Hindawi Publishing Corporation
EURASIP Journal on Image and Video Processing
Volume 2007, Article ID 73703, 19 pages
doi:10.1155/2007/73703

Research Article

Cued Speech Gesture Recognition: A First Prototype Based on Early Reduction

Thomas Burger,[1] Alice Caplier,[2] and Pascal Perret[1]

[1] *France Telecom R&D, 28 chemin du Vieux Chêne, 38240 Meylan, France*
[2] *GIPSA-Lab/DIS, 46 avenue Félix Viallet, 38031 Grenoble Cedex, France*

Received 10 January 2007; Revised 2 May 2007; Accepted 23 August 2007

Recommended by Dimitrios Tzovaras

Cued Speech is a specific linguistic code for hearing-impaired people. It is based on both lip reading and manual gestures. In the context of THIMP (Telephony for the Hearing-IMpaired Project), we work on automatic cued speech translation. In this paper, we only address the problem of automatic cued speech manual gesture recognition. Such a gesture recognition issue is really common from a theoretical point of view, but we approach it with respect to its particularities in order to derive an original method. This method is essentially built around a bioinspired method called *early reduction*. Prior to a complete analysis of each image of a sequence, the early reduction process automatically extracts a restricted number of key images which summarize the whole sequence. Only the key images are studied from a temporal point of view with lighter computation than the complete sequence.

1. INTRODUCTION

Among the various means of expression dedicated to the hearing impaired, the best known are sign languages (SLs). Most of the time, SLs have a structure completely different from oral languages. As a consequence, the mother tongue of the hearing impaired (any SL) is completely different from that which the hearing impaired are supposed to read fluently (i.e., French or English). This paper does not deal with the study and the recognition of SLs. Here, we are interested in a more recent and totally different means of communication, the importance of which is growing in the hearing-impaired community: cued speech (CS). It was developed by Cornett in 1967 [1]. Its purpose is to make the natural oral language accessible to the hearing impaired, by the extensive use of lip reading. But lip reading is ambiguous, for example, /p/ and /b/ are different phonemes with identical lip shape. Cornett suggests (1) replacing invisible articulators (such as vocal cords) that participate to the production of the sound by hand gestures and (2) keeping the visible articulators (such as lips). Basically, it means completing the lip reading with various manual gestures, so that phonemes which have similar lip shapes can be differentiated. Thanks to the combination of both lip shapes and manual gestures, each phoneme has a specific visual aspect. Such a "hand and lip reading" becomes as meaningful as the oral message. The interest of CS is to use a code which is similar to oral language. As a consequence, it prevents hearing-impaired people to have an under-specified representation of oral language and helps them to learn to verbalize properly.

The CS's message is formatted into a list of consonant-vowel syllables (CV syllables). Each CV syllable is coded by a specific manual gesture and combined to the corresponding lip shape, so that the whole looks unique. The concepts behind cued speech being rather common, it has been extended to several languages so far (around fifty). In this paper, we are concerned by the French cued speech (FCS).

Whatever the CS, the manual gesture is produced by a single hand, with the palm facing the coder. It contains two pieces of information.

(i) *The hand shape*, which is actually a particular configuration of stretched and folded fingers. It provides information with respect to the consonant of the CV syllable (Figure 1). In order to make the difference between the shape (as it is classically understood in pattern recognition) and the hand shape (as a meaningful gesture with respect to the CS), we call this latter a *configuration*.

(ii) *The location of the hand with respect to the face*. This location around the face is precisely defined by being touched by one of the stretched fingers during the coding (the touching finger is called the *pointing finger*). Its purpose is to provide information about the

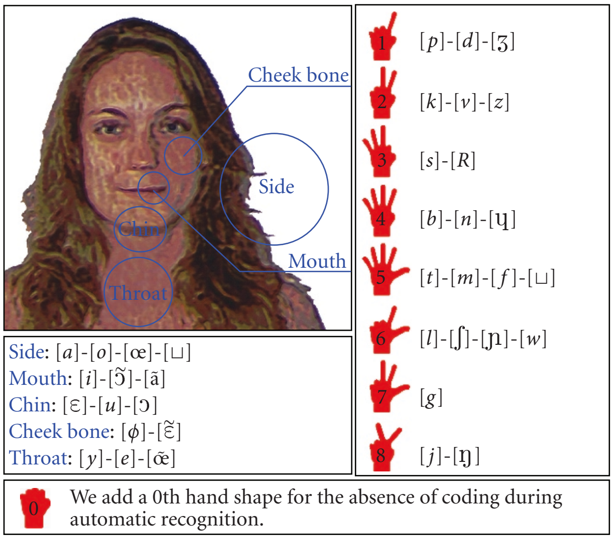


Figure 1: French cued speech specifications: on the left, 5 different hand locations coding vowels; on the right, 8 different hand shapes coding consonants.

vowel of the CV syllable (Figure 1). In the same way, it is necessary to make the difference between the morphologic part of the face being touched by the pointing finger and its semantic counterpart in the code. We call the first *pointed area* and keep the word *location* for the gesture itself.

Hand coding brings the same quantity of information as lip shape. This symmetry explains why

(i) a single gesture codes several phonemes of different lip shapes: it is as difficult to read on the lip without any CS hand gesture, as it is to understand the hand gestures without any vision of the mouth;

(ii) the code is compact: only eight configurations are necessary for the consonant coding and only five locations are necessary for the vowel coding. We add the configuration 0 (a closed fist) to specify the absence of coding, so that we consider a total of nine hand configurations (Figure 1). The configuration 0 has no meaning with respect to the CV coding and consequently it is not associated with any location (it is classically produced by the coder together with the side location but it has no interpretation in the code);

The presented work only deals with the automatic recognition of FCS manual gestures (configuration and location). Therefore, the automatic lip-reading functionality and the linguistic interpretation of the phonemic chain are beyond the scope of this paper. This work is included in the more general framework of THIMP (Telephony for the Hearing IMpaired Project) [2], the aim of which is to provide various modular tools which bring telephone accessible to French hearing-impaired people. To have an idea of the aspect of FCS coding, see examples of videos at [3].

In addition to the usual difficulties for recognition processes of dynamic sequences, CS has several particularities which are the source of extra technical obstacles.

(i) The inner variations of each class for the configurations are so wide that the classes intermingle with each other. Hence, in spite of the restricted number of classes, the recognition process is not straightforward. The same considerations prevail for the location recognition.

(ii) The hand is theoretically supposed to remain in a plan parallel to the camera len, but in practice, the hand moves and our method must be robust regarding minor orientation changes. In practice, this projection of a 3D motion into a 2D plan is of prime importance [4].

(iii) The rhythm of coding is really complicated as it is supposed to fit the oral rhythm: in case of succession of consonants (which are coded as CV with invisible vowels) the change of configuration is really fast. On the contrary, at the end of a sentence, the constraints are less strong and the hand often slows down. For a complete study of the FCS synchronization, from the productive and perceptive point of view of professional coders, see [5].

(iv) From an image processing point of view, when a gesture is repeated (there are twice the same location and configuration), the kinetic clues indicating such a repetition are almost inexistent.

(v) The finger which points the various locations around the face (the pointing finger) depends on the configuration performed at the same time. For instance, it is the medium for configuration 3 and the index for configuration 1.

(vi) Finally, some long transition sequences occur between key gestures. They are to be dealt in the proper way. At least some transition images can contain a hand shape which really looks like any of the configurations by chance, or equivalently, the pointing finger can cross or point a peculiar pointed area which does not correspond to the location of the current gesture: in the corresponding state machine, some states are on the path between two other states.

Knowing all these specifications, the problem is to associate a succession of states to each video sequence. The possible states correspond to the cross product of five locations and eight configurations, plus the configuration 0 (which is not associated to any location to specify the absence of coding), which makes a total of forty-one possible states. Thus, the theoretical frame of our work is widely addressed: the problem is to recognize a mathematical trajectory along time. The methods we should implement for our problem are likely to be inspired by the tremendous amount of work related to such trajectory recognition problems (robotic, speech recognition, financial forecast, DNA sequencing).

Basically, this field is dominated by graphical-based methods under the Markov property [6–8] (hidden Markov chain, hidden Markov model or HMM, Kalman filters, particles filters). These methods are so efficient that their use does not need to be justified anymore. Nonetheless, they suffer from some drawbacks [9].

(i) As the complexity of the problems increases, the models turn to become almost intractable.
(ii) To avoid such things, the models often lose in generality: the training sequence on which they are based is simplified so that both the state machine and the training set have reasonable size.
(iii) The training is only made of positive examples, which does not facilitate the discrimination required for a recognition task.
(iv) They require enormous amount of data to be trained on.

In practice, these technical drawbacks can lead to situations in which the method is not efficient. With respect to our application, difficult situations could materialize in several manners. For instance,

(i) the succession of manual gestures will only be recognized when performed by a specific coder whose inner dynamism is learned as a side effect;
(ii) the improbable successions of manual gestures with respect to the training datasets are discarded (which leads to understand the trajectory recognition problem on a semantic point of view which is far too sophisticated for the phonetic recognition required at the level we work in THIMP).

To avoid some of these drawbacks, several methods have been submitted so far. For a complete review on the matter, see [6].

For our problem, we could apply a method which fits the usual scheme of the state-of-the-art. Image by image processing permits to extract some local features, which are then transmitted to a dynamical process which deals with the data along time. However, we develop a method which is not based on this pattern. The reasons are twofold.

First, it is very difficult to have meaningful data; even if raising the interest of a part of the hearing impaired community, FCS is not that spread yet (it appeared in 1979, so only the younger have been trained since their infancy). Consequently, gathering enough sequences to perform complete training with respect to the French diversity and the potential coding hand variety is very difficult. Moreover, to have a proper coding which does not contain any noxious artifact for the training, one must only target certified or graduated FCS coders, who are very rare compared to the number of various coders we need.

Secondly, from our expertise on the particular topic of FCS gesture, we are convinced that thanks to the inner structure of the code, it is possible to drastically simplify the problem. This simplification leads to an important save in terms of computation. Such a saving is really meaningful for THIMP in the context of the future global integration of all the algorithms into a real-time terminal.

This simplification is the core of this paper and our main original contribution to the problem. It is based on some considerations which are rooted on the very specific structure of CS.

From a linguistic point of view, FCS is the complete visual counterpart of oral French. Hence, it has a comparable prosody and the same dynamic aspect. From a gesture recognition point of view, the interpretation is completely different: each FCS gesture configuration + location is a static gesture (named a *phonemic target* or PT in the remaining of the paper) as it does not contain any motion and can be represented in a single picture or a drawing such as Figure 1. Then, a coder is supposed to perform a succession of PTs. In real coding, the hand nevertheless moves from PT to PT (as the hand cannot simply appear and disappear) and *transition gestures* (TGs) are produced.

We are interested in decoding a series of phonemes (CVs) from a succession of manual gestures which are made of discrete PTs linked by continuous transitions. We formulate in a *first hypothesis* that PTs are sufficient to decode the continuous sentence. As a consequence, complete TG analysis is most of the time useless to be processed (with the saving in terms of complexity it implies).We do not assess that TGs have no meaning by themselves, as we do not want to engage the debate on linguistic purposes. These transitions may carry a lot of information such as paralinguistic clues or even be essential for the human brain FCS decoding task. But it is considered as not relevant here, as we focus on the message made by the succession of PTs.

We also suppose in a *second hypothesis* that the differentiation between TG and PT is possible thanks to low-level kinetic information that can be extracted before the complete recognition process. This is motivated by the analysis of FCS sequences. It shows that the hand is slowing down each time the hand is reaching a phonemic target. As a consequence, PTs are related to smaller hand motion than TGs. It nonetheless appears that there is almost always some residual motion during the realization of the PT (because of the gesture counterpart of the coarticulation).

These two hypotheses are the foundation of the *early reduction*: it is possible (1) to extract some key images via very low level kinetic information, and (2) to apprehend a continuous series of phonemes in a sequence thanks to the study of a discrete set of key images.

The advantages of the *early reduction* are twofold: (1) the computation is lighter as lots of images are discarded before being completely analyzed; (2) the complexity of the dynamical integration is far lower, as the size of the input data is smaller. In this purpose of *early reduction*, we worked in [10] to drastically reduce the number of input images by using the inner structure and dynamic of the gestures we are interested in. In this paper, we sum up and expand this analysis, while linking it with other new works related to segmentation and classification.

We develop a global architecture which is centered on the *early reduction* concept. It is made of several modules. The first one is made of the segmentation tools. We extract the hand shape, its pointing finger, and we define the pointed area of coding with respect to the coder's face position in the image. The second module performs the *early reduction*: its purpose is to reduce the whole image sequence to the images related to PTs. This is based on low-level kinetic information. The third module deals with the classification aspect of locations and configurations on each key image. This is summarized in the functional diagram of Figure 2.

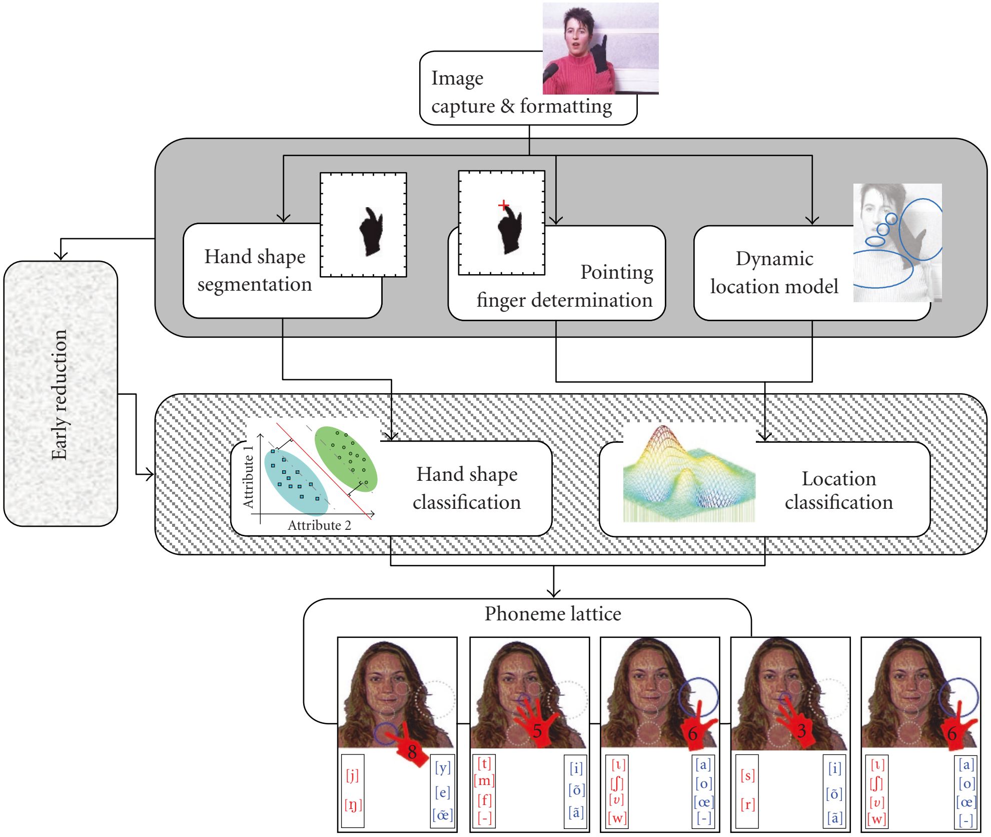


Figure 2: Global architecture for FCS gesture recognition.

In Section 2, we present the image segmentation algorithms required to extract the objects of interest from the video. Section 3 is the core of the paper as the *early reduction* is developed. The recognition itself is explained in Section 4. Finally, we globally discuss the presented work in Section 5: we develop the experimental setting on which the whole methodology has been tested and we give quantitative results on its efficiency.

2. SEGMENTATION

In this section, we rapidly cover the different aspects of our segmentation algorithm for the purpose of hand segmentation, pointing finger determination, and pointed area definition. Pointed area definition requires face detection. Moreover, even if the position of the face is known, the chin, as the lower border of the face, is really difficult to segment. As well, the cheek bone has no strict borders to be segmented from a low-level point of view. Hence, we define these pointed areas with respect to the features which are robustly detectable on a face: eyes, nose, and mouth.

2.1. Hand segmentation

As specified in the THIMP description [2], the coding hand is covered with a thin glove, and a *short* learning process on the color of the glove is done. This makes the hand segmentation easier: the hand often crosses the face region, and achieving a robust segmentation in such a case is still an open issue. The glove is supposed to be of uniform but undetermined color. Even if a glove with separated colors on each finger [11] would really be helpful, we reject such a use, for several reasons.

(i) *Ergonomic reason*: it is difficult for a coder to fluently code with a glove which does not perfectly fit the hand. Consequently, we want the coder to have the maximum freedom on the choice of the glove (thickness, material, color with respect to the hair/background/clothes, size, etc.).

(ii) *Technical reason*: in the long term, we expect to be able to deal with a glove free coder (natural coding). But fingers without glove are not of different color so that we do not want to develop an algorithm related to different colors in order to identify and separate fingers. The glove's presence has to be considered only as an intermediate step.

With the glove, the segmentation is not a real problem anymore. Our segmentation is based on the study of the Mahalanobis distance in the color space, between each pixel and the trained color of the glove. Here follows the description of the main steps of the segmentation process. This process is an evolution of prior works [12].

(1) *Training*: At the beginning of a video sequence, the color of the glove is learned from a statistical point of view and modeled by a 3D Gaussian model (Figure 3). We choose a color space where luminance

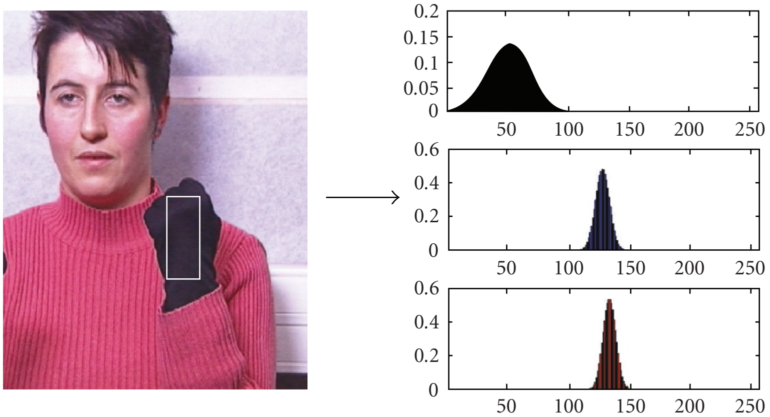


FIGURE 3: Projection in the YCbCr space of the modeling of the learning of the color of the glove.

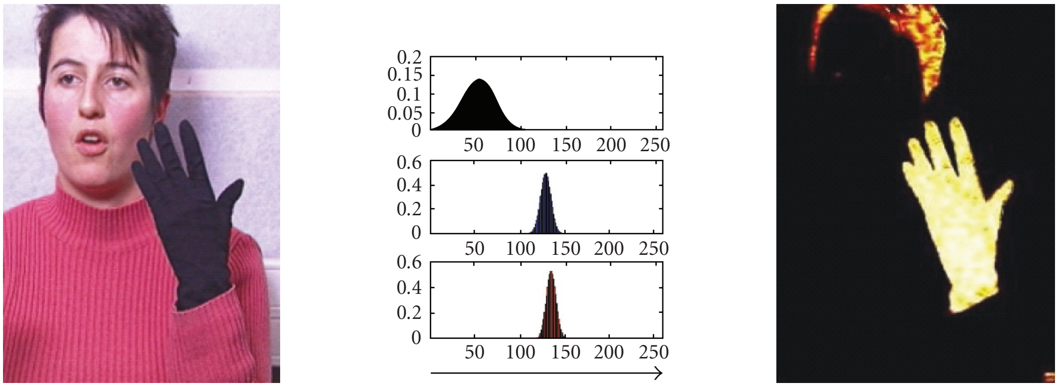


FIGURE 4: Similarity map computation.

and chrominance pieces of information are separated to cope better with illumination variations. Among all the possible color spaces, we use the YCbCr (or YUV) color space for the only reason that the transform from the RGB space is linear, and thus, demanding less computation resources.

(2) *Similarity map*: For each pixel, the Mahalanobis distance to the model of the color's glove is computed. It simply corresponds to evaluate the pixel p under the Gaussian model (m, σ) (Figure 4), where m is the mean of the Gaussian color model, and σ its covariance matrix. We call the corresponding Mahalanobis image the *Similarity Map* (SM). From a mathematical point of view, the *similarity map* is the Mahalanobis transform of the original image:

$$\mathrm{SM}(p) = \mathrm{MT}_{m,\sigma}(p) \quad \text{for } p \in \text{Image}$$
$$\text{with } \mathrm{MT}_{m,\sigma}(p) = 1 - \exp\left(\frac{(p-m)\cdot\sigma\cdot(p-m)'}{2\cdot\det(\sigma)}\right), \tag{1}$$

where det (σ) is the determinant of the covariance matrix σ.

(3) *Light correction*: On this SM, light variations are classically balanced under the assumption that the light distribution follows a centered Gaussian law. For each image, the distribution of the luminance is computed and if its mean is different from the mean of the previous images, then it is shifted so that the distribution remains centered.

(4) *Hand extraction*: Three consecutive automatic thresholds are applied to extract the glove's pixels from the rest of the image. We develop here the methods for an automatic definition of the thresholds.

(a) *Hand localization*: A first very restricting threshold T1 is applied on the SM in order to spot the region(s) of interest where the pixels of the glove are likely to be found (Figure 5(b)). This threshold is automatically set with respect to the values of the SM within the region in which the color is trained. If m is the mean of the color model, and *training* is the set of pixels on which the training was performed,

$$\mathrm{T1} = \frac{1}{2}\cdot\left(m + \frac{\max_{\text{Training}}(\mathrm{SM})}{\min_{\text{Training}}(\mathrm{SM})}\right). \tag{2}$$

(b) *Local coherence*: A second threshold T2 is applied to the not-yet-selected pixels. This threshold is derived from the first one, but its value varies with the number of already-selected pixels in the neighborhood of the current pixel $p(x, y)$: each pixel in the five-by-five neighborhood is attributed a weight according to its position with respect to $p(x, y)$. All the weights for the 25 pixels of the five-by-five neighborhood are summarized in the GWM matrix. The sum of all the weights is used to ponder the threshold T1. Practically, GWM is a matrix which contains a five-by-five sampling of a 2D Gaussian,

$$\mathrm{T2}(x, y) = \frac{3 \cdot \mathrm{T1}}{4} \cdot \left(\sum_{i=-2}^{2} \sum_{j=-2}^{2} \left(\mathrm{GWM}(i, j) \cdot Nbgr_{x,y}(i, j) \right) \right)^{-1} \tag{3}$$

with

$$Nbgr_{x,y} = \begin{pmatrix} \mathrm{SM}(x-2, y-2) \cdots & \cdots & \cdots \mathrm{SM}(x+2, y-2) \\ \vdots & \ddots \quad \iddots & \vdots \\ \vdots & \mathrm{SM}(x, y) & \vdots \\ \vdots & \iddots \quad \ddots & \vdots \\ \mathrm{SM}(x-2, y-1) \cdots & \cdots & \cdots \mathrm{SM}(x+2, y+2) \end{pmatrix},$$

$$\mathrm{GWM} = \begin{pmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{pmatrix}, \tag{4}$$

where $\mathrm{SM}(x, y)$ being the value for pixel $p(x, y)$ in SM. Such a method allows having a clue on the spatial coherence of the color and on its local variation. Moreover, this second threshold permits the pixels (the color of which is related to the glove one) to be connected (Figure 5(c)). This connectivity is important to extract a single object.

(c) *Holes filling*: A third threshold T3 is computed over the values of SM, and it is applied to the not-selected pixels in the fifteen-by-fifteen neighborhood of the selected pixels. It permits to fill the holes as a post processing (Figure 5(d)):

$$\mathrm{T3} = \frac{\max_{\mathrm{Training}}(\mathrm{SM})}{\min_{\mathrm{Training}}(\mathrm{SM})} - 0.1. \tag{5}$$

2.2. *Pointing finger determination*

The pointing finger is the finger among all the stretched fingers, which touches a particular zone on the face or around the face in order to determine the location. From the theoretical definition of CS, it is very easy to determine which finger is used to point the location around the coder's face: it is the longest one between those which are stretched (thumb excluded). Then, it is always the medium but in case of configurations, 0 (as there is no coding), 1 and 6 (where it is the index). This morphologic constraint is very easy to translate into an image processing constraint: the convex hull of the binary hand shape is computed and its vertex which is the furthest from the center of palm and which is higher than the gravity center is selected as the pointing finger (Figure 6).

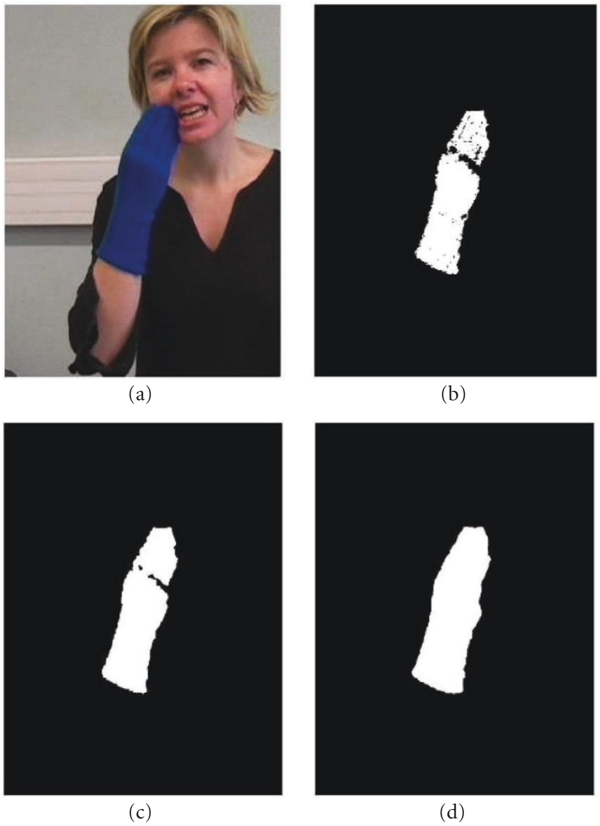


Figure 5: (a) Original image, (b) first threshold (step 3), (c) second threshold and postprocessing (d), third threshold and postprocessing.

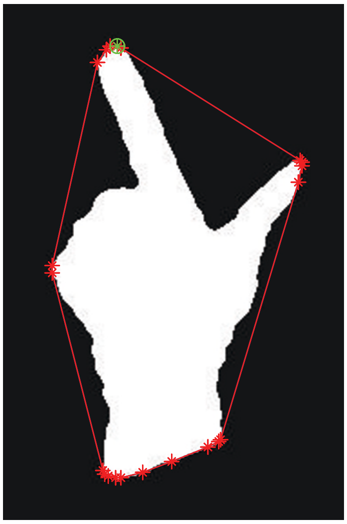


Figure 6: Pointing finger extraction from the convex hull of the hand shape.

2.3. Head, feature, and pointed area determination

In this application, it is mandatory to efficiently detect the coder's face and its main features, in order to define the regions of the image which correspond to each area potentially pointed by the pointing finger. Face and features are robustly detected with the convolutional face and feature finder (C3F) described in [13, 14] (Figure 7). From morphological and geometrical considerations, we define the five pointed areas required for coding with respect to the four features (both eyes, mouth, and nose) in the following way.

(i) *Side*: an ovoid horizontally positioned beside the face and vertically centered on the nose.

(ii) *Throat*: a horizontal oval positioned under the face and aligned with the nose and mouth centers.

(iii) *Cheek bone*: a circle which is vertically centered on the nose height and horizontally so that it is tangent to the vertical line which passes through the eye center (which is on the same side as the coding hand). Its radius is 2/3 of the vertical distance between nose and eyes.

(iv) *Mouth*: the same circle as the cheek bone one, but centered on the end of the lips. The end of the lips is roughly defined by the translation of the eyes centers so that the mouth center is in the middle of the so-defined segment.

(v) *Chin*: An ellipse below the mouth (within a distance equivalent to mouth center to nose center).

Despite the high detection accuracy [14], the definition of the pointed areas varies too much on consecutive images (video processing). Hence, the constellation of features needs to be smoothed. In that purpose, we use a monodirectional Kalman filter Figure 8 represented by the system of equations S:

$$S:\begin{cases} {}^{T}\!\left(x_{t+1}\ \ y_{t+1}\ \ \dfrac{dx_{t+1}}{dt}\ \ \dfrac{dy_{t+1}}{dt}\right) \\ \quad = \begin{pmatrix} \mathrm{Id}(8) & \mathrm{Id}(8) \\ \mathrm{ZERO}_{8\times 8} & \mathrm{Id}(8) \end{pmatrix} \cdot {}^{T}\!\left(x_t\ \ y_t\ \ \dfrac{dx_t}{dt}\ \ \dfrac{dy_t}{dt}\right) \\ \qquad + \propto \mathrm{N}(\mathrm{ZERO}_{8\times 1}, \mathrm{Id}(8)), \\ {}^{T}\!\left(X_t\ \ Y_t\ \ \dfrac{dX_t}{dt}\ \ \dfrac{dY_t}{dt}\right) \\ \quad = {}^{T}\!\left(x_t\ \ y_t\ \ \dfrac{dx_t}{dt}\ \ \dfrac{dy_t}{dt}\right) + \propto \mathrm{N}\left(\mathrm{ZERO}_{8\times 1}, \mathrm{cov}\left(\dfrac{dZ}{dt}\right)\right), \end{cases} \tag{6}$$

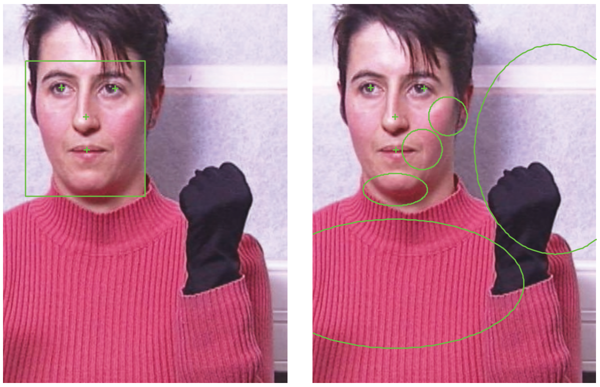
where

(i) x_t and y_t are the column vectors of the horizontal and vertical coordinates of the four features (both eyes, nose and mouth centres) in the image at time t and X_t and Y_t their respective observation vectors;

(ii) $Id(i)$ is the identity matrix of size i, and $\mathrm{ZERO}_{i\times j}$ is the null matrix of size $i \times j$;

(iii) $\propto$ N *(param1, param2)* is a random variable which follows a Gaussian law of mean *param1* and of covariance *param2*;

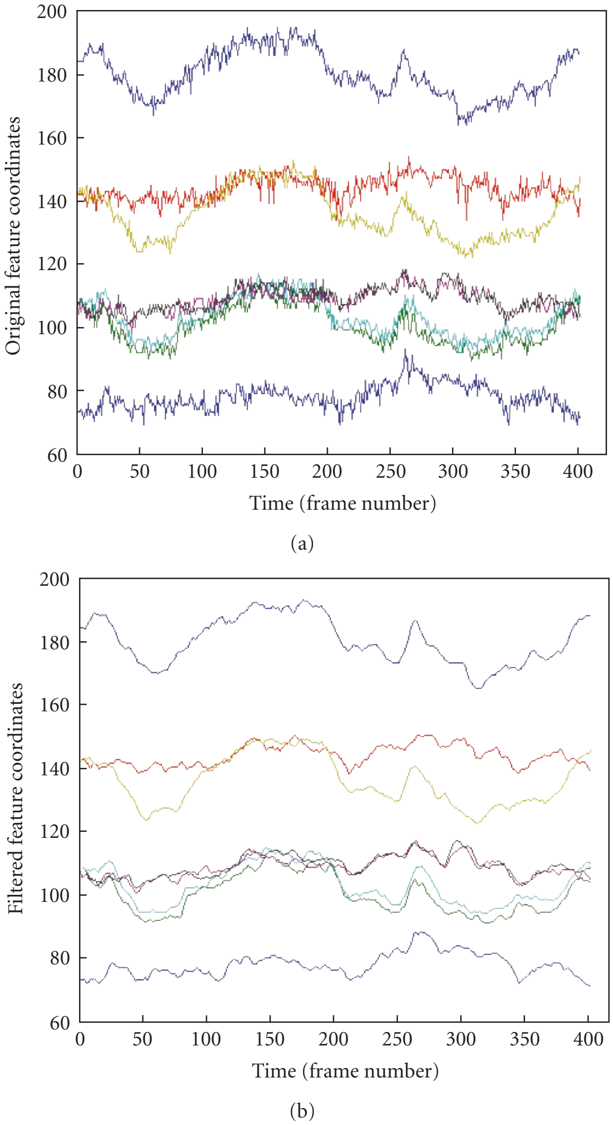
(iv) dZ/dt is a training set for the variability of the precision of the C3F with respect to the time.



(a) Convolutional Face and feature finder result [14]

(b) Pointed areas definition with respect to the features

Figure 7: determination of the pointed areas for the location recognition.



(a)

(b)

Figure 8: Projection of each of the eight components of the output vector of the C3F and the same projection after the Kalman filtering.

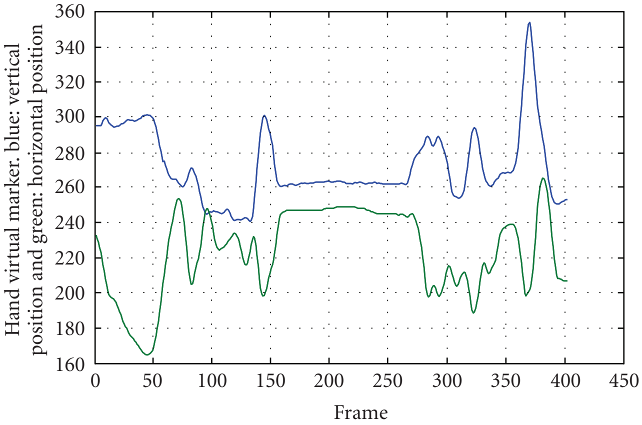
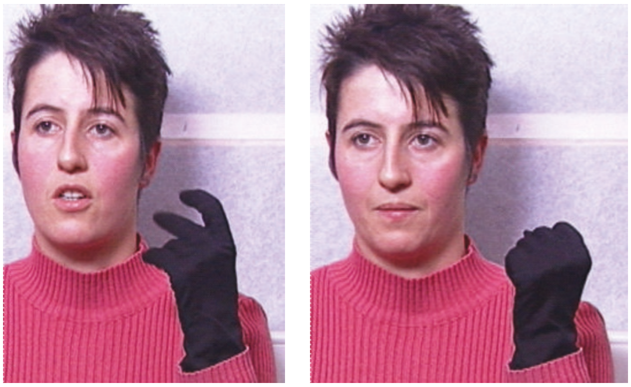


Figure 9: Example of the hand gravity center trajectory along time (x coordinate above and y coordinate below). Vertical scale: pixel. Horizontal scale: frame.



(a) During a transition no location is pointed

(b) Closed wrist does not refer to any position

Figure 10: Hand shapes with no pointing finger.

3. EARLY REDUCTION

3.1. Principle

The *early reduction* purpose is to simplify the manual gesture recognition problem so that its resolution becomes easier and less computationally expensive. Its general idea is to suppress processing for transition images and to focus on key images associated to PTs. The difficulty is to define the key images prior to any analysis of their content. As we explained in the introduction,

(i) images corresponding to PTs are key images in the meaning that they are sufficient to decode the global cued speech gesture sequence;

(ii) Around the instant of the realization of a PT, the hand motion decreases (but still exists, even during the PT itself) when compared to the TG.

The purpose of this section is to explain how to get low-level kinetic information which reflects this motion variation, so that the PTs instants can be inferred.

When coding, the hand motion is double: a global hand rigid motion associated to location and a local nonrigid fingers motion associated to configuration formation. The global rigid motion of the hand is supposed to be related to the trajectory of the hand gravity center. Such a trajectory is represented in Figure 9, where each curve represents the variation of a coordinate (x or y) along time. When the hand remains in the same position, the coordinates are stable (which means the motion is less important). When a precise location is reached, it corresponds to a local minimum on each curve. On the contrary, when two consecutive images have very different values for the gravity center coordinates, it means that the hand is moving fast. So, it gives very good understanding of the stabilization of the position around PTs (i.e., the motion decreases).

Unfortunately, this kinetic information is not accurate enough. The reasons are twofold:

(i) when the hand shape varies, the number of stretched fingers also varies and so varies the repartition of the mass of the hand. As a consequence, the shape variations make the gravity center moving and looking unstable along time;

(ii) the hand gravity center is closer to the wrist (the joint which rotates for most of the movement) than the pointing finger, and consequently, some motions from a position to another one are very difficult to spot.

As a matter of fact, the pointing finger position would be a better clue for the motion analysis and PTs detection, but on transition images as well as when the fist is closed, it is impossible to define any pointing finger. This is illustrated on the examples of Figure 10.

Thus, the position information (the gravity centre or the pointing finger) is not usable as it is, and we suggest focusing on the study of the deformation of the hand shape to get the required kinetic information.

Because of the lack of rigidity of the hand deformation, usual methods for motion analysis such as differential and block matching methods [15] or model-based methods [16] are not well suited. We propose to provide the *early reduction* thanks to a new algorithm for motion interpretation based on a bioinspired approach.

3.2. Retinal persistence

The retina of vertebrates is a complex and powerful system (of which the justification of the efficiency roots in natural selection process) and a large source of inspiration for computer vision. From an algorithmic point of view [17], a retina is a powerful processor, in addition, it is one of the most efficient sensors: the sensor functionality permits the acquisition of a video stream and a succession of various modules processing them, such as explained in Figure 11. Each module has a specific interest, such as smoothing the variations of illumination, enhancing the contours, detecting, and analyzing motions.

Among all these processes, there is the inner plexiform cells layer (IPL) filtering. It enhances moving edges, particularly edges perpendicular to the motion direction. Its output can easily be interpreted in terms of retinal persistence: the faster an object goes in front of the retina, the blurriest the

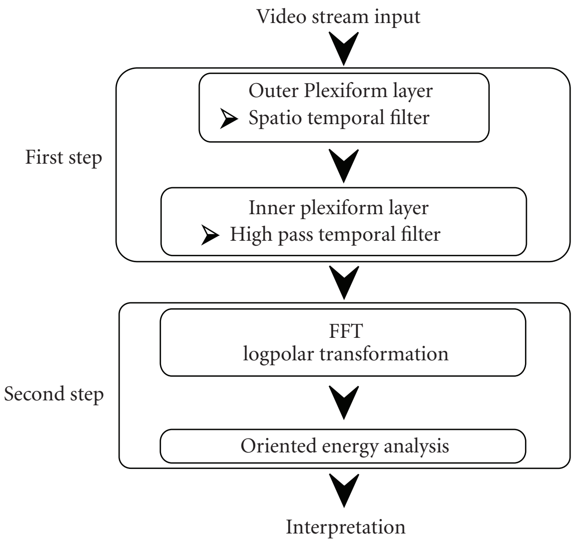


Figure 11: Modeling of the global algorithm for the retina processing [17].

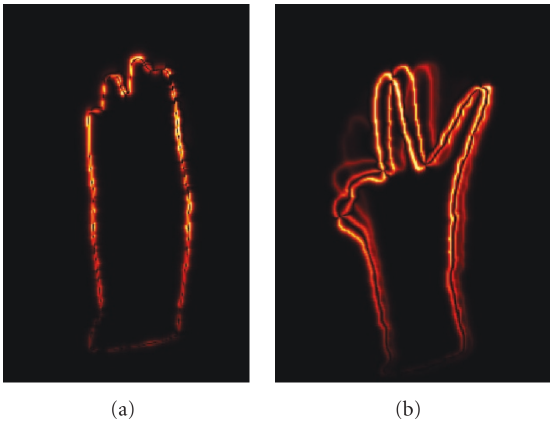


Figure 12: IPL output for (a) a potential target image (local minimum of the motion), (b) a transition image (important motion).

(perpendicular to motion) edges are. Roughly, the IPL filter can be approximated by a high-pass temporal filter, (as indicated in Figure 11) but for more comprehensive description, see [17].

By evaluating the amount of persistence at the IPL filter output, one can have a clue on the amount of motion in front of the retina sensor. This can be applied to our gesture recognition problem. As shown in Figure 12, it is sensible to use the retinal persistence to decide whether the hand is approximately stable (it is likely to be a target) or not (it is likely to be a transition).

Our purpose is to extract this specific functionality of the retina and to pipeline it to our other algorithms in order to create a complete "sensor and preprocessor" system which meets our expectation on the dedicated problem of gesture recognition: *the dedicated retina filter* .

3.3. Dedicated retina filter

The *dedicated retina filter* [9] is constituted of several elements which are chained together, as indicated in Figure 13.

(1) A *video sensor*. It is nothing more than a video camera.

(2) *Hand segmentation*, which has been described in Section 4. At the end of the segmentation process, the hand is rotated on each image so that on the global sequence, the wrist basis (which is linked to the forearm) remains still. In this way, the global motion is suppressed, and only the variation of shape is taken into account.

(3) An *edge extractor*, which provides the contours of the hand shape. It is wiser to work on the contour image because, from a biological point of view, the eye is more sensitive to edges for motion evaluation. As extracting a contour image from a binary image is rather trivial, we use a simple subtraction operator [18]. The length L of the closed contour is computed.

(4) A *finger enhancer*, which is a weighted mask applied to the contour binary image. It makes the possible positions of the fingers with respect to the hand more sensitive to the retinal persistence: as the changes in the hand shape are more related to finger motions that palm or wrist motion, these latter are underweighted (Figure 14(a)). The numerical values of the mask are not optimized, and there is no theoretical justification for the choice of the tuning described in Figure 14(b). This is discussed in the evaluation part.

(5) A *smoothing filter*, which is a 4 operations/byte approximation of a Gaussian smoother [17]. Such a filter appears at the retina preprocessing to the IPL.

(6) The *inner plexiform layer (IPL)* itself, which has already been presented in the previous paragraph 3.2 as the core of the retinal persistence.

(7) A *sum operator*, which integrates the output of the IPL filter in order to evaluate the "blurriness" of the edges, which can directly be interpreted as a motion energy measure. By dividing it by the edge length, we obtain a normalized measure which is homogenous with a speed measure:

$$\text{Motion Quantification}(\text{frame}_t) = \frac{1}{L}\cdot\sum_{x,y}\text{IPL output}_t(x,y), \tag{7}$$

where frame_t represents the current tth image, L represents the length of the contour of the shape computed in the *edge extractor* module, and $\text{IPL output}_t(x,y)$ represents the value of the pixel (x,y) in the image result of the processing of frame_t by modules (0) to (5) of the dedicated retina filter.

3.4. Phonemic target identification

The motional energy given as output of the *dedicated retina filter* is supposed to be interpreted as follows: at each time t, the higher the motional energy is, the more the frame at time t contains motion, and vice versa. On Figure 15, each

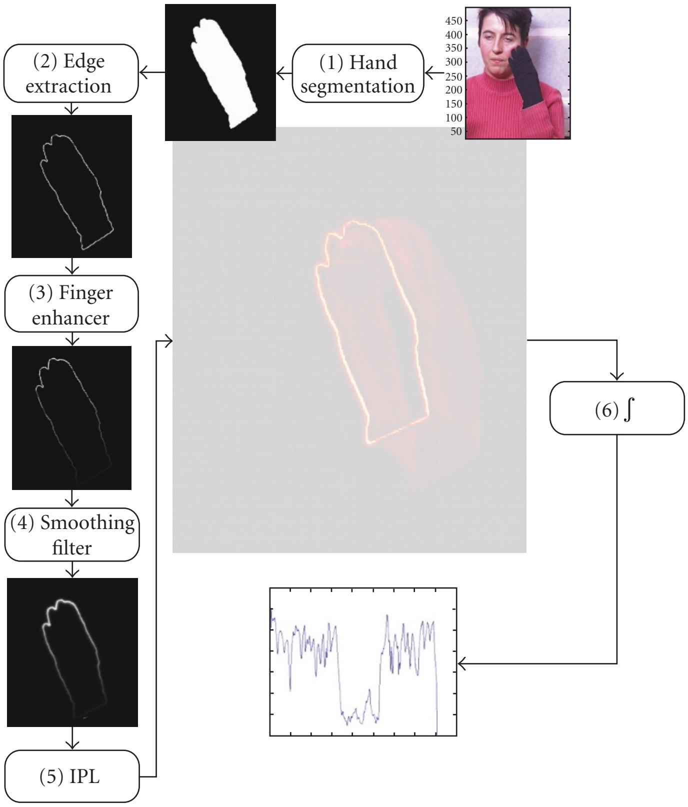
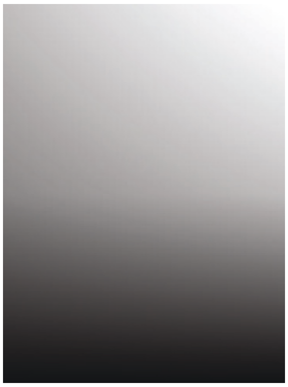


Figure 13: Dedicated retina filter functional diagram.



(a) grayscale representation of the weight mask (the darker the gray, the lower the weights)

Upper part: square root evolution of the weight $w(x, y)$ along the $(Y\max.\vec{y} + X\max.\vec{x}/2)$ vector.

$w(x, y) = 0.5$ if $y = 0$
$w(x, y) = 1$ if $(x, y) = (X\max, Y\max)$

Lower part: linear evolution of the weight $w(x, y)$ along the y vector.

$w(x, y) = 0$ if $y = 0$
$w(x, y) = 0.5$ if $y = Y\max/2$

(b) Expression of the mask for each pixel $p(x, y)$. The lower left-hand corner is the reference, and $(X\max, Y\max)$ are the dimensions of the image

Figure 14: Weight mask for the finger enhancement.

minimum of the curve is related to a slowing down or even a stopping motion. As the motion does not take into account any translation or rotation, which are global rigid motion, the amount of motion only refers to the amount of hand-shape deformations in the video (fingers motion).

Hence, any local minimum in the curve of Figure 15 corresponds to an image which contains less deformation than the previous and next images: such an image is related to the notion of PTs as defined above. Unfortunately, even if the relation is visible, the motional energy is too noisy a signal to allow *direct* correspondence between the local minima and the PTs: the local minima are too numerous.

Here are the reasons of such noisiness.

(i) A PT is defined from a phonemic point of view which is a high-level piece of information: whatever the manner the gesture is made, it remains a single PT per gesture. On the contrary, a local minimum in the motion can have several origins: the motion may be jerked, or the gesture may require several accelerations and decelerations for morphologic reasons; it is

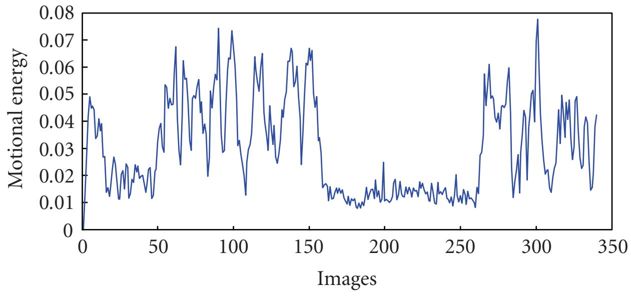


Figure 15: Dedicated retina filter output: the normalized motional energy per image along time.

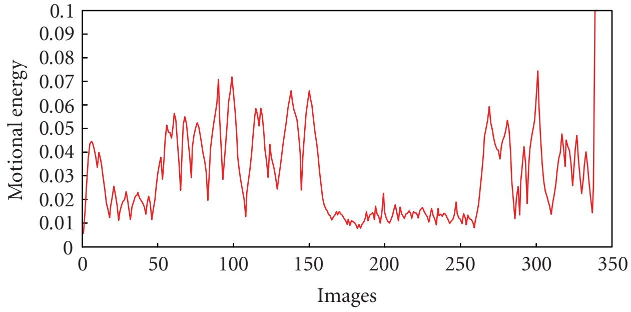


Figure 16: The filtered dedicated retina output.

simply related to the kind of motion (speed, acceleration, jerk) and it is a very low-level piece of information. Consequently, several such instants in which a relative stability is measured can appear in a single gesture. These instants must be filtered in order to keep only the ones which are likely to have a higher level of interpretation.

(ii) Because of the nature of the dedicated retina filter (especially the IPL filter), its output is noisy (there are lots of local minima of no meaning from a phonetic point of view).

(iii) Any mistake in the previous processing can also lead to unjustified local minima (noise in the segmentation, relative sensitivity of the captor to lighting variations, approximation of considering the motion as planar, etc.).

For all these reasons, it is impossible to simply associate local minima to PTs. On the contrary, it appears from common sense that any image which really corresponds to a PT is a local minimum. This is confirmed by experiments (see Section 5). Finally, the set of all the local minima is too big to be associated to the set of the phonemic targets, but contains it. We consider the set of local minima as a first step of the *early reduction*, and the corresponding images are considered as targets of a very low level called KT1 (which stands for kinetic target of type 1) on which set of targets of higher level will be defined.

The point is now to define a set of KT2 based on the set of KT1, (in which all the useless KT1 has been removed, and in which no PT is missing). For that purpose, the motion energy is filtered, so that the small variations are smoothed and the important variations are enhanced (Figure 16). The remaining KT1 on the filtered motional energy curve are considered as KT2. To perform such a filtering, we use a series of convolutions with the following kernel *ItKe*:

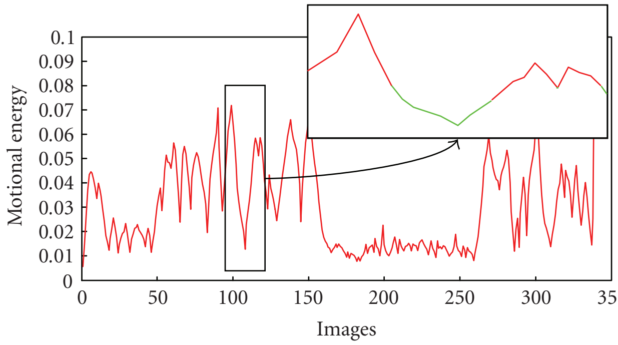


Figure 17: A zone of stability (in green bold) determined by hysteresis cycle. Its minimum value corresponds to its representing KT3.

$$ItKe = (0.1 \quad 0.2 \quad 0.4 \quad 0.2 \quad 0.1). \tag{8}$$

After each iteration, the remaining local extrema are set back to their original values, so that only the small variations are suppressed. In practice, three iterations are sufficient.

The KT2 images are images which potentially correspond to the PTs, but they still remain too numerous. The next step of the reduction is to define an equivalence class for all the KT2 which correspond to the same gesture. The difficulty is to group the images of the same gesture without analyzing the images of the sequence.

To create these equivalence classes, we simply group the consecutive images together, under the hypothesis that any change of gesture leads to an important amount of retinal persistence. As soon as the motional energy becomes too high (higher than *Inf Tresh*), the gesture is supposed to be not stable enough any more (one leaves the previous equivalence class of stable images). As soon as the motional energy becomes small enough (smaller than *Sup Tresh*) the gesture is supposed to be approximately stable back and enter a new class of images considered as being equivalent (Figure 17). *Sup Tresh* must be higher than *Inf Tresh* (it defines a hysteresis cycle) to take into account the delay induced by the temporal filtering of the IPL.

If the thresholds are not properly set, two kinds of error can appear. We call *error of type 1* a transition which is not spotted; in such a case the previous gesture and next gesture are merged together. We call *error of type 2* a transition which is detected whereas none occurs; in such a case, a gesture is split into two. As a matter of fact, errors of type 2 are really easy to correct. Then, it is not necessary for the thresholds to be precisely tuned, as long as they prevent any error of type 1. That is why we have roughly and manually selected them to the value of 30% and 40% of the maximum values reachable by the motional energy (which is 0.1): *Inf Tresh* = 0.03 and *Sup Tresh* = 0.04.

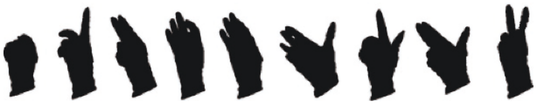


Figure 18: Examples from KT3 images for classes 0 to 8, respectively.

Once the equivalence classes of KT2 are defined, the most representative KT2 element of each class (the one which has the lowest motional energy value among the equivalence class) is defined as KT3, the kind of kinetic targets of the highest level of interpretation: the early reduction purpose is to define KT3s which are as closed as possible to the theoretical PTs.

To correct an error of type 2, it is sufficient to compare the result of the recognition for each KT3 (i.e., after the recognition stage which is described in the next section). If consecutive KT3s are recognized as containing the same configuration, it has two possible meanings:

(i) two identical configurations have been produced in two PTs, no mistake has been made;

(ii) a single PT has been cut in two by mistake and a single configuration has provided two KT3s.

To make the difference between these two cases, it is sufficient to process the single TG image which corresponds to the local maximum (the image of maximum motion) between the two considered KT3s. If the same hand shape is recognized during the TG, it means that a single PT has been artificially cut into two. In addition to all the RT3 images, some TGs images are processed (their number obviously varies, as it is discussed in Section 5).

The interests of using a hierarchical definition for the KTs (KT1s, KT2s, and KT3s) instead of using a direct method to extract KTs which correspond to PTs are manifold.

(i) *Stronger reduction*: we have got the following relationship which must be enforced:

$$\{\text{KT3s}\} \subseteq \{\text{KT2s}\} \subseteq \{\text{KT1s}\}. \tag{9}$$

Then, by explicitly defining intermediate level of KTs, we pedagogically explain that the target images must be recognized as such at various levels of interpretation. For example, several local minima after the smoothing by the *ItKe* kernel are not in KT2s, whereas they fulfil the other conditions: they do not belong to KT1s because of the nonzero phase of the convolutional filter.

(ii) *Computation resources*: the definition of intermediate KTs allows recognizing fewer images for the definition of equivalence classes with respect to the gesture contained in the image.

(iii) *Extension to future works*: in our future work we expect to automatically spot. Some of the mistakes due to the system. Then, a hierarchical definition of targets of various levels of interpretation would allow correcting them more easily by descending the level of interpretation.

4. HAND-SHAPE CLASSIFICATION

In this section, we are interested in the classification of a KT3 image. Working on KT3 simplifies the recognition task for two reasons.

(i) For each zone of stability that corresponds to an equivalence class for the KT2, all the images have their hand shape recognized through the recognition of the single corresponding KT3.

(ii) The configurations to recognize are fully realized on PTs. So, there is less variance to take into account, and the classes are well defined and bounded (in opposition to when transitions are taken into consideration).

Figure 18 is an example of the kind of images that are obtained in the KT3 set, and which are likely to be classified. Some images represent imperfect gestures, such as the examples of configurations 3 and 4. As explained in Section 2, by nature of FCS, the hand shapes obtained are prone to numerous artefacts which complicate the classification task. KT3 images are more stable, but this stability remains relative.

4.1. Preprocessing: the wrist removal

The wrist is a source of variation: (1) it is a joint the shape of which varies, and (2) its size varies with the glove. Hence, one does not want to perform any learning on it, and we simply remove it. We define the wrist as the part of the hand which is under the palm (Figure 19). We define the palm as the biggest inner circle of the hand. We find it via a distance transform which is computed over the binary image coming from the segmentation step. The purpose of the distance transform is to associate to each pixel of an object in the binary image a value which corresponds to the Euclidian distance between the considered pixel and the closest pixel belonging to the background of the image. For morphological reasons [19], the center of the palm is the point of the hand the value of which is the highest in the corresponding distance transform image (Figure 19).

4.2. Attributes definition

Several image descriptors exist in the image compression literature [20]. We focus on Hu invariants, which are successful in representing hand shapes [4]. Their purpose is to express the mass repartition of the shape via several inertial moments of various orders, on which specific transforms ensure invariance to similarities. Centered inertial moments are invariant to translation. The moment m_{pq} of order $p+q$ is defined as

$$m_{pq} = \iint_{x\,y} (x-\overline{x})^{p}(y-\overline{y})^{q}\delta(x,y)\,dx\,dy. \tag{10}$$

With $\overline{x}$ and $\overline{y}$ being the coordinates of the gravity center of the shape and $\delta(x,y) = 1$ if the pixel belongs to the hand and 0 otherwise. The following normalization makes them invariant to scale:

$$n_{pq} = \frac{m_{pq}}{{m_{00}}^{(p+q)/2+1}}. \tag{11}$$

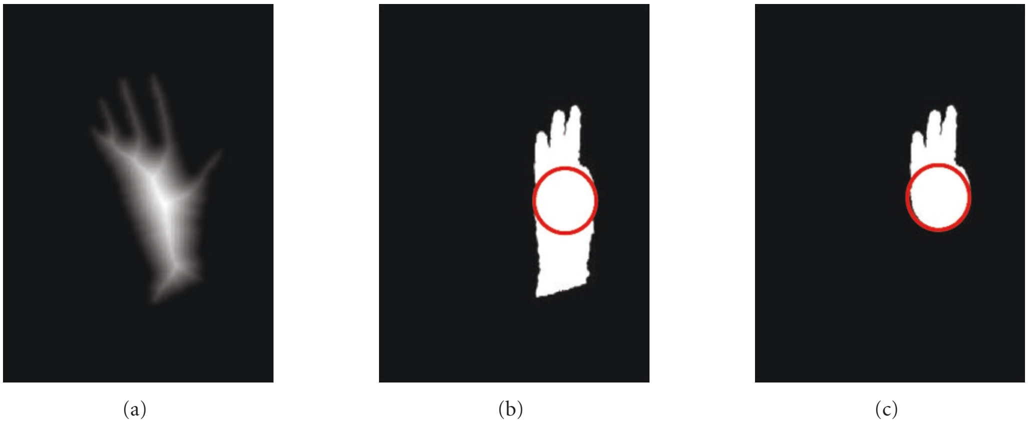


Figure 19: grayscale representation (the lighter the gray, the further from the border) of the distance transform of a stretched hand (the edge of each finger appears), and the use of such a transform applied to remove the wrist on a peculiar hand shape (b, c).

Then, we compute the seven Hu invariants, which are invariant to scale and rotation [4, 20].

$$\begin{aligned}
S_1 &= n_{20} + n_{02},\\
S_2 &= (n_{20} + n_{02})^2 + 4\cdot n_{11}^2,\\
S_3 &= (n_{30} - 3\cdot n_{12})^2 + (n_{03} - 3\cdot n_{21})^2,\\
S_4 &= (n_{30} + n_{12})^2 + (n_{03} + n_{21})^2,\\
S_5 &= (n_{30} - 3\cdot n_{12})\cdot(n_{30} + n_{12})\\
&\quad \cdot\left((n_{30} + n_{12})^2 - 3\cdot(n_{03} + n_{21})^2\right) - (n_{03} - 3\cdot n_{21})\\
&\quad \cdot(n_{03} + n_{21})\cdot\left(3\cdot(n_{30} + n_{12})^2 - (n_{03} + n_{21})^2\right),\\
S_6 &= (n_{20} + n_{02})\cdot\left((n_{30} + n_{12})^2 - (n_{03} + n_{21})^2\right)\\
&\quad + 4\cdot n_{11}^2\cdot(n_{30} + n_{12})\cdot(n_{03} + n_{21}),\\
S_7 &= (3\cdot n_{21} - n_{03})\cdot(n_{30} + n_{12})\cdot\left((n_{30} + n_{12})^2\right.\\
&\quad \left.- 3\cdot(n_{03} + n_{21})^2\right) - (n_{30} - 3\cdot n_{12})\cdot(n_{03} + n_{21})\\
&\quad \cdot\left(3\cdot(n_{30} + n_{12})^2 + (n_{03} + n_{21})^2\right).
\end{aligned} \tag{12}$$

4.3. Classification methodology

For the classification itself, we use support vector machines (SVMs) [21]. SVMs are binary classification tools based on the computation of an optimal hyperplane to separate the classes in the feature space. When the data are not linearly separable, a kernel function is used to map the feature space into another space of higher dimension in which the separation is possible. We use the following.

(i) "One versus one" methodology for the multi-classification aspect. It means that to deal with multiple classes, one uses a SVM per pair of classes, and the final classification is derived from the result of all the binary classifications.

(ii) A voting procedure: each SVM gives a vote to the class it selects and the final classification is achieved by choosing the class which gatheres the highest score.

(iii) The C-SVM algorithm [22].

(iv) Sigmoid kernels, in order to transform the attribute space so that it is linearly separable.

5. DISCUSSION ON THE OVERALL METHOD

In order to evaluate the algorithms presented in this paper, we mainly use a specific corpus which corresponds to a single experiment campaign. Its conditions of acquisition perfectly fit the general situation which our system is supposed to work on. Consequently, it is used to test all the algorithms. Sometimes, this setting is not sufficient to make all the evaluations and we use some other experiments in parallel which are dedicated to a peculiar algorithm (segmentation, classification, etc.). The first main data collection is described in the first paragraph. In the following paragraphs, each algorithm is evaluated with respect to this main corpus. If additional minor datasets are required, they are described in the corresponding paragraph.

5.1. Experimental setting and data collection

The main data collection deals with a corpus of 267 sentences of very uninteresting (or inexistent) meaning, but with the particularity of presenting all the potential transitions among the French phonemes. This lack of clarity leads to a coding which is not perfectly fluent, as there are some hesitations, or mistakes, which are finally also present in an unprepared talk. The sentences are long from five to twenty-five syllables and have elaborated structures. No learning is performed on their semantic level, so that the power of generalization of this dataset is complete: any new sentence acquired in the same conditions is processed in a similar way and gives similar result. Consequently, the mere content of the corpus in terms of linguistic meaning is not as important as the other variation factors (coder, lightening conditions, camera quality, etc.), which may have far more consequences.

The coder is a native French female, certified for FCS translation, and working regularly as a translator (in schools, meetings, etc.). She codes in a sound-proof room, with professional studio lightening conditions, sitting in front of a camera, and using the thin black-silk glove she usually used to protect her hands from cold (consequently, the glove is really chosen up to the coder). This is the first time she uses a glove for a coding acquisition, and after a short warm up, she is not bothered anymore by its presence. It appears that the glove color is very close to the coder's hair. In order to assess the choice of the glove with respect to (1) the comfort of its use, (2) its colour for segmentation purpose, (3) the difficulty of recognizing a badly chosen glove, we also made few acquisitions with a thick blue glove which is two sizes too big for the coder's hand.

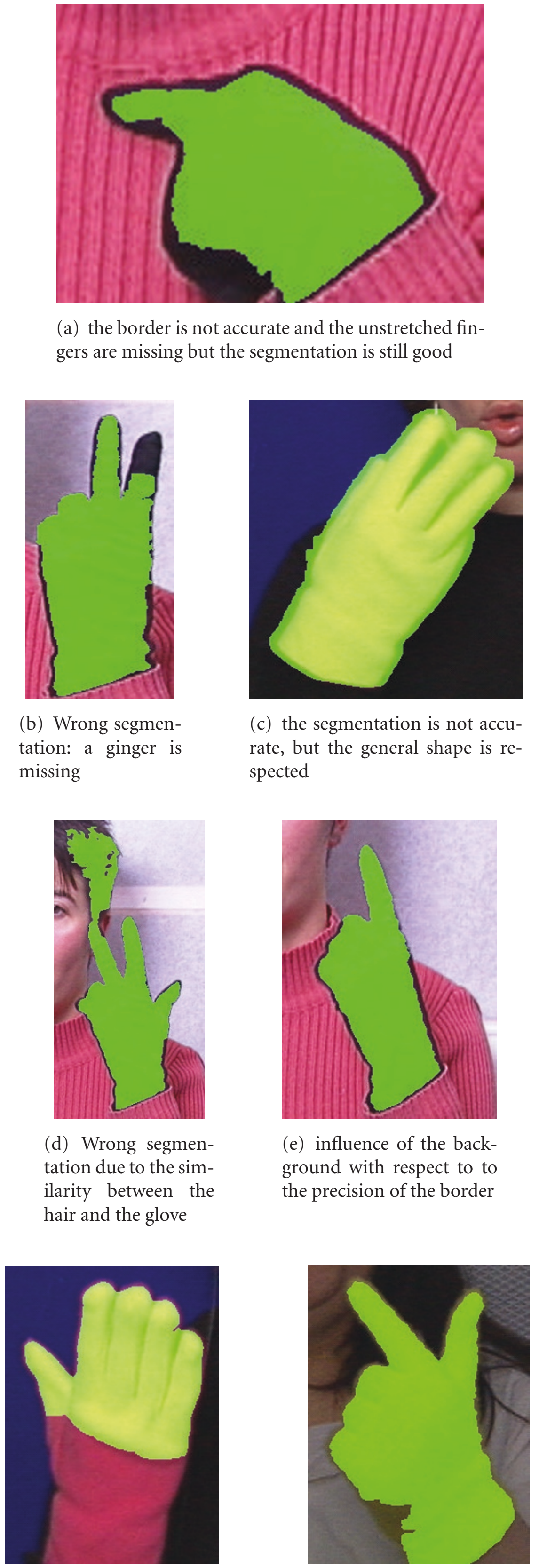
The acquisition is made at 25 images/second with a professional analogical camera of the highest quality. Then, the video is digitalized. Frames A and B are separated and each is used to recreate a complete image thanks to a mean interpolation process. Finally the video rate is 50 images/second, with the lowest quality, which is not a disturbance, as the original one is high enough to allow such a loss.

5.2. *Hand-segmentation evaluation*

For the evaluation of the hand segmentation process, we made the choice of using a qualitative approach defined in the following way: the hand is correctly segmented if the global shape is preserved in the sense that a human expert is able to recognize the right configuration. So segmentation "errors" such as small extension, border suppression, background fingers missing are not considered (Figures 20(a) and 20(c)). On the contrary, if a small mistake modifying the shape is observed, the segmentation is considered as mislead (Figures 20(b) and 20(d)). We do not consider an automatic and quantitative evaluation of the hand segmentation by comparing our results with a ground truth as our main goal is configuration recognition and not only hand segmentation.

The accuracy is defined as follows: for each video sequence, we count the proportion of images which are considered as correctly segmented (with respect to the previous conditions). With such a definition of the accuracy, the results for each sentence of the main corpus are the following: the lowest accuracy is 95.8% and the highest 100% with a mean of 99.4% on 1162 images. These results are equivalent with other gloves under the same conditions of acquisition: yellow glove (99.56%) and bright pink glove (98.98%), but the number of images for the test is less important. Concerning lower quality acquisition, it is really difficult to assess a score as it is far more depending on the conditions (lightening, glove, etc.). As an example, the result of Figure 5 is obtained with a portable digital camera in a classroom lit by the sun and with no particular constraint. Finally, the results are not corrupted by the addition of a salt and pepper noise of 60%. This result is not surprizing as the color is modeled by a Gaussian and the image smoothed by convolutions (Figure 21).

As we expect our segmentation to be accurate enough to provide precise descriptors to the recognition process, another mean to evaluate the efficiency of the segmentation is to consider the accuracy of the classification process: so long



(a) the border is not accurate and the unstretched fingers are missing but the segmentation is still good

(b) Wrong segmentation: a ginger is missing

(c) the segmentation is not accurate, but the general shape is respected

(d) Wrong segmentation due to the similarity between the hair and the glove

(e) influence of the background with respect to to the precision of the border

(f) Bad training consequence

(g) Bad training consequence

Figure 20: litigious segmentation illustration.

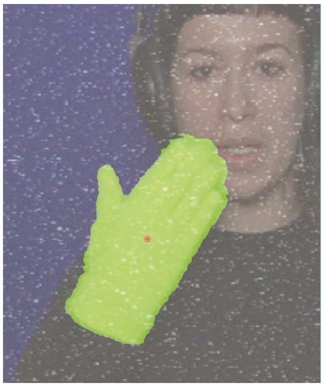


FIGURE 21: Segmentation of a noisy image (60% salt and pepper noise).

as this latter is efficient enough, it is not necessary to improve the segmentation.

Despite the number of steps (the error rates of which cumulate each other) involved in the extraction of the hand, our experiments have shown the good efficiency of this module.

We perform other acquisitions, with various webcam or intermediate quality digital cameras, and various gloves, in order to determine the robustness of the method with respect to the equipment. Of course, the quality of the results is somehow related to the quality of the camera. From our expertise, most of the errors are due to

(i) a too low shutter speed on the camera which leads to some images on which the fingers are blurred;

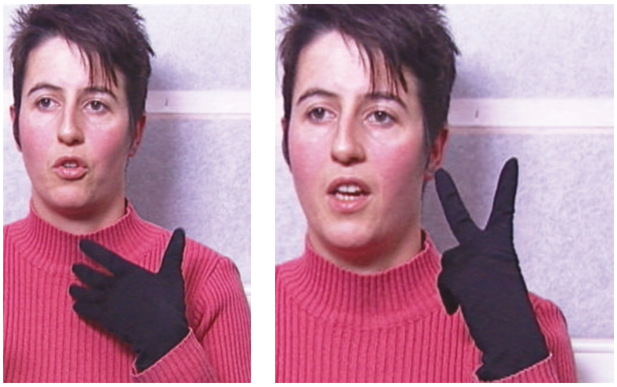
(ii) unwise choice of the color of the glove (it is more difficult to segment it when its color is closed to the color of an element of the background, to the skin color, or to the color of the hair);

(iii) bad color sensors, which prevent any color discrimination;

(iv) too dark gloves are difficult to segment as only the luminance of the pixel is meaningful. On the other hand, too light gloves are more sensitive to light variation and shadow effects (Figure 5). Intermediate colors are more efficient: luminance and chrominances are useful for the discrimination and shadows effects are dealt by the multiple thresholding;

(v) texture of the surrounding or background: because of the heavy use of convolution filter in the segmentation process, the border of the hand is not as accurate in case of textured area around it (Figure 20(e));

(vi) if the training step is not accurately performed, the segmentation results quality drastically decreases. We think an ergonomic study might be necessary to provide a convenient interface which ensures the coder who is not familiar with the program to have his/her glove well learned (Figures 20(f) and 20(g)). Of course, such a study is beyond the scope of scientific research and is of greater concern for a commercial application.



(a) The wrist is flexed: the other fingers appear shorter than they are with respect to the thumb

(b) The medium is not as parallel as the index with respect to the camera plan. Thus, the longest finger appears to be the index

FIGURE 22: Deformations of the hand which lead to a wrong determination of the pointing finger.

5.3. Pointed area and pointing finger evaluation

Practically, the definition of the pointed areas works efficiently. It is possible to evaluate the interest of the pointed areas with respect to their morphology for the coding task: it is actually interesting to check whether the chin is efficiently detected by the corresponding ellipse, (and for the mouth or the cheek bone as well) independently from the position of the pointing finger during the realization of a gesture. To do so, we applied our algorithm to 82 images of the BioID database [23]. It appears from this test that all the defined pointed areas give satisfactory results, but the area related to the chin remains less accurate than the others (due to the presence of beard, and the opening of the jaw). We do not provide accuracy rates, as there is no objective ground truth. On the contrary, we provide litigious cases (see Figure 23). The interesting point is that once this algorithm is coupled with its counter part on lip reading (a work lead by another team of THIMP), the mouth contour will be accurately segmented and it will improve mouth and chin detection as a side effect.

As shown on Figure 8 the coordinates of each feature are far more stable after being processed by the Kalman filter. Consequently, the pointed areas defined above are also more stable.

Concerning the pointing finger determination, the accuracy score is between 99% and 100% depending on the sentences of the corpus, (with a mean of 99.7%), so long as the hand perfectly remains in the acquisition plan. Otherwise, because of parallax distortions, the longest finger on the video is not the real one, as illustrated in Figure 22. As we expect the code to be correctly done, the images with parallax distortions are not taken into account in this evaluation.

5.4. Early reduction evaluation

PT definition. The PTs are only defined with respect to the change of configuration, and not with respect to the change

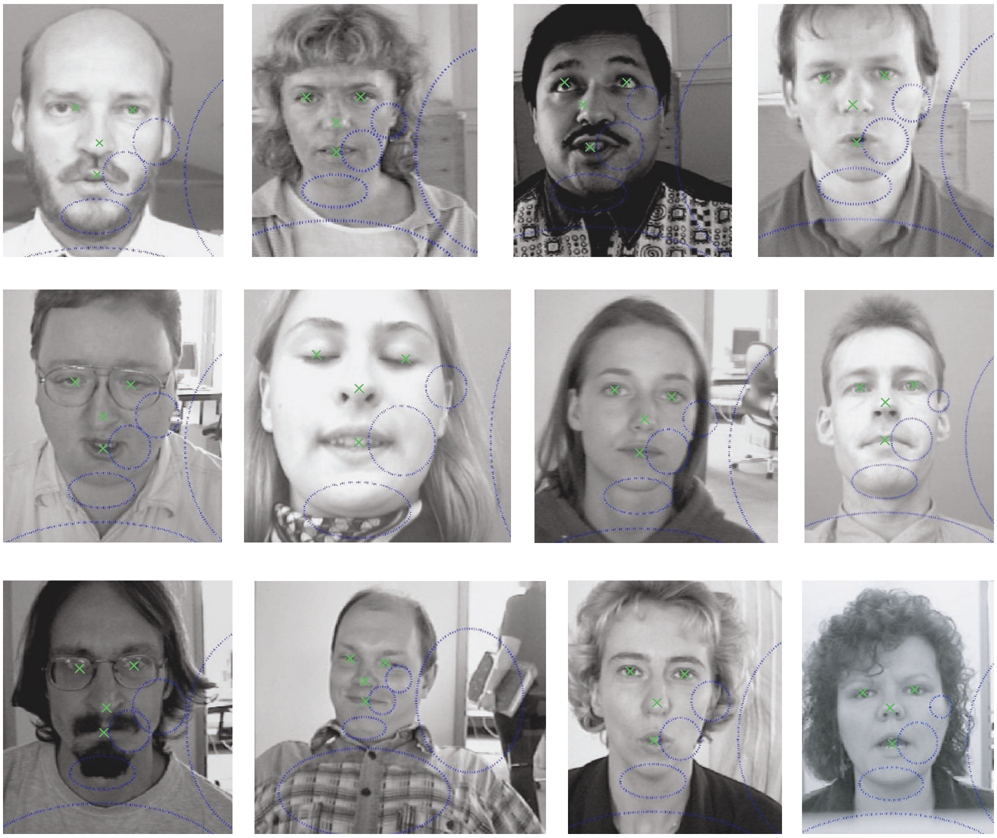


Figure 23: various litigious case from BioID database.

of location. It intuitively leads to a problem: when two consecutive gestures have the same configuration but different locations, a single PT should be detected and the other one should be potentially lost. In practice, there is a strong correlation on hand-shape deformation and global hand position, so it does not to occur too often: its proportion with respect to the other mistakes is not big enough to be quantified at the level of a phonemic evaluation of the system. On the contrary, it may lead to global inconsistencies for higher-level interpretation, such as complete sentence decoding. (Section 5.7).

The finger enhancer: The mask is manually set to correspond to the general pattern represented in (Figure 14(a)). In order to decide whether to use it or not, we simply qualitatively compare the output of the dedicated retina filter when it processes brute or enhanced data. As the results of the early reduction seem more adapted with than without the finger enhancer, it is kept with no longer optimization (although we concede that it could be optimized, there is no need for it at the moment).

KT selection. We have set a hierarchical definition of three types of kinetic targets. The last type of targets is associated to zones of stability (relative minimum in the motion) which are supposed to correspond to the full realization of gestures. The average proportion of images in a sequence which are KT1s is 40%. The proportion of the images in the sequence which are KT2 is between 25% and 30%, depending on the rhythm of coding, and this proportion is between 6% and 12% for KT3.

PTs are included in KT1 and KT2. From our experiments, this is true in more than 98% of the cases: the extremely rare errors are due to a very bad coding in which the corresponding gesture is not performed until its end, but completely smashed by the next gesture so that it appears as a transitive phenomenon. As these very few errors are not due to a failure of the algorithm, but to a bad coding, they are removed from the corpora for the evaluation. Concerning the error rate for KT3s, it is a bit more complicated, as there are two types of error (type 1 and type 2).

The rate of errors of type 1 (RT1%) is evaluated by an expert, and consequently is expert-dependent: it is based on the evaluation of stability by the visual perception of the expert. For each gesture the expert check that the selected set of images does not contain any motion; the configuration and the location must not change. From our experiments, RT1% ≈ 4%. Errors are most of the time due to an odd rhythm in the coding, which breaks the kinetic assumptions implicitly made in the way the motional energy is processed.

The rate of errors of type 2 (RT2%) is much higher just before the recognition step, but as they are dealt with later on, their number is not evaluated at this level. After the recognition step, these errors are dealt with, at the price of the addition of some TGs, which are processed until the recognition level. Then in addition to all the KT3 images, some

TGs images are processed (their number varies from zero to the number of KT3s). It leads to a total number of images, which is 13% to 18% of the total number of frame in the sequence. Of course, it is wiser to add some other images to prevent that any mistake has too large consequences. Practically, we found that the results do not improve if we process more than 25% of the images of the whole video. Eventually, this improvement in the robustness of the detection is correlated with more fake alarms, which finally annihilates the interest of using too many images. As a consequence, it validates a-posteriori the interest of the early reduction.

As long as no mistake is made at the recognition level, 100% of the errors of type 2 are properly adressed by considering the appropriate TGs. Hence, RT2% is directly related to the error rate of the recognition module (which is developed in the next section).

Concerning the identification of PTs by KT3s, the accuracy is really high from a gesture point of view, as we reach 89% to 100% depending on the sentences, with a mean of 93%. Nonetheless, these results must be cautiously interpreted, as they do not deal with the synchronization with "the pointing of the location gesture," and as the ground truth is specified for each PT. Hence, the results, although they are an important improvement, are still far from complete sentences recognition.

5.5. Classification evaluation

We use the LIBSVM library [22] for the implementation of the SVM algorithm. Thanks to a tenfold cross-validation, the classification parameters described above are set: the cost parameter is set to 100 000 and termination criterion to 0.001. The sigmoid kernel is

$$\begin{aligned}\mathrm{Ker}_{\gamma,R}(u,v) &= \tanh(\gamma\cdot u^T\cdot v + R)\\ &\text{with } \gamma = 0.001, R = -0.25.\end{aligned} \tag{13}$$

To evaluate the methodology (attributes and classifier selection, classification parameter tuning), we perform the following experiment: a hand-shape database is derived from our main dataset of FCS videos. The transition shapes are eliminated manually and the remaining shapes are labelled and stored in the database as binary images representing the nine configurations (Figure 18).

The training and test sets of the database are formed such that there is no strict correlation between them. Thus, two different corpuses are used in which a single coder is performing two completely different sets of sentences using Cued Speech. The respective distributions of the two corpora are given in Table 1. The statistical distribution of the configurations is not balanced at all within each corpus. The reason of such a distribution is related to the linguistics of cued speech.

For each image, the real labels are known. Thus, we use the following definition of the accuracy to evaluate the performance of the classifier:

$$\text{Accuracy} = 100\cdot\frac{\text{Number Of Well Classified Items}}{\text{Total Number Of Items}}. \tag{14}$$

Table 1: Details of the database.

Hand Shape	Training set	Test set
0	37	12
1	94	47
2	64	27
3	84	36
4	72	34
5	193	59
6	80	46
7	20	7
8	35	23
Total	679	291

On the test set, we obtain an accuracy of 90.7%. Most of the mistakes are due to the following.

(i) A strong overlap of the classes in the descriptor space: some rather different images have closed description and consequently, the Hu invariants, though efficient on really discriminated classes of hand shapes, are not powerful enough.

(ii) Classes 3 and4 are difficult to separate, because of the similarity of the configurations, as well as for classes 1 and 2 and for classes 6 and 7, when the fingers are kept grouped.

(iii) The descriptors are also not very successful for classes 3 and 7; it is due to the similarity between a mirror image of configuration 3 and an image of configuration 7 when both of them are performed with the fingers too much separated. The detection of the thumb, which is an easier finger to detect, would help to make the difference.

(iv) The fusion of the binary SVMs is not really efficient: to our point, 3% to 5% of the mistakes are due to the one-versus-one procedure. The final result is mistaken whereas the separated SVMs give consistent results.

In this experiment, both learning and test are made on a single corpus user. We nonetheless consider some small experiments to have an idea of the manner in which these results can be generalized to multiple coders: within our database, few acquisitions are made with another glove which is not as adapted as the main one (see Section 5.1). Consequently, the shape of the hand looks rather different. We used the learning made on the main glove in the database to classify the other few images with the "bad" glove. Consequently, we submit unknown glove shapes to the classification algorithm. We also capture few hand shape performed by non-cued-speech coder (consequently the configurations are performed out of coding context) in order to have a hint on the variability of the hands. The same classification process is applied with the same previous learning. It appears that the accuracy drops only from 1 to 3 points, depending on the corpora.

5.6. Camera calibration and computation cost

In terms of computation, we are now restricted to MatLab/C/C++ code (with no micro-processor optimizations) and Intel Pentium workstations running under Microsoft Windows, so the real time is not reachable yet. However, a processing rate of 5 image/s (image size: 480 x 360 pixels) is promising for future real time implementation on dedicated hardware. From our test, a real-time version of the algorithm needs to cope with rates higher than 40 image/sec: an acquisition frame rate of 50 image/s is really sufficient for no PT being lost by the subsampling, even in case of a fast coder. On the contrary, a frame rate of 25 and 30 image/s is not enough, some PTs are missing. Finally, the focus of the camera is a real issue, as it is required to have the face and the hand in a single picture, while having a high enough resolution to segment the lip (for the lip-reading task carried out by another team, as we do not expect to use several cameras in THIMP). So far, only the professional camera of the main corpus of data fulfils these requirements.

5.7. Sentence recognition

All the elements of our architecture have been described so far, and the whole system has to be evaluated. As our purpose is to decode sentence as a whole, let us select for each sentence from the corpus a lattice of potential phonemes, and define the overall accuracy OvAcc as

$$\text{OvAcc} = 100 \cdot \frac{\begin{array}{c}\text{Number Of Sentences completly included}\\ \text{in the proposed lattice of phonemes}\end{array}}{\text{Number Of Sentences}}. \tag{15}$$

This definition of the overall accuracy is very restrictive as a single omission of a gesture in a sentence is sufficient to consider the whole sentence as false. Of course, a rate on the number of correctly recognized gestures would lead to higher recognition rates. But as our final purpose is natural coding recognition (that is sentence decoding), we consider that it is better to evaluate the whole process with respect to this goal, even if the global resulting accuracy score is not as high.

In practice, the selection of PTs is very efficient, as our experiments showed it. But

(i) few mistakes remain;

(ii) for the PTs which have been correctly detected, there are several images that potentially correspond, and the *early reduction* does not always select the same one as the expert who defined the ground truth. Hence, we find that the set of PTs automatically and correctly detected via the *early reduction* has a bigger variance that the ground truth set.

Consequently, the accuracy of configuration recognition is slightly lower when considered in the whole process rather than isolated. Secondly, the same problem occurs in bigger proportion for the location recognition, for the simple reason that the PTs have not been designed to detect PTs for the location as precisely as to detect the PTs for the configuration. Finally, the synchronization problems between the two components (hand configuration and location) of the hand gesture (raised by [3]) are not yet addressed in the process we presented.

Table 2: Summary of the results.

Algorithm	Qualitative evaluation	Accuracy rate
Segmentation	Good results within the acquisition conditions specified	99.4%
Pointing area	The chin pointing area is less robust than the others. Definition is improved by Kalman filtering	—
Pointing finger	The hand must remain in the acquisition plan	99.7%
PT selection	—	96%
Configuration classification	—	90.7%
Camera calibration	Professional camera with frame rate > 40 image/s is required	—
Sentence recognition	There are synchronization problems which are not dealt yet	< 50%

For these three reasons, the overall accuracy OvAcc on the lattice of phonemes is far lower than acceptable rates. Hence, we have evaluated it only on 20 sentences randomly chosen among the part of the corpus which has not been used to extract the training set for the configuration classification. From our test, 40% ≤ OvAcc ≤ 50%, depending on the evaluations. This does not question our methodology and algorithms (specially the *early reduction*), as, taken individually, they all provide good or very good results (these results are summarized in Table 2) .

Moreover, despite giving still insufficient results on global sentence recognition, our method has a very powerful advantage: its use is not conditioned to any subset of language. Hence, the results which are announced are likely to be easily generalized to more complex sentences or even to natural dialogue. Classically, the systems described in the literature (see Section 1) propose accurate results on very restrictive cases for which any extension is bounded to reduce the performance. Hence, our method is really new and its performances need to be estimated with respect to this generalization capability.

Nonetheless, the results only points out the lack of global fusion or integration process. For the moment, such a process gathers all our efforts and is the main aspect of our future work. From our prime analysis, this integration module is likely to be far less complicated than expected thanks to the *early reduction*. Finally, for a perfect sentence-by-sentence interpretation such an integration module might not be sufficient and a language model might be necessary. These aspects will be the topic of our future works, but also that of the other teams of THIMP.

6. CONCLUSION

In this paper, we proposed a first complete automatic cued speech gesture recognition method. From an image-by-image processing point of view, the algorithms involved are rather classical (segmentation and classification steps), but from a video processing point of view, we provided an original method called the *early reduction*. From our experiments, all the proposed algorithms give satisfactory or very satisfactory results, at a gesture level. On the contrary, their integration into a global system leads to results at the level of complete sentence interpretation, which are not yet as satisfactory. This is due to the lack of a last module the purpose of which is to fuse the information from the various classifiers and the *Early Reduction*. Consequently, our future works will be focused on such a module.

ACKNOWLEDGMENTS

This work is supported by SIMILAR, European Network of Excellence. This work takes its origin in several cooperations: (1) Oya Aran and Lale Akarun, from Bogazici University, for the classification purpose, (2) Alexandre Benoit, from GIPSA, for his knowledge of the visual system, (3) Alexandra Urankar from France Telecom R&D who coded the segmentation algorithm, (4) Sebastien Roux and Franck Mamalet from France Telecom, for the implementation of the face and features-detection algorithm, (5) all the teams involved in THIMP.

REFERENCES

[1] R. O. Cornett, "Cued speech," *American Annals of the Deaf*, vol. 112, pp. 3–13, 1967.

[2] D. Beautemps, "Telephone for hearing impaired," French RNTS Report, 2005, Reseau National des Technologies pour la Santé.

[3] http://www.lis.inpg.fr/pages_perso/caplier/english/geste.html .en/geste_1.html.en.html.

[4] A. Caplier, L. Bonnaud, S. Malassiotis, and M. Strintzis, "Comparison of 2D and 3D analysis for automated cued speech gesture recognition," in *Proceedings of the 9th International Workshop on Speech and Computer (SPECOM '04)*, Saint-Petersburg, Russia, September 2004.

[5] V. Attina, D. Beautemps, M.-A. Cathiard, and M. Odisio, "A pilot study of temporal organization in cued speech production of French syllables: rules for a cued speech synthesizer," *Speech Communication*, vol. 44, no. 1–4, pp. 197–214, 2004.

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

[7] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[8] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[9] J. Bilmes, "What HMMs can do," Tech. Rep. UWEETR-2002-2003, University of Washington, Department Of EE, Seattle, Wash, USA, 2002.

[10] T. Burger, A. Benoit, and A. Caplier, "Extracting static hand gestures in dynamic context," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '06)*, pp. 2081–2084, Atlanta, Ga, USA, October 2006.

[11] B. Dorner and E. Hagen, "Towards an American sign language interface," *Artificial Intelligence Review*, vol. 8, no. 2-3, pp. 235–253, 1994.

[12] T. Burger, A. Caplier, and S. Mancini, "Cued speech hand gestures recognition tool," in *Proceedings of the 13th European Signal Processing Conference (EUSIPCO '05)*, Antalya, Turkey, September 2005.

[13] C. Garcia and M. Delakis, "Convolutional face finder: a neural architecture for fast and robust face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1408–1423, 2004.

[14] S. Duffner and C. Garcia, "A hierarchical approach for precise facial feature detection," in *Proceedings of Compression et Représentation des Signaux Audiovisuels (CORESA '05)*, Rennes, France, November 2005.

[15] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43–77, 1994.

[16] M. Irani, B. Rousso, and S. Peleg, "Computing occluding and transparent motions," *International Journal of Computer Vision*, vol. 12, no. 1, pp. 5–16, 1994.

[17] A. Benoit and A. Caplier, "Motion estimator inspired from biological model for head motion interpretation," in *Proceedings of the 6th European Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS '05)*, Montreux, Switzerland, April 2005.

[18] S. Wang, J. Zhang, Y. Wang, J. Zhang, and B. Li, "Simplest operator based edge detection of binary image," in *Proceedings of the International Computer Congress on Wavelet Analysis and Its Applications, and Active Media Technology*, vol. 1, pp. 51–56, Chongqing, China, May 2004.

[19] T. Morris and O. S. Elshehry, "Hand segmentation from live video," in *Proceedings of the International Conference on Imaging Science Systems and Technology (CISST '02)*, UMIST, Manchester, UK, August 2002.

[20] D. Zhang and G. Lu, "Evaluation of MPEG-7 shape descriptors against other shape descriptors," *Multimedia Systems*, vol. 9, no. 1, pp. 15–30, 2003.

[21] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[22] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2001.

[23] http://www.bioid.com/.