Proceedings of the 17th IEEE International Symposium on Robot and Human Interactive Communication, Technische Universität München, Munich, Germany, August 1-3, 2008

Development of a 3D Real Time Gesture Recognition Methodology for Virtual Environment Control

Otniel Portillo-Rodriguez, Oscar O. Sandoval-Gonzalez, Carlo A. Avizzano, Emanuele Ruffaldi,
Davide Vercelli and Massimo Bergamasco

Abstract— **In this paper, we propose a real time 3D gesture recognition system that relies on the state based approach. The novelty of this work is the introduction of probabilistic neural networks (PNNs) to characterize the uncertain boundaries of each state. The 3D gestures are modeled as a sequence of states in a configuration space; the number of states and their spatial parameters are calculated by dynamic k-means clustering on the training data of the gesture without temporal information. Gesture recognition is performed using a simple Finite State Machine (FSM), where, each state transition depends only on the output of its corresponding PNN and optionally on its time restrictions (minimum and maximum time permitted in the state). If a recognizer reaches its final state, then it could be said that a gesture is recognized. The approach is illustrated with the implementation of a real time system that recognizes the semantic meaning of seven basic gestures of the Indian Dance, the description of the system and the technologies used, it will be described in detail in the paper.**

I. Introduction

Gesture recognition is the process by which the gestures made by the user are recognized by the receiver and is considered to be a higher-level task. A gesture may also be perceived by the environment as a compression technique for the information to be transmitted elsewhere and subsequently reconstructed by the receiver. Applications can be found in automatic robot control sign language recognition, virtual reality, computer games, objects manipulation, etc. [1].

Diverse approaches to handle gesture recognition have been performed [2], ranging from mathematical models based on Hidden Markov Models (HMMs) [3], to tools or approaches based in soft computing [4]. In addition to the theoretical aspects, any practical implementation of gesture recognition typically requires the use of different imaging and tracking devices such as: gloves, body suits, and marker based optical tracking, etc. Each sensing technology varies along several variables, including accuracy, resolution, latency, range of motion, user comfort and cost [5].

Gesture can be static (the user assumes a certain pose or configuration) or dynamic. Some gestures also have both static and dynamic elements, as sign language. While static gesture recognition can typically be accomplished by template matching [6, 7], standard pattern recognition [8], and neural networks (multilayer perceptron and radial basis function network) [9, 10]; the dynamic gesture recognition involves the use of techniques such as time-compressing templates [11], dynamic time warping [12, 13], HMMs [14] and time delay neural networks (TDNNs) and recurrent neural networks (RNNs) [15, 16, 17].

For human activity or recognition of dynamic gestures, most efforts have been concentrated on using state-space approaches [18] to understand the human motion sequences. Each posture state (static gesture) is defined as a state. These states are connected by certain probabilities. Any motion sequence as a composition of these static poses is considered a walking path going though various states. Cumulative probabilities are associated to each path, and the maximum value is selected as the criterion for classification of activities. Under such a scenario, duration of motion is no longer an issue because each state can repeatedly visit itself. However, approaches using these methods usually need intrinsic nonlinear models and do not have closed-form solutions. Nonlinear modeling also requires searching for a global optimum in the training process and a relative complex computing. Meanwhile, selecting the proper number of states and dimension of the feature vector to avoid "underfitting" or "overfitting" remains an issue.

State space models have been widely used to predict, estimate, and detect signals over a large variety of applications. One representative model is perhaps the HMM, which is a probabilistic technique for the study of discrete time series. HMMs have been very popular in speech recognition, but only recently they have been adopted for recognition of human motion sequences in computer vision [14]. HMMs are trained on data that are temporally aligned. Given a new gesture, HMM use dynamic programming to recognize the observation sequence [12].

The advantage of a state approach is that it doesn't need a large set of data in order to train the model. Bobick and Wilson [19] proposed an approach that models a gesture as a sequence of states in a

Manuscript received February 1st, 2008. This work was supported in part by ENACTIVE and SKILLS European Excellence Networks.

Otniel Portillo-Rodriguez, Oscar O. Sandoval-Gonzalez, Carlo Alberto Avizzano, Emanuele Ruffaldi, Davide Vercelli and Massimo Bergamasco are with Perceptual Robotics Laboratory (PERCRO), Scuola Superiore Sant'Anna, Pisa, Italy (phone: +39 050-883-771; fax: +39 050-883-874; e-mail: otniel@ sssup.it).

configuration space. The training gesture data is first manually segmented and temporally aligned. A prototype curve is used to represent the data, and is parameterized according to a manually chosen arc length. Each segment of the prototype is used to define a fuzzy state, representing transversal through that phase of the gesture. Recognition is done by using dynamic programming technique to compute the average combined membership for a gesture.

Learning and recognizing 3D gestures is difficult since the position of data sampled from the trajectory of any given gesture varies from instance to instance. There are many reasons for this, such as sampling frequency, tracking errors or noise, and, most notably, human variation in performing the gesture, both temporally and spatially. Many conventional gesture-modeling techniques require labor-intensive data segmentation and alignment work.

The attempt of this work is develop a useful technique to segment and align data automatically, without involving exhaustive manual labor, at the same time, the representation used by our method captures the variance of gestures in spatial-temporal space.

In our approach, we have modeled 3D gestures as sequences of states in a configuration space. For each gesture, the number of its states and their coarse spatial parameters are calculated by dynamic k-means clustering on the training data of the gesture without temporal information. Each gesture performed online is recognized at frame rate using a simple Finite State Machine (FSM), its states' transitions are based on the current state estimated by one PNN and optionally of its time restrictions (minimum and maximum time permitted in each state). If a recognizer reaches its final state then the algorithm concludes that a gesture is recognized.

Training is done offline using several examples of each gesture, repeated continuously by the user when was requested, as a training data. After learning the spatial information, data segmentation and alignment become easy. The temporal information from the segmented data could be added to the states. The spatial information is also updated. This produces the state sequence that represents the gesture.

The rest of the paper is organized as follows: section II presents the basis of the probabilistic neural networks. Section III presents the description of operation of the online recognition system as well as the variables used. Finally in the section IV, the conclusions of this research are presented.

II. Probabilistics Neural Networks

The PNN is a Bayes–Parzen classifier [20] and is a special form of radial basis function (RBF) network used for classification. The foundation of the approach is well known decades ago (1960s); however, the method was not of widespread use due to the lack of sufficient computation power . The PNN was first introduced by Specht [21], who showed how the Bayes–Parzen classifier could be broken up into a large number of simple processes implemented in a multilayer neural network each of which could be run independently in parallel.

A PNN often learn more quickly than many neural network models such as backpropagation networks, and have had success on a variety of applications [22].

The network learns from a training set **T**, which is a collection of examples called instances. Each instance *i* has an input vector $\boldsymbol{y}_i$, and an output class, denoted as $class_i$. During execution, the network receives additional input vectors, denoted as $\boldsymbol{x}$, and outputs the class that $\boldsymbol{x}$ seems most likely to belong to.

In Fig. 1, a classical probabilistic neural network with four hidden neurons and two class nodes is shown to explain how they work. The first (leftmost) layer contains one input node for each input attribute in an application. All connections in the network have a weight of 1, which means that the input vector is passed directly to each hidden node. There is one hidden node for each training instance *i* in the training set. Each hidden node h_i has a spread factor, σ_i, which determines the size of its receptive field.

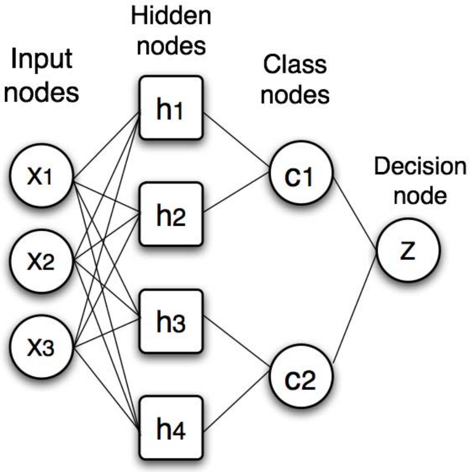

Fig. 1. Probabilistic Neural Network.

A hidden node receives an input vector $\boldsymbol{x}$ and outputs an activation given by the Gaussian function *g*, which returns a value of 1 if $\boldsymbol{x}$ and $\boldsymbol{y}_i$ are equal, and drops to an insignificant value as the distance grows:

$$g(\boldsymbol{x},\boldsymbol{y}_i,\sigma_i)=\exp[-D^2(\boldsymbol{x},\boldsymbol{y}_i)/2\sigma_i^2] \quad (1)$$

The distance function *D* determines how far apart the two vectors are. By far the most common distance function used in PNN's is Euclidean distance. Each hidden node h_i in the network is connected to a single class node. If the output class of instance i is j, then h_i is connected to class node c_j. Each *class node* c_j computes the sum of the activations of the hidden nodes that are connected to it (i.e., all the hidden nodes for a particular class) and passes this sum to a decision node. The decision node outputs the class with the highest summed activation.

One of the greatest advantages of this network is that it does not require any iterative training; it trains quickly since the training is done in one pass of each training vector, rather than several. However, one of the main disadvantages of this network is that it has one hidden node for each training instance and thus requires more computational resources (storage and time) during execution than other models. When is simulated on a serial machine, O(n) time is required to classify a single input vector. On a parallel system, only O(log n) time is required, but n nodes and nm connections are still required (where n is the number of instances in the training set, and m is the number of input attributes).

PNN can be used in real time because as soon as one input vector representing each class has been observed, the network can begin to generalize to new input vectors. As additional input vectors are observed and stored in the net, the generalization will improve and the decision boundary can get more complex. PNN saves the statistic feature of the training pattern as the weights between the input layer and the hidden layer, but it is not designed for spatiotemporal pattern recognition.

III. Methodology To Recognize 3D Gestures using The State Based Approach

Our work aim at extracting a representation of the data that encapsulates only the key aspect of the gesture and discard the intrinsic variability to each person's movements. Recognition and generalization is spanned from very small dataset, we have asked to the expert to reproduce just five examples of each gesture to be recognized.

The principal problem to model a gesture is the characterization of the optimal number of states and the establishment of their boundaries. For each gesture, the training data is obtained concatening the data of its five demonstrations. To define the number of states and their coarse spatial parameters we have used dynamic k-means clustering on the training data of the gesture without temporal information (see [23]). The temporal information from the segmented data is added to the states and finally the spatial information is updated. This produces the state sequence that represents the gesture. The analysis and recognition of this sequence is performed using a simple Finite State Machine (FSM), instead of use complex transitions conditions as in [24], the transitions depend only of the correct sequence of states for the gesture to be recognized and eventually of time restrictions i.e., minimum and maximum time permitted in a given state.

The novel idea is to use for each gesture a PNN to evaluate which is the nearest state (centroid in the configuration state) to the current input vector that represents the user's body position. The input layer has the same number of neurons as the input vector and the second layer has the same quantity of hidden neurons as states have the gesture. The main idea is to use the states' centroids obtained from the dynamic k-means as weights in its correspondent hidden neuron, in a parallel way where all the hidden neurons computes the similarities of the current student position and its corresponding state. In our architecture, each class node is connected just to one hidden neuron and the number of states in which the gesture is described defines the quantity of class nodes. Finally, the last layer, a decision network computes the class (state) with the highest summed activation.

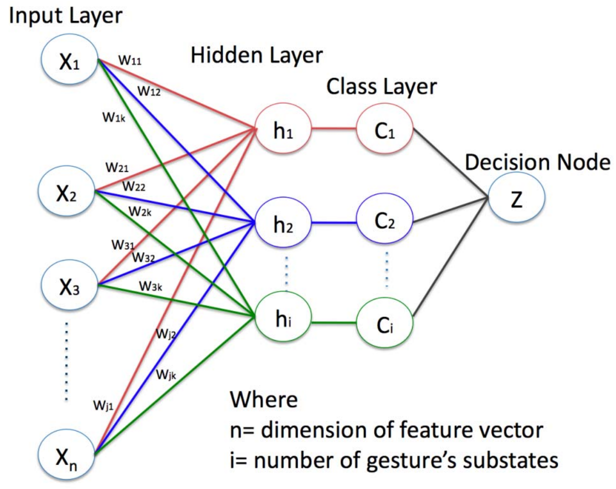

Fig. 2. PNN architecture used to estimate the most similar gesture's state from the current user's body position.

This approach allows real-time recognition while avoiding the classical disadvantages of this network: big computational resources (storage and time) during execution than many other models. An alternative for such computation is the use of RNN. In [25] the authors tested the use of RNNs for motion recogniton, according to their results, more than 500 nodes and more than 200,000 weight parameters between each node are needed in order to integrate the memorization process in the RNN. The RNN consist of motion elements neurons, symbol representation neurons and buffer neurons for treating time-series data. The required number of weights increases in proportion to the square of the number of all nodes. On the contrary in our methodology the number of parameters is proportional to the product of the number of hidden nodes (states in the gesture) and the dimension of the feature vector. To give a concrete example, a gesture typically has 10 states and the dimension of the feature vector is 13 (Section IV), resulting that with only 130 parameters a gesture is modeled, given as result a high information compression ratio.

The construction of the finite state machine is fast and simple; the transitions depend only of the correct sequence of states for the gesture to be recognized and eventually of time restrictions. If the FSM reaches its final state then the algorithm concludes that a gesture is recognized.

IV. Implementation of a Real Time Recognition System of The Indian Dance

We have implemented a system that recognizes seven basic movements (temporal gestures) of the Indian Dance. Each one has a meaning, thus, the scope of our system is to discover if the user/dancer has performed a valid known movement in order to translate its meaning in a artistic representation using graphics and sounds.

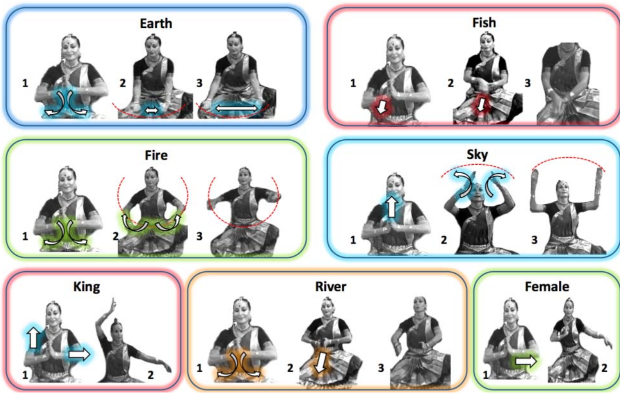

Fig. 3. The seven Indian Dance movements that our system recognizes.

The interaction with the user is simple and easy to learn (Fig. 4). At the beginning the user remains in front of the principal screen in a motionless state. Once the system has sensed the user's inactivity, it sends a message indicating that it is ready to capture the movement of the user. If the system recognizes the movement, it renders a sound and image corresponding to its meaning. After that, the system is again ready to capture a new movement. The hardware architecture of our system is composed of four components: VICON capture system (VCS), a host PC (with processor of 3 Ghz Intel Core Duo and 2 gigabytes of RAM memory), one video projector and sound system. In the host PC, three applications run in parallel: Vicon Nexus [26], Matlab Simulink® and XVR [27]. In the next paragraphs the operation of the recognition system it will be described using the Fig. 5 as reference.

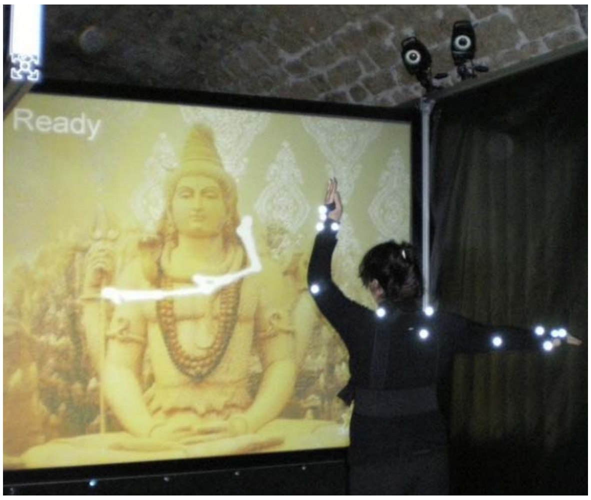

Fig. 4. The recognition system in action. It is possible observe how the cameras track the user movements using the reflective markers that are mounted on the suit dressed by the user. The user's avatar and the image that represents the meaning (a King) of the recognized gesture are displayed in the screen.

Eight cameras and their electronics acquisition unit compose the VCS; whose function is to acquire the 3D positions of reflective markers attached to a suit that the user dresses on its upper limbs and send the positions to Simulink through UDP protocol. In order to track the markers is necessary create the kinematics model of the user's upper limbs. The model is shown in Fig. 6; it is composed of 13 markers united for hinge and balls joints.

With Matlab Simulink's Real-Time Windows Target, we have developed a real-time recognition system with a sample rate of 50 Hz. Its operation at each frame rate it's as follows: the current 3D position of the markers is read from Nexus, then the data are filtered to avoid false inputs (sometimes VCS loses the position of the markers given zero inputs), then, the data are send simultaneously to XVR (to render a virtual avatar) and other block that converts the 3D positions of the 13 markers (vector of 39 elements) in a normalized feature vector (values from 0 to 1). The feature vector is composed by 13 elements:

1) Right & left elbows angles
2) Right & left wrists pitch angles
3) Right & left Shoulders angles
4) The magnitude of the distance between palms
5) Three spatial vector components from the back to the right palm
6) Three spatial vector components from the back to the left palm.

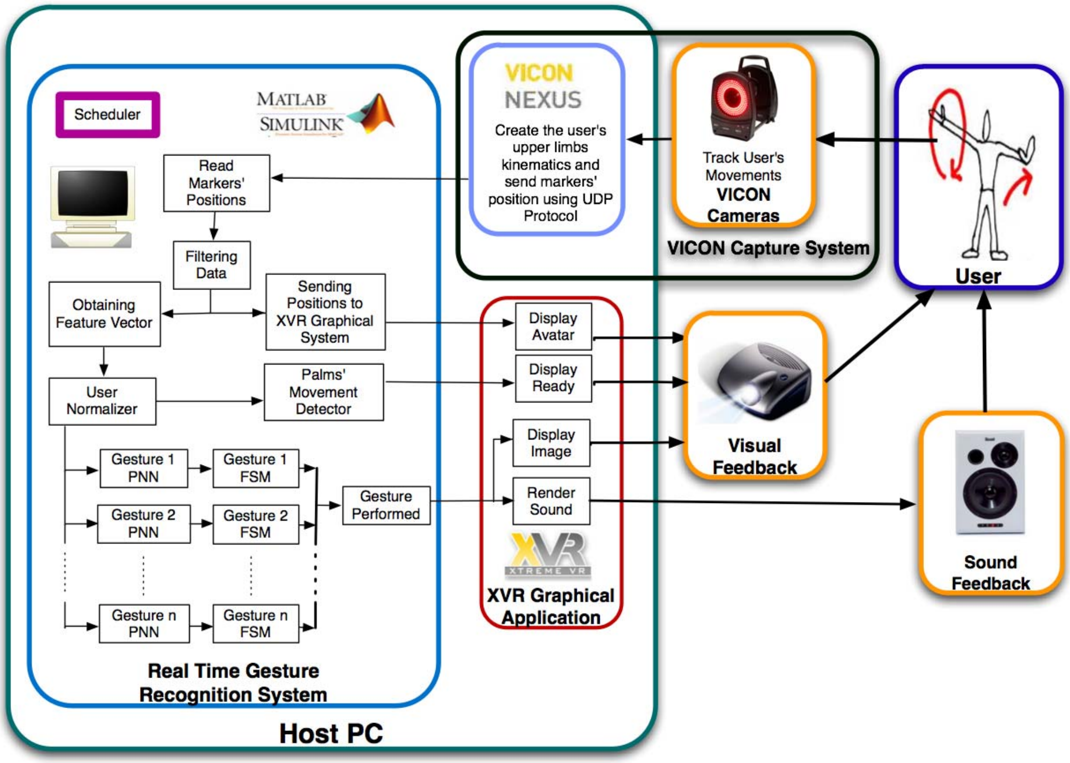

Fig. 5. Architecture of the Recognition System

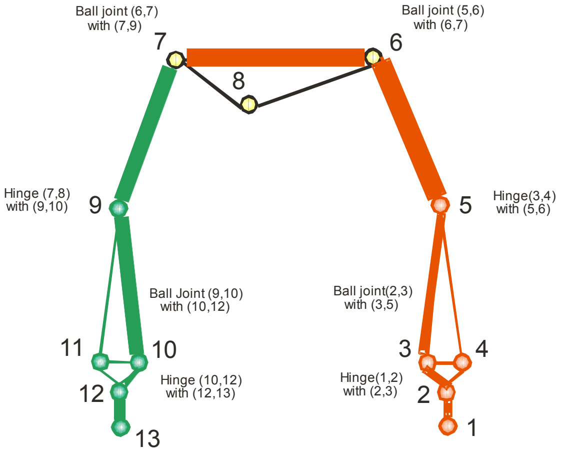
Ball joint (6,7) with (7,9)
Ball joint (5,6) with (6,7)
7
6
8
Hinge (7,8) with (9,10)
9
5
Hinge(3,4) with (5,6)
Ball Joint (9,10) with (10,12)
Ball joint(2,3) with (3,5)
11
10
3
4
Hinge (10,12) with (12,13)
Hinge(1,2) with (2,3)
12
2
13
1

Fig. 6. Kinematics for the upper limbs based on the marker placement on the arms and hands

The chosen elements are invariant to the position and orientation of the dancer inside the capture system's workspace; allowing to have enough degrees of freedom to model the movements of the Indian dance without over-fitting.

The movements of both palms are analyzed in order to determine if the user is or is not in motion, this information is useful in order to interact correctly with the user.

Our system recognizes movements of different users independently of their body sizes. It is important mention that the training data is obtained from the movements of a single person. In order to recognized movements from different persons, it is necessary normalize the data to the "pattern user". The normalization relies in the fact that in the gestures to be recognized, the ensemble of angles of the feature vector have approximately the same temporal behavior and their range of values are similar for different users independently of their body sizes due their arms can be seen as kinematics chains with the same joint variables with different lengths. The key idea is normalize just the components of the feature vector that involves distances. The normalize factor is obtained each time that a new user interact with the system measuring the length of his/her right arm.

Then, the modified feature vector is sent simultaneously to a group of PNN–FSM couples that work in parallel. Each couple is used to recognize one gesture, the PNN is used to determinate which is the nearest state respect to the current body position for one particular gesture. Recognition is performed using a FSM, where, its state transitions depend only of the output of the its PNN. If the FSM arrives to its final state the gesture is recognized. All the FSM are initialized when no motion of the palms is detected.

XVR, a virtual environment development platform is used to display visual and sound information to the user.

An avatar shows the users' movements, a silhouette is used to render the sensation of performing movements along the time. The graphical application also indicates when is ready to recognized a new movement. If the recognition system has detected a valid gesture, the meaning of the gesture recognized is displayed to the user using an appropriate image and sound (Fig. 7).

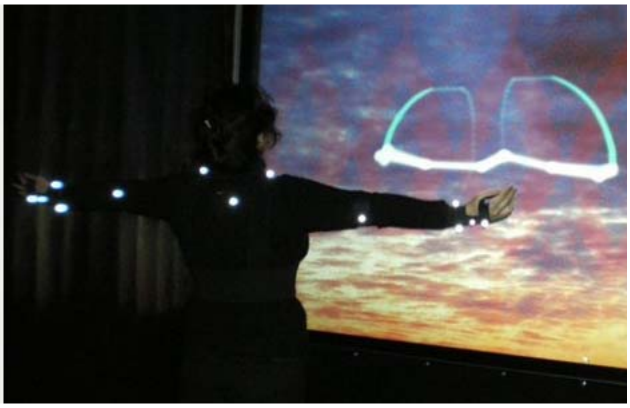

Fig. 7. Sky gesture has been recognized. The user can see the avatar, hear the sound of a storm and see the picture of the firmament.

IV. Conclusions

In this paper a new methodology to recognize 3D gestures using the state based approach has been developed, based on the efficacy of the PNNs to calculate the boundaries of each gesture's state and its capability to manage the transitions of its corresponding FSM. For each gesture, the methodology, automatically obtains the optimal number of states, which are characterized by their centroids. With them a PNN is trained fast and accurately, containing only the minimum number of neurons, reducing the computational cost in its real time implementation

The online recognition system of seven different basic movements of the Indian dance, shown the efficacy of our methodology, which can be applied to different areas of the Human Computer Interaction.

References

[1] D. M. Gavrila. *The visual analysis of human movement: A survey*. Comput. Vis. Image Understanding, vol. 73, pp. 82, 1999.

[2] V. I. Pavlovic, R. Sharma and T. S. Huang. *Visual interpretation of hand gestures for human computer interaction*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 19, pp. 677, Jul. 1997.

[3] L. R. Rabiner. *A tutorial on hidden Markov models and selected applications in speech recognition*. *Proc. IEEE*, vol. 77, pp. 257, Feb. 1989.

[4] S. Mitra and T. Acharya. *Data Mining: Multimedia, Soft Computing, and Bioinformatics*. New York: Wiley, 2003.

[5] Pantic, M., Pentland, A., Nijholt, A., and Huang, T. *Human computing and machine understanding of human behavior: a survey*. Proceedings of the 8th international Conference on Multimodal interfaces. Banff, Alberta, Canada, November 2006.

[6] A. F. Bobick and J. Davis. *Real-time recognition of activity using temporal templates*. In Proc. of IEEE Computer Society Workshop Applications on Computer Vision, pages 39-42, Sarasota, FL, 1996.

[7] Polana, R. Nelson, R. *Low level recognition of human motion (or how to get your manwithout finding his body parts)*. Proceedings of the IEEE Workshop on Motion of Non-Rigid and Articulated Objects, 1994.

[8] Gang Qian Feng Guo Ingalls, T. Olson, L. James, J. Rikakis, T. A gesture-driven multimodal interactive dance system. IEEE International Conference on Multimedia and Expo 2004. Taipei, Taiwan. June 2004.

[9] Tabb K, Davey N, Adams R & George S. Omni-directional Motion: Pedestrian Shape Classification using Neural Networks and active Contour Models. Image & Vision Computing conference. 387-392. New Zealand, Nov 2001.

[10] Daw-Tung Lin. *Spatio-temporal hand gesture recognition using neural networks*. IEEE World Congress on Computational Intelligence. Anchorage, AK, USA. May 1998.

[11] Bobick, A.F. Davis, J.W. *The recognition of human movement using temporal template*s. IEEE Transactions on Pattern Analysis and Machine Intelligence. On page(s): 257-267. Volume: 23, Issue: 3. Mar 2001.

[12] Richard Bellman. *Dynamic Programming*. Princeton University Press. 2003.

[13] Lementec, J.C. and Bajcsy, P. *Recognition of arm gestures using multiple orientation sensors: gesture classification*. 7th International IEEE Conference on Intelligent Transportation Systems, 2004.

[14] J. Yamato, J. Ohya and K. Ishii. *Recognizing human action in time-sequential images using Hidden Markov Model*. In Proc. IEEE Conference CPVPR, pages 379-385, Champaign, IL, June 1992.

[15] M. S. Yang and N. Ahuja. Recognizing hand gesture using motion trajectories. Proc. IEEE CS Conf. Comput. Vis. Pattern Recogn. Fort Collins. Jun. 1998, p. 466.

[16] Kouichi Murakami and Hitomi Taguchi. *Gesture Recognition using Recurrent Neural Networks*. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. New Orleans, Louisiana, United States. May, 1991.

[17] Vamplew, P. Adams, A. *Recognition and anticipation of hand motions using a recurrent neural network*. IEEE International Conference on Neural Networks, 1995.

[18] A. F. Bobick and A. D. Wilson. *A state-based technique for the summarization and recognition of gesture*. In Proc. of 5th International Conference on Computer Vision, pages 382-388, 1995.

[19] A. F. Bobick and A. D. Wilson. *A state-based approach to the representation and recognition of gesture*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 19, pp. 1235, Dec. 1997.

[20] E. Parzen. *On estimation of a probability density function and mode*. Annals of Mathematical Statistics 36 (1962), pp. 1065–1076.

[21] D.F. Specht. *Probabilistic neural networks for classification, mapping or associative memory*. IEEE International Conference on Neural Networks. 1988.

[22] D.F. Specht. *Probabilistic neural networks*. Neural Networks. Vol. 3 (1990), pp. 109–118.

[23] A.K. Jain, M. N. Murty, P.J. Flynn. *Data Clustering: A Review*. ACM Computing Surveys, Vol. 31, No. 3, September 1999.

[24] Pengyo Hong, Matthew Turk and Thomas S. Huang. *Gesture Modeling and Recognitition Using Finite State Machines*. Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition. 2000.

[25] Tetsunari Inamura, Yoshihiko Nakamura and Moriaki Shimozaki. *Associative Computational Model of Mirror Neurons that connects mising link betwwen behavior and symbols*. Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems. EPLF, Lausanne, Switzerland. October 2002.

[26] Vicon web site. Viewed 29 January 2008, http://www.vicon.com

[27] VRMedia – eXtremeVR: virtual reality on the web, viewed 29 January 2008, http://www.vrmedia.it