A contribution to mouth structure segmentation in images towards automatic mouth gesture recognition

Juan Bernardo Gómez-Mendoza

Thèse

A contribution
to mouth structure segmentation in images aimed towards automatic mouth gesture recognition

Présentée devant
L'institut national des sciences appliquées de Lyon

Pour obtenir
Le grade de docteur

École doctorale
École doctorale electronique, electrotechnique, automatique (EEA)

Par

Juan Bernardo Gómez-Mendoza
(Ingénieur)

Soutenúe le 15 mai 2012 devant la Commission d'examen

Jury MM.

P. BOLON	Professeur (Polytech Annecy-Chambéry)
C-A. PARRA-RODRÍGUEZ	Professeur (Universidad Javeriana)
M. ORKISZ	Professeur (Université Lyon I)
J-W. BRANCH-BEDOYA	Professeur (Universidad Nacional de Colombia)
H-T. REDARCE	Professeur (INSA de Lyon)
F-A. PRIETO-ORTIZ	Professeur (Universidad Nacional de Colombia)

To my family and friends.

Abstract

This document presents a series of elements for approaching the task of segmenting mouth structures in facial images, particularly focused in frames from video sequences. Each stage is treated separately in different Chapters, starting from image pre-processing and going up to segmentation labeling post-processing, discussing the technique selection and development in every case. The methodological approach suggests the use of a color based pixel classification strategy as the basis of the mouth structure segmentation scheme, complemented by a smart pre-processing and a later label refinement.

The main contribution of this work, along with the segmentation methodology itself, is based in the development of a color-independent label refinement technique. The technique, which is similar to a linear low pass filter in the segmentation labeling space followed by a non-linear selection operation, improves the image labeling iteratively by filling small gaps and eliminating spurious regions resulting from a prior pixel classification stage. Results presented in this document suggest that the refiner is complementary to image pre-processing, hence achieving a cumulative effect in segmentation quality.

At the end, the segmentation methodology comprised by input color transformation, pre-processing, pixel classification and label refinement, is put to test in the case of mouth gesture detection in images aimed to command three degrees of freedom of an endoscope holder.

Keywords: Image segmentation, lips segmentation, gesture classification, human-machine interface.

Résumé étendu

Ce travail présente une nouvelle méthodologie pour la reconnaissance automatique des gestes de la bouche visant à l'élaboration d'IHM pour la commande d'endoscope. Cette méthodologie comprend des étapes communes à la plupart des systèmes de vision artificielle, comme le traitement d'image et la segmentation, ainsi qu'une méthode pour l'amélioration progressive de l'étiquetage obtenu grâce à la segmentation. Contrairement aux autres approches, la méthodologie est conçue pour fonctionner avec poses statiques, qui ne comprennent pas les mouvements de la tête. Beaucoup d'intérêt est porté aux tâches de segmentation d'images, car cela s'est avéré être l'étape la plus importante dans la reconnaissance des gestes.

En bref, les principales contributions de cette recherche sont les suivantes:

- La conception et la mise en œuvre d'un algorithme de raffinement d'étiquettes qui dépend d'une première segmentation/pixel étiquetage et de deux paramètres corrélés. Le raffineur améliore la précision de la segmentation indiqué dans l'étiquetage de sortie pour les images de la bouche, il apporte également une amélioration acceptable lors de l'utilisation d'images naturelles.

- La définition de deux méthodes de segmentation pour les structures de la bouche dans les images; l'une fondée sur les propriétés de couleur des pixels, et l'autre sur des éléments de la texture locale, celles-ci se complètent pour obtenir une segmentation rapide et précise de la structure initiale. La palette de couleurs s'avère particulièrement importante dans la structure de séparation, tandis que la texture est excellente pour la séparation des couleurs de la bouche par rapport au fond.

- La dérivation d'une procédure basée sur la texture pour l'automatisation de la sélection des paramètres pour la technique de raffinement de segmentation discutée dans la première contribution.

- Une version améliorée de l'algorithme d'approximation bouche contour présenté dans l'ouvrage de Eveno et al. [1, 2], ce qui réduit le nombre d'itérations nécessaires pour la convergence et l'erreur d'approximation finale.

- La découverte de l'utilité de la composant de couleur CIE $a*$ statistiquement normalisée, dans la différenciation lèvres et la langue de la peau, permettant l'utilisation des valeurs seuils constantes pour effectuer la comparaison.

Le contenu de ce document suit les étapes du processus de reconnaissance bouche geste. Tout d'abord, le Chapitre 2 introduit une bibliographie sur la segmentation de la structure de la bouche dans les images, et décrit les bases de données et les mesures de performances

utilisées dans les expériences. Le Chapitre 3 traite des représentations de couleurs et plusieurs techniques de traitement d'image qui permettent de mettre en évidence les différences entre les structures de la bouche, tout en améliorant l'uniformité à l'intérieur de chaque structure. La modélisation stochastique et les techniques de classification, communes dans la reconnaissance des formes et d'exploration de données, sont utilisées pour obtenir des résultats rapides en matière d'étiquetage (segmentation de point de départ). Une nouvelle technique pour le post-traitement des images d'étiquettes résultant de la segmentation initiale par le biais d'un raffinement itératif des étiquettes est présentée dans le Chapitre 4. Le processus est également testé avec des images naturelles, afin d'établir une idée plus complète du comportement du raffineur. Le Chapitre 5 présente une étude sur l'utilisation de descripteurs locaux de texture afin d'améliorer la segmentation de la structure de la bouche. Le Chapitre 6 introduit une version modifiée de l'algorithme automatique d'extraction du contours des lèvres, initialement traité dans le travail par Eveno et al. [1, 2], conçu pour trouver la région d'intérêt de la bouche. Le Chapitre 7 propose une nouvelle méthodologie pour la reconnaissance des mouvements de la bouche, en utilisant les techniques traitées dans les chapitres précédents. Une illustration du travail proposé dans le cas spécifique de commande porte endoscope pour le système chirurgical Da Vinci est présentée. Finalement, les conclusions de ce travail sont montrées au Chapitre 8, des questions ouvertes et des travaux futurs sont discutés dans le Chapitre 9.

Mots-clés: segmentation, segmentation de lèvres, classement de gestes, interface human-machine.

Resumen extendido

En éste trabajo se presenta una nueva metodología para el reconocimiento automático de gestos de la boca orientada al desarrollo de una interfaz hombre-máquina para el comando de endoscopios. Dicha metodología comprende etapas comunes a la mayoría de sistemas de visión artificial, como lo son el tratamiento de la imagen y la segmentación, además de un método para el mejoramiento progresivo del etiquetado resultante de la segmentación inicial. A diferencia de otras aproximaciones, la metodología propuesta se adecua a gestos bucales y que no implican movimientos de la cabeza. A lo largo del documento se presta especial interés a la etapa de segmentación, ya que es ésta la que presenta mayores retos en el reconocimiento de gestos.
En resumen, las contribuciones principales de este trabajo son:

- El diseño y la implementación de un algoritmo de refinamiento de etiquetas que depende de una segmentación y etiquetado inicial, y de dos parámetros intrínsecos al algoritmo. La estrategia produce una mejora en el etiquetado de las regiones en imágenes faciales centradas en la región de la boca, mostrando también un rendimiento aceptable para imágenes naturales.

- La propuesta de dos métodos de segmentación de las estructuras de la boca en imágenes: uno basado en la clasificación de los píxeles por su color, y otro que incluye además algunas características locales de textura. El segundo método mostró ser particularmente útil para separar la boca del fondo, mientras que el primero es fuerte en la clasificación de las estructuras visibles de la boca entre sí.

- La derivación de un procedimiento basado en características locales de textura en las imágenes para la selección automática de los parámetros del algoritmo de refinamiento.

- Una versión mejorada del algoritmo de aproximación del contorno externo de la boca presentado por Eveno y otros [1, 2], en la cual se reducen tanto el número de iteraciones necesarias para alcanzar la convergencia como el error final de aproximación.

- Se notó la utilidad de la componente CIE$a*$ normalizada estadísticamente dentro de la región de interés de la boca, para la clasificación rápida de los labios y la boca a través del uso de comparación con un umbral fijo.

El contenido del documento se presenta como sigue: primero, en el Capítulo 2 se introduce el tema de segmentación de las estructuras de la boca, y se brinda una breve descripción de las técnicas de medida de rendimiento utilizadas y de la base de datos generada para este trabajo. Seguidamente, en el Capítulo 3 son tratadas algunas representaciones de color y su potencial en la tarea de clasificación de estructuras de la boca por comparación, y en el

modelado estocástico de la distribución de color de cada una de ellas. En el Capítulo 4 se presenta una nueva técnica de refinamiento progresivo de las etiquetas resultantes de procesos de clasificación/segmentación. Su comportamiento es también estudiado en el caso de uso de segmentación de imágenes naturales y de segmentación de las estructuras de la boca en imágenes faciales. El Capítulo 5 muestra un estudio acerca del uso de un grupo de características locales de textura para el enriquecimiento de la segmentación de la boca en imágenes. En el Capítulo 6 se introduce una versión modificada del algoritmo de aproximación del contorno externo de los labios presentado por Eveno y otros [1, 2]. El Capítulo 7 contiene la aplicación de la metodología de segmentación de estructuras de la boca y reconocimiento de gestos en la tarea de generar comandos para una interfaz hombre-máquina para la manipulación de robots porta-endoscopios, en particular orientado al sistema de cirugía asistida DaVinci. Finalmente, las conclusiones de este trabajo y el trabajo futuro se presentan en los Capítulos 8 y 9, respectivamente.

Palabras clave: Segmentación, segmentación de labios, clasificación de gestos, interfaz hombre-máquina.

Contents

List of Figures

List of Tables

1 Introduction

Vision based human machine interfaces have gained great interest in recent years. The increasing computational capabilities available in mainstream PCs and embedded systems such as mobile phones, digital cameras, tablets, etc., enable the massive deployment of a wide variety of applications using techniques hitherto constrained to specialized equipment. In example, it is not uncommon to see device unlocking systems based on face recognition, voice recognition and digital prints, among others, embedded in rather simple devices.
Nevertheless, security and consumer based products are not the only fields that have profited from the visual recognition race. Medical applications have taken advantage of this technological leap, translated in the arising of high performing data analysis and visualization techniques. These techniques interact in real-time with the working environment both in patient-wise and surgeon-wise levels, empowering sophisticated assessment systems and command interfaces.
Visual assessment human machine interfaces in surgical environments can be clearly exemplified by the use of the visible tip of surgical instruments, as well as other visual cues, in order to servo endoscope holders. This approach presents acceptable results if the endoscope movement is meant to be coupled with those carried out by the instruments, but its applicability seems to be limited if such movements should be independent from each other. In that case, commanding the holder requires the use of additional elements such as joysticks, pedals, buttons, etc. Wearable marks in the head of the surgeon have also been used to estimate the desired endoscope pose by measuring the relative pose of the surgeon face.
In a previous work [3, 4], several approaches for solving the endoscope holder command were discussed, highlighting the fact that mouth gestures could be regarded as a feasible alternative to prior approaches in tackling such task. The authors used a small subset of easily identifiable gestures conveying both mouth poses and head movements, in order to control three degrees of freedom (upwards-downwards, left-right, zoom in-zoom out). The approach presents a feasible solution whenever movements of surgeon's head are admissible during the intervention, but is not usable otherwise. This fact poses an important limitation since modern tools such as the DaVinci surgical system impose strict constraints regarding the surgeon pose during intervention.
In this work, a novel methodology for automatic mouth gesture recognition aimed towards the development of endoscope holder command HMIs is presented. The methodology comprises common stages in most artificial vision systems, such as image pre-processing and segmentation, along with a least treated one which is label post-processing or refinement. Unlike previous approaches, the methodology is designed to work with mouth poses that do not include head movements. Much interest is given to image segmentation related tasks, since it has proven to be the most challenging stage in the gesture recognition streamline.
Briefly, the main contributions of this research are:

- The design and implementation of a label refinement algorithm that depends on an

initial segmentation/pixel labeling and two correlated parameters. The refiner improves the segmentation accuracy shown in the output labeling for mouth images, also bringing an acceptable improvement when using natural images.

- The statement of two segmentation schemes for mouth structures in images, one utterly based in pixel color properties, and other including some local texture elements, which complements each other in obtaining a fast and accurate initial structure segmentation. The color-only scheme proves to be particularly accurate for structure from structure separation, while the texture and color scheme excels at mouth from background distinction.

- The derivation of a texture based procedure for automating the parameter selection for the segmentation refinement technique discussed in the first contribution.

- An improved version of the mouth contour approximation algorithm presented in the work by Eveno *et al.* [1, 2], which reduces both the number of iterations needed for convergence and the final approximation error.

- The discovery of the usefulness of the CIE$a*$ color component statistically normalized in differentiating lips and tongue from skin, enabling the use of constant threshold values in such comparison.

The outline of this document follow the stages that conform the mouth gesture recognition process. First, Chapter 2 introduces a review on mouth structure segmentation in images, and describes the databases and accuracy measurements used throughout the remainder of the document. Chapter 3 explores several color representations and pre-processing techniques that help in highlighting the differences between mouth structures, while improving the uniformity within each structure. Stochastic modeling and classification techniques, common in pattern recognition and data mining, are used to obtain fast labeling results usable as a starting point segmentation. A novel technique for post-processing the label images resulting from the initial segmentation through iterative label refinement is presented in Chapter 4. The method is also tested with natural images, in order to establish a broader idea of the refiner's behavior. Chapter 5 presents a study on the use of local texture descriptors in order to enhance mouth structure segmentation, following the methodological approach in Chapter 3. Chapter 6 presents a modified version of the automatic lip contour extraction algorithm originally introduced in the work by Eveno *et al.* [1, 2], aimed to find the mouth's region of interest. Chapter 7 proposes a new methodology for mouth gesture recognition in images using most of the elements treated in previous Chapters. It exemplifies the proposed workflow in the specific case of endoscope holder command for the DaVinci surgical system. Finally, the conclusions of this work are drawn in Chapter 8, and some open issues and future work are discussed in Chapter 9.

2 Mouth structure segmentation in images

The main goal of this Chapter is to serve as an extended introduction to the document. It presents the topic of mouth structures segmentation in images, giving insights on how this task has been addressed in the past, and how such task is approached in this work.
The Chapter can be split in two main parts. The first part, which is identified as Section 2.1, contains a synthesis on how the challenge of automatic mouth structure segmentation has been addressed in the past. The second part, denoted as Section 2.2, introduces a series of performance measurement tools that serve as a basis to present the results contained in the remaining chapters, as well as a brief description of the databases used to obtain the aforementioned results. The later is a common reference throughout the rest of this document.

2.1 Previous work

One major step in automatic object recognition in images and scene perception is image segmentation. In this step, all pixels in the image are classified according to their color, local texture, etc., preserving local compactness and connectedness. As in any other image segmentation task, the techniques that can be used to deal with the task may be approached from a taxonomic point of view regarding their inherent abstraction level. Figure 2.1 shows one of those possible approaches for the specific challenge of mouth structure segmentation, and used more specifically in lip segmentation. It is noteworthy that some techniques may overlap in more than one category (as in the case of active appearance models and Fuzzy C-Means).

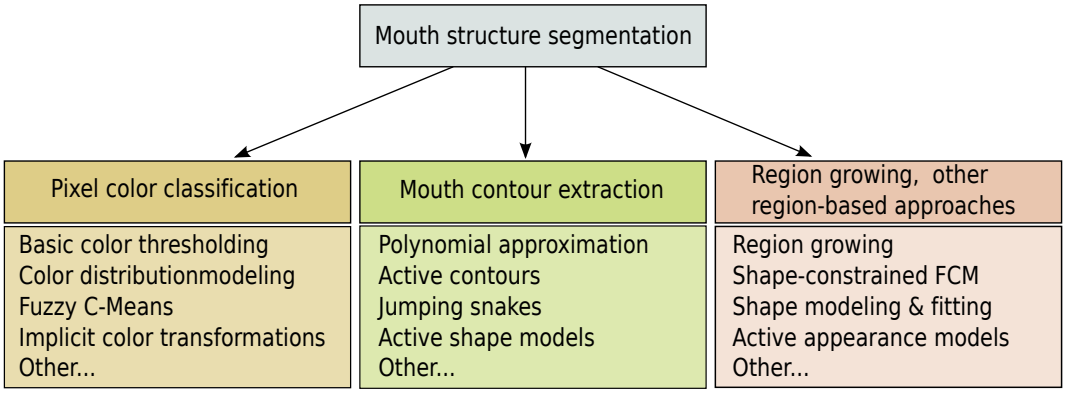


Figure 2.1: Mouth structure segmentation in images: taxonomic diagram.

Texture and region based approaches conform the basis of high level segmentation algorithms, while color and highly localized texture features and moments are more common to low level segmentation approaches. In high level segmentation strategies, like watershed or region growing-based segmentation, the most time-consuming tasks derive from these region constraints [5]. Thereby, region constraints are usually excluded at some extent when developing in favor of speed over quality.
Now, the possibilities narrow down when restraining the application to mouth segmentation in images. For instance, tongue and teeth segmentation have seen their development enclosed in rather punctual applications whose solution usually imply the use of specific illumination and acquisition hardware. For instance, works like [6, 7, 8, 9] exploit the inherent highlighting generated by hyperspectral or infra-red lighting in order to facilitate tongue segmentation for upwards and downwards tongue gestures. Using that configuration, tongue appears clearly highlighted from lips. Kondo *et al.* [10] used range imagery and X-ray images in order to perform teeth segmentation and reconstruction as three dimensional surfaces. Lai & Lin [11] also used X-ray images for teeth segmentation.
In the other hand, most of previous work on mouth structure segmentation in images has been focused in segmenting the lip region from skin, leaving aside the remaining visible mouth structures (namely teeth, gums and tongue). Therefore, the remaining of this Section focuses in exploring several approaches that arose to cope with such challenge.

2.1.1 Lip segmentation based on pixel color classification

Pixel-based segmentation encompasses pixel classification using region color distribution models. Commonly, those models are linear and non-linear approximations of the region separation boundaries, reducing the classification problem to comparing the input feature vectors against a set of thresholds. The thresholds define region boundaries in the feature space, conforming models of the regions' color distribution by themselves.
Thresholds can be set statically, taking into account any prior knowledge about the contents of the image, or they can also be dynamically adjusted to achieve a proper segmentation of the image. Dynamical threshold selection rely in either local statistics, like in the case of adaptive thresholding [5], or global statistics usually based in the image histogram, like in Otsu's technique [12]. In most cases, these techniques manage to maximize interclass variances while minimizing intraclass variances.
Succeeding pixel color based lip segmentation requires a proper input representation of the data. Thereupon, several approaches for finding appropriate color representations are treated.

Color representations used in mouth structure segmentation

The first step in solving a segmentation task is to find an adequate input representation which helps highlighting the existent differences among regions. Those representations can be classified in two different categories. The first category is composed by general purpose color transformations that prove to be helpful in the specific application; and the second comprises color transformations which are specifically designed aiming towards the application through the use of linear and non-linear transformations. In this Section, some approaches explored in both categories are treated briefly.

The lip region and skin are very similar in terms of color [13]. For this reason many different color transformations have been developed.
Some linear transformations of the RGB color space have led into fair results in term of color separability between lips and skin. For example, Guan [14, 15], and Morán & Pinto [16] made use of the Discrete Hartley transform (DHT) in order to improve color representation of the lip region. The component C_3 of the DHT transformation properly highlights lip area for subjects with pale skin and no beard, as shown in Figure 2.3d. Chiou & Hwang [17] used the Karhunen–Loève Transform in order to find the best color projection for linear dimension reduction.
Hue-based transformations are also used in lip segmentation. The pseudo hue transformation, proposed in the work of Hurlbert & Poggio [18], exhibits the difference between lips and skin under controlled conditions, as seen in Figure 2.3b. The pseudo hue transformation focuses in the relation between red and green information of each pixel, and is defined as $p_H = R/(R+G)$. It leads into a result that appears very similar to the one achieved using hue transformation, but its calculation implies less computational reckoning, and therefore was a preferred alternative in some work [19, 1, 20]. Particularly, a normalized version of this representation was used in the development of a new color transformation, called the *Chromatic Curve Map*[1]. As with hue, pseudo hue cannot separate properly the lips' color from the beard and the shadows. Pseudo-hue may generate unstable results when the image has been acquired under low illumination, or in dark areas; this effect is mainly due to a lower signal to noise ratio (SNR).
Chromaticity color space, introduced in 1931 by the CIE (Commission Internationale de l'Éclairage), have been used in order to remove the influences of varying lighting conditions, so that the lip region can be described in a uniform color space. Using a chromatic representation of color, the green values of lip pixels are higher than that of the skin pixels. Ma *et al.* [21] reported that in situations with strong/dim lights, skin and lip pixels are better separated in chromatic space than in the pseudo-hue plane. In order to address changes in lighting, in [22] the color distribution for each facial organ on the chromaticity color space was modeled. Chrominance transformations (YC_bC_r, YC_bC_g) have also been used in lip segmentation [23].
The use of pixel based, perceptual non-linear transformations, which presents a better color constancy over small intensity variations, has been a major trend in the late 90s. Two well-known perceptual color transformations, presented by the CIE, are the CIE$L*a*b*$ and CIE$Lu'v'$. The principle behind these transformations is the compensation of natural logarithmic behavior of the sensor. Work like [24, 25, 26] made use of these color representations in order to facilitate the lip segmentation process in images. Like in $Y'CbCr$, $L*a*b*$ and $Lu'v'$ representations theoretically isolate the effect of lighting and color in separated components[1].
In Gómez *et al.* [27], a combination of three different color representations is used. The resulting space enables the algorithm to be more selective, leading into a decrease of spurious regions segmentation. The authors perform a clipping of the region of interest (RoI) in order to discard the nostril region, as shown in Figure 2.2.
One major disadvantage of the pixel based techniques is the lack of connectivity or shape constrains in its methods. In order to deal with that problem, Lucey *et al.* [28] proposed

[1]Sensor operating noise and lighting response, as well as lens-related chromatic distortions increase the correlation between intensity and color.

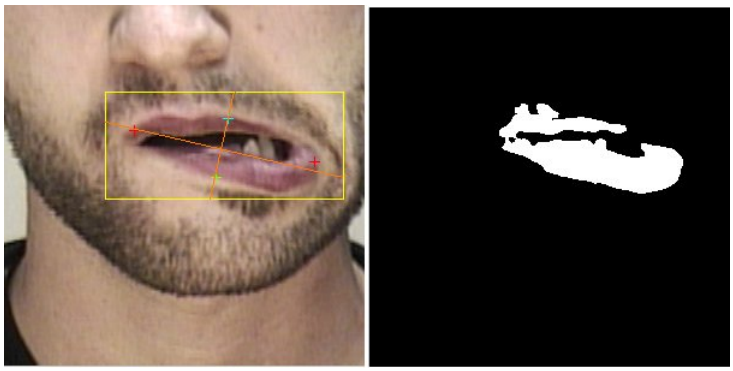


Figure 2.2: Example of mouth segmentation presented in [27]

.

a segmentation algorithm based on dynamic binarization technique that takes into account local information in the image. The first step in Lucey's method is to represent the image in a constrained version of the R/G ratio. Then, an entropy function that measures the uncertainty between classes (background and lips) is minimized, in terms of membership function parameters. After that, a second calculation based in neighboring region information is used to relax the threshold selection. Despite of the threshold adaptation, a later post-processing is needed in order to eliminate spurious regions.

Optimal color transformation design using Linear Discriminant Analysis

Skin, lip and tongue color tends to overlap greatly in every color representation treated in the previous Section. This task has proven to be uneasy even when using the so-called chromatic constant transformations–which try to make the lighting effect in color negligible, due to factors such as noise, sensor sensitivity, etc..
However, it is possible to search for optimal solutions to mouth structure classification. The simplest way to perform this optimization comprise the calculation of a set of linear combinations of the input features, each one aimed to distinguish a structure from the others. Notice that following this approaches one obtains a reduction in the feature space if the input representation dimension surpasses the number of desired output classes, and if only one projection is computed per each class. Another approach, which comprises the calculation of more elaborated color models for each mouth structure is discussed in the next section.
A common approach of linear optimization for classification, a technique which can also be used for dimensional reduction, is the *Fisher Linear Discriminant Analysis (FLDA)*. The *FLDA* is carried out by finding a projection matrix or vector (for multiclass or bi-class problems, respectively) which transforms the input space into another space with reduced dimensionality, aiming to maximize the inter-classes covariance while minimizing the intra-class covariance of the data. Conversely, the goal of *FLDA* can be translated into maximizing

$$S(\boldsymbol{w}) = \frac{\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{\Sigma}_B\boldsymbol{w}}{\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{\Sigma}_W\boldsymbol{w}} \tag{2.1}$$

where $\boldsymbol{w}$ is the projection vector, $\boldsymbol{\Sigma}_B$ represents the inter-class covariance matrix, and $\boldsymbol{\Sigma}_W$

represents the intra-class covariance matrix.
In the case where only two classes are considered, the closed solution for maximizing $S(\boldsymbol{w})$ is reduced to

$$\boldsymbol{w} \propto (\boldsymbol{\Sigma}_{Class1} + \boldsymbol{\Sigma}_{Class2})^{-1}(\boldsymbol{\mu}_{Class1} - \boldsymbol{\mu}_{Class2}) \tag{2.2}$$

where $\boldsymbol{\mu}$ stands for the class mean. Special care should be taken in the case where $(\boldsymbol{\Sigma}_{Class1} + \boldsymbol{\Sigma}_{Class2})$ is close to singularity, in which case the pseudo inverse of $(\boldsymbol{\Sigma}_{Class1} + \boldsymbol{\Sigma}_{Class2})$ should be used. Once $\boldsymbol{w}$ is computed, each input feature can be easily rated by studying its associated element in the projection vector, or the corresponding eigenvalue in $\boldsymbol{\Sigma}_W^{-1}\boldsymbol{\Sigma}_B$. Hence, the data can be classified by projecting each input pattern (denoted by $\boldsymbol{x}_i$) and comparing the result with a given threshold, as seen in (2.3).

$$x_i' = \boldsymbol{x}_i^T \boldsymbol{w} \tag{2.3}$$

One common choice for the threshold is obtained by averaging the position of the two class means in the projected space. In our tests, thresholds are computed taking into account the class standard deviations in the projected space, ensuring that the threshold will be at the same Mahalanobis distance from both class means (as in (2.4)).

$$th = \frac{(\boldsymbol{\mu}_{Class1}^T \boldsymbol{w})\sigma'_{Class2} + (\boldsymbol{\mu}_{Class2}^T \boldsymbol{w})\sigma'_{Class1}}{\sigma'_{Class1} + \sigma'_{Class2}} \tag{2.4}$$

Multi-class *LDA* can be extended by generalizing the objective function in (2.1), or by configuring an "one against the rest" scheme and then finding a projection for each cluster individually. In example, *LDA* has been applied in order to project common color spaces, like RGB, into specifically designed spaces oriented to perform the lip enhancement. Zhang *et al.* [29] made use of LDA in order to find an optimal linear separation for lip color and skin color, using as input space the green and blue color components. Kaucic & Blake [30] used the *FLD* to find the best linear separation in RGB, for a given image set; this strategy was also followed by Rongben *et al.* [31], and Wakasugi *et al.* [32]. In all those approaches, the authors report competitive performance in lip region segmentation if compared with previous works.
Zhang *et al.* [29] uses *FLDA* in order to find a linear transformation that maximizes the difference between the color in the skin and lips. After that, it performs an automatic threshold selection based in preserving the histogram area occupied by the lips' region. Kim *et al.* [33] proposes the use of manually annotated data in order to train a fuzzy inference system, which is used as a confidence index for automatic threshold selection.

Color distribution modeling in images

Modeling lip/skin color separation using optimal linear techniques like *FLDA* implies the assumption that a linear model is able to map pixel classification sufficiently. Nevertheless, such overlap appears to have a non linear solution aiming towards structure color classification. Hence, non-linear modeling approaches appear as a natural way to cope with these limitations.
In Loaiza *et al.* [34] the use of feed-forward artificial neural networks (FFNNs) in order to model

such difference is proposed. The network was trained using a wide set of color representations used for skin and lip color segmentation tasks as the input, and the desired class as the output. As in the case of *FLDA/LDA*, a set of manually labeled color patterns is needed in order to train the network. The results of the paper reported slightly better results by using the ANN approach rather than linear approaches in lips highlighting. A similar approach was followed by Dargham *et al.* [35]. In both cases, the black-box modeling given by the connections among the neural units provide a model set whose structure lacks of a direct physical interpretation. Moreover, neural networks usually comprise a parameter set that grows linearly in terms of the input feature space dimension, and the number-of and size-of intermediate or hidden layers.

Another approach, which is very common in pattern recognition and data mining, comprises the extension of linear approaches through the use of the kernel trick. In this case, the space is augmented with linear and non-linear combinations of typical input features in order to find linear solutions to non-linear separation problems.

A rather used tool in statistical modeling of data is Mixture Modeling. In particular, a well known and widely used alternative, the Gaussian Mixtures (GMs), has been used in classifiers, density estimators and function approximators [36]. For instance, the GM approximation of the probability distribution of a $d-$dimensional random variable X with realizations x can be described by

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{2.5}$$

where π_k is the mixing proportion, and holds for a) $0 < \pi_k < 1 \ \forall k \in 1, ..., K$, and b) $\sum_{k=1}^{K} \pi_k = 1$; and $\mathcal{N}(\cdot|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is the $d-$dimensional Gaussian function with mean $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$.

One common approach to estimate the parameter set in a mixture model is based on the Expectation-Maximization (EM) algorithm [37]. The goal in the EM formulation is, given $\boldsymbol{z}$ the unknown and $\boldsymbol{y}$ the known observations of X, to find the vector of parameters such that the $E_z[f(\mathbf{y}, \mathbf{z}|\theta)|\mathbf{y}]$ reaches its maximum; which, for mathematical facilities, turns into maximizing the term $Q(\theta)$ in (2.6), in terms of θ.

$$Q(\theta) = E_z[\log(f(\mathbf{y}, \mathbf{z}|\theta))|\mathbf{y}] \tag{2.6}$$

A more detailed description of Gaussian mixtures is presented in Section 3.2.

Gaussian Mixtures have been used extensively in lip segmentation. One straightforward use of them is to model the color distribution of skin and lips, in order to maximize its separability, as in Basu *et al.* [38, 39, 40] and Sadeghi *et al.* [41, 42]. In the later, a Sobol-based sampling is used in order to reduce the computational load of the EM algorithm. The same technique was used by Bouvier *et al.* [43] for estimating skin color distribution of the images.

Shape constraints can be also codified in generating the GMM. In Kelly *et al.* [44], a spatial model of the lips conformed by a set of four Gaussian functions is introduced. The model is adjusted with the information provided in a video sequence, using a set of five different constraints which enables the update of rotational, translational and prismatic deviations in the model (the later allows a non-rigid approximation of the data). In Gacon *et al.*[45], a

method for dynamic lip color characterization and lip segmentation is proposed. The method is based in statistical Gaussian models of the face color. The method is able to model both static and dynamic features of pixel color and mouth shape, and it was reported that the technique could compensate illumination changes and quick movements of the mouth.

GMMs have been also used for lip contour representation. Bregler & Omohundro [46] presented a technique which uses a high dimensional modeling using GMMs, and a later re-projection the model in the image space. The projection is used to constrain an active contour model that approximates the lips' outer contour. Chan [47] used a GMM with color and shape information, in order to model the lip contours.

Implicit color transformations

In cases in which there is no prior labeling information available, it is still possible to compensate some lighting related issues in the images by taking into account image statistics as regularization or normalization factors. Those transformations that imply the inclusion of such measures are called implicit color transformations.

Lucey *et al.* [48] combine chromatic representations with features which consider the localized second order statistics present in adjacent pixels, and then perform segmentation as a classification problem. They found that results were not improved by using those features. On the contrary, the outer mouth contour was degraded.

Implicit color transformations take into account image statistics as normalizing or stabilizing factors. Those transformations are somehow capable to compensate small changes in illumination for each image, and thus increasing constancy in threshold selection over a variable image set. Two remarkable implicit, non-linear transformations, are the *Mouth Map*, by Hsu *et al.* [49], and the *Chromatic Curve Map*, by Eveno *et al.* [1]. The first one is intended to adjust the overall color compensation in function of the color values in the whole image, based in the traditional $YCbCr$ color space representation. The transformation is described in (2.7).

$$MouthMap(x,y) = C_r(x,y)^2 \left(C_r(x,y)^2 - \eta \frac{C_r(x,y)}{C_b(x,y)} \right)^2 \tag{2.7}$$

with

$$\eta = 0.95 \frac{\frac{1}{N} \sum_{(x,y)\in FG} C_r(x,y)^2}{\frac{1}{N} \sum_{(x,y)\in FG} \frac{C_r(x,y)}{C_b(x,y)}} \tag{2.8}$$

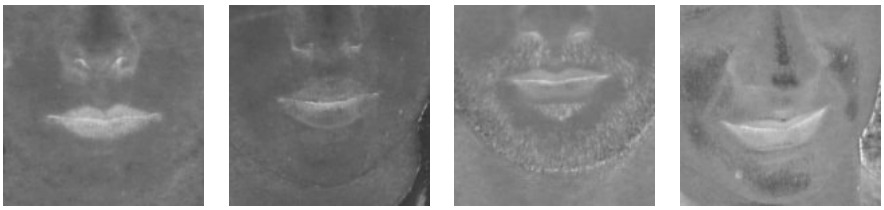
where N stands for the total number of pixels in the image, and FG is the set of all possible pixel locations in the image. The second one, uses the normalized pseudo-hue representation in order to compute a parabolic approximation of the local chromatic curvature in each pixel. The value of the *Chromatic Curve Map* at each pixel of the image can be computed as the higher-order coefficient of the 2nd order polynomial that passes through three points, whose

positions are expressed in (2.9).

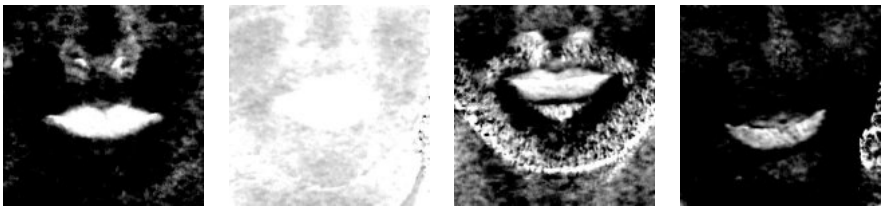
$$\begin{aligned} \boldsymbol{p}_1(x,y) &= \begin{pmatrix} -\alpha k(x,y) \\ B(x,y) + \beta k(x,y) \end{pmatrix} \\ \boldsymbol{p}_2(x,y) &= \begin{pmatrix} 0 \\ G(x,y) \end{pmatrix} \\ \boldsymbol{p}_3(x,y) &= \begin{pmatrix} 1 \\ \gamma k(x,y) \end{pmatrix} \end{aligned} \tag{2.9}$$

here, $k(x,y)$ is the normalized pseudo hue at the pixel (x,y). The authors chose the values of α, β and γ by sweeping the parameters' space. An example of images represented using the *Curve Map* can be seen in Figure 2.3c.

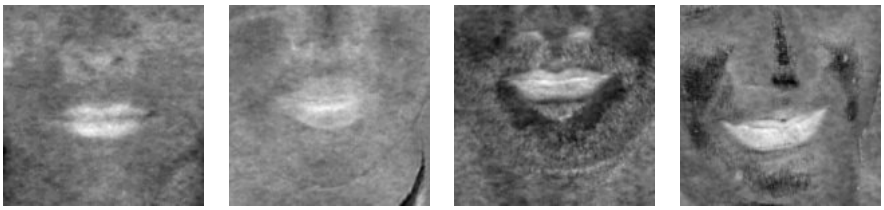
(a) Original images.



(b) Pseudo hue images (shown with contrast enhancement).



(c) Curve map [1].



(d) C_3 images [14].

Figure 2.3: Effect of color transformation in mouth images.

The two transformations reported lip enhancement for subjects with white skin and without beard. However, their accuracy decreases when dark skin or beard are present. Also, the reckoning inherent in implicit transformations make them less suitable in real–time applications.

Another non-linear implicit transformation, proposed by Liévin & Luthon [50], is the LUX Color Space (Logarithmic hUe eXtension). LUX components are given in (2.10).

$$
\begin{aligned}
L &= (R+1)^{0.3}(G+1)^{0.6}(B+1)^{0.6} - 1 \\
U &= \begin{cases} \frac{M}{2}\left(\frac{R+1}{L+1}\right) & \text{if } R < L, \\ M - \frac{M}{2}\left(\frac{L+1}{R+1}\right) & \text{otherwise} \end{cases} \\
X &= \begin{cases} \frac{M}{2}\left(\frac{B+1}{L+1}\right) & \text{if } B < L, \\ M - \frac{M}{2}\left(\frac{L+1}{B+1}\right) & \text{otherwise} \end{cases}
\end{aligned}
\tag{2.10}
$$

where M is the dynamic range of gray levels, equal to 256 for 8-bit coding. Since the hue of the face skin is mainly red, for face and lip segmentation consideration of the U component is enough. Related to C_r or H components the U transformation gains in contrast, but it is also insensitive to illumination variations.

A well-performing implicit transformation aimed for lip segmentation in RoI clipped facial images is the normalized $a*$ component presented in [4]. This technique is addressed in detail in Section 3.1.3.

The use of implicit transformation has also a few drawbacks, mostly derived from image uncertainty. In example, the transformations discussed before in this Section fail in highlighting the lip region when skin color varies considerably if compared to the one used in their development. Also, the presence of specific æsthetic or prosthetic elements in the image, as well as the presence of beards or mustache, modifies the relative relationship between lip and skin color in terms of image statistic. Thereby, implicit methods may outperform typical color transformations whenever the image statistics fits a pre-defined range.

Notice that implicit color transformations are designed for images which exhibit similar color characteristics. The behavior of an implicit transformation on an image that escapes from the design set is unpredictable. Hence, this kind of technique are not advised when image statistics cannot be safely asserted in a certain design range, or if its contents are undefined.

2.1.2 Mouth segmentation derived from contour extraction

Lip could be interpreted as a deformable object, whose shape or contours can be approximated by one or many parametric curves. Then, it might seem evident that one must first look for the mouth contour before trying to segment its inner structures. In this Section, some techniques that have been used in outer lip contour approximation in images in lip segmentation are briefly discussed.

Polynomials can be used as lip contour approximation, notably between second and fifth degrees. For instance, in Stillittano & Caplier [51] four cubics are used to represent mouth contour starting from a series of detected contour points, two in the upper lip and two in the lower lip. The keypoints are extracted using the *Jumping Snakes* technique presented by Eveno *et al.* [1, 20, 2]. The authors reported some issues due to the presence of gums or tongue.

Nevertheless, as stated in [52], low order polynomials (up to fourth degree) are not suitable for anthropometric applications since they lack in mapping capabilities for certain features; in the other hand, high order polynomials may exhibit undesired behavior on ill-conditioned

zones. For that reason, most of the work that use polynomials to approximate the lip contour are intended for lipreading applications.

Werda *et al.* [53] adjust the outer contour of the mouth by using three quadratic functions: two for the upper lip and one for the lower lip. The adjustment is performed after a binarization process in the $R_nG_nB_n$ color space where the lighting effect is reduced. Each component of the RGB space is transformed to the new $R_nG_nB_n$ by A_n=255*A/Y, with Y the intensity value. The final representation contains a strong geometric parametric model, whose parameters enable the contour to be deformed into a constrained set of possible shapes. In [54] three or four parabolas are used to extract mouth features for the closed and open mouth respectively. Rao & Mesereau [55] used linear operators in order to find the horizontal contour of lips, and then they approximate that contours with two parabolas. Delmas *et al.* [56] extract from the first frame of a video sequence, the inner and outer lip contour by using two quartics and three parabolas. The polynomials are defined by the corners and vertical extrema of the mouth, which are found by using [57].

Active contours or snakes are computer-generated curves that move within images to find object boundaries. In this case, the inner and outer contour of the mouth. They are often used in computer vision and image analysis to detect and locate objects, and to describe their shape. An active contour can be defined as a curve $\mathbf{v}(u,t) = (x(u,t), y(u,t)), u \in [0,1]$, with t being the temporal position of the point in the sequence, that moves in the space of the image [58]. Evolution of the curve is controlled by the energy function in (2.11).

$$E_{\text{ac}} = \int_0^1 E_{\text{int}}(\mathbf{v}(u)) + E_{\text{im}}(\mathbf{v}(u)) + E_{\text{ext}}(\mathbf{v}(u))du \tag{2.11}$$

E_{int} represents the internal energy of the curve, and controls the properties of stretching and bending of the curve. E_{im} is the image energy, and is related to properties in image data. E_{ext} is an external energy, and usually represents application-specific constraints in the evolution of the curve. A technique called Gradient Vector Flow (GVF) was developed in order to improve convergence and accuracy in representing high curvature regions in the contour model [59, 60]. More information about the internal energy of the active contours is described by (2.12) [61, 58].

$$E_{\text{int}}(\mathbf{v}(u)) = \frac{1}{2}\left(\alpha(u)|\mathbf{v}'(u)|^2 + \beta(u)|\mathbf{v}''(u)|^2\right) \tag{2.12}$$

An example of outer lip contour extraction performed with active contours can be seen in Figure 2.4.

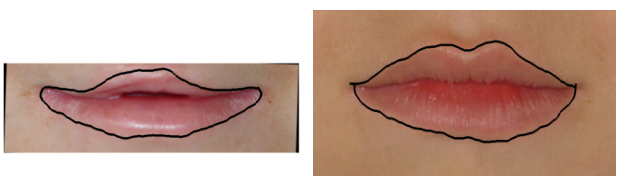


Figure 2.4: Examples of outer contour parametrization by active contours.

The generalized form of active contours was used in lip contour detection in the work of Lai & Chan [62] Wakasugi *et al.* [32] use a B-spline curve as the initialization of an active contour. Because the internal energy is related to the smoothness of the active contour, it is not considered since the B-spline representation maintains smoothness via implicit constraints. Okubo & Watanabe [63] used active contours approximated in the optical flow between the images in a sequence. In another approach, Ramos *et al.* [64] use an elliptic B-spline to approximate the lip contour, after a chromaticity clustering segmentation process. Hernández *et al.* [58] presented a simplified form of GVF for active contours, and applied it to mouth segmentation. A simplified parametrization of outer contour which uses fourth-order polynomials after active contour's convergence can be found in [52]. The outer mouth contour can be precisely described using such technique, but it is highly dependent on a prior segmentation stage.
Wu *et al.* [65] proposed a method combining a GVF snake and a parabolic template as external force, to improve outer lips extraction performance against random image noise and lip reflections. The technique did not provide good results for the extraction of inner lips, because the lips and the area inside the mouth are similar in color and texture. So, they used two parabolas as inner lips. Morán & Pinto [16] used the GVF in order to constrain the approximation of a parametric contour, bound by a set of landmarks, conforming an active shape model. The landmarks are meant to converge in the lip's inner and outer contour. Mouth region bounding box is found by clipping on the horizontal and vertical axis in the perpendicular projection of each axis. The GVF is computed in the $C_3 + U$ color space over time, where U represents the u component in CIELUV perceptual color space. Another approach that uses GVF is the one presented in [66]. In this case, the Viola-Jones face detector [67] is used to detect the face bounding box and the mouth bounding box. After that, a formulation of active contours using the level set method without re-initialization is implemented, in order to perform the model tuning.
In Eveno *et al.* [20], the authors carried out the lip contour representation by searching a set of key points in the horizontal and vertical intensity projections of the mouth region, and then approximating a set of polynomials to the resulting points. The point search and fine-tuning is controlled by a special image gradient called *hybrid edges*, based in both luminance and pseudo hue. This work evolved into a new approach, called *Jumping Snakes* [68, 2, 69]. This method allows lip contour detection just by giving an arbitrary point above the lip region in the image. At each iteration, a new pair of nodes are added at both corners of the model. Seyedarabi *et al.* [70] uses a two-step scheme of active contours in order to approximate lip's outer contour. First, a Canny operator is used in the image, and a high threshold is used for upper lip contour extraction. Once converged, a second lower threshold is used to deflate a deformable model that stops in the lower lip outer contour. Beaumesnil & Luthon [71, 72] presented a real-time 3D active contour based technique for mouth segmentation, in which a 3D model of the face is fitted directly to the image.

2.1.3 Shape or region constrained methods for lip segmentation

Color enhancement is a good first step in segmenting lips from skin, as well as other facial features. But, as shown in the previous Section, changing color representation is not enough to segment the different structures present in facial images, particularly in the mouth region.

Regions with shadows, beards, gums and tongue often overlap in color with the lips. Therefore, it is needed to include some other constraints in the problem of representing the image information, which allows to separate properly the mouth, and more specifically, the lips. One can cope with that problem by either including shape constraints in the segmentation, testing local connectivity in small spatial neighborhoods, or by trying to adjust the image contour information to a specific parameterizable template.
In this section, some alternatives to the first two categories–shape constraints and region based methods–which have been used to solve the lip segmentation problem are treated. Since most segmentation problems can be seen as classification and labeling problems, some of them have their roots in statistical classification methods and pattern recognition strategies.

Shape-Constrained Fuzzy C-Means

Fuzzy C-means (FCM) is a common clustering technique used in image segmentation. It was posted first in the mid 60s, and introduced in pattern recognition in the late 60s [73]. The FCM basics are summarized in the text by Bezdek [74, 73].
FCM segmentation is founded in the principle of feature dissimilarity, as follows: given $\boldsymbol{X} = \{\boldsymbol{x}_{1,1}, \boldsymbol{x}_{1,2}, \ldots, \boldsymbol{x}_{N,M}\}$ a set of features that corresponds to an image I of size $N \times M$, each $\boldsymbol{x}_{r,s} \in \mathbb{R}^q$ being the vector of features of the correspondent pixel in I, and C the number of fuzzy clusters in the image, the goal is to find a set $\boldsymbol{V} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_C\}$ of C different centroids $\boldsymbol{v}_i \in \mathbb{R}^q$, and a matrix $\boldsymbol{U}$ with size $M \times N \times C$ which is a fuzzy partition of $\boldsymbol{X}$, such that minimizes the cost function $J(\boldsymbol{U}, \boldsymbol{V})$ in (2.13).

$$J(\boldsymbol{U}, \boldsymbol{V}) = \sum_{r=1}^{N} \sum_{s=1}^{M} \sum_{i=0}^{C-1} u_{i,r,s} D_{i,r,s} \tag{2.13}$$

subject to

$$\sum_{i=0}^{C-1} u_{i,r,s} = 1, \quad \forall (r,s) \in I \tag{2.14}$$

$D_{i,r,s}$ is a term that reflects the distance between the feature vector $\boldsymbol{x}_{r,s}$ and the centroid $\boldsymbol{v}_i$; $u_{i,r,s}$ represents each element in the fuzzy partition $\boldsymbol{U}$. The feature set X is usually composed by different color representations. The optimum solution of the cost function can be referred as $J_m(\boldsymbol{U}, \boldsymbol{V})$, and is a stationary point at which the gradient of J_m equals to zero. For the case of the partial derivative of J_m with respect to $\boldsymbol{U}$ and setting it to zero, a closed-form expression of $u^m_{i,r,s}$ can be found as [75]

$$u^m_{i,r,s} = \left[\sum_{j=0}^{C-1} \left(\frac{D_{i,r,s}}{D_{j,r,s}} \right)^{\frac{1}{m-1}} \right]^{-1} \tag{2.15}$$

Commonly, the fuzzy partition $\boldsymbol{U}$ is compared with a threshold, in order to obtain a crisp set of disjoint regions in the source image. In such case, the FCM scheme acts as a combination of color representation and binarization algorithm.
Liew *et al.* [76] uses FCM in order to obtain the mouth's probability map from the basic

CIE$Lu'v'$ color representation. This map is used to facilitate the mouth's contour extraction, which is done by fitting a deformable template. Gordan *et al.* [77], report a modified version of the FCM algorithm which includes not only luminance information but also spatial information about the pixels in the image.

Segmentation techniques based in pixel color classification (like those treated in this section) suffer from the effects of noise or high color variations inside regions, usually producing spurious regions and gaps. FCM is, *per se*, a classification technique, and thus it can be arranged along with pixel classification techniques treated before. However, its formulation has been modified in order to include shape-based constrains. Leung, Wang & Lau [24, 75] introduced a modified version of a FCM-based unsupervised lip color segmentation engine which consider elliptical shape constraints in its formulation.

The introduction of an elliptical constrain in the cost function leads to formulation in (2.16) [24, 75, 78, 79]:

$$J_m(\boldsymbol{U},\boldsymbol{V},\boldsymbol{p}) = \sum_{r=1}^{N}\sum_{s=1}^{M}\sum_{i=0}^{C-1} u_{i,r,s}^m (d_{i,r,s}^2 + \alpha f(i,r,s,\boldsymbol{p})) \tag{2.16}$$

The term $d_{i,r,s}^2$ is the same as $D_{i,r,s}$ in (2.13). The term $\boldsymbol{p} = \{x_c, y_c, w, h, \theta\}$ is the set of parameters that describes the aspect and position of the constraining ellipse. The expression in (2.16) can be splatted in two parts, where the first one is conformed by typical FCM terms, and the second is $\boldsymbol{p}$-dependent, as shown in (2.17).

$$\begin{aligned} J_m(\boldsymbol{U},\boldsymbol{V},\boldsymbol{p}) &= J_{m1}(\boldsymbol{U},\boldsymbol{V}) + J_{m2}(\boldsymbol{U},\boldsymbol{p}) \\ J_{m1}(\boldsymbol{U},\boldsymbol{V}) &= \sum_{r=1}^{N}\sum_{s=1}^{M}\sum_{i=0}^{C-1} u_{i,r,s} d_{i,r,s}^2 \\ J_{m2}(\boldsymbol{U},\boldsymbol{p}) &= \alpha \sum_{r=1}^{N}\sum_{s=1}^{M}\sum_{i=0}^{C-1} u_{i,r,s} f(i,r,s,\boldsymbol{p}) \end{aligned} \tag{2.17}$$

The elliptical geometric constrain has proven to be effective in eliminating the effect of spurious regions with similar features than those present in the lip area [75, 78].

Despite of the improvement in FCM by the introduction of geometrical terms, there are some problems associated with complex backgrounds (i.e., the presence of beards). In this case, a multi-class shape-guided FCM variant (MS-FCM) can be used [80]. The method's formulation establishes a set of constraining functions $g_{i,BKG}$ which helps in modeling complex backgrounds. The cost function of FCM can be stated as follows [80]

$$\begin{aligned} J \;=\; & \sum_{r=1}^{N}\sum_{s=1}^{M} u_{0,r,s}^m d_{0,r,s}^2 + \sum_{r=1}^{N}\sum_{s=1}^{M}\sum_{i=1}^{C-1} u_{i,r,s}^m d_{i,r,s}^2 \\ & + \sum_{r=1}^{N}\sum_{s=1}^{M} f(u_{0,r,s}) g_{OBJ}(r,s) \\ & + \sum_{r=1}^{N}\sum_{s=1}^{M}\sum_{i=1}^{C-1} f(u_{i,r,s}) g_{i,BKG}(r,s) \end{aligned} \tag{2.18}$$

subject to condition in (2.14). The functions g_{OBJ} and $g_{i,BKG}$ are usually selected so they have a sigmoid shape, and include geometrical constrains like the one presented in [75, 78]. The authors reported the method to be more reliable in segmentation with presence of beards, compared with traditional FCM and with the work of Zhang & Mersereau [19].
This approach achieves better results than the traditional FCM, fundamentally improving the spurious region generation and filling small gaps. This improvement is somehow limited to almost symmetric mouth poses, in which lip shape can be closely defined by an ellipse. Moreover, ill-conditioned results may be obtained if the ellipse is not correctly initialized.

Other techniques used in shape or region lip segmentation

In the work by Goswami *et al.*[81], an automatic lip segmentation method based on two different statistical estimators is presented: a Minimum Covariance Determinant Estimator and a non-robust estimator. Both estimators are used to model skin color in images. The lip region is found as the largest non-skin connected region. The authors present a significant improvement over the results reported in [20]. This method uses the assumption that the skin region could be detected more easily than lips. Mpiperis *et al.* [82] introduced an algorithm which classifies lip color features using Maximum Likelihood criterion, assuming Gaussian probability distributions for the color of the skin and the lips. They also compensate gestures by using a geodesic face representation. Lie *et al.* [83] uses a set of morphological image operations in temporal difference images, in order to highlight the lip area.
Artificial intelligence has also been used in lip segmentation. Mitsukura *et al.* [84, 85] use two previously trained feed forward neural networks in order to model skin and lip color. Shape constraints are included in the weights of the lip detection neural network. Once a mouth candidate is detected, a test of skin is performed in its neighborhood, using the skin detector network. After that, a lip detection neural network is used in order to select the mouth region. In another work, the same authors presented a second scheme [86] based in evolutionary computation for lip modeling.

Active Shape Models (ASMs) and Active Appearance Models (AAMs)

ASMs are statistical shape models of objects, which iteratively deform to fit to an example of the object in a new image. They do not conform to what one may interpret as a segmentation technique, but they are nevertheless widely used in object detection and classification in images. The goal using ASMs is to approximate a set of points in the image (usually provided by the acquired object's contour information) by a point distribution model, composed by the mean shape of the object model $\bar{\mathbf{x}}$ plus a linear combination of the main modes of variation of the shape P, as shown in (2.19).

$$\mathbf{x} = \bar{\mathbf{x}} + P\mathbf{b} \tag{2.19}$$

$\mathbf{b}$ is a vector of weights related to each of the main modes. The matrix P is obtained from a set of training shapes of the object, as the t main eigenvectors of the covariance of the shapes' point position. $\mathbf{x}$ is represented in an object frame scale and rotation, and thus, the measured

data should be adjusted in order to be approximated by such model. Further information in how ASMs are trained can be found in [87, 88].

In the work of Caplier [89], a method for automatic lip detection and tracking is presented. The method makes use of an automatic landmark initialization, previously presented in [90]. Those landmarks serve to select and adapt an Active Model Shape (ASM) which describes the mouth gesture. Kalman filtering is used in order to speed up the algorithm's convergence through time.

In Shdaifat *et al.* [91] an ASM is used for lip detection and tracking in video sequences, the lip boundaries are model by five Bézier curves. In [92], a modified ASM algorithm is employed to search the mouth contour. The modification consist in the use both local gray intensity and texture information on each landmark point. In cases where it is possible that landmark points are incorrect (for example when the lip boundary is not clear), it is better to characterize the distribution of the shape parameter b by a Gaussian mixture rather than by single Gaussian. Jang *et al.* [93] developed a method for locating lip based on ASM and using a Gaussian mixture to represent the distribution of the shape parameter. In Jiang *et al.* [94], a mixture of deterministic particle filter model and stochastic ASM model is used, in order to improve convergence and accuracy in lip tracking. Jian *et al.* [95] used an approach called *radial vector*, which is similar to ASMs, but with an implicit labeling of model data. The authors performed the training of their model using particle filtering.

AAMs are a generalization of the ASMs approach, which include not only shape information in the statistical model, but also texture information [96]. The basic formulation of the AAM can be seen in (2.20).

$$\begin{aligned} \mathbf{x} &= \bar{\mathbf{x}} + Q_s \mathbf{c} \\ \mathbf{g} &= \bar{\mathbf{g}} + Q_g \mathbf{c} \end{aligned} \tag{2.20}$$

The term $\mathbf{g}$ represents the texture information contained in the model frame; $\bar{\mathbf{g}}$ represents the mean texture of the model, and Q_s and Q_g are the matrices of the main modes of variation in shape and texture, respectively; and $\mathbf{c}$ is the set of the appearance model parameters. Further information on how AAMs are trained can be found in [96].

An interesting work, in the context of lipreading, is that presented by Matthews *et al.* [97]. A comparison of three methods for representing lip image sequences for speech recognition is made. The first is an ASM lip contour tracker. The second is the ASM extension into an AAM. The last is a multi-scale spatial analysis (MSA) technique. The experiment was performed under identical conditions and using the same data. Like it was expected, better results were obtained when used AAM. Of course, an AAM is an improved ASM with the addition of appearance or texture. The work of Gacon *et al.* for detection of the mouth contour is based on an ASM but introducing two appearance models. One for the appearance corresponding to skin, lips, teeth or inner mouth, and a second for the mouth corners [98, 99]. In a posterior work [99], they focus on the detection of the inner mouth contour, by replacing the cost function used to fit the appearance model. In the previous work it was based on the difference with the real appearance and on the flow of a gradient operator through the curves of the shape. In the last work they used a criterion based on the response of Gaussian local descriptors predicted by using a nonlinear neural network. As an iterative method, AAM

relies on the initialization of shape and could suffer of the local minimum problem. Two possible solutions to this problem are to improve the shape initialization, or to model the local texture. The last solution may result in high computation complexity. Li and Hi [100] propose a AAM based mouth contour extraction algorithm. To reduce the local minimum problem, the algorithm first uses a texture-constrained shape prediction method to perform initialization, an then characterizes the local texture model with classifiers obtained by using Real AdaBoost [101]. Turkmani & Hilton [102] also use AAMs in talking sequences, in order to properly locate inner and outer mouth contours. They extended the basic AAM in order to avoid falling into local minimum, and also to eliminate the need of model reinitialization.

Other Parametric Approaches

Moghaddam & Safabakhsh [103] presented a fast algorithm for outer contour extraction which uses Self Organizing Maps (SOMs). In Khan *et al.* [104], a scheme of specially parametrized active contours–called *level set* representation–is used. The convergence to the lips' contour is achieved in a color representation obtained from the output of a support vector machine, trained to enhance the lip-skin difference from basic RGB. Chang *et al.* [105] also used the *level set* representation in order to approximate the lip contour, introducing additional shape constraints. Xie *et al.* [106] presented a method for lip segmentation which relies in mixing ASMs and cumulative projections, in order to improve the overall robustness in contour detection. In [107], the same principle behind Eigenfaces is used in order to detect and classify the gestures through mouth contour characterization. In the work of Basu *et al.* [38, 39, 40], a triangular 3D mesh model is registered to the face in the images. It uses finite element models in order to constrain the possible movements of the face and, thus, predicting the possible next states of the mouth. Mirhosseini *et al.* [108, 109] used a deformable template basis with shape constrains for outer lip contour tracking. The algorithm is able to adjust the number of control points in the template using hierarchical rules.

2.1.4 Performance measurement in mouth segmentation tasks

Most of lip detection and parametrization algorithms are created in order to supply valuable information to higher level processes like audio-visual speech recognition [25]. In that sense, much of the work in measuring detection quality is guided to reflect the application's specific performance rather than image segmentation, focusing most of the time in determining contour error rather than segmentation error. However, some measures have been tested in order to measure at which extent an algorithm is performing properly in lip segmentation.

Even when some of these measures can be somehow comparable, there is a concurrent lack of generalization in the reported results. In some cases, the measures are taken based on a limited set of images which have a selected condition such as specific lighting, presence/absence of shadows, presence/absence of beards, etc. In spite of this, a brief comparison between some of the methods is shown in Table 2.1.

It can be noticed that most of the measurement methods can be classified in two sets: a contour-based measurement, and a region based measurement. Measurements belonging to the first category quantify features such as point-to-point distances, model-to-point distances

Table 2.1: Methods for lip segmentation: measurement and comparison.

Method	*Type of Measurement*	*Test Data*	*Compared against*
FCMS - Wang *et. al.* [24]	Inner and Outer Lip Error (ILE & OLE).	Detailed results for two images.	FCM.
Liew *et. al.* [110]	• Overlap between regions. • Segmentation error.	70 test images taken from XM2VTS [111] and AR [112] databases.	The method itself.
FCMS - Leung *et. al.* [75]	• ILE & OLE, as in [24]. • Segmentation error.	Detailed results reported for three images; brief results for 27.	FCM, CT[1], LL[113], ZM[19]
MS-FCMS - Wang *et. al.* [80]	Same as [75].	Detailed results reported for three images.	FCM, LL[113], ZM[19]
RoHiLTA - Xie *et. al.* [106]	Annotated contour model error.	150 images taken from AVCONDIG database (cited in [106]); 50 of them with beard or shadows.	The method itself.
CT - Eveno *et. al.* [1]	Segmentation error.	Detailed results reported for three images.	The method itself.
Eveno *et. al.* [2]	• Normalized tracking error. • Normalized human error.	Results reported for 11 sequences (300 images); several annotations per image, performed by different subjects.	The method itself.
Hammal *et. al.* [69]	Same as [2].	Same as in [2].	The method itself.
Bouvier *et. al.* [43]	Same as [2].	Results reported for 12 sequences (450 images).	Eveno's work[2]; Gacon's work[45].
Guan [14]	Segmentation error as in [110].	Detailed results reported for four images; brief results for 35 images.	FCM, CGC.
Khan *et. al.* [104]	Quality of segmentation.	Results reported for 122 images.	The method itself.

and model deviations[2]. In the other hand, region-based measurements usually derive from computations extracted from the confusion matrix; common measures include the true positive rate (TPR, also known as sensitivity) and true negative rate (TNR, also known as specificity); othre measurements such as the Dice or Jaccard indexes can also be extracted from the confusion matrix. In [114] the authors present a contour-based measurement developed to counter the inconsistency exhibited by TPR and TNR under scale changes. Its computation requires several manually annotated versions of each image, which are not always available, to obtain significant results. Also, small global deviations in contour location are reflected by non-negligible increases in error.

2.2 Experimental workflow

The challenge of segmenting mouth structures for mouth gesture detection is addressed in depth through the course of this document, following the guideline given in Figure 2.5. This Section contains a detailed description on how each one of those steps is treated in this document.

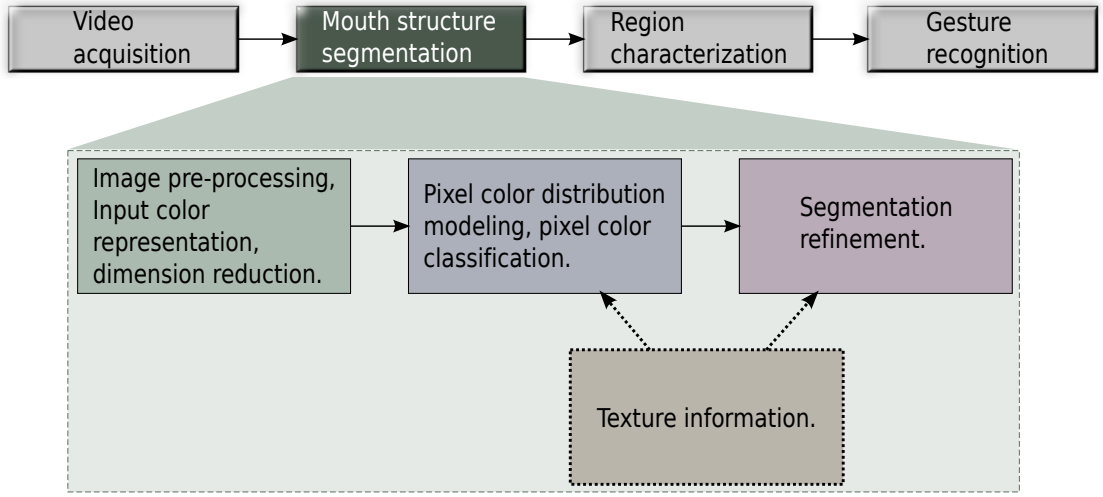


Figure 2.5: Mouth gesture detection in video sequences, as addressed in this work.

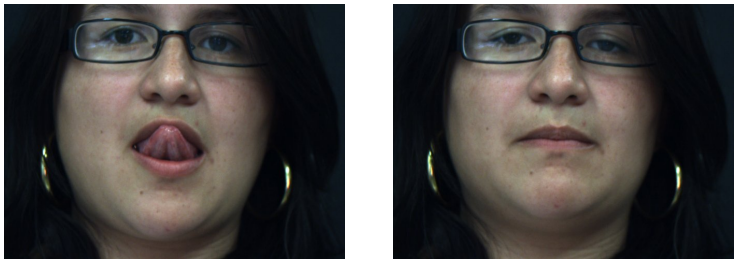
In the previous Section, it is shown how input color representation is exploited in order to highlight mouth structures among them, notably focused in lip/skin separation challenge. Each of the components that has been used traditionally to enhance such differences has its particular advantages and disadvantages that condition the level of accomplishment one may get by using it. Thus, in this work the individual discriminant capability of each component is measured using a "one against the rest" evaluation scheme per each mouth structure separately. Then, *FLDA* is used in order to find a three-some set of projection vectors that enables an input space dimension reduction prior to pixel color modeling of mouth regions. Results on comparing the mapping capabilities of the *FLDA* reduced input space against the full featured input space are exposed in Chapter 3.

[2]A clear example of contour based error measurement is presented in [2].

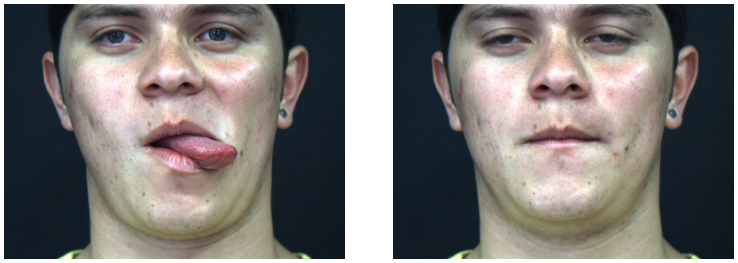
Besides input representation, image pre-processing plays a key role in image segmentation. Pre-processing improves compactedness of color distributions while increasing the signal-to-noise ratio. In this work, two classical approaches for general-purpose image pre-processing are tested: linear low pass Gaussian filters and non-linear statistical Median filters. Once again, results regarding such experiments are presented throughout Chapter 3.
In the document, image segmentation basis restricts to the field of pixel color classification using statistical modeling, namely K-Means and Gaussian mixture based. Their performance and mapping capabilities are tested in the task of mouth structure segmentation in images by direct comparison with Feed forward neural networks, using a scheme which resembles those in [34] and [35]. All of these approaches exploit the potential advantages resulting from using the selected input color transformations altogether. Several combinations regarding pre-processing and input space dimension reduction are tested, including brief yet clear conclusions regarding the experiments.
Once the segmentation benchmark is established, the resulting label images are post-processed using a new segmentation refinement strategy which is introduced in this Thesis. The strategy uses a perceptual unit array that process the input label iteratively, generated an increasingly improved version of the original labeling. The technique, as well as the results obtained by its application in part of the databases, are presented in Chapter 4.
The use of texture descriptors are introduced in Chapter 5, first as input features complimentary to color in the color distribution modeling engine, and then as a tool to automatize the segmentation refinement process. Performance is measured for both variations in terms of computational complexity and error reduction.
Aiming to settle down the previously discussed individual results, overall performance is tested in a mouth gesture detection application. The results of this experiment are presented and discussed in Chapter 7. As in the previous Chapters, both computational complexity and error measurements are provided.
It is noteworthy that in this experimental workflow lip contour tracking is excluded. The main reason for which tracking is not taken into account is that even in 50 frame-per-sec video sequences there are very rapid mouth movements which cannot be accurately followed by the techniques, and in some cases, mouth structures disappear completely from one frame to another, as shown in Figure 2.6. Mouth segmentation is assumed to happen at every frame in the sequences instead.

2.2.1 Database description

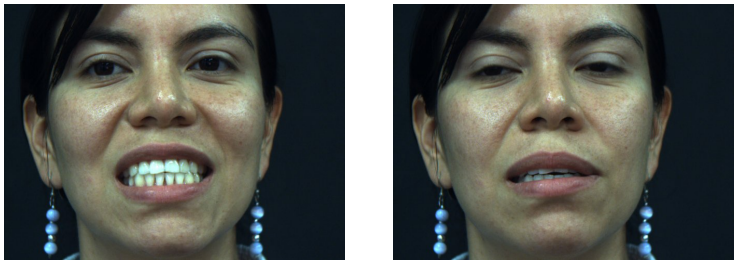
Aiming to establish a common base for comparison against work presented in literature, two well known databases were used. The first one, identified as BSDS300 [114], contains a set of training and testing images in both color and grayscale. Each image has at least one corresponding manually annotated ground truth. This database is particularly used in illustrative examples in Chapter 3, and for general purpose segmentation refinement measurement in Chapter 4.
The second database, named Color FERET (Facial Recognition Technology) database [115, 116], contain facial images taken from several subjects covering a wide variety of skin colors, poses and illuminations. Since this database has no associated ground truth, an image subset



(a) Tongue up to rest transition.



(b) Tongue right to rest transition.



(c) Teeth display to rest transition.

Figure 2.6: Examples of consecutive images in facial video sequences. Sequences were grabbed at 50fps.

was randomly selected and a manually annotated ground truth was established. These images are mainly used in pixel color distribution modeling robustness measurement in Chapter 3.
Besides the aforementioned databases, a proprietary database was constructed. The database is referred as "Own", and its contents is described subsequently.

The "Own" database

The first portion of the database contains images and sequences of five subjects speaking to the camera a pre-defined set of phrases and commands. This database was initially acquired for the work in [4, 117]. Data was acquired using a JVC video camera working at a native resolution of 720x480 pixels, at a frame rate of 25fps. There was no particular illumination setup, neither to control or compensate ambient lighting. Therefore, varying illumination conditions may be found. Particularly, most of the database present upper-left lighting, generating shadows in the right side of the faces, and under the nostrils and the lower lip.
The last portion of this database is conformed by video sequences of 16 subjects, male and female, imitating three different combinations of mouth gestures. All considered mouth gestures are clustered in seven different groups: resting mouth position (R), open mouth (O), close mouth showing the teeth (Th), tongue up (TU), tongue down (TD), tongue left (TL) and tongue right (TR). The gesture combinations are identified as follows:

- *Init* (initialization sequence): R - Th - R - O - R - TH - R - O - Th - R.
- *CCWS (counter-clockwise square)*: R - TR - R - TU - R - TL - R - TD - R.
- *CWLDC (clockwise left-down cube[3])*: R - TL - R - Th - R - TU - R - O - R - TR - R - TD - R - Th - R - TL - R - O - R - TU - R.

Video sequences were acquired in format YUV4:2:2 with a resolution of 658x492 pixels, and at frame rates of 12 and 50 fps. The acquisition setup comprised two diffused, unbalanced spotlights, and a Basler scout scA640-70fc video camera. Environmental lighting was compensated but uncontrolled, by the means of the spotlights. Daytime related lighting variations were allowed. Some sample images extracted from this database can be seen in Figure 2.6.

2.2.2 Error, quality and performance measurement

In spite of the reasons exposed in [114] that instruct to avoid region based confusion matrix based measures in favor of boundary based measures, the former is still the most common benchmark in image segmentation tasks[4]. In this work, Sensitivity and specificity are extensively used for rating classification and segmentation performance [5]. Sensitivity, also known as True Positive Rate (TPR), can be denoted by

$$TPR = p(A^*|A)/p(A) \tag{2.21}$$

[3]The actual shape described by the sequence does not match exactly a left-down cube.
[4]The technique in [114] was developed for the specific task of image segmentation evaluation and is not extendable to any classification problem; moreover, it requires several ground truth annotations per image. This fact makes it unsuitable when a large image set is to be tested.
[5]Both measurements are used in balanced classification tasks. For unbalanced classes, Dice index and Balanced Error Rate are used.

where $p(A^*|A)$ is the probability of a pixel being correctly labeled as to belong to class A, whilst $p(A)$ is the probability of the class A.
Now let $B = A^C$, therefore $p(B) = p(A^C) = 1 - p(A)$. The *True Negative Rate* (TNR), also known as specificity, can be defined in terms of class probabilities as follows:

$$TPR = p(B^*|B)/p(B) \tag{2.22}$$

where $p(B^*|B)$ is the probability for a pixel being correctly classified as to belong to A^C.
Prior information about the application may help in determining risk factors which weight the responsibility of TPR and TNR in data analysis. Nevertheless, in cases when risk does not sets a compromise between the former measures, it is helpful to establish a composite measure that unifies the effect of sensitivity and specificity in one value[6]. One way to accomplish this is by using the *Distance to Optimal*, or DTO (see Figure 2.7). The DTO is the euclidean two dimensional distance between the point (TPR, TNR) and the optimal value $(1, 1)$. Conversely, this definition translates to

$$DTO = \sqrt{(1 - TPR)^2 + (1 - TNR)^2} \tag{2.23}$$

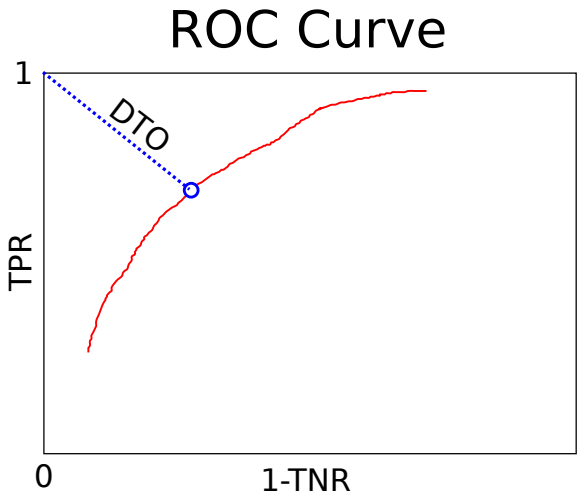


Figure 2.7: Geometrical interpretation of DTO using the Receiver Operating Characteristic (ROC) curve.

Notice that DTO values range in the interval $[0, \sqrt{2}]$, with zero being the optimal value. In that sense, DTO can be viewed as a non normalized measure of error rather than a performance measure. Absolute class confusion, given by $p(A^*|A^C) = 1$, generates $DTO = \sqrt{2}$; uniform random selection like a coin flip, given by $p(A^*|A) = p(A^*|A^C)$, leads to $DTO = 1$.
In multiclass problems both A and A^C may be seen as super-classes that may contain data from more than one class. As in example, the "one against the rest" evaluation scheme confronts data from one class against data from all the remaining classes.
Whenever possible, algorithm complexity is presented using the asymptotic notation $\mathcal{O}(\cdot)$

[6]When risk factor is taken into account, one may want to favor TPR over TNR or viceversa. In example, it is preferable to diagnose more false positives than false negatives in epidemiology and disease control.

with the center point being replaced by an order relationship. In some cases, Tables with parametrized computation times are provided.

2.2.3 Ground truth establishment

For each image database used in this work a subset of images was randomly selected, and manually annotated. Some of them were annotated by more than one evaluator in order to establish human variability. Every segmented image contains labeling information for Lips, Teeth and Tongue, as well as a mouth region of interest (RoI).
Also, the video sequence frames in the second part of the "Own" database are associated with one of the gestures in the set described in Section 2.2.1. As in the later case, one of the sequences was annotated by more than one evaluator in order to measure human-related variability in label selection.
Human perception is a definitive factor in ground truth establishment, and thus it cannot be neglected as a topic of exploration. Thereupon, the next issue discusses human-related variability and consistency in manual image segmentation and gesture detection in video sequences.

Image segmentation ground truth variability

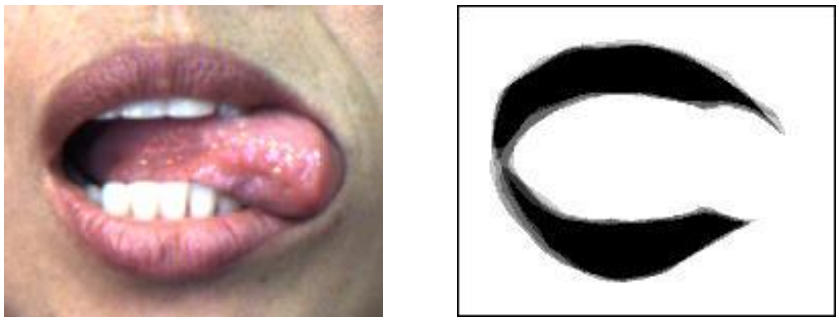
Manually annotated ground truth in image segmentation tends to vary from one evaluator to another. It is seldom taken into account the variability in the appraised segmented images. In this work, two measures are used in order to establish how consistent or variable the evaluators' advise tends to be for the "Own" database.
The first measure, denoted simply as hv, reflects the consistency evidenced among evaluators for a given region in the image, regarding all the pixels that have been chosen as to belong to such region. hv can be defined as

$$hv = p(x \in \cap_A)/p(x \in \cup_A) \tag{2.24}$$

where $p(\cap_A)$ stands for the probability of a given pixel x to be selected as to belong to region A by all subjects, and $p(\cup_A)$ represents the probability of a given pixel x to be selected as to belong to region A by at least one subject. A second measure, identified as hm, reflects the level of overlap in human perception among regions. The measure is defined by

$$hm = p(x \in \cap_A | x \in \cup_B)/\mathsf{min}(p(x \in \cap_A), p(x \in \cap_B)) \tag{2.25}$$

where $p(A|\cup_B)$ represents the conditional probability of a given pixel x to be selected as to belong to region A by at least one subject while being selected as to belong to a region B by one or more different subjects for every pair of disjoint classes A and B.. The two measures account a reflection of consistency in ground truth selection in pixel classification.
Table 2.2 indicates the value of hv for the "Own" database, for a total of five evaluators who provided three annotations to the same image each (the image can be seen in Figure 2.8).
Table 2.3 is the analogous of Table 2.2 for hm. It can be concluded that the task of manually stating the boundaries between mouth structures in images is not straightforward, as the consistency in region selection remained below 70% in average. In example, from all pixels



(a) Original image. (b) Lip segmentation.

(c) Teeth segmentation. (d) Tongue segmentation.

Figure 2.8: Human variation in manual ground-truth establishment. Darker regions mean higher consistency in pixel labeling selection; white represents the background.

labeled as to belong to Lip region by any evaluator, only the 69.4% were selected as to belong to such region by all of them. A similar analysis can be derived from the Table for Teeth and Tongue, for whom the measure achieved 65.2% and 64.67% respectively.

Table 2.2: Human variability measurement in manual mouth structure segmentation, measured using hv.

	Lips	Teeth	Tongue
hv	0.6940 (69.4%)	0.6520 (65.2%)	0.6467 (64.7%)

Table 2.3 can be used to interpret how much each of the structures can be interpreted by a human evaluator as to belong to a different one. The measures do not include the degree of confusion between the regions and the background; this, however, can be deduced by the combined results of Tables 2.2 and 2.3.

Table 2.3: Human variability measurement in manual mouth structure segmentation, measured using hm.

	Lips, Teeth	Lips, Tongue	Teeth, Tongue
hm	0.0386 (3.86%)	0.0080 (0.8%)	0.0254 (2.54%)

The values presented in the Tables should be taken into account when interpreting the results

in mouth structure segmentation provided in the following Chapters.

Gesture detection ground truth variability in video sequences

There is a noticeable difference in the way humans perceive gestures, mostly in transitional video segments in which one gesture is slowly transforming into another. Thus, it is important to establish how consistent human perception proves to be when segmenting gestures in video sequences.
In this test, one *CWLDC* sequence taken from the last portion of the "Own" database was provided to five evaluators. The evaluators were asked to establish the frame ranges in which a given gesture from the set described in Section 2.2.1 is a clear match. The results of the experiment were measured using the aforementioned hv measure, and are presented in Table 2.4. In the test, hm equals to zero for every gesture combination. Average hv equals 0.9355, which indicates high consistency among evaluators.

Table 2.4: Human variability measurement of gesture classification in video sequences.

	Tongue up	Tongue down	Tongue left	Tongue right	Open	Teeth
hv	0.9534	0.9955	0.9087	0.9494	0.9171	0.8890

2.3 Summary

This Chapter serves as an extended introduction, treating concisely but sufficiently the topic of mouth structures segmentation, as well as giving a methodological background concerning the tests and measurements used in the remainder of the document.
Mouth structure segmentation techniques are approached in a rather taxonomic manner, despite all possible methodological combinations and the sometimes weakly defined limits between categories, following the scheme in Figure 2.1. In most cases, special notices about benefits and drawbacks of the techniques are exposed. Special focus is given to color distribution modeling through linear optimization using $FLDA$ and non-linear modeling using mixtures of Gaussians.
The second part of the Chapter contains the description of the databases selected for training and testing purposes in this work, and the basic methodological background needed to unveil the results presented in the remaining of the document. Human-related ground truth variation is also treated from an objective point of view, and presented accompanying the database description.
The measures presented in Tables 2.2 and 2.3 give important notice about the fact that human factor introduces a level of uncertainty in ground truth establishment that cannot be neglected or isolated when interpreting automatic segmentation results. A relatively low consistency in ground truth consistency evidenced in Section 2.2.3 makes contour based error and performance measurements, like the ones in [1, 2, 114], unsuitable for mouth structure segmentation.

3 Pixel color classification for mouth segmentation

In the previous Chapter, several approaches for segmenting mouth structures in images were discussed. Those approaches are categorized in pixel color classification techniques, region based techniques and contour based techniques.
Region based segmentation techniques usually give good results when compared against color classification and contour detection. They, however, include a series of incremental updates and local connectivity tests which make them relatively expensive in machine time, and quite hard to parallelize. Shape constraints help improving their accuracy when working with a restrained set of mouth poses, but resulting in a loss of robustness.
Contour-based techniques, in the other hand, are fast segmentation techniques which may outperform other approaches in both quality and speed. They base their operation in maximizing the gradient flow through a contour located in the image. Their convergence is conditioned to the gradient flow function, which is often prone to suffer from local minima problems.
The remaining alternative, pixel color classification, is usually fast and leads to good results when image lighting could be predicted at some extent and, therefore, compensated. Poor results may be obtained for images acquired under varying lighting conditions, even when using the so called chromatic invariant color transformations. Despite such evident disadvantage, pixel color classification is more appropriate in cases in which pose may vary considerably among images, and when scale of the features in the image is unknown.

Figure 3.1: Image segmentation scheme based in pixel color classification.

In this Chapter, pixel color classification based mouth structure segmentation is treated. Figure 3.1 shows the basic sequence followed in pixel color based segmentation, in the same order they are addressed in the Chapter[1].
The remainder of this Chapter is organized as follows: Section 3.1 treats the application of optimal linear modeling of class separation for mouth structures using Fisher Linear Discriminant Analysis ($FLDA$), as well as the effect of linear and non-linear pre-processing in such process. An alternative for fast and coarse lips segmentation is also introduced. Section 3.2 discuss the use of Gaussian mixtures for modeling mouth structure color distributions. Important results are also presented in this Section, which are used as a benchmark in the majority

[1]Color representation and pre-processing stages may be freely interchanged.

of the remaining tests in the document. A brief summary and final notes are presented in Section 3.3.

3.1 Color representations for mouth segmentation

It has been reported in literature that skin and lip color distribution overlap considerably [27]. Figure 3.2 shows that such issue extends to all visible mouth structures, making them difficult to classify.

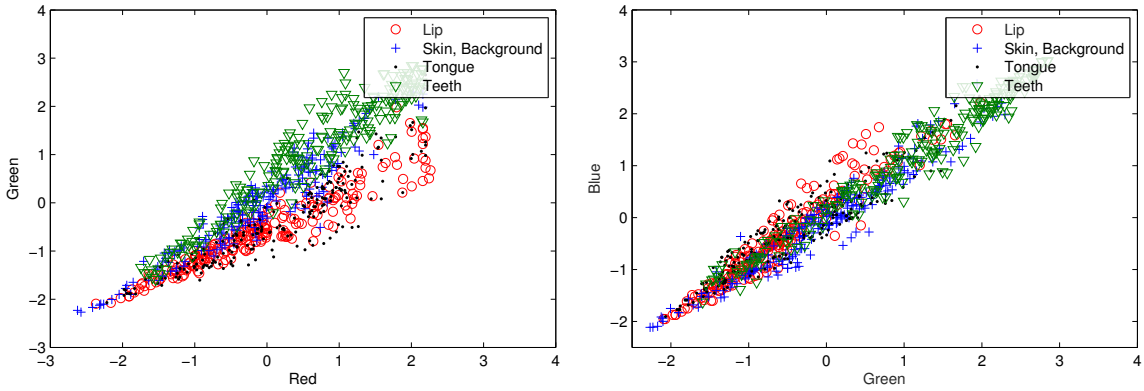


Figure 3.2: RGB distribution of mouth structures.

Nonetheless, it is also proven that combinations of 'weak' features - even linear - may lead to highly discriminant representations [118][2].
In this work, twelve color components, which are common in skin and lip segmentation tasks, are used for pixel color classification: Red, Green, Blue, Hue, Saturation, Value (defined as $\max(R, G, B)$), CIE$L*$, CIE$a*$, CIE$b*$, CIEu', CIEv' and Pseudo-hue[3]. Any linear combination of the base twelve components is avoided, as they do not provide extra information for the analysis[4].

3.1.1 Discriminant analysis of commonly-used color representations

In order to prove the effectiveness of the *FLDA* in selecting a good input color representation for mouth structure pixel classification a data set containing annotated pixel color data was selected. The data come from sixteen facial images randomly selected from the last portion of the "Own" database, acquired under compensated lighting conditions, aiming to preserve the same head pose but not the same gesture. Image color is treated using the *Grey-edge* color constancy method described in [119].
Each pixel is represented using twelve color representations which are commonly used in literature. Since component distribution and value range may differ greatly from one component to

[2]One approach to find optimal linear transformations in the sense of linear separability among classes is achieved through *Fisher Linear Discriminant Analysis*. This method is described in Section 3.1.1.
[3]In lip/skin segmentation, Hue component is usually rotated by $30°$ in order to concentrate reddish hue in a compact range.
[4]The restriction excludes color representations such as $YCbCr$, and the C_3 component of the Discrete Hartley Transform.

Table 3.1: Normalization factors and individual discrimination capabilities in mouth structure segmentation for several color representations.

	Reg. Parameters		Ind. Class. Capability ($\overline{DTO}$)		
	Mean	Std. Dev.	Lip	Teeth	Tongue
Red	0.5676	0.1982	0.6890	0.6976	0.8291
Green	0.4568	0.1861	0.7201	0.7548	0.8750
Blue	0.4427	0.1837	0.7010	0.7549	0.8378
Hue	0.4044	0.4200	**0.5603**	0.9161	**0.5921**
Saturation	0.2650	0.1269	0.7605	0.8134	0.8320
Value	0.5718	0.2008	0.6852	0.6956	0.8174
CIE $L*$	51.2836	18.6664	0.7069	0.8895	0.6549
CIE $a*$	11.2164	9.4382	0.5800	**0.5787**	0.8469
CIE $b*$	6.6904	7.5667	0.7050	0.7837	0.8356
CIE u'	0.2410	0.0246	0.8232	0.9102	0.7461
CIE v'	0.4951	0.0119	0.6489	0.6566	0.9624
Pseudo-hue	0.5600	0.0456	0.7957	0.9324	0.7572

another, each one of them was normalized using the mean and standard deviations consigned in the first two columns of Table 3.1. A benchmark for pixel classification was also established using the "one against the rest" approach [120] for each input component, measuring $\overline{DTO}$ for lip, teeth and tongue regions; the results of the experiment are shown in the last three columns of Table 3.1. In the table, the best components' $\overline{DTO}$ for classifying each region is presented in bold.

The projection spaces given by $\boldsymbol{w}_{Lip}$, $\boldsymbol{w}_{Teeth}$ and $\boldsymbol{w}_{Tongue}$ will, from now on, be used as the preferred color representation in further tests.

Table 3.2 contains the *FLDA* projection vectors for Lip, Teeth and Tongue regions, computed using the "one against the rest" approach. Classification performances in the projected spaces, measured using $\overline{DTO}$, are shown in Table 3.3. Training performance was tested using the same amount of data for both the testing class and "the rest"; such quantity was limited to cover the 30% of the total number of elements contained in the class subset which has the less elements (around 6000). Testing performance, in the other hand, was measured using all the input patterns contained in the selected image set. Its noteworthy that all regions are much more easily classifiable using the *FLDA* projection space than any of the original input representations; this can be effectively seen by comparing $\overline{DTO}$ measures in Tables 3.1 and 3.3. Mean $\overline{DTO}$ obtained using *FLDA* based projection and thresholding for the three regions was 0.2726 (average $\overline{DTO}$ in Table 3.3, corresponding to an 80% in combined TPR and TNR).

Features set covariance of the data, represented using the color spaces in Table 3.2, is shown in Figure 3.3. In the figure, white blocks represent positive values while black blocks represent negative values. Magnitudes are proportional to block area. Notice a high correlation between the red, green, blue, value, and CIE$b*$ components; also notice the high correlation between hue and CIE$L*$ and CIE$a*$. Covariance matrices are seldom used as feature selection tools; however, it has been proved that two or more variables that appear to be correlated may reach

Table 3.2: *FLDA* projection vectors and comparison thresholds for mouth structure classification.

	$\boldsymbol{w_{Lip}}$	$\boldsymbol{w_{Teeth}}$	$\boldsymbol{w_{Tongue}}$
Red	-14.9876	16.2401	-9.9318
Green	30.9664	-10.6025	17.7132
Blue	-16.7017	-1.1409	-3.9080
Hue	0.1505	-0.0264	-0.0903
Saturation	-1.0330	-0.5130	0.3971
Value	4.6850	0.4752	-1.2842
CIE $L*$	-4.6476	-3.9766	-3.0053
CIE $a*$	9.5569	-8.4148	6.7015
CIE $b*$	-4.8199	-1.2962	0.2016
CIE u'	-2.3136	6.2434	-1.5410
CIE v'	2.1365	-1.0480	-0.5272
Pseudo-Hue	1.7660	-4.6361	0.6532
Threshold	0.7788	0.6908	0.6714

Table 3.3: Projection classification performance, using the "one against the rest" approach.

	Lips	Teeth	Tongue
$\overline{DTO}$	0.2844	0.2258	0.3228
$\sigma_{\overline{DTO}}$	0.0018	0.0012	0.0021

higher classification performances when used altogether.

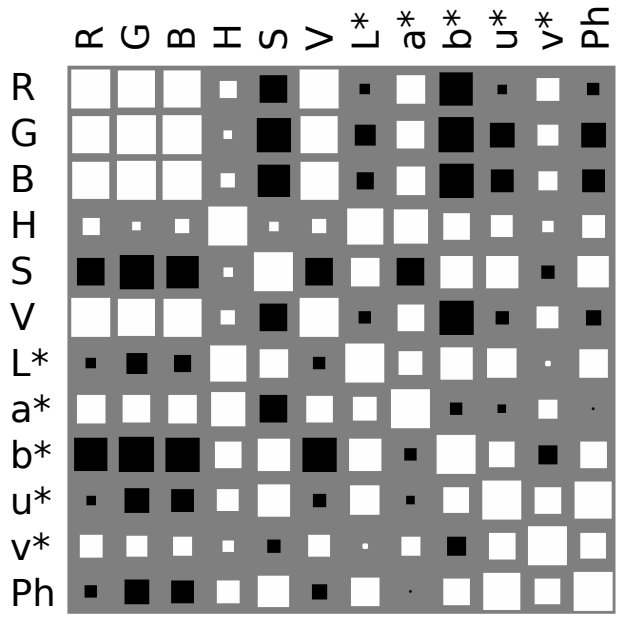


Figure 3.3: Covariance matrix codified in blocks. White blocks represent positive values, while black blocks represent negative values. Element magnitude in the matrix is represented proportionally regarding block area.

3.1.2 Effect of image pre-processing in FLDA

Digital image quality suffer the influence of several factors that distort the scene realization made by the sensor. Those distortions can be reflected in changes of relative shape and size of the objects in the image (like barrel distortion), chromatic changes due to lens phasing or sensor noise, etc.. Hence, techniques have been developed aimed to cope with each one of those problems. Shape and chromatic distortion correction usually requires an accurate lens model, or at least some key points in the image that help in establishing color and shape references. However, noise reduction can be achieved without prior knowledge about the acquisition setup[5].

The most common approach for tackling with noise reduction problem is the application of low pass linear and non-linear spatial filtering [5]. In this work, Gaussian linear low pass filters and non-linear stochastic Median filters are used.

The median filter is $\mathcal{O}(n)$ regarding the total number of pixels in the image, and typically $\mathcal{O}(n^2 \log n)$ regarding filter's mask width[6].

Particularly, separable linear filters can be computed by decomposing the 2-D/2-D convolution operation with two 2-D/1-D convolution operations, using a vertical and a horizontal kernel. Hence, separable linear filters are $\mathcal{O}(n)$ regarding the number of pixels of the input image,

[5]Nevertheless, the availability of prior information about the acquisition configuration would lead to better results in noise reduction.

[6]The computational complexity of the median filter varies depending the complexity of the implementation of the underlying sorting algorithm.

and $\mathcal{O}(n)$ regarding mask width (unlike non-separable filters, which are $\mathcal{O}(n^2)$ regarding mask width). Specialized CPU and GPU operations are able to handle per cycle floating point addition-multiplication operations, which dramatically improve computational time for mask widths equaling or below the floating point register size.

Figure 3.4 exposes the effect of filtering in pixel color classification, measured using mean DTO averaged over Lip, Teeth and Tongue. The experiment was conducted using the one-against all *FLDA* configuration for the regions. The analysis was repeated 50 times, each time using around 6000 input patterns from every region, as well as the background. The patterns were randomly selected from 16 images, for whom a manually annotated RoI is provided, each one corresponding to a different subject[7].

In 3.4a, the blue dashed line represents the base mean DTO measurement, with a value of 0.2726. The red solid line with round markers show the mean DTO obtained for patterns selected from filtered images using separable, centered Gaussian low pass filters, varying in each case the mask size. The solid black line with inverse triangle markers represent the same measure for a median filter whose mask size was varied alike. Notice that mean DTO tends to lower as the mask size increase. This trend stops when the filter's cut frequency surpasses the size of certain features in the image, like teeth, mouth corners, etc.; in that case, these structures get excessively blurred by the filter's softening effect, effectively worsening the DTO (see median filter behavior for mask widths above 11 pixels, or Gaussian filter behavior for mask widths above 25 pixels). DTO variation throughout trials remained stable for mask widths below 23 pixels for both Gaussian and median filters, as shown in Figure 3.4b. In that range, its value turn around 0.036 and 0.046. Following the results in the figure, a good quality/performance compromise can be achieved by choosing a 9×9 Gaussian filter. Using that selection, an improvement above 18% in DTO can be expected[8].

It is noteworthy that image filtering is a scale-variant operation, hence being affected by feature size in pixels. Results obtained using filtering should be accompanied with detailed descriptions about the acquisition process and a feature size indicator. In this test, mouth width is used as scale indicator; mouth width ranges between 138 and 216 pixels, with mean and standard deviation of 170.88 and 20 pixels, respectively. For all testing purposes, noise is assumed to have a Gaussian distribution, and to be spatially and chromatically uncorrelated.

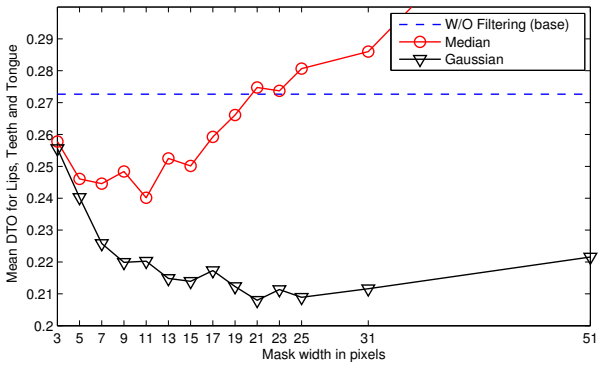
3.1.3 Case study: the normalized $a*$ component

In some cases, the computational complexity implied in computing several input color representations for pixel color classification cannot be afforded (this can be true notably for high-speed color classification). In this Section, a simpler alternative for lip and teeth segmentation in images is presented. The technique, based in the CIE$a*$ color representation, was developed as a part of a methodology for mouth gesture detection in images in video [4].

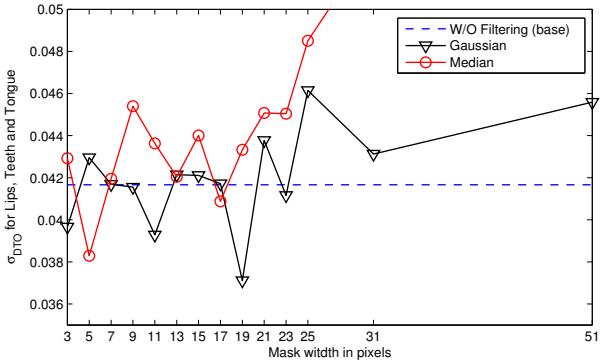
The $a*$ distribution for lip and skin regions tends to vary from one image to another–though at some extent disjoint among them, making it unsuitable for segmentation with static threshold

[7] Only eight images were taken into account in each turn, allowing a greater variation in DTO among turns.

[8] Such improvement can be obtained if the same specific testing conditions are used ("one against the rest" scheme, equal number of patterns for all classes). The improvement is measured using $(DTO_{base} - DTO_{improved})/DTO_{base}$.



(a) Color classification performance, measured using $\overline{DTO}$ (lower is better).



(b) Color classification performance variation among trials ($\sigma_{\overline{DTO}}$).

Figure 3.4: Effect of pre-processing in mouth structure color classification.

selection. That variability decays when the color distribution of the mouth region and its immediate surroundings in normalized, as shown in the following experiment. First, a set of twenty facial images taken from our facial image database were used. The images were manually segmented in order to facilitate threshold establishment and error measurement.
A rectangular ROI was manually established for each image used in the test. The ROI encloses all mouth structures, extended by around a 10% of the mouth width at each side. In order to set a benchmark, an optimal threshold was computed for each image represented in the $a*$ color component by minimizing the classification error using the training data. The experiment was repeated using a normalized $a*$ color representation of the input patterns (see Algorithm 1), with and without RoI selection. Table 3.4 shows the result of the experiment.

Algorithm 1 RGB to $a*$ color transformation.

Require: P (Training data, in RGB), L (labeling information associated to P), N (number of elements in P), RoI.
Ensure: $A' = a'_1, a'_2, , a'_N$ (normalized $a*$ representation of P).
  for $i = 0$ to N **do**
    $a_i \leftarrow a*$ representation of $\boldsymbol{p}_i$;
  end for
  $\overline{a} \leftarrow \left(\sum_{i=1}^{N} a_i\right)/N$;
  $\sigma_a^2 \leftarrow \left(\sum_{i=1}^{N} (\overline{a} - a_i)^2\right)/N$;
  for $i = 0$ to N **do**
    $a'_i \leftarrow (a_i - \overline{a})/\sqrt{\sigma_a^2}$;
  end for

Table 3.4: Mean threshold and threshold variances for the training data set.

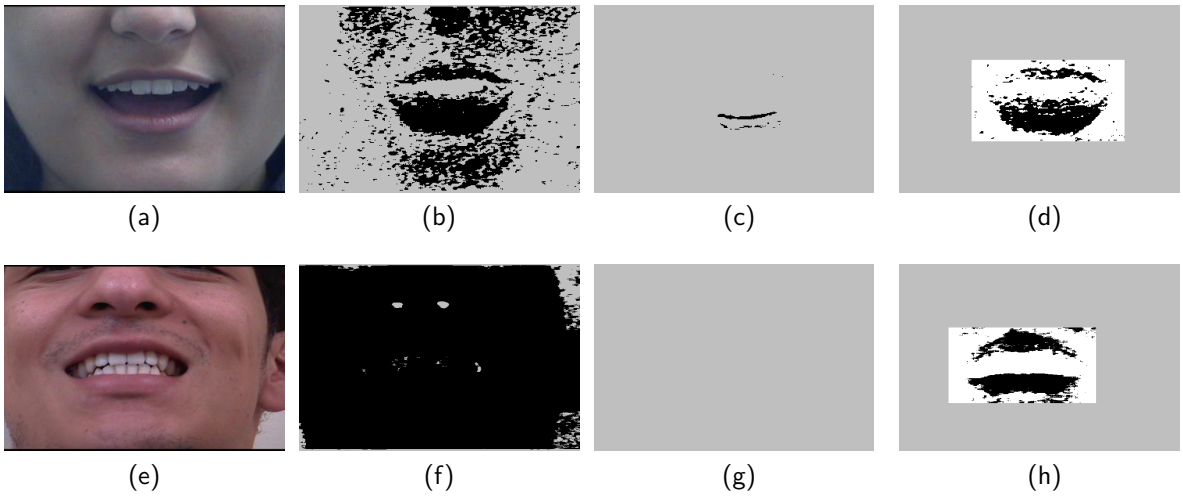
Base component	Mean ($\overline{th}$)	Variance (σ_{th}^2)
$a*$	4.9286	2.95202
Normalized $a*$ w/o RoI	1.4634	0.2301
Normalized $a*$ with RoI	0.6320	0.2016

Notice that threshold variance decays when ROI is set up prior to threshold selection. If RoI is not initialized, the *a** component behavior is close to that of the standard *a** component. Computing the later is a fully-parallel operation at pixel level, while the former requires serial calculations (specifically in the mean and variance calculation).
In the second test, the threshold set was averaged for both $a*$ and normalized $a*$ component, and they were later used to segment the test image set. Table 3.5 shows the results of this experiment, in terms of TPR, TNR and DTO. Notice that TPR value dropped drastically when the $a*$ component was normalized without caring about the RoI clipping. The corresponding decay in segmentation quality is exemplified in Figures 3.5c and 3.5g. The best compromise, reflected with the lowest mean DTO and a relatively small DTO deviation, was obtained using the normalized $a*$ component.
Evidence of the stability gain in threshold selection using the normalized $a*$ images is reflected in the segmentation process, as seen in Figures 3.5d and 3.5h. The relatively small threshold

Table 3.5: Lips-skin segmentation performance.

Base component	$\overline{TPR}$	σ_{TPR}	$\overline{TNR}$	σ_{TNR}	$\overline{DTO}$	σ_{DTO}
$a*$	0.9244	0.0597	0.6033	0.3129	0.4140	0.3019
Normalized $a*$ w/o RoI	0.0427	0.0392	0.9998	0.0004	0.9573	0.0392
Normalized $a*$ with RoI	0.7013	0.0725	0.9769	0.0032	0.2996	0.0725

deviation for normalized $a*$ color representations using RoI clipping indicates that threshold value can be safely chosen to be 0.632 for a wide range of facial images, expecting a mean TPR, TNR and DTO conforming to those in Table 3.5.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 3.5: Comparison of pixel color based lip segmentation: a, e) original images; b, f) lip segmentation using the $a*$ color representation, with $th = 4.9286$; c, g) lip segmentation using the normalized $a*$ color representation without RoI clipping, with $th = 1.4634$; d, h) lip segmentation using the normalized $a*$ representation with RoI clipping, with $th = 0.632$.

The $a*$ component normalization introduces important changes in computational complexity when compared to computing plain $a*$. Notably, calculating the mean and variance of the data inside the RoI is only partially parallelizable. In a massive-parallel SIMD[9] platform, computational complexity associated with $a*$ color representation can be downscaled from $\mathcal{O}(n)$ to $\mathcal{O}(1)$, regarding the number of pixels in the image. In the other hand, the normalized version of the $a*$ color representation can only be reduced from $\mathcal{O}(n)$ to $\mathcal{O}(\log(n))$, regarding the total number of pixels inside the RoI. It is also noteworthy that the normalization can become an important source of error if the RoI is nor properly selected, as suggested by the results in Table 3.5.

[9]Acronym for Single-Instruction, Multiple-Data.

3.2 Gaussian mixtures in color distribution modeling

The assumption is that a good approximation of $p(x)$ can be obtained through a weighted superposition of K Gaussians probability densities, as in

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{3.1}$$

where $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is a $\boldsymbol{\mu}_k$-centered Gaussian with covariance matrix given by $\boldsymbol{\Sigma}_k$. The values of π_k indicates the responsibility of the k^{th} Gaussian in representing the data distribution. The best fit to the input data, determined by the parameter set $\Theta = \{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$, can be found by maximizing the log likelihood function

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\} \tag{3.2}$$

Detailed information on how to perform the maximization of (3.2) can be found in Section 3.2.2. Initial values for the parameter set are usually obtained through the K-Means algorithm discussed in the next Section.

Gaussian mixture models can be used to model pixel color distributions in both supervised and unsupervised manners. In the first case more than one GMM shall be used, each one approximating the color distribution of the data belonging to a particular class. In unsupervised color distribution modeling one makes the assumption of no prior knowledge about the classes present in the image, and one GMM is used to approximate the image color histogram. In this case, classes may be assigned to subsets in the mixture (down to one Gaussian per class).

When GMMs are used to model class color distributions, new input patterns may be classified by looking for the model that generates the highest probability in the data.

3.2.1 The K-Means algorithm

The K-Means algorithm is a statistic modeling technique which aims to represent a data set distribution by the means of a centroid set. Each centroid in the feature space gather a subset of data points through the use of a distance measurement. This principle can be coded in the cost function depicted in (3.6). This function is also known as *distortion measure*, and quantifies the dispersion from a given data $\mathbf{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ regarding the centroid set $\boldsymbol{M} = \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K\}$.

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2 \tag{3.3}$$

The value of r_{nk} is used to associate each pattern or data point $\boldsymbol{x}_n$ with a centroid $\boldsymbol{\mu}_k$. In basic

K-Means, this coefficient is constrained to codify a hard class assignment for each pattern:

$$r_{nk} = \begin{cases} 1 & \text{if } k = \text{argmin}_j \|\boldsymbol{x}_n - \boldsymbol{\mu}_j\|^2 \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

The problem of fitting a K-Mean model to a data set can be regarded as to find a set of K centroid locations in the feature space given a data set. Due to the constrained form of r_{nk}, the problem of minimizing J in terms of $\boldsymbol{\mu}_k$ present a closed-form solution

$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk}\boldsymbol{x}_n}{\sum_n r_{nk}} \tag{3.5}$$

It can be seen that $\boldsymbol{\mu}_k$ becomes the mean of the patterns associated with the $k^{th}-$cluster (hence the name of K-Means). Convergence is achieved by alternating iteratively the steps in (3.4) and (3.5) until a stop criterion(a) is(are) met [121]. Since $\boldsymbol{x}_n$ is an euclidean variable and r_{nk} is also established in an euclidean space, the approximated cluster shape is either symmetric, circular, spheric or Hyper-spheric, for the case of higher-dimensional data. The floating parameter K is a pre-set in the algorithm, although its value can be refined during optimization like in the ISODATA algorithm [122]. Some extensions of the K-Means algorithm, such as the K-Medoids algorithm, use other dissimilarity measures rather than the euclidean distance in the *distortion measure* formulation, transforming (3.6) into

$$\tilde{J} = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\mathcal{V}(\boldsymbol{x}_n, \boldsymbol{\mu}_k) \tag{3.6}$$

It is possible to use K-Means modeling in order to approximate color distribution in both supervised and unsupervised manner. In the first case, the labeling information associated to the training data can be used in order to approximate a whole K-Means model for each class, thus obtaining as much models as labels are in the prior. Once converged, new data are classified by associating them to the model for which the euclidean distance is minimal. For unsupervised color modeling, only one model suffices to approximate the color distribution of the image, and the efforts are usually concentrated in establishing a proper value for K.

3.2.2 Gaussian mixture model estimation using Expectation-Maximization

A K-Means model can be regarded as a particular case of a Gaussian mixture where all covariance matrices equal to $\alpha\mathbf{I}$. Therefore, posterior parameter tunning is needed for mixtures initialized using K-Means in order to take benefit from Gaussian approximation potential.
The optimization of the likelihood function in (3.2) can be carried using the Expectation-Maximization method. In the expectation (E) step, current parameter values are used to evaluate the posterior probabilities given by (3.7). In turn, the maximization (M) step use such posterior probabilities in order to update the means, covariances and mixing coefficients, according to (3.8), (3.9) and (3.10). Iterative alternation between E and M steps ensures parameter convergence while maximizing the likelihood function, as shown in [123].

A summary of the expectation-maximization method applied to Gaussian mixtures extracted from [123] is shown in Algorithm 2.

Algorithm 2 Expectation-Maximization for Gaussian Mixtures (taken from [123]).

1. Initialize the means $\boldsymbol{\mu}_k$, covariances $\boldsymbol{\sigma}_k$ and mixing coefficients π_k, and evaluate the initial value of the log likelihood.

2. **E Step.** Evaluate the responsibilities using the current parameter values

$$\gamma(z_{nk}) \leftarrow \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \tag{3.7}$$

3. **M Step.** Re-estimate the parameters using the current responsibilities

$$\boldsymbol{\mu}_k^{\text{new}} \leftarrow \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \boldsymbol{x}_n \tag{3.8}$$

$$\boldsymbol{\Sigma}_k^{\text{new}} \leftarrow \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) (\boldsymbol{x}_n - \boldsymbol{\mu}_k^{\text{new}})(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{\text{new}})^{\mathsf{T}} \tag{3.9}$$

$$\pi_k^{\text{new}} \leftarrow \frac{N_k}{N} \tag{3.10}$$

with

$$N_k \leftarrow \sum_{n=1}^{N} \gamma(z_{nk}) \tag{3.11}$$

4. Evaluate the log likelihood

$$\ln p(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) \leftarrow \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\} \tag{3.12}$$

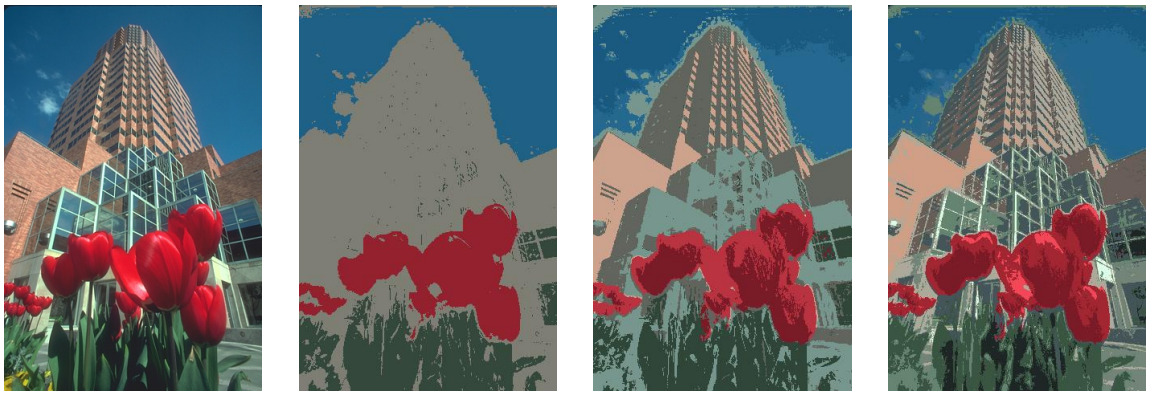
and check for convergence of either parameters or the log likelihood. If the convergence criterion is not satisfied return to step 2.

3.2.3 Case study: Color distribution modeling of natural images

Figure 3.6 shows the effect of grouping pixels by color using the K-Means algorithm, and then representing each cluster by its corresponding centroid. This approach can be used for color compression or image segmentation [123].
Figure 3.7 shows the analogous effect evidenced in Figure 3.6, this time using Gaussian mixtures.
Gaussian mixture based classification imposes the calculation of (3.1) for every new input pattern. Unlike the K-Means evaluation, this operation is of order $\mathcal{O}(n^2)$ regarding the input pattern dimension. Hence, it is always desirable to reduce the input representation

(a) Original image. (b) Compressed image with K=4. (c) Compressed image with K=8. (d) Compressed image with K=16.

Figure 3.6: Example of image color compression using K-Means pixel color modeling.



(a) Original image. (b) Compressed image with K=4. (c) Compressed image with K=8. (d) Compressed image with K=16.

Figure 3.7: Example of image color compression using Gaussian Mixture based pixel color modeling.

space (feature space) before carrying out the modeling process. Evaluating a D-dimensional Gaussian mixture with K Gaussians is approximately $2D \sim 3D$ times more expensive than evaluating a D-dimensional K-Means model with K centroids.

3.2.4 Mouth structure segmentation using K-Means and Gaussian mixtures

In this Section, K-Means modeling and Gaussian mixture modeling is put to test in the task of mouth structure segmentation. The feature space is comprised by the same twelve color representations enunciated in Section 3.1.1, contrasted with the three *FLDA* vector projections obtained in the same Section. Those representations serve to conform a 12-dimensional input feature space in the first case and a three-dimensional input feature space in the second case.
For the tests, a set of 16 images taken from the last portion of the "Own" database, coming from 16 different subjects, was used. The images were selected covering most of the gestures described in Section 2.2.1, in a way that pixels of every structure are contained in at least six of them[10].
Also, in order to establish the effect of pre-processing in color modeling of mouth structures, the configuration selected in Section 3.1.1 is used. Each test image was represented using four combinations: 12-feature non-filtered, 12-feature filtered, 3-feature non-filtered and 3-feature filtered. The 3-feature representations are obtained by projecting the 12-dimensional data using the *FLDA* vectors in Table 3.2.
As a special notice, mind that using three features instead of twelve reduces the number of parameters of each K-Means centroid from twelve to three, and those of each Gaussian from 91 to ten[11]. These numbers should be multiplied by the number of centroids (in the case of K-Means) or Gaussians used for a given model.
First of all, a proper value for the parameter K, which controls the number of clusters, should be established. The easiest way to obtain such value is carried out by sweeping the parameter's value over a range of possibilities, measuring classification performance in each case. In order to improve result significance several models were trained for each parameter combination, and then the classification results were averaged.
Table 3.6 exposes K-Means color classification performance for mouth structures. The results include the combination of three and twelve dimensional input feature spaces with and without pre-processing filtering, using data from within RoI and the whole image. In all cases, filtered versions show improvements in averaged DTO regarding the unfiltered versions. Notice that averaged DTO is bigger in value for Lips and Teeth regions when using data inside RoI than in the case of data coming from the whole image. This effect is due to an decrease in TNR, indicating a higher overlap between those regions' models and the background color model in mouth surroundings. The mean averaged DTO for the three regions was 0.3712 for 12-dimensional feature input vectors, and 0.3953 for 3-dimensional feature input vectors.

[10] Unlike Teeth and Tongue regions, Lip region is present in the 16 test images.

[11] Total parameter count is 157 for 12-dimensional Gaussians and 13 for 3-dimensional Gaussians. However, due to symmetry in the corresponding covariance matrices, the actual number of parameters is reduced to 91 and 10 respectively. Further reduction can be achieved if using non-rotated Gaussians, where only 25 and 7 parameters are needed.

Table 3.6: K-Means based pixel color classification: performance measurement using $\overline{DTO}$. Upwards arrows indicate improvement.

		12 Features			3 Features		
		Lip	Teeth	Tongue	Lip	Teeth	Tongue
Whole	Unfilt.	0.4170	0.4275	0.5873	0.4423	0.4174	0.6214
	Filt.	**0.2722 (↑)**	**0.3243 (↑)**	**0.4936 (↑)**	**0.3167 (↑)**	**0.2894 (↑)**	**0.5074 (↑)**
Clipped	Unfilt.	0.5055	0.4878	0.5220	0.5265	0.4580	0.5656
	Filt.	**0.3407 (↑)**	**0.3632 (↑)**	**0.4599 (↑)**	**0.3718 (↑)**	**0.3289 (↑)**	**0.4852 (↑)**

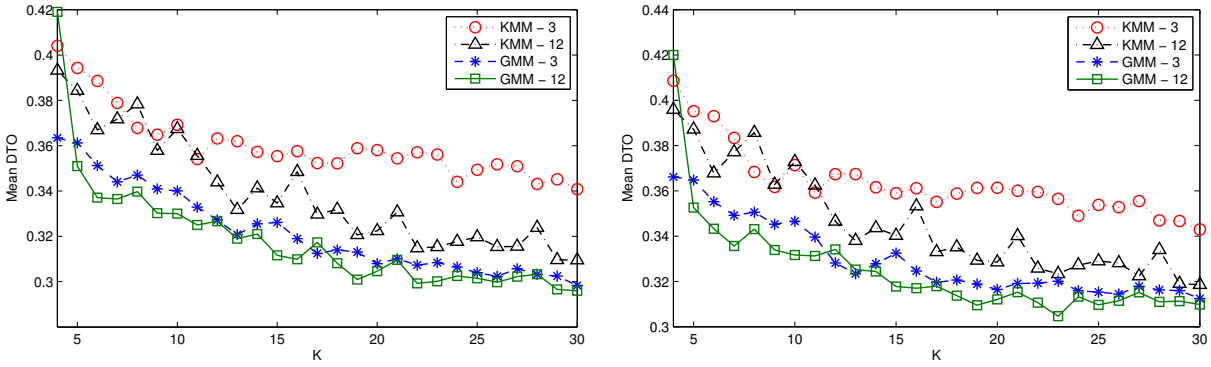
Analogously, Table 3.7 presents the results of mouth structure segmentation using Gaussian mixture model based color classification instead of K-Means. Once again, filtered versions show improvements in averaged DTO regarding the unfiltered versions. Mean averaged DTO for the three regions was 0.3621 for 12-dimensional feature input vectors and 0.3565 for three-dimensional input feature vectors, besting in both cases the results obtained using K-Means by a narrow margin. Figure 3.9 illustrates some results from the experiment conducted for both K-Means and Gaussian mixtures using two sample images.

Table 3.7: Gaussian mixture based pixel color classification: performance measurement using $\overline{DTO}$. Upwards arrows indicate improvement.

		12 Features			3 Features		
		Lip	Teeth	Tongue	Lip	Teeth	Tongue
Whole	Unfilt.	0.4977	0.4195	0.5302	0.5153	0.4192	0.5698
	Filt.	**0.2911 (↑)**	**0.2959 (↑)**	**0.4673 (↑)**	**0.3276 (↑)**	**0.2849 (↑)**	**0.4738 (↑)**
Clipped	Unfilt.	0.4876	0.4728	0.5228	0.5106	0.4454	0.5643
	Filt.	**0.3103 (↑)**	**0.3335 (↑)**	**0.4244 (↑)**	**0.3397 (↑)**	**0.2851 (↑)**	**0.4446 (↑)**

A summary of the accuracy obtained after carrying out the experiment can be seen in Figure 3.8. Notice that both sides of the Figure exhibit great similarity for most combinations; however, DTO achieves lower values for training data than for testing data. One can conclude that model overfitting has not been reached up to $K = 30$ since there is no noticeable increase in DTO for any of the modeling engines. As expected, the best results were obtained for Gaussian mixture modeling using three and twelve input features. DTO value starts to settle for $K > 20$; thereby, K value is set up at 20 for the rest of the tests in the Chapter[12].
Gaussian mixtures modeling capability was compared against that of feed-forward artificial neural networks (FFNNs), in the task of mouth structure color distribution modeling (as in [34, 35]). In the case of 12-feature input vectors a FFNN with one hidden layer with 428 neural units and four output units was used; for 3-feature input vectors, a FFNN with one hidden layer with 100 units and four outputs was used. Both network architectures were selected to match approximately the number of parameters of the corresponding GMM.
The networks were trained using the resilient backpropagation algorithm [124] using the same training data as in the case of GMM approximation. Results of this experiment are presented in Table 3.8. Notice that FFNNs perform a better color classification for 12-dimensional feature

[12]Having twenty Gaussians per model implies the estimation of 1820 parameters for 12-dimensional feature input vectors, and 200 parameters for three-dimensional feature input vectors.



(a) Training mean DTO for lips, teeth and tongue. (b) Testing mean DTO for lips, teeth and tongue.

Figure 3.8: Training and testing DTO measured for each mouth structure, regarding the number of centroids or Gaussians per model.

Table 3.8: FFNN based pixel color classification: performance measurement using $\overline{DTO}$. Upwards arrows indicate improvement.

		12 Features			3 Features		
		Lip	Teeth	Tongue	Lip	Teeth	Tongue
Whole	Unfilt.	0.5335	0.4843	0.4875	0.5037	0.5118	0.5020
	Filt.	**0.4151 (↑)**	**0.3545 (↑)**	**0.3995 (↑)**	**0.3687 (↑)**	**0.4601 (↑)**	**0.5049 (↓)**
Clipped	Unfilt.	0.4919	0.4326	0.5115	0.5184	0.4197	0.5468
	Filt.	**0.3363 (↑)**	**0.2939 (↑)**	**0.4134 (↑)**	**0.3886 (↑)**	**0.3139 (↑)**	**0.5126 (↑)**

input vectors than the one obtained with Gaussian mixtures; however, Gaussian mixtures outperform FFNNs using three dimensional feature input vectors.

In order to test the robustness of the GMMs, the structure classification was repeated using images from the FERET database and the second portion of the "Own" database. Mind that the images from the FERET database contain all the head and neck of the subjects, and in some cases the upper part of the torso, thus the mouth region is a very small portion in the images (around 5% of the pixels in the image). Color models were not re-trained neither adapted for the new data. Results of this experiment can be seen in Table 3.9.

Table 3.9: Robustness test - Color FERET database. Values measure $\overline{DTO}$.

		12 Features				3 Features			
		Lip	Teeth	Tongue	Avg.	Lip	Teeth	Tongue	Avg.
K-Means	Whole	0.6239	0.8613	0.9835	0.8229	0.6580	0.9431	0.9287	0.8433
	Clipped	0.6385	0.7664	0.9910	0.7986	0.6620	0.9315	0.9397	0.8444
GMM	Whole	0.6299	0.8512	0.9263	0.8025	0.6483	0.8566	0.9487	0.8179
	Clipped	0.6442	0.7689	0.9591	0.7907	0.6302	0.7904	1.0274	0.8160
FFNN	Whole	0.4918	0.8903	0.9855	0.7892	0.8570	0.9969	0.7263	0.8601
	Clipped	0.5072	0.8642	0.9867	0.7860	0.8574	0.9641	0.8877	0.9030

From Table 3.9, it can be concluded that pixel color classification error increased significantly by using a completely new database. This effect can be issued to variable lighting conditions present in the FERET database and the increased size and color variability of the background,

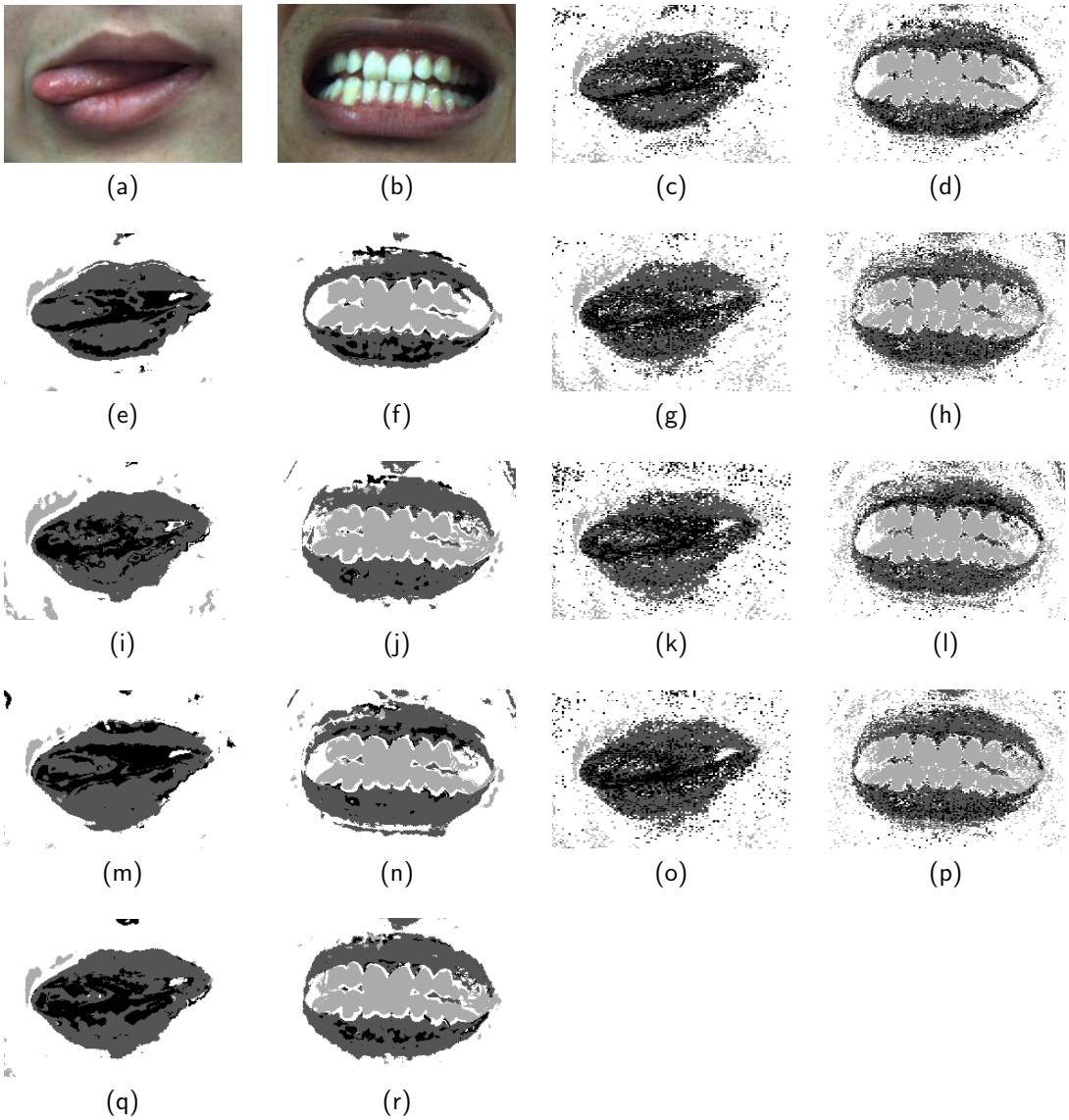


Figure 3.9: K-Means based and Gaussian mixture based pixel color classification examples. K-Means based: 3.9a, 3.9b Original images; 3.9c, 3.9d result using a 12-feature input space; 3.9e, 3.9f result using a 12-feature filtered input space; 3.9g, 3.9h result using a 3-feature input space; 3.9i, 3.9j result using a 3-feature filtered input space. GM based: 3.9k, 3.9l result using a 12-feature input space; 3.9m, 3.9n result using a 12-feature filtered input space; 3.9o, 3.9p result using a 3-feature input space; 3.9q, 3.9r result using a 3-feature filtered input space.

among other factors. TNR decreased considerably using the new data, which reflects negatively in DTO. As in the previous tests, FFNN bested Gaussian mixtures and K-Means by a narrow margin for 12-dimensional feature input vectors, and Gaussian mixtures bested K-Means and FFNN for three dimensional feature input vectors.

3.3 Summary

This Chapter presents a study in pixel color classification of mouth structures in facial images. The first part of the Chapter focuses in studying the individual and conjoint discriminant capabilities of several color components that have been used to tackle the aforementioned task. In the second part of the Chapter, these representations are used as features in stochastic modeling engines trained to model the color distributions of each visible mouth structure in the images.

Neural networks proved higher accuracy in color distribution modeling when using a 12-dimensional input feature vector than Gaussian mixtures. This effect is reversed when only three features were used. There is a considerable reduction in computational complexity when downscaling from 12 features to three; at the same time, a barely noticeable decrease in accuracy was obtained by performing that change. Thereby, results presented in following chapters will be referred to a three-dimensional feature Gaussian mixture model. The models use the configuration described in Section 3.2.4.

From the tests conducted over "Own" database and Color FERET database, it can be concluded that changes among databases can produce higher variations in color classification performance than changes among subjects in the same database. The tests clearly illustrate the complexity of isolating the influence of issues related to acquisition set-up from the final color register in the images.

As a side contribution of the study conducted in this Chapter, a fast alternative for coarse lip/skin segmentation based in pixel classification is introduced in Section 3.1.3. The segmentation technique is based in the use of the CIE$a*$ color component, with its value normalized using the values inside the mouth's region of interest (RoI). Classification results proved to be better than those obtained using other color components commonly used in lip/skin segmentation through pixel color thresholding.

4 A perceptual approach to segmentation refinement

Image segmentation produces label images that can be used in higher level processes in perception. Human-like interpretation of the scene is possible when an accurate region detection and characterization is available. When segmenting with pixel-color based techniques, which lack of region and task specific constraints (like connectedness tests), regions usually present jagged borders and holes, and may vary considerably in size and shape, as seen in Figure 4.1.

(a) Original image.

(b) Pixel color classification of mouth structures.

Figure 4.1: Example of pixel color classification based image segmentation. Notice the presence of jagged region borders, unconnected spurious regions and small holes and gaps.

There are some techniques that can be applied to label images in order to refine the labeling. For instance, localized application of binary morphological operations may improve region definition by smoothing jagged areas around the borders or filling small gaps between regions[1]. Nonetheless, applying these techniques involve setting up a wide variety of parameters that condition their behavior, like structuring element selection (shape and size), operation ordering (erosion, dilation, combinations of them), etc.. The varying nature of such parameters makes morphological operation selection a task in which tuning for the best results is commonly carried out by the means of heuristic searches or by following guidelines stated in the state of the art. Having all those possible parameter combinations also means that it is impractical to predict or characterize the final output for every combination of operations and structuring elements.

In this Chapter, a new technique for segmentation refinement is proposed. The technique was designed aiming to be both easily understandable, as well as predictable through time, based on a simplistic analogy to the first stage of the human visual perception system. Being predictable

[1] Given that Gray-scale morphological operations are based in ordering relationships that cannot be easily extended to classes, they are not applicable to segmentation refinement straightforwardly.

through time means that one may expect a predictable result when iteratively applying the refinement to an input labeling until convergence is achieved; therefore, the insights of the infinite-time behavior of the refiner are discussed thoroughly. The fundamentals of the method are stated in Section 4.1. Section 4.2 treats topics such as parameter selection and infinite behavior of the refiner through parameter study cases. Sections 4.3 and 4.4 provide some results obtained in segmentation refinement of natural images and mouth structures in images, respectively. Finally, a brief summary of this Chapter in given in Section 4.5.

4.1 Segmentation refinement using perceptual arrays

Image segmentation based in thresholding or pixel color classification gives fast results but compromising in segmentation quality. As discussed before, problems such as jagged edges and small holes or unconnected regions are common results of this kind of image segmentation. Therefore, their usage is usually accompanied by image pre-processing and output labeling refinement. In this section, a biologically inspired method for tackling segmentation refinement is proposed.

According to [125, 126], human nervous system may be viewed as a three-stage system with forward and backward propagation, where a set of receptors transform the input stimuli and propagate them through a complex neural net (the brain), ultimately generating a response through the effectors (please refer to Fig. 4.2). Similarly, the proposed segmentation scheme uses a set of color classifiers that transform visual color information coming from the scene into segmentation labels, thus taking the role of receptors, and a segmentation refiner who acts as the neural net and the effector, in the case where the desired output corresponds to a label image[2].

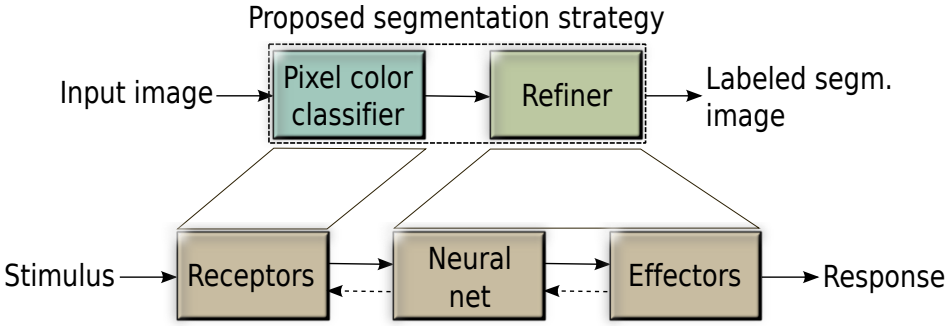


Figure 4.2: Segmentation methodology diagram and the dual block diagram representation of nervous system according to [125].

The refinement technique comprises a layer of organized units with forward, backward and lateral connections. From now on, this layer will be referred as a perceptual array, and its layout mimics the cone and rod distribution in human retina. Each unit in the array is connected with one unique input pixel class labeling, thus equaling the total number of patterns in the input array. The effect of the feedback connections is controlled by the joint influence of two

[2]Explicit feedback is avoided in the segmentation scheme (upper part of Figure 4.2) due to parameter tuning and color modeling being carried out off-line.

parameters: σ, which determines the size of the unit's perceptual field (set of neighboring units), as well as its associated weighting function; and α, which sets the proportion between the lateral and forward influence in the unit output.

The behavior of the perceptual layer is summarized in Equation (4.1). In the equation, P^0 stands for the input pattern set, and P^k and P^{k+1} represent the layer output at iterations k and $k+1$, respectively.

$$P^{k+1} = W\left(\alpha P^0 + (1-\alpha)(P^k * G_\sigma)\right), \tag{4.1}$$

It is noteworthy that P^0, P^k and P^{k+1} represent the class labeling for every pixel in the input image rather than its color or intensity representation. The "Winner" function, denoted by W, is a non-linear operator that generates a "winning" array or vector with the same size of its argument. In the case of pixel labeling, the "Winner" function will select the class which most likely correspond to a given pixel based in an input vector whose component codify the class rating associated to each class. A formal definition of the "Winner" function for vector is explained in Section 4.2, (please refer to formulation in Equation (4.3)). The operator $\cdot * \cdot$ stands for the spatial convolution operation, and G_σ is a centered bi-dimensional Gaussian window with variance σ^2.

It can be immediately noticed the analogy between a low-pass filter-like behavior and that of each iteration of the refiner, where the corresponding smoothing factor is controlled by the combination of parameters $(\alpha,\ \sigma)$. Nevertheless, the actual enhancement potential of the method is met when the process is recalled to convergence; that is, when the calculation of Equation (4.1) is repeated until one or more stop criteria are met, as shown in Algorithm 3. Common criteria are a maximum number of iterations, and a minimum number of label changes between consecutive values of P^k. The computational complexity of each refiner iteration is $\mathcal{O}(n)$ regarding the number of pixels in the image; particularly, its reckoning is slightly higher than the one associated with a linear filter with a window width and height equaling 5σ. This relationship can be established since most of the area contained by a Gaussian function is bound inside the domain range $(\mu - 2.5\sigma,\ \mu + 2.5\sigma)$[3].

Algorithm 3 Segmentation refinement.

Require: P^0, α, σ, *Stop criterion(a).*
Ensure: P^k
  $k \leftarrow 0$
  while *Stop criterion(a) is(are) NOT met,* **do**
    $P^{k+1} \leftarrow W\left(\alpha P^0 + (1-\alpha)(P^k * G_\sigma)\right)$
    $k \leftarrow k+1$
  end while

Thanks to symmetry in G_σ, and bounding its size to match that of the image (that is, $N \times M$

[3]The area contained by a Gaussian function inside the domain range $(\mu - 2.5\sigma,\ \mu + 2.5\sigma)$ is slightly greater than 0.987, which corresponds to a 98.7% of the total area of the curve.

elements), the pixel-wise form of Equation (4.1) can be expressed as

$$\mathbf{p}_{i,j}^{k+1} = \mathbf{w}\left(\alpha\mathbf{p}_{i,j}^{0} + (1-\alpha)\left(\sum_{u,v}\mathbf{p}_{u,v}^{k} g_{u,v,\sigma}(i,j)\right)\right), \tag{4.2}$$

with u, v varying to cover the image size. $G_\sigma(i,j)$ is a bi-dimensional Gaussian window with variance σ^2 centered at (i,j), and $g_{u,v,\sigma}(i,j)$ represents the element (u,v) in $G_\sigma(i,j)$. Notice that $\mathbf{p}_{i,j}$ represents a label vector resulting from a prior pixel classification, not the actual color or intensity of the pixel.

Figure 4.3 shows the pixel-wise principle of the technique. In the diagram, the combination $(\alpha = 0.25, \sigma = 0.6)$ is used. Each color in the colored squares represent a particular class labeling, while gray-shaded squares represent the values of G_σ, centered at i, j. The output block holds the color which corresponds to the winning class, as obtained by using the "Winner" operation.

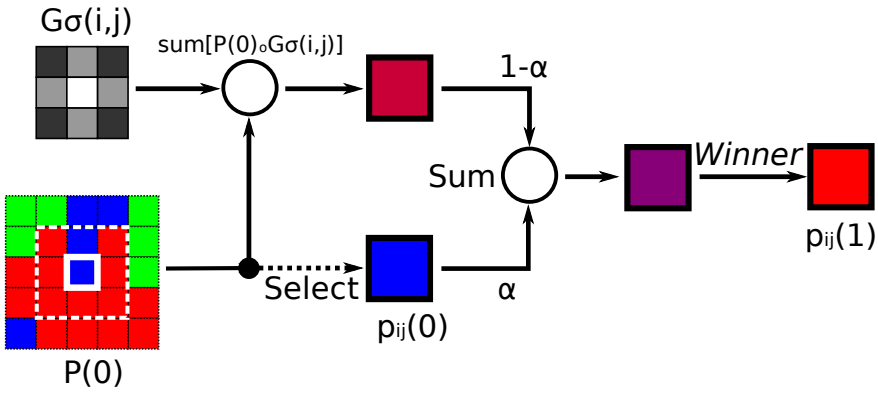
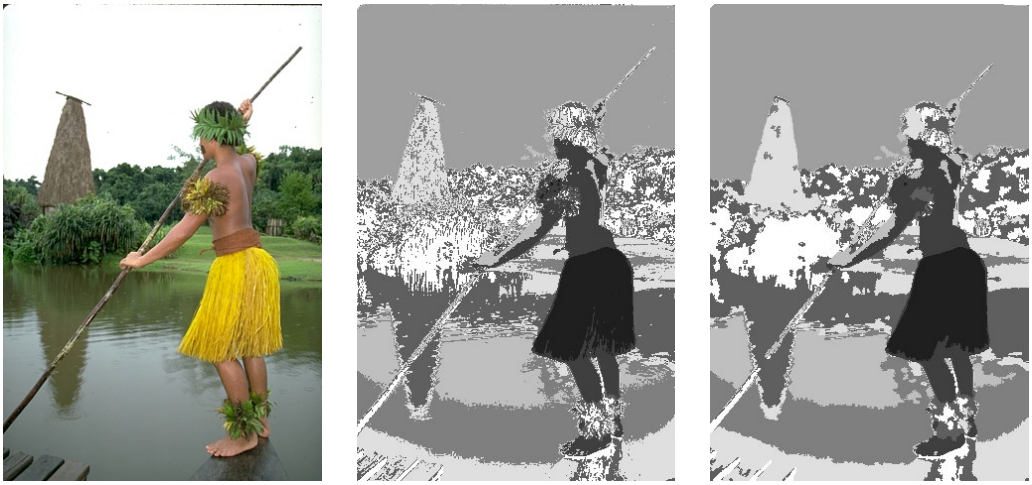


Figure 4.3: Refinement process example diagram for $\alpha = 0.25$ and $\sigma = 0.6$ (3x3 pixels perceptual field size). Colored squares represent class labeling, while gray-shaded squares represent weights.

Figure 4.4 exemplifies the refiner behavior after an image segmentation based in color classification. The analysis is focused in the detail in Figures 4.4d and 4.4e, where two phenomena can be regarded: neighborhoods with high variability in the input labeling do not produce changes in such labeling along iterations, as seen in the subject's right hand; and a clear softener effect in the bright side (left) of the subject's leaf headband. The former is an uncommon behavior that mimics the one of a high pass spatial filter applied to the labeling information. The later, a smoothing effect produced by the refiner, tends to blur or even eliminate small features through iterations depending on the perceptual field size and the weight of the lateral effect. Figure 4.5 shows the effect of varying the refiner parameters when the input is a label facial image with an initial mouth structure segmentation.

4.2 Special cases and infinite behavior of the refiner

In order to clarify the insights of the refinement process some definitions will be stated. First, let I be a given input image, with size $N \times M$ pixels. Each element $\mathbf{q}_{i,j} \in I$ is a vector



(a) Original image, taken from the Berkeley image database [114].

(b) Label image generated using K-Means with k=8.

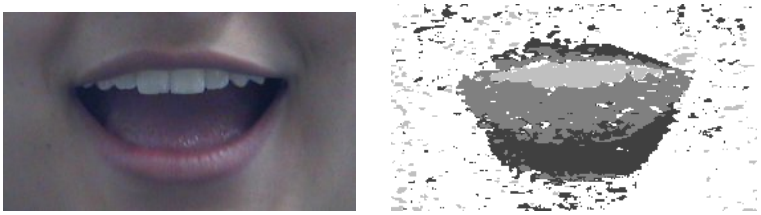
(c) Figure 4.4b after refinement.

(d) Detail of 4.4a.

(e) Corresponding detail in 4.4b.

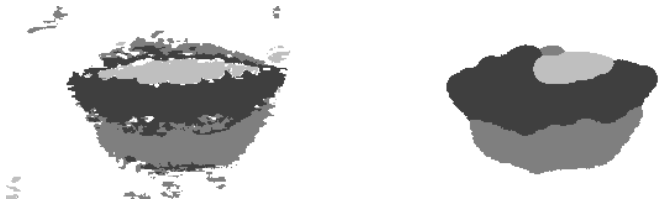
(f) Detail on 4.4b after refinement.

Figure 4.4: Effect of segmentation refinement in a K-Means based color classification.



(a) Original image.

(b) Initial classification image, coded in grayscale.



(c) Result of the refinement process for $(\sigma = 4.0,\ \alpha = 0.4)$.

(d) Result of the refinement process for $(\sigma = 4.0,\ \alpha = 0.1)$.

Figure 4.5: Example of segmentation refinement using a mouth image.

containing the intensity and/or color information of the pixel referred by indexes (i, j). Let P_0 be a vector array with size $N \times M$, containing an initial classification of the information in I. Each element $\mathbf{p}^0_{i,j}$ codifies the class label assigned to the input pattern $\mathbf{q}_{i,j}$. Any possible value for $\mathbf{p}^0_{i,j}$ lies in the set composed by the column vectors of the identity matrix of order C; C standing for the total number of detected classes.
Now, let $\mathbf{w}(\mathbf{p}) = (w_1, w_2, \ldots, w_C)^\top$ be a function from $\mathbb{R}^C$ to $\mathbb{R}^C$, defined by its elements as

$$w_j(\mathbf{p}) = \begin{cases} 1, & \exists! j : j = \text{argmax}_i\{p_i\} \\ indeterminable, & \exists j_1, j_2, j_1 \neq j_2 : j_1, j_2 = \text{argmax}_i\{p_i\} \\ 0, & \text{otherwise} \end{cases} . \tag{4.3}$$

From now on, $\mathbf{w}(\mathbf{p})$ can be referred as the "Winner" vector resulting from $\mathbf{p}$; in the same way, $W(P)$ can be referred as the "Winner" array resulting from vector array P. Namely speaking, the "Winner" vector $\mathbf{w}(\mathbf{p})$ contains zeros in all but one of its components; that component holds a value equal to one, and corresponds to the same component that holds the maximum value in the input vector $\mathbf{p}$. If the result of $\text{argmax}_i\{p_i\}$ is not unique, $\mathbf{w}(\mathbf{p})$ becomes indeterminable[4].
With these priors in mind, it follows that the expressions in Equations (4.1) and (4.2) lead to valid class labels for any value of k if α is bound to $[0, 1]$ and σ keeps its value in the range $[0, \infty)$. The stationary behavior of the refiner can be predicted if constraints for α and σ are imposed. In this section, some of those variations are studied.
First, let $P'^k = P^k - \{\mathbf{p}^k_{i,j}\}$ be a labeling vector set constructed as the array P^k minus the element at location (i, j); then, one can reformulate Equation (4.2) by splitting it as

$$\mathbf{p}^{k+1}_{i,j} = \mathbf{w}\left(\alpha\mathbf{p}^0_{i,j} + (1-\alpha)g_{i,j,\sigma}(i,j)\mathbf{p}^k_{i,j} + (1-\alpha)(\textstyle\sum_{u,v}\mathbf{p}'^k_{u,v}g_{u,v,\sigma}(i,j))\right), \tag{4.4}$$

Intuitively, big values of σ lead to small values of $g_{i,j,\sigma}(i, j)$; particularly, $\sigma \to \infty \implies g_{i,j} \to 1/(nm)^+$, and $\sigma \to 0 \implies g_{i,j} \to 1$. The splat version presented in Equation (4.4), which seems clearer to interpret than the original expression in Equation (4.2), is used as a basis for studying the effect of the parameter set $(\alpha,\ \sigma)$ in conditioning the refiner behavior.

Case 1: $\sigma \to \infty$**,** $\alpha = 0$

In this case the Gaussian window given by $G_\sigma(i, j)$ flattens completely, turning formulation in (4.4) into

$$\mathbf{p}^{k+1}_{i,j} \approx \mathbf{w}\left(\sum_{u,v}\frac{\mathbf{p}^k_{u,v}}{nm}\right) = \mathbf{mode}(P^k), \tag{4.5}$$

for any location (i, j) in the image. Particularly, when $k = 1$, $\mathbf{p}^1_{i,j} = \mathbf{mode}(P^0)$. This generalizes to any value of $k > 0$, as no further changes are produced, and therefore can be regarded as a degenerated case of parameter selection.

[4] In order to reduce the influence of indeterminable values in refiner computation, indeterminable $\mathbf{p}^{k+1}$ are forced to keep their older value; that is, $\mathbf{p}^k$.

Case 2: $\sigma = 0$**,** $\alpha = 0$

In this case, $g_{i,j} = 1$ and $\sum_{u,v} \mathbf{p}'^{k}_{u,v} g_{u,v,\sigma}(i,j) = 0$. Thus, the expression in Equation (4.4) becomes

$$\mathbf{p}^{k+1}_{i,j} = \mathbf{w}\left(\mathbf{p}^{k}_{i,j}\right). \tag{4.6}$$

This particular case leads to no changes between consecutive arrays P^k and P^{k+1}, therefore making $P^k = P^0$ for any value of k. No changes between consecutive labeling arrays mean no refinement at all, thus the combination $(\sigma = 0,\ \alpha = 0)$ is another degenerated case of parameter selection for the refiner.

Case 3: Every $\mathbf{p}^{k}_{u,v} \in P'^{k}$ **equals the mode, while** $\mathbf{p}^{k}_{i,j}$ **does not**

In this case, formulation in Equation (4.4) can be safely changed by

$$\mathbf{p}^{k+1}_{i,j} \;=\; \mathbf{w}\left(\alpha\mathbf{p}^{0}_{i,j} + (1-\alpha)g_{i,j}\mathbf{p}^{k}_{i,j} + (1-\alpha)(1-g_{i,j})\mathbf{mode}(P'^{k})\right). \tag{4.7}$$

By setting $k = 0$,

$$\mathbf{p}^{1}_{i,j} \;=\; \mathbf{w}\left((\alpha + (1-\alpha)g_{i,j})\mathbf{p}^{0}_{i,j} + (1-\alpha)(1-g_{i,j})\mathbf{mode}(P'^{0})\right). \tag{4.8}$$

As both $\mathbf{p}^{0}_{i,j}$ and $\mathbf{mode}(P'^{0})$ are vectors from the identity matrix of order C, the value of $\mathbf{p}^{1}_{i,j}$ is given by

$$\mathbf{p}^{1}_{i,j} = \begin{cases} \mathbf{p}^{0}_{i,j}, & \alpha + g_{i,j} - \alpha g_{i,j} > 1/2 \\ ind, & \alpha + g_{i,j} - \alpha g_{i,j} = 1/2 \\ \mathbf{mode}(P'^{0}), & \text{otherwise} \end{cases}. \tag{4.9}$$

This is the most favorable case for producing changes between $\mathbf{p}^{0}_{i,j}$ and $\mathbf{p}^{1}_{i,j}$, as the elements in P'^{0} do not compete each other but collaborate in introducing that change. In order to ensure that $\mathbf{p}^{1}_{i,j} = \mathbf{p}^{0}_{i,j}$ first condition in Equation (4.9) should be attained. Regardless the value of $\mathbf{p}^{1}_{i,j}$, no further changes in the labeling are be expected, and then

$$\mathbf{p}^{k}_{i,j} = \begin{cases} \mathbf{p}^{0}_{i,j}, & \alpha + g_{i,j} - \alpha g_{i,j} > 1/2 \\ ind, & \alpha + g_{i,j} - \alpha g_{i,j} = 1/2 \\ \mathbf{mode}(P'^{0}), & \text{otherwise} \end{cases}. \tag{4.10}$$

The derived condition $\alpha + g_{i,j} - \alpha g_{i,j} > 1/2$ is clearly established as necessary in order to avoid any change in the labeling; moreover, if the indetermination resulting from the "Winner" function is resolved by avoiding changes, the condition can be extended to $\alpha + g_{i,j} - \alpha g_{i,j} \geq 1/2$. Hence, a necessary condition for valid parameter selection is given by

$$\alpha + g_{i,j} - \alpha g_{i,j} < 1/2, \tag{4.11}$$

subject to avoid the already discussed degenerated parameter combinations.
Notice that the previous analytic study reveals a series of necessary but not sufficient conditions

for the refiner to work properly in label images. Hence, Figure 4.6 shows the result of an empirical study of the effect of σ and α in the refiner behavior through iterations, measured using the number of changes detected between consecutive labeling arrays. Notice that small values of α usually lead to a constant decay through time in the effect of the input P_0 over the output P_{k+1}. It is important to note that such behavior is consistent with the dual alertness-encoding factors in the stimuli-repetition effects theory presented in [127].

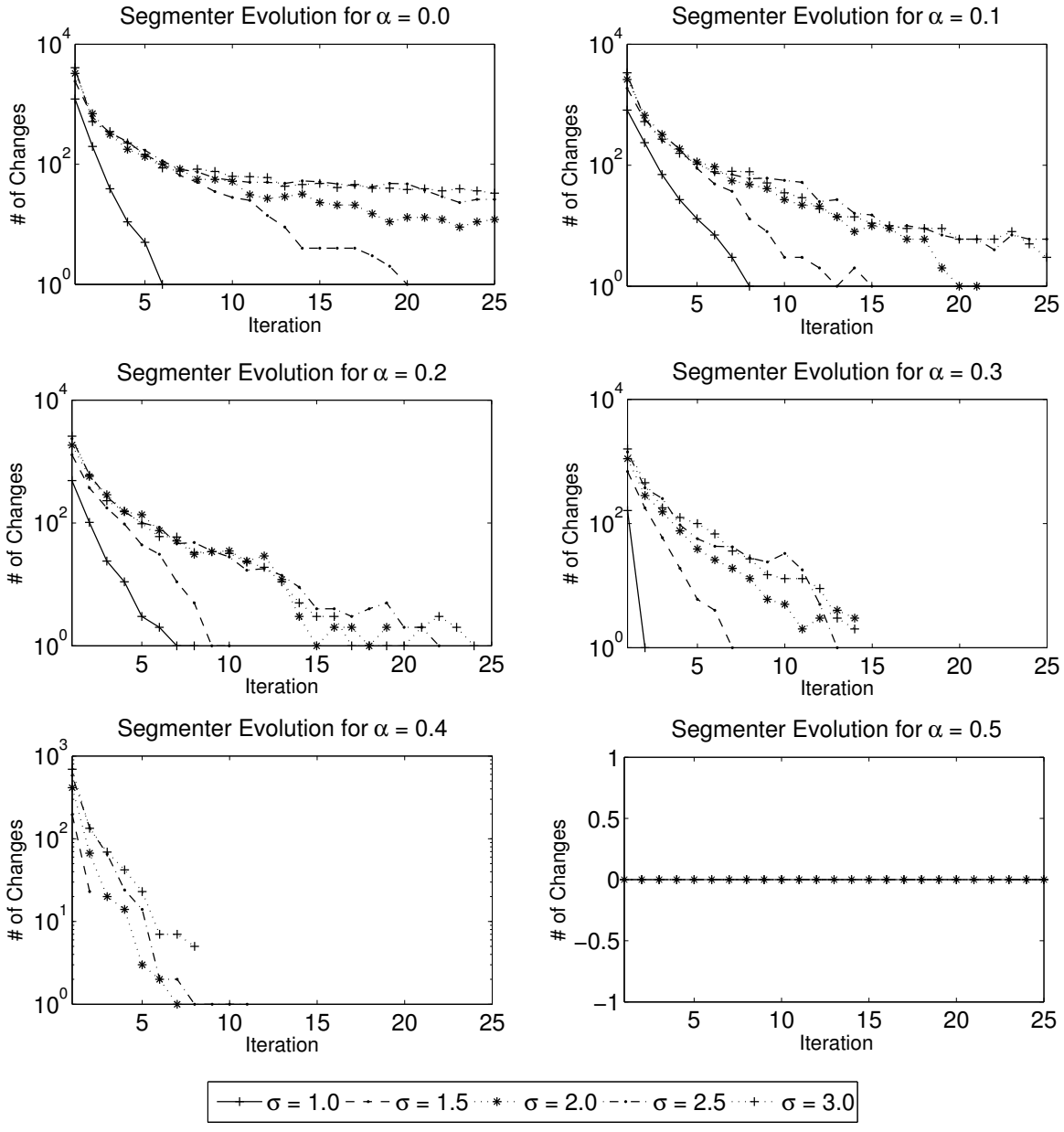


Figure 4.6: Refiner evolution through iteration for diverse parameter combinations.

4.3 Unsupervised natural image segmentation refinement

In order to clarify the effect of label refinement in pixel color based image segmentation a set of tests using natural images were carried out. In the tests, no prior knowledge about

the images is assumed, therefore leading to the application of unsupervised segmentation approaches. The data source selected for the experiment is the "test" subset of the Berekeley's database (BSDS300) [114], which comprises a hundred images along with corresponding manual annotations or ground truths. The BSDS300 database contain more than one ground truth image per source image; hence, TPR, TNR and DTO measures obtained using this database represent mean values rather than absolutes.
Three different techniques were selected to generate the reference segmentation measures: K-Means, Gaussian mixtures and Fuzzy C-Means. In the case of Gaussian mixtures, K-Means and Expectation-Maximization were used to initialize and fine tune the parameter set, like in Chapter 3.

4.3.1 Refiner parameter set-up

First of all, refiner parameters should be set in proper values for the dataset. One intuitive way of tuning σ and α is by minimizing $DTO(\alpha, \sigma)$. Since it is impossible to predict DTO's behavior within a natural image set, a sub-optimal solution can be found by sweeping the parameter space and computing the corresponding DTO value for each combination. Bounded limits for studying parameter influence over refining performance can be established using conditions in (4.10). In all case, the refiner stop criterion was set to either *(a)* reaching 25 iterations, or *(b)* obtaining no changes between consecutive label images.
Figure 4.7 exposes the effect of parameter variation in refinement performance, measured using the DTO, for images in the BSDS300 database. In the figure, darker shades mean lower DTO values and higher performance. The dashed white line shows local minima path of DTO in the parameter space. The lowest DTO value was obtained for $(\alpha = 0.05, \sigma = 1)$; this specific combination creates a relatively small perceptual field for each unit (5×5 pixels in size), and favors changes between iterations. There is a noticeable drop in performance for $\sigma > 5$, which corresponds to perceptual fields with more than 25×25 pixels in size. This effect is quite similar to the one evidenced from the usage of relatively big windows for Gaussian filters and Median filters in image pre-processing, as seen in Section 3.1.2 (reader may refer particularly to Figure 3.4). It can be concluded that important regions in the image are lost once that perceptual field size is surpassed.

4.3.2 Pixel color classification tuning

The three methods share one parameter which controls the desired number of clusters. Its value was swept from three to eight, and the best fit in terms of DTO for each technique in each image was selected. Then, the mean values for TPR, TNR and DTO were computed, along with their associated deviations, as seen in Table 4.1. In the table, upwards arrows indicate improvement, which in the case of DTO, σ_{TPR}, σ_{TNR}, OS and σ_{OS} correspond to values closing to zero. Notice that in most of the cases TPR and OS profit from the refinement process; however, this behavior does not hold for the other measures. Summarizing, the average changes obtained by using the refiner were: an increase of 4.85% in $\overline{TPR}$, a decrease of 2.97% in $\overline{TNR}$, and a decrease of 3.3% in $\overline{DTO}$.
High values in OS and σ_{OS} were obtained using the three segmentation techniques. This

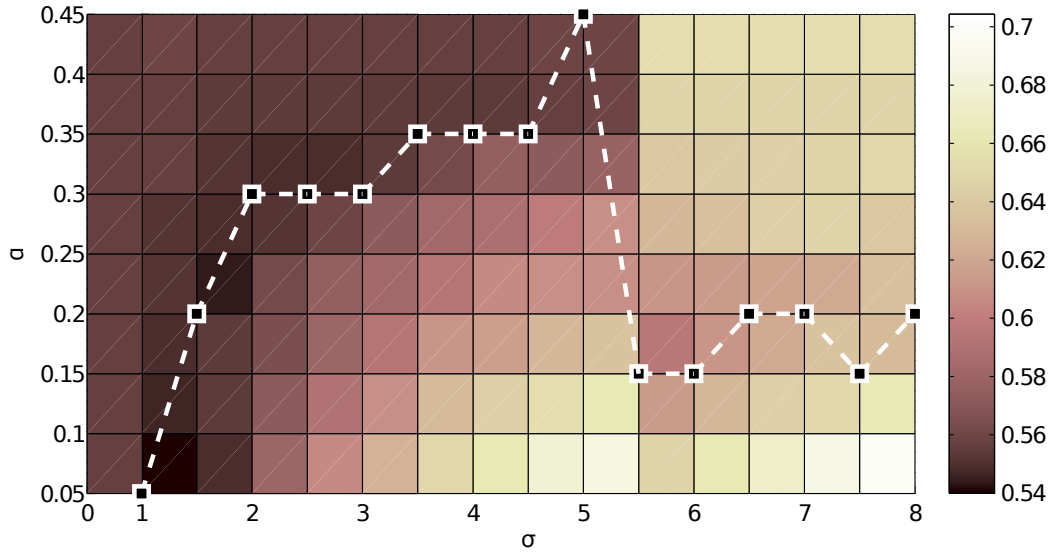
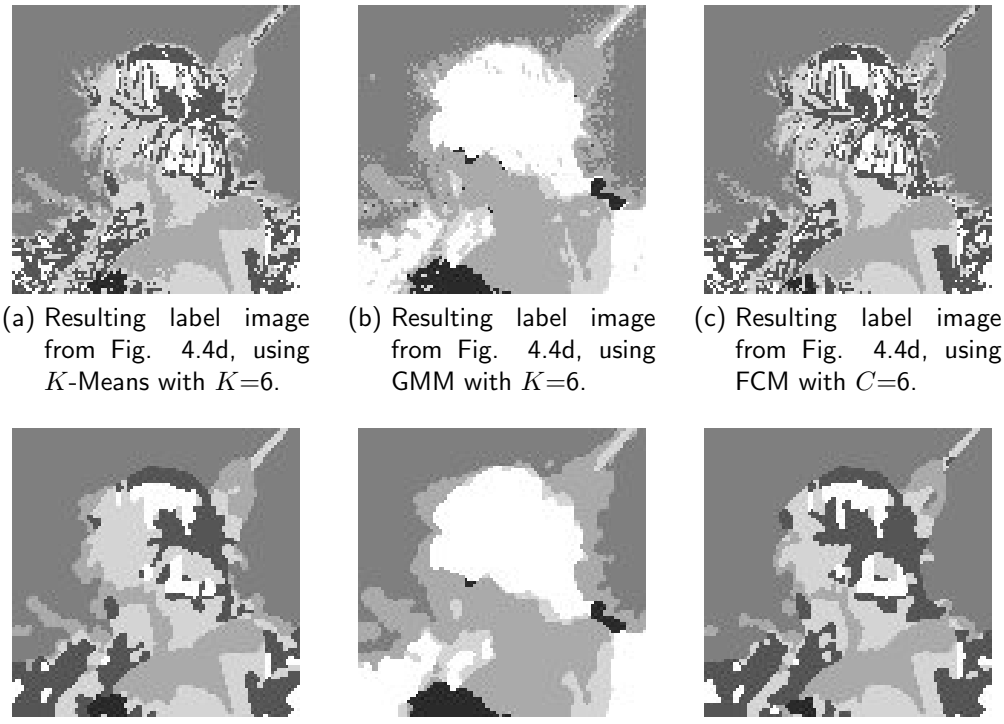


Figure 4.7: Influence of σ and α in segmentation refinement performance for BSDS300 database. Darker areas represent lower DTO values.

Table 4.1: Refinement applied to unsupervised segmentation of BSDS300 image database. Upwards arrows in right side of the table indicate improvement, whereas downwards arrows indicate worsening.

		K-Means	GMMs	FCM
$\overline{TPR}$	Base	0.4373	0.4368	0.4273
	Refined	**0.4628 (↑)**	**0.4425 (↑)**	**0.4590 (↑)**
σ_{TPR}	Base	0.0869	0.0790	0.0901
	Refined	**0.0793 (↑)**	**0.0826 (↓)**	**0.0810 (↑)**
$\overline{TNR}$	Base	0.9734	0.9350	0.9767
	Refined	**0.9519 (↓)**	**0.8923 (↓)**	**0.9559 (↓)**
σ_{TNR}	Base	0.0273	0.0497	0.0217
	Refined	**0.0389 (↓)**	**0.0744 (↓)**	**0.0382 (↓)**
$\overline{DTO}$	Base	0.5633	0.5669	0.5732
	Refined	**0.5393 (↑)**	**0.5678 (↓)**	**0.5428 (↑)**
$\overline{OS}$	Base	6400.3	3007.1	6998.8
	Refined	**968.35 (↑)**	**450.62 (↑)**	**1026.8 (↑)**
$\sigma_{\overline{OS}}$	Base	5010.7	1980.8	5116.6
	Refined	**792.33 (↑)**	**322.26 (↑)**	**789.21 (↑)**

behavior is very common in pixel color based segmentation since it usually produces spurious unconnected regions and gaps. Figure 4.8 illustrates how the refinement process cope with the aforementioned problem. A quick comparison between the figure pairs 4.8a and 4.8d, 4.8b and 4.8e and 4.8c and 4.8f demonstrates the smoothing effect of the refiner, and a proper elimination of small spurious features.



(a) Resulting label image from Fig. 4.4d, using K-Means with K=6.

(b) Resulting label image from Fig. 4.4d, using GMM with K=6.

(c) Resulting label image from Fig. 4.4d, using FCM with C=6.

(d) Figure 4.8a after refinement.

(e) Figure 4.8b after refinement.

(f) Figure 4.8c after refinement.

Figure 4.8: Example of K-Means, Gaussian mixtures and Fuzzy C-Means pixel color segmentation. Refined results were obtained with ($\alpha = 0.05$, $\sigma = 1.0$).

4.4 Mouth structures segmentation refinement

In this work, particular attention is given to mouth structure segmentation in images. Therefore, important notice on refiner's performance in mouth segmentation refinement is given in this Section. The experiment was conducted using sixteen images chosen from different subjects from the "Own" database, provided that each image has one corresponding manually annotated ground truth image. The label images used as the refiner input are the same that resulted from experiment in Section 3.2.4. Segmentation error, reflected by DTO, was computed using all of the sixteen images (including those five used for model training).
Following the same guideline proposed in natural images segmentation in Section 4.3.1, refiner parameters were tuned through a parameter sweep. From the experiment, the combination which led to better results in terms of average DTO was ($\sigma = 1.0,\ \alpha = 0.1$). Given this parameter selection, a good approximation for the inner term of Equation (4.1) can be achieved using a Gaussian window of five pixels width per five pixels height.

Table 4.2 contains the mouth structure segmentation average DTO measurements presented in Table 3.6, along with the corresponding average DTO obtained after the refinement. The Table is composed by measurements taken from all possible combinations of: 12-dimensional and 3-dimensional feature input vectors; whole image data and RoI clipped data; filtered and unfiltered; refined and unrefined segmentation labeling.

Table 4.2: Refinement results of K-Means based pixel classification.

			12 Features			3 Features		
			Lip	Teeth	Tongue	Lip	Teeth	Tongue
Whole	Unfilt.	Unref.	0.4170	0.4275	0.5873	0.4423	0.4174	0.6214
		Ref.	**0.3172 (↑)**	**0.3478 (↑)**	**0.6282 (↓)**	**0.4507 (↓)**	**0.4185 (↓)**	**0.4630 (↑)**
	Filt.	Unref.	0.2722	0.3243	0.4936	0.3167	0.2894	0.5074
		Ref.	**0.2443 (↑)**	**0.2958 (↑)**	**0.4992 (↓)**	**0.3154 (↑)**	**0.3326 (↓)**	**0.4436 (↑)**
Clipped	Unfilt.	Unref.	0.5055	0.4878	0.5220	0.5265	0.4580	0.5656
		Ref.	**0.3600 (↑)**	**0.3285 (↑)**	**0.6540 (↓)**	**0.4728 (↑)**	**0.3908 (↑)**	**0.5142 (↑)**
	Filt.	Unref.	0.3407	0.3632	0.4599	0.3718	0.3289	0.4852
		Ref.	**0.2978 (↑)**	**0.2561 (↑)**	**0.5109 (↓)**	**0.3519 (↑)**	**0.2978 (↑)**	**0.4706 (↑)**

Despite increases in some of DTO measurements occur, average DTO went from 0.3712 to 0.3549 for filtered 12-dimensional feature input vectors after refinement, reflecting a gain in classification accuracy of 1.56%. Similarly, this measurement decayed from 0.3953 to 0.3734 for filtered 3-dimensional feature input vectors after refinement, meaning a gain in classification accuracy of 3.62%. DTO for unfiltered inputs improved from 0.4482 to 0.4455, leading to an average gain in classification accuracy of 5.75%. The gain in average segmentation accuracy from unfiltered unrefined to filtered refined is 14.77% for 12-dimensional feature input vectors, and 14.5% for 3-dimensional feature input vectors. It can be thereupon deduced that image pre-processing and label refinement effects on segmentation accuracy are notably complementary when using K-Means pixel color classification.

Table 4.3 contains the mouth structure segmentation average DTO measurements presented in Table 3.7, along with the corresponding average DTO obtained after the refinement. Improvement in DTO measurements is more consistent than in the case of K-Means based classification, with average DTO going from 0.3537 to 0.3441 for filtered 12-dimensional feature input vectors after refinement, reflecting a gain in classification accuracy of 0.91%. This measurement went from 0.3593 to 0.3320 for filtered 3-dimensional feature input vectors after refinement, meaning a gain in classification accuracy of 2.59%. The gain in average segmentation accuracy from unfiltered unrefined to filtered refined is 17.56%. Particularly good results were obtained for 3-dimensional feature input vectors with pre-processing and refinement, closing to the accuracy of 12-dimensional feature input vectors case.

Figure 4.9 extend the example in Figure 3.9. The Figure illustrates the effect of segmentation refinement when segmenting using K-Means color classification and Gaussian based color classification. The joint effect of image pre-processing and refinement can be seen in Figures 4.9e and 4.9i.

Table 4.3: Refinement results of Gaussian mixture based pixel classification.

			12 Features			3 Features		
			Lip	Teeth	Tongue	Lip	Teeth	Tongue
Whole	Unfilt.	Unref.	0.4977	0.4195	0.5302	0.5153	0.4192	0.5698
		Ref.	**0.4230 (↑)**	**0.3324 (↑)**	**0.5116 (↑)**	**0.4241 (↑)**	**0.3829 (↑)**	**0.4497 (↑)**
	Filt.	Unref.	0.2911	0.2959	0.4673	0.3276	0.2849	0.4738
		Ref.	**0.2622 (↑)**	**0.2618 (↑)**	**0.4741 (↓)**	**0.2865 (↑)**	**0.3064 (↓)**	**0.4019 (↑)**
Clipped	Unfilt.	Unref.	0.4876	0.4728	0.5228	0.5106	0.4454	0.5643
		Ref.	**0.4450 (↑)**	**0.3244 (↑)**	**0.5444 (↓)**	**0.4484 (↑)**	**0.3546 (↑)**	**0.4983 (↑)**
	Filt.	Unref.	0.3103	0.3335	0.4244	0.3397	0.2851	0.4446
		Ref.	**0.3064 (↑)**	**0.2469 (↑)**	**0.4788 (↓)**	**0.3207 (↑)**	**0.2511 (↑)**	**0.4242 (↑)**

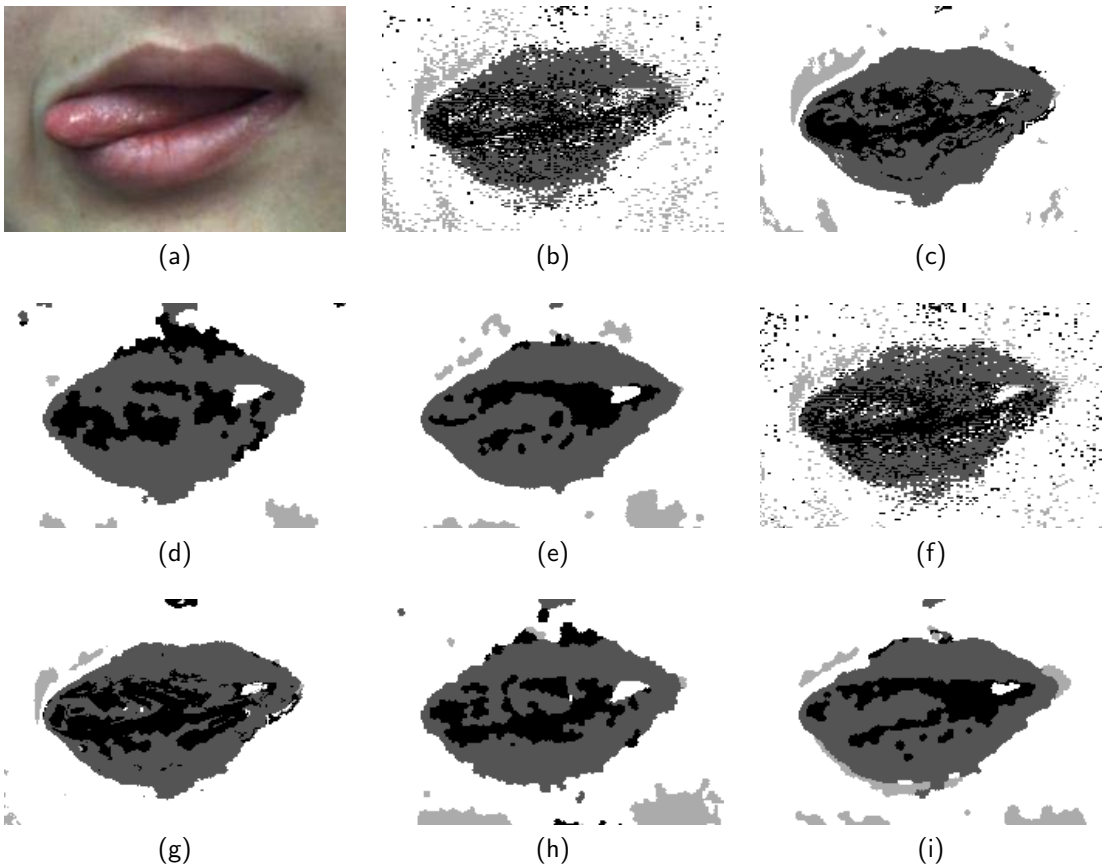


Figure 4.9: Segmentation refinement on K-Means based and Gaussian mixture based pixel color classification examples. 4.9a: Original image. 4.9b: K-Means based classification, $K = 3$. 4.9c: K-Means based classification, $K = 3$, filtered. 4.9d: Refined version of 4.9b. 4.9e: Refined version of 4.9c. 4.9f: Gaussian mixture based classification, $K = 3$. 4.9g: Gaussian mixture based classification, $K = 3$, filtered. 4.9h: Refined version of 4.9f. 4.9i: Refined version of 4.9g.

4.5 Summary

In this Chapter, a new technique for refining the label image resulting from segmentation is presented. The method, inspired in a simplistic model of the human visual perceptual system, improves iteratively the label image by smoothing region boundaries, filling small gaps inside or between regions, and eliminating small spurious regions.
The refiner is composed by a layer of perceptual units (one per pixel), each of them connected to one unique input label pattern, and to neighboring units' output. Two parameters, which are proven to be at some extent correlated in Sections 4.2 and 4.3.1, control the compromise between input labeling and field effect through iterations. The technique mimics the smoothing effect of low pass filters applied to labeling information, and its computational cost per iteration is also around the same as the one of such kind of filters. Refiner's behavior is analyzed in depth in Section 4.2, and numerical results are also provided in Sections 4.3 and 4.4.
In most cases, the refiner improves the output labeling resulting from unsupervised pixel color based segmentation of natural images. In the case of supervised mouth structures segmentation, the benefit is clearer by improving the results in all cases. The improvement is at some extent cumulative with the one obtained by the means of image pre-processing, thus proving to be complementary techniques. Individually, linear filtering and segmentation refinement increase segmentation accuracy by 5 to 10% approximately (reflected in DTO), while the combined effect of both techniques lead to an increment of 15% approximately. It is noteworthy that the computational complexity of each refinement iteration is comparable with that of the linear filter, and that the refiner usually takes between five and fifteen iterations to converge.

5 Texture in mouth structure segmentation

According to Shapiro & Stockman [128], local image texture can be defined in terms of local image statistic or structural behavior. From the first point of view, texture is regarded as basic image structures or neighborhoods in which some statistic or spectral measurements remain constant, or at least very similar, all along the textured region. The latter, closely related to human interpretation, sees texture as patterns which repeat themselves throughout some areas in the image preserving their appearance among occurrences. This approach to texture definition can be difficult to express in terms of measurable image features, despite of being easier to understand, thus the first definition is more commonly found in fields like image processing and artificial vision.

Texture features can be classified in two main categories: the first category, often identified as low-level texture features, encompasses the use of raw color or intensity data extracted from windows or pixel sets in the image. Then, texture is implicitly represented by the concatenation of that raw information provided by the pixels. The second category, referred as high level texture features, is based in local statistic measurements (i.e., moments), non-linear transformations or spectral information, extracted from the pixels and their corresponding neighborhoods.

Throughout this Chapter, texture descriptors are used in every stage of the proposed mouth segmentation scheme, discussing the benefits and disadvantages in every case. Particularly, Figure 5.1 shows how texture can interact within the stages involved in the proposed mouth structure segmentation scheme (as in Figure 2.5). These interactions are studied through this Chapter, discussing their usefulness in improving the initial color based pixel segmentation.

First, a brief introduction on low-level and high-level texture feature representation is presented in Sections 5.1 and 5.2, respectively. Then, in Section 5.3 the use of scale (a high level texture descriptor) in image preprocessing is discussed through an example, in the particular task of improving color definition and compactedness for pixel color classification of mouth structures. Section 5.4 shows a comparison between color and color+texture based pixel classification for mouth structure segmentation, using both low-level and high-level texture descriptors. Section 5.5 introduces the use of scale as a tool to automatize segmentation refinement, and studies the effect of this process in refined mouth structure segmentation based in pixel color/color+texture classification. Finally, a brief discussion on the topics explored in this Chapter is presented in Section 5.6.

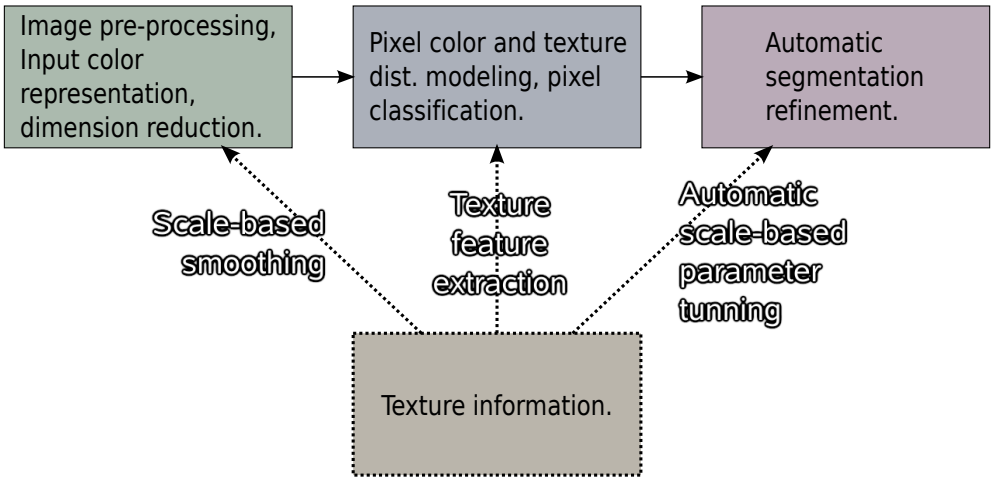


Figure 5.1: Mouth structure segmentation and refinement scheme, highlighting the alternative use of texture at each stage.

5.1 Low-level texture description

The simplest way to represent texture encompasses use of raw intensity and color data inside the supposed texture elements or units (usually referred as texels). Raw image data can be concatenated conforming feature vectors that can be used as texture indicators, like in the example in Figure 5.2. In this case, the pixel of interest is identified by the number five, and texture is encoded using the information contained on its immediate surroundings.

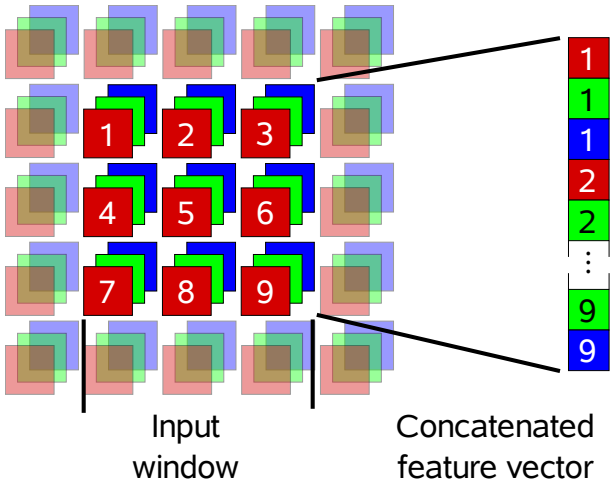


Figure 5.2: Low level feature vector conformation using RGB input and a 3x3 window.

In the example, the RGB information contained in a 3×3 pixels window is concatenated in a vector, which results in a 27-dimensional texture descriptor. It can be intuitively seen that pixel class modeling using a high-dimensional feature set suffers from the *Hughes effect*[1], meaning that great amounts of training data are needed in order to obtain a representative model for

[1] In other contexts, the *Hughes effect* or *Hughes phenomenon* is commonly referred as the *Dimensionality curse*.

each class. This limitation is usually overcome using dimension reduction techniques such as Principal Component Analysis (*PCA*) and Fisher Linear Discriminant Analysis (*FLDA*) [29, 30].

An example of mouth structure pixel classification in a bi-dimensional space conformed by the two principal components from an original low-level feature set is shown in Figure 5.3. The original input space encompasses color information contained inside pixel neighborhoods (windows of 3×3 pixels each) concatenated in a large feature vector like in Figure 5.2, using twelve color representations per pixel. Therefore, the original feature dimension is 108 components. After *PCA*, more than 99% of the input data variance was represented by the first two principal components.

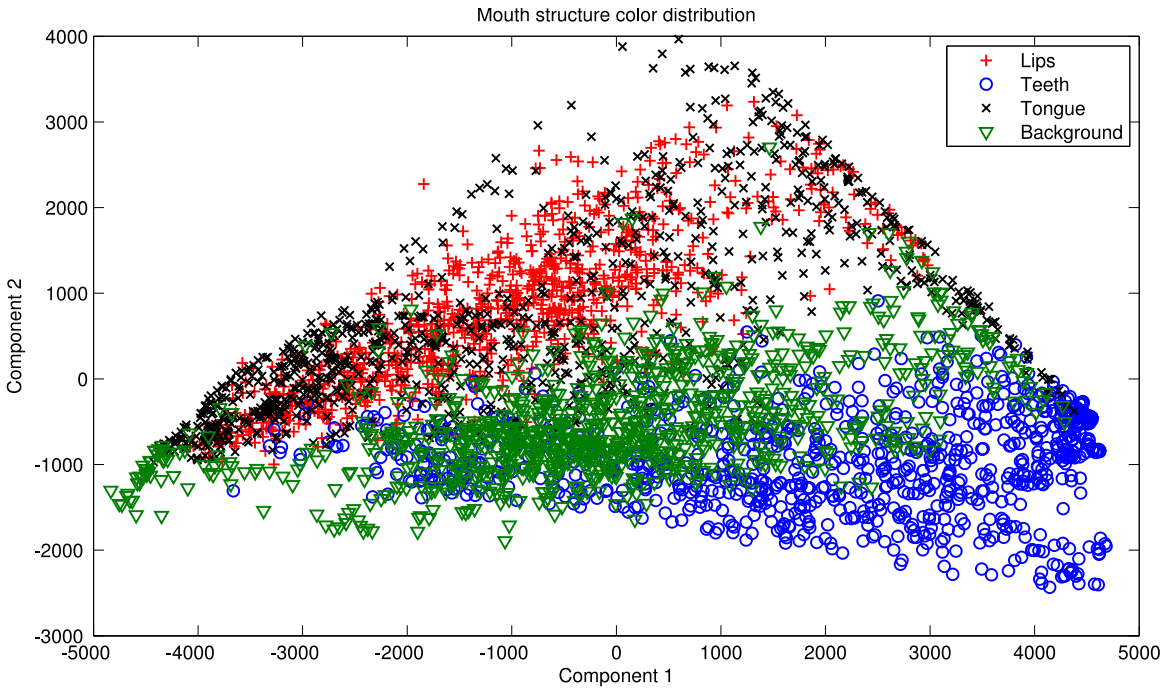


Figure 5.3: Mouth structure color distribution represented using first two principal components, which cover approximately 99.65% of data variance.

Notice that pre-defining the texture window size implies the assumption that texture can be properly described by the information contained inside that window. It also implies that the window has a size such that no relevant object features or objects are completely contained by it (therefore masked by the texture descriptors).

Low-level texture descriptors are commonly used in face detection, subject recognition, eyes and mouth detection, etc.. Techniques such as Eigenfaces [129], which is based in low-level texture description, proved to be effective in tackling automatic object recognition tasks. Nevertheless, their use pose some constraints like the availability of a large enough amount of annotated training data, the disallowance of considerable variations in pose, and a mandatory object scale standardization prior to classification.

5.2 High-level texture description

Low level texture description imposes a series of limitations, suffering notably from issues related to changes in scale, rotation, projection and shear. In cases in which texture preserves appearance features like local variability, orientation, etc., but varying in the aforementioned ones, a different approach is needed to texture characterization. Texture measurements based in more elaborated image analysis and complex transformations are grouped in a new category which is usually referred as high-level texture descriptors.

Under uncontrolled scale and rotation conditions, new texture samples can only be matched with base patterns or classes if it can be closely represented by them. Hence, it seems that a more adequate way to quantify texture should rely in the extraction of indexes that prove robustness against the aforementioned issues. One possibility is to define texture in terms of its local spectral behavior, codifying periodic or quasi-periodic elements using frequency bands and amplitudes.

Another possibility sees texture in terms of its local statistical behavior, considering mainly local first and second moments on the intensity and/or orientation. In those cases, texture patterns may be regarded as patches of information that retain their statistical behavior both locally and all throughout the textured region. Texture statistics are computed inside small windows, and continuity among patches is then measured in terms of continuity in their statistics. Clear examples of this kind of features are local anisotropy and contrast, both treated in Section 5.2.2.

Regardless the approach to texture concept, a texture patch is defined as a structural element inside the image, and is defined inside pre-defined pixel neighborhoods. A key issue in texture analysis is the estimation of the neighborhood size, as texture features may vary considerably depending on that size. In order to establish the size of the window of interest or neighborhood for each pixel in the image, a notion of local *integration scale* must be introduced. A common approach to estimate the local *integration scale*, based in local image statistics, is treated in the following Section.

5.2.1 Integration scale

The *integration scale*, *artificial scale* or simply scale (denoted as ${\sigma_S}^2$), is a measurement that reflects the variability in the local orientation inside an arbitrarily small window in the image. According to [130], one way to make scale notion concrete is to define it to be the width of the Gaussian window within which the gradient vectors of the image are pooled.

The integration scale is closely related to the statistical properties of the region surrounding the pixel, and its selection is not always straightforward. In [130] the authors present a technique for the implicit calculation of σ_S using an auxiliary quantity often referred to as polarity. Polarity is a measure of the extent to which the gradient vectors in a certain neighborhood all point in the same direction. Its value closes to one if the local orientation inside the neighborhood is uniform, and decreases when local orientation is sparse. Intuitively, once a proper scale value is achieved no further significant changes will be detected in local polarity.

[2]For any image location (x, y) the corresponding integration scale is denoted as $\sigma_S(x, y)$.

For any image location or pixel (x, y), polarity can be computed as

$$pol_\sigma(x,y) = \frac{|E_+ - E_-|}{E_+ + E_-} \tag{5.1}$$

with

$$E_+ = \sum_{x,y|\nabla I^T \hat{\mathbf{n}} > 0} \mathbf{G}_\sigma(x,y) \circ (\nabla I^T \hat{\mathbf{n}}) \tag{5.2}$$

$$E_- = - \sum_{x,y|\nabla I^T \hat{\mathbf{n}} \leq 0} \mathbf{G}_\sigma(x,y) \circ (\nabla I^T \hat{\mathbf{n}}) \tag{5.3}$$

where I denotes the image intensity value, $\hat{n}$ represents the main direction of ∇I at pixel (x, y)[3], $\mathbf{G}_\sigma(x, y)$ is a Gaussian window centered at point (x, y), and the $\circ$ operator denotes the Hadamard matrix product.

Once the polarity images are computed, $\sigma_S(x, y)$ is selected as the smallest value such that

$$\frac{\partial pol_{\sigma_S}(x,y)}{\partial \sigma_S(x,y)} \leq th \tag{5.4}$$

In [130], a threshold value of 2% ($th = 0.02$) is suggested. Since one cannot have the partial derivative directly, the authors swept the value of σ_S starting from 1.0 to 3.5 with steps of 0.5, stopping once the condition was met or the maximum value was reached. This allows them to limit the window size up to 10 pixels approximately[4].

The scale can be regarded as a texture descriptor by itself, but is more commonly used for computing features derived from the scale-based second moment matrix. Two well known features extracted from the second moment matrix are local anisotropy and contrast, both treated in the following Section.

5.2.2 Scale based features for image segmentation

Given the intensity (I) component of an image, the second moment matrix ($\mathbf{M}_{\sigma_S}(x, y)$) can be computed as in Equation (5.5).

$$\mathbf{M}_{\sigma_S}(x,y) = \mathbf{G}_{\sigma_S}(x,y) * (\nabla I)(\nabla I)^T \tag{5.5}$$

where $\mathbf{G}_{\sigma_S}(x, y)$ is a Gaussian kernel centered at (x, y) with a variance of ${\sigma_S}^2$. It is noteworthy that $*$ in Equation (5.5) does not represent an actual convolution since the operating window size and its corresponding weights depend on the value of $\sigma_S(x, y)$.

[3]The main direction $\hat{n}$ can be set as the unitary vector whose direction follows the main eigenvector of the second moment matrix inside the integration window generated by G_σ.

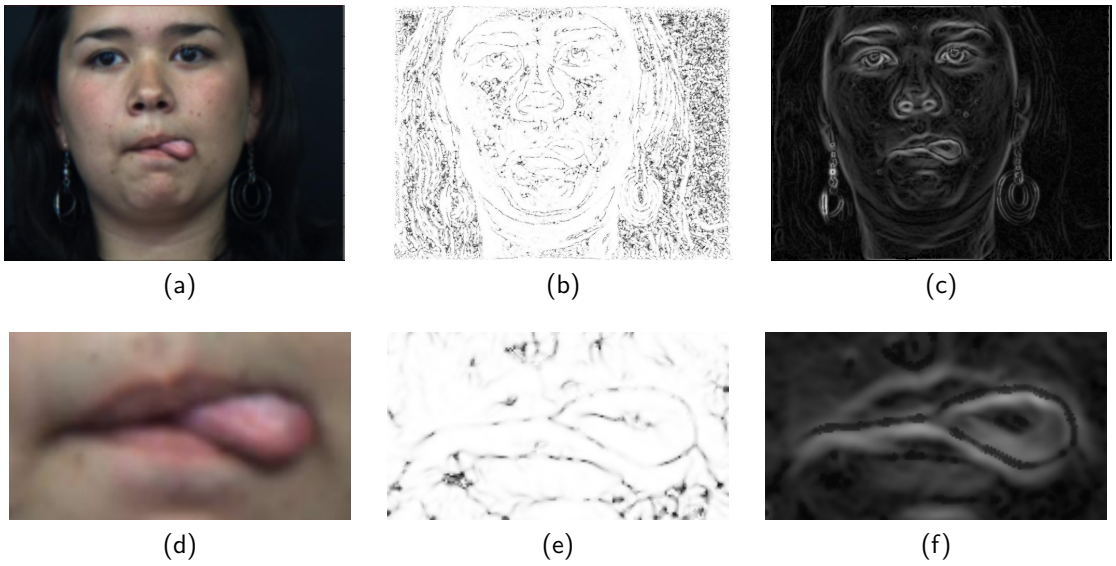
[4]Assuming that window width is approximately three times σ_S, covering around the 87% of the area enclosed by the Gaussian function. In this work, window sizes are computed as five times σ_S, thus covering almost a 99% of the area enclosed by the Gaussian function. This, however, increases computation time.

Local image contrast ($c(x,y)$) and local anisotropy ($a(x,y)$) can be obtained from the eigenvalues of $M_\sigma(x,y)$–denoted by $\lambda_1(x,y)$ and $\lambda_2(x,y)$, with $\lambda_1(x,y) \geq \lambda_2(x,y)$)–as in Equations (5.6) and (5.7).

$$c(x,y) = 2(\sqrt{\lambda_1(x,y) + \lambda_2(x,y)})^3 \tag{5.6}$$

$$a(x,y) = 1 - \frac{\lambda_2(x,y)}{\lambda_1(x,y)} \tag{5.7}$$

Figure 5.4 shows local anisotropy and contrast for test image in 5.4a. It can be easily noted that anisotropy take high values in relatively smooth regions, while contrast is higher in areas where local variation rises.



(a) (b) (c)

(d) (e) (f)

Figure 5.4: Local anisotropy and contrast. a) Original image; b) gray-coded local anisotropy (white: 1.0, black: 0.0); c) gray-coded local contrast (white: 1.0, black: 0.0); d), e), f) detail from a), b) and c), respectively.

5.3 Scale based image filtering for mouth structure classification

Since the *integration scale* reflects the extent at which orientation varies inside a window, its value can be used to setup anisotropic image pre-processing. If local scale can be estimated properly, a Gaussian filter whose deviation correspond to the pixel scale will theoretically smooth noise and localized orientation variation while preserving color, thus improving region color compactness.

In order to measure the extent of such pre-processing, a test base encompassing sixteen manually annotated images was selected. Each image was processed using a scale-variant Gaussian filter and a fixed scale Gaussian filter with $\sigma = 1.8$, corresponding to a window size of 9×9 pixels. For training and testing purposes, ten thousand pixels were randomly chosen

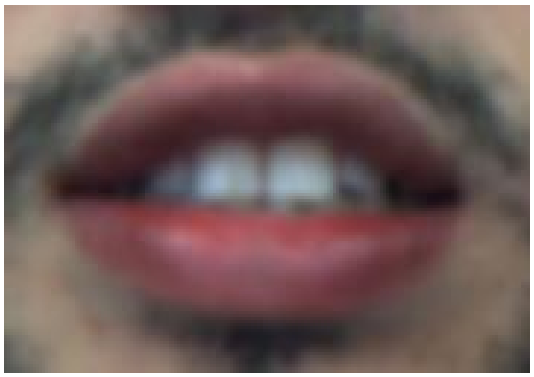
from each structure in the whole image set (lips, teeth, tongue and background), totaling forty thousand pixels. The pixels were equally distributed for training and testing.
Table 5.1 presents the pixel color classification performance before and after a scale-variant Gaussian filtering. Notice that the filter helps improving TPR, TNR and DTO for lips and tongue, while producing an adverse effect in the teeth region. The average improvement obtained for each region during training was 1.85% in DTO for all structures, with a standard deviation of 6.64%; using the testing database, the average improvement was 1.66% in DTO for all structures, with a standard deviation of 7.03%. The measured averaged changes in DTO are several times smaller than their corresponding deviations, which is inconclusive regarding the effect of the scale-based filtering process in mouth structure color classification. It is noteworthy that computing the scale implies complex calculations, including several sequential steps for each pixel in the image. Furthermore, fixed-scale filtering generated a clearer improvement in classification accuracy, as shown in Sections 3.1.2 and 3.2.4.

Table 5.1: Color classification performance using scale-based filtering.

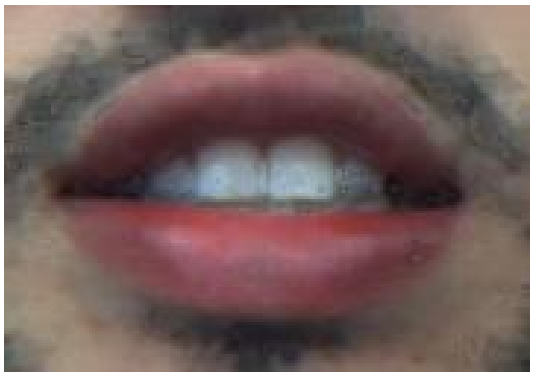
			Training			Testing		
			TPR	TNR	DTO	TPR	TNR	DTO
Lips	RGB	Unfilt.	0.3774	0.9448	0.625	0.383	0.9426	0.6197
		Filt.	**0.4274** (↑)	**0.9457** (↑)	**0.5752** (↑)	**0.4166** (↑)	**0.9456** (↑)	**0.5859** (↑)
	12-C	Unfilt.	0.648	0.9463	0.3561	0.6242	0.9439	0.3799
		Filt.	**0.6748** (↑)	**0.9501** (↑)	**0.329** (↑)	**0.6592** (↑)	**0.9503** (↑)	**0.3444** (↑)
Teeth	RGB	Unfilt.	0.7686	0.9767	0.2326	0.7698	0.9764	0.2314
		Filt.	**0.6798** (↓)	**0.9709** (↓)	**0.3215** (↓)	**0.6786** (↓)	**0.9694** (↓)	**0.3228** (↓)
	12-C	Unfilt.	0.8292	0.9742	0.1727	0.8294	0.9741	0.1726
		Filt.	**0.8026** (↓)	**0.9673** (↓)	**0.2001** (↓)	**0.787** (↓)	**0.9669** (↓)	**0.2156** (↓)
Tongue	RGB	Unfilt.	0.7454	0.9235	0.2658	0.7388	0.9266	0.2713
		Filt.	**0.8428** (↑)	**0.9391** (↑)	**0.1686** (↑)	**0.8304** (↑)	**0.9408** (↑)	**0.1796** (↑)
	12-C	Unfilt.	0.8558	0.9432	0.155	0.8368	0.9382	0.1745
		Filt.	**0.9086** (↑)	**0.9546** (↑)	**0.1021** (↑)	**0.9112** (↑)	**0.9507** (↑)	**0.1016** (↑)

Figure 5.5 presents a sample image filtered using fixed-scale and scale-variant Gaussian filters. Typical uniform smoothing obtained from fixed-scale filters is evidenced in Figure 5.5b. Notice that fixed-scale filtering affects negatively region features that are below its integration scale– in this case corresponding to approximately 1.8 pixels–by blurring them excessively, as in the case of region borders. Nevertheless, the filter stabilizes color inside regions, making them more compact and less prone to generate unconnected spurious regions after pixel color classification. This advantage is lost at some extent when using scale-variant filtering, as some localized lighting effects persist after the smoothing effect of the filter. By dissecting Figures 5.5c and 5.5d one can identify where the filter has effectively softened region color (i.e., near lip corners) and where it has preserved localized features that may lead to pixel misclassification (i.e., region borders with high color variability and specular light reflections like those found in the teeth).
Despite exposing a better behavior in preserving region borders, scale-variant filtering does not remove specular noise or highly variable texture hatching. It can also be noticed that the lack of continuity in scale among neighboring pixels introduces artifacts, easily identifiable at teeth and around some region borders.

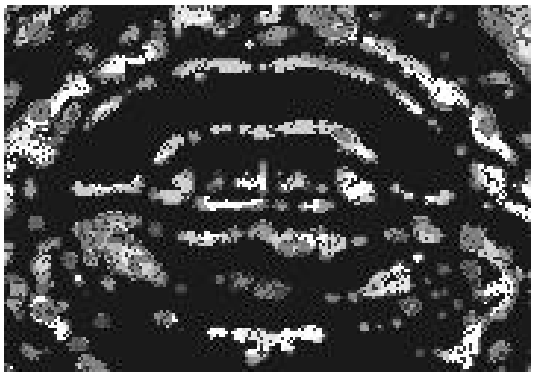
(a) Original image.



(b) Filtered image obtained using a 9×9 Gaussian filter.



(c) Filtered image obtained using a scale-variant Gaussian filter.



(d) Image scale, codified using grayscale (White: 10 pixels, Black: 1 pixel).

Figure 5.5: Example of scale-variant Gaussian filtering vs. fixed-scale Gaussian filtering.

5.4 Texture features in mouth structure classification

Texture descriptors can be used along with pixel color features to model the pixel distribution for each mouth structure. In order to expose the effect of using texture in that task, a mouth classification experiment was set up following the basic guideline stated in Section 3.2.4. There, 16 facial images from 16 subjects were chosen from the "Own" database, and a subset of pixels was extracted from each structure pixel data, totaling six thousand data points per structure for training the models and the whole image data for testing.
Table 5.2 shows the accuracy of pixel classification using color and texture features in a Gaussian mixture classification engine, using four different setups. The first combination, identified as LLTF-27, encompasses the use of low level texture features, particularly the first two principal components extracted from a concatenated version of the color information contained in windows with 3×3 pixels in size, like in Figure 5.2. The suffix "27" is used to denote the number of Gaussians in the distribution model corresponding to each structure. This number is chosen to match as close as possible the total number of parameters of the reference classification configuration chosen in Section 3.2.4: three dimensional input vector resulting from color-based LDA space reduction, twenty Gaussians per model. The second combination, referred as HLTF-27, uses local anisotropy and contrast in order to conform the input vectors, and like in the previous case, 27 Gaussians are chosen to model each mouth structure's pixel distribution. The third combination, called CATF-5, makes use of the color and texture information in order to conform the input vectors (seven features in total), and uses five Gaussians to model each pixel distribution. Finally, combination CATF-20 follows the same scheme as in CATF-5, only changing the total number of Gaussians used to model pixel distribution. While CATF-5 configuration was selected as a close match in number of parameters to the reference configuration, CATF-20 surpasses greatly that number.

Table 5.2: Pixel classification accuracy measured using DTO.

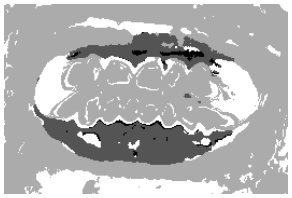
		Lips	Teeth	Tongue	Backg.	Mean
Whole	Base	0.6767	0.6763	0.3175	0.6893	0.4950
	LLTF-27	0.8697	0.4131	0.8033	0.5538	0.6600
	HLTF-27	0.8581	0.5323	0.6902	0.3737	0.6136
	CATF-5	0.8993	0.2894	0.6006	**0.2431**	0.5081
	CATF-20	0.8662	0.3635	0.5908	**0.2537**	0.5185
Clipped	Base	0.6893	0.3389	0.6468	0.3459	0.5052
	LLTF-27	0.8722	0.4294	0.7179	0.5530	0.6431
	HLTF-27	0.8760	0.5712	0.6952	0.5519	0.6736
	CATF-5	0.9001	0.3177	0.6051	**0.3092**	0.5330
	CATF20	0.8682	0.3796	0.6018	**0.3302**	0.5450

From the Table, it is noteworthy that background-associated DTO is lower (therefore better) for mixtures of color and texture features in the input vector, thus concentrating most of classification error among mouth structures. Hence, it is safe to advise the use of color and texture for tasks in which the main goal is to separate the mouth as a whole from the background. In the case of inner mouth structures separation, the best results were obtained

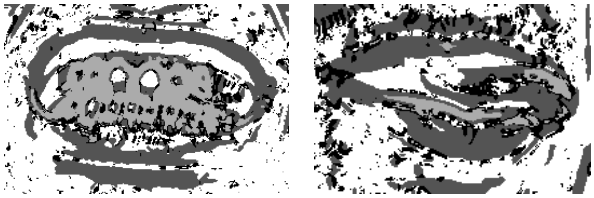
using feature vectors derived utterly from color.

Another important effect that can be observed from the Table lies in the fact that classification accuracy in structure from "the rest" classification dropped when increasing the number of modeling Gaussians from five to twenty (CATF-5 vs. CATF-20). This result indicates that, in the latter, the number of parameters introduces model overfitting. Specifically, this effect can be evidenced clearly in a tongue-related DTO increase. Figure 5.6 shows eight image labelings resulting from two RoI clipped input images, sweeping over the combinations studied in Table 5.2. Figure 5.6 summarizes the effect obtained by either using texture-only features in pixel classification, or using color and texture feature combinations. Notice that mouth from background distinction is both perceptually and numerically superior when both color and texture are used together.

(a) RoI clipped original images.

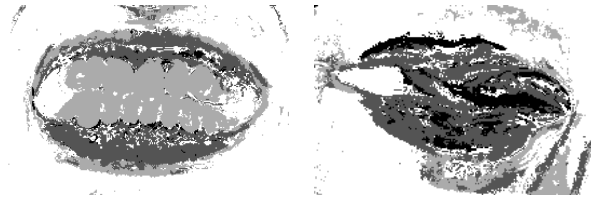


(b) Low-level texture feature based pixel classification.



(c) High-level texture feature based pixel classification.

(d) Color and texture feature based pixel classification, using five Gaussians per model.



(e) Color and texture feature based pixel classification, using twenty Gaussians per model.

Figure 5.6: Mouth structure pixel classification using texture and color features.

5.5 Automatic scale-based refiner parameter estimation

In the previous Chapter a method for segmentation refinement through iterative label updates was introduced. This method relies in the usage of two parameters that control the integration window size and the probability for a label to change due to the neighboring labels. The first parameter, which was denoted by σ, can be directly related to the *integration scale* studied in the previous Section, as they both intend to reflect the size of the smallest window inside whom

the local orientation is pooled[5]. Hence, this Section shows the behavior of the segmentation refinement process if σ is set after the value of the local *integration scale*. The experiment was conducted using the same basis as in Section 5.4, renaming it from "Base" to GMM-20, and applying both constant scale and variant scale segmentation refinement to the whole set .

Table 5.3 shows the result of comparing three different refinement combinations. The identifier GMM-20 represents mouth structure pixel classification results using three color features as input in color models obtained using twenty Gaussians per structure, following the scheme used in Chapter 3. GMMR-20, GMMCR-20 and GMMVR-20 points the accuracy obtained using fixed-scale fixed-proportion refinement, variable-scale fixed-proportion refinement and variable-scale variable-proportion refinement, respectively. It is noteworthy that despite that the first combination, denoted by GMMR-20, presents the best mean accuracy in structure segmentation, the variable-scale related combinations improve mouth from background distinction for data sampled inside the whole image and the RoI.

Table 5.3: Variations on label refinement of color-based mouth structure segmentation, measured using DTO.

		Lips	Teeth	Tongue	Backg.	Mean
Whole	GMM-20	0.6767	0.3095	0.6763	0.3175	0.4950
	GMMR-20	0.6639	0.2858	0.6626	0.3081	**0.4801**
	GMMCR-20	0.6774	0.2794	0.6998	**0.2840**	0.4851
	GMMVR-20	0.6726	0.2991	0.7075	**0.2889**	0.4920
Clipped	GMM-20	0.6893	0.3389	0.6468	0.3459	0.5052
	GMMR-20	0.6774	0.3200	0.6299	0.3387	**0.4915**
	GMMCR-20	0.6872	0.3310	0.6764	**0.3091**	0.5009
	GMMVR-20	0.6827	0.3431	0.6858	**0.3160**	0.5069

Table 5.4 presents a comparison between color and color + texture based based mouth structure classification.

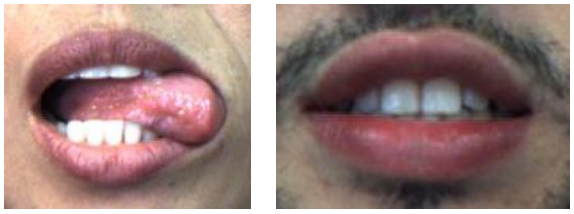
In both Tables, the compromise in accuracy favors mouth from background distinction over structure from structure distinction.

5.6 Summary

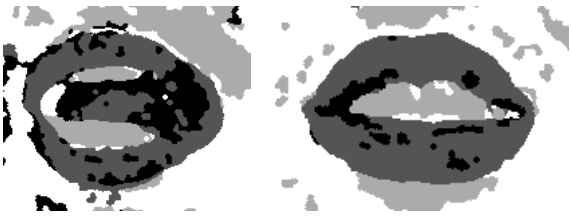
In this Chapter, the usage of texture features in mouth structure segmentation is evaluated. Since texture indexes can be used at each stage in the process, the evaluation extends to image pre-processing, pixel classification and segmentation refinement.

As shown in Chapter 3, image pre-processing proved to benefit pixel color classification, notably through the use of fixed-scale low pass linear filters. Particularly, the use of a 9×9 Gaussian filter improved pixel classification DTO for all mouth structures. In this Chapter, the Gaussian filter's size was made variable in terms of local scale, using the measured *integration scale* for

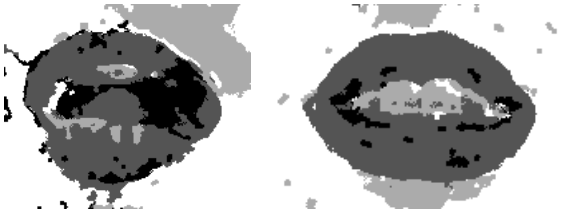
[5] This is particularly true if color and intensity variations can be associated with label changes in the segmented image.



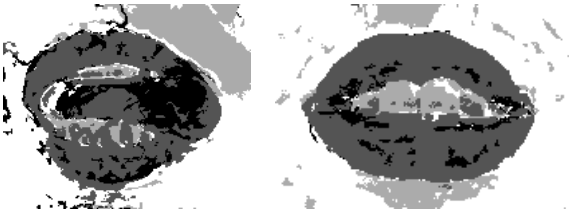
(a) RoI clipped original images.



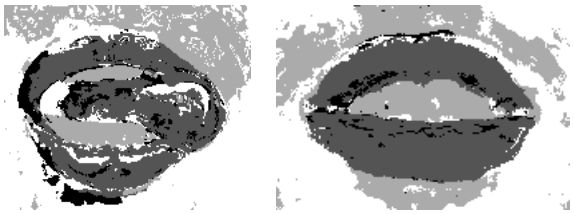
(b) Mouth structure classification using color features, refined with fixed parameters.



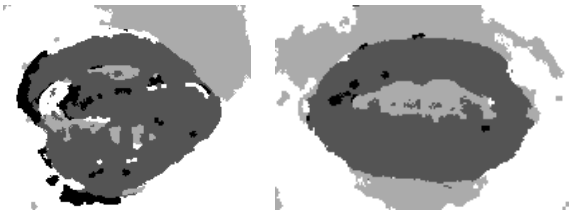
(c) Mouth structure classification using color features, refined with variable integration scale.



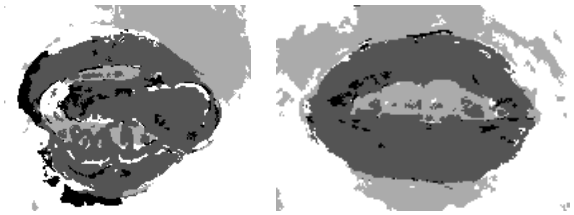
(d) Mouth structure classification using color features, refined with variable proportion and integration scale.



(e) Mouth structure classification using color and texture features, refined with fixed parameters.



(f) Mouth structure classification using color and texture features, refined with variable integration scale.



(g) Mouth structure classification using color and texture features, refined with variable proportion and integration scale.

Figure 5.7: Refinement approaches in color and color+texture based mouth structure classification.

Table 5.4: Effect of scale-based label refinement in structure segmentation accuracy using color and color+texture models.

		Lips	Teeth	Tongue	Backg.	Mean
Whole	GMM-20	0.6767	0.3095	0.6763	0.3175	0.4950
	GMMCR-20	0.6774	0.2794	0.6998	0.2840	**0.4851**
	GMMVR-20	0.6726	0.2991	0.7075	0.2889	0.4920
	CATF-5	0.8993	0.2894	0.6006	0.2431	0.5081
	CATFCR-5	0.9584	0.2373	0.6551	**0.2231**	0.5185
	CATFVR-5	0.9427	0.2620	0.6486	0.2305	0.5209
Clipped	GMM-20	0.6893	0.3389	0.6468	0.3459	0.5052
	GMMCR-20	0.6872	0.3310	0.6764	0.3091	**0.5009**
	GMMVR-20	0.6827	0.3431	0.6858	0.3160	0.5069
	CATF-5	0.9001	0.3177	0.6051	0.3092	0.5330
	CATFCR-5	0.9587	0.2976	0.6545	**0.2775**	0.5471
	CATFVR-5	0.9430	0.3115	0.6491	0.2872	0.5477

every pixel. Results of image filtering with the scale variable filter expose a clear retention of structure borders while smoothing the color information within each region. Nevertheless, features such as specular noises and strongly variable textures (like the bright hatched pattern in the lips) also remain after filtering. Hence, pixel classification performance was not clearly improved by the scale-variant filtering, as opposed to the fixed scale version. This fact makes it advisable to use a fixed-scale filter over a scale variant version.

In the next stage of the segmentation process, texture descriptors are used as part of the feature vector fed to the pixel classification engine. Texture is characterized using a reduced set of low-level features, and two more features derived from the *integration scale*, known as local contrast and anisotropy. The augmented feature set show a considerable improvement in mouth from background distinction, but the addition of the texture features raised the confusion between lips and tongue regions. The results of the conducted tests indicate that a good practice can be derived from the mixed use of texture and color features for initial mouth selection, and then the use of color-only features for structure from structure classification.

Finally, the *integration scale* was used to set up automatically the scale parameter σ for segmentation refinement. As in the case of the scale-variant filtering, the lack of continuity in the scale among neighboring pixels led to refinement results exhibiting poorer results than those obtained with the presets found in the previous Chapter.

At the end, texture proved to be particularly helpful in pixel color classification for mouth from background distinction, but its usage is bound to the quality/performance compromise for every particular application. However, its use in pre-processing and segmentation refinement in the task mouth structure classification can be safely avoided.

6 An active contour based alternative for RoI clipping

In recent years, lip contour extraction has regained attention from research community, mainly due to its great potential in human machine interface and communication systems development. Most of the lip contour extraction techniques have been designed for audio-visual speech recognition (AVSR) [2, 131], and in some cases extended to speaker identification. In this task only a few landmarks suffice to estimate key features, as in the case of MPEG-4 facial animation parameters [117].

Over-specialization of lip contour extraction methods for the task of AVSR makes them inappropriate in cases where mouth appearance may change subjected to pose changes and/or specific lip malformations. This also excludes them when an accurate description of the whole mouth contour is needed, rolling them out for general gesture detection or accurate mouth structures segmentation.

Indeed, mouth gestures cannot be solely described by the outer contour of the mouth. There is always a set of gestures that can match the same mouth contour, even though they may differ in the type of structures visible in the images, as well as their aspect, size and position. Thereby, features extracted from the different structures should be taken into account for precise gesture characterization.

The considerable overlap between lip and skin color distributions slanted mouth segmentation towards lip region segmentation. Elaborated methods, like those presented in [35, 105], seem to cope with some limitations of color-based techniques. Nevertheless, their associated computational complexity make them unsuitable for real time applications.

Even when the refinement algorithm proposed in Chapter 4 is able to correct most of the problems introduced by pixel-based color segmentation, some spurious regions may remain in the refinement output. In this section, the use of an outer lip contour extraction method aimed to constrain the Region of Interest (RoI) is proposed. The technique is based in the work of Eveno *et al.* [2], with the modification introduced in [132]. This technique is aimed to be a in-place alternative for the stages discussed in Subsections 7.3.4 and 7.3.5 in the segmentation streamline presented in Section 7.3.

6.1 Upper lip contour approximation

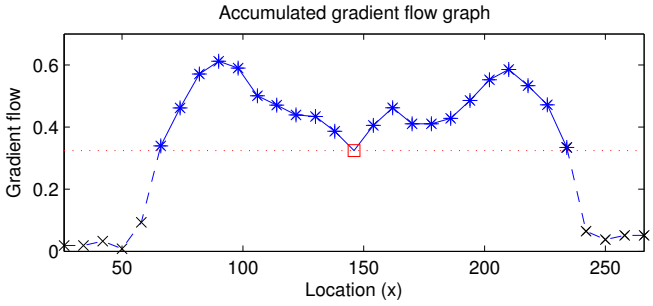
The core of the upper lip contour approximation in Eveno's technique is the gradient flow value of the Pseudo-hue color component minus the Luminance. Denoted as φ, its value aids in selecting those points that should be added to or trimmed from the snake. The gradient

flow passing through the line segment $|\mathbf{p}_{i-1}\mathbf{p}_i|$, denoted φ_i, is given by

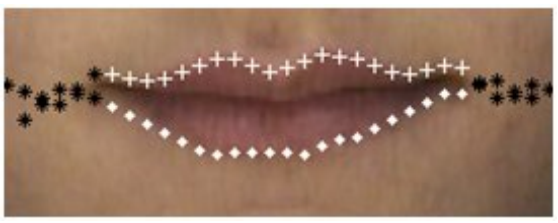
$$\varphi_i = \begin{cases} \int_{\mathbf{p}_i\mathbf{p}_{i+1}} \dfrac{[\nabla(ph - L))] \cdot \mathbf{dn_l}}{|\mathbf{p}_i\mathbf{p}_{i+1}|}, & i \in [1, N] \\ \int_{\mathbf{p}_{i-1}\mathbf{p}_i} \dfrac{[\nabla(ph - L))] \cdot \mathbf{dn_r}}{|\mathbf{p}_{i-1}\mathbf{p}_i|}, & i \in [N+2, 2N+1] \end{cases}, \tag{6.1}$$

where $2N + 1$ is the total number of points in the snake, $N + 1$ stands for the seed point index in the snake points set; ph represent the Pseudo-hue component values of the pixels in the line segment; L the corresponding Luminance value; and $\mathbf{dn}_l$ and $\mathbf{dn}_r$ are normalized vectors which lie perpendicular to line segments conformed by the points located at left or right side of the seed and the seed itself. According to [2], a seed point is chosen slightly over the upper lip, and then points are iteratively added at left and right sides of the seed. This process occurs subject to a constrained flow maximization rule, while preserving a constant horizontal distance in pixels (denoted as Δ).

In this Chapter, an alternative approach for seed updating, point addition and trimming are used. This modified methodology is described in depth in [132], and proposes a continuous point addition until a noticeable decrease in flow is obtained. Gradient flow usually increases in value when approaching mouth corners, and then decreases rapidly when points are added outside mouth contour. Points should be added until their associated φ decay below the minimum value of φ closest to the seed (this can be seen in Figure 6.1). Points are added preserving the line segment size, instead of preserving a constant horizontal distance, thus producing better results in near vertical contour approximation.



(a) Gradient flow plot.



(b) Detected mouth contour, showing trimmed points in black.

Figure 6.1: Gradient flow based point trimming.

Seed update is performed by maximizing the gradient flow passing through both seeds associated left and right line segments, ensuring that overall line segment size is kept bounded. The

goal can be translated into maximizing the objective function

$$\varphi_{N+1} = \int_{\mathbf{p}_N\mathbf{p}_{N+1}} \frac{[\nabla(ph - L))] \cdot \mathbf{dn}_-}{|\mathbf{p}_N\mathbf{p}_{N+1}|} + \int_{\mathbf{p}_{N+1}\mathbf{p}_{N+2}} \frac{[\nabla(ph - L))] \cdot \mathbf{dn}_+}{|\mathbf{p}_{N+1}\mathbf{p}_{N+2}|} \tag{6.2}$$

subject to

$$\begin{array}{l} |\mathbf{p}_N\mathbf{p}_{N+1}| < \gamma, \text{ and} \\ |\mathbf{p}_{N+1}\mathbf{p}_{N+2}| < \gamma \end{array}, \tag{6.3}$$

where $\mathbf{p}_{N+1}$ is the seed, φ_{N+1} is the gradient flow associated to point $\mathbf{p}_{N+1}$, $\mathbf{dn}_-$ and $\mathbf{dn}_+$ are the normalized gradient vectors of left and right line segments, and γ is a parameter that controls the average line segment length. The value of γ should lie inside the range $(\Delta, 2\Delta - 5]$. Smaller γ values will lead to smoother lip contour, but it increases the fitting error around corners in cupids arc. This method also ensures that seed final position lies closer to the actual lip contour, contrasting with [2]. Therefore, the seed position can be treated like any other snake point in further procedures.

6.2 Lower lip contour approximation

Lower lip contour approximation is performed following the same principle as for the upper lip contour, but in this case flow is computed using only the gradient of Pseudo-hue component. Therefore, the formulation in Equation (6.1) changes into

$$\varphi_i = \begin{cases} \int_{\mathbf{p}_i\mathbf{p}_{i+1}} \frac{(\nabla ph)] \cdot \mathbf{dn_l}}{|\mathbf{p}_i\mathbf{p}_{i+1}|}, & i \in [1, N] \\ \int_{\mathbf{p}_{i-1}\mathbf{p}_i} \frac{(\nabla ph)] \cdot \mathbf{dn_r}}{|\mathbf{p}_{i-1}\mathbf{p}_i|}, & i \in [N+2, 2N+1] \end{cases}. \tag{6.4}$$

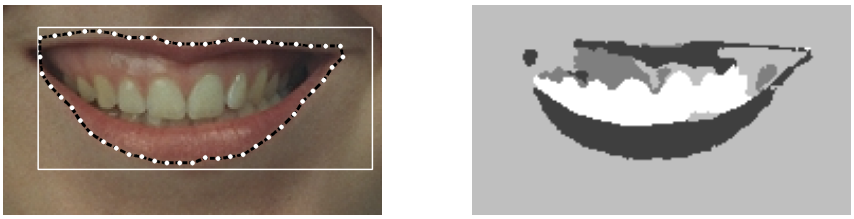
Usually, only one or two iterations suffice in order to achieve full convergence. Similarly, the seed update follows the formulation in (6.2), substituting $\nabla(ph - L)$ with ∇ph.

6.3 Automatic parameter selection

The initial value of Δ, as well as the location of upper and lower seeds, can be obtained by using the bounding box of the mouth as an initial RoI. Upper and lower seed initial position is computed by choosing the closest points labeled with "Lip" to the mid-points of upper and lower RoI boundaries. Then, the average of each pair of mid-point and closest "Lip" point is used as a seed.

An acceptable value for Δ can be chosen by dividing the RoI width by $4N$, whenever that operation leads to a value bigger than five pixels. Otherwise, decreasing the value of N is recommended. When Δ is chosen to be smaller than five pixels the flow through each line segment is highly unstable thus introducing undesired local minima in (6.1) and (6.2).

Once the outer lip contour extraction algorithm has converged, contour points can be used in order to conform a restricted RoI for mouth structures segmentation. The effect of this process can be seen in Figure 6.2.



(a) (b)

Figure 6.2: Lip contour based RoI clipping. (a) Lip contour approximation; (b) mouth structure segmentation after RoI clipping.

The computational complexity associated to the contour approximation technique has an order $\mathcal{O}(n)$, regarding the total number of contour points. This is usually much less than the complexity associated in computing the Pseudo-hue and Luminance transforms–both of them needed for contour approximation, whose order is $\mathcal{O}(n)$ regarding the total number of pixels in the image. This makes the technique much faster than the streamline conformed by the texture-based clipping with a later convex hull based region trimming, both discussed previously in this Chapter.

6.4 Tests and results

Images were selected from three different databases: the FERET database [115, 116], widely used in subject identification and gesture detection researches, and briefly treated in Chapter 2. A second database, conformed by facial images extracted from video sequences of children who have been operated for cleft lip and/or palate (from now on denoted as CLP); and a database comprised by facial images extracted from video sequences taken from different subjects during speech with uncontrolled lighting conditions (from now on denoted as "Other"). The FERET images were clipped in order to contain information primarily from the lower face. The image set used in the experiments contains 24 images from FERET database, 5 from CLP database, and 16 from "Other" database. Every image in the dataset was manually annotated in order to be used as ground truth.

In the first experiment, Gaussian mixtures were trained in a supervised manner in order to model the color distribution of each class. The image pixels were represented using feature vectors containing RGB, L*a*b* and Pseudo-hue color values. Four Gaussians were used in each mixture. Results for pixel classification performance can be seen in the confusion matrices in Table 6.1. The best results were achieved using the "Other" database images. This can be explained by the fact that the intra-class variation is lower than in FERET or CLP, as is the overlapping between classes. An illustrative example using one image from CLP and one image from "Other" is provided in Figure 6.3.

From the Table, it is noteworthy that overall classification performance improves greatly for

Table 6.1: Pixel classification performance for three databases, measured using DTO.

		Lips	Teeth	Tongue	Backg.
FERET	Class.	0.2154	0.2309	0.6923	0.4258
	Ref.	**0.1895**	0.1944	0.6723	0.3683
	C. RoI	0.1992	**0.0580**	**0.6711**	**0.2198**
"Other"	Class.	0.3368	0.1372	0.1418	0.1132
	Ref.	**0.2921**	**0.1040**	**0.0904**	**0.0506**
	C. RoI	0.3602	0.1457	0.1448	0.0777
CLP	Class.	0.1885	0.3016	N.A.	0.4719
	Ref.	0.1629	0.3108	N.A.	0.4684
	C. RoI	**0.0283**	**0.1205**	N.A.	**0.0242**

FERET and CLP images when contour based RoI clipping is used; however, the performance actually dropped for "Other" images. This effect is due to control points intruding inside the lips, thus over-cutting regions that should have been preserved. The impossibility to detect when a control point gets inside the mouth area makes the technique prone to rejection for fully automated systems, and therefore is not recommended nor included in the main streamline of the proposed gesture detection methodology. The technique is nevertheless advisable for semi-automatic systems where a human operator gives the seed points location manually. Other limitations of this technique are treated in the following notes.

Gradient flow related issues

The basis of the contour approximation algorithm is the local gradient flow, as computed in Equation 6.1. The possibility of a contour approximation method converging to a true contour depends in the quality of the gradient, and in turn the gradient depends on image quality.
There are several factors that impact gradient quality, most notably image noise. This challenge is tackled by the means of image pre-processing algorithms like filters and morphological operations. Unfortunately, coping with noise usually imply a negative affectation in border region definition, causing an excessive contour smoothing and softening textures. In some cases, image artifacts such as specular reflections and motion blurring are spread through their neighboring regions in the image, compromising region quality.
Figure 6.4 shows the effect of using a strong filter along with the gradient flow calculation for lip contour approximation. Notice how region borders get deflected from their initial position by the excessive image smoothing, hence affecting the location of the approximated contour. Figure 6.5, in the other hand, shows how poor filtering causes the contour approximation algorithm to get stuck outside the actual lip contour. This effect is caused by high fluctuations in the gradient flow due to image noise.

Tracking issues

Tracking is usually much less complex in computational effort than region based segmentation, achieving in some cases comparable results. Hence, lip contour detection and tracking has

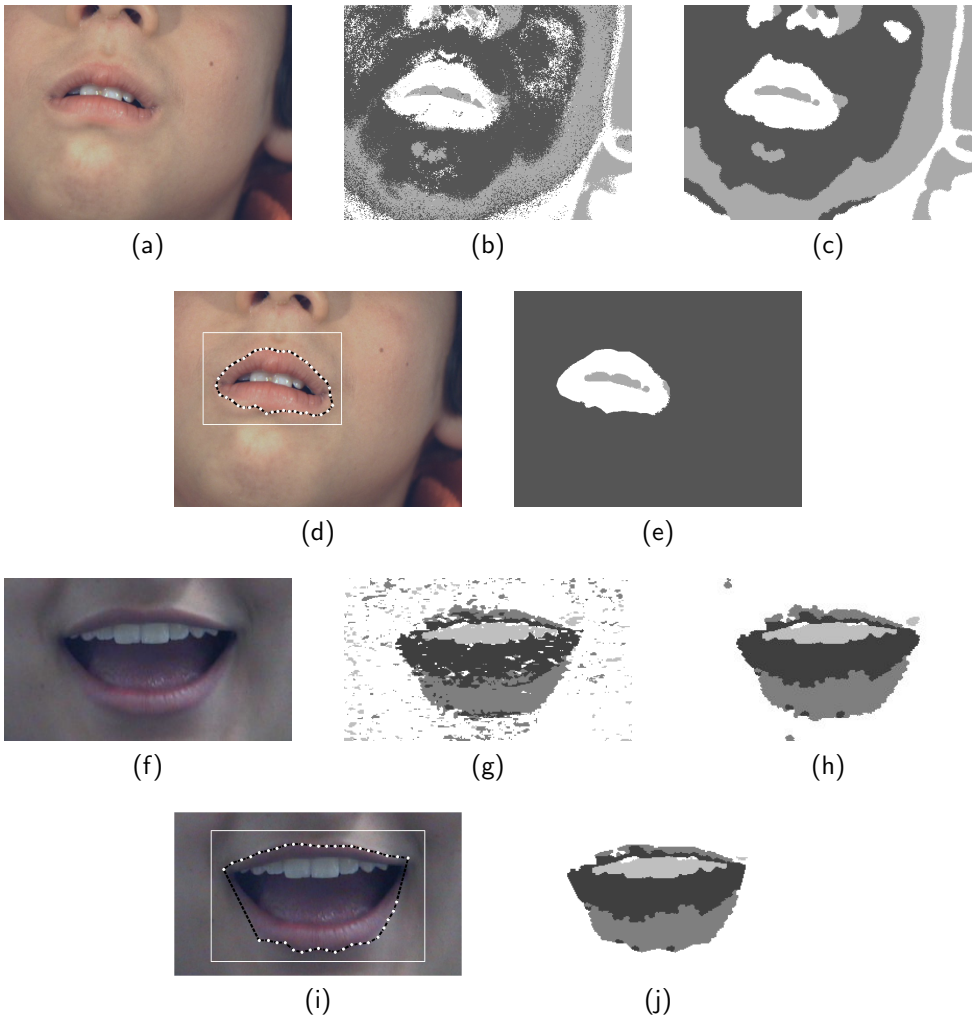
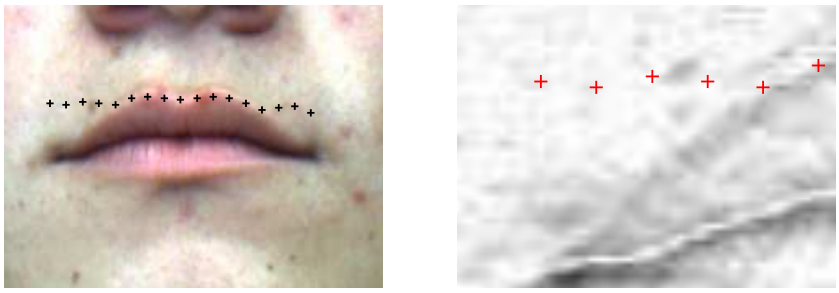


Figure 6.3: Behavior of the contour extraction methodology. (a), (f): original images. (b), (g): initial pixel color classification using GMMs. (c), (h): refined pixel classification. (d), (i): detected RoI and outer lip contour. (e), (j): final segmentation with contour-based RoI clipping.

(a) Base image.

(b) Local gradient approximation near the left mouth corner.

Figure 6.4: Contour approximation behavior under a excesively smoothed local color distribution.



(a) Base image.

(b) Local gradient approximation near the left mouth corner.

Figure 6.5: Contour approximation behavior under highly variable local color distribution.

been a preferred choice for some authors in order to cope with continuous video tasks (i.e., audio visual speech recognition).

Nevertheless, tracking is not always a better choice over image segmentation when processing video sequences. For instance, landmark detection algorithms tend to fail when there are non negligible changes among consecutive images in the sequence. Moreover, it is difficult in some cases to determine automatically whether a landmark has been properly tracked or not from one frame to the next. This proved to be particularly true in our tests, where subjects performed movements that change considerably mouth appearance at speeds close to that of the sensor.

Figure 6.6 shows a clear example of this situation using two consecutive frames from a sequence acquired at 50fps. Notice how the subject is able to change from a clear tongue pointing downwards gesture to a rest gesture from one frame to the next.

(a) Base frame.

(b) Next frame.

Figure 6.6: Gesture change between consecutive frames in a 50fps facial video sequence.

6.5 Summary

In this Chapter, a modification to the algorithm introduced in [1, 2] is proposed. The method encompasses the progressive update of two active contours–one for the upper lip, the other for the lower lip–which encloses tightly the mouth region.

The foundation of the algorithm is an iterative maximization of the gradient flow through a series of line segments conformed by the points in the active contour. The proposed modification performs a two-way propagation of the update mechanism, thus taking the most benefit from re-computed contour point location. This novel approach reduces both the number of iterations needed to achieve full contour convergence while reducing the approximation error. The technique exhibit great accuracy for outer lip contour extraction when image conditions permit a proper gradient flow extraction. This, however, cannot be guaranteed for every facial image in a video sequence, since image noise cannot be estimated precisely nor compensated without compromising border definition. Also, common image artifacts such as specular reflections also compromise the local gradient behavior in the base color representations. Current difficulties in detecting whether or not the contour has properly reached the mouth contour presents the method as a usable precise alternative for assisted mouth RoI delimitation, but not as an advisable alternative for fully automatic outer lip contour detection.

7 Automatic mouth gesture detection in facial video sequences

In this Chapter, a novel approach to mouth gesture recognition based in the mouth structures visible in the images is proposed. The methodology takes into account the basic scheme presented at the beginning of this document. Figure 7.1 presents a clear flowchart summarizing the proposed mouth gesture recognition proposal. This diagram is an updated version of the experimental workflow shown in Figure 2.5, and later modified in Figure 5.1. It covers the whole streamline, emphasizing in the actual sub-processes suggested by the proposed methodology at every stage. Unlike prior diagrams, the region characterization stage and the gesture recognition stage are highlighted, as they are briefly treated in this Chapter. Also, numbers enclosed by parenthesis in the Figure indicate the Chapters inside the document which contain the most information regarding the corresponding topic.

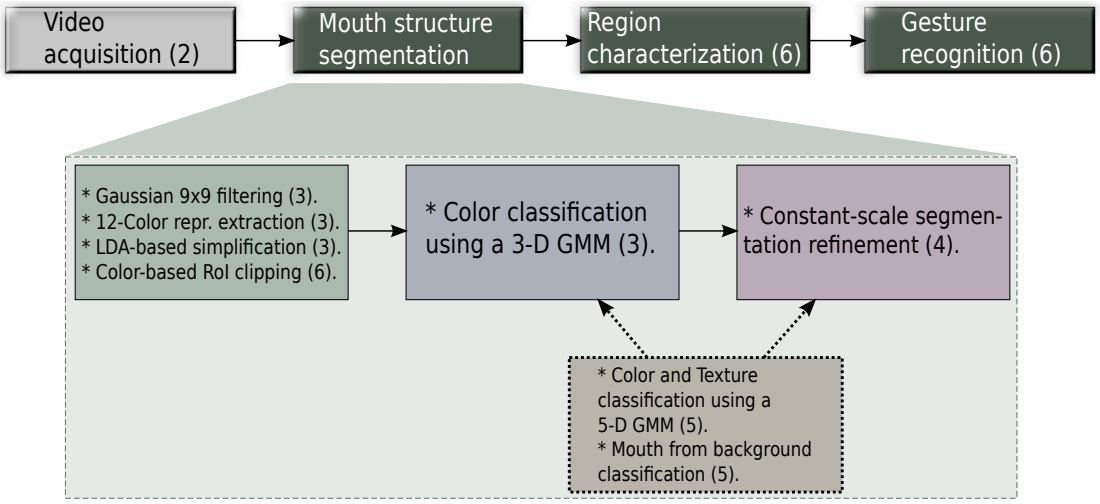


Figure 7.1: Proposed gesture recognition methodology, illustrated with an update to Figure 2.5. Numbers enclosed by parenthesis denote Chapters in this document.

7.1 Problem statement

Current approaches for facial gesture recognition cover fields like virtual avatar animation, security, and with less accuracy, automatic visual speech recognition. Human-machine interfaces based in actual mouth gestures has captured less attention, and typical applications focus in camera autoshoot systems, awareness detection, etc. The scope of this Chapter is related to

mouth-gesture based human-machine interfaces design; particularly, a study case encompassing the laparoscope movement assessment in robotic surgery is treated, stepping into each stage in automatic gesture recognition in video images.

7.1.1 Motivation

The DaVinci robotic surgical system [133] is a tele-operated system composed of three parts: a control console, a surgical arm cart, and a conventional monitor cart with vision system. The control console could be placed on one side of the operating room, or even in an adjoining room. A camera and robot-assisted instruments are controlled by the surgeon from this console with hand manipulators and foot pedals. The surgical arm cart consists of three or four arms with an endoscope and two or three surgical tools. The vision system provides three-dimensional imaging by using a stereo endoscope. Figure 7.2 illustrate the typical set-up of the system.

Figure 7.2: DaVinci surgical system set-up. Taken from Intuitive Surgical® homepage.

The surgical instruments and the camera, which are carried by the arms, are operated by the manipulation of two master controls on the surgeon's console. The surgeon has a visual feedback of the intervention area by the means of an endoscope connected to a stereoscopic viewer. In order to have a better view of the intervention area, the surgeon is forced to leave the joystick-based control of the instruments by pushing a pedal, hence enabling the camera command. In order to take back the instrument control, the surgeon must leave the camera by performing a similar sequence of movements. That swapping between camera and instrument command introduces small delays in surgery time, and in some cases affecting the surgeons concentration. Even when endoscope movements are not as frequent as tool movements (there can be more than fifteen minutes between two endoscope movements in a common intervention), it is desirable to perform both tool and endoscope movements at the same time.

Table 7.1: Approaches for endoscope holder robot command.

	Voice Command	*Joystick*	*Face movements*	*Tool-guided*	*Pedal*
Automation level	Assisted	Assisted	Assisted	Automatic	Assisted
Response time	**Considerable**	Negligible	Negligible	Negligible	Negligible
Expected Precision	Medium	High	Medium	High	Medium
Tremor	Absent	**Present**	Absent	**Present**	Absent
Speed	**Slow**	Variable	Variable	**Slow**	**Slow**
Hands-free	Yes	**No**	Yes	Yes	Yes
Feet-free	Yes	Yes	Yes	Yes	**No**

In the past twenty to thirty years, several approaches for automatic or assisted endoscope holder command have been developed. Table 7.1 presents briefly a summary of common approaches used to cope with the aforementioned task. Endoscope movements are not very frequent during intervention; however, current approaches based in spoken commands, joysticks, etc., do not comply with both time and accuracy constraints at the same time. In the other hand, tool-based guidance systems are aimed towards tool tracking and following, thus introducing tremor and making it impossible to decouple the endoscope movements from those of the tool (which is desirable in some cases).

The current benchmark holder is the speech-based system, which use a reduced set of voice commands in order to lead the endoscope movements [134]. Using voice commands, the surgeon is able to decouple tool and endoscope movements and to avoid introducing tremor in the movements, while escaping from the stress generated with more invasive command techniques. Nevertheless, voice commands take some time to be processed which is reflected in a noticeable time lag between thinking in the movement and actually performing it[1]; also, speech-based systems usually command endoscope movements at pre-defined speeds [117].

Since the console is usually located at least three meters far from the patient, there is no inherent need of covering the surgeon's mouth, and then mouth gestures can be used to command the endoscope. They can also be seen an interpreted at a speed corresponding to the camera's grabbing frame rate. Hence, mouth gestures arise as a natural alternative to existing interfaces.

7.1.2 Previous work

In a first approach, combinations of three distinguishable mouth gestures and bi-axial head movements were used in order to command three degrees of freedom of a robot [135, 27]. The gesture set was composed by resting position, wide open mouth and tightly closed mouth hiding part of the lips. Results from this first approximation led us to propose two new alternatives for the endoscopic camera command. The first one, a mouth gesture based approach that uses a set of basic mouth movements in order to drive the robot movements. The second one, a voice command based system which detects a small set of command words, like in [136].

[1] In a simple study presented in [117], an english voice command set was measured to range between 200ms and 700ms in duration–the time needed to utter them properly, with an average of 500ms.

7.1.3 Limitations and constraints

Non-permanent access to the system led to relax the data acquisition process to lab-based video sequences, somehow compliant with the lighting and pose conditions given by the physical examination of the DaVinci console. Since sequences were acquired under several daylight conditions, no hardware-based color calibration was to be performed prior to data acquisition. Following the advise from an expert in assisted surgery using the DaVinci system, the mouth gesture set was constrained to contain seven possibilities: resting mouth position (R), wide open mouth (OM), closed mouth showing teeth (Th), open mouth with tongue pointing up (TU), open mouth with tongue pointing down (TD), open mouth with tongue pointing left (TL), and open mouth with tongue pointing right (TR). An example of the gesture set can be seen in Figure 7.3.

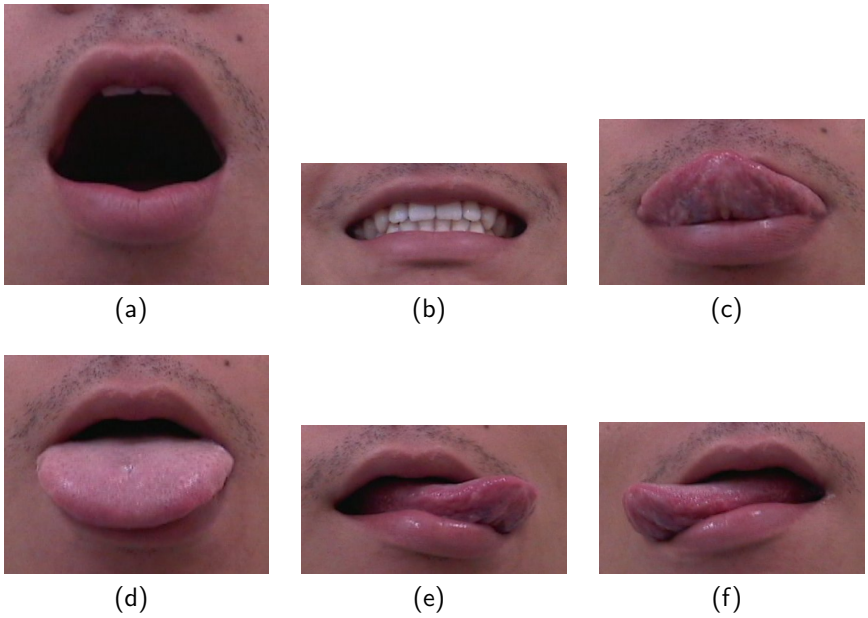


Figure 7.3: Selected mouth gesture set (rest position excluded).

7.2 Acquisition system set up

Figure 7.4 presents an approximate lateral view of the acquisition set-up. The black circle in the Figure represents the location of the spotlights, projected in the lateral view. In the actual set-up, two light were positioned at the left and right sides of the camera, both of them pointing slightly upwards towards the mouth. Ambient lighting was not completely controlled, thus introducing a noticeable chromatic variation in the data[2].

[2]The ambient light hit directly the left cheek of the subjects, thus generating a yellowish glow in this side of the face. This is evidenced all throughout the database.

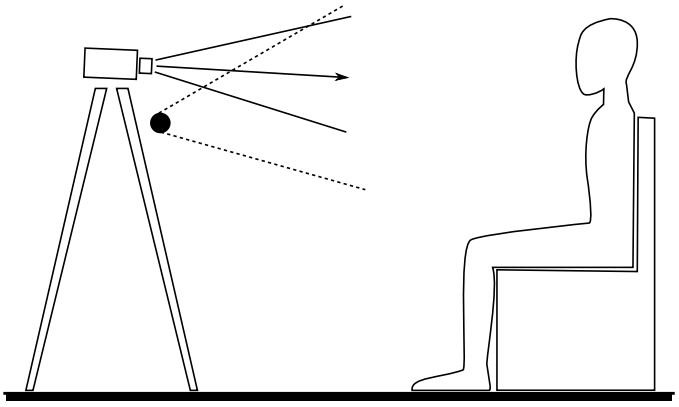


Figure 7.4: Illustrative diagram of the acquisition set-up: lateral view approximation.

7.3 Mouth structure segmentation

According to the scheme presented in Figure 7.1, the first step in automatic mouth gesture recognition in images encompasses a series of steps that lead to mouth structure segmentation. The streamline is composed by an image pre-processing stage, a pixel-color classification engine, a label refinement stage and a final region trimming stage.
The Section summarizes the combination of techniques which proved to be the fittest in the results presented in Chapters 3 to 5, aiming towards a fast [3] and accurate mouth structure segmentation. The following sections convey not only basic information about the techniques but also information regarding parameter selection and special considerations. Also, a region trimming method which has been specifically designed to work with our database is described briefly (an alternative replacement for RoI clipping and region trimming is presented in Section 6).

7.3.1 Pre-processing and initial RoI clipping

Based on the results obtained in Chapter 3, images were treated using a 9×9 Gaussian low-pass filter in order to reduce noise effects without compromising excessively in terms of quality. Image illumination was partially controlled in most of the database; hence, lighting related issues were neglected in later processes.
There are approaches in literature for mouth RoI clipping, most of them based in simple per-pixel color operations. In this work, RoI clipping is carried out by using per-row and per-column color profiles. As an example, Figure 7.5 shows the result of detecting RoI based in the Pseudo-Hue (ph) color representation of the input image. First, the skin region is segmented using a simple comparison between the ph color representation of the image and a pre-defined value (in this case, 0.46). Small gaps and holes are corrected by applying a morphological opening operation followed by an erosion, both of them using a radial structuring element.
Once the face region is separated from the background, a column profile conformed by the

[3] Close to standard video frame rates.

normalized summation of each row of the ph is computed[4], as seen in the lower left part of the Figure. The horizontal mouth axis (the red line in the lower right part of the figure) is chosen to be located at the row that corresponds to the maximum value in the column profile, while the closest minima in the upper and lower side of such axis determine the top and bottom limits of the RoI.

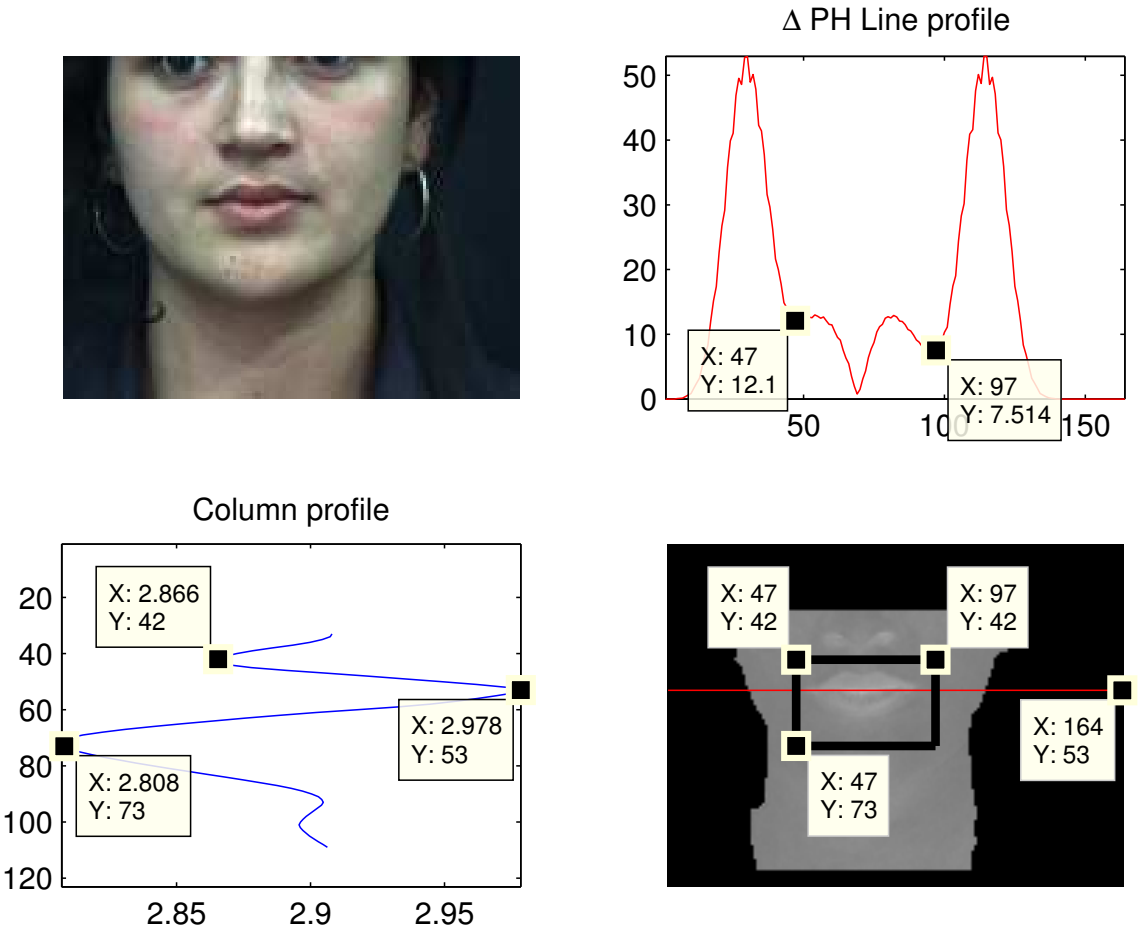


Figure 7.5: RoI clipping based in color profiles.

Then, the ph row profile corresponding to the main axis is used to determine the location of the left-most and right-most margins of the RoI. Good choices for those limits are selected as the closest local minima in the row profile which lie inwards between the two main maxima, as seen in the upper right part of Figure 7.5. The two maxima correspond to those points in the profile that separate background from skin. It is noteworthy that both the column profile and the row profile are heavily smoothed in order to avoid falling in undesired local minima.
The methodology presented for RoI clipping is relatively fast (taking around 1ms per image), and presents acceptable results for most of the subjects. However, its performance is not sustainable for non-resting mouth positions, since presence of teeth generate serious variations in the ph profiles. Hence, working with video sequences imply alternating between RoI detection (carried out using the proposed methodology) and tracking.

[4]Notice that only data from inside the face region is taken into account in conforming the profiles.

7.3.2 Mouth segmentation through pixel classification

Mouth structure segmentation is done by following the recommendations in Chapters 3 and 5, where color-based pixel classification using Gaussian mixtures outperforms other approaches in the compromise between computational reckoning and segmentation quality (measured in terms of DTO). The pixel color classification based mouth segmentation methodology, extracted from the best performing technique combination, encompasses the following stages:

- The basic scheme is comprised by representing pixel color in an augmented feature space containing the basic RGB components, along with other nine enunciated in Section 3.1.
- Next, a three-fold $FLDA$ projection is used in order to reduce the feature set dimension.
- The resulting three-dimensional feature vectors are fed to a pixel classifier which uses a set of four previously trained Gaussian mixture models–one model per structure, three Gaussians per model. Labels are assigned depending on how likely is every pixel to belong to each class.

Since each pixel is treated independently, the segmentation process can be carried out in parallel if the underlying platform supports it. Computational time is basically affected by the complexity of the color transformations, and is comparable with the one resulting from applying a 9×9 linear filter in the image, if executed in mainstream hardware. A deeper insight on the computational complexity involved in computing the color transformations and the pixel classification can be seen in Chapters 2 and 3.

7.3.3 Label refinement

Chapter 4 proved that the proposed refinement algorithm improves DTO in most cases for pixel color based classification for mouth structure segmentation. Also, the refiner presents a more consistent behavior if a fixed scale is selected for all the pixels in the image (as shown in Chapter 5). Particularly, it has been proven that choosing $\sigma \in [1.0, 2.0]$ leads to adequate results in most cases. Particularly, σ was set to 1.0 for the tests whose results are presented in the remainder of this Chapter. The refiner was set to iterate ten times in the label images.

7.3.4 Texture based mouth/background segmentation

As shown in Chapter 5, texture information complements color in mouth from background distinction. Consequently, the use of texture and color classification is advised in order to reduce the number of resulting spurious regions and undesired extensions in base regions, commonly found in bare color pixel based classification.
Fifteen images from different subjects were randomly selected from the annotated part of the database.
It is noteworthy that for the bi-class classification problem, both classes present the same DTO. Hereby, the last two rows of the Table hold the same values.
The computing time inherent to texture features is, in general, higher than most linear and non-linear filtering or morphological operation in the pixels. Due to this fact, texture features

Table 7.2: Effect of Texture-and-color based segmentation masking, measured using DTO.

	Whole image		Inside RoI	
	Base	Masked	Base	Masked
Lips	0.8114	**0.6210**	0.4166	**0.2966**
Teeth	0.9825	**0.9508**	0.7315	**0.5379**
Tongue	0.8932	**0.7133**	0.5938	**0.4994**
Background	0.9138	**0.7884**	0.4264	**0.2491**
Mouth region	0.9138	**0.7884**	0.4264	**0.2491**

are used only once mouth ROI has already been approximated, thus reducing the computational complexity of the masking process.

7.3.5 Region trimming using convex hulls

The last stage in the proposed segmentation methodology comprises the use of the biggest connected lip region and tongue region in order to establish if a given labeled connected region should be trimmed out or preserved. The assumption states that all preservable regions must be contained inside the convex hull conformed by the lips and the tongue. This is particularly true for unadorned mouths without prosthetic modifications, and if the camera is facing directly to the subject.
Figure 7.6 shows an example of the texture based RoI clipping complemented by region trimming. Notice that the texture based RoI clipping aids in removing some misclassified regions along mouth contour, while the region trimming finally cuts down spurious regions that may have outlived RoI clipping.
Table 7.3 shows DTO measures for the images in Figure 7.6. Notice that DTO associated to background improved dramatically after the convex hull based region trimming, indicating a huge improvement in mouth from background distinction. Despite lip and tongue regions DTO don't show numerical improvement after trimming, the overall appearance of the mouth seems to be cleaner and well defined.

Table 7.3: Illustrative DTO comparison for RoI clipping and region trimming in Figure 7.6.

	Lips	Teeth	Tongue	Backg.
Base	**0.2972**	0.2199	**0.4627**	0.4405
Tex. Clip.	0.3235	0.2004	0.4837	0.3440
C. Hull trim.	0.3230	**0.1479**	0.4828	**0.1662**

7.4 Mouth gesture classification

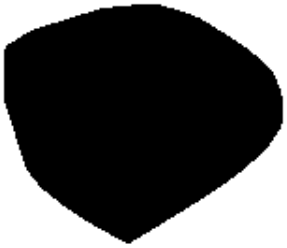
The last portion of the recognition scheme encompasses the mouth region characterization and posterior gesture detection. The remarkable results obtained in the previous stages, particularly

(a) Original RoI clipped image.

(b) Initial pixel color classification after refinement.

(c) Texture-based clipping mask.

(d) Refined color-based segmentation after texture-based clipping.



(e) Convex hull of the biggest lip and tongue region.

(f) Resulting region trimming.

Figure 7.6: Example of the use of texture-and-color based segmentation masking, trimmed using the convex hull of the biggest lip and tongue region.

in determining the mouth boundaries and shape, are exploited in choosing rather simple and fast region feature extraction and gesture classification techniques. In the remainder of this Section both processes are discussed.

7.4.1 Region feature selection

Mouth region segmentation delivers a well delimited mouth region, which permits an easy geometric characterization of the mouth area. Commonly used features that exploit geometric properties of the regions are the Center of Mass (CoM) location, the mass (number of pixels conforming the region), aspect ratio, tortuousness, circularity, etc. In this work, a subset composed by eleven geometric measurements is used for region characterization. Those features were selected so their computation does not imply significant increases in complexity, yet subjectively conveying enough information to perform an adequate gesture classification using the chosen gesture set. The measurements are computed using the convex hull of a mouth in resting position as reference. The measurements are enunciated and described briefly in Table 7.4.

7.4.2 Gesture classification

Once all region data is codified according the geometric feature set presented in the previous Subsection, a set of bi-class classifiers based in $FLDA$ is constructed. Each classifier is chosen to maximize class distinction between each gesture and the rest of the data, producing a set of seven projection vectors and seven comparison thresholds. Table 7.5 shows the frame composition rates of the "Own" database regarding its gesture contents.
Once projected using the $FLDA$ vectors, each resulting feature vector serves to classify a particular gesture in a "One against the rest" scheme. The result of comparing the projected feature vector with the seven thresholds can lead to a true for the pattern to belong to more than one class. In those cases, the gesture is marked as "Undefined", and for the test database took a 2.33% of the total of frames.
Table 7.6 summarizes the classification results obtained using the proposed scheme. It should be remarked that resulting DTOs for all gestures lie very close to the human variability measured for the database, thus being located inside the error tolerance region.
Figure 7.7 shows the results of gesture classification for a $CWDLC$[5] sequence. Each Subfigure represents a different gesture, and the three signals in them presents the ground truth, the gesture classification result and the classification error. From the Figures, it is noticeable that a considerable amount of the detection errors is located in transitional areas where the subject is changing his/her pose between two different gestures.

7.4.3 Gesture detection stabilization

Frame-based gesture classification brings results that can be physically unfeasible, like in example, instant gesture changes directly from "TD" to "Th". Nevertheless, temporal constraints can be used in order to stabilize such undesired behavior.

[5]The $CWDLC$ and $CCWS$ sequences are described in Section 2.2.1.

Table 7.4: Geometric feature set used for gesture classification.

Feature	Description	No. of ind.
Lip region CoM	Location of the lip region CoM relative to the resting mouth convex hull's CoM location.	2
Teeth region CoM	Location of the teeth region CoM relative to the resting mouth convex hull's CoM location.	2
Tongue region CoM	Location of the teeth region CoM relative to the resting mouth convex hull's CoM location.	2
Lip mass proportion	Number of pixels inside the convex hull classified as lip pixels, divided by the total number of pixels lying inside the resting mouth's convex hull.	1
Teeth mass proportion	Number of pixels inside the convex hull classified as teeth pixels, divided by the total number of pixels lying inside the resting mouth's convex hull.	1
Tongue mass proportion	Number of pixels inside the convex hull classified as tongue pixels, divided by the total number of pixels lying inside the resting mouth's convex hull.	1
Free mass proportion	Number of pixels inside the convex hull classified as background, divided by the total number of pixels lying inside the mouth's convex hull.	1
Normalized aspect ratio	Convex hull aspect ratio divided by resting mouth's aspect ratio.	1
TOTAL		11

Table 7.5: Frame composition rates of the "Own" database.

	Rest	T. Up	T. Down	T. Right	T. Left	Teeth	Open	Undef.
Init.	64.80	0	0	0	0	22.05	11.63	1.51
CCWS	60.57	7.37	8.97	7.25	9.09	0	0	6.76
CWDLC	56.95	4.24	3.76	4.24	7.9	7.66	7.53	7.74

Table 7.6: Gesture classification accuracy measured using DTO.

	Rest	T. Up	T. Down	T. Right	T. Left	Teeth	Open
Init	0	N.A.	N.A.	N.A.	N.A.	0.0058	0.0017
CCWS	0.0249	0.0053	0.0027	0.0013	0.0136	N.A.	N.A.
CWDLC	0.0558	0.056	0.0088	0.0128	0.0286	0.0046	0.0078

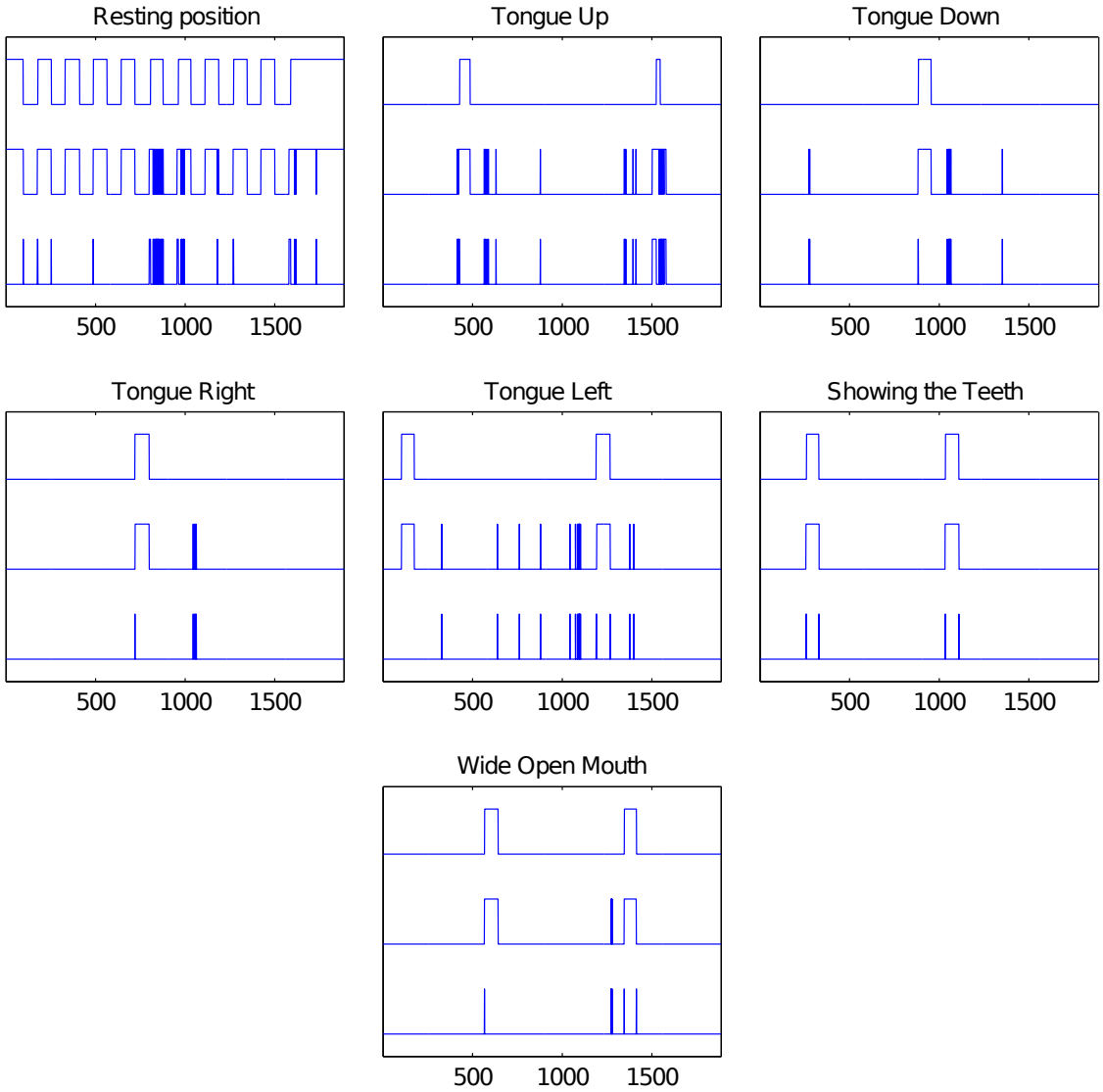


Figure 7.7: Example of gesture classification using the $CWDLC$ sequence. For all Sub-Figures, the topmost signal represents the ground truth, the medium signal represents the gesture detection result, and the bottommost signal show the instant error. The horizontal axis presents the frame number.

A first approach, presented in [4], makes use of a state machine which forbids changes between detected states that may be result from misinterpretation. The technique was applied to a reduced gesture set which included mixtures of head displacements and base gestures as completely new gestures.

In this document, a simpler approach for stabilizing gesture classification is proposed. The technique is based on a decision rule that verifies the existence of two consecutive detections of the same gesture followed by an undefined gesture. In that case, the former is kept instead of setting the output as "Undefined".

Figure 7.8 shows the effect of using the aforementioned decision rule in order to stabilize gesture detection. Notice that almost every "Undefined" gesture occurrence has been replaced with a feasible gesture, and the actual resulting gesture sequence resembles closely the CWDLC gesture sequence definition as stated in Section 2.2.1.

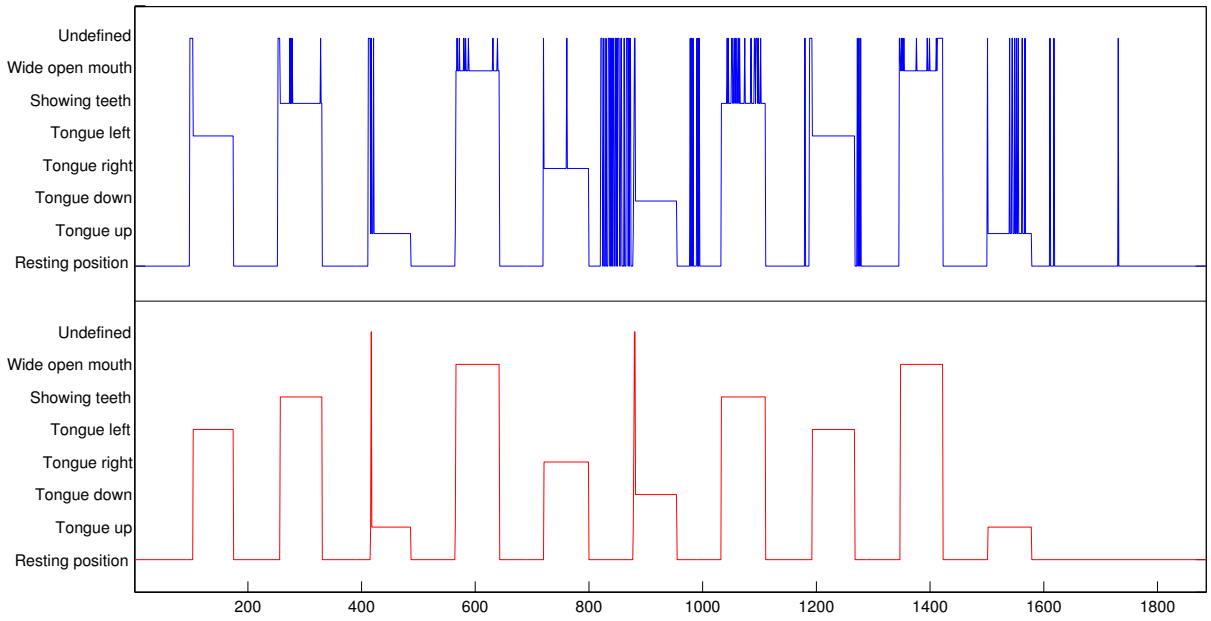


Figure 7.8: Effect of temporal stabilization in gesture detection in a CWDLC example sequence. The upper signal represents the initial gesture detection, and the lower signal represents the stabilized detection. The varying levels in the vertical axis represent the gesture, while the horizontal axis presents the frame number.

Despite of the inherent simplicity of the technique, the resulting gesture sequences corresponds almost completely to the actual gesture match and duration in the sequences. It is noteworthy that the use of such a simple rule for stabilization is possible due to a very accurate gesture detection stage. Also, it should be noticed that using two past frames in making the decision introduces a time lag that, in the worst case, corresponds to the time taken by two frames. That time is, for NTSC and PAL video standards, much smaller than the time needed to process a typical voice command [4].

7.5 Summary

In this Chapter, a methodology for automatic mouth gesture detection in images is discussed. The methodology is focused in detecting the gestures contained in the "Own" database described in Chapter 2, which in turn are the result of a selection process aimed towards human machine interface development for assisted surgery with the DaVinci surgical system.
The methodology, comprised by image pre-processing and segmentation, region feature extraction and gesture classification, proved to exhibit high accuracy for the selected gesture set. The results were further improved by the use of a gesture stabilization mechanism that forbids sudden changes between gestures by correcting undefined gestures detection.
In the image segmentation stage, a new method for RoI clipping and region trimming was discussed. The RoI clipping technique, along with the convex hull based region trimming, improves dramatically the DTO measure for mouth from background distinction. These two procedures are nevertheless non-advisable for any possible application since they were designed having in mind images with frontal face poses with unadorned mouths and a compensated lighting environment. Hence, those results may not hold for uncontrolled environments with face deviations and aesthetic or prosthetic facial modifications.
The overall time spent to process each image ranges between 600ms and 1200ms. The computational complexity indicators given throughout the document for the techniques used in the gesture detection scheme leads to think that an optimal implementation of the suggested methodology could achieve up to 6 frames per second in mainstream hardware. Although this speed is not considered real-time if compared with standard video formats, it may be sufficient for low speed appliances.

8 Conclusion

This document is focused in the introduction of a methodology for automatic mouth gesture recognition in images. The methodology comprise several stages from image pre-processing to gesture classification, choosing in each stage feasible techniques aimed towards the obtention of a good compromise between speed and recognition accuracy. Special attention is payed in pixel color representation, region color modeling, image filtering, pixel color classification, label segmentation refinement. Other topics such as the use of texture descriptors for mouth/skin distinction and geometric region feature extraction are also treated briefly.

The methodology, comprised by image pre-processing and segmentation, region feature extraction and gesture classification, exhibits high accuracy for the selected gesture set. The results were further improved by the use of a gesture stabilization mechanism that forbids sudden changes between gestures by correcting undefined gestures detection.

Color representation and modeling was carried out by starting with a 12-dimensional feature space conformed using diverse color representations of the image pixel data. Then, stochastic models of region color were approximated in order to characterize lips, tongue, teeth and background color using a subset of a proprietary database (widely referenced as "Own" throughout the document). Measurements obtained in the conducted tests revealed that Neural Networks exhibit higher accuracy in color distribution modeling when using a 12-dimensional input feature vector than Gaussian mixtures. This effect is reversed when only three features were used. There is a considerable reduction in computational complexity when downscaling from 12 features to three; at the same time, a barely noticeable decrease in accuracy was obtained by performing that change. Thereby, results presented in following chapters were referred to a three-dimensional feature Gaussian mixture model. The models use the configuration described in Section 3.2.4. High variations in color classification accuracy were detected when using data provided by different databases, higher than those obtained by changing subjects within the same database. The tests clearly illustrate the complexity of isolating the influence of issues related to acquisition set-up from the final color register in the images. Complementary tests varying among different pre-processing combinations resulted in the selection of a low pass Gaussian filter with a window size of 9×9 pixels as the best performing one, using as reference the average mouth size of the proprietary image database.

As a side contribution of the study conducted is a fast alternative for coarse lip/skin segmentation based in pixel classification (Section 3.1.3). The segmentation technique is based in the use of the CIE$a*$ color component, with its value normalized using the values inside the mouth's region of interest (RoI). Classification results proved to be better than those obtained using other color components commonly used in lip/skin segmentation through pixel color thresholding.

The region color models enable a quick pixel color classification that can be used as a starting point for mouth structures detection. However, the results obtained by classifying color gen-

erate label images with spurious regions and gaps within regions. Hence, a new segmentation refinement technique is introduced. The refiner is composed by a layer of perceptual units (one per pixel), each of them connected to one unique input label pattern, and to neighboring units' output. Two parameters, which are proven to be at some extent correlated in Sections 4.2 and 4.3.1, control the compromise between input labeling and field effect through iterations. The technique mimics the smoothing effect of low pass filters applied to labeling information, and its computational cost per iteration is also about the same as the one of such kind of filters. Refiner's behavior is analyzed in depth in Section 4.2, and numerical results are also provided in Sections 4.3 and 4.4.

In most cases, the refiner improves the output labeling resulting from unsupervised pixel color based segmentation of natural images. In the case of supervised mouth structures segmentation, the benefit is clearer by improving the results in all cases. The improvement is at some extent cumulative with the one obtained by the means of image pre-processing, thus proving to be complementary techniques. Individually, linear filtering and segmentation refinement increase segmentation accuracy by 5% to 10% approximately (reflected in DTO), while the combined effect of both techniques lead to an increment of 15% approximately. It is noteworthy that the computational complexity of each refinement iteration is comparable with that of the linear filter, and that the refiner usually takes between five and fifteen iterations to converge.

Image pre-processing proved to benefit pixel color classification, notably through the use of fixed-scale low pass linear filters (Chapters 3 and 4). Particularly, the use of a 9×9 Gaussian filter improved pixel classification DTO for all mouth structures. In this Chapter, the Gaussian filter's size was made variable in terms of local scale, using the measured *integration scale* for every pixel. Results of image filtering with the scale variable filter expose a clear retention of structure borders while smoothing the color information within each region. Nevertheless, features such as specular noises and strongly variable textures (like the bright hatched pattern in the lips) also remain after filtering. Hence, pixel classification performance was not clearly improved by the scale-variant filtering, as opposed to the fixed scale version. This fact makes it advisable to use a fixed-scale filter over a scale variant version.

In the next stage of the segmentation process, texture descriptors are used as part of the feature vector fed to the pixel classification engine. Texture is characterized using a reduced set of low-level features, and two more features derived from the *integration scale*, known as local contrast and anisotropy. The augmented feature set show a considerable improvement in mouth from background distinction, but the addition of the texture features raised the confusion between lips and tongue regions. The results of the conducted tests indicate that a good practice can be derived from the mixed use of texture and color features for initial mouth selection, and then the use of color-only features for structure from structure classification.

Finally, the *integration scale* was used to set up automatically the scale parameter σ for segmentation refinement. As in the case of the scale-variant filtering, the lack of continuity in the scale among neighboring pixels led to refinement results exhibiting poorer results than those obtained with the presets found in Chapter 4. At the end, texture proved to be particularly helpful in pixel color classification for mouth from background distinction, but its usage is bound to the quality/performance compromise for every particular application. However, its use in pre-processing and segmentation refinement in mouth structure classification can be

safely avoided.
In the image segmentation stage, a new method for RoI clipping and region trimming was discussed. The RoI clipping technique, along with the convex hull based region trimming, improves dramatically the DTO measure for mouth from background distinction. These two procedures are nevertheless non-advisable for any possible application since they were designed having in mind images with frontal face poses with unadorned mouths and a compensated lighting environment. Hence, those results may not hold for uncontrolled environments with face deviations and aesthetic or prosthetic facial modifications.
The overall time spent to proceed with the gesture recognition for each image varies between 600ms and 1200ms with the current implementation, which restricts its use to offline processing systems. The computational complexity indicators given throughout the document for the techniques used in the gesture detection scheme leads to think that an optimal implementation of the suggested methodology could achieve up to 6 frames per second in mainstream hardware. Although this speed is not considered real-time if compared with standard video formats, it may be sufficient for low speed gesture detection applications. It is noteworthy that the computational complexity is also associated to the size of the mouth in the image (in pixels), and can henceforth be reduced by sacrificing in segmentation accuracy.

9 Open issues and future work

Segmenting mouth structures for dark skin subjects is still the most challenging issue in mouth segmentation. The color information contained in the zone between the lips and the skin present high variability, low discriminance and higher noise levels than those found in brighter skins. Most color representations fail completely to codify the difference between lips and skin, while the others perform poorly. Thereupon, the need of a more robust color and/or texture representation that enables an accurate lips-from-skin differentiation arises.

The modified version of the mouth contour extraction introduced in [1, 2] that is proposed in this work exhibited an outstanding accuracy in outer lip contour approximation for images acquired under controlled conditions. Nevertheless, its robustness is put to test under any slight variation of these conditions that may increase noise or unstabilize the local gradient approximation. This fact opens two derived tasks that must be tackled in order to improve the contour approximation: first, the improvement of a local gradient approximation that proves to be robust against a wider noise level range; and second, the generation of a measure that codifies how fit the approximated contour points are to the actual location of the mouth contour in the image. The first task can be tackled through the development of an adaptive pre-processing filter that copes with noise while complying with preserving important features in facial images. In the other hand, the second task can be tackled by mixing the contour approximation method with a fast pixel classifier; in that way, the system would be able to estimate how much of the information contained inside the approximated contour correspond to mouth information, and how much of the mouth was left outside that contour.

The segmentation refinement technique proposed in this work proved to be complementary to image pre-processing in improving mouth structure segmentation. The results do depend on the refiner parameter selection, which in turn control the number of iterations needed for the algorithm to converge. A first approximation to the automatic parameter selection by means of local scale failed to perform better than static parameter selection, leaving a door open for new automatic parameter selection strategies to emerge. Such strategies should arise to improve both with final labeling accuracy and the number of iterations needed to achieve the former goal.

One of the biggest limitations when working in image segmentation, and particularly in mouth structure segmentation, is related to segmentation quality measurement. General purpose measures such as those derived from the confusion matrix, do not express properly some perceptual concepts such as shape conformance, overall shape definition, context conformance, etc. In the other hand, humans fail dramatically when trying to adjust subjective perception to machine-derived assessment, as evidenced in the human variability tests performed in the "Own" database. Hence, human expertise is still an isolated source of information waiting to be imbued in artificial vision solutions. This makes segmentation measure development an important yet very open issue in artificial vision.

Bibliography

[1] N. Eveno, A. Caplier, and P.Y. Coulon. New color transformation for lips segmentation. In *2001 IEEE Fourth Workshop on Multimedia Signal Processing, Cannes (FR), 3-5 Oct.*, pages 3–8, 2001.

[2] N. Eveno, A. Caplier, and P.Y. Coulon. Accurate and quasi-automatic lip tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(5):706–715, May 2004.

[3] J.B. Gómez, F. Prieto, and T. Redarce. Towards a mouth gesture based laparoscope camera command. In *IEEE International Workshop on Robotic and Sensors Environments, Ottawa (CA), 17-18 Oct.*, pages 35–40, 2008.

[4] J.B. Gómez, A. Ceballos, F. Prieto, and T. Redarce. Mouth Gesture and Voice Command Based Robot Command Interface. In *Proc. of the IEEE International Conference on Robotics and Automation ICRA'09, Kobe (JP), 12-17 May*, pages 333–338, 2009.

[5] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Pearson Education, 3rd edition, 2007.

[6] D. Jian-qiang, L. Yan-sheng, Z. Kang, Z. Ming-feng, and D. Cheng-hua. A Novel Approach of Tongue Body and Tongue Coating Separation Based on FCM. In *Proceedings of the 2nd International Conference on Bioinformatics and Biomedical Engineering ICBBE 2008, Shanghai (CN), 16-18 May* , pages 2499–2503, 2008.

[7] Z. Fu, W. Li, X. Li, F. Li, and Y. Wang. Automatic tongue location and segmentation. In *Proceedings of the International Conference on Audio, Language and Image Processing ICALIP 2008, Shanghai (CN), 7-9 July*, pages 1050–1055, 2008.

[8] L. Jiang, W. Xu, and J. Chen. Digital imaging system for physiological analysis by tongue colour inspection. In *Proceedings of the 3rd IEEE Conference on Industrial Electronics and Applications ICIEA 2008*, pages 1833–1836, 2008.

[9] Z. Yan, K. Wang, and N. Li. Segmentation of sublingual veins from near infrared sublingual images. In *Proceedings of the 8th IEEE International Conference on BioInformatics and BioEngineering BIBE 2008, Athens (GR), 8-10 Oct.*, pages 1–5, 2008.

[10] T. Kondo, S.H. Ong, and K.W.C. Foong. Tooth segmentation of dental study models using range images. *IEEE Transactions on Medical Imaging*, 23(3):350–362, 2004.

[11] Y.H. Lai and P.L. Lin. Effective Segmentation for Dental X-Ray Images Using Texture-Based Fuzzy Inference System. *Lecture Notes In Computer Science*, 5259:936–947, 2008.

[12] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[13] V. Vezhnevets, V. Sazonov, and A. Andreeva. A survey on pixel-based skin color detection techniques. In *Proceedings of GraphiCon 2003, Moscow (RU), 5-10 Sep.*, page 8pp, 2003.

[14] Y.P. Guan. Automatic extraction of lip based on wavelet edge detection. In *Eighth International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, 2006. SYNASC '06, Timisoara (RO), 26-29 Sep.*, pages 125–132, 2006.

[15] Y. Guan and L. Yang. An unsupervised face detection based on skin color and geometric information. In *Proceedings of Sixth International Conference on Intelligent Systems Design and Applications, ISDA 2006, Jinan (CN), 16-18 Oct.*, pages Vol. 2. 272–276, 2006.

[16] L.E. Morán and R. Pinto. Automatic extraction of the lips shape via statistical lips modelling and chromatic feature. In *Electronics, Robotics and Automotive Mechanics Conference (CERMA 2007), Cuernavaca (MX), 25-28 Sep.*, pages 241–246, 2007.

[17] G.I. Chiou and J.N. Hwang. Lipreading from color video. *IEEE Transactions on Image Processing*, 6(8):1192–1195, 1997.

[18] A.C. Hurlbert and T.A. Poggio. Synthesizing a color algorithm from examples. *Science*, 239(4839):482–485, 1988.

[19] X. Zhang and R.M. Mersereau. Lip feature extraction towards an automatic speechreading system. In *Proceedings of IEEE International Conference on Image Processing ICIP '00, Vancouver (CA), 10-13 Sep.*, volume 3, pages 226–229, 2000.

[20] N. Eveno, A. Caplier, and P.Y. Coulon. A parametric model for realistic lip segmentation. In *Seventh International Conference on Control, Automation, Robotics And Vision (ICARCV'O2), Singapore (MY), 2-5 Dec.*, pages 1426–1431, 2002.

[21] R. Ma, Y. Wu, W. Hu, J. Wang, T. Wang, Y. Zhang, and H. Lu. Robust lip localization on multi-view faces in video. In *Proceedings of the 2007 IEEE International Conference on Multimedia and Expo ICMA' 07, Beijing (CN), 2-5 July*, pages 1395–1398, 2007.

[22] Y. Nakata and M. Ando. Lipreading method using color extraction method and eigenspace technique. *Systems and Computers in Japan*, 35(3):1813–1822, December 2004.

[23] A. Salazar and F. Prieto. Extracción y clasificación de posturas labiales en niños entre 5 y 10 años de la ciudad de Manizales. *DYNA*, 150:75–88, november 2006.

[24] S.L. Wang, S.H. Leung, and W.H. Lau. Lip segmentation by fuzzy clustering incorporating with shape function. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP' 02, Orlando (US), 13-17 May*, volume 1, pages I–1077–I–1080, 2002.

[25] I. Arsic, R. Vilagut, and J.P. Thiran. Automatic extraction of geometric lip features with application to multi-modal speaker identification. In *2006 IEEE International Conference on Multimedia and Expo ICME' 06, Toronto (CA), 9-12 July*, pages 161–164, july 2006.

[26] Y. Wu, R. Ma, W. Hu, T. Wang, Y. Zhang, J. Cheng, and H. Lu. Robust lip localization on multi-view faces in video. In *Proceedings of International Conference on Image Processing, ICIP 2007, San Antonio (US), 16-19 Sep.*, pages IV 481–484, 2007.

[27] J.B. Gómez, J.E. Hernández, F. Prieto, and T. Redarce. Real-time robot manipulation using mouth gestures in facial video sequences. *Lecture Notes in Computer Science*, 4729(2007):224–233, 2007.

[28] S. Lucey, S. Sridharan, and V. Chandran. Chromatic lip tracking using a connectivity based fuzzy thresholding technique. In *Proceedings of the Fifth International Symposium on Signal Processing and its Applications, ISSPA '99, Brisbane (AU), 22-25 Aug.*, pages 669–672, 1999.

[29] J.M. Zhang, D.J. Wang, L.M. Niu, and Y.Z. Zhan. Research and implementation of real time approach to lip detection in video sequences. In *Proceedings of the Second International Conference on Machine Learning and Cybernetics, Xi'an (CN), 2-5 Nov.*, pages 2795–2799, 2003.

[30] R. Kaucic and A. Blake. Accurate, real-time, unadorned lip tracking. In *Proceedings of 6th International Conference on Computer Vision ICCV' 98, Bombay (IN), 4-7 Jan.*, pages 370–375, 1998.

[31] W. Rongben, G. Lie, T. Bingliang, and J. Lisheng. Monitoring mouth movement for driver fatigue or distraction with one camera. In *Proceedings of he 7th International IEEE Conference on Intelligent Transportation Systems, Washington (US), 3-6 Oct.*, pages 314–319, 2004.

[32] T. Wakasugi, M. Nishiura, and K. Fukui. Robust lip contour extraction using separability of multi-dimensional distributions. In *Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition, FGR 2004*, pages 415–420, 2004.

[33] J.Y. Kim, S.Y. Na, and R. Cole. Lip detection using confidence-based adaptive thresholding. *Lecture Notes in Computer Science*, 4291(2006):731–740, 2006.

[34] J. Loaiza, J.B. Gómez, and A. Ceballos. Análisis de discriminancia y selección de características de color en imágenes de labios utilizando redes neuronales. In *Memorias del XII Simposio de Tratamiento de Señales, Imágenes y Visión Artificial STSIVA' 07, Barranquilla (CO), 26-28 Sep.*, 2007.

[35] J.A. Dargham, A. Chekima, and S. Omatu. Lip detection by the use of neural networks. *Artificial Life and Robotics*, 12(1-2):301–306, march 2008.

[36] G. McLachlan and D. Peel. *Finite Mixture Models*. John Wiley & Sons, 2000.

[37] G. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. John Wiley & Sons, 1996.

[38] S. Basu, N. Oliver, and A. Pentland. 3D lip shapes from video: A combined physical-statistical model. *Speech Communication*, 26(1998):131–148, 1998.

[39] S. Basu, N. Oliver, and A. Pentland. Coding human lip motions with a learned 3D model. In *Proceedings of the International Workshop on Very Low Bitrate Video Coding (VLBV' 98), Urbana (US), 8-9 Oct.*, Oct. 1998.

[40] S. Basu, N. Oliver, and A. Pentland. 3D modeling and tracking of human lip motions. In *Proceedings of the Sixth International Conference on Computer Vision ICCV'98, Bombay (IN), 4-7 Jan.*, 1998.

[41] M. Sadeghi, J. Kittler, and K. Messer. Real time segmentation of lip pixels for lip tracker initialization. *Lecture Notes in Computer Science*, 2124(2001):317–324, 2001.

[42] M. Sadeghi, J. Kittler, and K. Messer. Modelling and segmentation of lip area in face images. In *IEE Proceedings - Vision, Image and Signal Processing*, volume 149, pages 179–184, 2002.

[43] C. Bouvier, P.Y. Coulon, and X. Maldague. Unsupervised lips segmentation based on ROI optimisation and parametric model. In *IEEE International Conference on Image Processing, 2007. ICIP 2007, San Antonio (US), 16-19 Sep.*, volume 4, pages IV–301–IV–304, 2007.

[44] P.H. Kelly, E.A. Hunter, K. Kreutz–Delgado, and R. Jain. Lip posture estimation using kinematically constrained mixture models. In *British Machine Vision Conference, Southampton (UK), Sep.*, page 10, 1998.

[45] P. Gacon, P.Y. Coulon, and G. Bailly. Statistical active model for mouth components segmentation. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP' 05, Philadelphia (US), 18-23 Mar.*, volume 2, pages ii/1021–ii/1024, 2005.

[46] C. Bregler and S.M. Omohundro. Nonlinear manifold learning for visual speech recognition. In *Fifth International Conference on Computer Vision (ICCV'95), Boston (US)*, pages 494–499, 1995.

[47] M.T. Chan. Automatic lip model extraction for constrained contour-based tracking. In *Proceedings of the 1999 International Conference on Image Processing ICIP' 99, Kobe (JP), 24-28 Oct.*, volume 2, pages 848–851, 1999.

[48] S. Lucey, S. Sridharan, and V. Chandran. Adaptive mouth segmentation using chromatic features. *Pattern Recognition Letters*, 23(2002):1293–1302, 2002.

[49] R.L. Hsu, M. Abdel-Mottaleb, and A.K. Jain. Face detection in color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):696–706, May 2002.

[50] M. Liévin and F Luthon. Nonlinear color space and spatiotemporal mrf for hierarchical segmentation of face features in video. *IEEE Transactions on Image Processing*, 13(1):63–71, january 2004.

[51] S. Stillittano and A. Caplier. Inner lip segmentation by combining active contours and parametric models. In *Proceedings of International Conference on Computer Vision Theory and Applications, VISAPP' 08, Funchal (PT), 22-25 Jan.*, page 5 pp., 2008.

[52] A.E. Salazar, J.E Hernández, and F. Prieto. Automatic quantitative mouth shape analysis. *Lecture Notes in Computer Science*, 4673(2007):416–423, 2007.

[53] S. Werda, W. Mahdi, and A. Ben-Hamadou. Colour and geometric based model for lip localisation: Application for lip-reading system. In *Proceedings of 14th International Conference on Image Analysis and Processing, ICIAP' 07, Modena (IT), 10-13 Sep.*, pages 9–14, 2007.

[54] L. Zhang. Estimation of the mouth features using deformable templates. In *Proceedings of International Conference on Image Processing, ICIP' 97, Washington (US), 26-29 Oct.*, pages Vol. 3: 328–331, 1997.

[55] R.A. Rao and R.M. Mersereau. Lip modeling for visual speech recognition. In *1994 Conference Record of the Twenty-Eighth Asilomar Conference on Signals, Systems and Computers, Pacific Grove (US), 30 Oct.-2 Nov.*, volume 1, pages 587–590, 1994.

[56] P. Delmas, N. Eveno, and M. Liévin. Towards robust lip tracking. In *Proceedings of the 16th International Conference on Pattern Recognition ICPR' 02, Quebec (CA), 11-15 Aug.*, volume 2, pages 528–531, 2002.

[57] P. Delmas. *Extraction des contours des lèvres d'un visage parlant par contours actifs. Application à la communication multimodale*. PhD thesis, Institut National Polytechnique de Grenoble (INPG), France, 2000.

[58] J.E. Hernández, F. Prieto, and T. Redarce. Fast active contours for sampling. In *Proceedings of Electronics, Robotics and Automotive Mechanics Conference*, volume 2, pages 9–13, 2006.

[59] C. Xu and J.L. Prince. Gradient Vector Flow: A new external force for snakes. In *Proceedings of Computer Vision and Pattern Recognition (CVPR '97), San Juan (PR), 17-19 June*, pages 66–71. IEEE, 1997.

[60] C. Xu and J.L. Prince. Snakes, shapes, and gradient vector flow. In *IEEE Transactions on Image Processing*, volume 7, pages 359–369, 1998.

[61] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[62] K.F. Lai, C.M. Ngo, and S. Chan. Tracking of deformable contours by synthesis and match. In *Proceedings of the 13th International Conference on Pattern Recognition ICPR' 96, Vienna (AT)* , volume 1, pages 657–661, 1996.

[63] M. Okubo and T. Watanabe. Lip motion capture and its application to 3-d molding. In *Proceedings of the 3rd. International Conference on Face & Gesture Recognition FG' 98, Nara (JP), 14-16 Apr.*, pages 187–192, 1998.

[64] M.U. Ramos-Sanchez, J. Matas, and J. Kittler. Statistical chromaticity-based lip tracking with b-splines. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP' 97, Munich (DE), Apr.*, pages IV 2973–IV 2976, 1997.

[65] Z. Wu, P.S. Aleksic, and A.K. Katsaggelos. Lip tracking for MPEG-4 facial animation. In *Proceedings of Fourth IEEE International Conference on Multimodal Interfaces, Pittsburgh (US), 14-16 Oct.*, pages 293–298, 2002.

[66] A.S.M. Sohail and P Bhattacharya. Automated lip contour detection using the level set segmentation method. In *Proceedings of 14th International Conference on Image Analysis and Processing, ICIAP' 07, Modena (IT), 10-13 Sep.*, pages 425–430, 2007.

[67] P. Viola and M.J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57:137–154, 2004.

[68] N. Eveno, A. Caplier, and P.Y. Coulon. Jumping snakes and parametric model for lip segmentation. In *Proceedings of International Conference on Image Processing, ICIP' 03, Barcelona (ES), 14-18 Sep.*, pages II 867–870, 2003.

[69] Z. Hammal, N. Eveno, A. Caplier, and P.Y. Coulon. Parametric models for facial features segmentation. *Signal Processing*, 86:399–413, february 2006.

[70] H. Seyedarabi, W.S. Lee, and A. Aghagolzadeh. Automatic lip tracking and action units classification using two-step active contours and probabilistic neural networks. In *Proceedings of the Canadian Conference on Electrical and Computer Engineering, CCECE '06, Ottawa (CA), 7-10 May*, pages 2021–2024, 2006.

[71] B. Beaumesnil and F. Luthon. Real time tracking for 3D realistic lip animation. In *Proceedings of the 18th International Conference on Pattern Recognition, ICPR' 06, Hong Kong (CN), 20-24 Aug.*, volume 1, pages 219–222, 2006.

[72] B. Beaumesnil, F. Luthon, and M. Chaumont. Liptracking and mpeg4 animation with feedback control. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP' 06, Toulouse (FR), 15-19 May*, pages II 677–II 680, 2006.

[73] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[74] J.C. Bezdek. *Pattern Recognition With Fuzzy Objective Function Algorithms*. Plenum Press, 1981.

[75] S.H. Leung, S.L. Wang, and W.H. Lau. Lip Image Segmentation Using Fuzzy Clustering Incorporating an Elliptic Shape Function. *IEEE Transactions on Image Processing*, 13(1):51–62, january 2004.

[76] A.W.C. Liew, S.H. Leung, and W.H. Lau. Lip contour extraction using a deformable model. In *International Conference on Image Processing ICIP' 00, Vancouver (CA), 10-13 Sep.*, volume 2, pages 255–258, 2000.

[77] M. Gordan, C. Kotropoulos, A. Georgakis, and I. Pitas. A new fuzzy c-means based segmentation strategy. applications to lip region identification. In *Proceedings of the 2002 IEEE-TTTC International Conference on Automation, Quality and Testing, Robotics*, page 6, 2002.

[78] S.L. Wang, W.H. Lau, S.H. Leung, and A.W.C. Liew. Lip segmentation with the presence of beards. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP' 04, Montreal (CA), 17-21 May*, volume 3, pages 529–532, 2004.

[79] S.L. Wang, W.H. Lau, and S.H. Leung. Automatic lip contour extraction from color images. *Pattern Recognition*, 37(12):2375–2387, December 2004.

[80] S.L. Wang, W.H. Lau, A.W.C. Liew, and S.H. Leung. Robust lip region segmentation for lip images with complex background. *Pattern Recognition*, 40:3481–3491, 2007.

[81] B. Goswami, W.J. Christmas, and J. Kittler. Statistical estimators for use in automatic lip segmentation. In *Proceedings of the 3rd European Conference on Visual Media Production, CVMP' 06, London (UK), 29-30 Nov.*, pages 79–86, 2006.

[82] I. Mpiperis, S. Malassiotis, and M.G. Strintzis. Expression compensation for face recognition using a polar geodesic representation. In *Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06), Chapel Hill (US), 14-19 June*, pages 224–231, 2006.

[83] W.N. Lie and H.C. Hsieh. Lips detection by morphological image processing. In *Proceedings of the 1998 Fourth International Conference on Signal Processing, ICSP '98, Beijing (CN), 12-16 Oct.*, volume 2, pages 1084–1087, 1998.

[84] Y. Mitsukura, M. Fukumi, and N. Akamatsu. A design of face detection system by using lip detection neuralnetwork and skin distinction neural network. In *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, volume 4, pages 2789–2793, 2000.

[85] H. Takimoto, Y. Mitsukura, M. Fukumi, and N. Akamatsu. Face detection and emotional extraction system using double structure neural network. In *Proceedings of the International Joint Conference on Neural Networks, Montreal (CA), 31 July-4 Aug.*, volume 2, pages 1253–1257, 2003.

[86] Y. Mitsukura, M. Fukumi, and N. Akamatsu. A design of face detection system using evolutionary computation. In *Proceedings of TENCON 2000*, volume 2, pages 398–402, 2000.

[87] T. Cootes and C. Taylor. Active shape models- - smart snakes. In *Proceedings of the British Machine Vision Conference*, pages 266–275, 1992.

[88] T.F. Cootes, D. Cooper, C.J. Taylor, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[89] A. Caplier. Lip detection and tracking. In *Proceedings of 11th International Conference on Image Analysis and Processing ICIAP' 01, Palermo (IT), 26-28 Sep.*, pages 8–13, 2001.

[90] A. Caplier, P. Delmas, and D. Lam. Robust initialisation for lips edges detection. In *Proceedings of 11th Scandinavian Conference on Image Analysis, Kangerlussuaq (GL), 7-11 June*, pages 523–528, 1999.

[91] I. Shdaifat, R. Grigat, and D. Langmann. Active shape lip modeling. In *Proceedings of the 2003 International Conference on Image Processing ICIP' 03, Barcelona (ES), 14-18 Sep.*, volume 3, pages II–875 – II–878, 2003.

[92] P.C. Yuen, J.H. Lai, and Q.Y. Huang. Mouth state estimation in mobile computing environment. In *Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition, FGR 2004, Seoul (KR), 17-19 May*, pages 705–710, 2004.

[93] K.S. Jang, S. Han, I. Lee, and Y.W. Woo. Lip localization based on active shape model and gaussian mixture model. *Lecture Notes in Computer Science*, 4319:1049–1058, December 2006.

[94] M. Jiang, Z.H. Gan, G.M. He, and W.Y. Gao. Combining particle filter and active shape models for lip tracking. In *Proceedings of the Sixth World Congress on Intelligent Control and Automation, WCICA' 06, Dalian (CN), 21-23 June*, volume 2, pages 9897– 9901, 2006.

[95] Y.D. Jian, W.Y. Chang, and C.S. Chen. Attractor-guided particle filtering for lip contour tracking. *Lecture Notes in Computer Science*, 3851(2006):653–663, 2006.

[96] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. In *Proceedings of the European Conference on Computer Vision*, volume 2, pages 484–498, 1998.

[97] I. Matthews, T.F. Cootes, J.A. Bangham, S. Cox, and R. Harvey. Extraction of visual features for lipreading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):198–213, february 2002.

[98] P. Gacon, P.-Y. Coulon, and G. Bailly. Shape and sampled-appearance model for mouth components segmentation. In *Proceedings of 5th International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS' 04, Lisbon (PT), 21-23 Apr.*, page 4pp, 2004.

[99] P. Gacon, P.Y. Coulon, and G. Bailly. Non-linear active model for mouth inner and outer contours detection. In *Proceedings of Eur. Signal Processing Conference, EUSIPCO' 05, Antalya (TR), 4-8 Sep.*, page 5 pp, 2005.

[100] Z. Li and H. Ai. Texture-constrained shape prediction for mouth contour extraction and its state estimation. In *18th International Conference on Pattern Recognition ICPR' 06, Hong Kong (CN), 20-24 Aug.*, volume 2, pages 88–91, 2006.

[101] R.E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

[102] A. Turkmani and A. Hilton. Appearance-based inner-lip detection. In *Proceedings of the 3rd European Conference on Visual Media Production, CVMP' 06, London (UK), 29-30 Nov.*, pages 176–176, 2006.

[103] M.K. Moghaddam and R. Safabakhsh. TASOM-based lip tracking using the color and geometry of the face. In *Proceedings of the Fourth International Conference onMachine Learning and Applications, ICMLA' 05, Los Angeles (US), 15-17 Dec.*, page (6pp), 2005.

[104] A. Khan, W. Christmas, and J. Kittler. Lip contour segmentation using kernel methods and level sets. *Lecture Notes in Computer Science*, 4842(2007):86–95, 2007.

[105] J.S. Chang, E.Y. Kim, and S.H. Park. Lip contour extraction using level set curve evolution with shape constraint. *Lecture Notes in Computer Science*, 4552:583–588, 2007.

[106] L. Xie, X.L. Cai, Z.H. Fu, R.C. Zhao, and D.M. Jiang. A robust hierarchical lip tracking approach for lipreading and audio visual speech recognition. In *Proceedings of 2004 International Conference on Machine Learning and Cybernetics*, volume 6, pages 3620–3624, 2004.

[107] S. Lucey, S. Sridharan, and V. Chandran. Initialised eigenlip estimator for fast lip tracking using linear regression. In *Proceedings of the 15th International Conference on Pattern Recognition ICPR' 00, Barcelona (ES), 3-8 Sep.*, volume 3, pages 178–181, 2000.

[108] A.R. Mirhosseini, K.M. Lam, and H. Yan. An adaptive deformable template for mouth boundary modeling. In *International Conference on Image Analysis and Processing ICIAP' 97, Florence (IT), 17-19 Sep.*, volume 1310, pages 559–566, 1997.

[109] A.R. Mirhosseini, C. Chen, K.M. Lam, and H. Yan. A hierarchical and adaptive deformable model for mouth boundary detection. In *Proceedings of the 1997 International Conference on Image Processing*, volume 2, pages 756–759, 1997.

[110] A.W.C. Liew, S.H. Leung, and W.H Lau. Segmentation of color lip images by spatial fuzzy clustering. *IEEE Transactions on Fuzzy Systems*, 11:542–549, august 2003.

[111] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre. XM2VTSDB: the extended M2VTS database. In *Proceedings of the Second International Conference on Audio- and Video-based Biometric Person Authentication, AVBPA' 99*, pages 72–77, 1999.

[112] A.M. Martínez and R. Benavente. The AR database. Technical report, CVC, june 1998.

[113] M. Liévin and F. Luthon. Unsupervised lip segmentation under natural conditions. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '99, Phoenix (US), 15-19 Mar.*, volume 6, pages 3065–3068, 1999.

[114] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[115] P.J. Phillips, H. Wechsler, J. Huang, and P. Rauss. The FERET database and evaluation procedure for face recognition algorithms. *Image and Vision Computing*, 16(5):295–306, 1998.

[116] P.J. Phillips, H. Moon, S.A. Rizvi, and P.J. Rauss. The FERET Evaluation Methodology for Face Recognition Algorithms. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000.

[117] A. Ceballos, J.B. Gómez, F. Prieto, and T. Redarce. Robot command interface using an audio-visual speech recognition system. In *Proc. of Congreso Iberoamericano de Reconocimiento de Patrones CIARP' 09, Guadalajara (MX), 15-18 Nov.*, pages 869–876, 2009.

[118] C.M. Bishop. *Pattern Recognition and Machine Learning*, chapter 4: Linear Models for Classification. Springer, 2007.

[119] J. Van de Weijer, T. Gevers, and A. Gijsenij. Edge-based color constancy. *IEEE Trans. on Image Processing*, 16(9):2207–2214, 2007.

[120] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley, 2001.

[121] J.B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, 1967.

[122] J.R. Jensen. *Introductory Digital Image Processing - A Remote Sensing Perspective*. Prentice Hall, 1996.

[123] C.M. Bishop. *Pattern Recognition and Machine Learning*, chapter 9: Mixture models and EM. Springer, 2007.

[124] M. Riedmiller and H. Braun. A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. In *IEEE International Conference on Neural Networks*, pages 586–591, 1993.

[125] M.A. Arbib. *Brains, Machines and Mathematics*. Springer-Verlag, 2nd. edition, 1987.

[126] S. Haykin. *Neural Networks and Learning Machines*. Pearson, 3rd edition, 2009.

[127] A.G. Kraut and D.W. Smothergill. A two-factor theory of stimulus-repetition effects. *Journal of Experimental Psychology: Human Perception and Performance*, 4(1):191–197, 1978.

[128] L. Shapiro and G.C. Stockman. *Computer Vision*. Prentice-Hall, 2001.

[129] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[130] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(8):1026–1038, 2002.

[131] S. Alizadeh, R. Boostani, and V. Adapour. Lip feature extraction and reduction for HMM-based visual speech recognition systems. In *Proc. of International Conference on Signal Processing ICSP' 08, Beijing (CN), 26-29 Oct.*, pages 561–564, 2008.

[132] J.B. Gómez, F. Prieto, and T. Redarce. Automatic Outer Lip Contour Extraction in Facial Images. In *Proceedings of the International Conference on Systems, Signal and Image Processing IWSSIP'10, Rio de Janeiro (BR), 17-19 June*, pages 336–339, 2010.

[133] WEB. www.intuitivesurgical.com/index.aspx.

[134] C.O. Nathan, V. Chakradeo, K. Malhotra, H. D'Agostino, and R. Patwardhan. The Voice-Controlled Robotic Assist Scope Holder AESOP for the Endoscopic Approach to the Sella. *Skull Base*, 16(3):123—-131, 2006.

[135] J.B. Gómez, F. Prieto, and T. Redarce. *Innovative Algorithms and Techniques in Automation, Industrial Electronics and Telecommunications*, chapter Lips Movement Segmentation and Features Extraction in Real Time, pages 205–210. Springer Netherlands, 2006.

[136] M.E. Allaf, S.V. Jackman, P.G. Schulam, J.A. Cadeddu, B.R. Lee, R.G. Moore, and L.R. Kavoussi. Voice vs foot pedal interfaces for control of the AESOP robot. *Surgical Endoscopy*, 12(12):1415–1418, 1998.