ADAPTIVE GESTURE RECOGNITION IN HUMAN COMPUTER INTERACTION

George Caridakis, Kostas Karpouzis, Nasos Drosopoulos and Stefanos Kollias

Image, Video and Multimedia Systems Laboratory
National Technical University of Athens
gcari, kkarpou, ndroso, skollias@image.ntua.gr

ABSTRACT

An adaptive, invariant to user performance fluctuation or noisy input signal, gesture recognition scheme is presented based on Self Organizing Maps, Markov Models and Levenshtein sequence distance. Multiple modalities, all based on the hand position during gesturing, train different classifiers which are then fused in a weak classifier boosting-like setup by weight assignment to each stream. The adaptability of the proposed approach consists of the incorporation of Self Organizing Maps during training, the exploitation of neighboring relations between states of the Markov models and the modified Levenshtein distance algorithm. The main focus of current work is to tackle intra and inter user variability during gesture performance by adding flexibility to the decoding procedure and allowing the algorithm to perform an optimal trajectory search while the processing speed of both the feature extraction and the recognition process indicate that the proposed architecture is appropriate for real time and large scale lexicon applications.

1. INTRODUCTION

Gesture recognition and gesture based Human Computer Interaction have been increasingly attracting attention from researchers across disciplinaries such as machine learning, pattern recognition, computer vision, human computer interaction (HCI) and linguistic and natural language processing. This multidisciplinary research area can find applications in several disciplinaries such as multimodal HCI, robotics control, psychological behavior studies and emotion analysis, sign language recognition, assistive e-learning technologies and virtual environments navigation. Human Computer Interaction is constantly defining new modalities of communication, and new ways of interacting with machines [1]. Gestures can convey information for which other modalities are not efficient or suitable. In natural and user-friendly interaction, gestures can be used, as a single modality, or combined in multimodal interaction schemes which involve speech, or textual media [2]. Emotion recognition is another domain where gesture analysis is crucial and could provide important cues in a multimodal emotion recognition framework in natural HCI [3].

There is an abundance of approaches for gesture recognition and methodologies well presented in [4], [5] and [6]. Mitra and Acharya focus on gesture recognition, while Ong and Ranganath extend their research on automatic sign language recognition. Both surveys deal with feature extraction techniques and classification issues related to automatic analysis of gestures. Wu and Huang focalize more on hand modeling (shape analysis, kinematics chain and dynamics), computer vision and pattern recognition issues associated to hand localization and feature extraction from image sequences. Concerning the input stream of each approach there are two dominant categories: motion capture (direct-measure device) data gloves and video input stream but time of flight cameras or accelerometers have been also used. While datagloves are quite expensive and really intrusive they provide a more robust, accurate, detailed and efficient way of capturing 3D hand location and finger flexion in real time when compared to vision based approaches. Visual input on the other hand could be used for outdoor scenarios where the user is not equipped with specialized devices. Vision based approaches have the advantage of being non intrusive but frequently several assumptions have to be made or constraints to be applied during the recording process concerning either the environment, the user or the camera(s) setting. Usually when some kind of skin color model is employed the user is asked to wear long sleeves and proper clothing so as not to have too many skin areas exposed that are not head or hands (e.g. decolletage, skin like colored clothes).

The rest of the paper is organized as follows: Section 2 introduces the proposed approach and is further refined in the training sections 2.1, 2.2, 2.3 and the evaluation section 2.4 dealing with the feature extraction process, training and testing stage of the classifier respectively. Section 2.5 presents the experimental results of the overall system, while section 3 concludes the article and presents ongoing and future work in addition to possible extensions and applications of the architecture.

2. PROPOSED ARCHITECTURE

Present work introduces the SOMM architecture for gesture recognition by fusing separate component models all based on

hand trajectory. A novel approach is presented by applying a combination of self organizing maps and markov models for gesture trajectory classification. The extracted features used in the trajectory module include the trajectory of the hand and the direction of motion in the various stages of the gesture. This classification scheme is based on the transformation of a gesture representation from a series of coordinates and movements to a symbolic form and on building probabilistic models using these transformed representations. Our study indicates that, although each of the classifiers (hand position, motion direction) can provide distinctive information in most cases, only an appropriate combination can result in robust and confident user independent gesture recognition.

The introduced procedure initiates with the image processing module described in [7]. Following, each gesture instance is represented by a time series of points, representing the hand's location with respect to the head of the user, using the mapping function of the SOM and a crisp quantization process for the hand direction. The discrete symbols (SOM nodes and direction angles) are then used to construct the transition probability matrix of two Markov models types. The proposed modeling scheme is based on the transformation of a gesture representation from a series of coordinates and movements to a symbolic form which, in turn, is used to build the respective probabilistic models. The first transformation is based on the relative position of the hand during the gesture and is achieved using a self-organizing map model. Despite the fact that the map units are treated as symbols, the map's neighborhood function provides a distance metric between them, that is used during the classification of an unlabeled gesture. Additionally, this enables the use of the Levenshtein distance metric for the comparison between these sequences of symbols and the definition of a 'mean' string of symbols representing e.g. the gestures included in a D_j set.

2.1. Position model

Let us suppose our gesture vocabulary consists of c gesture classes in the D gesture dataset. So our dataset D with every gesture class set D_j containing n_j gesture instances $D_j = \{G_{1_j}, G_{2_j}, \ldots, G_{n_j}\}$, n_j denoting the number of repetitions for gesture class j. Every gesture instance G_{j_i} contains l_{j_i} coordinates so that Hand coordinates relative to the head position in the specified frame. Relative coordinates are used since the user position during recording is not known and furthermore normalization is applied for every user since the user's height and arm length could pose a problem during gesture modeling as will become apparent in the next section 2.1.

$$\begin{aligned} T(G) &= (u_1, u_2, \ldots, u_l) \\ &: u_i = BMU(x_i, y_i), i \in [1, l] \end{aligned} \tag{1}$$

Function $BMU(x_i, y_i)$ returns the index of the best-matching unit for point (x_i, y_i) and $T(G)$ is the modified gesture representation. Given that u_i is the index of a map unit, this function is declared as $BMU : R^2 \rightarrow S$, where S is the set of the indices of all map units and can be treated as a set of symbols. In many cases, the u_i value of consequent points of a gesture remains the same since, although the continuous movement of the hand is represented by distinct points, consequent points are generally close in the input data space. Replacing consequent equal values of u_i with a single value results in the following gesture definition:

$$\begin{aligned} G^{'} &= N(T(G)) = \{u_1^{'}, u_2^{'}, \ldots, u_m^{'}\} \\ &: m \leq l, u_t^{'} \neq u_{t-1}^{'} \forall t \in [2, m] \end{aligned} \tag{2}$$

where N is a function that removes consecutive equal u_i values and $G^{'}$ is the transformed gesture instance. The transformation of the gestures with the use of the SOM can be considered as a transformation of the continuous trail to a sequence of m discrete symbols, different for every gesture class, that define the finite states to build first order Markov chain models. By removing consecutive equal values for symbols u, the self transition probability values in the Markov transition probability matrix would be zero. By applying the same transformation $N(T)$ to the gesture instance to be decoded, as will be explained in detail in section 2.4, self transition probability values will also be removed from the unknown gesture instance to be classified. This procedure leads into a loss of information regarding duration of a particular state but this information is not crucial for gesture recognition and additionally enhances the architecture with an abstraction layer.

A Markov model, for each of the c categories in the gestures' data set, is created. The sequence of the u_i values into the transformed gestures $G^{'}$ of $D_j^{'}$ set, will be used for the calculation of the transition probabilities of the model MM_j^{som} describing category j and for the evaluation of the first state probability function $\pi_j^{som}$of this model. The result is a set MM^{som} of c Markov models.

$$\begin{aligned} MM^{som} &= \{MM_1^{som}, MM_2^{som}, \ldots, MM_c^{som}\} \\ &: D_j^{'} = \{G_1^{'}, G_2^{'}, \ldots, G_{n_j}^{'}\} \rightarrow MM_j^{som} \end{aligned} \tag{3}$$

These models are used to evaluate a new unlabeled gesture in order to be classified in one of the c categories.

2.2. Direction model

With the purpose of providing a more descriptive representation of each gesture instance, an additional transformation is introduced, based on the optical flow sequence of each gesture. This describes the different directions that the gesture trajectory presents instead of the spatial position of hands relative to the head. In order to achieve such a representation, direction vectors are calculated from the consecutive gesture trajectory points according to equation 4 and quantized in 8

different symbolic values. In that sense, we define the transformation of a gesture instance G using the OF function as:

$$\begin{aligned} OF(G) &= \{v_1, v_2, ..., v_m\} \\ : v_i &= W_r(Q(\arctan(\frac{y_i - y_{i-1}}{x_i - x_{i-1}}))) \end{aligned} \tag{4}$$

where v_i are the quantized values, Q the quantization function and W_r a median filter applied to the values for a fixed length window of r around the input value. The purpose of the later is to smooth the quantized values against possible instabilities of the hand during the gesture. Applying the transformation function in conjunction with function N for the removal of the equal consecutive values we get:

$$G_i^{''} = N(OF(G)) = \{v_1, v_2, ..., v_m\} \tag{5}$$

where v_i values define the states for a new set of Markov models MM^{of} that is built using the transformed set $D_j^{''}$. The first state probability function π_j^{of} is also calculated using this set as follows:

$$\begin{aligned} MM^{of} &= \{MM_1^{of}, MM_2^{of}, ..., MM_c^{of}\} \\ : D_j^{''} &= \{G_1^{''}, G_2^{''}, ..., G_n^{''}\} \rightarrow MM_i^{of} \end{aligned} \tag{6}$$

2.3. Levenshtein

An additional model that is created per gesture class is the Generalized Median of the $D_j^{'}$ set. In general, a generalized median of a set of sequences S is defined as the sequence, that consists of a combination of all or some of the symbols used in the set that minimizes the sum of distances to every string of S [8]. In case the generalized median sequence belongs to the set S it is called Generalized Set Median.

$$\begin{aligned} M_j &= generalized\_median(D_j^{'}) \\ &= \arg\min_g \sum_{G^{'} \in D_j^{'}} L(g, G^{'}) \end{aligned} \tag{7}$$

Let M_j be the generalized median of the $D_j^{'}$ set, using a modified version of the Levenshtein distance L, a widely employed distance metric. This variation of the Levenshtein distance incorporates the neighboring relation between SOM nodes which are the symbols of the two sequences in question for assigning a cost for each symbol substitution and is also employed during the decoding stage. The original cost assignment algorithm is $IFstr1[i] = str2[j]THENcost := 0ELSEcost := 1$ which takes place during symbol replacement $(str1[i], str2[j])$ comparison of sequences $str1$ and $str2$ to decide which action achieves minimal cost. Such an algorithm though would not take into consideration how similar symbols $str1[i], str2[j]$ are, if any similarity measurement actually exists in the symbol set. In the case of the SOM, trained to map hand coordinates, such a relation subsists between the nodes which are actually the symbols of the set constituting each sequence. This way the cost for the substitution action should be smaller if the two symbols participating in the substitution are close in terms of some similarity measurement (SOM neighboring function in our case) and on the other hand should be greater if the two nodes are distant according to some similarity measurement. So the cost should be modified accordingly $cost := 1 - NF_{str1[i]}(str2[j])$. The mean Levenshtein distance between the members of each $D_j^{'}$ set and M_j is also calculated and denoted as ML_j. This is an informal way to measure the variation within the members of the set and will be used accordingly in the decoding stage (section 2.4).

$$ML_j = \frac{\sum_{i=1}^{n_j} L(G_i^{'}, M_j)}{n_j} \tag{8}$$

2.4. Gesture Decoding

The classification of an input gesture is based on the two sets of Markov models (equations 3 and 6). Let G_k be a gesture instance of unknown category, and $G_k^{'}$ and $G_k^{''}$ its transformed representations, according to equations 2 and 5. Using the MM^{som} set of models, the probability of this gesture belonging to category j can be calculated as:

$$\begin{aligned} P(G_k^{'}|MM_j^{som}) &= \frac{\prod_{i=1}^{q} S_i^{som}}{q} \\ : q &= \left|G_k^{'}\right| \end{aligned} \tag{9}$$

The above equation averages the values S_i^{som}, which represent an evaluation factor for each $u_i : i \in [1, q]$ value of the $G_k^{'}$ transformed gesture with respect to the MM_j^{som} Markov model. These values are calculated as:

$$\begin{aligned} S_1^{som} &= \max_z(NF_{u_1}^{som}(z)\pi_j^{som}) \\ S_i^{som} &= \max_z(NF_{u_i}^{som}(z)P(z|u_{i-i}, MM_j^{som})) \end{aligned} \tag{10}$$

For the first state, the system simply performs a search for the node that has the largest joint probability of:

- being close to u_1 which is $NF_{u_1}^{som}(z)$
- being the first state in MM_j^{som} which is π_j^{som}

For nodes that $\in [2, q]$, a similar search is performed but the second probability is not that of being the first state but instead is a transition probability $P(z|u_{i-i}, MM_j^{som})$. $NF_{u_i}^{som}(z)$ is the distance of unit z with node u_i as defined by the SOM Gaussian neighborhood function with the second unit as its center. As z varies across all the units of the map, this product will provide a unit that combines a considerable transition

probability from the previous state with a relative small distance onto the map grid from the current state. This unit will also be used as the previous state in the next step:

$$u_i = \arg\max_z(S_i^{som}) : i \in [1, q] \tag{11}$$

An almost identical decoding process is performed for the case of optical flow. The slight difference is that although for position NF^{som} was provided by the SOM, NF^{of} is arbitrarily defined and more detailed a value of $1/2$ is given for the closest direction neighbor and $1/4$ for the second closest neighbor in both directions. All other values are 0. As a result the respective equations are:

$$P(G_k^{''}|MM_j^{of}) = \frac{\prod_{i=1}^{q} S_i^{of}}{q} \\ : q = \left|G_k^{''}\right| \tag{12}$$

$$S_1^{of} = \max_z(NF_{u_1}^{of}(z)\pi_j^{of}) \\ S_i^{of} = \max_z(NF_{u_i}^{of}(z)P(z|u_{i-i}, MM_j^{of})) \tag{13}$$

$$u_i = \arg\max_z(S_i^{of}) : i \in [1, q] \tag{14}$$

Shorter gesture instances tend to gain an advantage by having less transitions and thus less probabilities multiplication. To tackle this problem we have introduced an additional similarity measurement based on M_j, the generalized median of each class, according to the Levenshtein distance. This can also tackle the partial gesture problem, where if the whole of a gesture instance is the starting part of a gesture class then it would get high ranking using just MM^{som} and MM^{of}.

$$P(G_k^{'}|M_j) = \frac{ML_j}{L(G_k^{'}, M_j)} \tag{15}$$

Please note that $P(G_k^{'}|M_j)$ is a similarity measurement and not a probability, since its value can be > 1.

Finally, the winner class is decided as:

$$\arg\max_j(P(G_k^{'}|MM_j^{som})P(G_k^{''}|MM_j^{of})P(G_k^{'}|M_j)) \tag{16}$$

Quality criteria can be further applied in the form of a threshold either to the overall evaluation of the gesture instance or to parts of equation 16 not allowing thus poor scoring gestures to be classified. Additionally in ambiguity situations the n first classes, ordered by score, can all have high evaluation scores. This can be resolved by monitoring score difference between the two best scoring classes. If the score is close, ambiguity is detected.

2.5. Experimental results

Validation of the proposed architecture was performed on a dataset formed by the 30 gestures consisting of 10 repetitions each with the classes varying in complexity from very simple directive gestures to very complex ones. Experiments were conducted, using the described dataset, in order to evaluate the recognition performance of the proposed method. Using all the gesture instances, for both the training and the testing phases of the system, in an attempt to validate the system's learning capabilities, resulted in 100% recognition percentages. For an evaluation of the generalization capabilities of the proposed method, another experiment was executed using the 10-fold cross validation strategy. In this case the average recognition rate was 93%. The experiments were performed using Matlab on a regular PC (2GHz Dual Core, 3GB RAM) and for training all thirty classes 0,23 seconds were required (0,073 for MM^{of} and 0,15 for MM^{som}. The decoding stage varies depending on the gesture length but the average was 0.843 msec per gesture instance per gesture class, a performance which establishes the overall architecture suitable for real time applications.

In order to compare the results of our system with the most commonly used approach in the literature we implemented a HMM based classifier. We trained one HMM per gesture class. We used continuous left-to-right models and a mixture of 3 Gaussian probability density functions. During the decoding phase a gesture instance was tested against all models and the one with the highest log-likelihood value was selected as the winner resulting an average recognition rate of 86,36%.

These experimental study indicates that the currently proposed architecture produces encouraging results and when compared to one of the most popular approach demonstrates superiority mainly due to the adaptability characteristics able to cope with gesture variability and input signal noise.

3. CONCLUSIONS AND FUTURE WORK DISCUSSION

In this paper we propose an original automatic gesture recognition architecture via a novel classification scheme incorporating Self-organizing maps and Markov chains. Extracted features train separate classifiers, which in turn are fused during the classification stage, enhancing the proposed architecture with robustness against noisy and unconstrained environments or gesture variation. Intra and inter user variability during gesture performance are tackled through the flexibility of the decoding procedure provided by the neighboring characteristic of the SOM nodes and the optimal trajectory search performed during classification. Additionally the computational cost and processing speed of both the feature extraction and the recognition process indicate that the proposed architecture is suitable for real life applications and all the require-

ments accompanying such scenarios.

An obvious extension to the proposed approach would be the incorporation of hand shape features in the overall decision mechanism. Although SOM mapping seems unsuitable for hand shapes because in the 2D representation of the map the neighboring function might not be so representative of the actual similarity between the actual unmapped hand shape features. The inclusion of handshape information would make the approach suitable for Sign Language Recognition, an apparent extension of gesture recognition. Hand shape information and possible knowledge based fusion would integrate the Sign Language recognition extension. Adding this final layer of knowledge assisted recognition of linguistic or grammatical phenomena provides the assertional component of a knowledge base.

Gaming environments is another area where gesture recognition could be applied. More specifically the Wii game platform has recently become quite popular with its user motion controlled interaction within the virtual gaming environment. The three accelerometers installed in the Wii Remote provide measurements of the acceleration in the 3 dimensions. A movement can be detected either automatically, when values monitored,are above a threshold, or by pressing one of the available buttons. Velocity and position are obtained via single and double integration. The proposed recognition scheme can be applied to the provided features and used for training and recognizing Wii gestures in the game environment, since it has been proven to be superior to the popular HMM recognition architecture proposed by [9] in gaming environments.

4. REFERENCES

[1] Antonio Camurri and Gualtiero Volpe, Eds., *Gesture-Based Communication in Human-Computer Interaction, 5th International Gesture Workshop, GW 2003, Genova, Italy, April 15-17, 2003, Selected Revised Papers*, vol. 2915 of *Lecture Notes in Computer Science*, Springer, 2004.

[2] Annelies Braffort, Rachid Gherbi, Sylvie Gibet, James Richardson, and Daniel Teil, Eds., *Gesture-Based Communication in Human-Computer Interaction*, Springer Berlin / Heidelberg, 1999.

[3] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, and J.G. Taylor, "Emotion recognition in human-computer interaction," *Signal Processing Magazine, IEEE*, vol. 18, no. 1, pp. 32–80, Jan 2001.

[4] S. Mitra and T. Acharya, "Gesture recognition: A survey," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 37, no. 3, pp. 311–324, 2007.

[5] S.C.W. Ong and S. Ranganath, "Automatic sign language analysis: a survey and the future beyond lexical meaning," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 873–891, 2005.

[6] Y. Wu and T. Huang, "Hand modeling, analysis, and recognition for vision-based human computer interaction," *IEEE Signal Processing Magazine*, vol. 18, pp. 51–60, 2001.

[7] G. Caridakis, A. Raouzaiou, E. Bevacqua, M. Mancini, K. Karpouzis, L. Malatesta, and C. Pelachaud, "Virtual agent multimodal mimicry of humans," *Language Resources and Evaluation 41, Special issue on Multimodal Corpora, Springer*, vol. 41, pp. 367–388, 2007.

[8] Y. Lee and S. Kassam, "Generalized median filtering and related nonlinear filtering techniques," *Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on*, vol. 33, no. 3, pp. 672–683, 1985.

[9] Thomas Schlömer, Benjamin Poppinga, Niels Henze, and Susanne Boll, "Gesture recognition with a wii controller," in *TEI '08: Proceedings of the 2nd international conference on Tangible and embedded interaction*, New York, NY, USA, 2008, pp. 11–14, ACM.