International Journal of Computer Science and Applications,

Vol. 7 No. 3, pp. 124 - 139, 2010

HUMAN-COMPUTER INTERFACE BASED ON VISUAL LIP MOVEMENT AND GESTURE RECOGNITION

Piotr Dalka

Gdansk University of Technology, Multimedia Systems Department,
Narutowicza 11/12, 80-233, Gdansk, Poland
piotr.dalka@sound.eti.pg.gda.pl
http://www.multimed.org

Andrzej Czyzewski

Gdansk University of Technology, Multimedia Systems Department,
Narutowicza 11/12, 80-233, Gdansk, Poland
ac@pg.gda.pl
http://www.multimed.org

The multimodal human-computer interface (HCI) called LipMouse is presented, allowing a user to work on a computer using movements and gestures made with his/her mouth only. Algorithms for lip movement tracking and lip gesture recognition are presented in details. User face images are captured with a standard webcam. Face detection is based on a cascade of boosted classifiers using Haar-like features. A mouth region is located in the lower part of the face region. Its position is used to track lip movements that allows a user to control a screen cursor. Three lip gestures are recognized: mouth opening, sticking out the tongue and forming puckered lips. Lip gesture recognition is performed by an artificial neural network and utilizes various image features of the lip region. An accurate lip shape is obtained by the means of lip image segmentation using fuzzy clustering.

Keywords: human-computer interface, image processing; lip gestures, artificial neural network.

1. Introduction

There is an increasing need for development of new human-computer interfaces (HCI) [Baecker *et al.* (1995)][Sears and Jacko (2007)]. They are especially useful in situations when it is not possible, difficult or ineffective to use traditional input devices, like a keyboard and a mouse. The main goal of each HCI application is to make working with a computer as natural, intuitive and effective as possible.

One of the main areas of applications of new human-computer interfaces is to make possible for people with permanent or temporal disabilities to use computers in an efficient way. There are two main types of such solutions [Aggarwal and Cai (1999)]. The first group utilizes devices mounted directly on the user's body. Applications in the second group are contactless and they use remote sensors only, therefore they are much more comfortable for a user. Amongst contactless solutions, vision-based human-computer interfaces are the most promising ones. They utilize cameras and image processing algorithms to detect signs and gestures made by a user and execute configured

actions. The most common vision-based applications employ eye and hand tracking [Duchowski (2002)][Shin and Chun (2007)], while lip movements and gestures does not seem to be used for this purpose.

Lip image segmentation and lip movement tracking is a very complicated task, mainly because of a very small contrast between lips and a face skin. Many approaches to this task may be found in the literature. The latest solutions do not require any user preparation, such as placing marks on a user face or particular make-up. Lip image is usually segmented by the means of transforming RGB color space into CIE-LUV, HSV, YCbCr or a similar space [de Dios and Garcia (2004)][Zhang and Mersereau (2000)][Tsapatsoulis *et al.* (2000)]. In [Eveno *et al.* (2001)] authors propose a new transformation called a chromatic curve map. In [Guan (2008)] an automatic lip segmentation algorithm is described based on the wavelet multi-scale edge detection across the discrete Hartley transform.

Another methods for lip segmentation utilize creating a lip shape model and fitting it to a lip image. Lip shape models may be based on deformable templates [Liewa *et al.* (2000)], active contour models [Sasaki *et al.* (2004)] or active shape models [Moran and Pinto (2007)] and they generally use a set of feature points to approximate the lip contours.

An interesting method, proposed by [Leung *et al.* (2004)], combines both color dissimilarity between lip and skin and a spatial distance from an ellipse approximating lip shape in order to facilitate lip segmentation. This method was adopted for the purpose of the human-computer interface called LipMouse that is presented in this paper.

The paper is organized as follows. LipMouse interface is described in Section 2. Section 3 presents an overview of a methodology used by the application. Section 4 describes how a face and a mouth are localized in video frames while section 5 explains a method for translating mouth (head) movements into screen cursor movements. Section 6 contains details regarding lip gesture recognition. Results of experiments are summarized in section 7. Section 8 concludes the paper.

2. Human-Computer Interface Description

LipMouse is a name of a novel, patent-pending, contactless, human-computer interface that allows a user to work on a computer using movements and gestures made with his or her mouth only [Dalka and Czyzewski (2009)]. LipMouse is an application running on a standard PC computer. It requires only one hardware component: a display-mounted, standard web camera that captures images of the user face.

The main task of LipMouse is to detect and analyze images of user's mouth region in a video stream acquired from a web-camera. All movements of mouth (or head) are converted to movements of the screen cursor. Various parameters regarding speed of the cursor movement may be set according to user preferences. LipMouse also detects three mouth gestures: opening the mouth, sticking out the tongue and forming puckered lips (as for kissing). Each gesture may be associated with an action, which may be freely chosen by a user. Possible actions include clicking or double-clicking various mouse buttons,

moving mouse wheel – both horizontally and vertically and others. Many actions may be defined as single or continuous ones. The single actions are executed only once, in the very moment when a new gesture is detected; continuous actions are executed as long, as a gesture is kept. For example, opening mouth gesture may be connected with an action executing single left mouse button click in the moment, when the mouth is opened, or with an action that keeps left mouse button pressed as long, as a user keeps his/her mouth open.

Additionally, based on the mouth (head) movement speed, LipMouse detects two other gestures (head shaking "Yes" and "No") that consist in shaking a head energetically in vertical or horizontal direction.

Fig. 1 presents the main window of the application. It allows a user to configure LipMouse according to his preferences. In the right part of the window, vertically-flipped video frames from the camera are displayed (a user sees his mirror-like reflection). In the frames, the mouth region is denoted with a rectangle, and the lip shape is denoted with an

Fig. 1. LipMouse application main window

ellipse.

Before a user starts working with LipMouse, a short calibration lasting about 30 seconds needs to be executed. During the calibration, the user is asked to perform some head movement and gestures according to the instructions seen on the screen. The purpose of the calibration is to tune LipMouse to detect gestures made by the user in the current lighting conditions.

The target users for the tool are people who, for any reason, cannot or do not want to use traditional input devices. Therefore LipMouse is a solution enabling severely disabled and paralyzed people to use a computer and communicate with the surrounding world. No user adaptation, such as placing marks on the face, is required in order to successfully work with LipMouse.

3. LipMouse Methodology Overview

Fig. 2 presents a scheme of the algorithm used in LipMouse. First, a user's face is detected in every image frame captured by a web camera. Further stages of the algorithm are restricted to the ROI containing the user's face. Then, a mouth region is localized and its shift from the reference mouth position is calculated. This shift is directly used to move a screen cursor. Simultaneously, a small region (blob) placed on user lips is found in mouth region. This blob is used as a starting condition for an iterative method for lip shape extraction. Lip shape and lip region image features are used by an intelligent decision system utilizing an artificial neural network to classify gestures made by a user.

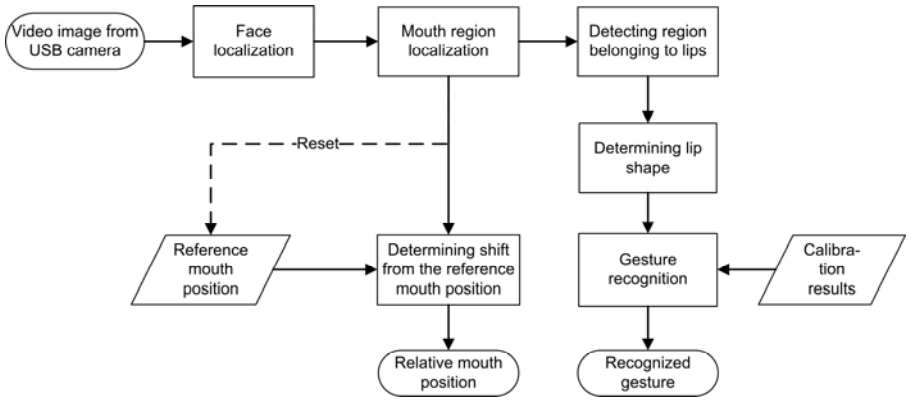

Fig. 2. Scheme of the LipMouse human-computer interface algorithm

4. Face and Mouth Position Detection

A cascade of boosted classifiers working with Haar-like features is used to detect a user's face in images captured by a web camera [Viola and Jones (2001)][Lienhart and Maydt (2002)]. It is a very efficient and effective algorithm for visual object detection.

Each classifier in the cascade consists of a set of weak classifiers based on one image feature each. Features used for face detection are grey-level differences between sums of pixel values in different, rectangle regions in an image window. The window slides over the image and changes its scale. Image features may be computed rapidly for any scale and location in a video frame using integral images [Viola and Jones (2001)].

For each window, the decision is made whether the window contains a face; all classifiers in the cascade must detect a face for the classification result to be positive. If any classifiers fails to detect a face, the classification process is halted and the final result is negative. Classifiers in the cascade are trained with AdaBoost algorithm that is tuned to minimize false negatives error ratio.

Classifiers in the cascade are combined in the order of increased complexity; initial classifies are based on a few features only. This makes possible for the algorithm to work

in the real time because it allows background regions of the image to be quickly discarded while spending more computation on promising regions.

Face detection algorithm finds location of all faces in every video frame. It is assumed, that only one person is present in the camera field of view therefore only the first face location is used for further processing.

In order to increase speed of the face detection and to make sure that the face is large enough to recognize lip gestures, the minimal width of a face was set to the half of the image frame width.

Sample results of face detection and mouth region finding are pictured in Fig. 3. The mouth region is localized arbitrary in the lower part of the face region detected. It is defined by the half-ellipse horizontally centered in the lower half of the face region. The width and the height of the half-ellipse is equal to the half of the height and half of the width of the face region, respectively. Only the mouth region of each video frame is used for lip gesture recognition.

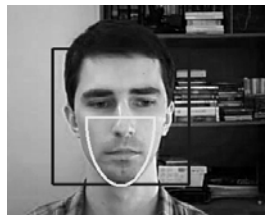

Fig. 3. Sample results of face detection (dark rectangle) and mouth region finding (light half-ellipse)

5. Cursor Movements

Mouth region localization, compared with a reference mouth position, is used to control the screen cursor. Generally, the greater the shift is, the faster the cursor moves in a given direction. The reference mouth position is saved at the application startup and may be altered at any time on the user request.

Translation between the current mouth position and the screen cursor movements is determined by three parameters: threshold t, sensitivity s and acceleration a, and is given with the equation (in vertical and horizontal direction separately):

$$\Delta x = \begin{cases} a(\Delta p - t)^2 + s(\Delta p - t), & \Delta \mathrm{p} > t \\ -a(\Delta p + t)^2 + s(\Delta p + t), & \Delta p < -t \\ 0, & -t \le \Delta p \le t \end{cases} \tag{1}$$

where Δx denotes the resulting screen cursor movement distance (in screen pixels) and Δp is the distance between the mouth position and the reference position. Threshold, sensitivity and acceleration have positive float values and are defined for horizontal and vertical direction, independently.

The mouth position shift Δp (in horizontal and vertical direction), is calculated as follows:

$$\Delta p_x = \frac{m_x - r_x}{w} \qquad \Delta p_y = \frac{m_y - r_y}{w} \tag{2}$$

where (m_x, m_y) denotes the current mouth position (the center of the mouth region upper boundary) in video frame pixels, (r_x, r_y) is the reference mouth position and w denotes the current mouth region width. Normalization of the mouth position shift by the mouth width assures that a screen cursor moves in the same way independently of the user face distance from the camera.

Threshold t is the minimal mouth shift from the reference position that is required for the screen cursor to start moving. The greater the threshold is the more the user head needs to be turned in order to move the screen cursor. Sensitivity s and acceleration a determine directly how the mouth shift value is translated into screen cursor movement speed. The greater values of these parameters are, the faster the screen cursor moves at the same mouth shift from the reference position.

6. Lip Gesture Recognition

Lip gesture recognition is performed by an artificial neural network (ANN). A feature vector for the ANN contains parameters describing image region containing lips only.

6.1. *Artificial Neural Network Description*

A feed-forward ANN with one hidden layer is used to detect lip gestures. Each image frame is classified independently. The number of ANN inputs corresponds with the number of lip image features and is equal 168 or 171, depending on the chosen variant of lip region extracting (section 6.4). There are 4 outputs from ANN, each one is related with one type of gestures recognized by ANN. Three of them are: opening the mouth, sticking out the tongue and forming puckered lips. A natural, neutral facial expression is the fourth gesture and means that no real lip gesture is present. Based on initial experiments, number of neurons in the hidden layer was set to eight. It is the minimum number of neurons sufficient for good effectiveness of lip gesture recognition. Sigmoid activation functions are used in all neurons.

A type of the gesture is determined by the maximum value of the ANN outputs. However, in the HCI application it is crucial to minimize false-positive rate of detection of all three, real gestures in order to prevent execution of actions not meant by a user. False-negative rate is less important – if a gesture is not recognized in some video frame, it will be recognized in succeeding frames when a user moves his head a little or change face expression.

In order to minimize number of false-positives, post processing of ANN output vector o is performed in order to determine reliability of classification. The maximum value of ANN output o_{max} is converted according to the equation:

$$o'_{\max} = \frac{o_{\max}}{\sum_{i=0}^{3} o_i} \cdot o_{\max}, \quad o_i \in [0,1] \tag{3}$$

If $o'_{\max}$ is greater or equal to the threshold T, a gesture connected with the output o_{max} is returned as the recognized gesture; otherwise, the neutral gesture is returned which means that no real gesture is detected. This method assures that if the neural network output is not firm, no gesture is detected in order to minimize false-positives ratio. It can be noticed that $T = 0$ turns off ANN output post-processing.

ANN is trained with a resilient backpropagation algorithm (RPROP) [Riedmiller and Braun (1993)]. Training data are acquired during calibration phase which is required at the beginning of every session with the application. The calibration consists of 4 stages. During each stage, a user is asked to move his head left, right, up and down while making one of the four gestures: neutral one in the stage 1, mouth opening in the stage 2, sticking out the tongue in the stage 3 and forming puckered lips in the stage 4. Each stage lasts 4 second, with 2 second break between stages when a user is asked to change a lip gesture. During each stage, 60 frames containing gesture images are gathered (video rate is 15 fps). Feature vectors obtained from these frames form training vectors (80% of all vectors) and validation vectors (every fifth vector). This means that total 192 feature vectors (48 for every gesture) are used for ANN training and 48 vectors are used to validate ANN after training (12 for every gesture). Five neural networks are trained based on the same data and the one with the smallest error rate of validation vector classification (with post-processing threshold $T = 0.5$) is used for lip gesture recognition.

6.2. *Lip Localization*

In order to facilitate lip gesture recognition by ANN, an algorithm for determining region of the image containing lips only must be very precise and has to be robust against head movements in the vertical and horizontal directions. In order to locate lips, a series of face image transformations is performed. First, an image is smoothed with Gaussian filter and converted from RGB color space to the CIE LUV space [Leung *et al.* (2004)]. Furthermore, the smoothed image is also transformed with DHT (Discrete Hartley Transform) [Moran *et al.* (2007)], according to the formula:

$$\begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix} = \begin{bmatrix} 0.5774 & 0.5774 & 0.5774 \\ 0.5774 & 0.2113 & -0.7887 \\ 0.5774 & -0.7887 & 0.2113 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{4}$$

The U component of LUV space and the third component C_3 of DHT transform are used for further processing, because they provide distinct, linear separation of lip and non-lip areas. These two components are multiplied in order to increase the differences (Fig. 4a).

Although simple, binary thresholding of the resulting image in order to obtain a lip region is now possible, its results could be faulty depending on lighting conditions. Therefore additional processing steps are performed. It is known, that user lips are always located near in center of the analyzed image region; they never touches boundaries of the region. For that reason, values of pixels are attenuated depending on their distances from

the mouth region center; the farther the pixel is from the center the more its value is decreased (Fig. 4b). This operation reduces the noise at the image border.

In the next step the image is saturated with the hyperbolic tangent curve according to the formula:

$$y = \frac{\tanh(c_s \cdot (x - t_s))}{2} + 0.5 \tag{5}$$

where x and y are input and output pixel values, saturation coefficient c_s is equal 5 and saturation threshold t_s is equal 0.5. This operation increases the difference between lip and non-lip regions by increasing high pixel values and decreasing small ones (Fig. 4c).

In the last step morphological closing and opening of the image is performed (Fig. 4d). Now the image is thresholded with the constant threshold located in the middle of pixel value range and resulting binary image is morphologically closed in order to connect all blobs lying close to each other.

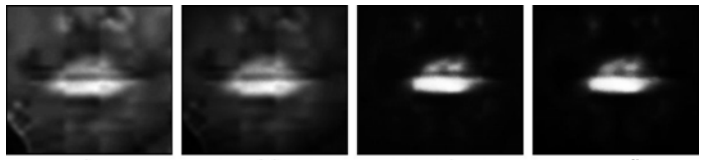

Fig. 4. Sample results of a) multiplication of U and C_3 components, b) attenuating values of pixels lying farther from the image center, c) saturating with hyperbolic tangent curve and d) morphological closing and opening

6.3. *Lip Shape Approximation*

In the result of the lip localization stage, one or more blobs placed on lips are obtained. Smaller blobs are removed. Based on horizontal axes of nose trills and a mouth (found by analyzing first derivate of a vector containing average values of every row in the mouth region), one blob lying both on upper and lower lips (or on a lower lip only) is chosen. This blob is used as a starting condition for an iterative process that approximates lip shape with an ellipse using fuzzy clustering [Leung *et al.* (2004)]. In the method, a dissimilarity measure that integrates the color dissimilarity and the spatial distance in terms of an elliptic shape function is used. Because of the presence of the elliptic shape function, the measure is able to differentiate pixels having similar color information but located in different regions.

Lip shape approximation algorithm requires providing three input parameters: mean colors of lip and non-lip regions and an initial localization of the ellipse. Pixel color is described with three channels containing U and V components of LUV color space and C_3 component of the DHT transform. In the first analyzed video frame, the mean color of the lip region v_0 is calculated as an average color of the starting blob interior and the mean color of the non-lip region (skin) v_1 is based on the color in the upper part of the mouth region (Fig. 3). In the following video frames, initial mean colors of lip and non-lip regions are based on the results of the analysis in the previous frame.

An ellipse approximating lip shape is denoted as $p = \{x_c, y_c, w, h, \theta\}$, where ($x_c$, y_c) is the ellipse center, its semi-axes are denoted with w and h, and θ is the inclination angle. Initial localization of the ellipse p is based on the starting blob position and size. The blob usually covers whole lips or is placed on the lower lip only. Therefore, the ellipse center is placed in the middle of the upper border of the starting blob bounding box. Its size is equal the size of the bounding box. The ellipse is oriented horizontally ($\theta = 0$).

A dissimilarity measure $D_{i,r,s}$ between the pixel (r, s) and the cluster i (lip or non-lip region) is given with the equation [Leung *et al.* (2004)]:

$$D_{i,r,s} = d_{i,r,s}^2 + \alpha \cdot f(i, r, s, p) \tag{6}$$

where $d_{i,r,s}$ denotes the Euclidian distance between the color of the pixel (r, s) and the mean color (cluster center) v_i of the ith cluster, and $f(i, r, s, p)$ is related to the spatial distance between the pixel and the ellipse p:

$$f(i, r, s, p) = \left\{ \frac{\left((r - x_c)\cos\theta + (s - y_c)\sin\theta\right)^2}{w^2} + \frac{\left((s - y_c)\cos\theta - (r - x_c)\sin\theta\right)^2}{h^2} \right\}^{p_i} \tag{7}$$

The parameter α adjusts the weight of the spatial distance compared to the distance in the color space. The exponent p_i ensures small f function values for pixels inside the cluster i and high values for pixels outside the cluster. Initially, $f(i, r, s, p) = 0$.

A membership $u_{i,r,s}$ of the pixel (r,s) in the cluster i is given with the equation [Leung *et al.* (2004)]:

$$u_{i,r,s} = \left[\sum_{j=0}^{1} \left(\frac{D_{i,r,s}}{D_{j,r,s}} \right)^{\frac{1}{m-1}} \right]^{-1}, \qquad \sum_{i=0}^{1} u_{i,r,s} = 1 \tag{8}$$

where $m \in (1, \infty)$ defines fuzziness of clustering.

During each iteration of the lip shape approximation algorithm, membership functions for every cluster are calculated according to (8). Then, the cluster centers v_i are updated as follows [Leung *et al.* (2004)]:

$$v_i = \frac{\sum_r \sum_s u_{i,r,s}^m \cdot x_{r,s}}{\sum_r \sum_s u_{i,r,s}^m} \tag{9}$$

where $x_{r,s}$ denotes the color of pixel (r,s).

In the next step, the ellipse p is replaced by an ellipse that is best-fitted to the u_0 function denoting membership of mouth region pixels to the lip cluster. Finally, spatial distances between pixels and the ellipse p are updated according to (7). The algorithm ends after I iterations. The ellipse p is the output of the algorithm. Fig. 5 presents results of lip shape approximation with an ellipse after iteration no. 1, 2, 3, 4, 6 and 10.

There are five parameters of the method that need to be set in order to achieve good results of lip shape approximation for every user in variant lighting conditions: m, p_0, p_1, α, and I. In order to facilitate this task, 14 presets of these parameters were created. A

user needs to choose one preset that provides the best-fitting, stable ellipse, i.e. the ellipse contour always overlaps lip edges, regardless of head movements and gestures made with the lips. In all experiments presented in the paper it is assumed that the best preset has been chosen for every person. Fig. 6 shows optimal results of lip shape approximation with an ellipse for all types of gestures recognized by ANN.

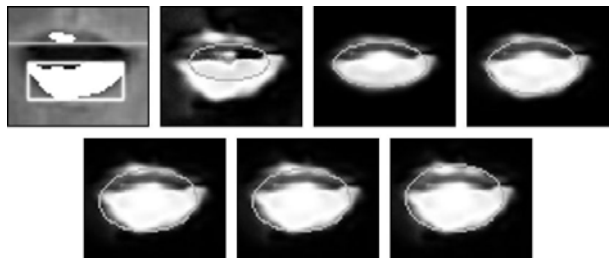

Fig. 5. Illustration of the algorithm for lip shape approximation; top left – mouth region with detected blobs placed on lips (white) and the blob chosen as a starting point for the algorithm (light rectangle); remaining images – membership function u_0 values and resulting ellipse p after iteration no. 1, 2, 3, 4, 6 and 10

6.4. *Lip Region Extraction*

Three different variants of lip region extracting are available. In the first variant (V_1), the lip region is based directly on the ellipse approximating the lip shape and is constituted by the rectangle containing the ellipse. It means that the lip region size is not constant and the region moves and tilts according to the results of lip shape approximation. In the second variant (V_2), horizontal, constant-size square is used as the lip region. Its center is always anchored at the center of the ellipse and the length of its sides is fixed and determined at the beginning of the calibration process by the width of the ellipse. In the third variant (V_3), influence of the ellipse on the lip region extracting is minimal. The lip region is formed by the square which size and position is fixed and determined at the beginning of the calibration phase. The center of the square is located at the center of the ellipse, and the length of sides is equal to the half of the width of the whole mouth region. The third variant is especially useful when the algorithm of lip shape approximation fails.

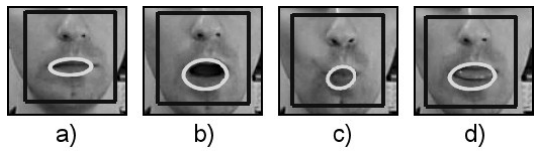

Fig. 6. Sample results of mouth region (dark rectangle) and lip shape detection (light ellipse) for the lack of the lip gesture (a) and for three gestures recognized: opening the mouth (b), forming puckered lips (c) and sticking out the tongue (d)

6.5. *Lip Image Features*

There are 171 lip image features used for lip gesture recognition. They can be divided into 4 groups. The first one is used only when the first variant V_1 of lip region extracting is chosen and it contains three parameters: the width and the height of the ellipse approximating the lip shape and the angular eccentricity of the ellipse which is given with the equation:

$$e = \arccos\left(\frac{a}{b}\right) \tag{10}$$

where a is the shorter, and b – the longer axis of the ellipse.

The second group of parameters is formed by the normalized, 20-point luminance histogram of the lip region.

The third group contains Hu sets of invariant image moments [Flusser (2000)][Flusser and Suk (2006)]. The moments H are invariant under translation, changes in scale and rotation and are given with equations:

$$H_1 = \eta_{20} + \eta_{02} \tag{11}$$

$$H_2 = (\eta_{20} - \eta_{02})^2 + (2\eta_{11})^2 \tag{12}$$

$$H_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \tag{13}$$

$$H_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \tag{14}$$

$$\begin{aligned} H_5 = {} & (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\ & + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \end{aligned} \tag{15}$$

$$H_6 = (\eta_{20} - \eta_{02})\left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \tag{16}$$

$$\begin{aligned} H_7 = {} & (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\ & - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \end{aligned} \tag{17}$$

where η_{xy} denotes a central moment μ_{xy} divided by the scaled, zeroth moment according to the formula:

$$\eta_{xy} = \frac{\mu_{xy}}{\mu_{00}^{1+\frac{x+y}{2}}}, \quad x + y \geq 2 \tag{18}$$

Four sets of Hu moments are calculated based on four equal-sized, non overlapping luminance images the lip region is divided into. Each Hu set contains 7 parameters which gives total number of 28 features in the third group.

The last group of parameters is based on co-occurrence matrices. A co-occurrence matrix, also referred to as a co-occurrence distribution, is defined over an image to be the distribution of co-occurring values at a given offset [Haralick *et al.* (1973)][Clausi (2002)]. It is commonly used as a texture description. A co-occurrence $N \times N$ matrix C

defined over an $n \times m$, *N*-colour image *I*, parameterized by a spatial offset $(\Delta x, \Delta y)$, is given with an equation:

$$C(i,j) = \sum_{p=1}^{n} \sum_{q=1}^{m} \begin{cases} 1, & I(p,q) = i \text{ and } I(p+\Delta x, q+\Delta y) = j \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

A co-occurrence matrix *C* is converted to the matrix symmetrical around the diagonal C_s by adding its own transposition:

$$C_s = C + C^{\mathrm{T}} \tag{20}$$

A symmetrical co-occurrence matrix is identical for offsets symmetrical around (0,0), e.g. (-1,0) and (1,0) or (-1, -1) and (1, 1). In the last step a matrix is expressed as a probability by normalizing its elements according to the formula:

$$P(i,j) = \frac{C_s(i,j)}{\sum_{i=0,j=0}^{N-1} C_s(i,j)} \tag{21}$$

A set of symmetrical, normalized, co-occurrence matrices *P* is calculated for three lip image representations: the luminance *L* and chrominance *U* of the CIE LUV color space and the first vertical derivative of the luminance image calculated with a Sobel operator [Young *et al.* (1998)]. Each set contains eight matrices *P* calculated for four different directions of offsets; for 0° (0,1) and (0,2) offsets are used, for 45°: (1,1) and (2,2), for 90°: (1,0) and (2,0), for 135°: (1, -1) and (2,-2). An input image color depth is quantized into 25 equally-spaced levels, therefore each co-occurrence matrix contains 25×25 elements.

Five statistical parameters are calculated for every co-occurrence matrix *P*. They are as follows:

$$\text{contrast} = \sum_{i,j} P_{i,j} (i-j)^2 \tag{22}$$

$$\text{energy} = \sqrt{\sum_{i,j} P_{i,j}^{\ 2}} \tag{23}$$

$$\text{mean} = \mu_i = \mu_j = \sum_{i,j} i \cdot P_{i,j} \tag{24}$$

$$\text{standard deviation} = \sigma_i = \sigma_j = \sqrt{\sum_{i,j} (i-\mu_i)^2 P_{i,j}} \tag{25}$$

$$\text{correlation} = \sum_{i,j} P_{i,j} \frac{(i-\mu_i)(j-\mu_j)}{\sqrt{\sigma_i^{\ 2} \sigma_j^{\ 2}}} \tag{26}$$

This gives total number of 120 parameters (3 image representations × 8 co-occurrence matrices × 5 parameters) based on co-occurrence matrices in the ANN feature vector.

7. Experiments and results

For the purpose of experiments, face recordings of 176 persons were collected during two recording sessions. Videos from the first session (Fig. 7), recorded in various places and in different lighting conditions, were used to develop and validate face localization, lip localization and lip shape approximation algorithms. Experiments prove, that the mouth region is localized with great accuracy. Controlling screen cursor with the mouth (head) movements is quiet convenient and does not pose any problems for anyone who uses the application for the first time.

Fig. 7. Sample frames from test recordings

102 video recordings from the second session were used for lip gesture recognition experiments. Each person was asked to carry out typical, calibration procedure twice. The first iteration was used to train ANN and the second iteration was used to obtain the effectiveness of lip gesture classification. All face images gathered during the second iteration were used for testing, therefore the testing set of vectors contained 25% more elements than the training set of vectors (20% of vectors gathered during the first iteration is used for instant ANN validation).

Tab. 1 presents summary of lip gestures recognition for different lip region extracting variants. It is seen that the best results were achieved for the third variant V_3 that relies the least on the results of lip shape approximation with an ellipse. This means that although the ellipse usually fits the real lip shape, its inter-frame variances might interfere with the ANN classification effectiveness. It was discovered that V_3 variant provides the best results for 82% of test recordings. Other variants were optimal (V_1 for 14% and V_2 for 4% of test recordings) for those recordings where the lip shape is approximated with an ellipse ideally in all movie frames. This allows to conclude that V_2 variant is not needed; in case of any problems with lip shape approximations V_3 wins, in other cases V_1 variant is usually sufficient.

Table 1. Summary results of lip gesture recognition for different lip region extracting variants.

Lip region extracting variant	Effectiveness of lip gesture classification				
	Neutral (no gesture)	Mouth opening	Forming puckered lips	Sticking out the tongue	All gestures
V1	86.1%	85.3%	85.4%	84.8%	85.4%
V2	80.2%	83.0%	75.0%	78.8%	79.3%
V3	91.3%	95.3%	92.0%	94.1%	93.2%

Detailed results of lip gesture classification are shown in Tab. 2. It is seen that increasing ANN post-processing threshold T improves effectiveness of neutral gesture recognition and worsens results of other three gestures classification. It is assumed that an optimal value of the T threshold is 0.5 which provides compromise between the effectiveness and the false-positive ratio of real gesture recognition.

Table 2. Results of lip gesture classification (optimal variant of lip gesture extraction is used for every test recording)

Gesture	No. of image frames	Effectiveness of lip gesture classification for different ANN post-processing thresholds			
		T = 0	T = 0.25	T = 0.5	T = 0.75
Neutral (no gesture)	6120	92.9%	93.8%	94.9%	96.1%
Mouth opening	6120	95.4%	94.8%	92.4%	89.2%
Forming puckered lips	6120	92.5%	91.8%	88.2%	83.6%
Sticking out the tongue	6120	94.1%	93.2%	91.3%	85.6%
All gestures	24480	93.7%	93.4%	91.7%	88.6%

Achieved results of lip gesture classification are satisfactory. Total effectiveness of recognition over 90% means that on average three recognition errors appear every two seconds of algorithm working. Furthermore, due to ANN output post-processing, the majority of the errors emerge when the neutral gesture is recognized instead of other three gestures. These errors do not pose much inconvenience to a user and may be attenuated further by the means of simple time-averaging of lip gesture detection results.

8. Conclusions

An algorithm for lip movement tracking and lip gesture recognition is presented in the paper. It forms the core of the multimodal human-computer interface (HCI) called LipMouse. Results of the experiments carried out show that the effectiveness of the algorithm is sufficient for comfortable and efficient usage of a computer by anyone who does not want or cannot use a traditional computer mouse.

Future work will focus on improvement and further development of the interface and its algorithms. Most of all, the search for optimal composition of the feature vector will continue. New parameters will be defined and their extraction method will be tuned. Another research thread will be focused on increasing the number of recognized gestures.

Development of new HCI solutions and improving existing ones is necessary to facilitate our everyday interactions with computers.

Acknowledgments

Research funded within the project No. POIG.01.03.01-22-017/08, entitled “Elaboration of a series of multimodal interfaces and their implementation to educational, medical, security and industrial applications”. The project is subsidized by the European regional development fund and by the Polish State budget.

References

Aggarwal J. K.; Cai Q. (1999): Human Motion Analysis: A Review, CVIU(73), No. 3, pp. 428-440.

Baecker R. M.; Grudin J.; Buxton W. A. S.; Greenberg S. (Eds.) (1995). *Readings in human–computer interaction. Toward the Year 2000*, 2nd edn. Morgan Kaufmann, San Francisco.

Clausi D. A. (2002): An analysis of co-occurrence texture statistics as a function of grey-level quantization, Canadian Journal of Remote Sensing, 28(1), pp. 45-62.

de Dios J.J.; Garcia, N. (2004): Fast face segmentation in component color space, Int. Conf. on Image Processing, ICIP, 1, pp. 191-194.

Dalka P.; Czyzewski A. (2009): Lip movement and gesture recognition for a multimodal human-computer interface, Proc. of the Int. Multiconf. on Computer Science and Information Technology, pp. 451-455.

Duchowski A. T. (2002): A Breadth-First Survey of Eye Tracking Applications, Behavior Research Methods, Instruments, & Computers (BRMIC), 34(4), pp.455-470.

Eveno N.; Caplier A.; Coulon P. Y. (2001): New color transformation for lips segmentation, IEEE 4th Workshop on Multimedia Signal Processing, pp. 3-8.

Flusser J. (2000): On the Independence of Rotation Moment Invariants, Pattern Recognition, 33, pp. 1405–1410.

Flusser J; Suk T. (2006): Rotation Moment Invariants for Recognition of Symmetric Objects, IEEE Trans. Image Proc., 15, pp. 3784–3790.

Guan Y. P. (2008): Automatic extraction of lips based on multi-scale wavelet edge detection, IET Comput. Vis., 2(1), pp. 23–33.

Haralick R. M.; Shanmugam K.; Dinstein I. (1973): Textural Features for Image Classification, IEEE Transactions on Systems, Man, and Cybernetics, 3(6), pp. 610–621.

Leung S.; Wang S.; Lau W. (2004): Lip image segmentation using fuzzy clustering incorporating an elliptic shape function, IEEE Transactions on Image Processing, 13(1), pp. 51-62.

Lienhart R.; Maydt J. (2002): An Extended Set of Haar-like Features for Rapid Object Detection, IEEE ICIP, Vol. 1, pp. 900-903..

Liewa W. C.; Leung S. H.; Lauw. H. (2000): Lip contour extraction using a deformable model, Proc. Int. Conf. Image Processing, Vancouver, Canada, 2, pp. 255–258.

Moran L. E. L.; Pinto R.E. (2007): Automatic Extraction Of The Lips Shape Via Statistical Lips Modelling and Chromatic Feature, Electronics, Robotics and Automotive Mechanics Conference, CERMA, pp. 241-246.

Riedmiller M.; Braun H. (1993): A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm, Proc. ICNN.

Sasaki Y.; Kawamura T.; Sugahara, K. (2004): Lip shape extraction for word recognition by using hardware active contour model, Proc. of Int. Symp. on Intell. Multimedia, Video and Speech Processing, pp. 370-373.

Sears A.; Jacko J. A. (Eds.) (2007). *Handbook for Human Computer Interaction*, 2nd edn. CRC Press.

Shin G.; Chun J. (2007): Vision-based Multimodal Human Computer Interface based on Parallel Tracking of Eye and Hand Motion, Int. Conf. on Convergence Information Technology, p. 2443-2448..

Tsapatsoulis N.; Avrithis Y.; Kollias S. (2000): Efficient Face Detection for Multimedia Applications, ICIPOO, TA07.11, Vancouver, Canada.

Viola P.; Jones M. (2001): Rapid Object Detection using a Boosted Cascade of Simple Features, IEEE CVPR.

Young I.; Gerbrands J.; Vliet L. (1998). *Fundamentals of Image Processing*. Delft University of Technology.

Zhang X,; Mersereau R. M. (2000): Lip Feature extraction Towards an Automatic Speechreading System, ICIPOO, WA07.05, Vancouver, Canada.