An Human-Computer Interface Using Facial Gestures for the Game of *Truco*

Gonzalo Castillo, Santiago Avendaño, and Norberto Adrián Goussies

Departamento de Computación,
Facultad de Ciencias Exactas y Naturales,
Universidad de Buenos Aires
Ciudad Universitaria, C1428EGA, Buenos Aires, Argentina
{gcastilo,savendano,ngoussie}@dc.uba.ar

Abstract. In this work we present a method to detect and recognize the signs of the card game of *Truco* which are a subset of facial gestures. The method uses temporal templates to represent motion and later extract features. The proposed method works in real time, allowing to use it as an human-computer interface , for example, in the context of the card game of *Truco* . To the best of our knowledge this is the first work that uses detection of facial gestures in the context of a game.

Keywords: facial gesture, temporal templates, truco.

1 Introduction

The *Truco* is a card game originary from Spain and played in South-America. One of its main objectives is to deceive the opponent in order to get a higher score. It is played using a spanish deck by two, four or six players divided into two teams. In order to develop a common strategy it is important that teammates inform each other the cards that they have in the hand. The players of each team use a set of facial gestures to secretly inform the others players of the same team the cards they have. The most generally accepted gestures are shown in Figure 1.

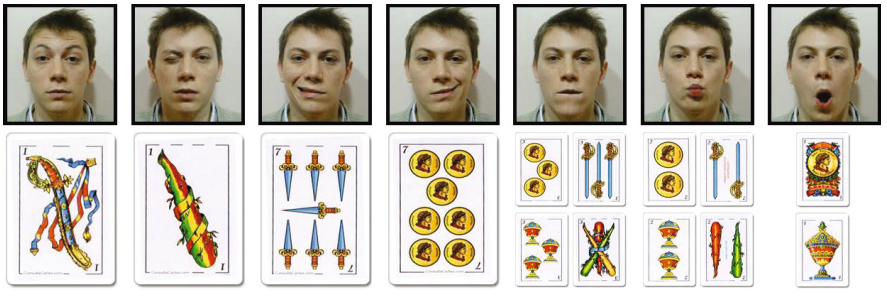

Fig. 1. Truco gestures and their corresponding cards. From left to right: ace of swords, ace of clubs, seven of swords, seven of gold, any three, any two, any other ace.

L. Alvarez et al. (Eds.): CIARP 2012, LNCS 7441, pp. 276–283, 2012.

In this paper we address the problem of detection and recognition of *Truco* signs which could be integrated into a computer *Truco* player in order to replace one of the human players by a computer player. The proposed interface will give the other players a more realistic and comfortable experience.

In other words, we present a novel human-computer interface that understands commands using facial gestures. This is a different type of interface to previous human-computer interfaces which usually understands commands using upper-body gestures [1, 2]. Additionally, the goal of this paper is different to the one of [3–6] where authors propose a system to understand human emotions using facial gestures. Note that when we design a facial gesture interface we should allow the user to do gesticulations that are not part of the interface, a challenge that is not present for other facial analysis systems. Also, the interface should work for different people and have a fast response time.

Our proposal presented in this paper uses temporal templates [2] to represent motion and later extract facial features. In addition, we developed a real-time on-line *Truco* sign recognition system. Figure 2 gives an overview of our system. First, we locate frontal faces [7] in each frame and at the same time we update the temporal template. Next, if the a *Truco* sign is spotted we extract features and recognize the type of *Truco* gesture.

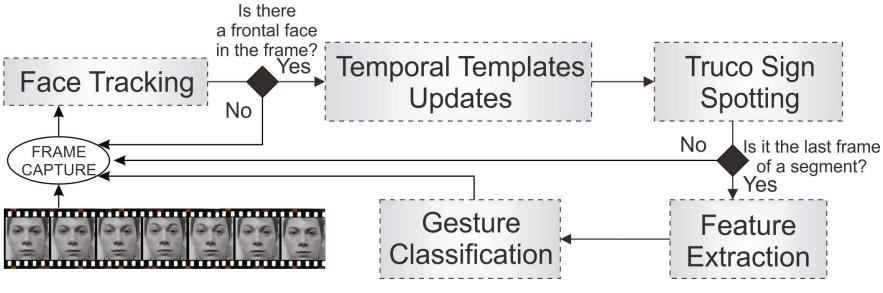

Fig. 2. Overview of the proposed system

The remainder of this paper is organized as follows: Section 2 provides an overview of related work. In Section 3 we describe the representation of the facial motion. In Section 4 we present different features for the facial motion. In Section 5 we explain the *Truco* gesture spotting method. We show experimental results in Section 6. Finally we conclude our work in Section 7.

2 Related Work

In this Section we mention the most relevant contributions for our work. Initially in [8] Kalman filters were used to predict and track the position of certain facial features in real time. Using these positions it is possible to detect gestures such as: yes, no, maybe, look up or down.

In [9–12] methods to detect facial gestures are proposed. In [9] vector spaces of facial gestures are built using the method of eigenfaces. Using these vector spaces, the facial gestures are represented by a limited number of descriptors. A different representation

based on optical flow was proposed in [10] to detect and recognize the six universal facial expressions (happiness, sadness, surprise, fear, anger and disgust). A cylindrical 3D model was built in [11] to perform face tracking. Additionally, the residual error was modeled as a linear combination of templates of facial movement to deduce the expression a person was doing. Another 3D model was proposed in [12] to detect the position and tilt of the face. Furthermore, the problem of detecting facial gestures in the presence of head movement was studied.

New methods for the problem of recognition of Action Units [13] were proposed in [3, 4, 6]. Temporal templates originary proposed to recognize human actions [2] were adapted in [3] for recognition of Action Units using nearest neighbor. In [4] the authors proposed a system that detects Action Units and analyses the temporal behavior, using spatio-temporal features computed by tracking 20 facial points. Both gestures and their temporal segments were recognized by Supports Vectors Machines. Later, [6] suggested two approaches based on dynamic textures to recognize Action Units and their temporary models. The first is an extension of the method of temporal templates based on Motion History Image. The second method relies on nonrigid registration using free-form deformations.

Our approach extends the work of [3, 6] on temporal templates and Motion History Images for facial gesture recognition. We use Directional Motion History Images [14] in order to overcome the problem of self-occluding gestures. Although the *Truco* gestures can be partly mapped to Action Units and there are methods based on temporal templates for recognizing Action Units, our method is not only able of recognize facial gestures but also performs real-time detection.

3 Motion Representation

The gestures of the face are described by temporal templates based on Motion History Image (MHI) and Motion Energy Image (MEI) presented by Bobick and Davis in [2]. The MHI is a scalar-valued image where intensity is a function of recency of motion. The MEI represents where motion has occurred in an image sequence. A temporal template is a vector-valued image where each component of each pixel is some function of the motion at that pixel location, for example the MEI and the MHI.

Let $\mathbf{I} = (I_1, \ldots, I_t)$ be an image sequence of t frames, ψ_k is a binary image indicating associated the motion in the I_k frame :

$$\psi_k(x,y) = \begin{cases} true & \text{if } \Delta_k^\beta(x,y) \geq \mu \\ false & \text{if } \Delta_k^\beta(x,y) < \mu \end{cases} \tag{1}$$

$$\Delta_k^\beta(x,y) = \begin{cases} |I_k(x,y) - I_1(x,y)| & \text{if } k \leq \beta \\ |I_k(x,y) - I_{k-\beta}(x,y)| & \text{if } k > \beta \end{cases} \tag{2}$$

where $\mu \in \mathbb{N}$ is the threshold of motion and $\beta \in \mathbb{N}$ is the temporal gap between subtracted frames.

In our work, we assume that the Truco gestures have a maximum duration of α frames. Therefore, we do not need to take into account all the frames but only α previous

frames when computing the temporal template for a given frame $i \leq t$. We encode the motion information for the range of time in the MHI M as follows:

$$M_i^\alpha(x,y) = \begin{cases} \min(i,\alpha) & \text{if } \psi_i(x,y) \\ M_{i-1}^\alpha(x,y) & \text{if } \neg\psi_i(x,y) \text{ and } i < \alpha \\ \max(0, M_{i-1}^\alpha(x,y) - 1) & \text{if } \neg\psi_i(x,y) \text{ and } i \geq \alpha \end{cases} \tag{3}$$

The range of the MHI are the natural numbers in $[0, \alpha]$. The pixels with highest value are those that had motion most recently in the last α frames. The comparison of two MHIs with different α is meaningless because their range depends on α. In order to solve this problem we define the Normalized MHI, $NMHI$, whose range is $[0, 1]$:

$$NMHI_i^\alpha(x,y) = \begin{cases} 0 & \text{if } t = 0 \\ M_i^\alpha(x,y)/t & \text{if } 1 < t < \alpha \\ M_i^\alpha(x,y)/\alpha & \text{if } t \geq \alpha \end{cases} \tag{4}$$

The Motion Energy Image E^α can be computed using the MHI M^α:

$$E_i^\alpha(x,y) = M_i^\alpha(x,y) > 0 \tag{5}$$

3.1 Directional Motion History Image (DMHI)

Most of the Truco signs can be separated in three different stages. In the first stage the facial muscles are contracted. Then, the sign is maintained for a few frames. Finally, the facial muscles are relaxed. Since, the stages are performed in the same spatial area, if we use the MHI as defined above each stage overrides the information of the previous one. In other words, the Truco signs are self-occluding. In order to solve this problem in [14, 15] the authors proposed to extend the information coded in the MHI. In their approach, gradient-based optical flow is calculated between two consecutive frames and split it into four channels. Based on such a strategy, a four-directional motion templates for left, right, up and down directions can be obtained. The Directional Motion History Image (DMHI) is defined as:

$$DM_t^{\alpha,\gamma}(x,y) = \begin{cases} M_t^\alpha(x,y) & \text{if } \angle_t^\alpha(x,y) \in \gamma \\ DM_{t-1}^{\alpha,\gamma}(x,y) & \text{if } \angle_t^\alpha(x,y) \notin \gamma \text{ and } i < \alpha \\ \max(0, DM_{t-1}^{\alpha,\gamma}(x,y) - 1) & \text{if } \angle_t^\alpha(x,y) \notin \gamma \text{ and } i \geq \alpha \end{cases} \tag{6}$$

where γ is one of the possible directions : left, right, up and down. The function $\angle$ is defined as:

$$\angle_t^\alpha(x,y) = \arctan\left(\frac{G_y * M_t^\alpha}{G_x * M_t^\alpha}\right) \tag{7}$$

where $G_y * MHI_t$ and $G_x * MHI_t$ are the images resulting from convolving the MHI with the vertical and horizontal Sobel filters respectively. The function $\angle_t$ provides a good approximation of the motion angle at each pixel. Although there are more exact implementations for obtaining the motion angle (for example [6]) we have chosen this one because is the fastest one, allowing the system to work in real-time.

4 Feature Extraction and Classification

In this section we describe the feature extraction method which will be used as input for the classifiers. Two main types of features has been proposed for MHIs. The first type are based on image moments [16] and are common features for body motion [2]. The second type are features common in face motion [6] and aggregate the motion in different areas of the face. In this work we use the second type of feature.

The first step is to divide the face using an uniform grid of r rows and c columns [3]. The features computed using a regular grid have tolerance to small rotations, displacements and scale changes. The level of tolerance is given by the number of cells in the grid. In this work we use cells without overlapping of the same size.

Then, the motion of a DMHI, NMHI or MEI I is aggregated for each region R. We use the function C_R defined as:

$$C_R(I) = \frac{1}{|R|} \sum_{(x,y)\in R} I(x,y) \tag{8}$$

Note that the function C_R assigns higher values to recent facial motion than to motion that occurs further in the past when computed over a NMHI or DMHI while this is not true for MEI. The final feature vector consists of the concatenation for all regions R of the values $C_R(NM^\alpha)$, $C_R(E^\alpha)$ and $C_R(DM_t^{\alpha,\gamma})$ and for all directions γ. As a result the feature vector has information about where and how the motion is present and the direction of that motion.

We use the LIBSVM [17] implementation of Support Vector Machines (SVM). We use the one-versus-one approach for mutliclass classification with the radial basis function. Given a feature vector x our classifier f outputs a vector y containing one probability per *Truco* sign which are estimated using the implementation of [18].

5 *Truco* Sign Spotting

During the development of a round in the *Truco* the teammates talk and make facial gestures that are not always *Truco* signs. Therefore it is important to distinguish between the *Truco* signs and other types of gestures. This is an important requirement so the players can enjoy the game without any restrictions. In other words in this step the objective is to distinguish relevant gestures from other type of movements. The output of this step is the temporal location of the gestures, i.e. the first and last frame of a *Truco* gesture.

This is an extremely difficult task because there is a huge number of facial gestures that has similar motion patterns and occur in the same facial area. Furthermore, we are interested in building a system that works online, i.e. we do not have the entire image sequence in advance, each frame arrives as it is captured by the camera. Therefore, the method proposed here has to find the beginning and ending of the gesture using only the information prior the current frame.

A similar problem was studied in the design of human-computer interfaces for video games using computer vision techniques [1, 19]. The main difference with this work is

that the methods proposed by those authors are mostly for recognition of upper-body gestures but in this paper we are concerned with facial gestures. It is interesting to note that this problem has not received much attention in the area of facial gestures, probably because previous work in the area was not focused in human-computer interfaces using facial gestures.

Bearing in mind that we are interested in spotting *Truco* gestures and that they are only sent to the other teammate when both players are looking at each other, the first step in our gesture spotting method is to consider only the frames that have a frontal face.

In the following step we look for facial movements that are a potential *Truco* gesture. Again, we are going to use specific knowledge on *Truco* gestures. First we note that *Truco* gestures are expressed moving parts of the face that have approximately the same size. Therefore when a *Truco* gesture is signaled areas of approximately the same size must appear in the MEI. Following this idea we define the amount of motion at frame k as:

$$AM_k = C_R(E_k^\alpha) \tag{9}$$

where C_R is the function defined in Eq. (8) and R is the union of the regions of the face.

We define two thresholds τ, δ as the minimum and maximum of AM_k, respectively, that a frame must have in order to be considered as part of a gesture. The definition of a maximum amount of motion helps to differentiate motion generated by a gesture as opposed to motion produced by movement of the head. Also, *Truco* gestures have a minimum duration, therefore we define another threshold υ as the minimum number of continuous frames of motion a gesture must have. Experimentally we determined that $\tau = 0.02$, $\delta = 0.14$, $\upsilon = 10$ gives excellent results. To sum up, we only consider intervals such that the amount of motion for each frame is between τ and δ and have at least υ frames.

6 Experiments

Two types of experiments were performed: off-line isolated *Truco* sign recognition and real-time on-line gesture recognition. The goal of the first set is to examine the effectiveness of the features as well as the classification performance and generalization. In the second set the complete Truco sign system is tested.

6.1 Isolated Truco Sign Recognition

Two subjects were asked to perform all the Truco signs in front of a video camera. The total number of Truco signs samples collected were 229. Each sample gesture lasts about 1s and runs at 25 frames per second, the spatial resolution is 640×480 pixels and the face is at least 100×100 pixels.

To test robustness and generalization ability we train using the samples of one subject and test using the samples of the other subject. This way, we can also find out if the system works with different people without having to be trained for each person. When we train using Subject A and test on Subject B we obtain an accuracy of 89%, in the other case we obtain an accuracy of 92%. As expected, the *Truco* signs that have motion in the area of the mouth, are the ones that were harder to differentiate.

6.2 Real-Time Truco Signs Spotting and Recognition

Here we analyze the experiments for the real-time on-line system to detect and recognize the Truco gestures. Our system was run on a desktop computer having four core AMD Phenom II X4 64bits and 4 Gb of RAM. We trained the system using the complete dataset of 229 samples. The accuracy of the on-line system ranged from 90% to 100%. The system runs at 22 frames per second, most of the time was spent in updating the MHI and DMHI.

7 Conclusions

We described a real-time on-line system for detecting and recognizing *Truco* signs. As far as we know this is the first paper in developing an human-computer interface that understands commands using facial gestures in the context of a game. Finally, our approach works with different people without having to be trained for each person.

In the future we hope to develop a complete *Truco* computer player, capable of sending and receiving signs and build a strategy for the round. Also, the player should have a verbal interface in order to understand the different commands of the teammate and the other players.

References

1. Kang, H., Lee, C.W., Jung, K.: Recognition-based gesture spotting in video games. Pattern Recogn. Lett. 25, 1701–1714 (2004)
2. Davis, J.W., Bobick, A.F.: The Representation and Recognition of Human Movement Using Temporal Templates. In: CVPR 1997: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR 1997). IEEE Computer Society Press (1997)
3. Valstar, M.F., Patras, I., Pantic, M.: Facial action unit recognition using temporal templates. In: Proceedings of IEEE Int'l Workshop on Robot-Human Interaction (RO-MAN 2004), Kurashiki, Japan, pp. 253–258 (September 2004)
4. Valstar, M.F., Pantic, M.: Fully automatic facial action unit detection and temporal analysis. In: Proceedings of IEEE Int'l Conf. Computer Vision and Pattern Recognition (CVPR 2006), New York, USA, vol. 3, p. 149 (June 2006)
5. Valstar, M., Pantic, M., Patras, I.: Motion history for facial action detection from face video. In: Proceedings of IEEE Int'l Conf. Systems, Man and Cybernetics (SMC 2004), The Hague, Netherlands, pp. 635–640 (October 2004)
6. Koelstra, S., Pantic, M., Patras, I.: A dynamic texture based approach to recognition of facial actions and their temporal models. IEEE Transactions on Pattern Analysis and Machine Intelligence 32, 1940–1954 (2010)
7. Viola, P., Jones, M.: Robust real-time face detection. International Journal of Computer Vision 57, 137–154 (2004)
8. Zelinsky, A., Zelinsky, E., Heinzmann, J.: Real-time visual recognition of facial gestures for human-computer interaction. In: Proceedings of the Int. Conf. on Automatic Face and Gesture Recognition, pp. 351–356. IEEE Computer Society Press (1996)
9. Algorri, M.E., Escobar, A.: Facial gesture recognition for interactive applications. In: Proceedings of the Fifth Mexican International Conference in Computer Science, pp. 188–195. IEEE Computer Society, Washington, DC (2004)

10. Naghsh-Nilchi, A.R., et al.: An efficient algorithm for motion detection based facial expression recognition using optical flow (2006)
11. La Cascia, M., Valenti, L., Sclaroff, S.: Fully automatic, real-time detection of facial gestures from generic video. In: IEEE 6th Workshop on Multimedia Signal Processing 2004, pp. 175–178 (2004)
12. Liao, W.-K., Cohen, I.: Classifying facial gestures in presence of head motion. In: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005) Workshops, vol. 03, p. 77. IEEE Computer Society, Washington, DC (2005)
13. Ekman, P., Friesen, W.: Facial Action Coding System: A Technique for the Measurement of Facial Movement. Consulting Psychologists Press, Palo Alto (1978)
14. Ahad, M.A.R., Ogata, T., Tan, J.K., Kim, H., Ishikawa, S.: Motion recognition approach to solve overwriting in complex actions. In: FG, pp. 1–6. IEEE (2008)
15. Ahad, M., Tan, J., Kim, H., Ishikawa, S.: Solutions to motion self-occlusion problem in human activity analysis. In: 11th International Conference on Computer and Information Technology, ICCIT 2008, pp. 201–206 (December 2008)
16. Hu, M.-K.: Visual pattern recognition by moment invariants. IRE Transactions on Information Theory 8, 179–187 (1962)
17. Chang, C.-C., Lin, C.-J.: Libsvm: A library for support vector machines. ACM Trans. Intell. Syst. Technol. 2, 27:1–27:27 (2011)
18. Lin, H.-T., Lin, C.-J., Weng, R.C.: A note on platt's probabilistic outputs for support vector machines. Mach. Learn. 68, 267–276 (2007)
19. Freeman, W.T., Anderson, D.B., Beardsley, P.A., Dodge, C.N., Roth, M., Weissman, C.D., Yerazunis, W.S., Kage, H., Kyuma, K., Miyake, Y., Tanaka, K.-i.: Computer vision for interactive computer graphics. IEEE Comput. Graph. Appl. 18, 42–53 (1998)