Proc. IEEE Internat. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2000), Istanbul 2000, pp. IV 2361-2364, IEEE Press 2000

IMPLEMENTATION AND COMPARISON OF THREE ARCHITECTURES FOR GESTURE RECOGNITION

Andrea Corradini and Horst-Michael Gross

Technical University of Ilmenau, Department of Neuroinformatics
Ilmenau, Federal Republic of Germany
e-mail:*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 three architectures for the recognition of dynamic gestures using the above mentioned techniques or a hybrid combination of them are presented and compared. For all architectures a common preprocessor receives as input a sequence of color images, and produces as output a sequence of feature vectors of continuous parameters.

The first two systems are hybrid architectures consisting of a combination of neural networks and hidden Markov models (HMM). NNs are used for the classification of single feature vectors while HMMs for the modeling of sequences of them with the aim to exploit the properties of both these tools. More precisely, in the first system a Kohonen feature map (SOM) clusters the input space. Further, each codebook is transformed into a symbol from a discrete alphabet and fed into a discrete HMM for classification. In the second approach a Radial Basis Function (RBF) network is directly used to compute the HMM state observation probabilities. In the last system only dynamic programming techniques are employed. An input sequence of feature vectors is matched by some predefined templates by using the dynamic time warping (DTW) algorithm.

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

Recently, there have been strong efforts to develop intelligent, natural interfaces between users and systems based on gesture recognition. The optimal interaction has to be natural, intuitive, not require any remembrance and is similar to that we are familiar, thus the interaction with other people. 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 [6][8].

A. Corradini is supported by the European Commission through the TMR Marie Curie Grant # ERBFMBICT972613

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 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, and so on.

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 containing the actual user's posture, and it 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 two systems to determine that estimate, we formally consider the recognition problem as a statistical classification task. Given as many gesture classes $C_j \mid j = 1..C$ as the movements to detect, and a parametric representation of the movement as sequence of feature vectors $X_N = \{x_1, .., x_N\}$ we face the classification problem from a statistical point of view using the maximum a posteriori (MAP) criterion. By means of the Bayes' rule this latter can be formulated as finding the class for which the posterior probability $P(C_j|X_N)$ is maximized. To calculate this probability we make use of a class of probabilistic models describing the user's movements and variations: the hidden Markov models.

In the third system we face the recognition task as a template matching problem. Therefore we make use of dynamic programming techniques in order to find the minimal distance between an input sequence and the (previous defined representants of the) classes.

This paper is structured as follows. In the next section we give the definitions of gesture which we base on, and we briefly review previous studies in this field. Starting from our saliency system for person localization [4], in Sec. 3 we provide an overview of the process which is to be carried out to describe the user's postures. Sec. 4 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.

2. RELATED WORK AND DEFINITIONS

Different attempts at gesture recognition have appeared in the literature over the past years. Most of them deal with the two-dimensional set of movement patterns. An early effort by Rubine [13] used mathematical functions as recognizer for the 2D trajectory patterns of mouse gestures. Due to the extraction of some low-level features from the raw data, the choice of the proper feature set is very important. Howell and Buxton [7] presented experiments using a radial basis function variant of the time-delay neural network able to learn simple gestures. The movement considered were simple pointing and waving gestures. Yamato [15] used discrete HMMs to recognize image sequence of six tennis strokes with respectable accuracy. Another papers exploiting HMMs for continuous gesture recognition based on time-sequential camera images was proposed in [14]. In [10] some three-dimensional hand movements were modeled as a sequence of movement primes as the unit of recognition. A recent work of Davids and Bobick [16] exploits the view-based representation of motion over time from different points of view. They basically construct a vector template image which has to be matched against stored representation of the actions to detect. That system is able to recognize 18 aerobic exercises in real-time.

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.*

3. VIEW-BASED POSTURE DESCRIPTION

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. This function represents a parametric model for the skin color and is unequivocally described by a mean vector μ and a covariance matrix Σ (the parameters). Using the Mahalanobis distance between the mean vector and an image pixel, this latter is classified to be or not a member of the skin class according to a threshold value.

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). Then to avoid problems deriving by the shape of each region due to the choice of the color threshold, we model each of these regions as a circle around their COGs (Fig. 1,b). 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. As first ten feature vector elements we calculate the so-called *scale normalized moments* [9] up to the third order. They remain unchanged under translation and size change.

In addition, to compensate the shift variation of the person gesticulating in front of the camera we choose for each image a suitable coordinate system by fixing its origin point at the current determined head's center of mass (Fig.1,c). It allows to calculate the remaining four feature vector elements relating to the head position and regardless to the user's position within the image. In this new coordinate system in order to ensure invariance also with respect to image size change, we use the polar coordinates of both hands's COG. 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 (Fig. 1,d). For more details on the pose segmentation and the feature extraction task see [5].

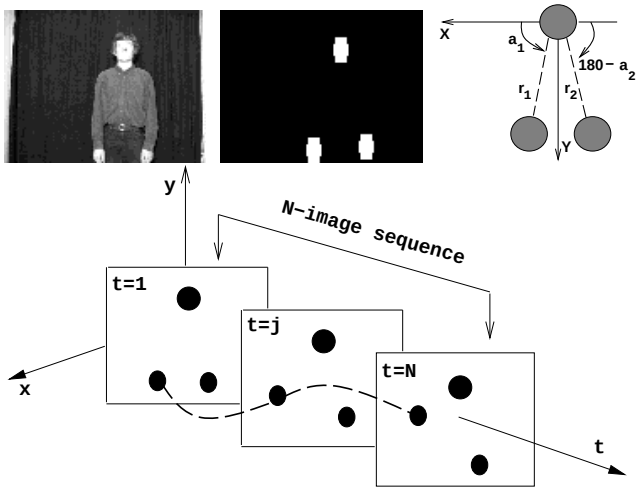

Figure 1: *From above to below and left to right: input image, hand and head regions modeling, new defined coordinate system, and finally gesture as hand and head trajectories within the feature space extended along the time.*

Finally, 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.

4. GESTURE RECOGNITION SYSTEMS

4.1. 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 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 [12] 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.

4.2. 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 distributions 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

Figure 2: *From left to right: the waving-right gesture is divided into 9 labeled subregions each covering exactly 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.*

according to the label of the corresponding subgesture (see Sec. 4.1). Absorbing as well the bias parameters as the normalizing factors of the Gaussian functions into the weights, and remembering the output represents a probability value, the output of the l-th node is given by

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

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 the parameters mean vector μ and covariance matrix Σ 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.

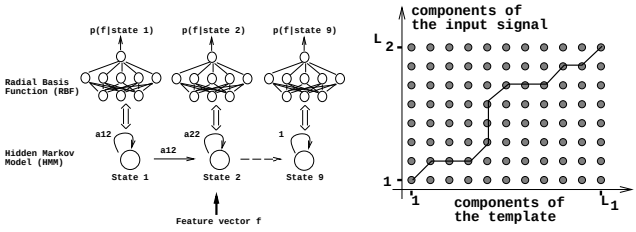

Figure 3: *From left to right: HMM/RBF hybrid architecture overview, and example of DTW time alignment and normalization of two pattern sequences of length L_1 and L_2, respectively.*

4.3. 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 [11]. 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.

5. 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.

Gest.	Recognition rate in %			% false class. gestures		
	SOM	RBF	DTW	SOM	RBF	DTW
stop	83.6	84.2	76.2	6.4	5.2	6.6
hi r.	85.4	88.9	76.2	7.1	4.3	7.4
hi l.	85.0	88.5	75.0	7.4	4.7	7.0
go r.	85.5	87.2	74.6	6.3	5.0	6.0
go l.	84.2	91.2	77.3	7.4	4.4	6.1

Table 1: Recognition results with the different architectures. The abbreviations r. and l. mean right and left, respectively.

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 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 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 systems, 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][12]. 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 [11].

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.

6. 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. *Proc. of MVA '98, IAPR Workshop on Machine Vision Appl.*, pp. 93-96.

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

[6] T. Darrell, S. Basu, C. Wren & A. Pentland (1999) Perceptually-driven Avatars and Interfaces: Active Methods for Direct Control. *M.I.T Tech. Rep. 416.*

[7] A. J. Howell & H. Buxton (1998) Learning Gestures with Time-Delay RBF Networks. *Proc. of the 8th ICANN, vol. 1*, pp. 239-244.

[8] R. E. Kahn (1996) PERSEUS: An Extensible Vision System for Human-Machine Interaction. *PhD-Thesis, Univ. of Chicago, Dpt. of Comp. Science.*

[9] K. Li (1992) Reforming the Theory of Invariant Moments for Patt. Recognition. *Patt. Rec.* **25**(7):723-730.

[10] Y. Nam & K. Wohn (1996) Recognition of Space-Time Hand-Gestures using HMMs. *Proc. of Workshop on Integration of Gest. in Lang. and Speech*, pp. 175-184.

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

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

[13] D. Rubine (1991) Specifying Gesture by Example. *Computer Graphics* **25**(4):329-337.

[14] T. Starner & A. Pentland (1995) Visual Recognition of American Sign Language using HMMs. *Proc. of the Int. Workshop on Autom. Face and Gesture Rec.*, pp. 189-194.

[15] J. Yamato, J. Ohya & K. Ishii (1992) Recognizing Human Action in Time-Sequential Images using HMMs. *Proc. Comp. Vis. and Pattern Rec.*, pp. 379-385.

[16] J. W. Davis, & A. F. Bobick (1997) The Representation and Recognition of Action Using Temporal Templates. *M.I.T. Tech. Rep. 402.*.