A random walk through human behavior,

Youssef Chahir*[a], Youssef Zinbi[a], Mahmoud Ghoniem[a] , Abder Elmoataz[a]
[a]GREYC - CNRS UMR 6072, Computer Science Department. University of Caen, Bd Maréchal Juin, BP 5186, 14032 Caen , France

ABSTRACT

In many applications, such as the video monitoring, the archiving and the video indexing, it is significant to recognize the movements of the people to be able to interpret their behaviors. This recognition of activity requires the extraction of multiple data, the automatic interpretation of image sequences, and called upon techniques of video analysis and techniques of data analysis and data classification. A human action being strongly related to the movement, we propose in this paper, an approach for tracking people to form a volume in 3D space (2d+t). This volume which represents a given action will be characterized by 3D geometrical moments which are invariants with the translation and the scaling.

In this article, we present a new approach of people actions categorization based on the Markov random walks on graph. The basic idea is to regard the whole of the actions (videos) as a weighted graph G=(V,E). This graph is defined as a set of vertices V which are represented by 3D volumes of the action, and a set of edges E which represent the similarity between actions.. This similarity will be calculated by an Euclidean distance between the vectors characteristic of the actions. Then, we will describe the implementation of our approach and we show results of validation on a corpus of actions which represent different actions of several people.

Keywords: Random walk, Nystrom, human action, 3D geometrical moments, graph cut, video

1. INTRODUCTION

Recognizing of Human Motion Actions from videos is a challenging re-search problem in computer vision; it is a key component in many computer vision applications, such as video surveillance, human-computer interface, video indexing and browsing, recognition of gestures, analysis of sports events, and dance choreography. It is of relevance to both the scientific and industrial communities. Recent works in the computer vision literature have proposed a number of successful motion recognition approaches based on nonlinear manifold learning techniques. Despite significant recent developments, general human motion recognition is still an open problem. This problem of identification becomes crucial when one has an increasing number individuals under various points of view of cameras, and in complex environments. To simplify the problem of identification of the actions, a common strategy was adopted by a majority of researchers which consists in treating the actions from only one point of view.

Most action recognition approaches rely on supervised learning methods where training is done on pre-defined sets of choreographed actions in order to do recognition. This differs from surveillance where the set of actions are often unknown or hand labeling of specific actions not applicable. Usually two distinctions are made between approaches to action recognition. Firstly, template matching[1,2] approaches convert a video sequence of events to a static representation such as a single silhouette based image. These templates of activity are then compared to stored action prototypes for classification. Template matching methods suffer from the varying styles and different temporal extents of an action where the template should ideally represent the whole temporal extent of an action. Secondly, state space methods [3,4,5,6] usually provide a solution to the phase problem with the use of time varying models such as the HMM. A common approach is to define each static feature of an action as a state and learn the relationship between these features. Then a motion sequence can be considered as a tour through the state space of these features. To classify an action the joint probability with the maximum value is selected as the criterion for action classification. Most methods, both state space and template matching, are based on computing either appearance based features such as from a silhouette or motion description such as optical flow.

Various methods based on silhouette features have been proposed. Bobick and Davis[7] proposed a view-based approach to the representation and recognition of temporal templates. They introduce Motion History Images (MHI) to represent how an action was performed using different levels of intensity based on the time since the silhouette was captured. A Motion Energy Image (MEI) is the accumulative shape of the person over time and captures where the action was performed and as introduced, can be used for pose invariance. A set of rotational invariant central moments, Hu moments, were extracted from each action and classification achieved using a nearest neighbour approach. Using temporal-spatial filters Chomat and Crowley [8] generated motion templates computed by Principal Compenent Analysis (PCA) using an Bayesian approach to do action classification. Ali and Aggarwal [9] also make use of a person silhouette's to classify a continuous set of actions by extracting skeleton properties from the shape. Star skeleton features were introduced by Fujiyoshi and Lipton [10] to extract 2D posture from a silhouette in real time.

In this research, we aim to categorize and recognize 10 human actions such as: walking, bending, jumping, jumping in the same place, running, skipping, walking a side, waving two hands, waving one hand and jacking. Then we propose a random walk approach as a learning algorithm for categorizing these ten actions. Our learning algorithm is unsupervised method, we don't give any previous knowledge on our training set.

The outline of the paper continues as follows. The proposed human action segmentation is presented in section 2. Section 3 explains the global features used for shape descriptors. Then, in Section 4, we describe in detail our categorization and recognition approaches. Section 5 describes the experimental results while Section 6 present concluding remarks.

2. HUMAN ACTION EXTRACTION

We wish to partition human action video into thwo parts "person and the background". To segment the ROI (Region Of Interest) and track the person o each successive frame, we use an approach based on graph-cut technique. Motion-based estimation defines regions of foreground, background and boundary blocks. An automatic segmentation is realized by obtaining prior knowledge from foreground and background blocks. The result also can be improved by user adjustment.

The segmentation problem is formulated as an energy minimization problem which is settled by using graph cut algorithm, according to the user-imposed constrains. As a pioneer work of object segmentation using a graph cut, the user-interactive segmentation technique was proposed by Boykov and Jolly [11]. They assumed that a given user imposes certain hard constraints for segmentation by indicating certain pixels (seeds) that belong to the object and certain pixels that belong to the background. The main contribution of our approach is that the object and background seeds(regions) are estimated in every frame of sequences without user interaction. Basically each pixel in the image is viewed as a node in a graph, edges are formed between nodes with weights denotes how alike two pixels are, given some measure of similarity, as well as the distance between them. The edges for each pixel can be formed between the pixel with all the other pixels, In attempt to reduce the number of edges in the graph, we will predetermine neighborhood N that describes the neighbors of each pixel and we will be interested in the similarity "distance" between each pixel and its neighbors.

There are two additional terminal nodes: an "object" terminal (a SOURCE) and a "background" terminal (a SINK) (cf. fig.2). These two terminal nodes don't correspond to any pixel in the image but instead they represent the object and the background respectively. The source is connected by edges to all nodes identified as object seeds and the sink is connected to all background seeds. Edges are formed between the source and sink and all other non-terminal nodes, where the corresponding weights are determined using models for the object and background. The min-cut of the resulting graph will then be the segmentation of the image. This segmentation should then be a partition such that, similar pixels close to each other will belong to the same partition. In addition, as a result of the terminal weights, pixels should also be segmented in such a manner so they end up in the same partition as the terminal node corresponding to the model *(object or background)* they are most similar to.

Given an image, we try to find the labeling X that minimizes the energy E:

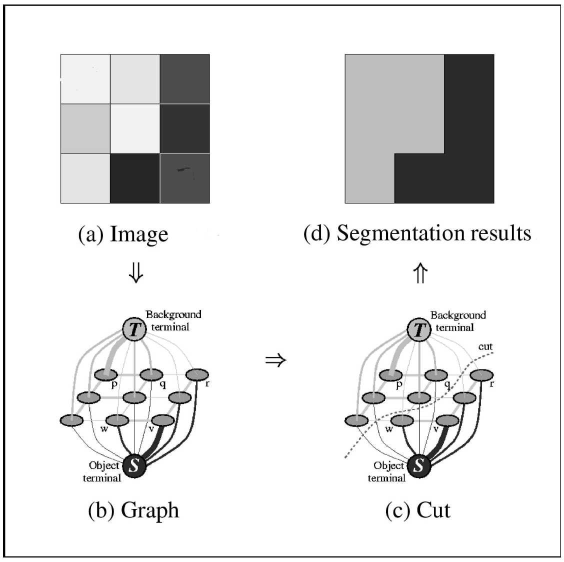
$$E(X)=\lambda\sum_{p\in P} D_p(x_p)+\sum_{pq} B_{pq}\delta(x_p,x_q) \quad p,q\in E$$

In the above equation, coefficient λ specifies the relative importance of the data term D(.) and the smoothness term B(.).

$X:=(x_1,x_2....x_p...x_{|P|})$ is a binary vector whose component xp specifies labels to pixels p. Each x_p value can be either 1 or 0 where 1 represents an object and 0 represents a background area. The vector X defines segmentation. $\delta(x_p,x_q)$ denotes the delta function defined by 1 if $x_p \neq x_q$, and 0 otherwise. The B_{pq} are defined by:

$$B_{pq}=K.exp\left(-\frac{\|I_p-I_q\|^2}{\sigma_2}\right)$$

I_p and I_q are intensities/colors at pixels p and q. K is a constant. The data term Dp measures how well label xp fits pixel p given the observed data. We modeled the object and background color likelihood's of $P(./"Obj.")\equiv P(\cdot/1).$ and $P(./"Back.")\equiv P(\cdot/0)$ using Gaussian mixtures in the RGB color space, using the data taken from the previous frame according to the labels given by the output of the segmentation process.


(a) Image

(d) Segmentation results

(b) Graph

(c) Cut

Fig.1. Example segmentation of a very simple 3-by-3 image. Edge thickness corresponds to the associated edge weight. (Image courtesy of Yuri Boykov.)

In every frame, our object and background estimation processes determine O and B which are the sets of pixels belonging to the estimated object and background regions respectively.

We compute the edge weights between pixels as the following. The edge weight between pixels p and q will denoted as $W(p,q)$ and the terminal weights (source and sink) between pixel p are given by:

$$W(p,S)=-\lambda\ln(P(I_p/"Background").$$

$$W(p,T)=-\lambda\ln(P(I_p/"object").$$

$$W(p,q)=B_{pq}.$$

$W(p,q)$ contains the inter-pixel similarity, that ensures that the segmentation more coherent. $W(p,S)$ and $W(p,T)$ describe how likely a pixel is to being background and foreground respectively.

In a video we construct a 3D graph that is obtained from a series of images that describes the video. Each node from the graph is connected to 26 (Pixels) neighbors, that means it has a 26 edges with weights calculated as described in the 2D graph. We applied the same ideas as above with slightly changes in 3D (cf fig. 2).

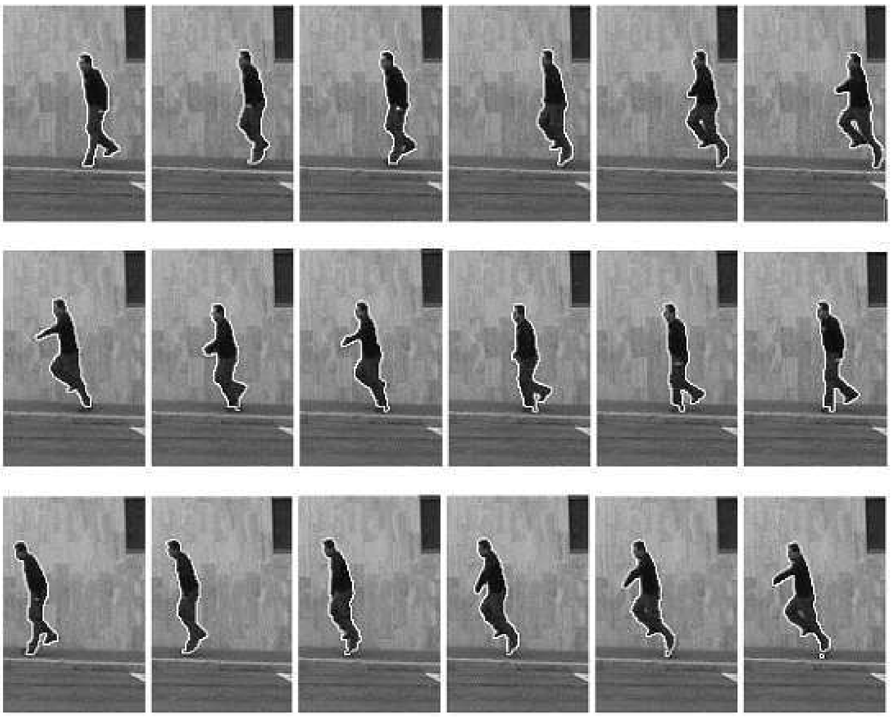


Fig.2. Human action extracted by graph cut approach

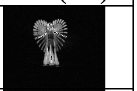
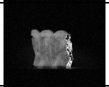
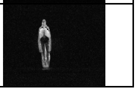
3. GLOBAL FEATURES

1.1 Motion History Image Density

Motion History Image Density uses the same technique as MHI [12] with one major change that it takes into account how many times the pixel is belonging to the object in a video. It takes a video as an input and returns 2D image which represents the historical information about this video that contains the projection of all the images in a video into one image 2d.

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

where H_τ is Motion History Image and D is the binary difference between successively images. x,y and t are pixels coordinates. τ is a threshold for extraction of moving patterns in video image sequence. Thus, MHI is a scalar-valued image where more recently moving pixels are brighter. We extend the MHI filter by giving the pixel a value equal to how many times it is white in the video (it belongs to the object), except pixels that had never changed.

Action	Example	MHI (2D)
« jack »		
« jump »		
« pjump »		

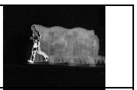
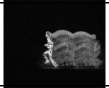
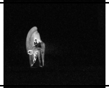
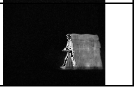
« run »		
« side »		
« skip »		
« bend »		
« Walk »		
	a	b

Fig.3. Example of human actions and results of MHID filter

1.2 3D-Geometrical moments

In general terms, shape descriptors are a set of numbers that are found to describe a shape in compact form. A shape descriptor should ideally be a simplification of its representative region but still hold enough information so that different shapes are discriminated. Usually it either describes the shape boundary or the image region. In our approach we use the features based region description. Moments are a measure of the spatial distribution of 'mass' of the shape of an object. Objects in a binary image are represented as a set of white pixels (in 2D) and voxels (in 3D), videos are a 2D+t (time) images, so we can represent it as a 3D image with Depth equal to t. A set of 14 moments derived by HU gives information about region-based shape descriptor that are rotation, scaling and translation invariant in a 3D dimensions.

Let (x,y,t) be a binary video, that means its voxels values equal to 1 for the voxels belonging to the object (hand) and zero for the background. We can define the moment as:

$$A_{pqr} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^{p} y^{q} t^{r} \, dxdydt$$

A_{000} represent the area of the object and $(A_{100}, A_{010}, A_{001})$ the center of the object.

$$M_{pqr} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left(\frac{x - A_{100}}{A_{200}^{1/4} * A_{020}^{1/4}}\right)^{p} \left(\frac{y - A_{010}}{A_{200}^{1/4} * A_{020}^{1/4}}\right)^{q} \left(\frac{t - A_{001}}{A_{002}^{1/2}}\right)^{r} dxdydt$$

In discrete, the integration is changed to summation. The 3D geometrical feature descriptors are calculated from our videos by applying the 3D geometrical moments directly on the videos. 14 moments are extracted as a feature vector:

$$M_{3D} = \begin{cases} M_{200}, M_{011}, M_{101}, M_{110}, M_{300}, M_{030}, M_{003}, \\ M_{210}, M_{201}, M_{120}, M_{021}, M_{102}, M_{012}, M_{111} \end{cases}$$

4. RECOGNITION BY DIFFUSION MAPS

1.3 Kernel Methods basics

Diffusion maps (DM) are based on defining the Markov random walk on the graph of data. By performing the random walk for a number of time steps, a measure for proximity of the data points is obtained. Using this measure, the so-called diffusion distance is defined. In diffusion maps the graph of the data is constructed first.

Let G = (V, E) be an undirected graph with vertex set $V=\{v_1,...,v_n\}$. In the following we assume that the graph G is weighted, we compute the weights of the edges in the graph using the Gaussian kernel function, leading to the similarity

matrix W of the graph G.

$$w_{ij}=e^{-\frac{d_{ij}}{2\sigma^2}}$$

Where σ indicates the variance of the Gaussian and d_{ij} denotes to the distance between v_i and v_j. The degree of a vertex $v_i \in V$ is defined as : $d_i=\sum_{i=0}^{n-1} w_{ij}$

We define the diagonal matrix D by: $D_{ii}=D(v_i,v_i)=d_i, and\ D_{ij}=0\ for\ i \neq j$

Defining the matrix L such as:

$$L_{ij} = L(v_i, v_j) = \begin{cases} d_i - W_{ii} & if \quad v_i = v_i \\ -W_{ij} & if\ v_i\ and\ v_j\ are\ adjacents \\ 0 & otherwise \end{cases}$$

While the Laplacian of graph G can be defined by: $\Im = D_{-1/2} LD_{-1/2}\ where\ \ D_{ii}^{-1} \equiv 0\ if\ d_i=0$

The transition probability of vertex v_i to vertex v_j in each step is: $p_{ij}=w_{ij}/d_i$

This defines the transition matrix P of the chain Markov.

$\forall v_i, \forall v_j, 0 \leq p_{ij} \leq 1 and \sum_{j \in V} p_{ij}=1$,

TABLE I: Categorization by Diffusion Maps

Keywords: Diffusion Maps Approach

Input: *graph*

vertices $X = \{x_1, x_2, ..., x_n\} \subset R^d, t, m, \varepsilon$

- *Construct a matrix of similarity. Let W be its weighted adjacency matrix.*

$$w_{ij}^{\varepsilon} = exp(\quad \|x_i - x_j\|^2 / \varepsilon)$$

- *Normalization by using the Laplace-Beltrami method :*

$$\tilde{w}_{ij}^{\varepsilon} = w_{ij}^{\varepsilon} /(d_i d_j)$$

- *Calculating the transition matrix:*

$$p_{ij} = \tilde{w}_{ij}^{\varepsilon} /(\sqrt{\tilde{d}_i \tilde{d}_j}) \text{ with } d_i=\sum_{i=0}^{n-1} w_{ij}^{\varepsilon}$$

- *Diagonalization of matrix P*

Diffusion space:

- *Compute the first k eigenvectors* $v_1....v_k$ *of P.*
- *Normalize the eigenvectors, dividing each row by its first value* λ_0. $\lambda_i = \lambda_i / \lambda_0$
- *Y = sorting the vectors by λ1, λ2, λ3.*
- *Cluster the points* $(y_i)_{i=1....n}$ *in* R^k *with the k-means algorithm into clusters* $C_1....C_k$.

Output: *Clusters* $A_1....A_k$ *with* $A_i = \{j | y_j \in C_i\}$

We can also write $P=D_{-1}W$. Now, we define our approach of categorization based diffusion map in graph. The table 1 shows our categorization approach.

1.4 Kernel Methods and the Nystrom extension:

Let $\Omega=\{x_1,x_2,......x_n\} \subset \Re^d$ be the set of training points. The kernel is a function k : Ω × Ω → R, such that there exist a mapping φ : Ω → H, where H is a Hilbert space and the following inner-product relationship holds

$$k(x_i,x_j)=p_{ij}=\langle \varphi(x_i),\varphi(x_j) \rangle \ i,j=1,...,n$$

Let $K \in M_{n \times n}$ be the matrix containing the kernel values, $K_{ij}=k(x_i,x_j)$. If this matrix is semi definite positive, then k is a kernel over the set Ω. A mapping satisfying the dot product property (1) can be found by the Eigen-decomposition of the kernel matrix K:

$K=U \Lambda U_T = U \Lambda_{1/2} (U \Lambda_{1/2})_T$,Where U is the matrix whose columns are the eigenvectors $\phi_i j=1,...,n,)$ and $\Lambda=diag(\lambda_1,\lambda_2,....,\lambda_n$ is the diagonal matrix of the Eigenvalues in decreasing order. If we define $\varphi(x_i)$ to be the i-th row of $U\Lambda_{1/2}$, and since the Eigenvectors are non-negative (positive semi definite matrix), we obtain the desired mapping:

$$\varphi(x_i) = \left[\sqrt{\lambda_1}\phi_1(x_i), \sqrt{\lambda_2}\phi_2(x_i), ..., \sqrt{\lambda_n}\phi_n(x_i)\right]$$

The kernel function can then be considered as a generalization of the dot product, and therefore it is a measure of similarity between the input points. The Hilbert space H is called the feature space. When the algorithm to be applied

in the feature space uses only the corresponding dot products, only the kernel values are needed, without the need for the explicit computation of the mapping functions. This is called the kernel trick.

Let $x\in\Re^d$ be a new input point not in the training set. The Nystrom extension, states that the j-th coordinate of the kernel mapping □ for this point can be approximated as:

$$\varphi_j(x)=\frac{1}{\sqrt{\lambda_j}}\sum_{i=1}^{n}k(x,x_i)\phi_j(x_i)\quad j=1,2,...,n$$

or in vector form:

$$\varphi(x)=\frac{1}{\sqrt{\Lambda}}U^T k_x,$$

where $k_x=\left[k(x,x_1),....,k(x,x_n)\right]$, and $\frac{1}{\sqrt{\Lambda}}$ stands for $(\sqrt{\Lambda})^{-1}=diag(\frac{1}{\sqrt{\lambda_1}},....,\frac{1}{\sqrt{\lambda_n}})$. In other words, the new point x is mapped as a weighted linear combination of the corresponding maps for the training points x_i. The weights are given, modulo normalization by the Eigen values, by the kernel relationship $k(x,x_i)$ representing the similarity between x and x_i.

Observe that while extending the mapping, we also need to extend the kernel. This is straightforward when the kernel defined over Ω is simply a known function defined in the ambient space $\Re^d$. In other cases the extension of the kernel is not trivial.

When the data is sampled from a distribution, it has been shown, that the functions defined by the Nystrom extension converge uniformly to the Eigen functions of the limit of the sequence of data driven kernels, given that this limit exists and that their Eigen functions also converge. This asymptotic property makes the Nystrom extension an appealing approach for the out-of-sample extension problem [13].

5. EXPERIMENTS

We aim to categorize these ten human actions: walking, ending, jumping, jumping in the same place, running, skipping, walking a side, waving two hands, waving one hand and jacking. We used data-base of 89 videos from [14], which are 9 videos for each action except 8 videos for the skip action. They were taken from 9 persons. Each video contains a stable background which makes it easy to segment and extract the moving body in a video as the object. In the next parts we work on the binary videos where the human body is the object in each video.

It turned out to be hard to find which features are good to categorize these ten actions, because the diffusion maps uses the distances between all the examples. The distances are different in the same group of action videos. This leads to the need of finding a strategy for the categorization of the ten actions. First we categorize the ten actions as two classes: classI contains actions that need to move all the body like (Walking, Jumping, Running, Skipping and Walking a Side); classII contains actions that need to move only part of the body (mostly the hands) or to move the body in the same place like (Bending, Jumping in the same place, Waving two hands, Waving one hand and Jacking). To do this categorization we used the three features described as the first group features -(difference area, Distance of Action and Center variance on x axis)- as vector descriptor extracted from the MHI transformed image of our videos. Results showed that by using these features as input to a Diffusion Maps, it is possible to categorize the ten actions into the two classes like we described above.

Now, after the two classes were categorized well, we applied the categorization of a diffusion maps using 14 geometrical moments in 3d (2d+t) as features descriptor to categorize the second class (class II) aiming to recognize what is the action in this class. The 14 moments were calculated on the MHIDT images; it gave good results.

There was only one mistake in all the videos in this class (45 videos of classII), After the process of the spectral clustering we can decide what is the action in this class by applying the recognition function using the Nystrom extension method. We will get then λ_1 ; λ_2 ; λ_3 as result. From the results, we found that the actions in this group can be categorized by only λ_1 as following:

λ_1	**Action**
≤ -0.33	Bend
$-0.23 < \lambda_1 \leq -0.1$	Jump in place
$-0.33 < \lambda_1 \leq -0.23$	Wave one hand
$-0.1 < \lambda_1 \leq 0.1$	Jack
$\lambda_1 \geq 0.1$	Wave two hands

For classI actions, we tried several features the geometrical moments, shape descriptors and 2d moments. It was difficult to categorize the actions in this lass correctly. We found that the best way to categorize these actions is by separating again this class into some classes, for some action groups we get some good results; For the bad results we will show later what their action is, by using anther features. We found that using the five features (Length Difference of the global rectangle, Length Variance of the global rectangle, Width Difference of the upper rectangle, Center variance on the X axis, Width Difference of the Width Difference of the upper rectangle, Center variance on the X axis, Width Difference of the bottom rectangle), described in the features part as the second features group, it is possible to categorize the five action in this class into three classes using again the Diffusion Maps techniques "spectral clustering" with σ =0.005.

Finally we calculated two features, described in the features section as the third group, (variance of the (convex Hull - object), and the Difference between convex Hull and object). To decide what the action is from the third class we use here a new diffusion map to separate the three actions. We define the categorization in the recognition process by λ_1 , λ_2 , σ = 0.365 as the following:

λ_1	λ_2	**Action**
< 0.1	> 0	Skip
< 0.1	< 0	Jump
> 0.1	*Whatever*	Run

Results were quite good here. In these three actions we have 26 examples of three actions, our approach categorized and recognized well 24 videos. However, two videos from present the skip action gave wrong answers. Finally, our results for all the data-base videos achieved activity recognition rates above 96.6\%. This demonstrates that, without any previous learning, our technique performs very well as human motion categorization method.

The categorization and recognition process flow:

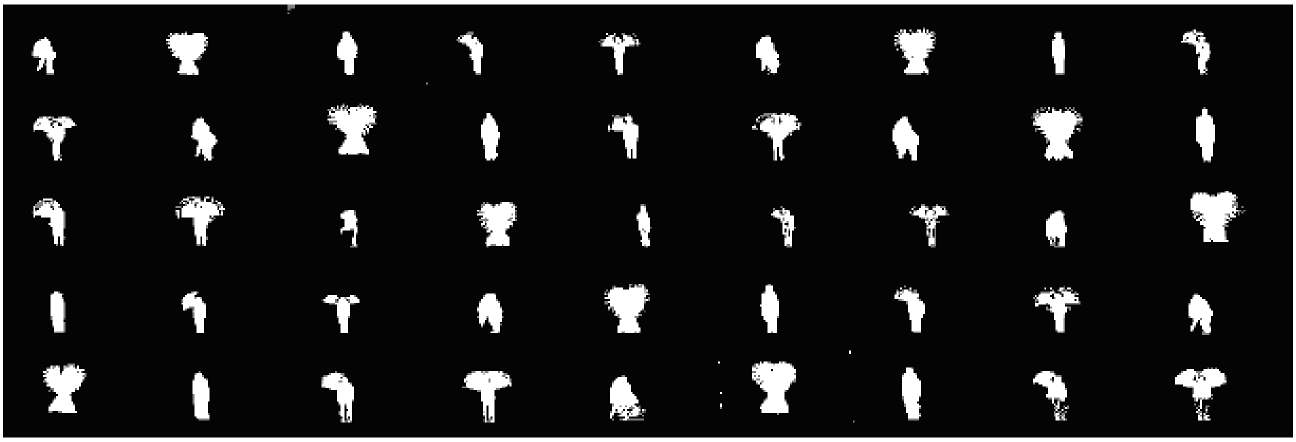


Fig.4. MHI of the five actions belonging to class II

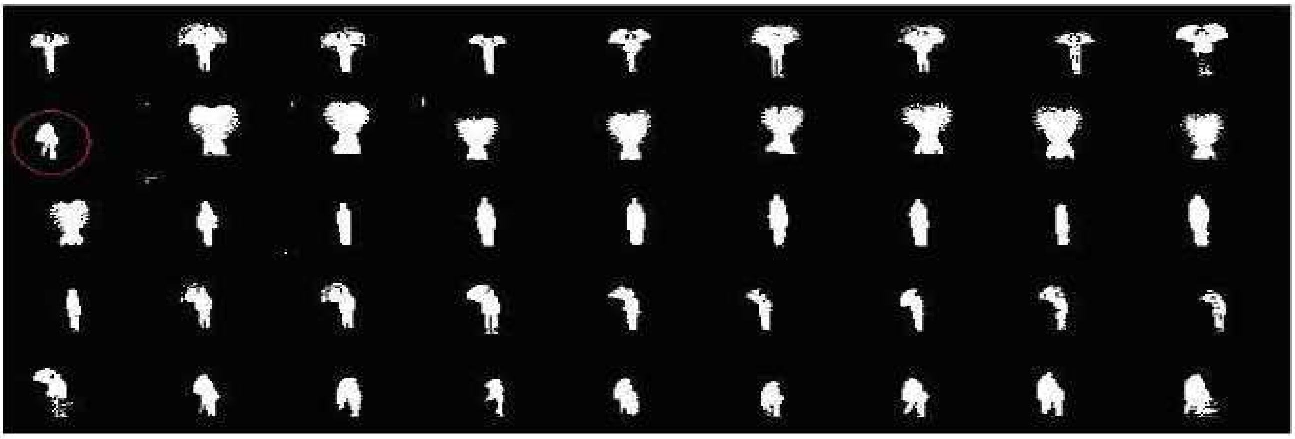


Fig.5. MHI of the five actions belonging to class II, the order was done by the first Eigen vector with σ = 0:000045

6. CONCLUSION AND FUTURE WORKS

A new approach based on Motion Descriptors, mass features, 3D geometrical moments and Diffusion Maps for human motion action categorization and recognition is presented. The proposed framework classifies haptic properties through the video analysis of human motion actions. Ten motion action have been tested walking, bending, jumping, jumping in the same place, running, skipping, walking a side, waving two hands, waving one hand and jacking.

We present Binary videos, each video contains a human who is doing an activity. We calculate three mass features -(difference area, Distance of Action and Center variance on x axis)- from the MHI image of the video to separate the ten actions into two classes, using the Diffusion Maps. By the Kernel method and the Nystrom extension algorithm we define the two classes.

For the first class I, five features -Length Difference of the global rectangle, some descriptors (Length Variance of the global rectangle, Width Difference of the upper rectangle, Center variance on the X axis, Width Difference of the bottom rectangle) were calculated and presented as input vectors. We categorized the five actions in this class as three groups (First: walk, Second: walk a side, third: (run, skip, jump)). We used two more features -(variance of the (convex Hull - object), and the Difference between convex Hull and object)- to categorize the (run, skip, jump) actions. The all categorization processes here were done by using the Diffusion Maps (spectral clustering) algorithm; the recognition processes were done by using the Kernel method and Nystrom extension method.

Robust global features are extracted, based on 3D geometrical moments for the class II actions, These vectors are then used to categorize the five actions classified as the second class by Diffusion Maps.

We tested the proposed algorithm only on our 89 videos; the results for all data-base videos achieved activity recognition rates above 96.6\%. This demonstrates that, without any previous learning, our technique performs very well as human motion recognition methods.

There are many avenues of future work for this part, including the recognition of the human motion actions tracked by the live camera, improve the recognition system in order to reduce the recognition errors, the recognition system should correct training data online, we also plan to improve segmentation algorithm to extract the human body in real time, also it will be very important to do a virtual reality system which simulates the human motion action.

7. REFERENCES

[1] Cui, Y. and Weng, J. (1997). Hand segmentation using learning based prediction and verification for hand sign recognition. In Proc. of IEEE CS Conf. on Computer Vision and Pattern Recognition., pages 88–93.

[2] Rabiner, L. R. (1990). A tutorial on hidden markov models and selected applications in speech recognition. pages 267–296

[3] Bobick, A. F. and Wilson, A. D. (1995). A state-based technique for the summarization and recognition of gesture. In ICCV '95: Proceedings of the Fifth International Conference on Computer Vision, page 382, Washington, DC, USA. IEEEComputer Society.

[4] Brand, M., Oliver, N., and Pentland, A. (1997). Coupled hidden markov models for complex action recognition. In ICVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (Computer Vision and Pattern Recognition '97), page 994, Washington, DC, USA. IEEE Computer Society.

[5] Bregler, C. (1997). Learning and recognizing human dynamics in video sequences. Computer Vision and Pattern Recognition, 00:568.

[6] J. Yamato, J. O. and Ishii, K. (1992). Recognizing human action in time sequential images using hidden markov model. In Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, pages 379–385. IEEE Computer Society.

[7] Bobick, A. F. and Davis, J. W. (2001). The recognition of human movement using temporal templates. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(3):257–267.

[8] Chomat, O. and Crowley, J. (1998). Recognizing motion using local appearance. In Proceedings of the Sixth International Symposium on Intelligent Robotic Systems, Edinburgh, Scotland, pages 271–279.

[9] Ali, A. and Aggarwal, J. K. (2001). Segmentation and recognition of continuous human activity. IEEE Workshop on Detection and Recognition of Events in Video, 00:28.

[10] Fujiyoshi, H. and Lipton, A. J. (1998). Real-time human motion analysis by image skeletonization. IEEE Workshop on Applications of Computer Vision, 00:15.

[11] Y. Boykov, and M. Jolly. Iterative graph cuts for optimal boundary and region segmentation of objects in N-D Images. Proc. IEEE 8th Int. Conf. on Computer Vision, Canada, CD-ROM, 2001.

[12] G. Bradski and J. Davis, Motion Segmentation and Pose Recognition with Motion History Gradients, IEEE Workshop on Applications of Computer Vision, December 2000.

[13] Pablo Arias, Gregory Randall, Guillermo Sapiro, Connecting the Out-of-Sample and Pre-Image Problems in Kernel Methods, Universidad de la Republica, Universidad de la Republica, University of Minnesota, IEEE Computer Society Conference on Computer Vision and Pattern Recognition - 18-23 jun 2007

[14] M.Blank, L. Gorelick, E. Sechtman, M. Irani, and R. Basri. Actions as Space-Time Shapes, IN ICCV 2005.