PAPER

Detection of Tongue Protrusion Gestures from Video

Luis Ricardo SAPAICO[†a)], ***Student Member***, **Hamid LAGA**[††], ***and*** **Masayuki NAKAJIMA**[†], ***Members***

SUMMARY We propose a system that, using video information, segments the mouth region from a face image and then detects the protrusion of the tongue from inside the oral cavity. Initially, under the assumption that the mouth is closed, we detect both mouth corners. We use a set of specifically oriented Gabor filters for enhancing horizontal features corresponding to the shadow existing between the upper and lower lips. After applying the Hough line detector, the extremes of the line that was found are regarded as the mouth corners. Detection rate for mouth corner localization is 85.33%. These points are then input to a mouth appearance model which fits a mouth contour to the image. By segmenting its bounding box we obtain a mouth template. Next, considering the symmetric nature of the mouth, we divide the template into right and left halves. Thus, our system makes use of three templates. We track the mouth in the following frames using normalized correlation for mouth template matching. Changes happening in the mouth region are directly described by the correlation value, i.e., the appearance of the tongue in the surface of the mouth will cause a decrease in the correlation coefficient through time. These coefficients are used for detecting the tongue protrusion. The right and left tongue protrusion positions will be detected by analyzing similarity changes between the right and left half-mouth templates and the currently tracked ones. Detection rates under the default parameters of our system are 90.20% for the tongue protrusion regardless of the position, and 84.78% for the right and left tongue protrusion positions. Our results demonstrate the feasibility of real-time tongue protrusion detection in vision-based systems and motivates further investigating the usage of this new modality in human-computer communication.

key words: *face gestures, mouth segmentation, perceptual user interface, tongue protrusion, vision-based systems*

1. Introduction

Perceptual User Interfaces [1] aim at establishing a natural communication between humans and machines, so that it resembles the way we interact with other people. One of the simplest forms of communicating is by producing face gestures. They have been extensively used previously for human-computer communication. While speech is probably the most natural means of sending a message, other non-standard possibilities include using eye blinks or eyebrow movements [2], head nodding and shaking [3]; or nose movements [4]. Conveying emotions is also a very important part of communication, for which detecting changes in certain points in the face becomes essential. In such situations, the mouth region plays an essential role. Accurate mouth segmentation has been a challenge given its highly deformable characteristic.

Controlling the tongue organ is especially suited for vision-based approaches because:

1. It is universal: not only is the tongue a highly controllable muscle, but its controllability remains intact even after major body injuries.
2. It is usable: the tongue is naturally located inside the mouth cavity, and its appearance on the face surface can be interpreted as the user's intention for conveying a message.
3. It is useful: tongue gestures can complement other face gestures for a richer communication.

Furthermore, the tongue protrusion gesture may be thought of as one of the most simple means for facial communication that can be carried out. Indeed, even neonates are able to imitate this oral movement just by observation [5].

In this paper we propose a vision-based system for detecting the tongue protrusion gestures illustrated in Fig. 1. Two tongue events are to be detected: tongue protrusion (1) to the right side, and (2) to the left side of the mouth. The only constraint is that the mouth should remain closed during the tongue appearance. The system requires an off-the-shelf webcamera and detects these gestures in real-time using video information.

Our method operates in three stages: (1) segmentation of the mouth region, in which a closed mouth region template is automatically obtained from the first image frame that is captured; (2) tracking of the mouth template in the subsequent frames, and (3) detection of the protrusion of the tongue by interpreting the changes in the mouth template through time.

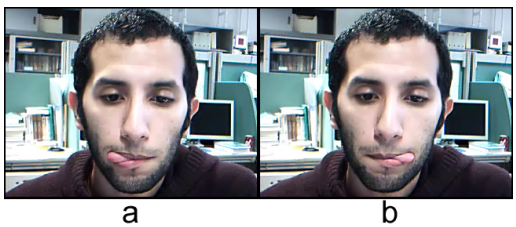

Fig. 1 Tongue gestures to detect: (a) Right. (b) Left.

Manuscript received August 1, 2010.
Manuscript revised February 19, 2011.

[†]The authors are with the Graduate School of Information Science and Engineering, Tokyo Institute of Technology, Tokyo, 152-8552, Japan.

[††]The author is with the Institut Telecom, Telecom Lille1, LIFL UMR8022, France.

a) E-mail: rsapaicov@yahoo.co.jp

DOI: 10.1587/transinf.E94.D.1671

Particularly, the main contributions of this paper are: (1) we propose a method that finds the mouth corners, based on intrinsic characteristics of the mouth; (2) we propose an algorithm for detecting a tongue protrusion gesture from video; and (3) we extend the previous framework in order to locate the position of the tongue that was protruded from the mouth. The positions we detect are right and left. Additionally, we perform a thorough analysis of the parameters required for the detection and we demonstrate the feasibility of utilizing the tongue for vision-based systems.

The remainder of the paper is organized as follows: Section 2 discusses existing related work; the mouth region segmentation algorithm is described in Sect. 3; Sect. 4 contains details regarding the mouth tracking and the detection of the tongue protrusion position. Results and a discussion on the system settings are presented in Sect. 5. We conclude and outline directions for future work in Sect. 6.

2. Related Work

Our system is divided into two steps: mouth region segmentation, and tongue protrusion detection. Consequently, we will briefly review the most relevant literature separately for each process.

2.1 Mouth Region Segmentation

Segmenting the mouth region from face-background images has been found useful in several applications, e.g., audio-visual speech recognition or affective technologies; various different approaches have been proposed. We expect the user of our system to have the mouth closed when it is being segmented. Therefore, the most representative work on closed mouth segmentation can be classified into three categories:

1. **Model-based [6], [7]:** Training using a Database with manually selected feature points such as eyes, nose and mouth is necessary for the system to learn a set of nodes that represents the object of interest. Using shape and texture information makes the system highly adaptive to new images. However; for each unseen image this method requires a good set of initialization points, since its search converges locally to the best match.
2. **Color-based [8], [9]:** Color information from training images are used for building Skin or Hair Likelihood Maps that, in turn, will segment the image into skin and non-skin organs. The mouth is found considering that it belongs to the latter category; or by discriminating pixels based on their chromatic color information. This technique is successful under controlled environments. However, it is sensitive not only to color changes due to illumination, but also to skin color variations among different races. In particular, cases with low contrast between lips and skin surrounding them tend to fail.
3. **Feature-based [10], [11]:** It is spread to use the Integral Projection method, which builds a histogram by scanning each pixel row and finding the total sum of gray values. Naturally, the face orientation should be known in advance. Next, peaks that are assumed to correspond to distinctive face features such as eyebrows, eyes and mouth, are found. The downside is that finding sharp and distinctive histogram peaks may not be possible. Thereby, making it difficult to achieve a robust performance due to the low accuracy of feature selection.

In summary, if a mouth model is constructed with sufficient training images, and good initialization points are passed on; the model-based approach can outperform other methods. Consequently, because the first stage of our system detects accurately the mouth corners, these fiducial points are used as reference. The initial mouth model is, hence, resized according to the corner's distance, and the corners are also fixed as starting points for the fitting process.

2.2 Vision-based Tongue Protrusion Detection

The literature corresponding to the detection of the tongue protrusion in vision-based systems is very limited. To the best of our knowledge, the first attempt was the work of Sapaico et al. [12], which used a cascade of three Support Vector Machines classifiers for detecting the tongue protrusion in three positions: left, right and middle. Results for the left/right detection (last stage of the cascade) showed an overall efficiency of approximately 70%. Additionally, the appearance of the tongue in the center was in many cases neglected, given its perceptual similarity in a gray-scale image to a closed mouth, i.e., using the tongue protrusion in the center hindered the detection effectiveness. Lastly, due to the lack of sufficient training data the model cannot achieve reliable classification.

A video-based tongue switch for a child with severe cerebral palsy was developed in [13]; which allowed to detect the tongue gestures from different view points. The mouth was segmented using the method proposed in [8]; and the tongue protrusion was detected by analyzing the change in saturation of red color in the mouth region. They reported a 82% success rate for the tongue gestures. Their system was only evaluated with a seven year-old boy, and it is not able to detect the position of the tongue protrusion. Additionally, similar to color-based mouth segmentation methods, using "redness" for the detection may not generalize well to other participants, since it may not be trivial to distinguish the tongue from the lips just by using that information. We note that this method does not detect the position of the tongue protrusion.

A system called LipMouse was presented in [14]. It detects three mouth gestures: (1) opened mouth, (2) sticking out the tongue from the mouth (tongue protrusion), and, (3) forming puckered lips. They extracted a set of feature vectors from the image after it is transformed to the CIE LUV color space, and they trained an Artificial Neural Net-

work for each user during an initial calibration procedure. In particular, their classification results of the tongue "sticking-out" from the mouth showed a success rate of 91.3%. Unfortunately, there is no detailed information available regarding the database, nor the exact position or duration of the tongue protrusion gestures, which appears to be fixed to the center of the mouth. Therefore, in this system the position of the tongue is not detected either.

In the previous systems, the detection is done at each incoming frame, i.e., although the input signal is a video, no time-related information is considered and each image is treated separately. In contrast, in this paper we argue that better results can be obtained by using time-varying perceptual information from the tongue gestures. Moreover, our method relies solely on intensity images, which requires less processing time. Finally, in our approach not only we detect the tongue protrusion, but its left/right position is also located. This makes the system more robust compared to [12], since these events are clearer to perceive; and richer in comparison to [13], [14], which only detect the tongue protrusion in the center.

3. Mouth Region Segmentation and Tracking

In order to detect the tongue protrusion, we first need to locate the mouth region. Several techniques have been proposed; however, they were difficult to generalize and often worked only under the conditions they were tested. In our system, we assume the following context for the segmentation:

1. There is only one user who accesses the system.
2. The user is located in front of the computer, in an upright position.
3. During the initial segmentation process, the user is asked to maintain the mouth closed.

Therefore, we propose to use an approach that combines feature, knowledge and model-based methods, which is robust to changes in environment and users. It searches for a unique characteristic of the human mouth: the shadow line that appears between the upper and lower lip, under the only constraint that the mouth should be closed. Extremities of this line correspond to the mouth corners, used as reliable initialization points for a model-based mouth contour search. The framework, described in Fig. 2, is explained next.

3.1 Approximate Segmentation of the Mouth

Our method uses the Haar feature-based face detection algorithm [15] for extracting the face of the user from the rest of the scene. As we have assumed that there is only one person situated in front of the computer, the criterion is to take as the input face the largest of all potential faces.

After obtaining the face region, we need to further segment the image in order to minimize the search region for the mouth. Locating directly an approximate mouth region

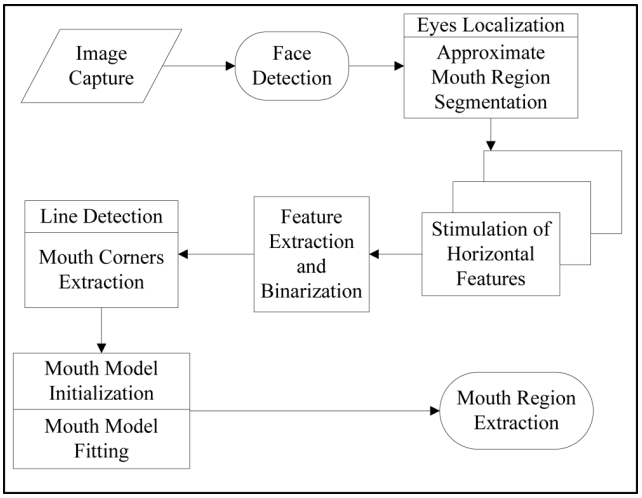

Fig. 2 Mouth region segmentation framework.

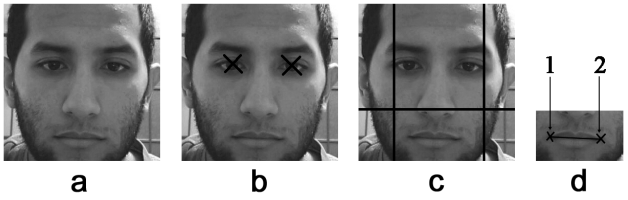

Fig. 3 Initial segmentation. (a) Input image. (b) Eyes detection. (c) Reference lines. (d) Segmented mouth region.

by using anthropometric measurements [16] is not reliable when a face shape is an *outlier* of the previous statistical descriptor. On the other hand, using a model-based technique for searching a region where *important* mouth features such as the corners would be located requires a precise set of initialization points, as explained in Sect. 2.1. We adopt a simple hypothesis: that given a frontal face image, the mouth should be located in the lower third part of that face. This gives us the height of the approximate mouth region.

For obtaining the width of the approximate mouth region, we locate the eyes since their distance provides an estimate of the width of the mouth. Within the face, eyes are much easier to detect among humans because of their symmetry and small variance in both shape and structure. We use the eye detection output proposed in [17]. In case the eye location is not accurate, or that the mouth is larger than the distance between the eyes, we widen the distance between the eyes by half. Finally, we trace two parallel lines for limiting the width of our search region for the mouth. Figure 3 illustrates the complete segmentation process. By using these knowledge-based criteria, we are able to segment a region which contained the mouth for each image in our face database, including cases where the eye detection was not precise.

3.2 Mouth Corners Extraction

The previous process has reduced the mouth search area.

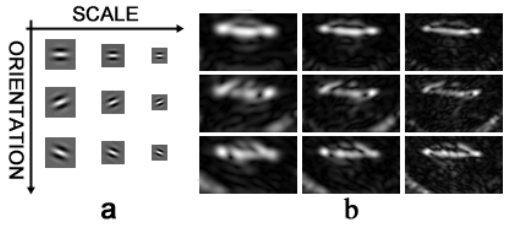

Fig. 4 (a) Set of filters used for feature enhancing.(b) Magnitude values obtained after applying (a) to an image.

Next, we need to detect accurately the position of the mouth corners. We will use a feature extraction method as follows.

3.2.1 Gabor Filters Processing

We will utilize of a family of Gabor filters that are invariant to uniform illumination changes, as proposed in [18]:

$$\psi_j(\vec{x}) = \frac{k_j^2}{\sigma^2}\exp(-\frac{k_j^2 x^2}{2\sigma^2})[\exp(i\vec{k_j}\vec{x}) - \exp(-\frac{\sigma^2}{2})] \qquad (1)$$

where:

$$\vec{k_j} = \begin{pmatrix} k_{jx} \\ k_{jy} \end{pmatrix} = \begin{pmatrix} k_v \cos\varphi_\mu \\ k_v \sin\varphi_\mu \end{pmatrix}, k_\nu = 2^{(-\frac{\nu+2}{2})\pi}, \varphi_\mu = \mu\frac{\pi}{8}$$

Parameters μ and ν from Eq. 1 define the orientation and scale of the Gabor kernel. The width of the Gaussian window is controlled by σ.

We use an orientation parameter μ such as the filters enhance only horizontal features. Doing so, we can find the line that lays between both lips, and connects the leftmost and rightmost corners of the mouth. This line is illustrated between points 1 and 2 in Fig. 3(d). Additionally, our images are captured from a low-resolution webcamera, thus we do not need to find features for relatively high scales. This limits the range of values for the parameter ν. Consequently, we consider parameters $\mu \in$ [3,4,5] corresponding to $3\pi/8$, $\pi/2$ and $5\pi/8$ orientations; and $\nu \in$ [0,1,2] corresponding to $1/2$, $1/2\sqrt{2}$ and $1/4$ scales. We obtain the set of Gabor filters shown in Fig. 4(a). They are applied to an image segmented according to the procedure described in Sect. 3.1. Magnitudes of the corresponding filter results are shown in Fig. 4(b).

3.2.2 Detecting the line connecting the mouth corners

We have obtained nine different outputs for one single image. We need to find the *important*† feature points. We localize them by using the method in [19], which automatically finds peaks from a given gray-scale image.

In this process, a pixel centered at (x_o, y_o) inside a window W_o of size $(W \times W)$ is selected as a feature point if: (1) its value is a local maximum, and (2) its value is also greater than the average value of the image. Consequently, we obtain a set of nine binary images whose "one" pixels are

†in this context, *important* means having high values.

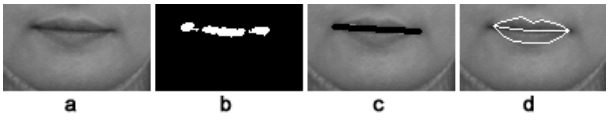

Fig. 5 (a) Input image from Sect. 3.1. (b) Binary image after feature extraction using images from Fig. 4. (c) Line detection result. (d) Model fitting result.

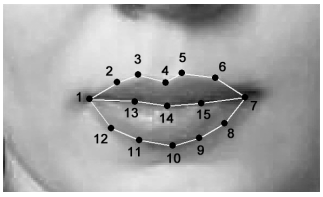

Fig. 6 The 15-point model used for training the AAM.

global maximum values. We found empirically that using a window of size $W = 7$ gives the best binarization results. Finally, we merge the set of images by applying the logical AND operator to the nine images, and we obtain a final image in which the most *important* horizontal features are present. A sample binary image is shown in Fig. 5(b).

To detect a line whose extremities correspond approximately to the corners of the mouth, we apply the Progressive Probabilistic Hough Transform [20] to our feature-point image because of its suitability for real-time applications. As a result, we find the longest horizontal line present in the binary image. Figure 5(c) shows some line detection results.

3.3 Model Fitting

Active Appearance Models (AAM) is a technique that fits an average shape and an average texture model to an image by tuning some parameters. We have built a training database of closed mouth shape samples under office-alike environments. We use a 15-point mouth model, shown in Fig. 6.

The fitting process requires an initial location for the points, and searches locally for the best match; i.e., the initialization is critical for the final result. We use the line obtained in Sect. 3.2.2 for the model initialization so that: (1) the model is rescaled according to the length of that line, and (2) initial points for the mouth corners (points **1** and **7** in Fig. 6) correspond to the extremities of that line.

Finally, the bounding box containing the fitted model, illustrated in Fig. 5(d), is segmented. This image will be used in the next section.

4. Mouth Tracking and Tongue Protrusion Detection

Running at each frame the mouth segmentation method described in the previous section is computationally demanding. Instead we use it as an initialization for detecting the mouth region in the first frame. In subsequent frames we propose to track the mouth using a template matching method based on correlation.

In particular, using the Normalized Correlation Coefficient (NCC) as a similarity measure for gray-scale images permits a fast processing time, and gives strength against uniform illumination changes. The NCC has been used before for tracking other face features such as eyes or eyebrows [21], [22]. It has been found that it is the most accurate algorithm for vision-based tasks [23].

We detect the tongue protrusion, and we locate its position, by performing an analysis of the change in the NCC value through time. Our method is perceptual in the sense that changes happening in the mouth are directly described by the NCC value. For example the lower the NCC gets the more the mouth region has changed. Since there are no other objects in the mouth surroundings capable of changing its appearance; these variations in the mouth region are attributed to the occlusion caused by the tongue appearance.

In our approach, one tongue protrusion gesture is started when the tongue appears on the surface on the mouth, and it is ended when the tongue returns to the oral cavity. Our system detects one gesture at a time, i.e., the user needs to put the tongue back to the mouth cavity before the system can detect another tongue protrusion gesture. Details regarding the tracking and detection process, shown in Fig. 7, are explained next.

4.1 Initialization of the Tracking

Using the method proposed in Sect. 3, we have segmented a mouth region given an upright frontal face. We need to initialize the templates for the matching. Before the templates are stored, the following preprocessing is executed.

In order to remove noise while preserving important edges, the image is processed using the Bilateral filter [24], followed by an unsharpening mask. Thus, we obtain and store the mouth template (**MT**).

It is a physical constraint that the tongue cannot appear at the same time in both left and right side of the mouth, i.e., it can protrude to either one of those two regions. Therefore, it is plausible that changes in the side where the tongue protrudes are greater than in the other. Consequently, we hypothesize that the position of the tongue protrusion can be detected by comparing the NCC values for both areas. For this reason, the **MT** is divided into halves, and its left half template (**LHT**) and right half template (**RHT**) are stored. These three templates will be utilized for the detection of the tongue protrusion position. They are shown in Fig. 8.

4.2 Detecting Tongue Protrusion Gestures

For each incoming frame, a NCC is obtained for each pixel inside a search region, and the largest NCC is found. This is equivalent to finding the best match for the **MT**. The pixel position of the best match is taken as the new location for the mouth. The search area for the new mouth region is defined using the current mouth position, and extending the search over an area twice as large. Thereby, the search area is updated at every frame, allowing smooth rotation movements

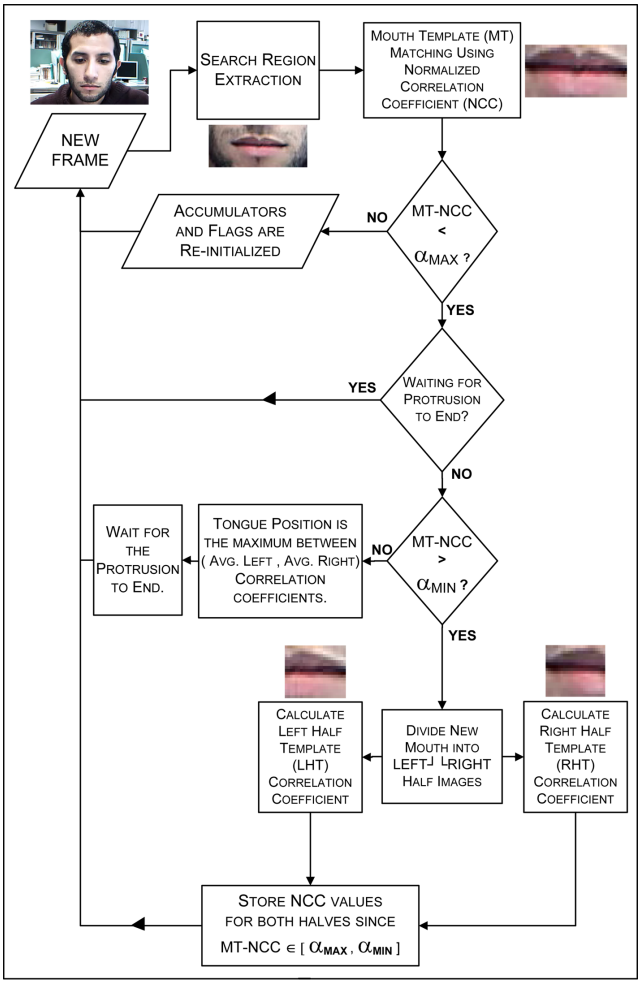

Fig. 7 Flowchart for the tongue protrusion detection.

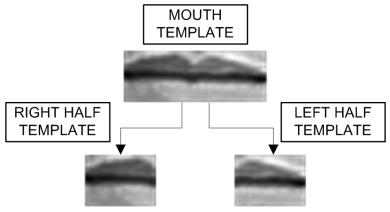

Fig. 8 Templates used for the detection: a mouth template and its corresponding left and right-half templates.

of the face.

The tracking behavior is detailed next. While the tongue protrudes from the mouth, the mouth region starts to change significantly, i.e., the NCC for the best match starts decreasing. Likewise, there is one point at which changes in the mouth region are so large that the NCC tracking is not able to find the new location for the mouth correctly. This produces a drift in the tracked template. This is illustrated in Fig. 9 where we can observe that in the first frame the NCC is high, but as the tongue starts protruding, the NCC decreases and by the sixth frame of the sequence ($NCC = 0.546$), the mouth region is not tracked accurately,

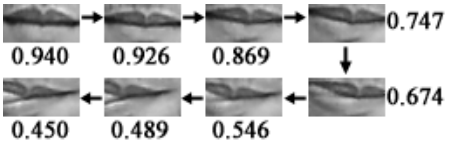

Fig. 9 NCC values for a tracking sequence as protrusion to the right side of the mouth occurs.

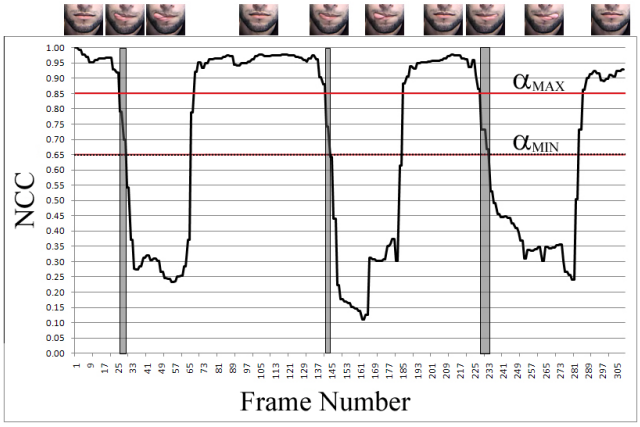

Fig. 10 Changes in the NCC value as the mouth template is tracked in a video sequence. The shadowed box areas are used for the detection of the position of the tongue. Thumbnails of the mouth region are illustrated on top for key frames.

drifting towards the right side due to the tongue protrusion.

4.2.1 Detection of Tongue Protrusion Occurrence

In this section we detect the occurrence of the tongue protrusion, irrespective of the place where it happened. To know if the tongue has been protruded from the mouth, we analyze the changes in the new mouth region compared to the mouth template, i.e., the correlation value for the best match.

Only when the changes are significant, it is safe to consider that the tongue has been taken out of the mouth. In other words, that the NCC is below a certain threshold indicates that the tongue has been protruded. Figure 10 contains the best-match NCC values for a sample video that contains tongue protrusion gestures. In this case, the valleys in the coefficient values correspond to the occurrence of tongue gestures. Consequently, if for instance we set a threshold value $\alpha = 0.85$ (shown as α_{MAX} in Fig. 10), we are able to detect the occurrence of tongue protrusion gestures whenever the coefficient goes below α.

4.2.2 Detection of Tongue Protrusion Position

In the previous subsection, we have used the NCC as a similarity measure between the **MT** and the new mouth region. We have detected the tongue protrusion; however, we have not located it yet. Hence, we will utilize the same measure for comparing the **LHT** and the **RHT** with their corresponding "new" tracked images. As we have discussed, from Fig. 9, for low correlation values we cannot assure that a reliable mouth region has been tracked. Thus, we cannot assume we will permanently obtain precise new left and right images.

From our observations, the issue shown in Fig. 9 is common to all the protrusion events in our video database, i.e., there is a NCC value such as if we go below it, the mouth tracking will drift towards one of the sides. For instance, in the case shown in Fig. 9, this value can be 0.65. We call this value the Minimum Threshold (α_{MIN}).

On the other hand, from Sect. 4.2.1, we know that there exists one threshold value α such as it indicates when the mouth is starting to change, possibly due to a tongue protrusion. We call this value the Maximum Threshold (α_{MAX}).

The position of the tongue protrusion is found by analyzing the NCC values for the images acquired in the time between α_{MAX} and α_{MIN}. For instance, considering now the data shown in Fig. 10, we could choose in addition to $\alpha_{MAX} = 0.85$, $\alpha_{MIN} = 0.65$ (shown as another horizontal line). We observe that in this video sequence, there are three protrusion gestures occurring. Therefore, the data contained inside the three gray regions in Fig. 10 is used for detecting the tongue protrusion position. Figure 11 illustrates closely the changes in correlation values around frame 230. These changes increase according to the amount of tongue that comes out from the mouth.

A flowchart for the threshold-based detection process was previously presented in Fig. 7. The thresholds cooperate as follows:

1. While the **MT**-NCC remains higher than α_{MAX}, we do nothing, as the changes in the mouth are not meaningful.
2. If the **MT**-NCC goes below α_{MAX}, it means that a significant change has started to occur in the mouth image, e.g., due to tongue protrusion.
3. While the **MT**-NCC stays above α_{MIN}, we assume that the new mouth region we have obtained by tracking is still reliable and that the protrusion is still in progress. Thus, we divide the new **MT** into its left and right halves. We compare them with **LHT** and **RHT** of Sect. 4.1. We obtain two more correlation-based similarity measures. These coefficients show us how much each half-side of the mouth has changed.
4. While the **MT**-NCC stays between α_{MAX} and α_{MIN}, we repeat the process described in Step 3. For each new incoming mouth region, we compare and find new **LHT** and **RHT** coefficients. During this interval, we obtain for each new image, a **LHT** and a **RHT-**NCC. In Fig. 11 we have a clear view of the left-half template (LHT-NCC), and the right-half template (RHT-NCC) values for the frames between 229 and 232 (α_{MAX} >**MT**-NCC> α_{MIN}). Hence, values inside this period (gray area) will be stored and used next.
5. Immediately after the **MT**-NCC goes below α_{MIN}, we obtain the average values for the **LHT** and **RHT** that were stored in Step 4. Comparing these values allows detecting the position of the tongue protrusion:

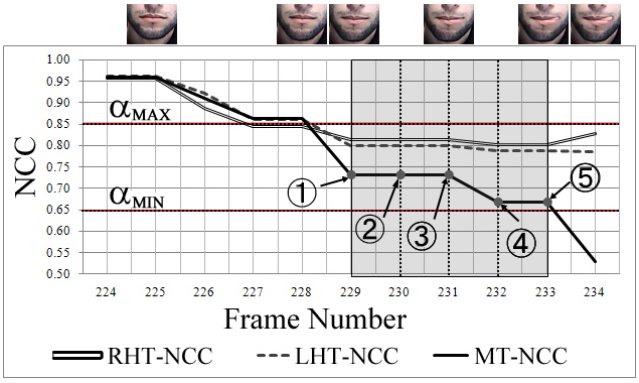

Fig. 11 Close-up of the last protrusion event in Fig. 10. NCC values of the left and right templates, from frame 229 to 233 (Points 1–5), will be used for calculating the tongue position.

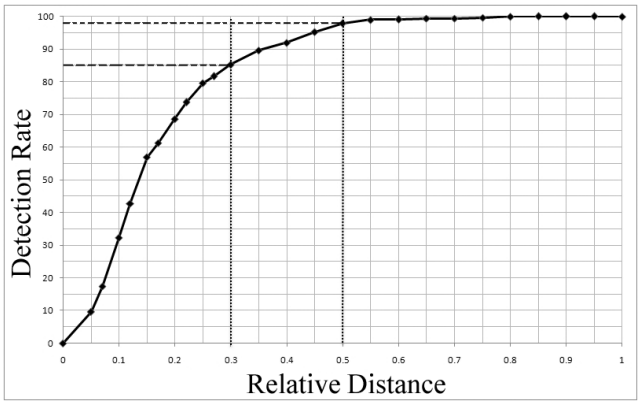

Fig. 12 Detection rate for relative distance values.

the smallest value corresponds to the mouth half that has changed the most; which is assumed to correspond to the mouth half where the tongue protrusion has occurred. For instance, referring again to Fig. 11, the average value is calculated using the information stored in frames indicated from Points 1 to Point 5, since during this interval the **MT**-NCC is between α_{MAX} and α_{MIN}. In this case, just by observing the changes of both **LHT** and **RHT**, we can estimate that the average left value obtained inside the gray area would be smaller. Perceptually, this means that the left side template has changed more in this period of time. Therefore, for this example, we can visually infer that a protrusion has occurred on the left side.

6. If at any moment, the **MT**-NCC surpasses the α_{MAX} value, the system automatically returns to the initial state (Step 1).

This algorithm, therefore, ignores changes that are not as large as the tongue protrusion. For instance, in cases of small head rotations or minor lip movements, the **MT**-NCC goes below α_{MAX} but stays above α_{MIN}; consequently, nothing is detected and those changes are discarded.

5. Results and Discussion

In this section we present the results for the mouth corner detection algorithm proposed in Sect. 3, and for the tongue protrusion detection method and its corresponding position location, given in Sect. 4. The former is executed only once, when the first frame is captured. The latter is executed starting from the second frame until the system is stopped. We also include a brief discussion on the parameters used for the detection to clarify their influence on the results.

5.1 Results for Mouth Corners Detection

We have gathered 252 images from the VALID Database [25], 24 from the CALTECH's Faces 1999 Database[†], and 23 images captured by ourselves, for a total of 300 face images. The former databases are used for facial expression recognition; therefore, we have used only images where users had their mouth closed. The VALID Database provides groundtruth data for the mouth corners. We have manually located the corners in the rest of the images.

For each input image, (x_r, y_r) and (x_l, y_l) locations for right and left mouth corners are output. We could measure the accuracy of our method by comparing our results with the ground truth data directly, i.e., measuring the absolute Euclidean distance that separates them. However; distance values can be large or small depending on the size of the image. Therefore, we need a "relative distance" (rel_d) measure that lets us quantify the accuracy regardless of the image size.

For evaluating eye detection, the BioID Face Database [26] proposed a normalization measure. We adapt it so that it fits our mouth corner scheme. Thus, for each mouth corner pair obtained we obtain rel_d in the following way: (1) find absolute pixel distances d_r and d_l, from the ground truth data, for both mouth corners; (2) choose the greater value between d_r and d_l; (3) rel_d is found by dividing that value by the absolute pixel distance between both mouth corners in the ground truth data. Thereby, distances are normalized, becoming independent from the size of the mouth in the image.

In Fig. 12 we plotted the ROC curve that describes the detection accuracy for rel_d values. For a given rel_d, a mouth is considered as successfully found, if its own *relative distance* measure is less or equal than the given rel_d. The detection rate is then calculated by dividing the number of correctly found mouths by the total number of mouths in the database (300). For instance, for $rel_d = 0.25$, which corresponds to an accuracy of about one quarter of the width of the mouth in the image, we obtain a 79.67% detection rate. If we relax the accuracy so that it is equal to half the width of the mouth region, i.e., $rel_d = 0.5$; the detection rate is 98%. Additionally, for $rel_d \geq 0.8$ the detection rate is 100%, which means that among all images, the greatest error we obtain is no greater that 0.8 times the mouth region width.

[†]http://www.vision.caltech.edu/archive.html

Table 1 Ground truth of the tongue protrusion events.

Video	Length	# Left Pr.	# Right Pr.	# Total
V001	30sec.	3	4	7
V002	36sec.	2	1	3
V003	31sec.	2	4	6
V004	25sec.	2	2	4
V005	33sec.	2	2	4
V006	17sec.	2	1	3
V007	19sec.	3	2	5
V008	19sec.	4	4	8
V009	36sec.	3	2	5
V010	36sec.	3	3	6
TOTAL	282sec.	26	25	51

The BioID Face Database [26] recommends to rate an "eye" as found if $rel_d \leq 0.25$. Different from the eyes location, the appearance of the mouth corners makes it rather difficult to restrict its location to just one particular pixel location. Therefore, the manually labeled data could be slightly different had other person labeled it. In order to account for this small human error, we rate a mouth image as "found" if $rel_d \leq 0.3$. Consequently, from Fig. 12, the detection rate for our system is 85.33%.

The next step consists of fitting a mouth model using as initialization points the mouth corners, as indicated in Sect. 3.3. For this purpose, we have built a database consisting of our own acquired images and images from the BioID Face Database [26], for a total of 360 training images. We have manually registered a 15-point model for each image, and trained the AAM offline using the interface provided in [7]. The model needs 30 parameters for explaining 95.14% of the training database.

Thus, we fit a mouth model in the current image. Finally, we segment a mouth image by extracting the model's bounding box. The complete mouth segmentation takes in average 150ms. for a 320x240 pixel image, using a 2.4GHz CPU and Windows XP-SP3 OS. While the segmentation is not suited for real-time application, it is done only once; hence it does not affect the subsequent processing.

5.2 Results for Tongue Protrusion Detection

It is a novel idea to detect the tongue protrusion from video; hence there are no available databases that include such gesture. We have recorded ten videos from seven different people inside an office environment, and collected a total of 51 tongue protrusion gestures. Table 5.2 shows details about the database we have created. The users have been asked to follow the next guidelines:

- To keep the mouth closed at all times, including when a tongue protrusion occurs. Lips can be relaxed as long as the mouth is not opened.
- Each protrusion must be finished by returning the tongue back into the oral cavity. Since the detection is done while the tongue is sticking out from the mouth, the time the tongue protrusion gesture stays on the surface of the mouth is unimportant.

5.2.1 Evaluation Procedure

We have labeled and counted the following tongue protrusion-related events from our dataset:

1. Missed Events (ME): Protrusions that were not detected.
2. Detected Events (DE): Protrusions that were detected. Events are further separated into:
   a. Uncertain Event (UE): a position could not be calculated because there is no **LHT**-NCC and **RHT**-NCC data, i.e., the number of points between α_{MAX} and α_{MIN} in Fig. 11 is zero.
   b. Misdetected Event (MDE): a decision was taken; however, Right was detected as Left, and vice versa.
   c. Correctly Detected Event (CDE): a correct position decision was taken.

Therefore: $\mathrm{DE} = \mathrm{UE} + \mathrm{MDE} + \mathrm{CDE}$, and $\mathrm{ME} + \mathrm{DE} = 51$, the number of gestures in the database.

Selection of proper threshold parameters α_{MAX} and α_{MIN} is essential for the tongue protrusion detection. Thus, the following measures are analyzed for each pair of threshold parameters:

1. Sensitivity: the proportion of tongue protrusion gestures that were detected. Hence:
$$\mathrm{Sensitivity} = \frac{\mathrm{DE}}{\mathrm{ME} + \mathrm{DE}} \tag{2}$$
2. Accuracy: the proportion of detected tongue protrusion gestures whose position was detected correctly. Hence:
$$\mathrm{Accuracy} = \frac{\mathrm{CDE}}{\mathrm{UE} + \mathrm{CDE} + \mathrm{MDE}} \tag{3}$$

5.2.2 Selection of Thresholds for the Evaluation

We need to choose a set of threshold parameters that allows evaluating the Sensitivity and Accuracy of our method. The upper bound for parameter α_{MAX} should be such as it is lower than the correlation value when the closed mouth is being tracked. The lower bound should be such that small mouth movements are neglected and only meaningful changes in the mouth region trigger a detection. Consequently, for the experiments we have selected two α_{MAX} values: 0.85, 0.80.

We have discussed in Sect. 4.2 that as the tongue comes out from the mouth, the correlation coefficient decreases. We also illustrated in Fig. 9 that the protrusion of the tongue causes a drift in the tracking. The "drifted" template is not symmetric; hence, correlation information obtained from it is not reliable because the left and right "drifted" halves do not correspond to the left and right halves of the original mouth. Therefore, the lower bound for the α_{MIN} parameter should be such that the NCC calculated is still reliable.

Likewise, the upper bound for α_{MIN} is given by α_{MAX}, so that it is feasible to obtain correlation data for the interval $\alpha_{MAX} > NCC > \alpha_{MIN}$. Consequently, the following α_{MIN} values are evaluated: 0.60, 0.65 and 0.70.

5.2.3 Detection Results

Following the evaluation procedure described in Sect. 5.2.1, detection results for one pair of parameters ($\alpha_{MAX} = 0.80$ and $\alpha_{MIN} = 0.70$) are shown in Table 2. Therefore, using Eq. 2 and Eq. 3 yields Sensitivity and Accuracy values of 98.04% and 76.00%, respectively.

Sensitivity and Accuracy values for the remainder pairs of threshold parameters are illustrated in Fig. 13. We can infer from this figure that the Sensitivity is directly proportional to α_{MIN}, and inversely proportional to α_{MAX}. Hence, the best performance is obtained by the pair $\{\alpha_{\text{MAX}}, \alpha_{\text{MIN}}\}$ = {0.80, 0.70}. However, the Accuracy for this pair was inferior compared to other pairs of parameters.

Indeed, we can observe in Tab. 2 that there were 3 Uncertain events (UE) and 9 Misdetections (MDE) when α_{MAX}=0.80 and α_{MIN}=0.70. Both scenarios, exemplified in Fig. 14, are detailed next.

First, a UE from video V003 is shown in Fig. 14(a). In this figure, we can observe that the NCC value at frame 767 was higher than α_{MAX}; and that the NCC value in the next frame (768) was already lower than α_{MIN}, because the protrusion of the tongue occurred very rapidly. Therefore, there was no data between the thresholds α_{MAX} and α_{MIN}; thus, the position of the tongue could not be calculated. UEs are always caused in the same manner.

Then, a MDE from video V004 is shown in Fig. 14(b). In this figure, we can observe the data that is used for the position detection, between frames 54 and 64. In this case, although the tongue protruded to the right side, the average **LHT**-NCC and **RHT**-NCC values between α_{MAX} and α_{MIN} were 0.685 and 0.721, respectively. Thus, since the **LHT**-NCC was lower, a tongue protrusion to the left was detected. Consequently, the detection method for the tongue position that we proposed in Sect. 4.2.2 failed for this particular case. Other misdetections take place similarly; however, their occurrence can be minimized if suitable thresholds are chosen.

In summary, there exists a trade-off between Sensitivity and Accuracy according to the parameters that are chosen. Results shown in Fig. 13 correspond to applying the same pair of parameters to the entire database, allowing us to determine that the best threshold pair in terms of Sensitivity vs. Accuracy is $\{\alpha_{MAX}, \alpha_{MIN}\}$ = {0.85, 0.70}. Thus, these parameters are chosen to be the "default" parameters; and the overall Sensitivity and Accuracy of the system we propose are 90.20% and 84.78%, respectively. New users are expected to obtain satisfactory results with these "default" values on actual use of the proposed system. However, we

Table 2 Detection results for the parameters: α_{MAX}= 0.80 and α_{MIN}= 0.70. **ME:** Missed Events. **DE:** Detected Events. **UE:** Uncertain Events. **CDE:** Correctly Detected Events; and **MDE:** Misdetected Events.

Video ID	ME	DE	UE	CDE	MDE
V001	0	7	2	5	0
V002	0	3	0	3	0
V003	0	6	1	5	0
V004	0	4	0	3	1
V005	1	3	0	2	1
V006	0	3	0	1	2
V007	0	5	0	3	2
V008	0	8	0	7	1
V009	0	5	0	3	2
V010	0	6	0	6	0
TOTAL	1	50	3	38	9

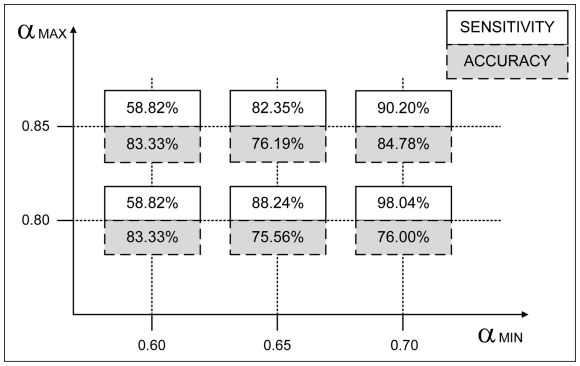

Fig. 13 Sensitivity and Accuracy for each pair of parameters.

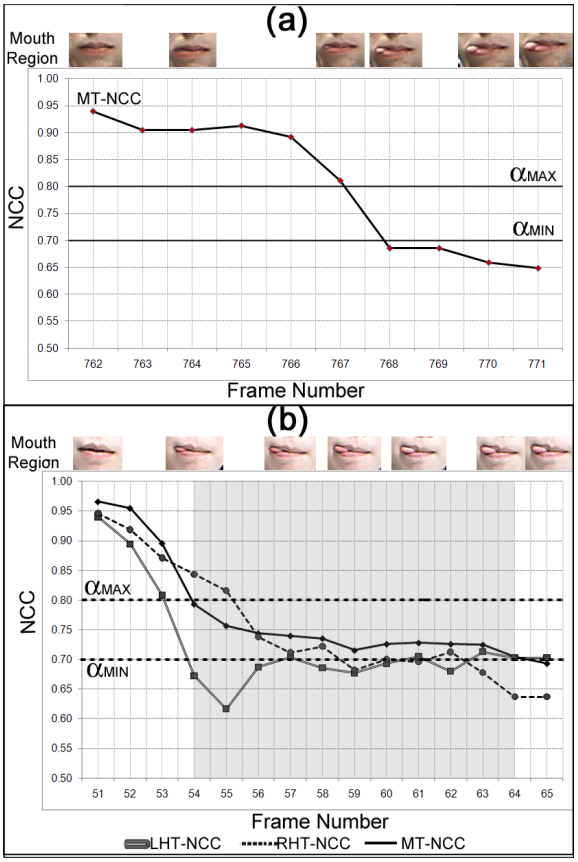

Fig. 14 (a) UE and (b) MDE examples from V003 and V004.

note that an increase in the effectiveness of the detection is possible, if a pair of personalized parameters is chosen for each user during a calibration stage.

The machine learning method previously proposed in [12] was also evaluated using the video database from this paper. As a result, the Sensitivity and Accuracy values for that method were 88.24% and 66.67%, respectively. Hence, overall results from our proposed system are superior. In particular, the detection of the position of the tongue protrusion (the Accuracy) is significantly improved. Finally, the tracking and detection process takes in average 40ms. for a 320×240-pixel image, which demonstrates the real-time capability of the proposed algorithm.

5.3 Discussion on the Parameters

In this section we will discuss the influence of adjusting the "default" parameters for each person, in case it is required in order to increase the effectiveness of the detection.

5.3.1 Lower Threshold Parameter

Decreasing the value of α_{MIN}, may produce the following results. A lower α_{MIN} value requires a larger change in the image for the protrusion to be detected, i.e., a larger portion of the tongue needs to be taken outside the mouth. Nonetheless, a larger portion of protruded tongue means less mouth similarity compared to the original template, which causes the tracking to drift as discussed in Sect. 4.2. If the tracked mouth region is not accurate, the new detected mouth may not be a good match of the original one. This, thereby, affects the "new" right and left templates used for the protrusion detection. Consequently, the left/right matching comparison is not reliable any longer.

Increasing the value of α_{MIN} have the opposite effect: a smaller portion of the tongue now suffices for the protrusion to be detected. However, other actions such as head rotation or subtle lip movement could trigger false positives.

5.3.2 Higher Threshold Parameter

Using higher values of α_{MAX} produces a larger time window for tracking the protrusion occurrence, i.e., the gray region in Fig. 11 is widened. Moreover, for high correlation values the tracked mouth region is very accurate. Thus, correlation coefficients obtained for the left and right template are highly reliable. Nevertheless, since α_{MAX} is used also for detecting when the protrusion event has finished, it could hinder the recovery of the system from the protrusion event if correlation values larger than α_{MAX} are not obtained.

Oppositely, decreasing α_{MAX} produces less information during the protrusion occurrence, i.e., the gray region in Fig. 11 is tightened.

Additionally, if α_{MAX} is too low and α_{MIN} is too high, the tongue protrusion should happen slowly. Otherwise, it might happen that the tongue is protruded so quick that no data is obtained for the interval $\in [\alpha_{MAX}, \alpha_{MIN}]$, i.e., the width of the gray region in Fig. 11 is zero. This will prevent the system from taking a decision about the position.

6. Conclusions and Future Work

In this paper we have proposed a robust method for mouth corners detection from a frontal face image. We have also proposed a system that automatically detects in real-time the tongue protrusion and finds its position.

We have assumed that the dark line between the lips when the mouth is closed is detectable. Hence, the mouth corners detection algorithm used Gabor filters whose orientation and scale parameters were chosen so as to enhance horizontal features in the mouth image. Next, a line whose extremes corresponded to the mouth corners was detected. Obtaining a detection ratio which surpasses 85% on the 300-image dataset showed the correctness of the initial assumption. Finally, by detecting these points, we were able to fit a mouth model and extract the mouth region given by its bounding box.

The tongue protrusion detection utilized video information and relied on two parameters, α_{MIN} and α_{MAX}, that describe the perceptual mouth changes through time. Choosing the proper values makes small lip and mouth movements to be neglected, which in turn allows the user to produce utterances in case speech communication is required.

Figure 13 showed that the detection of the tongue protrusion occurrence has an overall sensitivity of 90.20%. This would prove the effectiveness of our method if our goal were to merely detect the protrusion, as it was in both [14] and [13]. Our method surpasses detection results from [13] using the "default parameters", and surpasses results from [14] when the best parameters are selected. For instance, the Sensitivity rate using parameters $\{\alpha_{\text{MIN}},\alpha_{\text{MAX}}\}$ was 98.04%. Detecting additionally the tongue protrusion position, as either right or left, had an accuracy of 84.78%, which surpasses the accuracy of the only previous work that detected the position of the tongue protrusion [12]. In conclusion, working with video information has been found to perform better than detecting the tongue gestures separately for each frame.

The suitability of parameters depends on physical characteristics of the user such as tongue length or mouth size; thus, if the "default" parameters do not perform well for a user; optimal parameters can be calibrated: (1) heuristically, so that each pair of thresholds is tested for a brief period of time, and the one that performs the best is chosen; or (2) statistically, so that training video data is recorded for tongue gestures made by the potential user, and parameters are calculated offline based on knowledge about the contents of the training information. When building an actual interface that uses tongue protrusion gestures, the latter option may be more accurate; hence, building the statistical calibration method is part of our future work.

As plausible applications for this system, the tongue

protrusion events can be utilized for triggering tasks in the computer, as a hands-free Human-Computer Interface. In particular, it is well suited for supporting physically disabled people by helping them to control a computer. Furthermore, for people who frequently use a computer, the tongue protrusion can work as a complementary interface to be used with the traditional keyboard or mouse, e.g., as a "hands-free hotkey" button that executes a pre-defined task. Finally, possible future work includes the tracking of continuous tongue movement. Doing so would enhance both the utility and usability of the system.

Acknowledgments

The authors would like to thank the reviewers for their useful comments. Luis Ricardo Sapaico was supported by the KDDI Foundation through the Fellowship for Graduate Students from Abroad.

References

[1] M. Turk and G. Robertson, "Perceptual user interfaces (introduction)," Commun. ACM, vol.43, no.3, pp.32–34, 2000.
[2] K. Grauman, M. Betke, J. Lombardi, J. Gips, and G.R. Bradski, "Communication via eye blinks and eyebrow raises: Video-based human-computer interfaces," Universal Access in the Information Society, vol.2, no.4, pp.359–373, 2003.
[3] J.W. Davis and S. Vaks, "A perceptual user interface for recognizing head gesture acknowledgements," PUI '01: Proceedings of the 2001 workshop on Perceptive user interfaces, pp.1–7, 2001.
[4] D.O. Gorodnichy and G. Roth, "Nouse 'use your nose as a mouse' perceptual vision technology for hands-free games and interfaces," Image Vis. Comput., vol.22, no.12, pp.931–942, October 2004.
[5] M. Anisfeld, "Only tongue protrusion modeling is matched by neonates," Developmental Review, vol.16, pp.149–161, June 1996.
[6] T.F. Cootes, G.J. Edwards, and C.J. Taylor, "Active appearance models," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.23, no.6, pp.681–685, 2001.
[7] M.B. Stegmann, B.K. Ersbøll, and R. Larsen, "FAME - a flexible appearance modelling environment," IEEE Transactions on Medical Imaging, vol.22, no.10, pp.1319–1331, 2003.
[8] S. Lucey, S. Sridharan, and V. Chandran, "Adaptive mouth segmentation using chromatic features," Pattern Recognition Letters, vol.23, no.11, pp.1293–1302, 2002.
[9] X. Zhang, C.C. Broun, R.M. Mersereau, and M.A. Clements, "Automatic speechreading with applications to human-computer interfaces," EURASIP J. Appl. Signal Process., vol.2002, no.1, pp.1228–1247, 2002.
[10] F. Jiao, W. Gao, X. Chen, G. Cui, and S. Shan, "A face recognition method based on local feature analysis," Proceedings of the 5th Asian Conference on Computer Vision, pp.188–192, 2002.
[11] M. Lyons, J. Budynek, A. Plante, and S. Akamatsu, "Classifying facial attributes using a 2-D gabor wavelet representation and discriminant analysis," Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on, pp.202–207, 2000.
[12] L.R. Sapaico and M. Nakajima, "Toward a tongue-based task triggering interface for computer interaction," Applications of Digital Image Processing XXX, p.66960J, SPIE, 2007.
[13] B. Leung and T. Chau, "A multiple camera tongue switch for a child with severe spastic quadriplegic cerebral palsy," Disability & Rehabilitation: Assistive Technology, vol.5, no.1, pp.58–68, 2010.
[14] P. Dalka and A. Czyżewski, "Controlling computer by lip gestures employing neural networks," Rough Sets and Current Trends in Computing, pp.80–89, 2010.
[15] P. Viola and M.J. Jones, "Robust real-time face detection," Int. J. Comput. Vision, vol.57, no.2, pp.137–154, 2004.
[16] D. DeCarlo, D. Metaxas, and M. Stone, "An anthropometric face model using variational techniques," SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques, pp.67–74, 1998.
[17] K. Peng, L. Chen, S. Ruan, and G. Kukharev, "A robust algorithm for eye detection on gray intensity face without spectacles," Journal of Computer Science & Technology, vol.5, no.3, pp.127–132, October 2005.
[18] L. Wiskott, J.M. Fellous, N. Krüger, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.19, no.7, pp.775–779, 1997.
[19] B. Kepenekci, "Face recognition using gabor wavelet transform," Master's thesis, The Middle East Technical University, September 2001.
[20] J. Matas, "Robust detection of lines using the progressive probabilistic hough transform," Computer Vision and Image Understanding, vol.78, no.1, pp.119–137, 2000.
[21] M. Betke, J. Gips, and P. Fleming, "The camera mouse: Visual tracking of body features to provide computer access for people with severe disabilities," Neural Systems and Rehabilitation Engineering, IEEE Transactions on, vol.10, no.1, pp.1–10, march 2002.
[22] H. Kim and D. Ryu, "Computer control by tracking head movements for the disabled," Computers Helping People with Special Needs (ICCHP), 10th International Conference on, pp.709–715, 2006.
[23] C. Fagiani, M. Betke, and J. Gips, "Evaluation of tracking methods for human-computer interaction," Applications of Computer Vision, 2002. (WACV 2002). Proceedings. Sixth IEEE Workshop on, pp.121–126, 2002.
[24] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," Computer Vision, 1998. Sixth International Conference on, pp.839–846, January 1998.
[25] N. Fox, B. O'Mullane, and R. Reilly, "Valid: A new practical audio-visual database, and comparative results," Audio- and Video-Based Biometric Person Authentication, pp.777–786, 2005.
[26] O. Jesorsky, K.J. Kirchberg, and R.W. Frischholz, "Robust face detection using the hausdorff distance," Third International Conference on Audio- and Video-based Biometric Person Authentication, pp.90–95, 2001.

Luis Ricardo Sapaico was born in Lima, Peru, on 1979. He received the B.Eng degree of Electronic Engineering from the Pontifical Catholic University of Peru (PUCP) in 2002. He was a research student at the Department of Computer Science, the Graduate School of Information Science and Engineering, Tokyo Institute of Technology, Japan, from April 2004 to March 2005, where he received the Master degree of Computer Science in 2007. He is now a candidate for a PhD degree of Computer Science at the same department. His fields of interest include computer vision, human-computer interaction and interfaces, and assistive technologies.

Hamid Laga received his M.Sc. and PhD degrees in 2003 and 2006, respectively, from Tokyo Institute of Technology. Before joining the Institut Telecom / Telecom Lille1 as an Associate Professor, he has been an Assistant Professor at Tokyo Institute of Technology (2006-2010) and a JSPS post-doctoral fellow at Nara Institute of Science and Technology, Japan (2006). His research interests include 3D-based people detection and tracking, 3D shape analysis and retrieval, non-rigid shape analysis, digital geometry processing, and 3D acquisition. He is the recipient of the Best Paper Award at IEEE International Conference on Shape Modeling and Applications (2006), and the Best Paper Award from the Society of Art and Science Japan (2008).

Masayuki Nakajima received the B.E.E. degree and the Dr. Eng degree from the Tokyo Institute of Technology, Japan, in 1969 and 1975, respectively. Since 1975 he has been with the Department of Imaging Science and Engineering, Tokyo Institute of Technology, Yokohama, Japan. He is now a professor at the Department of Computer Science, the Graduate School of Information Science and Engineering, Tokyo Institute of Technology, Tokyo, Japan. His fields of interest are computer graphics, pattern recognition, image processing and virtual reality.