily adapted to other 3D representations, such as surface points. The proposed descriptor can also achieve a good performance for shape matching and retrieval of human-body-like shapes. Moreover, the same approach can be used to fit a skeleton to a 3D shape by connecting the centers of circles in each layer.

Acknowledgments

We would like to thank Rana Nelson for proof reading. This work was partially supported by TUBITAK grant 104E065.

References

[1] Q. Cai, J.K. Aggarwal, Human motion analysis: a review, Journal of Computer Vision and Image Understanding 73 (1999) 428–440.

[2] W. Hu, T. Tan, L. Wang, S. Maybank, A survey on visual surveillance of object motion and behaviors, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 34 (2004) 334–352.

[3] D. Forsyth, O. Arikan, L. Ikemoto, J. OBrien, D. Ramanan, Computational studies of human motion I: tracking and animation, Foundations and Trends in Computer Graphics and Vision 1 (2006) 1–255.

[4] L.W. Campbell, D.A. Becker, A. Azarbayejani, A.F. Bobick, A. Pentland, Invariant features for 3-D gesture recognition, in: Automatic Face and Gesture Recognition, 1996.

[5] D.M. Gavrila, L.S. Davis, 3-D model-based tracking of humans in action: a multi-view approach, in: IEEE Conference on Computer Vision and Pattern Recognition, 1996.

[6] Y. Yacoob, M.J. Black, Parameterized modeling and recognition of activities, in: IEEE International Conference on Computer Vision, 1998.

[7] A. Bobick, J. Davis, The recognition of human movement using temporal templates, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2001) 257–267.

[8] C. Rao, A. Yilmaz, M. Shah, View-invariant representation and recognition of actions, International Journal of Computer Vision 50 (2002) 203–226.

[9] A. Efros, A. Berg, G. Mori, J. Malik, Recognizing action at a distance, in: IEEE International Conference on Computer Vision, 2003.

[10] M. Blank, L. Gorelick, E. Shechtman, M. Irani, R. Basri, Actions as space-time shapes, in: IEEE International Conference on Computer Vision, 2005.

[11] A. Yilmaz, M. Shah, Actions sketch: a novel action representation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2005.

[12] V. Parameswaran, R. Chellappa, Human action-recognition using mutual invariants, Computer Vision and Image Understanding 98 (2005) 294–324.

[13] T. Syeda-Mahmood, M. Vasilescu, S. Sethi, Recognizing action events from multiple viewpoints, in: IEEE Workshop on Detection and Recognition of Events in Video, 2001.

[14] I. Cohen, H. Li, Inference of human postures by classification of 3D human body shape, in: IEEE International Workshop on Analysis and Modeling of Faces and Gestures, 2003.

[15] K. Huang, M. Trivedi, 3D shape context based gesture analysis integrated with tracking using omni video array, in: IEEE Workshop on Vision for Human–Computer Interaction (V4HCI), in Conjunction with CVPR, 2005.

[16] S. Carlsson, J. Sullivan, Action Recognition by Shape Matching to Key Frames, Workshop on Models versus Exemplars in Computer Vision, 2001.

[17] G. Loy, J. Sullivan, S. Carlsson, Pose Based Clustering in Action Sequences, Workshop on Higher-level Knowledge in 3D Modeling and Motion Analysis, 2003.

[18] K. Schindler, L. Van Gool, Action snippets: how many frames does human action recognition require?, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008.

[19] M. Ankerst, G. Kastenmueller, H. Kriegel, T. Seidl, 3D shape histograms for similarity search and classification in spatial databases, Lecture Notes in Computer Science (1999) 207–228.

[20] A. Johnson, M. Hebert, Using spin images for efficient object recognition in cluttered 3D scenes, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (1999) 433–449.

[21] M. Kazhdan, T. Funkhouser, S. Rusinkiewicz, Rotation invariant spherical harmonic representation of 3D shape descriptors, in: Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, 2003.

[22] C. Canton-Ferrer, J.R. Casas, M. Pardas, Human model and motion based 3D action recognition in multiple view scenarios, in: European Signal Processing Conference, 2006.

[23] D. Weinland, R. Ronfard, E. Boyer, Free viewpoint action recognition using motion history volumes, Computer Vision and Image Understanding 104 (2006) 249–257.

[24] D. Weinland, E. Boyer, R. Ronfard, Action recognition from arbitrary views using 3D exemplars, in: IEEE International Conference on Computer Vision, 2007.

[25] F. Lv, R. Nevatia, Single view human action recognition using key pose matching and viterbi path searching, in: IEEE Conference on Computer Vision and Pattern Recognition, 2007.

[26] N. İkizler, D.A. Forsyth, Searching for complex human activities with no visual examples, International Journal of Computer Vision 80 (2008) 337–357.

[27] P. Yan, S. Khan, M. Shah, Learning 4D action feature models for arbitrary view action recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008.

[28] R. Souvenir, J. Babbs, Learning the viewpoint manifold for action recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008.

[29] J. Liu, M. Shah, Learning human actions via information maximization, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008.

[30] Y. Rubner, C. Tomasi, L. Guibas, The earth mover's distance as a metric for image retrieval, International Journal of Computer Vision 40 (2000) 99–121.

[31] H. Ling, K. Okada, Diffusion distance for histogram comparison, in: IEEE Conference on Computer Vision and Pattern Recognition, 2006.

[32] H. Ling, K. Okada, An efficient earth mover's distance algorithm for robust histogram comparison, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (2007) 840–853.

[33] R. Kehl, L.V. Gool, Markerless tracking of complex human motions from multiple views, Computer Vision and Image Understanding 104 (2006) 190–209.

[34] K.M. Cheung, T. Kanade, J.Y. Bouguet, M. Holler, A real time system for robust 3D voxel reconstruction of human motions, in: IEEE Conference on Computer Vision and Pattern Recognition, 2000.