

Face Fusion: An Automatic Method For Virtual Plastic Surgery

Seyed Alireza Rabi
Parham Aarabi

The Edward S. Rogers Sr. Department of Electrical and Computer Engineering
University of Toronto
Toronto, ON, Canada
rabi@ecf.utoronto.ca
parham@ecf.utoronto.ca

Abstract - *This work describes a system that replaces an individual's facial features with corresponding features of another individual -possibly of different skin color- and fuses the replaced features with the original face, such that the resulting face looks natural. The final face resulting from the fusion of the original face with exogenous features, lacks the characteristic discontinuities that would have been expected if only a replacement operation was performed. The proposed system could be used to simulate and predict the outcome of aesthetic maxillofacial plastic surgeries.*

To achieve its task, the system uses five modules: face detection, feature detection, replacement, shifting and blending. While these modules are designed to address the problem of face fusion, some of the novel algorithms and techniques introduced in this work could be useful in other image processing and fusion applications..

Keywords: Virtual Surgeon, face detection, feature detection, edge detection, blending, shifting, RGB and HSV (a.k.a. HSL) color models.

1 Introduction

Motivated by reports of increased self-confidence [1], and driven by shifting cultural values, number of aesthetic maxillofacial plastic surgeries have increased dramatically in the past few decades. More recently, advances in computer modeling have allowed pre-surgical observation and simulation of the effects of such operations, thus reducing the risk of unwanted results.

Most of these systems focus on overall three dimensional simulation, thus ignoring finer details that nevertheless contribute immensely to the success of plastic surgeries [2], [3]. For example, prior work [4] showed that symmetries and proportions of a frontal face play a fundamental role in our concept of beauty. As a result, in this work we have tried to focus on fine-tuning the frontal, two dimensional image that results from replacement of patients facial features with those of the model.

In addition to the classical problems of face finding and feature detection, our system addresses and solves the problem of fusing a face with facial features belonging to another individual -possible of different color. This is required in order to achieve a natural looking result.

The problem of face detection has been extensively studied in the past and numerous solutions with varying degrees of success have been proposed -see [5] for a thorough review of the most successful algorithms. Recent work illustrates increased interest in algorithms -such as Neural Networks- that require extensive prior training. For example, in [6] and [7] very successful methods of detecting faces were developed, which can detect faces regardless of pose and orientation. However, due to our exclusive interest in frontal views of the face, we have chosen a simpler algorithm that uses a stored, generic face template, which was first proposed in [8]. Whereas the algorithm in [8] does not use the color of skin, an alternative solution proposed in [9], utilizes the color of skin in detecting the face. In our face detection algorithm we use aspects of both solutions.

Blending and replacing facial features requires a two level processing. On one hand, it is required that the shape of the original shape be changed to that of the desired object, and on the other hand it is desired that the color of the replaced feature be same as the original feature. The final requirement is crucial if we are to achieve a seamless transition from the replaced feature to the underlying face. To solve this problem we use a probabilistic model which takes advantage of pre-assumed information regarding the location of the object, as well as information about the color of the face. If a pixel is at the right place to be a feature and has a color different from that of the skin, then most probably it belongs to a facial feature, such as the eyes, nose or lips.

2 Overview

Figure 1 shows the block diagram representation of the virtual surgeon system. Aside from the two faces, the user has to input -by clicking- the approximate

coordinate of the tip of the nose belonging to the face of interest. As will be discussed in subsequent sections, this additional information will be used to find the face in an image, as well as giving us an idea of the skin color for each face.

The first step is to detect faces in the input images. The face detection module achieves this goal by using a stored template of a generic face -figure 3(a). Once the face is detected, the feature detection module finds three rectangular regions containing the left eye, the right eye and the lips. This information is then fed into the replacement module which replaces the eyes and lips of the original face with those of the model face. Together, the shifting and blending modules achieve smooth blending of the replaced features in the original face, such that the result will look natural. Figure 2 shows the outcome of these operations performed on two sample faces.

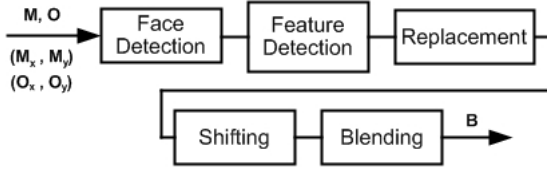


Figure 1: Block diagram representation of the system. Aside from the two faces, the user has to input -by clicking- the approximate coordinate of the tip of the nose belonging to the face of interest. B is the face resulting from replacement of the original features of the face with the corresponding features of the model face. M and O are two pictures containing the faces of interest and (M_x, M_y) and (O_x, O_y) are the coordinates of the tip of the nose in the two faces.

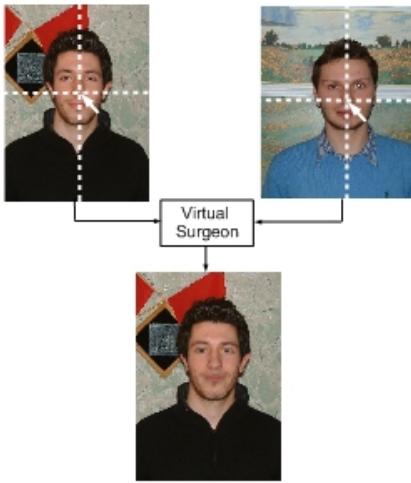


Figure 2: The eyes and lips of the top left image is replaced with those of the top right image. The resulting picture is smoothed for the most natural look.

In the subsequent sections, we will discuss each of the system modules in detail.

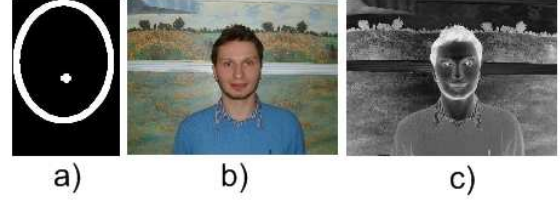


Figure 3: (a)Face template and (b)an image before and after the transformation of equation 1

3 Face Detection

Our face detection algorithm requires the user to input -by clicking- the approximate coordinate of the tip of the nose belonging to the face of interest. The program will then superimpose on the picture a scaled version of a generic face template -shown in figure 3(a). The template is enlarged in small intervals until it falls outside of the face boundary.

The first step is to transform the image according to the following formula:

$$\mathbf{F} : M^{m \times n \times 3} \mapsto T^{m \times n \times 1} \quad (1)$$

such that,

$$T(i, j) = \frac{1}{3} \sum_{k=1}^3 \|M(i, j, k) - \bar{m}_k\| \quad (2)$$

Where $M(i, j, k)$ is the Red, Green or Blue -for $k=1, 2$ and 3 , respectively- value of a pixel in the i^{th} row and j^{th} column of the original face. And T is the resulting image and \bar{m}_1, \bar{m}_2 and \bar{m}_3 are the mean RGB values for a box of size 10 pixel \times 10 pixel centered around the tip of the nose.

In the resulting image, T , those pixels with colors different from the skin will appear lighter, whereas those whose color resemble that of the skin appear darker. Figures 3(b) and 3(c) show an image, before and after the transformation.

As the stepwise enlargement of the generic face template superimposed on the transformed image, T , continues, the sum of the intensities of pixels in T that lie on the boundary of the face template is calculated. Let us call this value I , where I is a function of the scaling factor used to enlarge the face template.

Storing the values of I , normalized by the size of the template, enables us to detect the boundaries of the face. As long as the template is small enough to lie within the outline of the face, I values remain small, but as soon as the right scaling factor is reached, for which the template lies completely outside the face, I values increase substantially. This sudden increase in values of I , the location of which can be found mathematically by finding where the first derivative of the graph of I vs. scale factor is maximum, could be used to find the location of the face. Since the location of facial features -such as eyes and lips- for all human faces obey certain rules [11], once we have found the general boundary of the face we can envisage an eye box and a lip box containing these features. Figure 4 summarizes these steps.

4 Feature Detection

As discussed in the previous section, using the face finding algorithm and utilizing the homology among human faces, it is possible to find rectangular regions -such as the ones shown in figure 4- that contain the eyes and lips. However, keeping in mind our ultimate goal of smooth blending of features belonging to two distinct faces, regions of the size shown in figure 4 are too large. We have to pinpoint the exact position of the eyes and the lips with more accuracy. This is achieved using the feature detection algorithm discussed in this section.

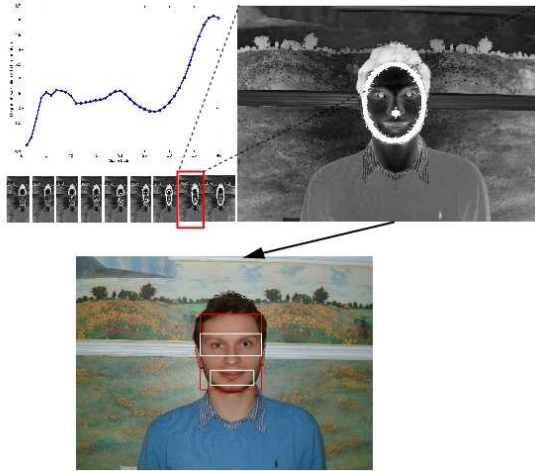


Figure 4: Once we have found the general location of the face, utilizing the homology among all human faces, it is possible to construct rectangular regions containing the eyes and lips.

Let us consider the eyes first; from the previous step, we have an eye box of size $L_y \times L_x$ which contains both eyes -figure 5(a). Let us agree that two smaller boxes of size $l_y \times l_x$, centered around the pupil of each eye capture the salient features of an eye, without including unnecessary regions. Our task then, is to find two boxes of size $l_y \times l_x$ within the original eye box, E.

Figure 5(b) is obtained by performing the Prewitt edge detection algorithm [12] on E. Notice that as we approach the eyes more edges become visible. These edges include the outline of the eyes themselves, the eyelids and the eyebrows.

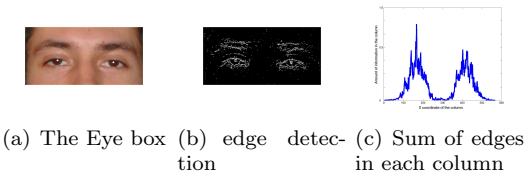


Figure 5: The result of performing the Prewitt edge detection algorithm on the eye box.

Next, for each column along the length of E, we sum

the edges in that column:

$$H(i) = \sum_{j=1}^{L_y} E_d(i, j) \quad (3)$$

where $1 \leq i \leq L_x$, and E_d is the image representing the edges in the eye box of size $L_y \times L_x$. Figure 5(c) shows the result of this operation. Notice that peaks are observed around the location of the eyes reflecting our previous observation that the majority of edges occur around the eyes. The idea of using projecting edges in horizontal and vertical axis as an indication of information content was first introduced in [10]; however, whereas they used this technique to detect the location of faces, we have applied it to pinpoint the exact location of the eye.

Finding two intervals of equal length, l_x -one on each side of the face- for which H is maximized yield the horizontal edges of the desired boxes.

$$k = \arg \max_{\kappa} \sum_{i=\kappa}^{\kappa+l_x} H(i) \quad (4)$$

where $k \leq i \leq k + l_x$ is the horizontal edge of the left eye box or the right eye box, depending on whether the equation 4 was performed on the left or right side of the face.

Performing the exact same algorithm in the vertical direction, but this time summing the edges in each row instead of each column, yields the vertical edges of the eye boxes. Figure 6 shows these steps.

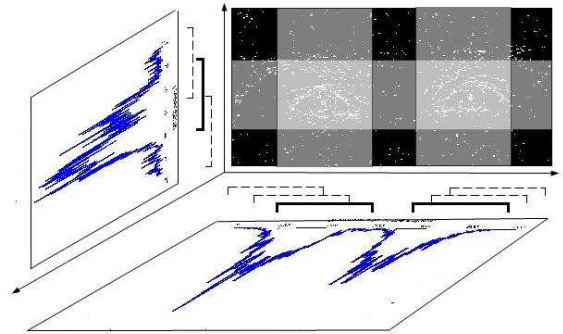


Figure 6: Feature detection algorithm: The edges in the face box are projected in the horizontal and vertical directions, as in equation 3. The horizontal and vertical interval for which the projected areas are maximum form the horizontal and vertical edges of the eye box, respectively.

Since the original eye box, E, could contain regions outside of the face, some spikes may be observed on either ends of the graph of H, corresponding to regions outside of the face. This could cause some problems for our feature detection algorithm. Multiplying H by a double gaussian with means around the expected location of the eyes and standard deviations of l_x will eliminate this problem.

Using the same algorithm, we can detect the exact location of the lips.

5 Replacement

Once we have found the two eye boxes containing the left and right eyes and a lip box containing the lips, we can proceed with the ultimate goal of replacing the current features of the original face with the desired features of the model face. The first step is to resize the eye and the lip boxes belonging to the model face such that they fit their appropriate places in the original face, and then superimposing them on the original face. Figure 7 shows the result of this operation. It is evident that this simple replacement will not result in a natural looking face, due to the possible mismatch in the skin colors. The subsequent sections address the problem of “blending” the replaced features such that the resulting face will look natural.

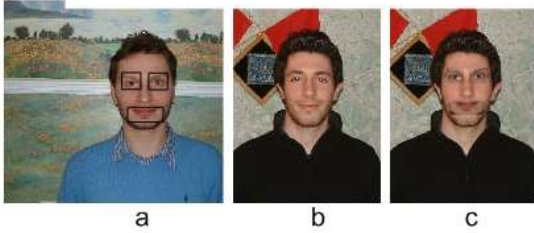


Figure 7: (a) and (b) are the model face and the original face, respectively. Simple replacement of features will not result in a natural-looking result, as seen in (c).

5.1 Blending

In this section we will focus on the eyes. However, everything that we state could also be applied to the lips.

We define $P(x_i, x_j)$ to be the probability that pixel (x_i, x_j) constitutes the eye in the original face. Similarly, $P(x'_i, x'_j)$ is the probability that pixel (x'_i, x'_j) is part of the desired eye. For the moment, let us assume that we know the probability distribution functions (pdf) $P(x_i, x_j)$ and $P(x'_i, x'_j)$. Consider the new eye box, N , constructed from the original eye box, O , and the desired eye box, D , using the following formula:

$$\begin{aligned} N(i, j, k) &= P(x'_i, x'_j)D(i, j, k) \\ &+ (1 - P(x'_i, x'_j))(1 - P(x_i, x_j))O(i, j, k) \\ &= (1 - P(x'_i, x'_j))P(x_i, x_j)M_k \end{aligned} \quad (5)$$

for $k = 1, 2, 3$; where N , D and O are of equal size, and M_1 , M_2 and M_3 are the mean RGB values for the skin color of the original face.

We argue that equation 5 gives us a way of replacing the original eye with the desired eye, while maintaining a smooth and natural looking face. To see this, let us consider a pixel in the i^{th} row and j^{th} column of the newly constructed eye box, N . the corresponding pixel in D , $D(i, j)$, is either a part of the desired eye, or it is not. If it is, $N(i, j)$ must equal $D(i, j)$. This is achieved

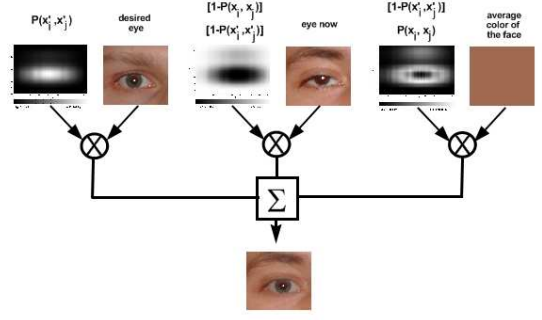


Figure 8: The result of replacing the left eye in the original face with the left eye of the model face. Together, the shifting and blending modules achieve a smooth, natural-looking result.

in equation 5, since in this case $P(x_i, x_j)$ is close to one, making the result to be dominated by the first term.

On the other hand, if $D(i, j)$ is not part of the desired eye, we have two cases: either $O(i, j)$ is part of the original eye or it is not. If $O(i, j)$ is not part of the original eye, $N(i, j)$ must equal $O(i, j)$; effectively this ensures that the skin color and texture of the original face is preserved, which would achieve a smooth and natural blending. On the other hand, in the second case where neither $O(i, j)$ nor $D(i, j)$ are eye pixels, we need to $N(i, j)$ with the “skin” of the original face. Close examination of equation 5 reveals that in all of the possible cases, the correct result is obtained.

Thus, our task is to obtain the probability distribution functions (pdf) $P(x_i, x_j)$ and $P(x'_i, x'_j)$. Define $P(x_j)$ to be the probability that the pixels in the i^{th} row of the original eye box, O , are eye pixels. Similarly, let $P(x_j)$ to be the probability that the pixels in the j^{th} column of the original eye box, O , are eye pixels. $P(x'_i)$ and $P(x'_j)$ are the corresponding pdf’s for the desired eye box, D . We can write $P(x_i, x_j)$ in terms of $P(x_i)$ and $P(x_j)$ as follows:

$$P(x_i, x_j) = P(x_i \cap x_j) \quad (6)$$

$$= P(x_i)P(x_j) \quad (7)$$

Where we have assumed independence of random variables x_i and x_j . Similarly, for $P(x'_i, x'_j)$ we can write:

$$P(x'_i, x'_j) = P(x'_i \cap x'_j) \quad (8)$$

$$= P(x'_i)P(x'_j) \quad (9)$$

In section 4 we devised a method for calculating the pdf’s $P(x_i)$, $P(x_j)$, $P(x'_i)$, and $P(x'_j)$. For example, we can calculate $P(x_j)$ by summing the edges in the j^{th} column of the original face’s eye box.

The pdf’s derived from equations 6 and 8, and the sum of edges in rows and columns as $P(x_i)$, $P(x_j)$, $P(x'_i)$, and $P(x'_j)$, suffer from one major shortcoming: They consist of discontinuous sharp peaks, and deep valleys between the peaks rather than continuous functions of position, which is what we would ideally need

if we are to obtain smooth fusion of faces. We have addressed this problem by convolving the pdf's of equations 6 and 8 by the sinc function. Note that we have not investigated the possibility of using other functions for this smoothing task. Potentially, any other differentiable, slowly changing function, such as a gaussian, could be used. Figure 9 shows the result of this operation.

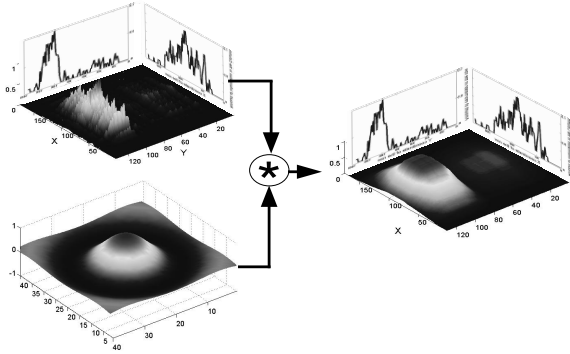


Figure 9: The top left figure shows $P(x_i, x_j)$ calculated using equation 6. Convoluting this with the sinc function -bottom left figure- leads to a continuous pdf.

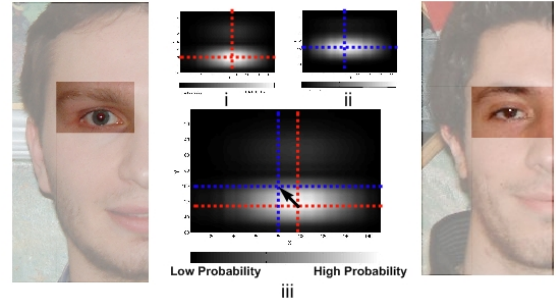
Up until this point, we have not mentioned anything about the relative position of the eyes in the original eye box, O , and the desired eye box, D . For instance, it is possible for the eye to be located in the upper right corner of O and the desired eye to be in the lower left corner of D . In this case, equation 5 will construct a new eye box, N , in which the upper right corner is covered by the “skin” of the original face, and the lower left corner is occupied by the desired eye. It can be predicted that in this case, the constructed eye box, N , will not look as natural as desired. This is because the average skin color M_k in equation 5 is an average for the entire face, which might not be a good representative of the skin color in the region of interest, i.e. around the eyes. This problem is resolved if we shift the desired eye box, D , such that the desired eye in D is located at around the same relative position that the original eye is located in O . If such shifting is performed prior to using equation 5, the discussed problem is effectively eliminated, since there are a few pixels that need to be covered by “skin”. This is achieved using the shifting module in figure 1.

6 Shifting

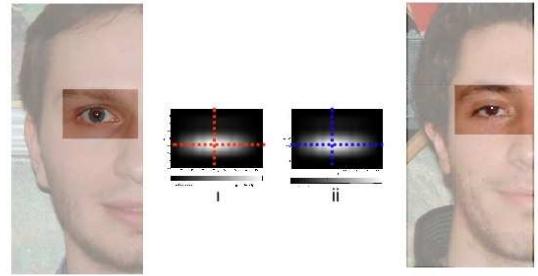
We have used the location of the center of mass of $P(x_i, x_j)$ and $P(x'_i, x'_j)$ to determine where the eyes are located within the original eye box and the desired eye box, respectively. The x and y coordinates of the center of mass, CM_x and CM_y for a given pdf $P(x_i, x_j)$ could be found using the following formulae:

$$CM_x = \sum_{j=1}^n P(x_i, x_j) \quad (10)$$

$$CM_y = \sum_{i=1}^m P(x_i, x_j) \quad (11)$$



(a) i and ii are the eye boxes for the faces shown center of mass in each figure. In iii, i and ii are superimposed to find the translation vector under which the eyes will have the same location.



(b) i and ii are the eye boxes for the faces shown on the left and right, respectively, after the eye box in figure 10(a) i was shifted.

Figure 10: Shifting the eye box, so that the eyes are in the same general location.

In figure 10(a) i and ii show the center mass of the original eye box and the desired eye box. In iii we have superimposed the two images and indicated the translation under which the eye in the desired eye box will be transferred to a location same as the location of the eye in the original eye box. Alternatively, instead of moving the eye using this translation vector, we can move the eye box using the inverse of the translation.

Figure 10(b) shows the resulting situation if the eye box is translated as discussed above; both the desired eye and the original eye will be located at approximately the same relative position within their corresponding eye boxes.

The same shifting operation is performed on the lip boxes. The face resulting from the replacement of features of the face in figure 7(b) with the corresponding features of the face in figure 7(a), followed by shifting and blending, is shown in 11. Comparing this picture with the one shown in 7(c), shows the importance of the shifting and blending modules in obtaining a smooth and natural-looking result.

7 Conclusion

In this work we described details of a system that achieves fusion of a face with facial features belonging to another individual -possibly of different skin color. The resulting face looks natural, without discontinuities in color or texture. This system could be used as a tool to predict the outcome of aesthetic maxillofacial

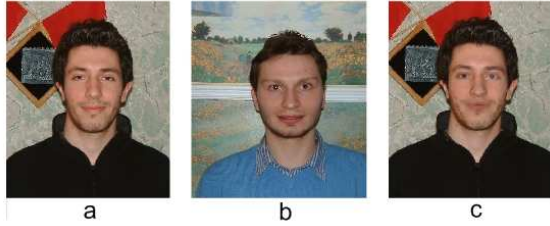


Figure 11: The face resulting from the replacement of features of a) the original face with the corresponding features of b) the model face, followed by shifting and blending. Comparison with figure 7(c) shows the improvement gained as a result of performing the shifting and blending algorithms described above.

surgeries.

Our algorithm uses five modules: face detection, feature detection, replacement, shifting and blending. To detect faces, we combine aspects of two well-known face detection algorithms by utilizing the color information as well as using a generic face template. In the facial feature module we have used vertical and horizontal projection of edges in a face to find the exact location of features -such as eyes and lips. This information was also used to construct probability distribution functions, used to fuse the replaced features within the original face.

While this work targets the problem of facial feature fusion, some of the techniques outlined here could be used in other areas of image processing and fusion. However, there are certain limitations in our system, which need to be addressed.

References

- [1] Goin MK, Rees TD, A prospective study of patients' psychological reactions to rhinoplasty, *Annals of Plast Surgery*, 27(3):210-5, Sep 1991.
- [2] Luboz V, Chabanas M, Swider P, Payan Y. Orbital and maxillofacial computer aided surgery: patient-specific finite element models to predict surgical outcomes, *Computer Based Biomechanics and Biomedical Engineering*, 1997, 8(4):259-65, Aug 2005.
- [3] Yamada T, Mori Y, Minami K, Mishima K, Sugahara T, Sakuda M., Computer aided three-dimensional analysis of nostril forms: application in normal and operated cleft lip patients., *Journal of Craniomaxillofac Surgery*, 27(6):345-53, Dec 1999.
- [4] Aarabi, P., Hughes, D., Mohajer, K., Emami, M., The Automatic Measurement of Beauty, *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics 2001*, October 2001.
- [5] Hjelmas E., Face Detection: A Survey, *Computer Vision and Image Understanding*, 83(3):236-274, Sept. 2001.
- [6] Anagnostopoulos, C. Anagnostopoulos, I. Vergados, D. Papaleonidopoulos, I. Kayafas, E. Loumos, V. Stasinopoulos, G., A probabilistic neural network for face detection on segmented skin areas based on fuzzy rules, *Electrotechnical Conference, 2002. MELECON 2002. 11th Mediterranean*, 493 - 497, May 2002.
- [7] Ming-Jung Seow Valaparla, D. Asari, V.K., Neural network based skin color model for face detection, *Applied Imagery Pattern Recognition Workshop, 2003. Proceedings. 32nd*, 141 - 145, Oct 2006.
- [8] Andreas Lanitis, Chris J. Taylor, Timothy F. Cootes, Automatic Interpretation and Coding of Face Images Using Flexible Models, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997, (19): 743-757, July 1997.
- [9] M.J. Jones and J.M. Rehg, Statistical Color Models with Application to Skin Detection, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1999, 274-280, 1999.
- [10] C. Kotropoulos I. Pitas, Rule-based face detection in frontal views, *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 1997, (4): 2537-2540, 1997.
- [11] A. Desilva K. Aizawa, Detection and tracking of facial features by using a facial feature model and deformable circular template, *IEICE Trans. Inform. Systems*, 1995, (9): 1195-1207, 1995.
- [12] J.M.S. Prewitt, Object enhancement and extraction in picture Processing and Pcyhopictorics, *B. S. Liplin and A. Rosenfeld, Eds.* New York, Academic Press, 1970.
- [13] Smith A. R., Color gamut transform pairs, *Computer and Graphics*, (3): 12-19, 1978.
- [14] Gonzalez R. Woods . E., Digital Image Processing, *Prentice Hall, Second Edition*, 2002.