HUMAN ACTION RECOGNITION IN 3D MOTION SEQUENCES

Konstantinos Kelgeorgiadis, Nikos Nikolaidis

Artificial Intelligence and Information Analysis Laboratory
Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece

ABSTRACT

In this paper we propose a method for learning and recognizing human actions on dynamic binary volumetric (voxel-based) or 3D mesh movement data. The orientation of the human body in each 3D posture is estimated by detecting its feet and this information is used to orient all postures in a consistent manner. K-means is applied on the 3D postures space of the training data to discover characteristic movement patterns namely 3D dynemes. Subsequently, fuzzy vector quantization (FVQ) is utilized to represent each 3D posture in the 3D dynemes space and then information from all time instances is combined to represent the entire action sequence. Linear discriminant analysis (LDA) is then applied. The actual classification step utilizes support vector machines (SVM). Results on a 3D action database verified that the method can achieve good performance.

Index Terms— human activity recognition, 3D data

1. INTRODUCTION

Human activity recognition is a research area that has attracted the attention of numerous researchers during the last fifteen years [1]. Activity recognition is usually applied on video data and is a challenging task. Its applications include automatic annotation and semantic description of video data for indexing, organization and other purposes, human-computer interaction, detection of dangerous situations in surveillance setups, etc. Activity recognition on 3D human movement data i.e., dynamic volumetric (voxel-based) or 3D mesh data has been rarely studied [2], [3], [4], [5], [6] although, it can have important applications similar to those of video-based activity recognition, such as semantic annotation of animation sequences for summarization, indexing and browsing or human-machine interaction in setups where acquisition of such data is feasible. In addition, since video feeds from multi-view camera setups are frequently used to generate 3D movement data, such methods can be used to achieve activity recognition on multi-camera environments. Motion capture data is the only type of 3D data that attracted a significant amount of activity recognition research [7], [8], [9], [10]. Activity recognition can be formulated as a problem of classification of time varying data, usually involving the matching of a movement data sequence (video / 3D) with a set of already labelled reference sequences. Thus, action recognition involves in many cases two distinct phases: the training phase and the actual recognition (recall) phase.

This paper presents a method that operates on dynamic binary volumetric (voxel-based) or 3D mesh movement data. An important advantage of the method is that it does not require temporal segmentation of the sequences into atomic actions, e.g. steps. In addition, the utilized action representation makes the method fairly insensitive to speed changes or changes in the action start and end point.

2. METHOD DESCRIPTION

The method assumes that each single-person action, belonging to one of R distinct action classes, is represented as a sequence of binary volumes (binary 3D voxel arrays) or 3D postures. In case 3D body posture data are provided as 3D meshes, a voxelization procedure can be applied in order to transform them to volumetric representation. Since the proposed method is not invariant with respect to body orientation, its orientation within each volume should be derived. A method based on the orientation of the feet is used for this purpose.

2.1. Estimating body orientation

The human feet can provide body orientation information since each foot has a primary axis (major elongation) and the bisector of the angle formed by the primary axes of the two feet, can usually be used to define the human body forward direction i.e., its orientation is very close to the orientation of the vector that is perpendicular to the human chest.

Thus, we can assume that, by localizing each foot in the binary volume of the body and deriving its primary axis we can approximate the body orientation, namely its forward direction. However, in order to do so we have to derive not only the primary axis of each foot but also its forward direction, i.e. we have to come up with a vector. The sum of these vectors can then provide the body forward direction. An algorithm for deriving these vectors is presented below.

Initialization: Let N to be the maximum expected volume (in voxels) of a foot up to the height of the ankle and d be a small portion of this volume e.g., $d = N/4$.

Step 1, feet localization: Crop and keep the lower part of the volume (Fig. 2-(2)), i.e., the part that corresponds to $z < K$, z being the vertical direction (height). Begin to scan the cropped volume in a bottom to top fashion and, when a foreground (body) voxel is found, apply a bottom-to-top floodfill algorithm, collecting the filled voxels into a new list, L_i, and removing them from the volume. By doing so we obtain D lists of voxels that correspond to D disjoint objects in the cropped volume. The type of the floodfill ensures that the collected voxels are sufficiently sorted according to their height without the need to apply a sorting algorithm. Discard the lists of voxels (objects) that contain less that V_T voxels, where V_T is a threshold used to separate body parts from noise. D' objects are retained after this step. From each remaining list of voxels, $L_i, i = 1, \ldots, D'$, select the first N voxels (green area B in Fig. 1) and apply PCA to them in order to find the primary axis, $\bar{V}_1$ (Fig.1) and the variance along the primary and secondary axis, $\sigma^2_{(N,pr)}$ and $\sigma^2_{(N,sec)}$. If $\sigma^2_{(N,pr)}/\sigma^2_{(N,sec)} < \sigma^2_v = 1.5$ we discard this object as being not oblong enough and thus not corresponding to a foot. Since

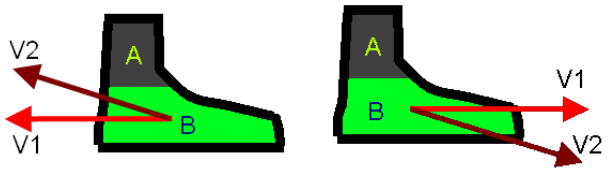

Fig. 1. Foot direction estimation procedure.

we expect to find two feet, if more than two objects are present after the above rejection procedures we keep only two of them (Fig. 2-(4)) by discarding the ones that reside at higher heights.

Step 2, estimation of feet direction: For each of the two remaining lists of voxels, $L_i, i = 1, 2$, select the first $N + d$ voxels (green and gray area A+B in Fig. 1) and, again, apply PCA to them in order to find the new primary axis, $\bar{V}_2$. Check the value of the inner product $\bar{V}_1 \cdot \bar{V}_2$. If $\bar{V}_1 \cdot \bar{V}_2 < 0$, i.e., if the vectors are forming an obtuse angle, set $\bar{V}_2 = -\bar{V}_2$ to make the two vectors point towards the same direction. Assuming that the vectors $\bar{V}_1 = (v_{1x}, v_{1y}, v_{1z})$ and $\bar{V}_2 = (v_{2x}, v_{2y}, v_{2z})$ are normalized, if $u_{1z} < u_{2z}$, then $\bar{V}_{Fi} = -\bar{V}_1$, else $\bar{V}_{Fi} = \bar{V}_1$ where $\bar{V}_{Fi}$ is the vector representing the foot forward direction. This is because, if voxels in area A, when added to voxels in area B, cause the primary vector to rotate downwards, the new voxels are placed in a location oposite the direction of the primary vector (see Fig.1). Conversely, if the additional voxels cause the primary vector to rotate upwards, the new voxels are placed in a location towards the direction the primary vector points to.

Step 3, estimation of the body orientation: Set $\bar{V}_{final} = \bar{V}_{F1} + \bar{V}_{F2}$. The orientation of the subject can be estimated by projecting the vector $\bar{V}_{final}$ on the plane $z = 0$.

This algorithm has been found to provide orientation estimates that are accurate enough for our purpose when 3D data of moderate/good quality are provided. When applied on the 3D postures sequence of an action, it provides a sequence of angles that indicate the direction of the human body at each time instance. Using this sequence, the body can be rotated in each instance around the z-axis so that it has a consistent orientation (e.g., heading towards 0^0) in all postures and all action sequences. In action sequences where body orientation does not change significantly from one time instance to the next, one can obtain the average orientation over time and orient all postures towards this direction.

The proposed body orientation estimation method can fail under certain circumstances. Indeed, subjects having their legs tight together can cause the algorithm to fail in step 1. Moreover, when the method is applied on subjects wearing loose clothes such as wide trousers it generates wrong orientation estimates. This problem occurred on one of the subjects in our dataset (see Section 3). Obviously, the precision of the orientation estimation depends on the resolution of the binary pose volume. Volumes with poor resolution provide poor orientation estimates.

2.2. Normalizing and centering the volumes

After consistently orienting the human body within each volume, normalization and centering take place. All the resulting volumes in each action are processed so that the centroid of the subject is always on the center of the volume. After centering all volumes with respect to their centroid, we normalize them by uniformly subsampling the initial volume so that, while centered, the subject occupies the maximum space possible in a volume of specific dimensions (e.g., $64 \times 64 \times 64$ voxels) without any part of it being clipped. Some normalized 3D postures are presented in Fig.3.

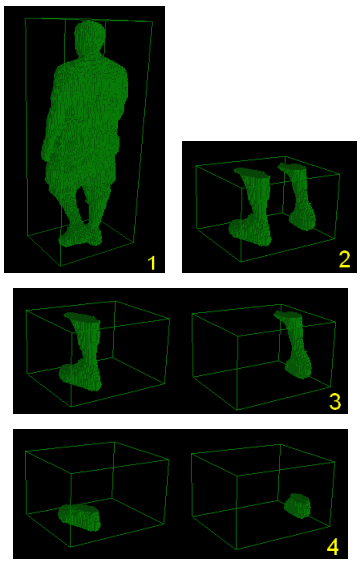

Fig. 2. The voxel data on different stages of the orientation estimation algorithm.

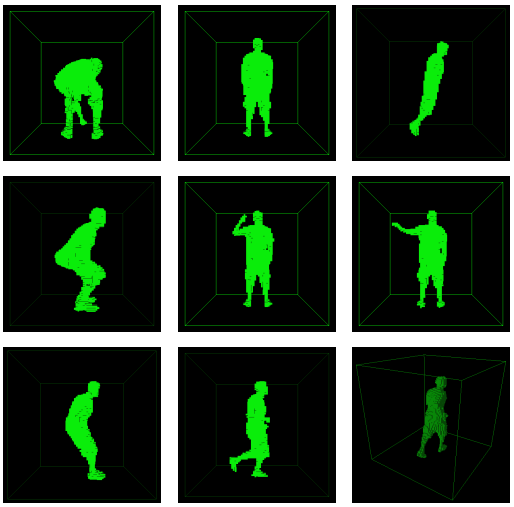

Fig. 3. Sample body centroid-centered and normalized 3D postures ($64 \times 64 \times 64$ voxels).

2.3. Dyneme extraction

Similar to the method in [11], the proposed method is based on clustering the 3D postures in the training set in order to come up with characteristic patterns, called dynemes, or, in our case, 3D dynemes. Indeed, in order to identify characteristic posture groups from the training set and extract cluster prototypes, K-Means was applied on all 3D postures of all actions in the training set. Other clustering al-

gorithms, like Fuzzy C-Means (FCM), can be used for this step but K-Means provided the best results. Each 3D volume (3D posture) is transcribed into a vector called *posture vector*, $\mathbf{x_i}$ and the K-Means algorithm is applied on these vectors. The number of K-Means clusters that need to be used for the method to work efficiently depends on the number of actions R that are to be recognized, the different ways an action can be performed by different people, the different body types, etc and it is usually selected empirically. At the end of the clustering procedure the space of all 3D postures in the training set is partitioned into a number of clusters, each containing similar postures. Let $\mathbf{v_c}, c = 1, \ldots, C$ be the extracted centers of the clustering algorithm. Each such center corresponds to the average of all postures in the cluster, and represents one dyneme. Some sample dynemes are presented in Fig.4. It should be noted that since each dyneme is the result of averaging, it no longer represents a binary volume.

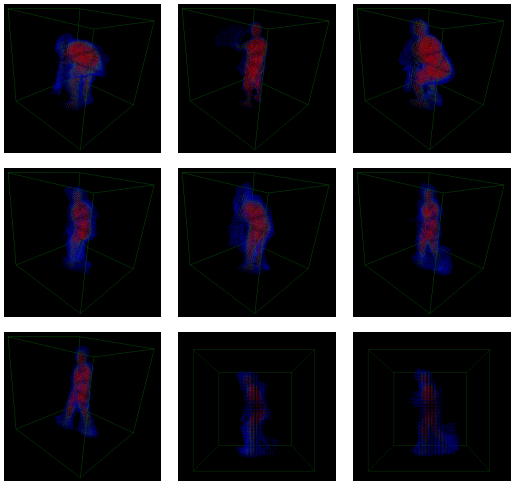

Fig. 4. Sample dynemes extracted by K-Means. The blue areas are the ones with low intensity and the red areas are the ones with high intensity.

2.4. Mapping postures and actions to the dyneme space

Using the centers (dynemes) acquired from the clustering algorithm of the previous step, we now map all the vectorized postures (posture vectors), $\mathbf{x_i}$, in the training set into the new space spanned by the dynemes, namely the dyneme space. To do so we create for each $\mathbf{x_i}$ a "membership" vector as follows:

$$\mathbf{u_i} = [u_{1,i}, \ldots, u_{C,i}]^T \tag{1}$$

$$u_{c,i} = \left(\|\mathbf{x_i} - \mathbf{v_c}\|_2\right)^{2/(1-m)}$$

m being the fuzzification parameter ($m > 1$). Subsequently, we normalize this vector to obtain the final representation of the posture in the dyneme space:

$$\phi_\mathbf{i} = \frac{\mathbf{u_i}}{\|\mathbf{u_i}\|_1} \tag{2}$$

Next, we calculate representations for each action in the dyneme space by simply averaging the normalized membership vectors of all postures of this action:

$$\mathbf{s} = \frac{1}{L}\sum_{i=1}^{L}\phi_\mathbf{i} \tag{3}$$

It is obvious that such an action representation retains essentially no information regarding the temporal succession of the various postures within the action, its length, speed and start/end points, making it advantageous for recognition purposes, despite its apparent simplicity.

2.5. LDA projection

In order to reduce the dimensionality of the action representations generated in the previous step and, at the same time, keep them as discriminant as possible, we utilize Linear Discriminant Analysis (LDA). LDA aims at calculating a linear projection matrix $\mathbf{\Psi_{opt}^T} \in \Re^{C\times R-1}$ (R being the number of actions we wish to recognize) so that:

$$\mathbf{\Psi_{opt}} = \underset{\mathbf{\Psi}}{argmax}(\mathbf{J_{LDA}}\left(\mathbf{\Psi}\right)) \tag{4}$$

$$\mathbf{J_{LDA}}\left(\mathbf{\Psi}\right) = \left|\mathbf{\Psi^T S_b \Psi}\right| / \left|\mathbf{\Psi^T S_w \Psi}\right| \tag{5}$$

with $\mathbf{S_b}, \mathbf{S_w} \in \Re^{C\times C}$ being the between and the within class scatter matrices, respectively.

After calculating the projection matrix we project action representations $\mathbf{s}$ (Eq. 3) thus reducing their dimension to $R-1$. Thus, the final representation of an action will be:

$$\mathbf{y_n} = \mathbf{\Psi_{opt}^T s_n} \tag{6}$$

2.6. Classification

In order to classify a 3D posture (volume) sequence depicting an unknown action to one of the actions the algorithm has been trained to recognize, we first consistently orient the 3D postures of the sequence and then normalize, center and sub-sample them to produce volumes of the same dimension to the ones used for the training. Subsequently, we use the centers (dynemes) of the K-Means algorithm that were extracted from the training data in order to map each posture in the test sequence into the dyneme space using Eq. (1),(2). Then, we calculate the representation of the entire sequence in the dyneme space using Eq. (3), project the representation using Eq. (6) and the projection matrix $\mathbf{\Psi_{opt}}$ evaluated from the training data and, finally, proceed into the classification using the resulting vector, τ on an SVM classifier. The projected representations of the action sequences that belong to the training set obtained by Eq. (6) were used to train a classifier.

3. EXPERIMENTAL RESULTS

The proposed method has been tested on the i3DPost multi-view and 3D human action/interaction database [12]. This database has been created by using a convergent setup of eight synchronized and calibrated cameras in order to produce high definition multi-view videos. In each multi-view video, one of the eight persons in the database is performing one of the seven actions defined in

the database namely "*walk*" (wk), "*run*" (rn), "*jump forward*" (jf), "*jump in place*" (jp), "*bend*" (bd), "*one hand wave*" (wv) and "*sit*" (st), which is actually a combined action consisting of "sit down" and "stand up". The subjects have different body sizes, clothing and are of different sex and nationalities. The multi-view videos have been used to construct a 3D mesh at each frame, describing the respective 3D human body surface (see [12] for more details).

The volumes on which the proposed technique was applied were obtained by converting the mesh data of the database into binary volumes (see [13]) of dimensions $256 \times 256 \times 256$ voxels. Initially, all action sequences in the database were oriented properly using the orientation estimation technique presented in Section 2.1. Since, in all sequences of the database, the person doesn't change his/her direction of motion within each action, it was decided to consistently orient the bodies by applying a single rotation to all postures in each sequence. The angle of this rotation was the mean of all body orientation angles in the sequence. Next, the postures were centered with respect to their centroid and normalized to fit into a $64 \times 64 \times 64$ volume, as described in the previous Section. Since the dataset was relatively small, a Leave One Out Cross Validation (LOOCV) procedure was deployed to infer the correct recognition rate of the proposed method. The summary of this LOOCV procedure is given in Fig.5.

- Set N_S to be the number of subjects in the database. Set N_A to be the number of actions in the database.
- Select a value for C (the number of centers used in K-Means).
- **For** $s = 1$ to N_S, **For** $a = 1$ to N_A,
  - Select the sequence of subject s performing action a to be the test set and all remaining sequences to be the training set.

  TRAINING
  - Compute the dyneme vectors, $\mathbf{v_c}, c = 1, \ldots, C$, from the training set using K-Means.
  - Compute the movement representation in the dyneme space for each sequence in the training set – Eq. (1)(2)(3).
  - Calculate the LDA matrix, M_{LDA}, and apply LDA projection – Eq. (6).

  TESTING
  - Map each posture vector in the test sequence into the dyneme space using $\mathbf{v_c}, c = 1, \ldots, C$ (the dyneme vectors obtained from the training phase) and compute its movement representation – Eq. (1)(2)(3).
  - Apply LDA projection using M_{LDA} obtained from the training phase – Eq. (6).
  - Classify the action using an SVM trained on the projected representations of the actions of the training data ($\mathbf{y_n}$).
- **End For** (a), **End For** (s)

Fig. 5. The LOOCV procedure.

After a number of trials the best configuration for the SVM was proven to be a linear kernel with $C_{SVM} = 10^5$. The number of centers of the K-Means algorithm (dynemes) that produced the best results was determined to be $C = 23$. The fuzzification parameter was selected to be $m = 1.33$. The overall correct classification rate was equal to 80.3%. The classification results in the form of the confusion matrix are given in Table 1. It should be noted however that some classification errors were due to the fact that one subject was wearing loose trousers that led to errors in the body orientation procedure (the orientations of the other subjects were correctly estimated).

Table 1. Confusion matrix (80.3% correct classification rate).

		Recognised Action						
Real Action		wk	rn	jf	bd	wv	jp	st
	wk	6	1	1	0	0	0	0
	rn	0	8	0	0	0	0	0
	jf	0	0	8	0	0	0	0
	bd	0	0	2	6	0	0	0
	wv	0	0	1	0	3	4	0
	jp	0	0	2	0	0	6	0
	st	0	0	0	0	0	0	8

4. CONCLUSIONS

In this paper a method for learning and recognizing human actions on dynamic binary volumetric (voxel-based) or 3D mesh movement data has been presented. The method was shown to achieve good recognition performance, without requiring temporal segmentation of the sequences into atomic actions, e.g. steps, or alignment of the start and end points to specific key body postures. Future plans include testing the method in larger datasets.

REFERENCES

[1] R. Chellappa, A. K. Roy-Chowdhury, and S. K. Zhou, "Recognition of humans and their activities using video," *Morgan and Claypool*, 2005.

[2] M.B. Holte, Cuong Tran, M.M. Trivedi, and T.B. Moeslund, "Human pose estimation and activity recognition from multi-view videos: Comparative explorations of recent developments," *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 5, pp. 538–552, Sept 2012.

[3] Pingkun Yan, Saad Khan, and Mubarak Shah, "Learning 4D action feature models for arbitrary view action recognition," in *In Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2008.

[4] Daniel Weinland, Edmond Boyer, and Remi Ronfard, "Action recognition from arbitrary views using 3D exemplars," in *Proceedings of the International Conference on Computer Vision*, 2007.

[5] Daniel Weinland, Remi Ronfard, and Edmond Boyer, "Free viewpoint action recognition using motion history volumes," *Computer Vision and Image Understanding*, July 2006.

[6] Cristian Canton-Ferrer, Josep R. Casas, and Montse Pardas, "Human model and motion based 3D action recognition in multiple view scenarios," *European Signal Processing Conference (EUSIPCO)*, September 2006.

[7] Ferda Ofli, Rizwan Chaudhry, Gregorij Kurillo, Ren Vidal, and Ruzena Bajcsy, "Sequence of the most informative joints (smij): A new representation for human skeletal action recognition," *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 24 – 38, 2014.

[8] Ioannis Kapsouras and Nikos Nikolaidis, "Action recognition on motion capture data using a dynemes and forward differences representation," *Journal of Visual Communication and Image Representation*, 2014, in press.

[9] Eshed Ohn-Bar and Mohan M. Trivedi, "Joint angles similarities and HOG2 for action recognition," in *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 465–470.

[10] Liqun Deng, Howard Leung, Naijie Gu, and Yang Yang, "Generalized model-based human motion recognition with body partition index maps," *Computer Graphics Forum*, vol. 31, no. 1, pp. 202–215, 2012.

[11] Nikolaos Gkalelis, Anastasios Tefas, and Ioannis Pitas, "Combining fuzzy vector quantization with linear discriminant analysis for continuous human movement recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 11, November 2008.

[12] Nikolaos Gkalelis, Hansung Kim, Adrian Hilton, Nikos Nikolaidis, and Ioannis Pitas, "The i3dpost multi-view and 3d human action/interaction database," in *IEEE Conference for Visual Media Production*, 2009.

[13] Fakir Nooruddin and Greg Turk, "Simplification and repair of polygonal models using volumetric techniques," *IEEE Trans. on Visualization and Computer Graphics*, vol. 9, no. 2, pp. 191–205, April 2003.