Proc. IEEE-INNS-ENNS Internat. Joint Conference on Neural Networks (IJCNN 2000), Como, Italy, 2000, pp. IV 133-138, IEEE Computer Society Press 2000

Camera-based Gesture Recognition for Robot Control

Andrea Corradini, Horst-Michael Gross
Technical University of Ilmenau
Department of Neuroinformatics
D-98684 Ilmenau, Federal Republic of Germany
andreac@informatik.tu-ilmenau.de

Abstract

Several systems for automatic gesture recognition have been developed using different strategies and approaches. In these systems the recognition engine is mainly based on three algorithms: dynamic pattern matching, statistical classification, and neural networks (NN).

In that paper we present four architectures for gesture-based interaction between a human being and an autonomous mobile robot using the above mentioned techniques or a hybrid combination of them. Each of our gesture recognition architecture consists of a preprocessor *and a* decoder. *The preprocessor, which is common to every system, receives an image as input and produces a continuous feature vector. The task of the decoder is to decode a sequence of these vectors into an estimate of the underlying movement. In the first three systems to determine that estimate, we formally consider the recognition problem as a statistical classification task. Three different hybrid stochastic/connectionist architectures are considered. In the first approach NNs are used for the classification of single feature vectors while Hidden Markov Models (HMM) for the modeling of sequences of them. In the second a Radial Basis Function (RBF) network is directly used to compute the HMM state observation probabilities. In the third system that probabilities is calculated by means of recurrent neural networks (RNN) in order to take into account the context information from the previously presented feature vectors.*

In the last system we face the recognition task as a template matching problem by making use of dynamic programming techniques. Here the strategy is to find the minimal distance between a continuous input feature sequence and the classes.

Preliminary experiments with our baseline systems achieved a recognition accuracy up to 92%. All systems use input from a monocular color video camera, are user-independent but so far, they are not yet real-time.

1 Introduction

Visual-based automatic gesture recognition has recently acquired much attention. In this context strong efforts have been carried out to develop intelligent and natural interfaces between users and computer systems based on body movements. The operational area of such intelligent interfaces covers a broad range of application fields in which an arbitrary system is to be controlled by an external user or in which system and user have to interact immediately [9,10]. These interfaces not only substitute the common interface devices but also can be exploited to extend their functionality. Especially for the interaction between a mobile system and a user the visual communication is very important because it gives the system the capability to observe its operational environment in an active manner.

Our superior longterm goal is to develop an intelligent system for an autonomous mobile robot able to act in a supermarket environment. The robot should be capable to get into contact with a customer, to follow, lead, support and interact with him/her. Considering now that the customers of the same supermarket can belong to very different cultures and social layers the use of few natural and cross-cultural gestures for robot command is mandatory. In addition, the optimal interaction has to be familiar and intuitive, just like the interaction between people. The commands to be used have to be highly instructive, everybody should be able as well to understand as to perform them without requiring any remembrance.

One of the crucial problems in automatic gesture recognition is to deal with the varying temporal structure of dynamic gestures. The difficulty of gesture recognition stems from the high variability of each movement associated with a gesture to

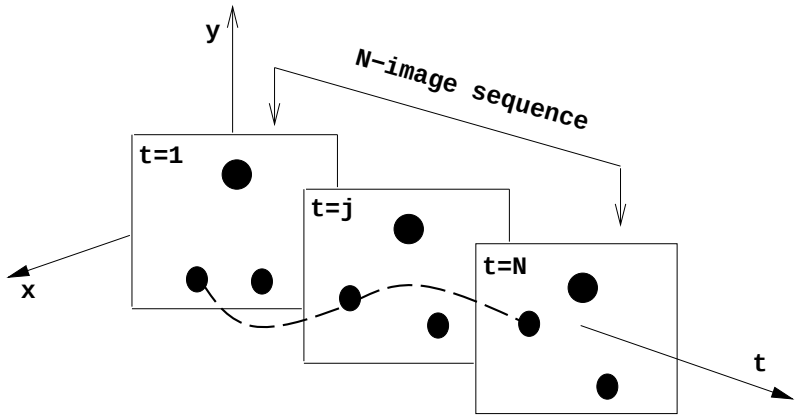

Figure 1. *Gesture as hand and head trajectories within the feature space extended along the time.*

be detect. Gesture's segments may overlap, have varying lengths, and vary across speakers. Even the same user is not ever able to produce exactly the same movement for the same gesture. Moreover the complexity of the automatic recognition task is related to robustness to environmental conditions, vocabulary size, number and movement characteristics of users in user independent recognizers, real-time operational capability and so on.

This paper is structured as follows. Starting from our saliency system for person localization [4], in Sec. 2 we provide an overview of the process which is to be carried out to describe the user's postures. Sec. 3 presents the recognition engines employed. Finally, the last Section reports the preliminary results achieved with the systems, and contains as well conclusions as suggestions for future work.

Throughout this paper the following definitions are considered.

Definition 1 (Posture/Pose) *A posture or pose is a couple determined by the only static hand locations with respect to the head position. The spatial relation of face and hands determines the behavioral meaning of each posture.*

Definition 2 (Gesture) *A gesture is a series of postures over a time span connected by motions.*

2 Feature Extraction

We think that a good person localization task is essential for any further gesture recognition process. In an our previous work [4] we proposed a multi-cue approach consisting of three feature modules sensitive to *skin color*, *facial structure* and *structure of the head-shoulder-contour*, respectively. Due to its reliability and robustness against varying environmental circumstances, that system represents our starting point for any further precessing step. After detecting the location of the head and considering a subregion around it, we characterize the distribution of the pixel values inside that window by a multidimensional normal distribution function representing a parametric model for the skin color. Using the Mahalanobis distance between that distribution and an image pixel, this latter is classified to be or not a member of the skin class.

From the resulting binary image we determine the centers of gravity (COG) of the hand and head regions (which we suppose to be the three greatest ones) and further we model each of these regions as a circle around their COGs (Fig. 1). From that binary image we compute a feature vector $\vec{v}$ containing 14 translation and scale invariant elements characterizing the shape of the segmented scene. The goal of the posture analysis is the extraction of local features along the hand trajectory, yielding a sequence of time ordered multi-dimensional feature vectors. See [5] for more details on the pose segmentation and the feature extraction task.

After that we process the whole training set. Because the feature vector components have values which differ by several orders of magnitude we proceed with a rescaling of them. We perform the *whitening* linear rescaling [2] with respect to the test patterns. This procedure does not treat the input variables independently but allows for correlations amongst the variables. In the transformed coordinates the data set has zero mean and a unit covariance matrix.

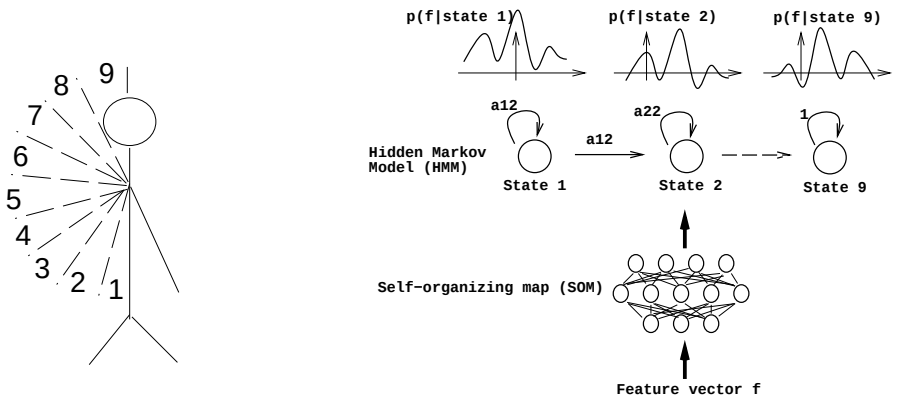

Figure 2. *From left to right: the waving-right gesture is divided into 9 labeled subregions each covering about 20 grads of the two-dimensional image surface, and SOM/HMM system overview showing how each HMM state is responsible for one gesture's subregion.*

3 Gesture Recognition Systems

3.1 Hybrid SOM/discrete HMM

Given an input feature sequence the further step is concerned with the quantization of that feature vectors into a sequence of symbols.

To do that a self-organizing map (SOM) is used to preserve the topology of the high-dimensional feature space by mapping the feature vectors to a two-dimensional space. Due to the sequential nature underlying each gesture such a topology-preserving map can be exploited to constitute trajectories where the SOM best-matching neurons are recorded during the process. The SOM clusters the unlabeled training feature vectors which lie near one other in the feature space. As well the codebook vector most sensitive to the actual training vector as those in its time-variable neighborhood, are tuned maintaining a well-balanced set of weight values with respect to the input density function.

In the training phase the weight adjustment is carried out using the Euclidean distance between the actual 14-dimensional input vector and the connecting weight vectors, times a time-dependent learning rate. We start the learning process with a large radius covering all the units in order to prevent the formation of undesired outliers in the clustering due to the limited training data set. During the training we decrease the neighborhood radius up to 1 and the learning rate from 0.9 to 0 in $(100 * xsize * ysize)$ iterations. Our SOM consists of 800 units organized into a $(xsize = 40 \times ysize = 20)$ square array.

In order to utilize the SOM for classification we divide each gesture of our vocabulary in *subgestures* (see e.g. Fig. 2 for the waving-right movement). We divide the gesture of our vocabulary into altogether 32 subgestures/symbols (9 for each left-,right-waving; 5 for each go left/right; 4 for stop). For class discrimination purposes we label each SOM clusters. That labels were assigned to the units according to the subgesture subdivision (Fig. 2) by using hand-labeled training samples as input.

The need of a vector quantizer to map the continuous observation vectors into discrete codebook symbols arises from the use of HMMs [7] with discrete observation symbols as recognizer for symbol sequences deriving from time-sequential images. For each movement to be detected, we create one left-to-right discrete HMM with as many states as the subregions which this gesture is divided in (Fig. 2). In the learning phase the HMM parameters are optimized in order to model the training symbol sequences from the corresponding gesture. The recognition phase consists in comparing a given sequence of symbols with each HMM. The gesture associated with the model which best matches the observed symbol sequence is chosen as the recognized movement.

To estimate the parameters of the discrete HMMs we use the Baum-Welch reestimation method [1] which is based on the maximum likelihood criterion [2], aiming at maximizing the probability of the samples given the model at hand.

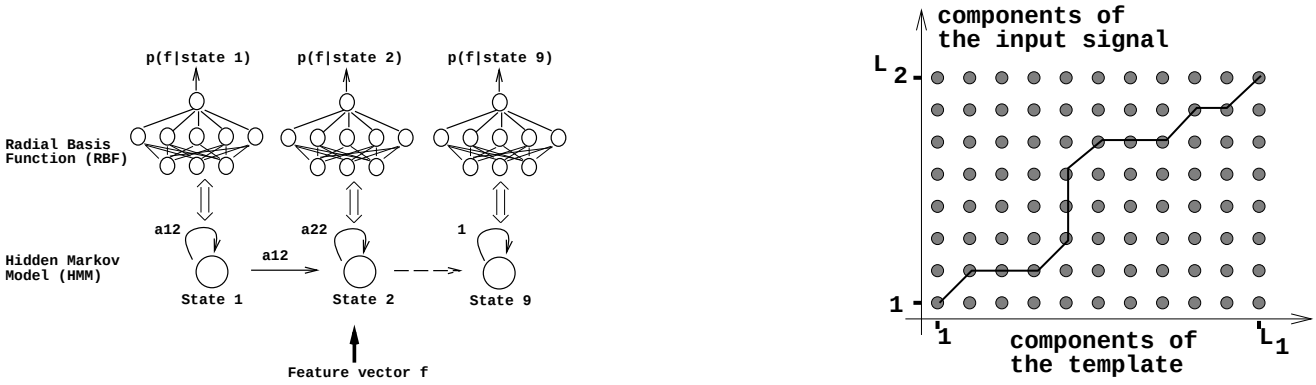

Figure 3. *From left to right: HMM/Neural Networks hybrid architecture overview, and example of DTW time alignment and normalization of two pattern sequences of length L_1 and L_2, respectively.*

3.2 Hybrid continuous HMM/RBF

In the second approach we consider the use of RBF networks for HMM state probability estimation (Fig. 3). In our RBF network the number of output neurons N_O is determined by the number of subgestures and therefore is as equal as the HMM state's number. As basis functions we choose $N_G = 5 \times \#subgestures$ Gaussian probability distribution functions each with own mean vector and different covariance matrix.

Our data set consists of input vectors $\mathbf{v}$, together with binary vector targets $\mathbf{t}$ whose j-th element alone is set to 1 according to the label of the corresponding subgesture (see Sec. 3.1). Absorbing as well the bias parameters as the normalizing factors of the Gaussian functions into the weights, and remembering the output has to represent a probability value, the output of the l-th node is given by

$$y_l(\mathbf{x}) = \frac{\sum_{j=0}^{N_G} w_{jl} \exp\{{}^t(\mathbf{x}-\mu_j)\mathbf{\Sigma}_j^{-1}(\mathbf{x}-\mu_j)\}}{\sum_{u=1}^{N_O}\sum_{j=0}^{N_G} w_{ju} \exp\{{}^t(\mathbf{x}-\mu_j)\mathbf{\Sigma}_j^{-1}(\mathbf{x}-\mu_j)\}} \tag{1}$$

A bias which extra basis function whose activation is set to 1, is included in the hidden layer. The determination of suitable parameters of the basis functions is accomplished by the iterative k-means clustering algorithm to more accurately reflect the training data distribution. Obviously the number of clusters corresponds to that of the basis functions.

After determining that parameters for each Gaussian, they are kept fixed while the weights of the second layer are found out by using a gradient descent technique. The RBF networks and the corresponding HMM are not trained jointly: we first train the RBF networks and then we apply the Baum-Welch reestimation algorithm [1] to determine the HMM parameters.

3.3 Hybrid continuous HMM/RNN

In that approach we consider the use of a Jordan network with an additional time window in the input layer as in the Time Delay Neural Network (TDNN) [11] for HMM state probability estimation (Fig. 3). That neural network is structured as following. The input layer is divided into two parts: the context units and the actual and last N input vectors. The context units hold a copy of the activations of the output layer from the previous time step and also from themselves (Fig. 4). That recurrence permits the network to remember some aspects of the most recent past giving the network some memory. At a given time t the network state depends as well on the current input as on an aggregate of past values. Because the feed-back connections are fixed they do not perceptibly complicate the training which can be easily performed by the backpropagation algorithm.

Our network consists of as many output neurons as the subgestures and therefore is as equal as the HMM state's number. Each neuron calculate the output probability for the underlying HMM state. We choose a time window of length $N = 5$. Because the output values are to be interpreted as probabilities they must sum to unity and lie in the range $(0, 1)$. This can be easily achieved by using the softmax activation function [8]. The hidden neurons simply compute the sigmoid activation function.

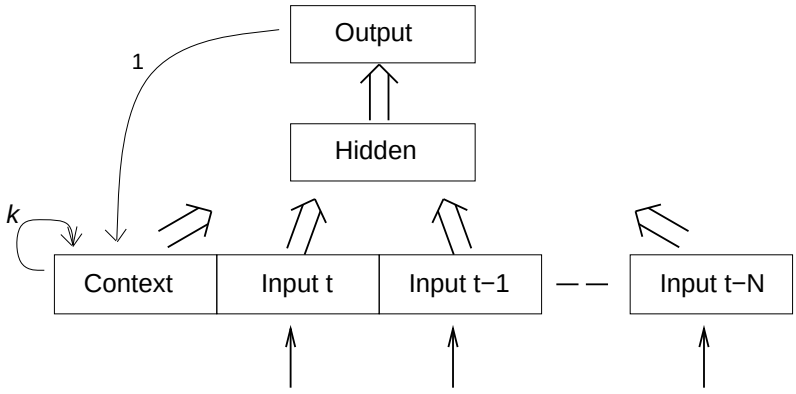

Figure 4. *One state of the HMM/RNN hybrid architecture. The output layer consists of only one neuron with softmax activation function.*

The RNN is trained in classification mode. Our data set consists of sequence of input vectors, together with one binary vector target whose j-th element alone is set to 1 according to the label of the expected subgesture (see Sec. 3.1) after the given sequence.

As in the previous system the neural networks and the corresponding HMM are not trained jointly: we first train the RNN networks and then we apply the Baum-Welch reestimation algorithm [1] to determine the HMM parameters.

3.4 Template Matching using DTW

Considering a gesture as feature vector sequence one can think to define one representative sequence template for every movement to detect and then find the minimal distance between these representants and a new input sequence. The input signal is further classified as belonging to the class whose representant is the nearest to it according to the choice of the distance function.

Some problems arise from such an approach. How many and how do we choose the class representants ? And how do we compute the distance between two signals allowing them to have different length one other ?

Concerning the first question and considering the training data of a given gesture, we simply calculate its mean sequence length and take the average sequence vector over the sequences with length equals to that mean value. We determine exactly one template for every movement class. As solution to the second problem we turn on to use the DTW algorithm with local constraints on path specification of Type I [6]. The DTW performs a time alignment and normalization by computing a temporal transformation function allowing two signals to be matched. Given two signals to compare, if we consider a table having the signals in the first row and column, respectively, that temporal function can be seen as a path in the table (Fig 3). The global path cost (locally accumulated over the time) represents the dissimilarity between the signals while the template signal with the more little path cost is the closest from the input.

4 Preliminary Results and Future Work

To train and test each model we gathered the data from five people performing 45 repetitions of each gesture to be recognized. The categories to be recognized were five.

The performances were captured by a color camera (25 frames/second) and digitized into 120×90 pixel RGB images. Table 1 summarizes the achieved performance concerning the recognition task. Associated to each system there is an acceptance threshold. Considering the hybrid approaches, an input is not classified if after feeding it into each HMM either the difference between the highest and the second highest output is not over that heuristically determined threshold or all the outputs are under its value. With the template matching technique we act in the same way but considering the two minimal distances from the input signal. The table does not show the recognition rates concerning the hybrid HMM/RNN because we are currently testing and improving it.

The performance of the systems depend not only on the number of training patterns but also how well that patterns are representative for each class. It means that the training patterns have to cover the maximum test pattern range as possible. In

Gesture	Recognition rate in %			% false class. gestures		
	SOM/HMM	HMM/RBF	DTW	SOM/HMM	HMM/RBF	DTW
stop	83.6	84.2	76.2	6.4	5.2	6.6
hello right	85.4	88.9	76.2	7.1	4.3	7.4
hello left	85.0	88.5	75.0	7.4	4.7	7.0
go right	85.5	87.2	74.6	6.3	5.0	6.0
go left	84.2	91.2	77.3	7.4	4.4	6.1

Table 1. Recognition results with the different architectures.

spite of our experimental results we do not state that the HMM/RBF-based system *always* outperforms the other ones. Due to the limited training data it would be a shaky conclusion strongly dependent from the implementation and the few data at the hand.

Anyway the recognition rate of the hybrid system HMM/RBF, up to now promising, can be improved by using a discriminative training algorithm instead of the Baum-Welch algorithm giving arise to a poor discriminative power among different models [3,7]. We think that also a jointly training between the HMM and the RBF network can improve the recognition rate. Regarding the DTW approach we are actually considering the case of several class templates, and different local path constraints [6].

So far, the methods proposed for gesture recognition were tested on a small sets of simple gestures and thus have very limited scope. We are currently extending the systems in order to overcome these limitations. The aim is to design real-time architectures that can work with a larger vocabulary of gestures, and remain user independent.

Acknowledgments

Andrea Corradini is supported by the European Commission through the TMR Marie Curie Grant # ERBFMBICT972613. We are also deeply grateful to Lara Neumann for her continuous help during this research.

5 References

[1] L. Baum & T. Petrie (1966) Statistical Inference for Probabilistic Functions of Finite State Markov Chains. *Ann. Math. Stat. 37*, pp. 1554-1563.

[2] C. Bishop (1995) Neural Networks for Pattern Recognition. *Clarendon Press, Oxford.*

[3] H. A. Bourlard & N. Morgan (1994) Connectionist Speech Recognition - A Hybrid Approach. *Kluwer Academic Publishers.*

[4] A. Corradini, U. -D. Braumann, H. -J. Boehme & H. -M. Gross (1998) Contour-based Person Localization by 3d Neural Fields and Steerable Filters. *Proceedings of MVA '98, IAPR Workshop on Machine Vision Applications*, pp. 93-96.

[5] A. Corradini, H. -J. Boehme & H. -M. Gross (1999) Visual-based Posture Recognition using Hybrid Neural Networks. *Proceedings of ESANN'99*, pp. 81-86.

[6] L. R. Rabiner & B. H. Juang (1993) Fundamentals of Speech Recognition. *Prentice-Hall Inc.*

[7] L. R. Rabiner & B. H. Juang (1986) An Introduction to Hidden Markov Models. *IEEE ASSP Magazine*, pp. 4–16.

[8] J. S. Bridle (1990) Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition. *Neurocomputing: Algorithms, Architectures and Applications, Springer Verlag.*

[9] T. Darrell, S. Basu, C. Wren & A. Pentland (1997) Perceptually-driven Avatars and Interfaces: Active Methods for Direct Control. *M.I.T Media Laboratory Perceptual Computing Section Technical Report No. 416.*

[10] R. E. Kahn (1996) PERSEUS: An Extensible Vision System for Human-Machine Interaction. *PhD-Thesis, University of Chicago, Department. of Computer Science.*

[11] A. H. Waibel (1989) Modular Construction of Time-Delay Neural Networks for Speech Recognition. *Neural Computation (1), pp. 39-46.*