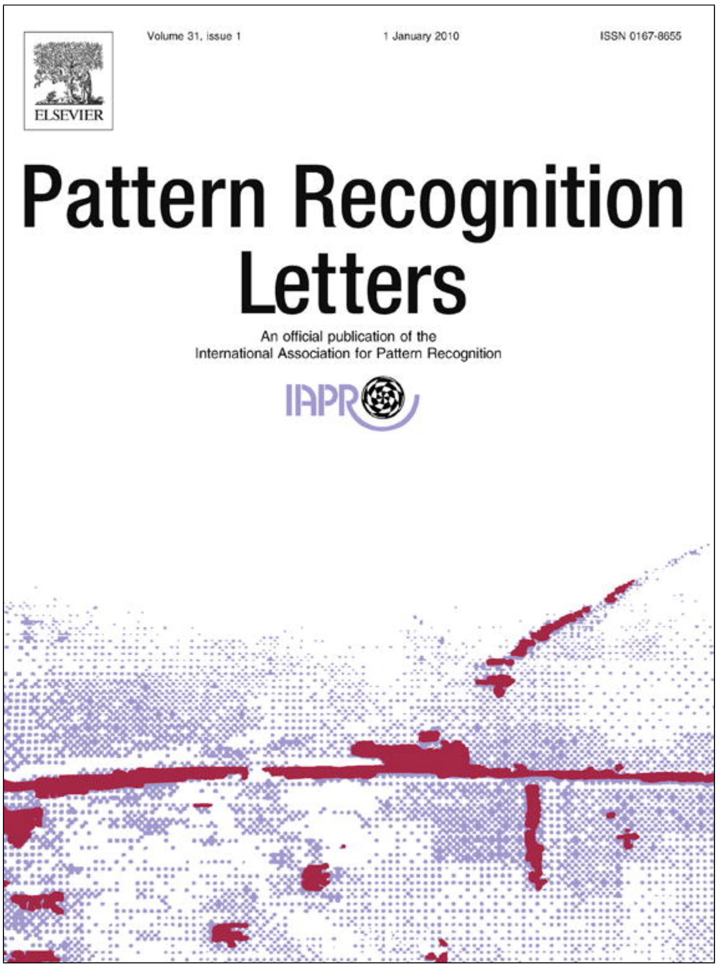

Volume 31, issue 1 1 January 2010 ISSN 0167-8655

ELSEVIER

Pattern Recognition Letters

An official publication of the
International Association for Pattern Recognition

IAPR

Pattern Recognition Letters 31 (2010) 52–59

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

SOMM: Self organizing Markov map for gesture recognition

George Caridakis *, Kostas Karpouzis, Athanasios Drosopoulos, Stefanos Kollias

Image, Video and Multimedia Systems Laboratory, National Technical University of Athens, Athens, Greece

ARTICLE INFO

Article history:
Received 13 December 2008
Received in revised form 16 July 2009
Available online 8 September 2009

Communicated by G. Borgefors

PACS:
42.30.Sy

Keywords:
Gesture recognition
Spatiotemporal pattern recognition
Self organizing maps
Markov models
Gesture-based interaction

ABSTRACT

Present work introduces a probabilistic recognition scheme for hand gestures. Self organizing feature maps are used to model spatiotemporal information extracted through image processing. Two models are built for each gesture category and, along with appropriate distance metrics, produce a validated classification mechanism that performs consistently during experiments on acted gestures video sequences. 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 finds diverse applications in 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 (Camurri et al., 2004), since gestures can convey information for which other modalities are not efficient or suitable. In natural interaction, gestures can be used as a single modality, or combined in multimodal interaction schemes which involve speech, or textual media (Braffort et al., 1999). Emotion (Cowie et al., 2001) and social (Vinciarelli et al., 2008) signal recognition is another domain where gesture analysis is crucial and could provide important cues in a multimodal recognition framework in natural environments.

A gesture is a motion of the body that conveys information; in this paper, we focus on hand gestures and information conveyed from these gestures. A gesture taxonomy can be formalized in a scaling continuum: gesticulation, speech-linked, pantomime, emblems and sign languages as proposed by McNeill (1992). An alternative gesture taxonomy can be defined according to their functionality:

- Symbolic gestures: gestures that, within each culture, have come to have a single meaning.
- Deictic gestures: types of gestures most generally seen in HCI and are the gestures of pointing to entities or direction.
- Iconic gestures: gestures used to convey information about the size, spatial relations, actions, shape or orientation of the object of discourse display.
- Pantomimic gestures: gestures typically used to mimic an action, object or concept.

Preliminary versions of this work can be found at Caridakis et al. (2008, 2007). The rest of the paper is organized as follows: Section 2 discuss related work and the challenges involved in gesture recognition in general. Section 3 introduces the proposed approach and is further refined in Sections 3.1, 3.2 and 3.3 dealing with the feature extraction process, training and testing stage of the classifier, respectively. Section 3.5 presents the experimental results of the overall system, while Section 4 concludes the article and presents ongoing and future work in addition to possible extensions and applications of the architecture.

* Corresponding author. Fax: +30 2107722492.
E-mail addresses: gcari@image.ntua.gr (G. Caridakis), kkarpou@image.ntua.gr (K. Karpouzis), ndroso@image.ntua.gr (A. Drosopoulos), skollias@image.ntua.gr (S. Kollias).

0167-8655/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2009.09.009

2. Related work

There is an abundance of approaches for gesture recognition and methodologies well presented in (Mitra and Acharya, 2007; Ong and Ranganath, 2005; Wu et al., 2001). 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.

2.1. Hidden Markov models

Coogan et al. (2006) present a quite common architecture involving color based hand detection, scale and rotation invariant PCA classification for hand shape and discrete hidden Markov models (HMMs) for position information. While they report an adequate recognition rate of 94.5% for static hand shape recognition, for dynamic two subject test the recognition rate varies from 83% to 98.6% depending on the training/testing ratio. Although the fusion of hand shape and position is an interesting approach discrete HMMs do not seem to be able to cope with intra/interuser gesture variation. Continuous HMM are also used by Huang et al. (2000); in this work, hand shape detection and tracking is based on the Active Shape Model method. In order to discriminate attention and non-attention seeking gestures, Hossain et al. (2005) present a HMM variation, called the Implicit/Explicit Temporal Information Encoded, which parameterizes the emission probability in the hidden states. Mantyla et al. (2000) focus on the applicability of their architecture on mobile devices using accelerometer-provided features, utilizing self-organizing maps (SOM) for static and HMM for dynamic gestures. Miners et al. (2002) decompose gestures into primitives in an attempt to improve scalability, while reducing complexity; these gesture primitives are synthesized into concepts and associated with a knowledge base using an conceptual approximate graph matching technique. Starner et al. (1998), in one of the most cited publications in the research area of sign/gesture language recognition, propose an HMM-based, sentence-level continuous recognition scheme in the framework of American Sign Language (ASL) and experiment with desk mounted and cap mounted matchstick-sized cameras. Wilson and Bobick (1999) tackle the problem of systematic variation in sensor output or contextual information, cases where conventional HMMs suffer, by proposing a parametric HMM variation where training assumes that each input feature vector is labeled with the value of the parameterization.

2.2. Other

Additionally to the dominant HMM-based approach, several researchers have experimented with artificial intelligence and other pattern recognition techniques. The CONDitional dENSity propagATION (CONDENSATION) algorithm was initially used by Black and Jepson (1998) for classification; Patwardhan and Roy (2007) extended the standard algorithm with the 'predictive' extension for hand detection and tracking, in conjunction with Mahalanobis distance, while (Wong and Cipolla, 2006) tackled the issue of gesture spotting in continuous gesturing with a sparse Bayesian classifier. Hong et al. (2000) employed Finite State Machines (FSM), while (Raytchev et al., 2000) used a structure consisting of dynamic buffers, which followed the projection of primitive features to discriminant feature space. Su (2000) employed fuzzy logic and rule-based approaches, extracting crisp rules from the values of synaptic weights of a trained hyper-rectangular composite neural network (HRCNNs), which are then fuzzified in the process. Similarly, Juang and Ku (2005) use a fuzzified Takagi–Sugeno–Kang (TSK) type recurrent network, in which rules are constructed online by concurrent structure and parameter learning. Neural networks which model dynamics have been also used in (Yang et al., 2002) (Time Delay Neural Network) and in (Huang and Huang, 1998) with a 3D Hopfield Neural Network.

3. System overview

This paper introduces a SOMM-based architecture for gesture recognition, fusing separate component models all of which are based on hand trajectory. The approach involves a combination of self organizing maps and Markov models for gesture trajectory classification, using the trajectory of the hand segment and direction of motion during a 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 based on 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 (shown in Fig. 1) begins with the image processing module responsible for the detection and tracking of head and hands as described in Section 3.1. Following, each detected gesture instance is represented by a time series of points, corresponding to the location of the hands 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 (direction angles and SOM nodes for the left and right branch of the flowchart, respectively) are then used to construct the transition probability matrix of a Markov model for each hand. Using the classification results along with a similarity measurement to tackle the scale variety of available gestures, the system forms a decision for the detected motion.

3.1. Feature extraction

Wu et al. (2001) and Ong and Ranganath (2005) provide excellent reviews of head and hand detection and tracking approaches. From these, only video based methods were considered here, since motion capture or other intrusive techniques (for instance, using data gloves) were deemed inappropriate for the reasons mentioned in Section 1. Besides natural interaction, the major factors taken under consideration are computational cost and robustness, resulting in an accurate near real-time skin detection and tracking module. The overall image processing is described in detail in (Caridakis et al., 2007).

The described algorithm is lightweight, allowing a quasi-real time rate on a usual PC during our experiments, which is enough for continuous gesture tracking; an OpenCV implementation of the algorithm, accelerated by the NVIDIA CUDA technology, has proven to perform even faster averaging a rate of 10 ms per frame. The object correspondence heuristic makes it possible to individually track the hand segments correctly, at least during usual everyday gesture sequences. In addition, the fusion of skin color and motion information eliminates any background noise or artifacts, thus reinforcing the robustness of the proposed approach.

3.2. Gesture modeling

The proposed modeling scheme is based on the transformation of a gesture representation from a series of coordinates and

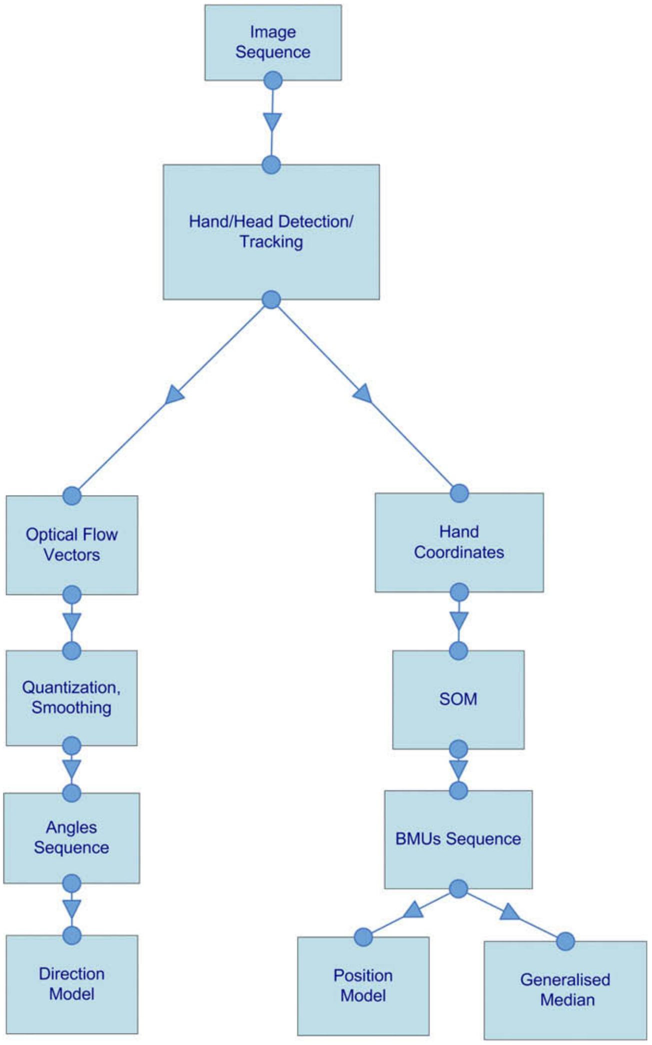

Fig. 1. System flowchart.

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, which 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 gesture instances included in a gesture class.

An additional transformation is based on the optical flow of the gesture, aiming to describe the changes in hand direction during gesturing. This transformation is based on quantized values of the subsequent angles of the hand's trajectory to create an additional set of Markov models.

During the classification stage, all the above mentioned trained models are evaluated against an unlabeled gesture instance. Recognition rates from each individual model are used as weights to fuse respective participation probabilities, resulting in a recognition scheme able to tackle cases of low confidence or ambiguity.

Let us suppose our gesture vocabulary consists of c gesture classes in the D gesture dataset. So our dataset is $D = \{D_1, D_2, \ldots, D_c\}$ 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 $G_{j_i} = \{(x_{1_{j_i}}, y_{1_{j_i}}), (x_{2_{j_i}}, y_{2_{j_i}}), \ldots, (x_{l_{j_i}}, y_{l_{j_i}})\}$, with l_{j_i} denoting the number of coordinates belonging to the hand trajectory for repetition i of class j. x, y are hand coordinates relative to the head position in the specified frame. Relative coordinates are used, since the distance from the camera and the position of the user within the frame during recording is not known beforehand. Also, differences in distance from the camera would result in differences of height and arm length across users, posing problems during gesture modeling as will become apparent in Section 3.2.1; thus, the sequences are normalized, relative to the head size (diagonal of the rectangle containing the detected face) of each user.

3.2.1. Position model

Although self organizing maps are used as a data visualization or dimension reduction technique we choose to utilize SOM as a clustering tool to derive a more abstract representation of the gesturing space and let the data decide on the neighboring relations between the map's nodes. A more simplistic approach would be to crisply quantify the gesturing space into blocks and assigning arbitrary neighboring relations. This approach seems to be unable to generalize well, while the self organizing attribute of SOMs, based on node competing for representation of the samples, is a more robust and adaptable approach. Here, the weight of a SOM node is allowed to change by learning, so as to better adapt to samples in hopes of achieving the minimum distance according to some distance metric; it is this selection-and-learning process that makes the weights organize themselves into a map representing similarities. Neighboring nodes are also affected during the learning process, thus node-neighboring relations are learned in addition to weight learning. This neighboring characteristic becomes crucial in the overall decoding process, elaborated in Section 3.3.

The coordinates (x, y) of all the points from all the gesture repetitions from all the gesture classes are used to train a hexagonal, two-dimensional grid SOM with the batch mode learning procedure. The structure of the grid of the SOM units is hexagonal in order to improve quality (isotropy) of the mapping (Kohonen, 1998) and avoid bias towards horizontal and vertical directions (Kohonen, 2001), while the size of the map is determined by a trial and error procedure. The coordinates are preprocessed in order to be normalized and position invariant: normalization for every user ensures that users that tend to use a larger or smaller gesture space or are anatomically larger or smaller do not introduce noise, bias or scaling issues in the training and classification processes. Additionally, hand positions are relative to the position of the head so that hand position in the frame is invariant of the actual position of the user in the image. These points are fed to the map in an unordered form, regardless of the gesture instance they belong to and to their ordered position into the gesture in order to avoid biasing the training procedure. The training of the SOM is performed once for the whole dataset D and not for every class, reducing the training time required, the storing resources required and adding to the simplicity of the system design. Furthermore, and given that an adequate number of gestures have been used to train the SOM, once a new class is introduced to the vocabulary one can assume that no additional training is required since the gesturing space has been well modeled and represented.

Following training, each point is assigned to the respective best matching unit (BMU) on the map, i.e. the unit of the map closer to the point in the input data space, according to the Euclidean distance of the two vectors. Thus, a gesture G_i can be transformed from a series of points to a series of map units according to the transformation function T. For simplicity we will replace the notation $G_{j_i} = \{(x_{1_{j_i}}, y_{1_{j_i}}), (x_{2_{j_i}}, y_{2_{j_i}}), \ldots, (x_{l_{j_i}}, y_{l_{j_i}})\}$ with $G = \{(x_1, y_1), (x_2, y_2), \ldots, (x_l, y_l)\}$. So, gesture instance G will be mapped to the SOM:

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

where 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 $BMU : R^2 \to S$, 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:

$$G' = N(T(G)) = \{u'_1, u'_2, \ldots, u'_m\} : m \leqslant l, \quad u'_t \neq u'_{t-1} \forall t \in [2, m] \tag{2}$$

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 and indicative for every gesture class, which define the finite states needed to build first order Markov chain models. Replacing consecutive equal values for symbols u with a single value, would result in zeroing the self transition probability values in the Markov transition probability matrix. By applying the same transformation $N(T)$ to the gesture instance to be decoded, as will be explained in detail in Section 3.3, 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 π_j^{som} of this model. The result is a set MM^{som} of c Markov models.

$$MM^{som} = \{MM_1^{som}, MM_2^{som}, \ldots, MM_c^{som}\}$$
$$: D'_j = \{G'_1, G'_2, \ldots, G'_{n_j}\} \to MM_j^{som} \tag{3}$$

These models are used to evaluate a new unlabeled gesture in order to be classified in one of the c categories. Fig. 2 depicts the above described transformation for a gesture instance in a more intuitive way.

3.2.2. 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 in the gesture trajectory, 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 Eq. (4). These angles are then quantized in eight different symbolic values as depicted in Fig. 3. The segments of coordinates in Figs. 2 and 3 are considered to be a set of coordinates that belong to the same cluster (BMU and Quantized Angle for Figs. 2 and 3, respectively). In that sense, we define the transformation of a gesture instance G using the OF function as:

$$OF(G) = \{v_1, v_2, \ldots, v_m\} : v_i = W_r\left(Q\left(\arctan\left(\frac{y_i - y_{i-1}}{x_i - x_{i-1}}\right)\right)\right) \tag{4}$$

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:

$$G''_i = N(OF(G)) = \{v_1, v_2, \ldots, v_m\} \tag{5}$$

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:

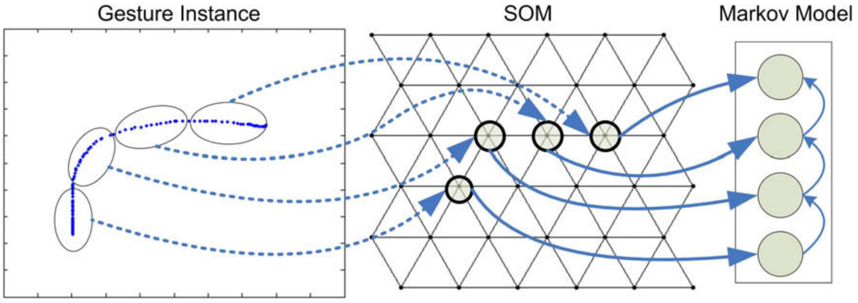

Fig. 2. Correspondence of gesture trajectory points to their respective BMUs on the SOM. These BMUs constitute the states of the Markov models.

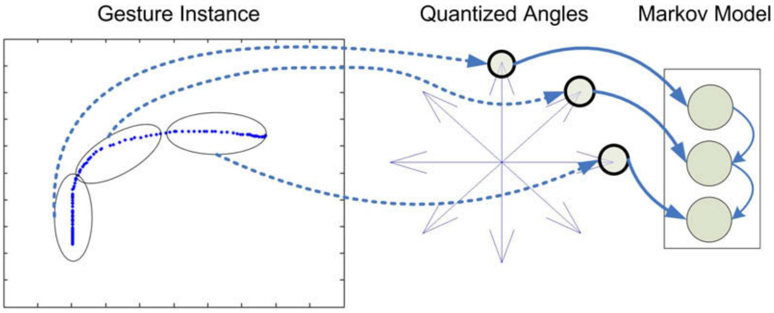

Fig. 3. Building a Markov model for a gesture's optical flow.

$$MM^{of} = \{MM_1^{of}, MM_2^{of}, \dots, MM_c^{of}\} : D''_j = \{G''_1, G''_2, \dots, G''_n\} \rightarrow MM_i^{of} \tag{6}$$

3.2.3. 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. In case the generalized median sequence belongs to the set S it is called Generalized Set Median.

Let M_j be the generalized median of the D'_j set, using as the Levenshtein distance L, a widely employed distance metric.

$$M_j = generalized\_median(D'_j) = \arg\min_g \sum_{G' \in D'_j} L(g, G') \tag{7}$$

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 3.3).

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

3.3. Gesture decoding

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

$$P\left(G'_k \mid MM_j^{som}\right) = \frac{\prod_{i=1}^{q} S_i^{som}}{q} : q = |G'_k| \tag{9}$$

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:

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

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 $P(z \mid u_{i-i}, MM_j^{som})$. $NF_{u_i}^{som}(z)$ 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:

$$P(G''_k \mid MM_j^{of}) = \frac{\prod_{i=1}^{q} S_i^{of}}{q} : q = |G''_k| \tag{12}$$

$$\begin{aligned} S_1^{of} &= \max_z \left(NF_{u_1}^{of}(z)\pi_j^{of}\right) \\ S_i^{of} &= \max_z \left(NF_{u_i}^{of}(z)P(z \mid u_{i-i}, MM_j^{of})\right) \end{aligned} \tag{13}$$

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

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}.

$$P(G'_k \mid M_j) = \frac{ML_j}{L(G'_k, M_j)} \tag{15}$$

Please note that $P(G'_k \mid M_j)$ is a similarity measurement and not a probability, since its value can be >1.

Finally, $P\left(G'_k \mid MM_j^{som}\right)P\left(G''_k \mid MM_j^{of}\right)P(G'_k \mid M_j)$ is calculated for all classes and the highest valued one is selected. Quality criteria can be further applied in the form of a threshold either to the overall evaluation of the gesture instance or to terms of the above equation, thus not allowing 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 and is resolved appropriately.

3.4. Error propagation study

In general the error definition for function $f(x, y)$, for independent error $\delta x, \delta y$ is:

$$\delta f^2 = \left(\frac{\partial f}{\partial x}\delta x\right)^2 + \left(\frac{\partial f}{\partial y}\delta y\right)^2 \tag{16}$$

We performed an error propagation study of the proposed system, focusing on the SOM decoding stage, i.e. on the evaluation of Eq. (9) $P\left(G'_k \mid MM_j^{som}\right)$ and $\prod_{i=1}^{q} S_i^{som}$. We investigated the effect that a random error $\delta x, \delta y$ in the detection of the hand point x, y in trajectory G_k (so that the detected trajectory point is $x + \delta x, y + \delta y$) has on the evaluation of $S_i^{som} = \max_z\left(NF_{u_i}^{som}(z)P\left(z \mid u_{i-i}, MM_j^{som}\right)\right)$. For simplification reasons, we assume that each trajectory point in G_k is mapped to a distinct BMU on the SOM so that $u'_t \neq u'_{t-1} \forall t \in [2, m]$ in G'_k. This assumption does not affect the overall error study, since without it the possible error would appear in a later repetition within the decoding loop, and simply provides a more straightforward way of dealing with the introduction of a random error.

In the case that the error is small enough resulting in mapping to the same node of the SOM, then no error is introduced in the decoding stage, since the error is absorbed by the SOM and not propagated to the rest of the decoding chain. This is because hand coordinates with relatively small variation are mapped to the same node of the SOM so that $BMU(x + \delta x, y + \delta y) = BMU(x, y)$. Simply put, for the rest of the decoding chain this error will disappear since it is compensated during the clustering process the SOM performs. On the other hand, when $\delta x, \delta y \gg : BMU(x + \delta x, y + \delta y) \neq BMU(x, y)$ then:

$$\begin{aligned} &u_i \neq u' \\ &u' = BMU(x,y) \\ &u_i = BMU(x+\delta x, y+\delta y) \end{aligned} \tag{17}$$

and the introduced error $\delta x, \delta y$ will affect S_i. Let u' be the most probable transition from node u_{i-1}, in MM_j^{som}, as resulted from the training stage and all other transitions probabilities from u_{i-1} are negligible:

$$\begin{aligned} &P\left(u'|u_{i-1}, MM_j^{som}\right) \to 1^- \\ &P\left(u|u_{i-1}, MM_j^{som}\right) \ll \forall u \neq u' \end{aligned} \tag{18}$$

As a consequence, $\delta S_i^{SOM} \approx NF_{u_i}^{som}(u')$, which is the neighboring relation of u_i, u' and is relative to the $\delta x, \delta y$ input error. Since u' becomes the new u_i, according to Eq. (11), the error is not propagated to the next steps of the recognition process.

In an abstraction attempt, the overall decoding stage operates as an energy maximization algorithm: at every step, it seeks to maximize S_i^{som} at the possible cost of choosing a different, but more probable BMU node for every model. The cost is located at $NF_{u_i}^{som}(u')$, since when choosing a different node the evaluation of a specific instance/model pair is penalized by this cost term in order to compensate for the node replacement. In other words, the algorithm appears to strive to converge to the model's most probable path, penalizing each deviation from this path with the respective nodes' neighboring relation. On the other hand, no node transition is eliminated, since all transitions are initialized with small but non-zero values and if a node has significant transition probability in the model's transition matrix and is also adjacent to the original node then it can be selected as the winner node.

Concluding, random errors in the input stream are not propagated in the decoding process, increasing the robustness of the proposed architecture. Additionally, user variability in the performance of a gesture is tackled by incorporating the neighboring relationship of the SOM nodes during the decoding of the gesture, which penalizes but does not exclude gesture instance variations of the same class from the classification process. The appropriate inclusion of the neighboring characteristic in the overall decoding process also ensures adaptive performance in dynamic backgrounds and user diversity in terms of gesture performance or anatomical and ergonomic characteristics. Errors introduced by feature extraction algorithms or deviations introduced by individual differences in user performance are either absorbed by internal mechanisms (median filter or SOM mapping) or influence the evaluation of a particular node or transition and not the entire trajectory.

3.5. Experimental results

Validation of the proposed architecture was performed on an artificial dataset formed by the 30 gestures seen in Fig. 4 and consisting of 10 repetitions each. The set of coordinates of the right hand were gathered according to the procedure described above (Section 3.1) and formed a gesture dataset containing 30 gesture categories, 10 repetitions each. As can be seen in Fig. 4 the classes vary in complexity from very simple directive gestures to very complex ones.

Fig. 5 shows the U-matrix of the self organizing map, trained with all the gestures instances of the dataset. Blue areas on the U-matrix depict regions where the vectors of map units on the input space are close, while red areas depict the opposite relation between the map units. The gesture coordinates cover the larger part of the input space, due to the diversity of the gesture categories, and of the variation of the gesture instances within each category. Under these conditions the trained map presents, as expected, uniformity in its larger part, which is the desirable result for the map's role in the gesture transformation.

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

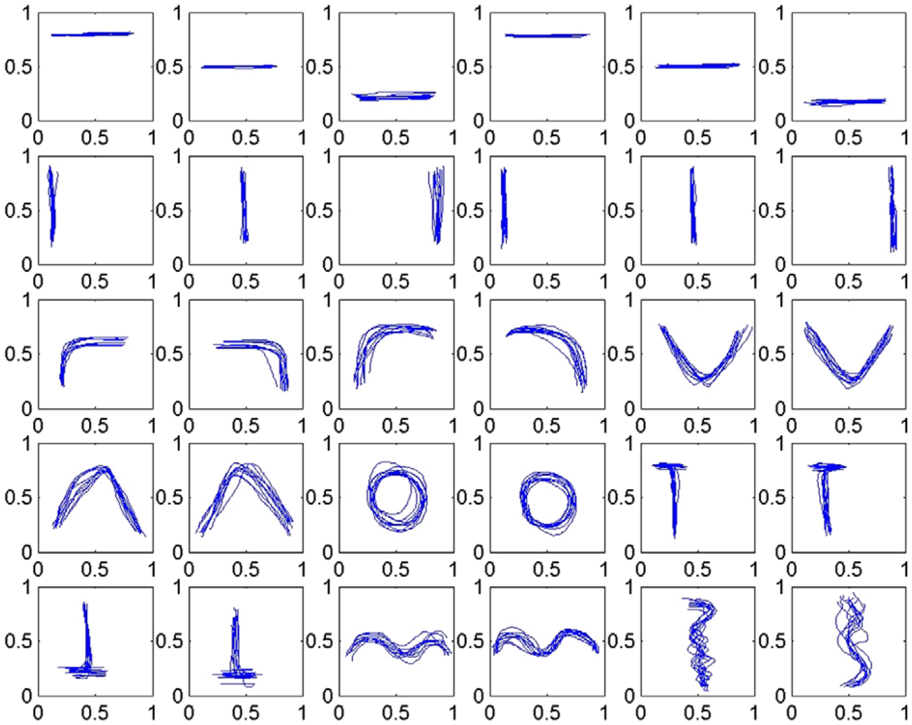

Fig. 4. The gesture dataset.

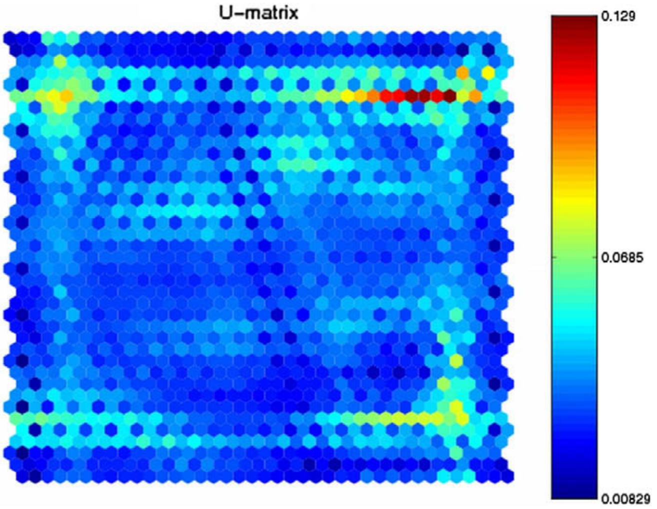

Fig. 5. The trained SOM U-matrix.

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 (2 GHz Dual Core, 3GB RAM) and for training all thirty classes 0.23 s 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 ms per gesture instance per gesture class, a performance which establishes the overall architecture suitable for real time applications.

Additional experiments were performed concerning the SOM's neighborhood radius and function. We concluded that a radius of 2 and gaussian function type performed better. Finally, the thresholds for the quality criteria, both absolute participation probability and winner-second difference, were determined experimentally.

In order to compare the results of our system with the most commonly used approach in the literature we implemented a HMM-based classifier, training one continuous left-to-right model per gesture class using a mixture of three 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 in an average recognition rate of 86.36%.

This experimental study indicates that the 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.

4. Conclusions and future work discussion

In this paper, we proposed 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 time applications.

An obvious extension to the proposed approach would be the incorporation of hand shape features in the overall decision mechanism. Even though SOM mapping seems unsuitable for hand shapes, since 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, utilizing hand shape information and possible knowledge based fusion. Adding a 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 Nintendo 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 three dimensions. A movement of the user's hand can be detected either automatically, when acceleration values are above a certain threshold, using velocity and position measurements obtained via single and double integration of the acceleration signal, respectively. 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 Schlömer et al. (2008) in gaming environments.

Furthermore, testing on additional corpora and comparing with other more sophisticated classifiers used in gesture recognition or HMM variations is certainly needed in order to validate the overall method. Gesture prediction and continuous gesture analysis are also included in future work.

References

Black, M.J., Jepson, A.D., 1998. Recognizing temporal trajectories using the condensation algorithm. In: 3rd Internat. Conf. on Face and Gesture Recognition.

Braffort, Annelies, Gherbi, Rachid, Gibet, Sylvie, Richardson, James, Teil, Daniel (Eds.), 1999. Gesture-Based Communication in Human–Computer Interaction. Springer, Berlin, Heidelberg.

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

Caridakis, G., Karpouzis, K., Pateritsas, C., Drosopoulos, A., Stafylopatis, A., Kollias, S., 2008. Hand trajectory-based gesture recognition using self-organizing feature maps and Markov models. In: IEEE Internat. Conf. on Multimedia and Expo.

Caridakis, G., Pateritsas, C., Drosopoulos, A., Stafylopatis, A., Kollias, S., 2007. Probabilistic video-based gesture recognition using self-organizing feature maps. In: 17th Internat. Conf. on Artificial Neural Networks (ICANN 2007), September 9–13, Porto, Portugal.

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

Coogan, Thomas, Awad, George, Han, Junwei, Sutherland, Alistair, 2006. Real time hand gesture recognition including hand segmentation and tracking. Adv. Visual Comput., 495–504.

Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., Votsis, G., Kollias, S., Fellenz, W., Taylor, J.G., 2001. Emotion recognition in human–computer interaction. Signal Process. Magaz., IEEE 18 (1), 32–80.

Hong, P., Turk, M., Huang, T.S., 2000. Gesture modeling and recognition using finite state machines. In: 4th IEEE Internat. Conf. and Gesture Recognition.

Hossain, Monowar, Jenkin, Michael, 2005. Recognizing hand-raising gestures using hmm. In: CRV'05: Proc. 2nd Canadian Conf. on Computer and Robot Vision. IEEE Computer Society, Washington, DC, USA, pp. 405–412.

Huang, Chung-Lin, Huang, Wen-Yi, 1998. Sign language recognition using model-based tracking and a 3d Hopfield neural network. Machine Vision Appl. 10 (5–6), 292–307.

Huang, Chung-Lin, Wu, Ming-Shan, Jeng, Sheng-Hung, 2000. Gesture recognition using the multi-pdm method and hidden Markov model. Image Vision Comput. 18 (11), 865–879.

Juang, C.-F., Ku, K.C., 2005. A recurrent fuzzy network for fuzzy temporal sequence processing and gesture recognition. IEEE Trans. Systems Man Cybernet., Part B 35, 646–658.

Kohonen, Teuvo, 1998. The self-organizing map. Neurocomputing 21 (1–3), 1–6.

Kohonen, Teuvo, 2001. Self-Organizing Maps, third ed. Springer.

Mantyla, V.-M. , Mantyjarvi, J., Seppanen, T., Tuulari, E., 2000. Hand gesture recognition of a mobile device user. In: IEEE Internat. Conf. on Multimedia and Expo.

McNeill, David, 1992. Hand and Mind: What Gestures Reveal About Thought. University of Chicago Press.

Miners, B.W., Basir, O.A., Kamel, M., 2002. Knowledge-based disambiguation of hand gestures. IEEE Internat. Conf. Systems Man Cybernet. 5 (October), 5–6.

Mitra, S., Acharya, T., 2007. Gesture recognition: A survey. IEEE Trans. Systems Man Cybernet., Part C: Appl. Rev. 37 (3), 311–324.

Ong, S.C.W., Ranganath, S., 2005. Automatic sign language analysis: A survey and the future beyond lexical meaning. Trans. Pattern Anal. Machine Intell. 27 (6), 873–891.

Patwardhan, Kaustubh Srikrishna, Roy, Sumantra Dutta, 2007. Hand gesture modelling and recognition involving changing shapes and trajectories, using a predictive eigentracker. Pattern Recognition Lett. 28 (3), 329–334.

Raytchev, Bisser, Hasegawa, Osamu, Otsu, Nobuyuki, 2000. User-independent online gesture recognition by relative motion extraction. Pattern Recognition Lett. 21 (1), 69–82.

Schlömer, Thomas, Poppinga, Benjamin, Henze, Niels, Boll, Susanne, 2008. Gesture recognition with a Wii controller. In: TEI'08: Proc. 2nd Internat. Conf. on Tangible and Embedded Interaction. ACM, New York, NY, USA, pp. 11–14.

Starner, T., Weaver, J., Pentland, A., 1998. Real-time American Sign Language recognition using desk and wearable computer-based video. IEEE Trans. Pattern Anal. Machine Intell..

Su, Mu-Chun, 2000. A fuzzy rule-based approach to spatio-temporal hand gesture recognition. IEEE Trans. Systems Man Cybernet., Part C: Appl. Rev. 30 (2), 276–281.

Vinciarelli, A., Pantic, M., Bourlard, H., Pentland, A., 2008. Social signal processing: state-of-the-art and future perspectives of an emerging domain. ACM Multimedia, 1061–1070.

Wilson, A., Bobick, A., 1999. Parametric hidden Markov models for gesture recognition. IEEE Trans. Pattern Anal. Machine Intell. 21 (9).

Wong, Shu-Fai., Cipolla, R., 2006. Continuous gesture recognition using a sparse Bayesian classifier. In: 18th Internat. Conf. on Pattern Recognition, 2006. ICPR 2006, vol. 1, pp. 1084–1087.

Wu, Y., Huang, T., 2001. Hand modeling, analysis, and recognition for vision-based human–computer interaction. IEEE Signal Process. Magaz. 18, 51–60.

Yang, M.H., Ahuja, N., Tabb, M., 2002. Extraction of 2d motion trajectories and its application to hand gesture recognition. IEEE Trans. Pattern Anal. Machine Intell. 24, 1061–1074.