Recognizing human motion using eigensequences

Andrea Bottino
Politecnico di Torino
Dipartimento di Automatica e
Informatica
Corso Duca degli Abruzzi, 24
10129 Torino Italy

andrea.bottino@polito.it

Matteo De Simone
Politecnico di Torino
Dipartimento di Automatica e
Informatica
Corso Duca degli Abruzzi, 24
10129 Torino Italy

matteo.desimone@polito.it

Aldo Laurentini
Politecnico di Torino
Dipartimento di Automatica e
Informatica
Corso Duca degli Abruzzi, 24
10129 Torino Italy

aldo.laurentini@polito.it

ABSTRACT

This paper presents a novel method for motion recognition. The approach is based on 3D motion data. The captured motion is divided into sequences, which are sets of contiguous postures over time. Each sequence is then classified into one of the recognizable action classes by means of a PCA based method. The proposed approach is able to perform automatic recognition of movements containing more than one class of action. The advantages of this technique are that it can be easily extended to recognize many action classes and, most of all, that the recognition process is real-time. In order to fully understand the capabilities of the proposed method, the approach has been implemented and tested in a virtual environment. Several experimental results are also provided and discussed.

Keywords

Motion recognition, real time recognition, PCA, model-based motion capture.

1. INTRODUCTION

The recognition of human movements is an important topic in computer vision and has many promising applications in entertainment, human computer interaction, automatic video indexing [Nam97], video surveillance and intrusion detection [Har00]. An important requirement is speed. Especially for interactive applications, real-time recognition rates are needed. In applications like computer games or human computer interactions, the user should not perceive any noticeable delay between the action performed and the system response.

This paper presents a novel technique for recognizing human motion in real time. The movement of a performer is first captured in 3D by means of a model-based technique. The basic idea of our approach is the following: a movement is a curve in the (normalized) model parameter space, and these curves are characteristic of the type of action performed. Comparing the whole curves for

recognizing actions does not seem to be a good idea, for instance because the same action can have different lengths, but we can extract small segments of curves and use them for comparison. In our idea these segments, called *sequences*, are still characteristics of the kind of action performed, as we will demonstrate in the paper. Using sequences allows for an "on line" recognition, necessary in several applications, since we do not need to acquire and process the whole motion to classify it. Exploiting *Principal Component Analysis* (PCA), a reduced dimensionality model of the sequences can be used to recognize several basic actions, like walking, running or waiting.

The contribution of this work is a system that, after a proper training, is capable of recognizing many different action classes in real time. Furthermore, the approach can be easily extended to deal with other motion classes. The proposed method exploits 3D motion data, since 2D techniques often impose constraints on the characteristics of the motion to be analysed and of the available data. For instance, in several approaches the performer's motion must be frontal or parallel to the image plane ([Yac99], [Rah05], [Dav01], [Cao04]), which are too limiting for applications like smart surveillance systems or sports analysis.

The extraction of 3D motion data and their classification are actually two independent processes, and this work focuses on motion recognition only.

Extracting 3D motion data is a complex task. However, several real-time non-intrusive motion capture techniques, such as [The03] or [Hig03], are available and can be coupled with our motion classifier.

This paper is aimed at demonstrating the capabilities of the proposed approach. Since, in principle, the recognition process is independent on how motion data have been acquired, we initially tested the approach with motion data acquired with an optical capture system. However, for several applications, non-intrusive capture techniques must be used. Therefore, we also experimented the proposed approach with a non-intrusive tracker in a virtual environment in order to understand how the reconstruction errors introduced by those methods affect the classification process. The preliminary results are promising, demonstrating a good capability of separating the analyzed movements.

The contents of the paper are the following. In section 2 we outline the contents of this work. Section 3 describes the motion recognition approach and the experimental results are shown in section 4.

Related works

PCA is a popular technique in pattern recognition. Eigenspace analysis and PCA methods have been successfully applied to several computer vision problems, like face recognition, background modelling, object detection and tracking ([Del01]). Several approaches to motion recognition use PCA on various 2D or 2½D motion data extracted from monocular sequences ([Yac99], [Per02], [Ben02]) or directly from video sequences ([Rah05]). In order to recognize segmented 2D motion trajectories, the use of a reduced PCA space has been coupled in several approaches to other techniques, like Hidden Markov Models ([Bas05]) or neural networks ([Hyu05]). However, in our knowledge, this is the first work that deals with PCA classification of 3D motion data.

In literature, state-space and template matching are the most popular approaches to human motion classification. In state-space approaches ([Sun02], [Kel05], [Ime03], [Gal01]), the set of features extracted from the incoming images, for instance 2D/3D postures or optical flow data are represented in a suitable manner to define different states connected by transition probabilities. A motion sequence can be thought as a tour between different states. Given the state sequence for an incoming motion, activities recognition is achieved modeling each activity as a state diagram and matching the incoming sequence with the most probable model. On the contrary, template-matching techniques compare the extracted features with pre-stored patterns for recognition. Those techniques can exploit several clues, like *Motion History Images*

([Dav01], [Bab03]), Gabor features ([Nak01]) and contours ([Rit99]). In both cases, the performances of these algorithms are strictly related to the type and quality of the extracted features used. A review of the literature on the field can be found in [Agg99].

2. SURVEY OF THE APPROACH

Our approach to motion recognition consists in analysing the 3D motion data of a performer and identifying the actions he performs. Therefore, in order of principle, the approach is absolutely independent on the way motion data are obtained. As we will show, the recognition process can be performed in few milliseconds. When coupled with a real-time motion capture system, all the tracking-recognition chain can be performed in real-time.

Figure 1 outlines our recognition framework. The motion curves in the parameter space are split into *sequences*, which are segments of fixed length. The PCA based representations of the training sequences for the action classes to be recognized (right) are compared for recognition with the same representation of the sequences of the incoming movement (left).
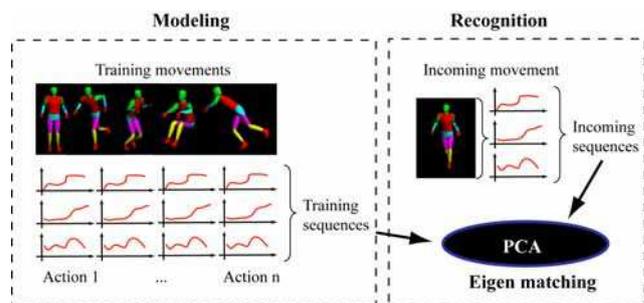

Figure 1. Outline of the recognition process

Before describing in details the proposed method, let us give some definitions. The attitude of the human body is referred to as a *posture*. A *motion* is a sequence of contiguous postures over time. An *action* is a specific type of motion, for instance walking, running or sitting. The aim of our approach is to automatically segment a motion, classifying the various actions it contains.

Figure 2: the human body model and its skeleton

To represent and reconstruct the motion of a human performer, we use a human body model that is defined by an articulated structure whose skeleton has 23 segments and whose surface is described by a triangular mesh (see Figure 2). Skeleton segments are organized into a tree whose root is located in the pelvis. The model has 29 degrees of freedom, 26 to define the rotation of the segments and 3 for the (x,y,z) position of the pelvis.

In order to deal with 3D data coming from different performers, we have to perform spatial normalization. Motion data can be made almost independent from performer's characteristics and from motion orientation with respect to a global reference system considering only angular data. In particular, all information about model's measures are discarded, the skeleton's root is translated into the origin of the reference system and the model is rotated in order to make the pelvis segment coincident with the z axis of the world coordinate systems, heading towards the x axis. Therefore, the dimension of the posture vector can be reduced to 26. We did not perform any temporal normalization for two reasons. First, in our opinion, the speed of a gesture provides a strong clue to detect its class of movement. For instance, speed is necessary to tell a slap from a caress, being the main distinctive parameter between the two actions. Second, temporal normalization would require the acquisition of the whole motion, preventing real time recognition. However, the PCA representation of the data, described in the following, introduces a sort of temporal normalization.
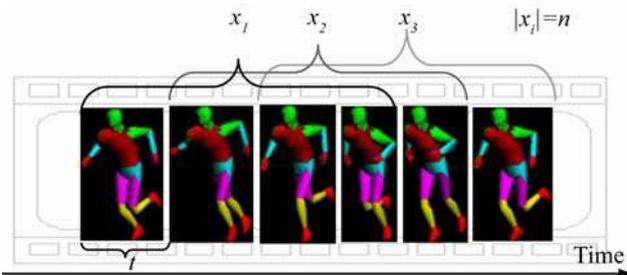

Figure 3: a motion is divided into three sequences x_1, x_2 and x_3. Each sequence contains n frames, and the interval between the beginnings of two consecutive sequences is t frames

Every movement can be represented into the reduced posture space as a curve, or motion curve. But how can we use motion curves for recognition? Our approach consists in splitting a complex movement into a set of "atomic" motions. We define as *sequence* a set of n consecutive frames, the starting frames of two sequences being at a distance of t frames (see Figure 3). Each sequence is defined by a unique vector containing the model parameters of its composing postures. A sequence represents the atomic quantity of motion that we want to recognize. In the posture space, a sequence represents a segment of a motion curve. Our idea, which will be demonstrated in the next section, is that these segments are characteristic of the type of action performed, that is sequences belonging to a walk are different from sequences belonging to a run. In this way it is possible to create classes of sequences that can be labelled as belonging to a specific action class. When a movement has to be recognized, we can extract its composing sequences and classify each of them, obtaining an indication of the actual action performed. The choice of proper values for t and n has been a fundamental part of our tests, and it will be discussed in section 4.

The motion recognition process is based on the principal component analysis. All sequences are represented by vectors of the same size. PCA can be applied to decompose the original sequences into a set of characteristic feature data, called *eigensequences*, which can be seen as the principal components of the original sequences. Eigensequences form the orthogonal basis of a linear subspace, called the *sequence space*. Therefore we can recognize an input sequence projecting it on the sequence space and comparing its position with those of known samples. If we see eigenfaces as a set of "standardized face ingredients", the eigensequences can be thought as being a set of standardized motion ingredients.

The advantages of the proposed approach are the following:

- the set of recognizable action classes can be increased at will, as soon as training motions for the desired action classes are available
- short detail movements can be easily recognized
- changes of the actions performed by the subject can be immediately identified, which is necessary for some applications like real-time video surveillance
- PCA allows reducing greatly the data dimension, providing for their real-time processing

3. RECOGNIZING MOTION

Let a sequence be composed of n normalized pose vectors and represented by a vector x. Let Z be the number of action classes to recognize. For each action class, we use a set of training motions and each one of them is split into its composing sequences. Let $\{x_i \mid i=1, …, S\}$ be the complete set of

training sequences. The average sequence for this set is defined as:

$$\mu_x = 1/S \sum_{i=1}^{S} x_i$$

The covariance matrix of the set is given by:

$$C_x = A \cdot A^t$$

where

$$A = [(x_1 - \mu_x),...,(x_S - \mu_x)]$$

is the matrix containing the differences between the training sequences and the mean sequence. The basis vectors of the training sequence space are the orthogonal eigenvectors of C_x.

The dimension of a sequence can be reduced expressing its components in terms of the *eigensequences* $e_1,...,$ e_k that are the eigenvectors corresponding to the largest k eigenvalues of C_x. These eigensequences form a reduced linear subspace, that we will call the *sequence space*. Therefore, each training sequence x_j can be expressed in the sequence space as a characteristic vector $g_j=[g_{j1},..., g_{jk}]$, that can be obtained as:

$$g_j = [e_1,...,e_k] \cdot (x_j - \mu_x)$$

In order to minimize the error introduced projecting the original vector on the sequence space, the value of k corresponds to the value for which the ratio of the eigenvalue sum is above a predefined threshold ε, as follows:

$$k = \min_r [r \mid (\sum_{i=1}^{r} \lambda_i / \sum_{i=1}^{n} \lambda_i) > \varepsilon]$$

In the recognition phase, each motion is divided into sequences; an incoming sequence x is projected onto the sequence space, obtaining its characteristic vector g. Therefore, each motion gives rise to a set of points in the eigenspace.

For comparison, we use the so-called point-set representation, describing a motion class by the set of vectors projected from all its training sequences. The distance d_j to each training sample in sequence space is defined as:

$$d_j = \|g - g_j\|^2 \qquad j = 1,...,k$$

The action class of the training sequence at minimal distance gives the action class of x. It is worth noting that the distance in frame of the beginnings of two inputs sequences can be different from the one used to extract the training sequences. As a matter of fact, in all our tests, the input sequences have a distance of one frame. This allows a per-frame classification of the input motion.

The training motion sequences used have been acquired with an optical motion capture system and are part of Motek's StockMoves[tm] library ([Motek]). In order to test our approach we selected several animations corresponding to seven action classes: *walk, run, wait, jump, slide, rowing* and *squat*. For each class, we picked out a subset of the available animations to extract the training sequences. The

animations have been chosen in order to provide the widest variability possible for the specific action. For instance, normal and lame walks, runs at different speed, normal jumps and jumps with turns, and so on. Some frames extracted from a running motion are shown in Figure 4.
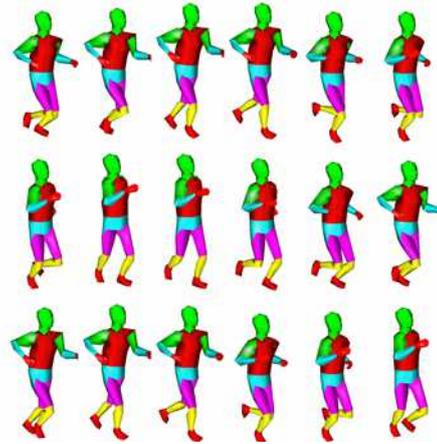

Figure 4. A running motion

An indication of the fact that the sequences are characteristic of the type of action performed can be seen in Figure 5, where the eigenspace representation of the training sequences is shown. For the sake of visibility, only the three eigensequences corresponding to the three largest eigenvalues are used to create the graphical representation. As it can be seen, sequences corresponding to the same action class cluster in the eigenspace representation, and the different action classes are considerably separated.
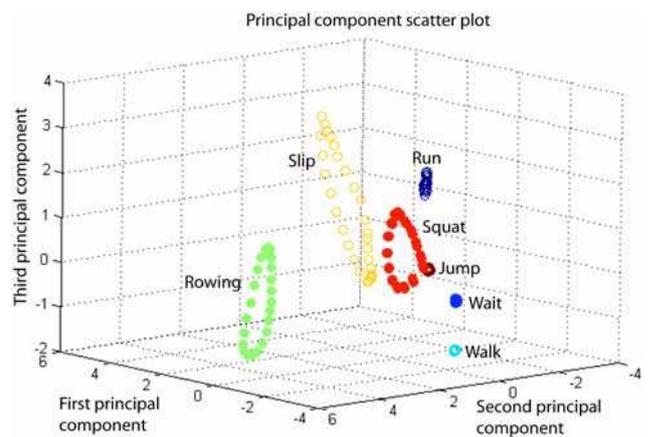

Figure 5. Eigenspace representation of the training sequences

4. EXPERIMENTAL RESULTS

Several experiments have been carried out in order to study the behaviour of the proposed approach with respect to the different parameters involved. These parameters are:

- The number n of frames composing a sequence
- The distance t in frames between two consecutive sequences in a training motion
- The value of the threshold ε used to select the k eigensequences defining the sequence space
- The number S of training sequences
- The representative motions used to extract the training sequences for each action class

The experimental work has been divided in two phases.

In the first phase the approach has been tested directly on the available motion capture data obtained from Motek in order to investigate the precision of the classification process for several motions and several arrangements of the parameters.

In the second phase we used a non-intrusive motion capture approach in a virtual environment. In this case, the motion data are used to animate a dummy. Its motion is reconstructed using the non-intrusive motion capture technique described in [Bot01]. The aim of this second phase is to understand how the reconstruction error introduced by a non-intrusive capture method affects the classification process. It should be outlined, however, that the proposed approach to motion recognition is independent on how motion data are acquired, that is with optical or magnetic tracking or any other motion capture method, until they are presented to the classifier in the desired format.
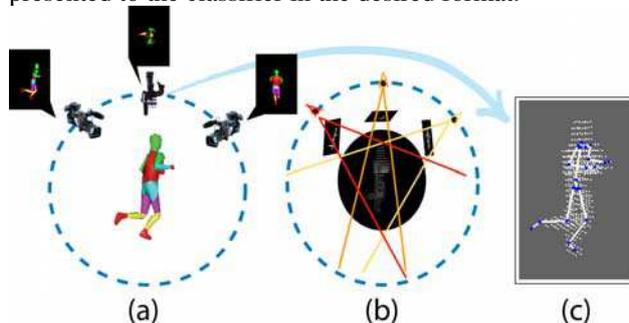

Figure 6. The motion capture technique

The motion capture technique used is depicted in Figure 6. In short, several images of the performer, taken with calibrated cameras, are used (Figure 6(a)). On each image plane, the silhouette of the performer is extracted, and it is back-projected in 3D from the view center the corresponding camera. The obtained viewing cones are intersected, and the resulting volume is represented as the set of voxels containing its surface (Figure 6(b)). The model is then fitted in 3D to the reconstructed surface by minimizing a suitable distance function between the surface of the model and the set of voxel centers (Figure 6(c)). It

should be noted that the motion capture technique used is not real-time. However, more complex reconstruction techniques fulfilling this requirement, such as [The03] or [Hig03], can be used.

In the two phases we used both animations from Motek's StockMoves[tm] library, different from the ones containing the training sequences, and motion-captured sessions, still provided by Motek, where the actors perform motions containing different kinds of actions. Those sequences have been pre-labelled at hand in order to compare the classifier's results with ground truth data. The frame rate of all the motion-captured data is 30 frames/sec.

Despite the value of the parameter t used to separate training sequences, the distance between sequences extracted from the incoming motions is always one frame. This allows having an instantaneous classification of the current motion. We also consider only the instantaneous classification of a single sequence, while other approaches considering the classification of contiguous sequences, that is performing some kind of filtering on the output data, are not taken into account in this work.

The performances of the different tests are measured using the true/false recognition rates, defined as the ratio of correctly/incorrectly classified input sequences to the total number of incoming samples. In the following tables, the results presented can be read in the following way. The first column is the test-set reference name, the second is the dimension n of the used sequences, the third is the distance t in frames between two training sequences, the fourth the number of training sequences S, the fifth the value of ε used to select the k eigensequences defining the sequence space, the sixth the cataloguing average time in milliseconds and the last two columns the true and false recognition rates.

The aim of the first battery of tests, whose results are summarized in Figure 7, is to understand how the sequence dimension affects the recognition rate. As it can be seen, varying the dimension of n from 5 to 20, the recognition rates are essentially equal. However, a smaller value is preferable, since it allows detecting the changes of the performed actions in a shorter time. We found the value of 5 being the best but also a limit to the sequence dimension, since for fewer values the recognition rates decrease significantly, as can be seen in Figure 7, where we have 59% of correct recognitions for 3 poses sequences. Finally, we have run our experiments in order to check the use of PCA on single poses, and the recognition falls below 54%, demonstrating that PCA based single posture recognition is not effective.

The table in Figure 8 outlines the variation of the recognition rates as function of the distance t in

frames between two consecutive training sequences. Also in this case, the results are almost constant and varying t does not significantly affect the quality of the reconstruction. Moreover, increasing the value of t reduces the number of available training sequences S and, consequently, the mean cataloguing time of the input sequences.

Apparently, this intuition is contradicted by the results detailed in Figure 9, where the recognition rates as function of the number S of training sequences are shown. It can be seen that recognition rates are decreasing when a lower number of training sequences are used. This somewhat contradictory behaviour can be explained by the following consideration. In the tests of Figure 8 the number of training motions for each action class is the same for all the experiments, while in Figure 9 the training sequences are reduced by discarding some of the selected training motions. The result is that the training sequences are less representative of the variability of the single action class, and the recognition rate decreases.

A predictable result is the improvements of the recognition rates determined by the parameter ε (Figure 10). Increasing its value, the classifier uses a greater number of eigensequences, the data are represented with higher accuracy in the sequence space and the recognition error is reduced.

The confusion matrix (Figure 11) shows the relationship between the ground-truth labelled data and the results of our classifier. Each row represents the probability of a sequence belonging to an action class to be classified into the available action classes. The diagonal of this matrix, therefore, shows the correct classification probabilities.

Summarizing the results obtained in the first phase of the experimental work, we have that:

- the recognition rate is substantially unaffected by the number of postures composing a sequence; this allows choosing a smaller value of n, providing for faster recognition of action changes
- PCA recognition applied to single postures is not effective
- reducing the number of training sequences, keeping constant the number of training motions used to create them, does not affect substantially the recognition rate; this allows to increase the value of t, that is to have a reduced number of sequences which are more spaced on the motion curves in the parameter spaces, reducing also the recognition time
- provided the same values for n and t, the recognition rates increases when the classifier uses a larger set of training

motions for each action class; as a matter of fact, increasing the number of training motions allows to describe a wider variability for the specific action

In order to test the reliability of the classification results when input data are affected by reconstruction noise, the motions used to create the table in Figure 8 have been used to animate a dummy in a virtual environment. The motion data to be classified have been captured with the non-intrusive motion capture system previously described. In Figure 12, for each test set, the recognition rates using directly the motion data are compared with the recognition rates of the captured data. In all the cases, the best performances are obtained with the direct data. However, the loss of quality is lower than 1% in all the cases, showing that the recognition rates are relatively unaffected by the reconstruction error introduced by the non-intrusive MC system.

Finally, we ran several experiments on motion data of an actor performing several actions in the same motion (Figure 13). The motions have been labelled at hand in order to provide ground-truth data for the experiments. As it can be seen, recognition rates are somewhat lower than the previous experiments. This is primarily due to the inter-action frames, which were difficult to classify even for a human observer and are somewhat confusing the classifier.

Concerning computational times, we ran our experiments on a 2.5GHz PC with 1 GByte of RAM. Constructing the sequence space can be done off-line as a pre-processing stage and takes between 3 and 6 seconds, depending on the number of training sequences. The mean classification time is always below 11 ms, which means that the classification can be performed in real-time. The complexity of the recognition process is O(Sn), where S is the number of training sequences and n is the sequence dimension.

5. CONCLUSIONS

This paper introduces a new approach to motion recognition. The 3D motion of a performer is first captured and then divided into sequences, each one representing a small segment of the curve describing the motion in a normalized posture space.

The motion recognition process is based on the principal component analysis. PCA can be applied to decompose the original sequences into a set of characteristic feature data, called *eigensequences*, which can be seen as the principal components of the original sequences. Given the eigensequences, every training sequence can be represented as a vector of weights; the weights are obtained by projecting the sequences into the sequence space. When an input sequence has to be identified, its vector of weights

also represents it. Identification is done by locating the training sequence whose weights are the closest to the weights of the incoming sequence.

The approach has been tested in a virtual environment in order to understand the capabilities of the method, and the influences of the various parameters on the recognition rate. Results of the recognition process in a controlled environment have been presented, showing encouraging recognition rates (about 89% in the best configurations). The ability of achieving real-time classification has also been demonstrated, since the recognition time is always lower than 11 ms.

One interesting research line could be to test the system in a real environment. Another research line involves the vector comparison method, which can influence the system's performance dramatically. For example, PCA algorithms can use either the angle or the Euclidean distance between two projection vectors, the Euclidean distance can be weighted or unweighted, or we can use non-Euclidean metrics. Also LDA algorithms, which have often showed promising results, will be tested. Finally, we are planning to experiment different clustering techniques, like *kernel PCA* [Smo98], to see if recognition rates can be improved.

6. REFERENCES

[Agg99] J.K. Aggarwal and Q. Cai: Human motion analysis: a review, CVIU, vol. 73(3), pp 428-440, 1999

[Bab03] R.V. Babu, K.R. Ramakrishnan: Compressed domain human motion recognition using motion history information, Proc. ICIP 2003, vol. 3, pp 321-324

[Bas05] F. Bashir, W. Qu, A. Khokhar and D. Schonfeld: HMM-based motion recognition system using segmented PCA, Proc. ICIP, 2005

[Ben02] C. BenAbdelkader, R. Cutler and L. Davis: Motion-based recognition of people in eigengait space, Proc. 5th IEEE Conf. on Automatic Face and Gesture Recognition, pp. 267, 2002

[Bot01] A. Bottino, A. Laurentini: A Silhouette-based technique for the Reconstruction of Human Movement. CVIU, Vol. 83, No. 1, pp.79-95, 2001

[Cao04] D. Cao, O. Masoud, N. Papanikolopoulos: Online Motion Classification using Support Vector Machines. Proc. Int. Conf on Robotics and Automation, 2004

[Dav01] J.W. Davis: Hierarchical motion history images for recognizing human motion. Proc. IEEE Workshop on Detection and Recognition of Events in Video, 2001, pp.39-46

[Del01] F. De la Torre, M. Black: Robust principal component analysis for computer vision,

Proceedings Int. Conf. on Computer Vision, ICCV 2001, Vancouver, BC, Vol. I, pp. 362-369.

[Gal01] A. Galata, N. Johnson and D. Hogg: Learning variable.lenght markov models of behaviour, Computer Vision and Image Processing, Vol. 81, no. 3, pp. 398-413, 2001

[Har00] I. Haritaoglu, D. Harwood, and L. S. Davis: W4: real time surveillance of people and their activities. IEEE Trans. PAMI, Vol. 22, No. 8, pp. 809-830, 2000.

[Hig03] J.Higgins, D. Molloy: Model Based Approach to Non-Intrusive Human Motion Capture, ISSC 2003, Limerick, July 1-2

[Hyu05] H. Yu, G. Sung, W. Song and X. Li: Human motion recognition based on neural network, Proc. IEEE Int. Conf. on Communications, Circuits and Systems, pp. 979-982, 2005

[Ime03] N. Imennov, S.L. Dockstader and M. Tekalp: A robust Bayesian network for articulated motion classification, Proc. ICIP 2003. Vol. 3, Pp. 305-308

[Kel05] V. Kellokumpu, M. Pietikäinen and J. Heikkilä: Human Activity Recognition Using Sequences of Postures, Proc. IAPR Conference on Machine Vision Applications (MVA 2005), Tsukuba Science City, Japan, 570-573

[Motek] Motek StockMovestm GENERIC, accessible at www.e-motek.com/entertainment/stockmoves/

[Nam97] J. Nam and A. Tewfik: Motion based video indexing using multi-resolution analysis. Proc. SPIE, vol 3309, pp. 688-696, 1997

[Nak01] H. Nakano, Y. Yoshida: Classifying human body motions using gabor features, Proc. International Conference on Image Processing, 2001, vol. 2, pp. 351 - 354

[Per02] J. Pers, S. Kovacic: Multi-scale recognition in squash match, Proc. ERK 2002, Portorož, Slovenija, pp. 295-298

[Rah05] M.M. Rahman, S. Ishikawa: Human motion recognition using an eigenspace, Pattern Recognition Letters 26 (2005) 687–697

[Rit99] J. Rittscher, A. Blake: Classification of human body motion, Proc. 7th IEEE International Conference on Computer Vision, 1999, pp:634 – 639

[Sun02] X. Sun, C. Chen and B.S. Manjunath: Probabilistic motion parameter models for human activity recognition, Proc. ICPR, 2002. Vol. 1, pp. 443 - 446

[Smo98] A.J. Smola, B. Schölkopf, K. Müller: The Connection between Regularization Operators and Support Vector Kernels, Neural Networks, vol. 11(4), 1998, pp. 637-649

[The03] C. Theobalt, J. Carranza, M. Magnor, H.P. Seidel: A Parallel Framework for Silhouette-based Human Motion Capture, in Proc. of Vision,

Modeling and Visualization, p.207-214, Munich, Germany, 2003

[Yac99] Y. Yacoob, M.J. Black: Parametrized modelling and recognition of activities, CVIU. 73(2), pp. 232-247, 1999

	Sequence frames (n)	Training step time (t)	Training sequences (S)	Eigenvector threshold (epsilon)	catalogation time average	true %	false %
exp50-x1-f1	1	1 / 30 sec	893	0,99	7,72 msec	53,98%	46,02%
exp50-x1-f3	3	1 / 30 sec	865	0,99	7,64 msec	58,15%	41,85%
exp50-x1-f5	5	1 / 30 sec	798	0,99	8,72 msec	88,41%	11,59%
exp50-x1-f10	10	1 / 30 sec	728	0,99	8,05 msec	87,42%	12,58%
exp50-x1-f15	15	1 / 30 sec	658	0,99	7,52 msec	87,30%	12,70%
exp50-x1-f20	20	1 / 30 sec	588	0,99	7,03 msec	87,56%	12,44%

Figure 7. Recognition rates vs. number of frames of a sequence

	Sequence frames (n)	Training step time (t)	Training sequences (S)	Eigenvector threshold (epsilon)	catalogation time average	true %	false %
exp50-x1-f5	5	1 / 30 sec	798	0,99	8,72 msec	88,41%	11,59%
exp50-x2-f5	5	2 / 30 sec	399	0,99	4,20 msec	89,01%	10,99%
exp50-x3-f5	5	3 / 30 sec	272	0,99	2,88 msec	88,23%	11,77%
exp50-x4-f5	5	4 / 30 sec	204	0,99	2,12 msec	89,67%	10,33%

Figure 8. Recognition rates vs. distance between sequences

	Sequence frames (n)	Training step time (t)	Training sequences (S)	Eigenvector threshold (epsilon)	catalogation time average	true %	false %
exp50-x1-f5-50p	5	1 / 30 sec	399	0,99	4,28 msec	77,24%	22,76%
exp50-x1-f5-75p	5	1 / 30 sec	603	0,99	6,28 msec	88,11%	11,89%
exp50-x1-f5-90p	5	1 / 30 sec	725	0,99	7,68 msec	87,93%	12,07%
exp50-x1-f5	5	1 / 30 sec	798	0,99	8,72 msec	88,41%	11,59%

Figure 9. Recognition rates vs. number of training sequences

	Sequence frames (n)	Training step time (t)	Training sequences (S)	Eigenvector threshold (epsilon)	catalogation time average	true %	false %
exp50-x1-f5-e80	5	1 / 30 sec	798	0,8	7,56 msec	76,64%	23,36%
exp50-x1-f5-e90	5	1 / 30 sec	798	0,9	7,48 msec	84,47%	15,53%
exp50-x1-f5-e99	5	1 / 30 sec	798	0,99	8,72 msec	88,41%	11,59%

Figure 10. Recognition rates vs. ε

In \ Out	Run	Wait	Walk	Rowing	Slips	Squat	Jumps
Run	96,37%	0,00%	0,00%	0,00%	0,00%	1,21%	2,42%
Wait	0,00%	100,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Walk	0,00%	0,00%	75,00%	0,00%	0,00%	0,00%	25,00%
Rowing	0,00%	0,00%	0,00%	100,00%	0,00%	0,00%	0,00%
Slips	0,00%	0,00%	0,27%	14,52%	81,64%	2,19%	1,37%
Squat	0,00%	0,00%	0,00%	2,03%	2,54%	95,43%	0,00%
Jumps	0,54%	0,00%	0,00%	0,00%	0,00%	27,72%	71,74%

Figure 11. Confusion matrix

	direct data		reconstructed data	
	true %	false %	true %	false %
exp50-x1-f5	88,41%	11,59%	88,11%	11,89%
exp50-x2-f5	88,11%	11,89%	88,89%	11,11%
exp50-x3-f5	88,23%	11,77%	87,63%	12,37%
exp50-x4-f5	89,67%	10,33%	89,31%	10,69%

Figure 12. Direct data vs. reconstructed data

	Sequence frames (n)	Training step time (t)	Training sequences (S)	Eigenvector threshold (epsilon)	catalogation time average	true %	false %
long1-x1-f5	5	1 / 30 sec	393	0,99	6,08 msec	79,41%	20,59%
long1-x1-f10	10	1 / 30 sec	353	0,99	5,97 msec	76,47%	23,53%
long1-x1-f15	15	1 / 30 sec	313	0,99	5,89 msec	75,59%	24,41%
long1-x1-f20	20	1 / 30 sec	273	0,99	5,83 msec	76,18%	23,82%
long2-x1-f5	5	1 / 30 sec	798	0,99	10,76 msec	82,65%	17,35%
long2-x1-f10	10	1 / 30 sec	728	0,99	10,76 msec	80,59%	19,41%
long2-x1-f15	15	1 / 30 sec	658	0,99	9,61 msec	79,41%	20,59%
long2-x1-f20	20	1 / 30 sec	588	0,99	9,61 msec	80,00%	20,00%

Figure 13. Reconstruction of complex sequences