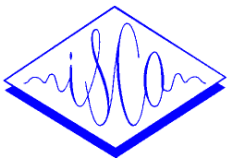

AVSP 2003 – International Conference
on Audio-Visual Speech Processing
St. Jorioz, France
September 4-7, 2003

Low Resource Lip Finding and Tracking Algorithm for Embedded Devices

Jesús F. Guitarte Pérez [1] , *Klaus Lukas* [1] , *and Alejandro F. Frangi* [2]
[1] Siemens AG, Corporate Technology, Otto-Hahn-Ring 6, Munich, Germany.
[2] Computer Vision Group, Aragon Institute of Engineering Research,
University of Zaragoza, María de Luna 1, Zaragoza, Spain.
jguitarte@yahoo.es, lukas@siemens.com, afrangi@unizar.es

Abstract

One of the best challenges in Lip Reading is to apply this technology in embedded devices. In current solutions the high use of resources, especially in reference to visual processing, makes the implementation and integration into a small device very difficult.

In this article a new and efficient algorithm for detection and tracking of lips is presented. Lip Finding and Tracking are customary first steps in visual processing for Lip Reading. In our approach Lips are found among a small number of blobs, which should fulfill geometric constraints. The proposed algorithm runs on an ARM920T embedded device using on average less than 4 MHz[1] (2,7% of CPU load). This algorithm shows promising results in a realistic environment accomplishing successful lip finding and tracking in 94.2% of more than 4900 image frames.

1. Introduction

Automatic Speech Recognition (ASR) will play an important role in future embedded devices, such as mobile phones, PDAs and also in car environments.

One of the most important problems of ASR is the degradation of the acoustic signal, which usually induces a significant recognition rate decrease. Several techniques, such as, for example, Noise Compensation based on acoustic signal processing [1,2] have been developed to improve the robustness of ASR when the acoustic signal is corrupted. Another approach uses visual information as it will be available in future devices, e.g. by using the camera of 3G mobile devices. This technique is called "Lip Reading" and it exploits the additional information that can be found in the lips movement during speech [3]. This visual information can be combined with the acoustic information in order to improve the recognition rate [4]. Compared to conventional acoustic recognition, the combined system can achieve a 55% error rate reduction for various signal/noise conditions [5]. This technique can also be used in combination with Noise Compensation Algorithms [6].

Lip Reading systems have been developed in several laboratories and research projects but without taking special care on embedment restrictions. In this article we describe an embeddable algorithm for the first stage of most Lip Reading systems: Lip Finding and Tracking. We propose a system that finds the position of the lips: Detection and Tracking of the Region of Interest (ROI). Several approaches can be found in the literature to this end; some of them are very robust and can work under complex lightning conditions. However, they usually require many computing and storage resources like, e.g., those based on large Neural Networks [5]. Other solutions work in Real-Time but only in desktop workstations where resource availability is less restrictive than in a small device. Yang et al. [7] proposed a top-down approach, which works by first finding the face using color information. Subsequently, for every frame, a set of facial features is extracted (eyes, nostrils and lip corners). In contrast, our approach extracts a smaller set of features that do not require a top-down search. The eyebrows and lips are searched only when there is no information about the position of the lip in the last frame. Otherwise, only a lip tracking is performed. Real-Time implementation constrains of small embedded devices must be taken into account. Kaucic et al. [8] presented an unadorned, accurate Lip Tracking algorithm. It works in a desktop workstation (200MHz) and it uses in an efficient way B-splines and Fischer linear discriminant analysis. This system solves the tracking of the lips but it does not take into account the lip search, so it assumes a first known position of the lips and it performs the tracking.

The aim of our paper is to describe a system that can automatically perform the Lip Finding and Tracking. The system must be able to work without special light conditions as well as without any kind of reflected markers or special make up placed on speaker's lips.

The applications of a Lip Finding and Tracking system are not only related to Lip Reading (ASR). It is proven that people catch the attention of the desired speaker by looking at him, and in the same way a system that knows when the speaker asks for its attention can be imagined. Our approach is very appropriate for this task because lips are found only when the speaker looks at the camera (a frontal view is required). An automatic "push to talk" system could be implemented; the conventional speech recognition system is activated only when the speaker wants to communicate with the device.

This paper is organized as follows. In Section 2, the Lip Finding and Tracking System is described. In Section 3, the requirements for implementing the system in a commercial

[1] ARM920T: 150 MHz, 16Kbyte bi-directional cache. External Memory Access Speed: 150 *nsec*. for non sequential and 10 *nsec.* for sequential access.

embedded system are introduced. Section 4 presents the experimental results of the evaluation of the system. Finally, Section 5 provides some conclusions.

2. Lip Finding and Tracking System

In order to describe properly our algorithm we have to make a reference to three different functions: Lip_Finding, Lip_Tracking and finally the Features_Extraction.

A comment regarding the Lip Finding and the Lip Tracking algorithms should be made. Lip Finding is applied when no previous information of the lip position is available. This happens in the first frame of a sequence or whenever the lips cannot be correctly located in the previous frame. Lip Tracking proceeds when knowledge on the position of the lips in the previous frame is available. This information can be used to update the lips' coordinates by inspecting a region close to the last position rather than in the whole image. It has been observed that given approx. 15 frames per second or higher rates, the location of the lips cannot differ too much from one frame to the next one for the application scenarios we are dealing with. For example, in a mobile phone application the device is held by the speaker or in a car environment the relative position of the speaker does not change too much in 1/15 sec. Furthermore, Lip_Tracking is more reliable and requires less resource than Lip_Finding.

Finally, Feature_Extraction receives the coordinates of the possible lips given by Lip_Finding or Lip_Tracking and tries to extract the descriptors of the lips in that region. The matching of such descriptors with the typical lips descriptors gives a criterion to verify whether the lips have been found or not. This three algorithms will be profusely described in the next paragraphs.

2.1 Lip Finding Algorithm

This algorithm is based on a geometric model of the face. Structures of pixels are evaluated in order to know if their relative positions match a simplified prior model of the face. In particular, this model accounts only for the relationships between location of the eyebrow(s) and the mouth.

The algorithm starts with a Directional Filtering, where eyebrows and lips are extracted. After Segmentation the system stops working with pixel based processing and it resumes operating with structures of pixels, called blobs. Finally the Search and Matching Process will be accomplished using blob descriptors.

2.1.1 Directional Filtering

In order to save memory no color information is used; only the luminance (Y) component from the YUV color space will be taken into account (see Figure 1.a This grayscale image is filtered by a horizontal filter. An evaluation of edge detection algorithms yields a simplified version of the Sobel operator to be the best choice, only the horizontal component of the simplified operator is used:

$$G_x(n,m) = Y(n-1,m) - Y(n+1,m) \tag{1}$$

After filtering, image thresholding takes place, where the threshold could be fixed or adapted to the variance of the filtered image. A binary image is obtained where all pixels which belongs to a contour with a horizontal component are set to "one" (white in Figure 1.b.)

To obtain a better contrast, especially in bad light conditions, a histogram equalisation can be achieved before filtering. A faster execution can be run by using the previous frame histogram information for the equalisation of the current frame.

In the same image scanning a run-length coding (RLC) [9] is created in order to obtain a faster segmentation.

2.1.2 Segmentation

A segmented structure is obtained from the run-length coding. A blob is defined as a group of pixels connected according to a specific neighborhood relationship and sharing a common characteristic [10]. In this case the common characteristic is to belong to the same horizontal contour.

Each blob is described only by the area and the coordinates of its center. Blobs are filtered according to their area, therefore very small or very big blobs will be disregarded. This will only imply that our algorithm will work for a limited range of distances between the camera and speaker. In our applications either the speaker holds the device in his hands or the camera is located at a fixed distance on the dashboard (car environment). The remainder blobs after filtering are showed in Figure 1.c.

2.1.3 Search and Matching Process

The search process is performed by only taking into account those blobs whose positions make them likely to belong to parts of the face. Only approximately 10-25 blobs per frame are left to take part in the search. The algorithm takes the set of blobs that best matches the prior model of a face. The center of the ideal mouth:

$$\underline{C}_{id} = f\{\underline{C}(e_r), \underline{C}(e_l)\} \tag{2}$$

is computed from the centers of mass of the detected eyebrows. This location is subsequently compared to the position of the nearest blob $\underline{C}(m)$, and the distance between both centers is considered as a measure of the resemblance with the face model. The set of three blobs that minimizes this distance is considered as the detected face. When this measure exceeds a certain value the algorithm assumes that no face has been found. This search process is shown in Figure 1.d.

Let $\underline{C}(e_r)$, $\underline{C}(e_l)$, and $\underline{C}(m)$ be the coordinates of the blob centers representing the right eyebrow, the left eyebrow, and the mouth, respectively. The objective is to find the three blobs e_r, e_l, m that minimize the distance:

$$dist\{\underline{C}_{id}, \underline{C}(m)\}_{e_r,e_l,m} \tag{3}$$

where:

$$\underline{C}_{id} = \begin{cases} C_{id\_x} = \max\{C_x(e_r), C_x(e_l)\} - 0.5 * abs\{C_x(e_r) - C_x(e_l)\} \\ \qquad + K * \{C_y(e_r) - C_y(e_l)\} \\ C_{id\_y} = \max\{C_y(e_r), C_y(e_l)\} - 0.5 * abs\{C_y(e_r) - C_y(e_l)\} \\ \qquad + K * \{C_x(e_r) - C_x(e_l)\} \end{cases} \tag{4}$$

where K is a geometrical aspect ratio obtained empirically from a large face database. It is defined, as shown in Figure 1.d, as K = b/a = 1.2.

Fig.1.a

Fig. 1.b

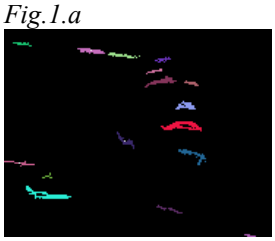

Fig. 1.c

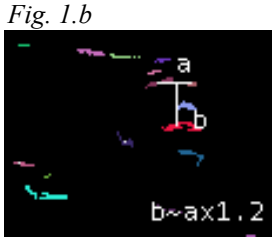

Fig. 1.d

Figure1: Stages of the Lip Finding algorithm. a) Grayscale image, b) horizontal filtering and thresholding, c) Segments, and d) Search process.

Problems could arise with people with very bushy eyebrows or with glasses. In these cases only one single segment would be recognized as an eyebrow. This situation was taken into account and if there are no segments that satisfy the condition of the face according to the first search algorithm, a second model with a single large eyebrow is assumed and the process is repeated.

2.2. Lip Tracking Algorithm

Lip Tracking is applied only when in the last image the lips were found. In this case we rely on the hypothesis that the position of the lips will not be very different between one frame and the next one. Also, a movement vector can be used to give a better approximation between frames. Lips will be searched in an area that is 10% larger than the region where the lips were located in the previous frame, see Figure 2. The searching process consists on extracting the features of the lips. It will be explained in the next paragraph.

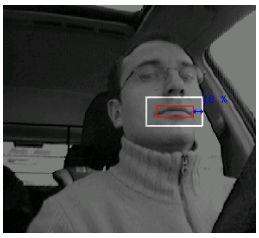

Fig. 2: Lip Tracking

2.3. Features Extraction

This part of the algorithm receives the position where the lips are supposed to be located from Lip Finding or Lip Tracking. It finds the features that describe the lips. If the typical features of the lips are found, the verification is completed and the lip coordinates are updated.
In the Feature Extraction implemented in this article, the upper and lower lip contours are sought. First, the upper lip contour is obtained with a gradient filter that highlights bright-to-dark intensity changes (from top to bottom in the image). Then another dark-to-bright filter is applied to obtain the lower contour of the lips. If both segments have the properties of a lip, the verification will be completed, and the lip position is updated. In this implementation, we know that the lips are correctly located using the relative position between the two lips and their geometrical properties.

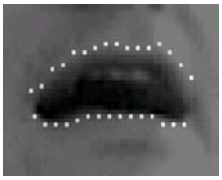

Fig. 3: Feature Extraction

We would like to point out that the Features Extraction used in this implementation can be improved by using other kind of algorithms. The used features are good enough for the tracking but they do not have enough information and are also not robust enough to allow the recognition of the visemes, Lip Reading. Although for the aim of this article -find the ROI- they are appropriate other Feature Extraction algorithms are being investigated. Some studies have being carried out with the intention of using ASM Active Shape Model [11], particularly there are important works oriented to Lip Reading [12]. The implementation of ASM was always difficult to run in Real-Time. Providing an approximation of the region of interest (ROI) the required computational time will be smaller because the ASM algorithm should now fit the lips only in a small region of the image, not in the whole image. It can be said that a coarse Rigid Registration (position, orientation and scaling) of the lips can be obtained by using our algorithm. A Real-Time implementation of the ASM for the lips was built by using our Lip Finding and Tracking proposal.

3. Requirements

Since the algorithm is intended for integration in an embedded device, it is important to restrict the resources that can be consumed. The Lip Finding algorithm saves an important amount of resources by performing the search only between a small amount of blobs as indicated in Section 2.1.2. Extra savings have been accomplished since the Lip Tracking process can be applied most of the time in comparison to Lip Finding. The former process searches in a small region of approximately 5% of the area of an image.

The fulfillment of the requirements is tested with an emulator for the ARM920T μC, an exemplary microprocessor suitable for Third Generation of Mobile Devices (UMTS). Table 1 lists the algorithm demands

	CPU (MHz)
Lip Finding	40 MHz
Lip Tracking	1.8 MHz

Table 1: Demands of Lip Finding and Tracking.

In the requirements tests a frame rate of 15 f.p.s. has been used. It must be taken into account that, as long as the lips are being found, only Lip Tracking is applied, so Lip Finding is used only when the lips were not found in the previous image and they must be searched within the whole image.

In the sequences used to test the system the percentage of images where Lip Finding was applied was only 5%, so we would obtain a mean CPU use of 4 MHz[1] (2,7% of CPU load). The Code Memory consumption is about 9 Kbytes without consideration of the linked C standard libraries that may be shared by several software modules and therefore do not increase the memory consumption.

These results were obtained without any kind of platform-specific or assembler optimization of the algorithm. Even in this situation the current software module is compliant with the available resources in an embedded device

4. Results

The algorithm was evaluated verifying results by the visual inspection. A set of 33 speakers of both sexes, with ages between 20 and 60 years, with different skin colors and with different grades of facial hair have tested the system in sessions of 10 s each (150 frames each speaker). No special light conditions have been used and no reflected markers or special make up were placed on the speaker's lips. The distance between the camera and the speaker was between 10 and 70 cm, and people were asked to look at the camera. From the total of 4950 images using the Lip Finding and Tracking algorithm the error rate was 5.8%. We have also obtained the results using only Lip Finding without Tracking, so without using the information about the previous frame; in this situation the error rate raises up to 27.4%, see Figure 4.

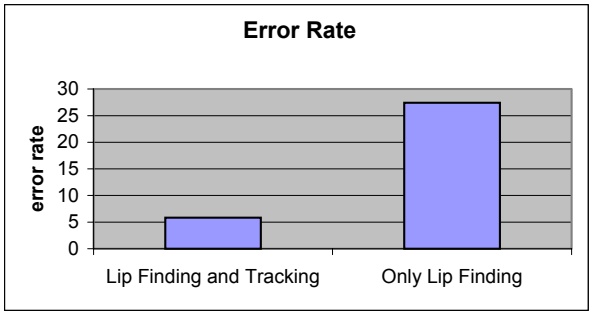

Figure 4: Error rate

In addition, to give a better criterion of the generalization ability of the results this algorithm is applied to different users, it is important to know how the erroneous frames are distributed over the different speakers. Figure 5 shows that the 78.8% of the speakers have an error rate smaller than 5%.

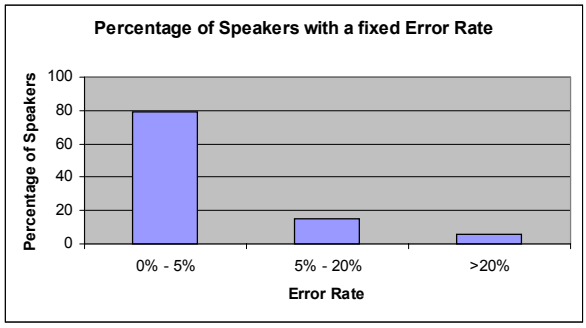

Figure 5: Percentage of speakers vs. error rate

We have classified the error frames on one hand as false alarm, when our system has found a structure that does not match the lips, and on the other hand as *not found*, when the system knows that it cannot find the lips in the image and gives therefore no output. The percentage of errors classified like false alarm for the Lip Finding algorithm is 39.2 % and this value is increased to 47.2%, see Figure 6, when Lip Tracking is applied, since some erroneous frames are propagated with the tracking system.

For Lip Reading systems it will be very interesting to have a small false alarm rate because that means a reduction on the video noise. In the same way as we have acoustic noise we will have visual noise when the image information -region of interest given by our system- is not the right one: the lips. So, as long as we can provide a small false alarm rate we will be able to repeal the visual noise. An improved version of the Features Extraction is being investigated. Providing a better description of the lips and therefore giving an improved criterion to discriminate between lips and other structure the false alarm rate will be reduced.

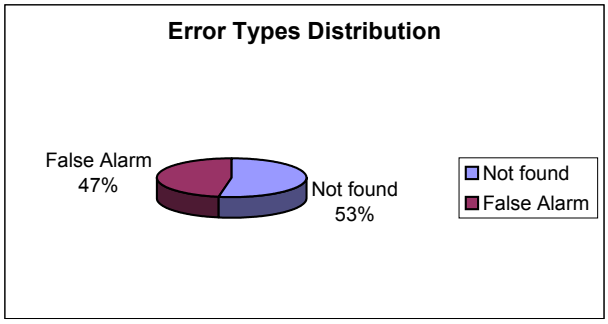

Figure 6: Error type distribution for Lip Finding and Tracking

The bursts of errors have been checked. A burst of errors happens either when by chance several errors are consecutively found or when the Feature Extraction criterion fails on identifying a false structure. In this situation the wrong structure will be tracked and a burst of false alarms will be caused. The mean length of the burst of errors has been measured in our test set. When the tracking system is used the mean length value is 5.05 frames and it decreases to 3,67 frames when the tracking is not applied. As it was expected the bursts of errors are deeper when tracking is applied, because of false structure tracking. But assuming that the system works with 15 frames every second, the errors could be interpolated in many cases.

In figure 7 we can see several examples of the performance of Lip Finding algorithm, the right column shows the different horizontal regions taken into account in order to look for the mouth structure and on the left column the Lip Finding result is shown. In figure 7 a. the nose is found instead of the mouth, which is a common kind of error, due to the different geometry of the faces.

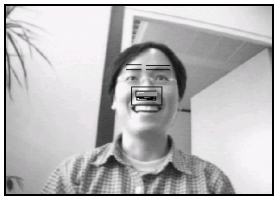

Fig.7.a

Fig.7.b

Fig.7.c

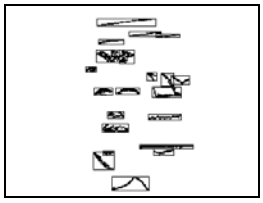

Fig7.d

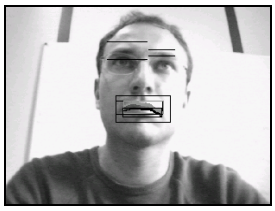

Fig.7.e

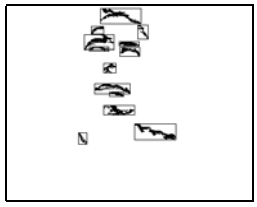

Fig.7.f

Figure7: examples of performance for different people, on the right column the different segments used for Lip Finding are shown.

5. Conclusion

An efficient algorithm to find the lips has been presented. It has good performance in a realistic environment, with a 5.8% error rate. Since it consumes very few resources it can be implemented in an embedded device. The error rate is lower when tracking is used, but if we want to reduce the false alarm and the burst of errors length a better Feature Extraction must be used in order to guarantee a good tracking. It is also necessary for Lip Reading to provide a set of characteristics useful for recognition; ASM is one of the best chances. We can say that there is a synergy between ASM and Lip Finding/Tracking. ASM will improve the robustness of Lip Finding/Tracking providing a more reliable verification criterion, and the Lip Finding/Tracking will allow a Real-Time implementation of the ASM.

False alarm and error rate could also be reduced by using first a color classification algorithm. The idea is to apply the algorithm explained in this paper only in the regions where the color is similar to that of the skin. Some pilot tests were already performed along this lines and important improvements were found especially in backgrounds full of horizontal structures like plants' leaves.

6. Acknowledgement

The authors wish to express special thanks to Markus Simon, Ewald Frensch, Nikos Paragios, Visvanathan Ramesh and Eduardo Lleida for their good advises and interesting discussions. Siemens AG has supported the work of J. F. Guitarte under a PhD. contract. The work of A. F. Frangi was supported by Grants TIC2002-04495-C02 and FIT-070000-2002-935 of Spanish Ministry of Science and Technology and a Ramón y Cajal Research Fellowship.

7. References

[1] R. Singh, R. M. Stern, and B. Raj, "Signal and Feature Compensation Methods for Robust Speech Recognition," *CRC Press LLC., pp. 219-243, 2002.*

[2] ETSI ES 202 050 V.1.1.1, "Speech Processing Transmission and Quality aspects (STQ); Distributed Speech Recognition; Advanced front-end feature extraction algorithm; Compression algorithms," *European Telecommunications Standards Institute, October 2002.*

[3] McGurk H., and MacDonald J., "Hearing lips and seeing voices," *Nature, 264, pp. 746-748, December 1976.*

[4] Petajan E. D., "Automatic lipreading to enhance speech recognition," *Proc. IEEE Computer Vision and Pattern Recognition, pp. 44-47, 1985.*

[5] U. Meier, R. Stiefelhagen, J. Yang, and A. Waibel, "Toward Unrestricted Lip Reading," *International Journal of pattern Recognition and Artificial Intelligence, Vol. 14, No. 5, pp. 571-785, 2000.*

[6] S. J. Cox, I. A. Matthews, and J. A. Bangham, "Combining noise compensation with visual information in speech recognition," *Proc. ESCA/ESCOP Audio-Visual Speech Processing, pp. 53-56, 1997*.

[7] J. Yang, R. Stiefelhagen, U. Meier and A. Waibel, "Real-time Face and Facial Feature Tracking and Applications," *Proc. Audio-Visual Speech Processing, pp. 79-84, 1998.*

[8] R. Kaucic and A. Blake, "Accurate, Real-Time, Unadorned Lip Tracking," *Proc. International Conference on Computer Vision, pp. 370-375, 1998.*

[9] S. W. Smith, "The Scientist and Engineer's Guide to Digital Signal Processing," *California Technical Publishing, 1997.*

[10] R. C. Gonzalez, and R. E. Woods, "Digital image processing," Prentice Hall, 2001.

[11] T. F. Cootes, C. J Taylor, D. H. Cooper, and J. Graham, "Active Shape Models – Their training and application," *Comp. Vis. Image Understand., vol. 61. no. 1, pp. 38-59, 1995*.

[12] J. Luettin, and N. A. Thacker, "Speechreading using probabilistic models,Computer Vision and Image Understanding," *Vol. 65, no. 2, pp.163-178, 1997.*