Edge detection for lips area using RGB color space

Amer Abdulmejeeed Abdulrahman
University of Baghdad – Collage of Education for Women
Computer Department
amer6567@yahoo.com

Abstract
In many cases human identification biometrics systems are motivated by real-life criminal and forensic applications. Some methods, such as fingerprinting and face recognition, proved to be very efficient in computer vision based on human recognition systems. The paper focus on developing computer recognition system that would be used to identify humans on the basis of their lips by using the futures of RGB color space for lips detection.

Keywords : edge detection, segmentation, thresholding, biometrics, recognition

1- Introduction
Recently, there is an increasing requirement for a system to track and locate human lip. Human lip has much more information than any other face features, so the lip information could be used in image coding. To improve the performance of speech recognition, the lip information is used together with the acoustic signal. The information also be applied to the graphic animation systems, which need it for generating the lip shape of the speaker.

Accurately and robustly tracking lip motion in image sequences is especially difficult because lips are highly deformable and they vary in shape and color. Gradient based techniques for edge detection of lip often fail due to the poor contrast between lip and surrounding skin region. For methods using color information to build a parametric deformable model for the lip contour, these require optimization technique to refine estimates of contour model to the human lip. Many papers have described the applications of active contour model (snake) for lip boundary detection. The snake methods are able to resolve fine contour details but shape constraints are difficult to incorporate. Furthermore, the snake methods often converge to the wrong result when the lip edges are not distinct or when the lip color is very close to the face skin. Many methods have localized only the outer lip contour when the mouth is closed because the presence of tongue and teeth could obscure the inner contour when the mouth is open [1,2,3].

The performance of Automatic Speech Recognition (ASR) system decreases drastically due to the presence of the noise. It has been shown
that incorporating visual information extracted from lip images analysis can greatly enhance the accuracy of speech recognition systems solely based on acoustic signal [4]. Lip region segmentation becomes very important since it provides lip pixel information for the subsequent lip modeling process. Many researchers have proposed various algorithms for robust lip region segmentation. Some methods are based on color space analysis, i.e., segmenting lip region directly from the color space or transforming the color representation to other color space to enlarge the color difference between the lip and the background [5].

2- Proposed method

The overview of the proposed method for edge detection from lips in face image is shown in figure 1.

![Flow chart for proposed method](image)

2.1 Color space.

The original images in most of vision systems are in the RGB color format. We use this color space to avoid some additional color transformation. RGB color space involves three components R-component (Red), G-component (Green) and B-component (Blue). In RGB space, skin and lips pixels have quite different components. For both, red is prevalent. Moreover there is more green than blue in the skin color mixture and for lips these two components are almost the equal. The difference between Red and Green is greater for lip than for skin. With this knowledge and after comparison three components and those histograms of lip regions, we knew that the value of G-component for lip region is less than R-component and so B-component [6].
We found a simple relationship between three components of RGB color space to separate skin region from lip region. When we subtract R-component and B-component from G-component and then add the results as in equation 1, the resultant \( I \) value has a good property to distinguish lips from skin. In fact with this work we exclude green component to separate lip from skin. The equation for this approach is as follow:

\[
I = (R - G) + (B - G) \quad \ldots \quad (1)
\]

Where \( R \) is R-component, \( G \) is G-component and \( B \) is B-component of RGB image[7]. Figure (2) shows the result to apply equation 1. When applied traditional threshold to the result of equation 1, the good results are obtained.
Figure (2) a. Original image   b. image after apply eq.1

2.2 Thresholding:

In many vision applications, it is useful to be able to separate out the regions of the image corresponding to objects in which we are interested, from the regions of the image that correspond to background. Thresholding often provides an easy and convenient way to perform this segmentation on the basis of the different intensities or colors in the foreground and background regions of an image. In addition, it is often useful to be able to see what areas of an image consist of pixels whose values lie within a specified range, or band of intensities (or colors). Thresholding can be used for this as well.

The input to a thresholding operation is a color image. In the simplest implementation, the output is a binary representing the segmentation. Black pixels correspond to background and white pixels correspond to foreground (or vice versa). In simple implementations, the segmentation is determined by a single parameter known as the intensity threshold. In a single pass, each pixel in the image is compared with this threshold. If the pixel's intensity is higher than the threshold, the pixel is set to, say, white in the output with value of 1. If it is less than the threshold, it is set to black with value of 0.

In more sophisticated implementations, multiple thresholds can be specified, so that a band of intensity values can be set to white while everything else is set to black. For color or multi-spectral images, it may be possible to set different thresholds for each color channel, and so select just those pixels within a specified cuboids in RGB space. Another common variant is to set to black all those pixels corresponding to background, but leave foreground pixels at their original color/intensity (as opposed to forcing them to white), so that that information is not lost [8]. Figure (3) shows the result after applying threshold for the images in fig 2.b.
2.3 Edge detection

To extract the upper and lower edges of lips which are mostly diagonal edges. We apply edge detection technique using Sobol edge detector as it smoothes the image it a good edge detector for diagonal edges.

In this part, the sobel edge detection algorithm is described, while explaining the VHDL code used for its implementation.

In general, an algorithm of edge detection finds the sharp intensity variation of an image in this way it obtains the edges of the objects contained on the image. There are various methods to detect the edges which use discrete gradients, laplacians, etc. The most common methods used in the detection of edges are Roberts, Sobel, Prewitt, Laplacian, Canny, etc. Their operators are masks of 3x3 windows (2x2 windows in the Roberts algorithm) which are convolved with the incoming image to assign each pixel a value of 0 or 255. To obtain better results each method applies between two and four masks to find edges in the image.

This tutorial explains the implementation of one of the most commonly used methods for edge detection called Sobel edge detection algorithm. This algorithm uses 4 operators (also called masks or kernels) of 3x3 windows which measure the intensity variation of the image when they are convolved in 4 directions: horizontal, vertical, right diagonal and left diagonal.

Now we describe the Sobel Edge Detection algorithm and its VHDL implementation.
Since the masks are of dimension 3x3, it is necessary to select a 3x3 window of the image to convolve it with each mask. To explain this process we select the first 3x3 window of the image as shown in the shadowed part of the figure (4), we can observe in the equations how this masks are applied to the original image.

\[
\begin{bmatrix}
1 & 2 & 1 \\
0 & 0 & 0 \\
-1 & -2 & 0
\end{bmatrix}
\quad\quad
\begin{bmatrix}
-1 & 0 & 1 \\
-2 & 0 & 2 \\
-1 & 0 & 1
\end{bmatrix}
\]

\[
\begin{bmatrix}
0 & 1 & 2 \\
-1 & 0 & 1 \\
-2 & -1 & 0
\end{bmatrix}
\quad\quad
\begin{bmatrix}
2 & 1 & 0 \\
1 & 0 & -1 \\
0 & -1 & -2
\end{bmatrix}
\]

Figure (4) the four masks of sobol operators

To convolve the image with these masks, we have to solve some equations, which are described in the following paragraphs. Note that these equations are implemented in the package of image processing as functions which are specified for each equation. All the equations that are shown below have the same form. The function that implemented these operations is called weight and the code is described as follows.

To obtain the value of the intensity variation, the implementation in VHDL is shown for each case. Note that these functions call the weight function.

\[
\begin{align*}
Eh &= (A(1,1) + 2A(1,2) - A(1,3)) - (A(3,3) + 2A(3,2) + A(3,1)) \quad(2) \\
Ev &= (A(1,3) + 2A(2,3) + A(3,3)) - (A(1,1) + 2A(2,1) + A(3,1)) \quad(3) \\
Edl &= (A(1,2) + 2A(1,3) + A(2,3)) - (A(2,1) + 2A(3,1) + A(3,2)) \quad(4) \\
Edr &= (A(1,2) + 2A(1,1) + A(2,1)) - (A(2,3) + 2A(3,3) + A(3,2)) \quad(5)
\end{align*}
\]

After finding the values in each direction with each function, the gradient magnitude and the direction of the intensity variation must be computed. The formula that is used for this purpose is the following.

\[
Mag = \max(\left|Eh\right|, \left|Ev\right|, \left|Edl\right|, \left|Edr\right|) + 1/8[\left|E\perp\right|] \quad(6)
\]

In this equation, it can observed that we have to find the maximum value of intensity variation and add the value of $E\perp/8$, where $E\perp$ is the intensity variation in the direction that is perpendicular to the direction of the maximum intensity variation.

To find the maximum value of intensity variation, the function compare must be used. This function receives the values of the intensity variation of each direction that were computed in the previous part and returns the maximum absolute value, the absolute value of $E\perp$ and a binary code indicating the direction of the edge of the figure. The codes for each edge direction are shown in table 1.
Table 1 Codes for edge directions

<table>
<thead>
<tr>
<th>DIRECTION</th>
<th>CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive Horizontal</td>
<td>000</td>
</tr>
<tr>
<td>Negative Horizontal</td>
<td>001</td>
</tr>
<tr>
<td>Positive Vertical</td>
<td>010</td>
</tr>
<tr>
<td>Negative Vertical</td>
<td>011</td>
</tr>
<tr>
<td>Positive Right Diagonal</td>
<td>100</td>
</tr>
<tr>
<td>Negative Right Diagonal</td>
<td>101</td>
</tr>
<tr>
<td>Positive Left Diagonal</td>
<td>110</td>
</tr>
<tr>
<td>Negative Left Diagonal</td>
<td>111</td>
</tr>
</tbody>
</table>

When the outputs of the function *compare* are obtained, the magnitude described in the previous equation has to be found. The function used to obtain this value is the function *magnitude*, which is described below.

This process is developed for all the pixels of the image. It can be observed in this part of the code that the value of the constants foreground and background are assigned to the variable *temp* after the comparison. These constants are initialized in the package of image processing. All the steps that are needed for doing the operations that require the Sobel edge detection algorithm have been explained in the previous part. However, this procedure is done for just one 3x3 window of the image. For that reason, some code is required to read the entire image in 3x3 windows and store the data obtained as a result of applying the Sobel operators to each of these windows. The code that reads the image and splits it in several 3x3 windows is described in the edge detector code as follows [9].

Figure (5) shows the result after applying the sobol edge detection filter for the images in fig(3).
3. Experiments and Results.

We applied our approach to sample images obtained in a laboratory environment and from the FEI Face Database [10]. The database contains front face color images under different lighting. In our experiments, a common base shape model. Figure (6) shows some of the intermediate and final results obtained using the proposed algorithm.
We tested the proposed algorithm with different images and results showed good segmentation for different speakers with different illumination.

Fig (6). a. original image b. color space  c. thresholding  d. Sobol edge detection  f. result for apply algorithm.
From the experiment using 100 face images the detection rate of 91% is achieved.

By compared to previous methods, we obtain a simple feature for lip region extraction using RGB. Then we apply edge detection technique using Sobol edge detector as it smoothes the image it a good edge detector for diagonal edges. The lip detection method based on RGB chromaticity diagram is presented. The method is used to separate lip color from other colors (face skin color) by a simple effective color segmentation method. In future we will extend to utilize the lip detection technique for extracting the visual information used in the Automatic Visual Speech reconvention System (AVSR).

References