ÇANKAYA UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
COMPUTER ENGINEERING

MASTER THESIS

FACIAL MODELING SOFTWARE

ZAFER YAPICI

SEPTEMBER 2012

Title of the Thesis **: Facial Modeling Software**

Submitted by **Zafer YAPICI**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University

Prof. Dr. Taner ALTUNOK
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Murat SARAN
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

(Title and Name) Co-Supervisor	Prof. Dr. Mehmet R. TOLUN Supervisor

Examination Date : 17.09.2012

Examining Committee Members (first name belongs to the chairperson of the jury and the second name belongs to supervisor)

Asst. Prof. Dr. Reza HASSANPOUR (Çankaya Univ.) ....

Prof. Dr. Mehmet R. TOLUN (TED Univ.) ....

Asst. Prof. Dr. Kasım Öztoprak (Karatay Univ.) ....

STATEMENT OF NON-PLAGIARISM

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : ZAFER YAPICI
Signature :
Date : 17.09.2012

ABSTRACT

FACIAL MODELING SOFTWARE

Yapıcı, Zafer
M.S.c., Department of Computer Engineering
Supervisor : Prof. Dr. Mehmet R. TOLUN

September 2012, 60 pages

The subject of this thesis is related to modeling human face according to ideal face size which is obtained using symmetry and Golden ratio rules. A software is developed during this study. It has mainly two missions. First of all it measures and calculates the ideal face size. While realizing this step, it uses the magnificent Phi Number (Golden Ratio). Because each calculation about beauty is based on this ratio. After finding the ideal face size for each facial part, the program enables to change any of this parts which spoils the symmetry or Golden ratio rules. Because spoiling of these rules leads to ugliness at the human face. The software gives the true proportions that must be. Another mission of this software is changing the face with help of translate, shrink, and grow tools as a whole according to the conditions told above. Each step of the software is tried and the output gives the right resultant according to the calculations made in Golden ratio.

Keywords: Facial Modeling, Modifying Face

ÖZ

YÜZ MODELLEME YAZILIMI

Yapıcı, Zafer
Yüksek Lisans, Bilgisayar Mühendisliği Anabilim Dalı
Tez Yöneticisi : Prof. Dr. Mehmet R. TOLUN

Eylül 2012, 60 sayfa

Yapılan çalışmanın konusu, simetri ve Altın oran kurallarına göre ideal yüz ölçülerini modelleme ile ilgilidir. Bu çalışma boyunca bir yazılım geliştirilmiştir. Yazılımın iki ana görevi bulunmaktadır. İlk görevinde öncelikle ideal yüz ölçülerini ölçüp, hesaplar. Bunu yaparken muhteşem Fi sayısını (diğer adıyla Altın Oranı) kullanır. Çünkü güzellikle ilgili yapılan her hesaplama bu orana dayanmaktadır. Yüzdeki her parça için ideal yüz ölçüleri bulunduktan sonra program bu parçalardan simetri veya Altın Oran kurallarını bozan parçaların değişimine izin veriyor. Çünkü bu kurallar bozulduğunda insan yüzünde bir çirkinlik oluşuyor. Sonuçta yazılım olması gereken doğru oranları veriyor. Bu çalışmadaki yazılımın bir diğer görevi ise yüzün tamamını değiştirme, küçültme ve büyütme araçlar yardımıyla bahsedilen koşullara uygun şekilde değiştirmek. Yazılımın her aşaması denenmiş ve sonuçta Altın oran a uygun şekilde yapılan hesaplamalar elde edilmiştir.

Keywords: Yüz Modelleme, Yüz Değiştirme

ACKNOWLEDGMENTS

I want to thank to my teachers who helped during my master education and preparation of this thesis. Moreover, I want to introduce my great thanks to Prof. Dr. Mehmet R. TOLUN, Asst. Prof. Dr. Reza HASSANPOUR who have huge labor in every step of this work. Finally, thanks to external member of jury Asst. Prof. Dr. Kasım Öztoprak.

TABLE OF CONTENTS

LIST OF FIGURES

INTRODUCTION

This thesis is related to the facial modeling software which is very popular recently. In the first part, modification of human face according to the ideal face size is investigated. Then required facial changes with using calculations and measurement is studied. I have searched what is the criterion of beauty and ideal size of human face. I have come across that ideal human face is measured by gold ratio with division of some height and width of face features, so I have found some methods that to determine the real size human face features in centimeter. I had some restrictions during take photos and to test this photo in the implementation, so I have succeed to fixed one pixel to centimeter. Gold Ratio is 1,618, it is come from Fibonacci numbers. If division of some face features is near to gold ratio, human face more beautiful. I have found gold ratio also in the nature, for example; pyramids in Egypt, the painting of Leonardo Da Vinci which is named Mona Lisa, sunflower, snail, Pine Cone, Heart Beats, Human body, face, etc... The other criterion for ideal human face is symmetry on face. Symmetry on face is determined by vertical fifths, and vertical horizontally methods. In the implementation of this thesis, I have tried to find how this methods will work on the human face, are they really good methods to determine human face beauty or not? I will define our methods on my implementation and I will discuss the result in the conclusion and discussion part. My implementation is on .Net with C# language, I have used some available tools on .Net.

CHAPTER I

1.1. USAGE AREAS

Human face defines the front of the human head from the forehead to the chin. It is very important because face is the most curious part of a human by visually. People may affect environment and have succeed in their social and business life. This importance leads to making some cosmetic or plastic interventions to the faces according to the face type. After having studied on human face for every ways in detail since many years, scientists arrive at a consensus about ideal face size. This is golden ratio and symmetry on face. So, finding a human’s golden ratio and symmetry will give people about their ideal face size and finally beauty. This subject has been the major problem of plastic surgery for years. With the help of this solution, medical doctors can develop new ways on their patients. It may provide good solution during take a decision to change on human face.

Physical attraction can be studied from a number of perspectives, including facial symmetry body structure, genetics, youth and mimics. This thesis interests in facial beauty. Beauty is one of the most important issues from the pre historic ages so that people suffer agony in addition to spending lots of money and time. This situation drags the science into making more deeply researches how to improve and realize this.

Beauty is accepted as an attribute of a human, animal, place, or idea that is well accepted by people and is studied as a component of aesthetics, culture and sociology. An "ideal beauty" is a quantity which is appreciated for excellence. However beauty is a relative concept that can change from human to human. In addition to the definition of beauty there is a certain fact. The eyes of humans are created and designed to look for an equilibrium in everything. If the objects are found in correct proportion then human eyes will accept them beautiful. In this thesis, calculating ideal face size is investigated. Many scientific researches show that ideal face size is found by calculating a human’s golden ratio and the symmetry of his/her face.

1.2. GOLDEN RATIO (Phi Number)

There are a lot of examples of architectural and artistic artworks from all over the world that are accepted as masterpieces by everybody. The reason of being magnificent is making them built in the perfect balance and proportion. Scientists have calculated and proven this equilibrium. This excellent balance is attributed by scientists as “Phi Number” or "The Golden Ratio." Many people are acquainted with Pi number, because it is one of the most well-known irrational numbers. However there is another irrational number which is as popular as Pi. This wonderful number is Phi, which is also named as “Golden Ratio” or “The Golden Mean”. There is a technique for finding Phi is to consulting the solutions of the equation mentioned below

$$x^2 - x - 1 = 0$$

When the equation is solved, the roots are obtained as;

$$x = \frac{1+\sqrt{5}}{2} \sim 1.618...$$

Phi (Φ = 1.618033988749895…) is the solution of the quadratic equation. Phi number comes from Fibonacci numbers which is sum of previous two numbers; 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584. After number 233, if you divide to the previous number you will get only 1,618.

For example;

233 / 144 = 1,618

377 / 233 = 1,618

610 / 377 = 1,618

987 / 610 = 1,618

1597 / 987 = 1,618

Phi number or golden ratio was first used in Ancient Egypt and Ancient Greece so that, it is accepted that the Egyptians have used both pi and phi numbers in building the Great Pyramids. Moreover, in designing the Parthenon Ancient Greeks have taken the advantage of this proportion. The Golden Ratio is used in mathematics, geometry, life, the whole universe and its applications are limitless from mathematics to beauty. In this thesis, the usage of Golden ratio in face beauty is researched and studied. The ideal face size is obtained by using Golden ratio. In spite of being a relative concept, there is a common beauty comprehension that is proven scientifically. The physical attraction of a human body is increased if the organs are symmetrical and in balanced proportion. Correspondingly, if human face is in equilibrium, then it is more appreciable and people find it beautiful. So it can be summarized as physical attraction depends on golden ratio.

1.3.SYMMETRY

Symmetry has no exact definition, however it creates a charming effect and makes people feel pleasant in both aesthetically and proportionality; so that it reflects excellence. Today, symmetry of the face becomes beautiful to human eye. Also this is proven scientifically. Moreover, the symmetry is assigned not only with the balanced proportions, but also the likeness between the right and left parts of the face. Facial symmetry is just one component of bodily symmetry which has been being studied on for finding the correlation of attraction by many scientists. Computer graphic methods prove that symmetrical faces are more attractive than relatively asymmetric faces. Although beauty is a relative concept, humans who have symmetrical properties look much more attractive to other people because the visual perception system of human can comprehend symmetric issues more easily than any kind of asymmetric issues. An example of symmetric face is shown in the figure below.

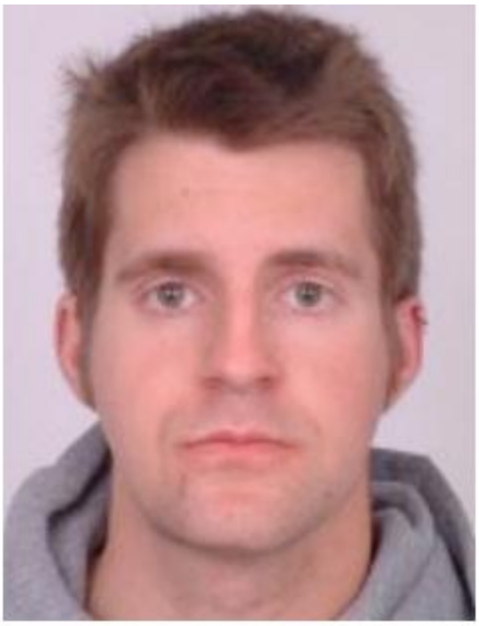


Figure 1 - Symmetry on Face

Everything is in order if Golden ratio and symmetry takes place in a human's face. However asymmetry generally makes the sense of irritation. In spite of that beauty changes from human to human, there is a standard well accepted of beauty because of its mathematical proportion. An example of an asymmetric face can be seen below.

Figure 2 - Symmetry on Face

Asymmetric faces affect not only the outer appearance but also the psychology of people. With the help of developing technology and medical science, it is possible to make some changes on people’s faces using plastic surgery. In this thesis, two types of changes are researched. In the first type, the wanted facial part can be changed. For example a human is not happy with his/her nose and the nose is not suitable for that person’s face. Then the plastic surgeon should change it according to the most suitable choice with his/her face. There is a sequence of mathematical operations should be done and the most suitable proportioned nose is occurred. In the second type of changes, all of the face is changed with translate, shrink, and grow tools according to the symmetry and golden ratio. This one is more used for detailed and complicated situations.

1.4. MOTIVATION APPLICATION

Human beauty depends on the “Golden Ratio”. So the human face is full of with examples of it. The head generates a rectangle called as golden rectangle that has eyes in the middle of it. The nose and mouth are found at a place which is named as golden sections and have distance between eyes and the bottom of the chin.

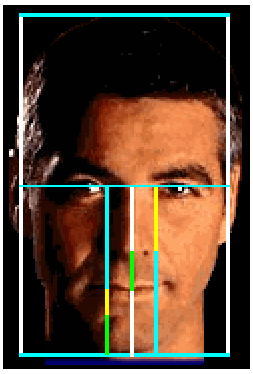


Figure 3 - Golden Ratio Front Face

Even the portrait view of human face also emphasizes the Number Phi. The first golden section is shown with blue and start from the front of the head that assigns the ear opening position. The sub sequential golden sections are assigned the neck viewed in yellow, the back of the eye shown in green and the front of the eye and back of the nose and mouth signed as magenta. The facial dimensions from top of the head to the bottom also reflects the Golden Ratio, in the eyebrow positions shown with blue, nose signed with yellow and mouth assigned with green and magenta.

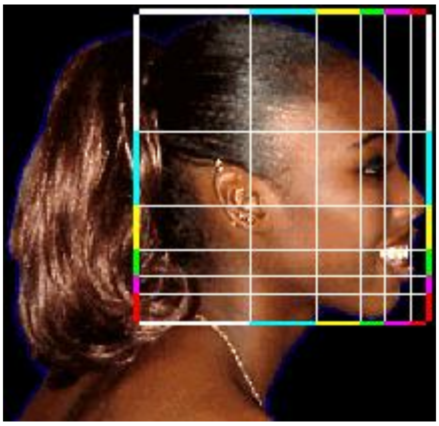


Figure 4 - Golden Ratio on Side

Each beautiful part of the human body is based on Phi Number. The structure and dimensions of teeth are also depends on this number. So the Golden Ratio can be expressed as true balance and correct proportion in itself. According to this ratio, an ideal human face is divided into three parts. The first section starts from the top of the forehead and ends with the top of the eyebrows. The second section begins from the top of the eyebrows terminates at the bottom of nose. The third one begins from top of bow of Cupid to the bottom of the chin. To satisfy the Golden Ratio, all of these three sections should have same length. In addition to the definition of facial balance, the excellent facial shape is accepted as the 2/3 width of its length, and should be symmetrical on both sides.

A-First, both the facial width and length are measured. Then, the width of the face is divided into the length. The perfect result should be 1.6 which represents the golden ratio value, and also this situation is explained as the face of the beautiful person is about 1 1/2 times longer than its width.

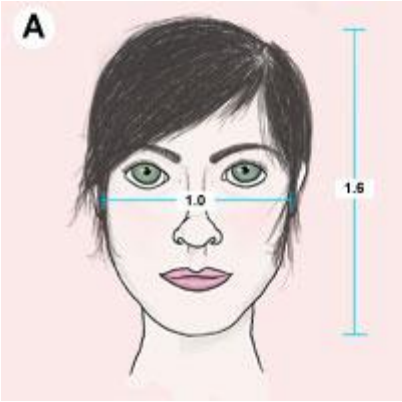


Figure 5 - Golden Ratio A

B-Next, the face consists of three segments. These segments are measured which starts from the forehead hairline to a specific dot between the eyes, from that dot between the eyes to the bottom of the nose, and from the bottom of the nose to the bottom of the chin. If the obtained proportions are equal to each other, then that human is accepted as much more beautiful. Also these segments are called as horizontal thirds.

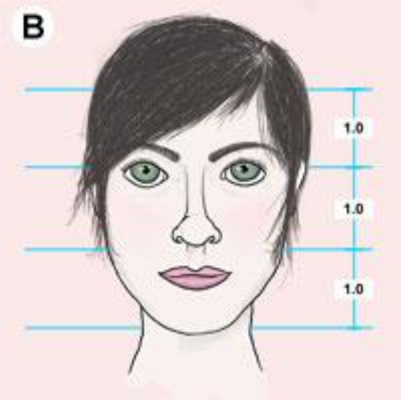


Figure 6 - Golden Ratio B

C-Lastly, other properties of faces are measured by scientists to identify the facial proportion and symmetry. Because the way that goes to beauty passes through equable proportions and symmetry. Moreover it represents an equivalence of facial components with respect to shape, size and relative positions on opposite sides of a dividing line or about a center. On an excellent face, the nose length and the ear length should be equal. In addition the width of an eye should have the same distance between two eyes.

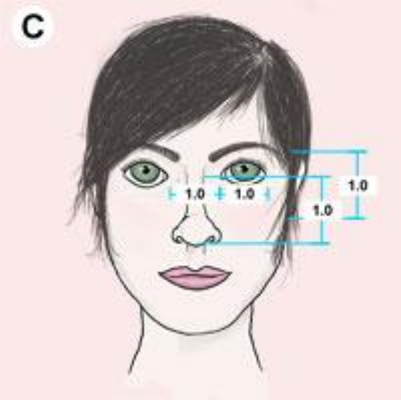


Figure 7 - Golden Ratio C

The perfect is one is 10. However no one can reach this value and most of the people have a score between 4 and 6.

- Face height / Face Width
- Lips - between Eyebrows / Nose height
- Face height / Chin - between Eyebrows
- Mouth width / Nose width

- Nose width / Between nares
- Between pupils / Between Eyebrows

The results of all these operations should give the golden ratio (1,618) for the perfect beautiful face.

The face is divided into five segments that use the eye width from corner to corner as a measurement point. It starts from the outside edge of one ear to the other ear. An ideal and excellent face should have a width of 5 eyes distinctly. Because of not being designed by devices or computers, humans who don't have 5 eye widths on face can not be said ugly. The width of the nose, at the nostrils, should be 1/5 of all the face, or one eye width. This is called vertical fifths. It can be viewed like below.

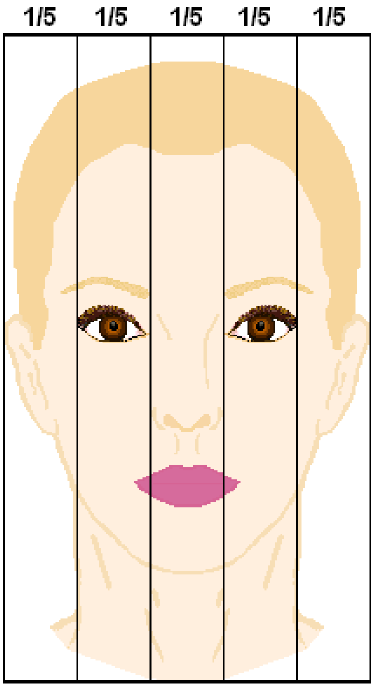


Figure 8 - Vertical Lines on Face

Another issue to calculate face symmetry is Frankfort Line. In order to identifying Frankfort Horizontal Line or red horizontal line in other words, a photograph from side view should be taken and a vertical line should be lined from glabella to the chin. This line must be vertical to have an excellent face. The next line is drawn horizontally from right above the auricular canal, to right above the tragus. The Frankfort line should pass from the highest point of the ear canal through to the lowest point of the eye socket. An example can be seen below.

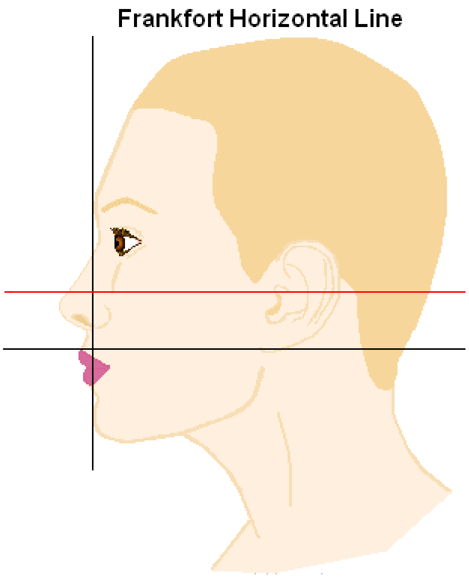


Figure 9 - Frankfurt Horizantal Lines on Face

Shortly, face symmetry is calculated by horizontal thirds, vertical fifths and Frankfort horizontally which defines the exact beauty scientifically depend on the Golden Ratio on Face.

Beauty is being such a popular subject for ages, everybody wants to be beautiful naturally. In addition to this, with the developing technology science makes lots of researches on this subject to go further. In all these studies the fundamental concept is Golden Ratio, because all the beauty concepts in the universe base on this ratio. In this thesis; methods, calculations and measurements of modifying human faces for required facial parts or for all face according to Golden ratio are investigated. All studies prove that Golden ratio is very important and is used in all calculations.

CHAPTER II

2.1. BASIC INFORMATION ABOUT WORK AND SIMILAR WORKS

Aesthetic is very important for both men and women. However, especially women spend lots of money for being beautiful. Some people need little changes and they can overcome this problem with the help of cosmetics and make up. However some defects are not so light every time and cannot be corrected with cosmetics. This time medical corrections are needed to correct these defected parts. Plastic surgery is directly related into this area. Especially, after developing some computer programs, these studies get acceleration all over the world. In early times, people can see themselves after operations and there is no turning back except another operation if there is a fault or unwanted situation about the process. This is a risky and time consuming process. However now everything is easier. Many computer programs used by plastic surgeons help both doctors and patients to satisfy pleasure. Also every step of the operations can be seen before the process and changes are interfered in before the operation. So there is no need to turn back. After the developing of Photoshop, it helps to mark an era for these kinds of works. Every facial parts or the face as a whole can be changed and changed situation can be seen. So, this program prevents not only the loss of money, time but also decreases the physical and psychological pains of patients.

Another program used by surgeons is called Virtual Plastic Surgery Software. It works like Photoshop, but has much more specialties like calculating and measuring the facial parts more deeply. This program works taking base of Golden ratio in making calculations. These programs depend on what changes user do. There is no any calculation of gold ratio on human face.

They are for general use to change and create any photos. In my work, user will work on human face only, and will see exact real measurement of human features and will work on it, so for example for doctors the real measurement will help them for operations.

In this thesis, Luxand and C# are used to calculate the ideal face sizes according to the Golden Ratio. Ideal face is obtained by using Golden ratio calculations. This measurements and calculations are done by using Luxand. Because of Golden ratio is a constant number, it is easier to make calculations. After making calculations, needed or wanted changes take place as the second part of the thesis. Two types of changes can be done. First of all changing the wanted facial parts. After taking the measurements Luxand calculates according to the Golden ratio and ideal face size. So the most suitable proportion is detected and the main program written in C# shows the changed facial part. The second step is changing the face as a whole. After loading the face into the program, it calculates the all face according to the Golden ratio again and formalize it with the closest proportion automatically. Then visualize the changed face.

Luxand face software development kit is a tool can be add to the visual studio project to work and develop anything. Thanks to this tool, I have succeed to find 66 point on face features. eyes, eye contours, eyebrows, lip contours, nose tip, chin and so on. It draws a square on face and circle 66 point on face on uploaded any photo. Then I have tried to find measurement of face features to be able to calculate available face ratio from photo. Then I have decided to restrict uploaded photo with some rules to be able to give real measurements of ratio in centimeter to the user. The photo should be taken from 180 cm far from face, photo should cover all face, photo should be in 1024x768 resolution. Then software gives the measurement and gold ratio of human face, in the modify to gold ratio tab, it is changed face features measurement with help of image tools to be able to near to the gold ratio. In the modify to gold ratio part, I have used again a tool for .Net which have grow, shrink, translate functions. I have used dll import in C# to use ImageWarper.dll. With these functions I have modified the related features on

face to able to near to the gold ratio. I have applied these functions on points of related features which I tried to change to the gold ratio proportion. In the required changes, again I have used ImageWarper, but I have remain the changes to the user, and again I have calculated the measurement and gold ratio in the below of photo, so user may decide which function should be used to be near to gold ratio.

Changes in facial dimensions assessed from lateral and frontal photographs: is a work done by Samir E. Bishara, D. Ortho., Gregory J. Jorgensen, and Jane R. Jakobsen in the USA. This research is made on children between the ages 4 and 13, because these are the most suitable ages for a human having changes on face dimensions. By the help of this work, facial changes can be evaluated and obtained standards can be used for different aims. The experimentals are 20 people; 10 of girls and 10 of boys. First their frontal and lateral photos are taken and then measurements are done. There are 32 signs identified and 29 dimensions measured. The boys' and girls' measurements with respect to their age are evaluated with the statistical method of analysis of variance. It is shortly called ANOVA.

As a result, at the end of this work there are some results obtained as mentioned. The length and width of a face is proportional amongst them. Length is two times of the width. Secondly, the change is most constant parameter for the dimensions of the eyes. Moreover, if there is a change in some part of face part, the standard deviation is affected much more than the change in that part. Then, if there is an increment in the length and vertical depth of the nose, it is twice of width of nose. Finally, the most volatile change occurs in lip size. Therefore, the taken photos are standardized and gives a correct description of the changes in face while growing.

Anatomy- and physics-Based Facial Animation for Craniofacial Surgery Simulations: A research done by E. Gladilin S. Zachow P. Deuflhard H.-C. Hege in Germany. In this work, modeling is presented in a general definition.

The aim of this modeling is using it especially in plastic surgeries. The facial expressions can change according to feelings. In this study, they are simulated as closest as the real world. From this aspect, this modeling technique is different from the traditional animation techniques that used in computer graphics. The facial muscles impose upon emotions and give a steady form to face physically. Finite element method is a numerical solution technique that is used to find boundary values of face. In this approach the human face data is taken from head tomography and applied on a geometric model. So, by this way personal emotional expressions on face are developed and prepared for using on a computer.

Human face modeling is a very popular, developing and complicated subject because it is also applied to many different fields both related to computer and science. The subject of interest in this paper, face modeling is investigated for craniofacial surgery (CAS). CAS is a kind of plastic surgery that especially interests in deformities of bones and skull. Therefore, this is a very important working area for the sufferers from these kinds of diseases on their skulls especially. The physical distortions on human faces affect people psychologically because physical appearance is very important in social life and people evaluate each other according to their face values at first. The conclusions of this work will give idea to this kind of plastic surgeries how to correct the wrong places and satisfy facial expressions.

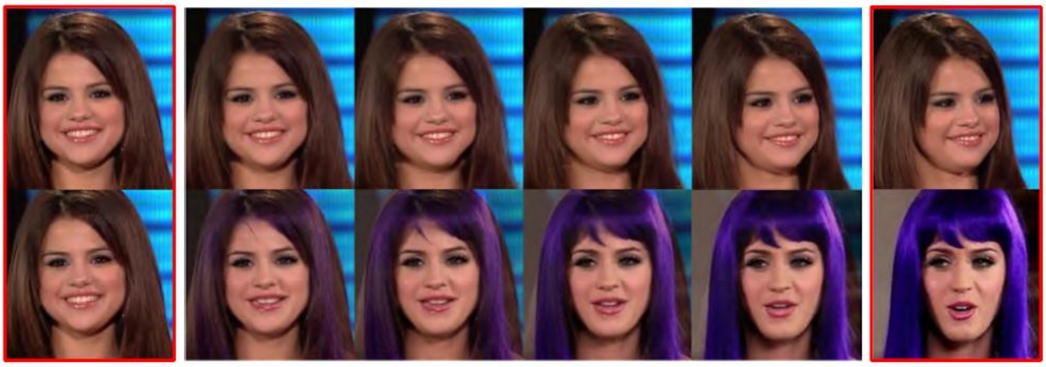
A geometric model is needed to see how the patient looks after the surgery. To obtain a realistic view, modeling must be done in detail especially for muscles and tissues. On the other hand, computer tomography (CT) and magnetic resonance imaging (MRI) are the useful technologies to realize medical imaging in three dimension. In addition finite element method (FEM) models muscles and tissues in computer context. These techniques support each other and are applied to the patient idiosyncratically.

The researchers study on 3 modeling techniques in this work. These are muscle modeling, soft tissue modeling and geometric modeling. These methods give opinion how to get the most suitable solution. In addition to this, these modeling techniques are very modern and innovator for correcting the facial image according to emotions, and also help deformed muscles and tissues are recreated.

In summary, this paper deals with facial expressions according to emotions. Physical and anatomical techniques are used for developing the most correct result which is as close as the real. Some facial structures can not be seen with eyes. These can only be obtained by microscopes and not be taken with tomography. There is a constant value and relationship between muscles and tissues. So at this kind of situations, the muscles and tissues are behaved according to this constant. A statistical method, finite element method (FEM), is used for correcting the distortions on facial muscles. Both single and complex facial movements are simulated with this technique. The results of this study will be used in computer aided techniques for facial surgeries.

Face Morphing using 3D-Aware Appearance Optimization: a study done by Fei Yang, Eli Shechtman, Jue Wang, Lubomir Bourdev and Dimitris Metaxas in the USA for Adobe Systems. Morphing is a special technique that is used for transition between pictures. There are some standard morphing techniques applied. In these kinds of applications, there is blurriness in photo frames between transitions.

However a new technique for face morphing is developed in this study. The input photo is recovered by using three dimension face geometry. Then the general posture and the facial expression in the posture are combined for the geometry and satisfy the transition fluently and orderly. Eventually, facial morphing is accepted as repetitive in this paper. With the help of geometrical techniques similarity between adjacent frames of the photo is detected and frames are combined. In addition to this, morphing technique experiments done for this paper provides better quality in results. The facial postures, expressions in the postures and final images satisfy much more successful solutions than former works.



In the paper studied, system enables generating facial morphing between distinct postures or different expressions. These can be done fully automated and in high quality. Images found on the top show morphing on the same face, images found at the bottom indicate morphing between different faces.   The input images are the first and last column and shown with red frame.

In summary, this paper is mentioned facial morphing animation between two faces with different postures and facial expressions. The aim is realizing this work automatically and with a high quality. Three dimensional facial models have been selected from a small database that includes realistic facial expressions. Some casual photos are also added to constraint the facial distortions. In traditional warping methods the system looks for a convenience between the given two faces, however the work done in this paper bring an innovation to morphing. It uses three dimensional face model flow and evaluate independently. So the holes are avoided. The optimization of view balances wrong alignments and changes in uncovered shape models for minor faults. This allows the system to work without blurriness and hosting artifacts. The purposed work in the future is obtaining correct results in facial optimization, face extrapolation, non-linear interpolation and applying this approach to other three dimensional models. For example if it is applied on a posture with sunglasses, it should give the correct value.

A Morphable Model For The Synthesis Of 3D Faces : a work done by Volker Blanz Thomas Vetter in Germany. Morphing, modeling and three dimensional face synthesis are the main working areas of this paper. Three dimensional facial images can be obtained in two ways. Either automatically from a few number of photos or human modeled interface can be used. However there are some problems in computer based facial modeling. One of them is both facial or three dimensional facial images are automatically computed by one to one correspondence. The other one is avoiding unwanted face appearance. The modeled face is aimed to arrange for a natural look. Vector space representations of three dimensional face models are needed to obtain. To satisfy this, the shapes and tissues of 3D example face sets are used from a sample database. Moreover, new faces and facial expressions can be added for creating new combinations.

These three dimensional facial sets are restricted according to shape and tissue. Then the obtained data is used in algorithms for modeling manually and matching automatically. Facial image processes are taken from these three dimensional reconstructed data for a realistic work. More complex values like gender, the integrity or disjunction on the faces are also obtained from facial processes in this paper.

In this work, there are 200 faces created for using in the example database. The morphing facial images are built automatically. An interactive modeling is used for facial processes that allow users modifying facial properties and generating characters. This is realized by using coefficients. In spite of restricting the probabilistic values, it enables an extensive selection opportunity about faces and their natural outlook. There is an important point to be mentioned here is the automatic reconstructed three dimensional faces of people from the Caucasus are not included in the sample database. They have a strict and powerful morphing modality. All the facial photos are taken with an ordinary digital camera under regular circumstances. As it is defined before, the faces are taken from the database which is made up of 200 faces, 100 include shape related and 100 include texture related. So a morphing technique is formed. Each of these 200 faces are divided into 4 parts and matching is repeated 105 times as a whole. The calculation time of this process is completed in 50 minutes with the processor SGIR10000. It is a real trouble that regenerating a single photo image to a realistic three dimensional facial shape and texture. Except that, if only the input image is known, then the consequence of the technique will give correct result. In addition to this, there can be observable diversities between real image and rotated photo which is converted more than a 60 degree angle. Also there are great number of programs that is used for converting two dimensional images into three dimensional ones. The obtained results are used in automatic processing of facial image whether it is a photo taken or a sequence in a movie. The three dimensional form of a face enables dividing both facial area and background. Therefore the face is allowed to be integrated with other three dimensional objects found in the photo. These objects can be defined as scarves, hats or glasses. After that the lights or shadows in the photo are computed and differentiated from the photo as background or foreground. In addition to this, some different properties like mentioned can be added or removed from the photo in order to change the face in the photo. Sometimes backgrounds are invisible and become apparent after doing these operations. In such situations, the holes of adjacent pixels are fulfilled. This technique is also applied to popular paintings but correcting the texture is nearly impossible. For this paper it is tried on Mona Lisa and the result is undesirable because of shadows and lights. This situation gives birth to developing distinct method which will be used in paintings. In such a method the paintings are going to be converted to novel views. To obtain a new enlightenment, two three dimensional face images are restored with a distinct lightning technique and new pixel values are combined with the original ones. For obtaining a new posture the same technique is applied to shading. At the end the recreated paintings are obtained by using both two dimensional and three dimensional forms.

Fat Distribution: a Morphologic Study of the Aging Face : a work done by Lisa M. Donofrio in the USA. All people get older by time and while getting older some alterations take place in humans. In this paper the changing on aging faces are investigated. The result of aging is not only the skin loss or wrinkles on the surface of the face, but also psychological changes occur. These changes can be observed by detailed graphical representation in 3D. The geometrical changes occur in fat proportion which is the symptom of agedness and results in fat anomaly. The solution of correcting this fat diffusion and key of youth is balancing the facial fat structure. This also helps gaining mimics on face. Therefore, the reason of facial changes is fat distribution. For example there is a plenty of fat in the area of cheek and this is a probable result. There are some definite differences between young and old faces. First of all the young face consists enough amount of fat. The facial surface looks smooth and sleek. There is no need for using cosmetics to make the skin more beautiful. Moreover convolutions are definite in 3D looking young faces. There is a tight line from ear to ear and multiple arcs on the lips which can be easily seen from the front view of the face. In addition, the young faces have three primary arcs in profile. The first arc is the most important one. It starts from top of eyelid through the bottom of face. The second arc is presented at jawline from lateral inferior maxillary bone to pogonion. The third and final arc is found between forehead and top of eyebrows. Apart from this, there are a few numbers of secondary arcs that surround the circumference of mouth. All these properties can be said as a brief definition of young faces commonly. Exactly the opposite, there are some common properties of old faces. First of all, there is a mild state of dissociation on the face surface. It has pockmarked topographic view. It has no homogenous appearance. For that reason, there is use of cosmetics needed. The arcs defined above have irregular, unbalanced and non uniform shape. From the front view, the cheeks and jawline have rough view. The arcs of lips have lost their angles and beautiful looking. From the profile looking, the arc of the cheek becomes unmarked. The forehead and eyebrows fall down.

Briefly, it is mentioned in the study of this paper that the face of a human changes with respect to age. In addition to wrinkles, there are special kinds of distortions in the arcs of skin and it causes drooping and skin excess on the skin.

Toward Automatic Simulation of Aging Effects on Face Images : is done by Andreas Lanitis, Chris J. Taylor, Timothy F. Cootes in the USA. The human beings change both physically and mentally while getting older. In this paper, physical changes and especially the alterations on the face are investigated. In the experiments done, it is obtained that in spite of having individual properties for each person in getting older, there are some definite and common characteristics present on facial images in young versus old people. The conditions of getting older occur stepwise. However lifestyle, gender, ethnicity and genetic factors affect this senility process. So, the facial appearances begin to change at different ages for everybody. In this study, the facial effects of getting older are defined by learned techniques and results of the experiments prove that correct predictions can be done about oldness without seeing senility photos. Moreover it is shown that every people get old in a distinct way. As it is mentioned before, some factors like genetic properties affect aging. The technique used in this paper differentiate from others as applying these aging techniques on new facial images in spite of guessing the past or future look of that human. Another property of this technique used in the paper is creating a face determination system. This system is powerful with respect to changing age. In this sense, the appreciable age of the experimentals their facial photos are normalized for training and testing. This normalization takes place before training and determination steps, because this situation separates changes in aging. The obtained results from experiments show that normalization of aging helps increasing and developing a higher quality for face determination system.

There is a cooperation between human faces and computer systems for face recognition. Also the use of computerization techniques here increases the improvement of face determination. A specified number of facial images are practiced for facial determination. Therefore each face properties are distinct from each other for every different people and this leads to define facial identification in new photos. There is an important point to be paid attention for having more accurate results in facial determination. The physical appearance of the face should be put distinct not to be impressed by the changes in intra personal.

Each working area may come up against some difficulties and there is a number of problems that has been faced in this paper. They can be summarized as the multiplicity of changes in aging, firmness of personal properties and aggregating the practiced data values. To overcome this problem a number of statistical techniques applied and used with facial changing rules while getting older. I want to mention some of these rules according to the paper. The view and structure of both neck and face change with respect to age. The functions of muscles consume their performance and this leads to flabbiness in overview. The cheek droops and “double chin” situation occurs. Moreover the length of nose can increase in some humans and this makes the nose look more in the foreground. In addition to this the surface of the skin may change. There can be generation of dark spots and the dimensions or amounts of them may increase. The reason of this situation may come from sun rays or genetic factors. Other disadvantages of passing time are classified as thin, dry and wrinkled skin. Even though the facial wrinkles are unavoidable for everybody, sun rays and smoking accelerate this process. The gristles in the ears may cause elongating them for some people. Especially men complain about their big ears while getting older. Another problem observed related to ear is about wax in it. The amount of wax decrease and get dried. This dry wax cause blockage in ear canals which leads to loss in hearing. In addition to hair color, the colours of eyelashes and eyebrows turn into grey. The eye contour is the most sensitive part of the face. For that reason, the wrinkling starts from here and high dense of wrinkles create "crow's feet" situation. The movement of eyes remains limited and eyes look flabby. The bags and lividness occurs under eyes. The eye cornea has deformation in shape which is called “arcus corneus”. A light grey ring occurs in the outer section of eye. Iris, which is the coloured part of the eye has also sacrifice in pigments. Therefore very old people have grey or blue eyes. After a certain age people lose their teeth. This loss causes pursy mouth view. Another effect of dental problems is alterations in the mouth. The bones of the chin forfeit their specialities which results in changing the whole face. For finding facial determination the implemented statistical techniques are applied according to the properties mentioned above.

Age Classification from Facial Images: a research done by Young Ho Kwont and Niels da Vitoria Lobo in the USA. In this paper the researchers have investigated finding human age from the facial images. Alterations in facial ratios and wrinkling analysis of skin satisfy both the theory and applications of determining age from face images. To oil the wheels, age categories are divided into three for this paper. First of these stages is infancy (babyhood). Second one is adolescence hood and finally senescence hood. However the prediction and ordination of humans according to their age from their facial photos is a still being studied subject. Therefore the computerized part has not been completed yet. One of the most important property of this paper is it is the first written study which has been worked on wrinkling and age classification. The steps that have been followed in this paper can be summarized as;

1. Firstly finding the specifications of a face. This is the most important step to be examined. For achieving that the following steps should be followed.

1.1 Finding facial trait roughly

1.2 Then determining chin,

1.3 And finding face sides.

1.4 After that the top of the head should be computed.

1.5 The eyes should be found.

1.6 Finding the mouth.

1.7 Finding the nose.

2. The second step enables computing facial parts’ ratios.

3. Analysis of wrinkling should be done.

4. Finally, the second and third step should be integrated and an analysis is generated for determining the age group of a human.

If there is a need to sum up this paper, it can be said that the work has been completed by using a database which is made up of 47 faces that include babies, young people and old people. A theory about classifying age groups from face images has been developed for this paper. However the age groups are restricted with three as babyhood, adolescence hood and senescence hood for this paper as mentioned before. In the next step, finding facial properties, determining facial ratios, wrinkling analysis and finally age categorization take place orderly. Encouraging and hopeful results about this field are obtained at the end of this work.

Photoshop is one of the most useful tool for making facial changes. I have also used this program in my thesis work. However distinct from it I want to explain how to make someone look older by using Photoshop. Load the image that you want to retouch to the computer by transferring it from a digital origin or by scanning it from a mark. Open Photoshop then choose "File" and click "Open". A dialog box is opened, browse to and open the loaded image. Let select "Magnetic Lasso" for making selection around the hair in the photo or image. Select the "Brush" tool for adding grey and white highlights to the hair. In addition "Magnetic Lasso" also helps making selections at the neck and face skin. Select "Image" from main menu and hit "Hue/Saturation." From dialog box the "Saturation" should be adjusted and the "Brightness" descends until the preview looks fading away. Push "OK." Select the "Brush" tool, then set the colour to beige, the "Opacity" works optimum at 30 percent and paint in any dark spots. Alter the color slowly as working. Select "Filter" and click "Liquefy." Use the "Push" tool for making section on the face like under the eyes and cheeks where distortion occurs. Hit "OK." A slight selection can be made at one eye. Click "FX" button and choose "Drop Shadow." Arrange the settings in the dialog box for producing wrinkles under the eye. Hit "OK." Repeat this step to add more wrinkling under both eyes and next to the lips. Save the work. In contrast, photoshop also enables to make someone look younger. Here are steps to realize this. Copy the image choosing "Layer" then "Duplicate" from the Layer menu that fount on the top of the screen. Follow "Filter," "Blur," and "Surface Blur" for selecting a setting which softens the skin slightly. If the photo is different, blur settings will have to be examined. Click "OK." Then choose the Eraser Tool, and clean out everything from the image that needs sharpening in hair, eyes and jewelry. The

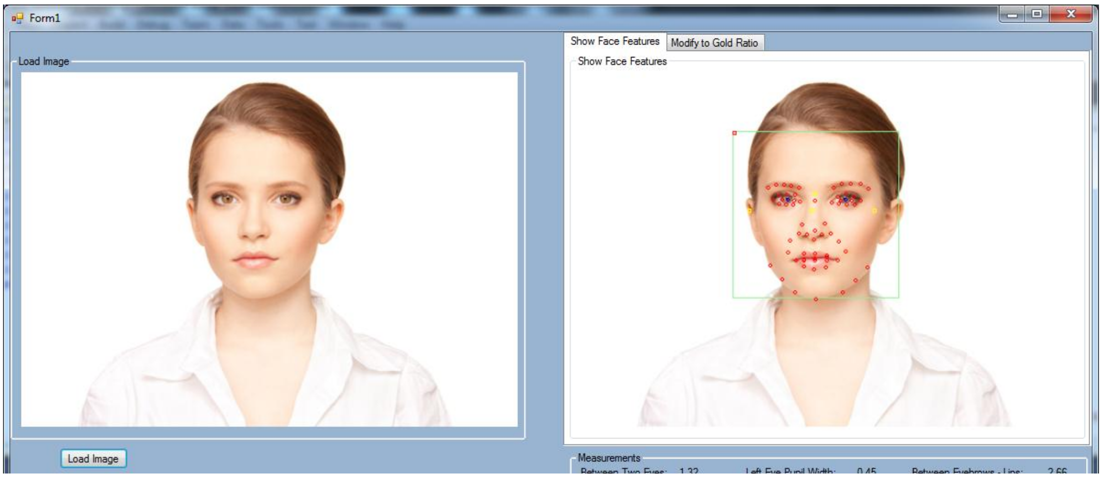
Zoom Tool will help getting close. Go to layers menu on the screen found at the right side. The opacity of the blurred layer could be decreased until the intended image is obtained.

In addition to photoshop, some other software programs are used for making facial changes. These programs are especially used in plastic surgery area. Facetouchup, Mirror Image Digital Imaging Systems, and LiftMagic can be given as examples for these programs. These programs and some morphing programs told above can be used both in medical area and security. The main difference of my thesis work is making the adjustments manually. After selecting the face, in modifying the gold ratio tab, any wanted measurements can be changed automatically. Also the changing ratio values are calculated again and written on the program. In the end, the changing face takes place on the screen.

CHAPTER III

3.1. IMPLEMENTATION AND DISCUSSIONS

The implementation is very important in this thesis. The most time taking part is the implementation part during the preparation of this thesis. In this part I will also discuss my restrictions and weakness of my work and ideas. I have started to search about face detection tools, and which environment I will work and develop. I have started to work first in the Matlab and try some ready code and tools for face detection. But I am not expert on Matlab, and couldn't find tools for detection of face features in one package. I am more expert on .Net with C#. I have found a software development kit for face detection in Luxand company, I have downloaded it with demo period, it is about $4000. I have added it as reference to my .Net project. I have designed a windows form to upload any photo and implement to find face features as below.



Between Two Eyes: 1.32 Left Eye Pupil Width: 0.45 Between Eyebrows - Lips: 2.66

Figure 10 - Face features

I should find now real width and length of face features. I have develop method and restrictions the user to able to succeed. I have measured my face features with ruler, and note them. Then I have taken a photo from 180 cm far from face, and I am restricting for all uploaded photos in same thing.

With this restrictions I have calculated face features size. I have calculated one pixel how many cm in real life. But I have got different cm to one pixel in each face features. Because my measurement with ruler is not exact the point of program mark it. I have used length of between two point formula to calculate length of face features in program. Please find my calculations below.

- Pupils length with ruler is 1 cm.

```
faceFeaturesSize.solGozBebegiGenisligi =
Math.Sqrt((((PointPlaces)pointPlacesArray[33]).point.X -
((PointPlaces)pointPlacesArray[34]).point.X) *
(((PointPlaces)pointPlacesArray[33]).point.X -
((PointPlaces)pointPlacesArray[34]).point.X) +
(((PointPlaces)pointPlacesArray[33]).point.Y -
((PointPlaces)pointPlacesArray[34]).point.Y) *
(((PointPlaces)pointPlacesArray[33]).point.Y -
((PointPlaces)pointPlacesArray[34]).point.Y)) = 26.08 pixel;
```

1 cm= 26.08 pixel.

- Face length with ruler is 17.5 cm

```
faceFeaturesSize.faceHeight = Math.Sqrt((facePosition.xc -
((PointPlaces)pointPlacesArray[11]).point.X) * (facePosition.xc -
((PointPlaces)pointPlacesArray[11]).point.X) +
(facePosition.yc - facePosition.w / 2 - ((PointPlaces)pointPlacesArray[11]).point.Y) *
(facePosition.yc - facePosition.w / 2 - ((PointPlaces)pointPlacesArray[11]).point.Y))
= 405.0025 pixel
```

1cm = 45.0025 / 17.5 cm = 23.143 pixel;

1cm = 23.14 pixel.

- Mouth width with ruler is 5 cm.

```
faceFeaturesSize.agizGenisligi =
Math.Sqrt((((PointPlaces)pointPlacesArray[3]).point.X -
((PointPlaces)pointPlacesArray[4]).point.X) *
(((PointPlaces)pointPlacesArray[3]).point.X -
((PointPlaces)pointPlacesArray[4]).point.X) +
(((PointPlaces)pointPlacesArray[3]).point.Y -
((PointPlaces)pointPlacesArray[4]).point.Y) *
(((PointPlaces)pointPlacesArray[3]).point.Y -
((PointPlaces)pointPlacesArray[4]).point.Y)) = 120.02 pixel
```

1 cm = 120.02 / 5 cm = 24.004 pixel
1cm = 24.00 pixel.

- Nose Height with ruler is 4 cm

```
faceFeaturesSize.noseHeight =
Math.Sqrt((((PointPlaces)pointPlacesArray[22]).point.X -
((PointPlaces)pointPlacesArray[49]).point.X) *
(((PointPlaces)pointPlacesArray[22]).point.X -
((PointPlaces)pointPlacesArray[49]).point.X) +
(((PointPlaces)pointPlacesArray[22]).point.Y -
((PointPlaces)pointPlacesArray[49]).point.Y) *
(((PointPlaces)pointPlacesArray[22]).point.Y -
((PointPlaces)pointPlacesArray[49]).point.Y)) = 87.02 pixel
```

1 cm = 87.02 / 4 cm = 21.76 pixel
1 cm = 21.76 pixel

- Eye Width with ruler is 3 cm

```
faceFeaturesSize.sagGozGenisligi =
Math.Sqrt((((PointPlaces)pointPlacesArray[23]).point.X -
((PointPlaces)pointPlacesArray[24]).point.X) *
(((PointPlaces)pointPlacesArray[23]).point.X -
((PointPlaces)pointPlacesArray[24]).point.X) +
(((PointPlaces)pointPlacesArray[23]).point.Y -
((PointPlaces)pointPlacesArray[24]).point.Y) *
(((PointPlaces)pointPlacesArray[23]).point.Y -
((PointPlaces)pointPlacesArray[24]).point.Y)) = 67.12 pixel
```

1 cm = 67.12 / 3 cm = 22.37 pixel
1 cm = 22.37 pixel

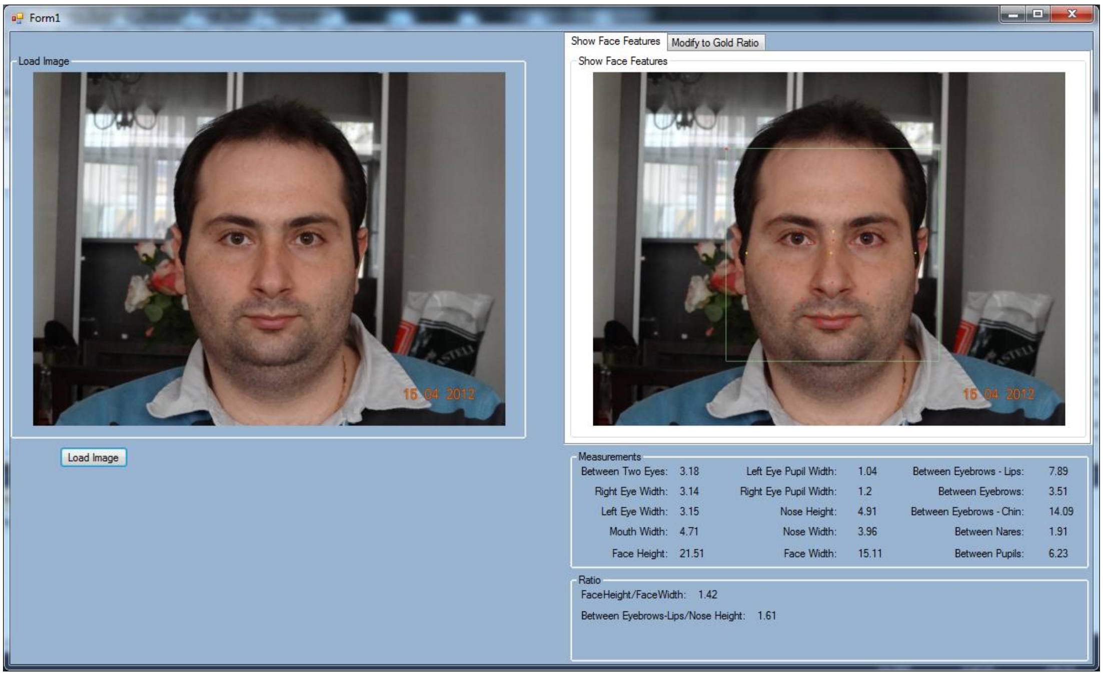
- Between Two Eyes with ruler 3 cm

```
faceFeaturesSize.ikiGozArasi =
Math.Sqrt((((PointPlaces)pointPlacesArray[24]).point.X -
((PointPlaces)pointPlacesArray[25]).point.X) *
(((PointPlaces)pointPlacesArray[24]).point.X -
((PointPlaces)pointPlacesArray[25]).point.X) +
(((PointPlaces)pointPlacesArray[24]).point.Y -
((PointPlaces)pointPlacesArray[25]).point.Y) *
(((PointPlaces)pointPlacesArray[24]).point.Y -
((PointPlaces)pointPlacesArray[25]).point.Y))  = 84.02 pixel
```

1 cm = 84.02 / 3 cm = 28.01 pixel.
1 cm = 28.01 pixel.

As you will see that in each features of face I have found different size for one cm to pixel. It may be worked on and find the more realistic size on it.
So I have decided to take average of this different size.
1 cm = (28.01 + 22.37 + 21.76 + 24.00 + 23.14 + 26.08) / = 24.226 pixel
So I have decided to take 1 cm = 24.23 pixel as real measurement.
So, this is may weakness of work.

Please find and screenshot from program calculate the real size of face features and calculate gold ratio from face features and give results.



Measurements

Between Two Eyes:	3.18	Left Eye Pupil Width:	1.04	Between Eyebrows - Lips:	7.89
Right Eye Width:	3.14	Right Eye Pupil Width:	1.2	Between Eyebrows:	3.51
Left Eye Width:	3.15	Nose Height:	4.91	Between Eyebrows - Chin:	14.09
Mouth Width:	4.71	Nose Width:	3.96	Between Nares:	1.91
Face Height:	21.51	Face Width:	15.11	Between Pupils:	6.23

Ratio

FaceHeight/FaceWidth: 1.42

Between Eyebrows-Lips/Nose Height: 1.61

Figure 11 - Simulate Golden Ratio on Face

The facial parts are measured and then calculated according to the ideal face size which is obtained from Golden ratio.

The other part of the thesis is studied on changes in modify to gold ratio tab, to realize these changes with image tools. When the image is loaded, the system formalize the face as a whole by using Golden ratio. After that the changed face with its last type is obtained, when any image tools is used.

Then the wanted facial part is changed. It can be changed manually.

The wanted part, for example this can be a nose, can be enlarged or minimized. The dimension of the brush can be arranged. This enlarging operation is realized with Grow button. The Shrink button is used for minimizing operation. The Translate button is used for changing the place of facial part. For example a nose can be low and the distance between mouth and tip of the nose is closer to each other than the standards defined in Golden ratio. The main program written in C# makes the operations and gives how much it is closer to the Golden ratio. Then the doctor can change whatever he/she wants with following below measurement with knowledge of gold ratio.

CONCLUSION

It is accepted that beauty changes for each individual because of being a relative concept. However there are some deterministic properties which indicate differences according to culture or race of people. Moreover, the studies that have been made up to here prove the apprehension of beauty is related to people's features of symmetry and phi in their proportions of the face.

Ratio and proportion are very important issues in aesthetic. The right proportion gives the perfect beauty. This is proven by scientists many many years ago. Also the Phi Number or Golden Ratio is the secret of equilibrium of beauty in the universe. The importance and necessity of Golden Ratio for catching the absolute beauty has been found and used since Ancient times. So it can be said that Golden Ratio is a scale of beauty and beauty attracts attention of everybody in the first look.

In this thesis, the measurement and calculations of creating a beautiful face are studied. For creating such a perfect face is possible using face symmetry and Golden ratio rules, which are explained in detail in the previous chapters. Such a popular subject needs improving itself. Moreover, the faces which are not suitable according to symmetry and Golden ratio can be arranged with some programs.
To satisfy the symmetry and Golden Ratio rules there is a group of issues that have to be paid attention. These are named as;

- Horizontal equal thirds (the first starts from the hairline and ends to eyebrows, the second begins from eyebrows until just under the nose, and the third starts from under the nose and ends with chin) below.

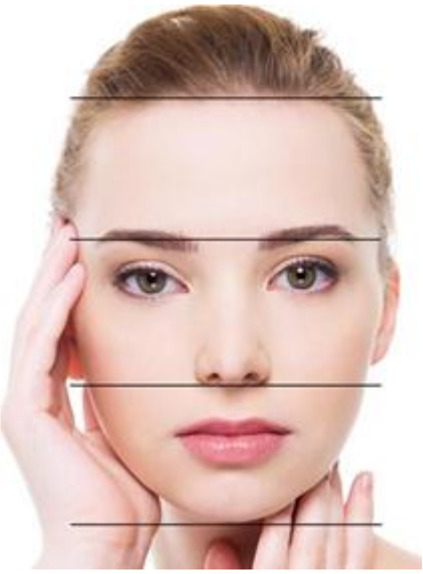


Figure 12 - Equal Third Horizantally

- It is also divided into equal fifths vertically, that starts from the upper tip of one ear to the upper tip of the other ear. Each fifth has wideness nearly as the width of one eye.

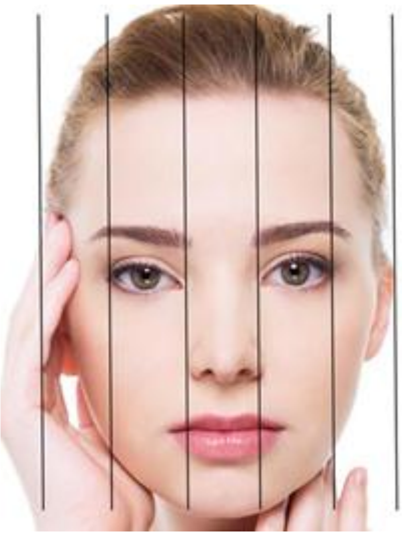


Figure 13 - Equal Fifth Vertically

An example of the difference between a symmetric and asymmetric face is shown below.

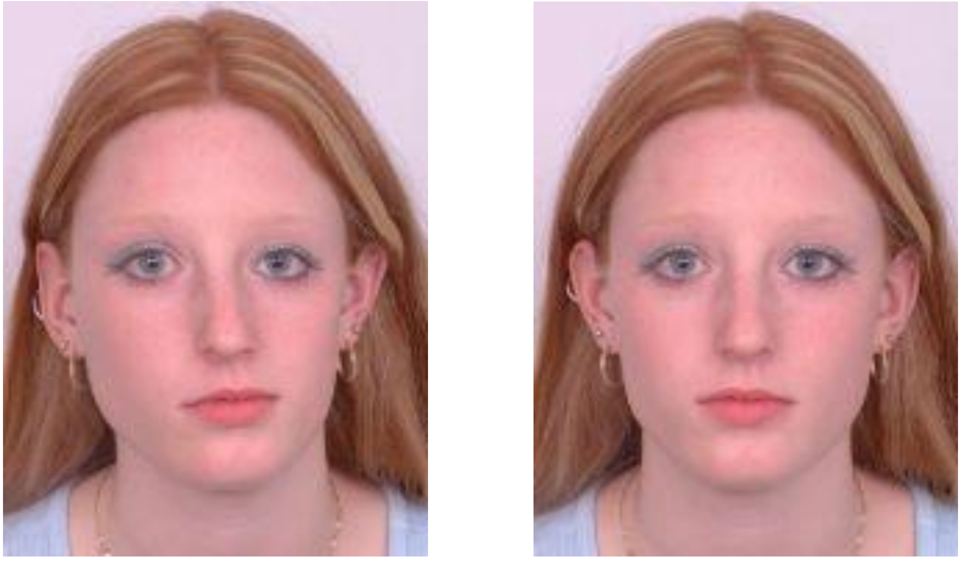


Figure 14 - Modified Asymmetry Face

The great differences between two faces need to develop a program to overcome this problem. In the thesis Luxand and C# are used to create such a program. This is realized in two ways. In the first way, after loading the facial image of a person, the ideal face size is calculated and when somebody wants to change one of his/her face part, the changing part is calculated again according to symmetry and Golden ratio rules. Then changed face type can be seen. After that the user can make any changes like growing, minimizing or changing its place. In the example below, the woman’s nose is calculated according to the Golden ratio and the result is in front of the eyes.

The second way is used for more complex situations. If the face is not suitable according to the symmetry and Golden ratio rules as a whole, this type of change will be more suitable. The facial image is loaded into the system, then measurements and calculations take place. After that the regenerated situation of the face according to the rules is created and shown. In this part of the work, there is Artificial Intelligence is also used. This type of change is used in deeply deformed faces like old people. An example is shown below.

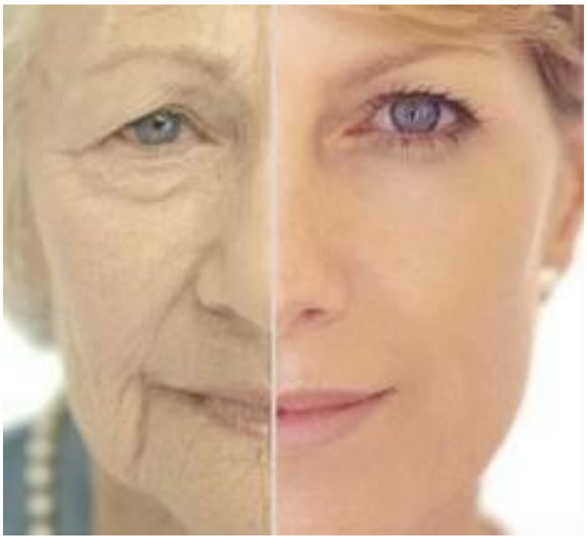


Figure 15 - Esthetic on All Face

The left part of the face is the natural face type but the right part represents the changed whole face. Moreover, the right part shows the situation which is done after nose esthetic, brow and eyebrow esthetic, chin esthetic, mouth and lips esthetic. The ratio and proportion of these plastic surgeries are done after the measurements and calculations according to symmetry and Golden Ratio rules that are defined in this thesis study. It also represents the change of all face.

REFERENCES

[1] Luxand - Download, Luxand Face SDK 4.0, http://www.luxand.com/download/, (Released: 22/07/2011)

[2] Ideal Beauty: Facial Analysis & Symmetry: Section 1, YTF Cosmetic Plastic Surgery Network, http://www.yestheyrefake.net/ideal_beauty.htm

[3] Measuring Facial Perfection, The Oprah, http://www.oprah.com/oprahshow/Measuring-Facial-Perfection-The-Golden-Ratio

[4] Lisa DeBruine, Ben Jones, Face Research, http://www.faceresearch.org/students/notes/symmetry.pdf
http://www.faceresearch.org/images/examples/sym/asym_female.jpg
http://www.faceresearch.org/images/examples/sym/sym_female.jpg
http://www.faceresearch.org/images/examples/sym/orig_female.jpg

[5] The Beauty of the Golden Ratio, ThinkQuest, http://library.thinkquest.org/trio/TTQ05063/phibeauty1.htm

[6] A Model's Secrets: The Perfect Face - Golden Ratio Beauty Calculator, A Model's Secrets by Shelley GoodStein, http://facethis.blogspot.com/2012/01/perfect-face-golden-ratio-beauty.html

[7] Golden Ratio, Wikipedia, http://en.wikipedia.org/wiki/Golden_ratio

[8] Human Face And Phi, the Golden Ratio, The Golden Number,
http://goldennumber.net/face

[9] Does Phi make a beatiful face, NYC Cosmetics Surgeons, http://nyccosmeticsurg.blogspot.com/2011/05/does-phi-make-beautiful-face.html

[10] Transforming Images for Fun: A Local Grid-based Image Warper, Code Project, http://www.codeproject.com/Articles/182242/Transforming-Images-for-Fun-A-Local-Grid-based-Ima

[11] Advanced Skin Wisdom, http://www.advancedskinwisdom.com/wordpress/wp-content/uploads/2011/09/facial-proportions-vertical-and-horizontal-cropped.jpg

[12] Prof Dr. Erol Kışlalı, Yüz Germe Estetik Cerrahi, http://www.estetikcerrahi.com/estetik/6-YuZ-GERME.html

[13] Samir E. Bishara, BDS, DDS, D. Ortho., MS, Gregory J. Jorgensen, BS, MS, and Jane R. Jakobsen, BS, MA, Changes in facial dimensions assessed from lateral and frontal photographs. Part I-Methodology, Iowa City, Iowa, and Rio Rancho, N.M., http://www.sciencedirect.com/science/article/pii/S0889540695700366

[14] E. Gladilin, S. Zachow, P. Deuflhard, H.C. Hege, Anatomy- and physics-based facial animation for craniofacial surgery simulations, Berlin, Germany, http://link.springer.com/article/10.1007%2FBF02344627?LI=true

[15] Fei Yang, Eli Shechtman, Jue Wang, Lubomir Bourdev, Dimitris Metaxas, Face Morphing using 3D-Aware Appearance Optimization, Rutgers University, Adobe Systems, http://www.research.rutgers.edu/~feiyang/paper/Feiyang_GI2012.pdf

[16] Volker Blanz, Thomas Vetter, A Morphable Model For The Synthesis Of 3D Faces, Tübingen, Germany, http://dl.acm.org/citation.cfm?id=311556

[17] Lisa M. Donofrio, MD, Fat Distribution: a Morphologic Study of the Aging Face, Department of Dermatology, Yale University School of Medicine, New Haven, Connectiut, http://www.ncbi.nlm.nih.gov/pubmed/11134986

[18] Andreas Lanitis, Member, IEEE, Chris J. Taylor. and Timothy F. Cootes, Toward Automatic Simulation of Aging Effects on Face Images, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.24, No.4, April 2002, http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=00993553

[19] Young Ho Kwont, and Niels da Vitoria Lobo, Age Classification from Facial Images, Dept. of Computer Science, University of Central Florida, Orlando, Florida 328 16, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=323894

[20] Age Progression , Photoshop Tutorials, http://www.exguides.org/photoshop-tutorials/age-progression.html

[21] A study on face morphing algorithms, EE368 Project, https://ccrma.stanford.edu/~jacobliu/368Report/index.html

[22] Online Plastic Surgery At Your Fingertips, Virtual Plastic Surgery for Nose Job, http://www.facetouchup.com/

[23] Aesthetic Imaging Software, Tools for Planning Aesthetic Surgical Treatments, Total Patient Imaging Technologies, http://digitale-photographie.info/mirror-software_en.asp

APPENDIX I - SOME PARTS OF CODES

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using Luxand;
using System.Collections;
namespace HumanFace_GoldRatio
{
    public partial class Form1 : Form
    {

        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            if (FSDK.FSDKE_OK != 
FSDK.ActivateLibrary("NEhE0W8svetqEJTww/8OqRIZYPYOjYHdRZ8tfNg/R+/9hg6CRGxN0DZV
Xdofuqdw9IIqkzNFRsZ8ZGL5KaFAMDX0mTgKJniR5OTISntYI2eCh0ArdvLMPUUxYLh4v96rJMSogd
zxiwrFNtTfqMDaSDxYLvx74asbWhFmMpjYZNE="))
            {
                MessageBox.Show("Please run the License Key Wizard (Start - 
Luxand - FaceSDK - License Key Wizard)", "Error activating FaceSDK", 
MessageBoxButtons.OK, MessageBoxIcon.Error);
                System.Windows.Forms.Application.Exit();
            }


            FSDK.InitializeLibrary();
            FSDK.SetFaceDetectionParameters(false, true, 384);
        }

        public string imageLocation = null;
        ArrayList pointsArray = new ArrayList();
        ArrayList pointPlacesArray = new ArrayList();
        FaceFeaturesSize faceFeaturesSize = new FaceFeaturesSize();
        public double onepixel = 24.23;
        FSDK.TFacePosition facePosition;

        private void btnLoadImage_Click(object sender, EventArgs e)
        {
            OpenFileDialog openFile = new OpenFileDialog();
            if (openFile.ShowDialog() == DialogResult.OK)
            {
                FSDK.CImage image = new FSDK.CImage(openFile.FileName);
Image frameImage = image.ToCLRImage();
                inputpictureBox.Image = frameImage;
                inputpictureBox.Refresh();
                imageLocation = openFile.FileName;
            }
            ShowFaceFeatures();
            calculateFaceFeatures(facePosition);
            calculateGoldRatio();
        }

        private void ShowFaceFeatures()
        {

            pointsArray.Clear();
            pointPlacesArray.Clear();

            FSDK.CImage image = new FSDK.CImage(imageLocation);
            Image frameImage = image.ToCLRImage();
            Graphics gr = Graphics.FromImage(frameImage);
            facePosition = image.DetectFace();
            if (0 == facePosition.w)
                MessageBox.Show("No faces detected", "Face Detection");
            else
            {
                int left = facePosition.xc - facePosition.w / 2;
                int top = facePosition.yc - facePosition.w / 2;
                gr.DrawRectangle(Pens.LightGreen, left, top, facePosition.w
+10, facePosition.w +10);
                gr.DrawRectangle(Pens.Red, left, top, 3, 3);
                gr.DrawRectangle(Pens.Yellow, facePosition.xc,
facePosition.yc, 3, 3);
                gr.DrawRectangle(Pens.Yellow, facePosition.xc-
(facePosition.xc-left)*81/100, facePosition.yc, 3, 3);
                gr.DrawRectangle(Pens.Yellow, facePosition.xc +
(facePosition.xc - left) * 81 / 100, facePosition.yc, 3, 3);


                FSDK.TPoint[] facialFeatures =
image.DetectFacialFeaturesInRegion(ref facePosition);

                int i = 0;
                foreach (FSDK.TPoint point in facialFeatures)
                    gr.DrawEllipse((++i > 2) ? Pens.Red : Pens.Blue, point.x,
point.y, 3, 3);

                findPlaces(facialFeatures);
                gr.DrawRectangle(Pens.Yellow,
((PointPlaces)pointPlacesArray[22]).point.X,
(int)(((PointPlaces)pointPlacesArray[22]).point.Y * 0.95), 3, 3);
                gr.Flush();

           }

            // display image
            outputpictureBox.Image = frameImage;
            outputpictureBox.Refresh();

        }
```

```
/////////////////////////////////////////////////////CalculateFaceFeatures_Sta
rt/////////////////////////////////////////////////////
        public void calculateFaceFeatures(FSDK.TFacePosition facePosition)
        {
            faceFeaturesSize.ikiGozArasi =
Math.Sqrt((((PointPlaces)pointPlacesArray[24]).point.X -
((PointPlaces)pointPlacesArray[25]).point.X) *
(((PointPlaces)pointPlacesArray[24]).point.X -
((PointPlaces)pointPlacesArray[25]).point.X) +

(((PointPlaces)pointPlacesArray[24]).point.Y -
((PointPlaces)pointPlacesArray[25]).point.Y) *
(((PointPlaces)pointPlacesArray[24]).point.Y -
((PointPlaces)pointPlacesArray[25]).point.Y)) / onepixel;

            faceFeaturesSize.sagGozGenisligi =
Math.Sqrt((((PointPlaces)pointPlacesArray[23]).point.X -
((PointPlaces)pointPlacesArray[24]).point.X) *
(((PointPlaces)pointPlacesArray[23]).point.X -
((PointPlaces)pointPlacesArray[24]).point.X) +

(((PointPlaces)pointPlacesArray[23]).point.Y -
((PointPlaces)pointPlacesArray[24]).point.Y) *
(((PointPlaces)pointPlacesArray[23]).point.Y -
((PointPlaces)pointPlacesArray[24]).point.Y)) / onepixel;

            faceFeaturesSize.solGozGenisligi =
Math.Sqrt((((PointPlaces)pointPlacesArray[25]).point.X -
((PointPlaces)pointPlacesArray[26]).point.X) *
(((PointPlaces)pointPlacesArray[25]).point.X -
((PointPlaces)pointPlacesArray[26]).point.X) +

(((PointPlaces)pointPlacesArray[25]).point.Y -
((PointPlaces)pointPlacesArray[26]).point.Y) *
(((PointPlaces)pointPlacesArray[25]).point.Y -
((PointPlaces)pointPlacesArray[26]).point.Y)) / onepixel;

            faceFeaturesSize.agizGenisligi =
Math.Sqrt((((PointPlaces)pointPlacesArray[3]).point.X -
((PointPlaces)pointPlacesArray[4]).point.X) *
(((PointPlaces)pointPlacesArray[3]).point.X -
((PointPlaces)pointPlacesArray[4]).point.X) +

(((PointPlaces)pointPlacesArray[3]).point.Y -
((PointPlaces)pointPlacesArray[4]).point.Y) *
(((PointPlaces)pointPlacesArray[3]).point.Y -
((PointPlaces)pointPlacesArray[4]).point.Y)) / onepixel;

            faceFeaturesSize.solGozBebegiGenisligi =
Math.Sqrt((((PointPlaces)pointPlacesArray[33]).point.X -
((PointPlaces)pointPlacesArray[34]).point.X) *
(((PointPlaces)pointPlacesArray[33]).point.X -
((PointPlaces)pointPlacesArray[34]).point.X) +

(((PointPlaces)pointPlacesArray[33]).point.Y -
((PointPlaces)pointPlacesArray[34]).point.Y) *
(((PointPlaces)pointPlacesArray[33]).point.Y -
((PointPlaces)pointPlacesArray[34]).point.Y)) / onepixel;
```

```
            faceFeaturesSize.sagGozBebegiGenisligi =
Math.Sqrt((((PointPlaces)pointPlacesArray[29]).point.X -
((PointPlaces)pointPlacesArray[30]).point.X) *
(((PointPlaces)pointPlacesArray[29]).point.X -
((PointPlaces)pointPlacesArray[30]).point.X) +

(((PointPlaces)pointPlacesArray[29]).point.Y -
((PointPlaces)pointPlacesArray[30]).point.Y) *
(((PointPlaces)pointPlacesArray[29]).point.Y -
((PointPlaces)pointPlacesArray[30]).point.Y)) / onepixel;

            faceFeaturesSize.noseHeight =
Math.Sqrt((((PointPlaces)pointPlacesArray[22]).point.X -
((PointPlaces)pointPlacesArray[49]).point.X) *
(((PointPlaces)pointPlacesArray[22]).point.X -
((PointPlaces)pointPlacesArray[49]).point.X) +

(((PointPlaces)pointPlacesArray[22]).point.Y -
((PointPlaces)pointPlacesArray[49]).point.Y) *
(((PointPlaces)pointPlacesArray[22]).point.Y -
((PointPlaces)pointPlacesArray[49]).point.Y)) / onepixel;

            faceFeaturesSize.noseWidth =
Math.Sqrt((((PointPlaces)pointPlacesArray[45]).point.X -
((PointPlaces)pointPlacesArray[46]).point.X) *
(((PointPlaces)pointPlacesArray[45]).point.X -
((PointPlaces)pointPlacesArray[46]).point.X) +

(((PointPlaces)pointPlacesArray[45]).point.Y -
((PointPlaces)pointPlacesArray[46]).point.Y) *
(((PointPlaces)pointPlacesArray[45]).point.Y -
((PointPlaces)pointPlacesArray[46]).point.Y)) / onepixel;

            faceFeaturesSize.faceHeight = Math.Sqrt((facePosition.xc -
((PointPlaces)pointPlacesArray[11]).point.X) * (facePosition.xc -
((PointPlaces)pointPlacesArray[11]).point.X) +
                                                 (facePosition.yc -
facePosition.w / 2 - ((PointPlaces)pointPlacesArray[11]).point.Y) *
(facePosition.yc - facePosition.w / 2 -
((PointPlaces)pointPlacesArray[11]).point.Y)) / onepixel;

            faceFeaturesSize.betweenPupils =
Math.Sqrt((((PointPlaces)pointPlacesArray[0]).point.X -
((PointPlaces)pointPlacesArray[1]).point.X) *
(((PointPlaces)pointPlacesArray[0]).point.X -
((PointPlaces)pointPlacesArray[1]).point.X) +

(((PointPlaces)pointPlacesArray[0]).point.Y -
((PointPlaces)pointPlacesArray[1]).point.Y) *
(((PointPlaces)pointPlacesArray[0]).point.Y -
((PointPlaces)pointPlacesArray[1]).point.Y)) / onepixel;

            faceFeaturesSize.betweenNares =
Math.Sqrt((((PointPlaces)pointPlacesArray[47]).point.X -
((PointPlaces)pointPlacesArray[48]).point.X) *
(((PointPlaces)pointPlacesArray[47]).point.X -
((PointPlaces)pointPlacesArray[48]).point.X) +

(((PointPlaces)pointPlacesArray[47]).point.Y -
((PointPlaces)pointPlacesArray[48]).point.Y) *
(((PointPlaces)pointPlacesArray[47]).point.Y -
((PointPlaces)pointPlacesArray[48]).point.Y)) / onepixel;

            faceFeaturesSize.betweenEyeBrowsChin =
Math.Sqrt((((PointPlaces)pointPlacesArray[22]).point.X -
((PointPlaces)pointPlacesArray[11]).point.X) *
(((PointPlaces)pointPlacesArray[22]).point.X -
((PointPlaces)pointPlacesArray[11]).point.X) +

(((PointPlaces)pointPlacesArray[22]).point.Y * 0.95 -
((PointPlaces)pointPlacesArray[11]).point.Y) *
(((PointPlaces)pointPlacesArray[22]).point.Y * 0.95 -
((PointPlaces)pointPlacesArray[11]).point.Y)) / onepixel;

            faceFeaturesSize.betweenEyeBrows =
Math.Sqrt((((PointPlaces)pointPlacesArray[13]).point.X -
((PointPlaces)pointPlacesArray[14]).point.X) *
(((PointPlaces)pointPlacesArray[13]).point.X -
((PointPlaces)pointPlacesArray[14]).point.X) +

(((PointPlaces)pointPlacesArray[13]).point.Y -
((PointPlaces)pointPlacesArray[14]).point.Y) *
(((PointPlaces)pointPlacesArray[13]).point.Y -
((PointPlaces)pointPlacesArray[14]).point.Y)) / onepixel;

            faceFeaturesSize.betweenEyeBrowsLips =
Math.Sqrt((((PointPlaces)pointPlacesArray[22]).point.X -
((PointPlaces)pointPlacesArray[61]).point.X) *
(((PointPlaces)pointPlacesArray[22]).point.X -
((PointPlaces)pointPlacesArray[61]).point.X) +

(((PointPlaces)pointPlacesArray[22]).point.Y*0.95  -
((PointPlaces)pointPlacesArray[61]).point.Y) *
(((PointPlaces)pointPlacesArray[22]).point.Y*0.95 -
((PointPlaces)pointPlacesArray[61]).point.Y)) / onepixel;
            faceFeaturesSize.faceWidth = (2 * (facePosition.w / 2) * 81 / 100)
/ onepixel;

            lblIkiGozArasi.Text =
faceFeaturesSize.ikiGozArasi.ToString("0.##");
            lblRightEyeWidth.Text =
faceFeaturesSize.sagGozGenisligi.ToString("0.##");
            lblLeftEyeWidth.Text =
faceFeaturesSize.solGozGenisligi.ToString("0.##");
            lblMouthWidth.Text =
faceFeaturesSize.agizGenisligi.ToString("0.##");
            lblLeftEyePupilWidth.Text =
faceFeaturesSize.solGozBebegiGenisligi.ToString("0.##");
            lblRightEyePupilWidth.Text =
faceFeaturesSize.sagGozBebegiGenisligi.ToString("0.##");
            lblNoseHeight.Text = faceFeaturesSize.noseHeight.ToString("0.##");
            lblNoseWidth.Text = faceFeaturesSize.noseWidth.ToString("0.##");
            lblFaceHeight.Text = faceFeaturesSize.faceHeight.ToString("0.##");
            lblBetweenPupils.Text =
faceFeaturesSize.betweenPupils.ToString("0.##");
            lblBetweenNares.Text =
faceFeaturesSize.betweenNares.ToString("0.##");
            lblBetweenEyeBrowsChin.Text =
faceFeaturesSize.betweenEyeBrowsChin.ToString("0.##");
```

```
            lblBetweenEyeBrows.Text = 
faceFeaturesSize.betweenEyeBrows.ToString("0.##");
            lblbetEyeBrows_Lips.Text = 
faceFeaturesSize.betweenEyeBrowsLips.ToString("0.##");
            lblFaceWidth.Text = faceFeaturesSize.faceWidth.ToString("0.##");
        }
        public void calculateGoldRatio()
        {
            lblFaceHeight_FaceWidth.Text = (faceFeaturesSize.faceHeight / 
faceFeaturesSize.faceWidth).ToString("0.##");
            lblbetweenEyebrowsLips_NoseHeight.Text = 
(faceFeaturesSize.betweenEyeBrowsLips / 
faceFeaturesSize.noseHeight).ToString("0.##");
        }

/////////////////////////////////////////////////////CalculateFaceFeatures_End
/////////////////////////////////////////////////////


//////////////////////////////////CLASSES_START///////////////////////////////
///////////////////////////////////////////////
        public class FaceFeaturesSize
        {
            public double ikiGozArasi;
            public double sagGozGenisligi;
            public double solGozGenisligi;
            public double agizGenisligi;
            public double solGozBebegiGenisligi;
            public double sagGozBebegiGenisligi;
            public double noseHeight;
            public double noseWidth;
            public double faceHeight;
            public double betweenPupils;
            public double betweenNares;
            public double betweenEyeBrowsChin;
            public double betweenEyeBrows;
            public double betweenEyeBrowsLips;
            public double faceWidth;

        }

        public class PointPlaces
        {
            public string place;
            public Point point = new Point();
            public PointPlaces(string place, Point point)
            {
                this.place = place;
                this.point = point;
            }

        }
//////////////////////////////////CLASSES_END/////////////////////////////////
////////////////////////////////////////////////

//////////////////////////////////////////////Find_Places_Start///////////////
///////////////////////////////////////////////////
        public void findPlaces(FSDK.TPoint[] facialFeatures)
        {
            for (int j = 0; j < facialFeatures.Length; j++)
            {
                pointsArray.Add(new Point(facialFeatures[j].x, 
facialFeatures[j].y));
            }


            for (int k = 0; k < pointsArray.Count; k++)
            {
                Point point = (Point)pointsArray[k];

                if (k == 0)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağ göz bebeği", 
point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ göz 
bebeği\r\n";
                }
                else if (k == 1)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol göz bebeği", 
point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol göz 
bebeği\r\n";
                }
                else if (k == 2)
                {
                    pointPlacesArray.Add(new PointPlaces("Burun üst ucu", 
point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Burun 
üst ucu\r\n";
                }
                else if (k == 3)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağdan ağız 
başlangıcı 1", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağdan 
ağız başlangıcı 1\r\n";
                }
                else if (k == 4)
                {
                    pointPlacesArray.Add(new PointPlaces("Soldan ağız 
başlangıcı 1", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Soldan 
ağız başlangıcı 1\r\n";
                }

                else if (k == 5)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağdan çene 
başlangıcı 1", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağdan 
çene başlangıcı 1\r\n";
                }
                else if (k == 6)
                {
                    pointPlacesArray.Add(new PointPlaces("Soldan çene 
başlangıcı 1", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Soldan 
çene başlangıcı 1\r\n";
                }
                else if (k == 7)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağdan çene
başlangıcı 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağdan
çene başlangıcı 2\r\n";
                }
                else if (k == 8)
                {
                    pointPlacesArray.Add(new PointPlaces("Soldan çene
başlangıcı 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Soldan
çene başlangıcı 2\r\n";
                }
                else if (k == 9)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağdan çene
başlangıcı 3", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağdan
çene başlangıcı 3\r\n";
                }
                else if (k == 10)
                {
                    pointPlacesArray.Add(new PointPlaces("Soldan çene
başlangıcı 3", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Soldan
çene başlangıcı 3\r\n";
                }
                else if (k == 11)
                {
                    pointPlacesArray.Add(new PointPlaces("Çene Ucu", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Çene
Ucu\r\n";
                }
                else if (k == 12)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağ kaş başlangıcı
sağdan 1", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ kaş
başlangıcı sağdan 1\r\n";
                }
                else if (k == 13)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağ kaş başlangıcı
sağdan 5", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ kaş
başlangıcı sağdan 5\r\n";
                }
                else if (k == 14)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol kaş başlangıcı
soldan 5", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol kaş
başlangıcı soldan 5\r\n";
                }
                else if (k == 15)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol kaş başlangıcı
soldan 1", point));
```

```
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol kaş
başlangıcı soldan 1\r\n";
                }

                else if (k == 16)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağ kaş başlangıcı
sağdan 3", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ kaş
başlangıcı sağdan 3\r\n";
                }

                else if (k == 17)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol kaş başlangıcı
soldan 3", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol kaş
başlangıcı soldan 3\r\n";
                }

                else if (k == 18)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağ kaş başlangıcı
sağdan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ kaş
başlangıcı sağdan 2\r\n";
                }

                else if (k == 19)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağ kaş başlangıcı
sağdan 4", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ kaş
başlangıcı sağdan 4\r\n";
                }

                else if (k == 20)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol kaş başlangıcı
soldan 4", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol kaş
başlangıcı soldan 4\r\n";
                }

                else if (k == 21)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol kaş başlangıcı
soldan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol kaş
başlangıcı soldan 2\r\n";
                }

                else if (k == 22)
                {
                    pointPlacesArray.Add(new PointPlaces("Burun Başlangıcı",
point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Burun
Başlangıcı\r\n";
                }
```

```
        else if (k == 23)
        {
          pointPlacesArray.Add(new PointPlaces("Sağ göz başlangıcı
sağdan 1", point));
          //textBox1.Text += point.X + " " + point.Y + "  - Sağ göz
bebeği\r\n";
        }

        else if (k == 24)
        {
          pointPlacesArray.Add(new PointPlaces("Sağ göz başlangıcı
soldan 1", point));
          //textBox1.Text += point.X + " " + point.Y + "  - Sağ göz
başlangıcı soldan 1\r\n";
        }

        else if (k == 25)
        {
          pointPlacesArray.Add(new PointPlaces("Sol göz başlangıcı
sağdan 1", point));
          //textBox1.Text += point.X + " " + point.Y + "  - Sol göz
başlangıcı sağdan 1\r\n";
        }

        else if (k == 26)
        {
          pointPlacesArray.Add(new PointPlaces("Sol göz başlangıcı
soldan 1", point));
          //textBox1.Text += point.X + " " + point.Y + "  - Sol göz
başlangıcı soldan 1\r\n";
        }

        else if (k == 27)
        {
          pointPlacesArray.Add(new PointPlaces("Sağ göz başlangıcı
alttan orta", point));
          //textBox1.Text += point.X + " " + point.Y + "  - Sağ göz
başlangıcı alttan orta\r\n";
        }

        else if (k == 28)
        {
          pointPlacesArray.Add(new PointPlaces("Sağ göz başlangıcı
üstten orta", point));
          //textBox1.Text += point.X + " " + point.Y + "  - Sağ göz
başlangıcı üstten orta\r\n";
        }

        else if (k == 29)
        {
          pointPlacesArray.Add(new PointPlaces("Sağ göz başlangıcı
ortadan sağdan 2", point));
          //textBox1.Text += point.X + " " + point.Y + "  - Sağ göz
başlangıcı ortadan sağdan 2\r\n";
        }

        else if (k == 30)
        {
```

```
                    pointPlacesArray.Add(new PointPlaces("Sağ göz başlangıcı
ortadan soldan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ göz
başlangıcı ortadan soldan 2\r\n";
                }

                else if (k == 31)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol göz başlangıcı
alttan orta", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol göz
başlangıcı alttan orta\r\n";
                }

                else if (k == 32)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol göz başlangıcı
üstten orta", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol göz
başlangıcı üstten orta\r\n";
                }

                else if (k == 33)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol göz başlangıcı
ortadan sağdan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol göz
başlangıcı ortadan sağdan 2\r\n";
                }

                else if (k == 34)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol göz başlangıcı
ortadan soldan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol göz
başlangıcı ortadan soldan 2\r\n";
                }

                else if (k == 35)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağ göz başlangıcı
üstten sağdan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ göz
başlangıcı üstten sağdan 2\r\n";
                }

                else if (k == 36)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağ göz başlangıcı
üstten soldan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ göz
başlangıcı üstten soldan 2\r\n";
                }

                else if (k == 37)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağ göz başlangıcı
alttan sağdan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ göz
başlangıcı alttan sağdan 2\r\n";
```

```
                }

                else if (k == 38)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağ göz başlangıcı
alttan soldan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ göz
başlangıcı alttan soldan 2\r\n";
                }

                else if (k == 39)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol göz başlangıcı
üstten sağdan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol göz
başlangıcı üstten sağdan 2\r\n";
                }

                else if (k == 40)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol göz başlangıcı
üstten soldan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol göz
başlangıcı üstten soldan 2\r\n";
                }

                else if (k == 41)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol göz başlangıcı
alttan sağdan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol göz
başlangıcı alttan sağdan 2\r\n";
                }

                else if (k == 42)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol göz başlangıcı
alttan sağdan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol göz
başlangıcı alttan sağdan 2\r\n";
                }

                else if (k == 43)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağ burun üstü",
point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ
burun üstü\r\n";
                }

                else if (k == 44)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol burun üstü",
point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol
burun üstü\r\n";
                }

                else if (k == 45)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağ burun
başlangıcı", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ
burun başlangıcı\r\n";
                }

                else if (k == 46)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol burun
başlangıcı", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol
burun başlangıcı\r\n";
                }

                else if (k == 47)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağ burun deliği",
point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ
burun deliği\r\n";
                }

                else if (k == 48)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol burun deliği",
point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol
burun deliği\r\n";
                }

                else if (k == 49)
                {
                    pointPlacesArray.Add(new PointPlaces("Burun alt ucu",
point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Burun
alt ucu\r\n";
                }

                else if (k == 50)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağ burun altı
üstten 1", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ
burun altı üstten 1\r\n";
                }

                else if (k == 51)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol burun altı
üstten 1", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol
burun altı üstten 1\r\n";
                }

                else if (k == 52)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağ burun altı
üstten 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağ
burun altı üstten 2\r\n";
```

```
                }

                else if (k == 53)
                {
                    pointPlacesArray.Add(new PointPlaces("Sol burun altı
üssten 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sol
burun altı üssten 2\r\n";
                }

                else if (k == 54)
                {
                    pointPlacesArray.Add(new PointPlaces("Üst dudak üst orta",
point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Üst
dudak üst orta\r\n";
                }

                else if (k == 55)
                {
                    pointPlacesArray.Add(new PointPlaces("Alt Dudak alt orta",
point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Alt
Dudak alt orta\r\n";
                }

                else if (k == 56)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağdan ağız
başlangıcı üst dudak üstten 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağdan
ağız başlangıcı üst dudak üstten 2\r\n";
                }

                else if (k == 57)
                {
                    pointPlacesArray.Add(new PointPlaces("Soldan ağız
başlangıcı üst dudak üstten 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Soldan
ağız başlangıcı üst dudak üstten 2\r\n";
                }

                else if (k == 58)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağdan ağız
başlangıcı alt dudak alttan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağdan
ağız başlangıcı alt dudak alttan 2\r\n";
                }

                else if (k == 59)
                {
                    pointPlacesArray.Add(new PointPlaces("Soldan ağız
başlangıcı alt dudak alttan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Soldan
ağız başlangıcı alt dudak alttan 2\r\n";
                }

                else if (k == 60)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağdan ağız 
başlangıcı üst dudak alttan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağdan 
ağız başlangıcı üst dudak alttan 2\r\n";
                }

                else if (k == 61)
                {
                    pointPlacesArray.Add(new PointPlaces("Üst Dudak alt orta", 
point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Üst 
Dudak alt orta\r\n";
                }

                else if (k == 62)
                {
                    pointPlacesArray.Add(new PointPlaces("Soldan ağız 
başlangıcı üst dudak alttan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Soldan 
ağız başlangıcı üst dudak alttan 2\r\n";
                }

                else if (k == 63)
                {
                    pointPlacesArray.Add(new PointPlaces("Sağdan ağız 
başlangıcı üst dudak üstten 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Sağdan 
ağız başlangıcı üst dudak üstten 2\r\n";
                }

                else if (k == 64)
                {
                    pointPlacesArray.Add(new PointPlaces("Alt dudak üst orta", 
point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Alt 
dudak üst orta\r\n";
                }

                else if (k == 65)
                {
                    //gr.DrawEllipse(Pens.Black, point.X, point.Y, 3, 3);
                    pointPlacesArray.Add(new PointPlaces("Soldan ağız 
başlangıcı alt dudak alttan 2", point));
                    //textBox1.Text += point.X + " " + point.Y + "  - Soldan 
ağız başlangıcı alt dudak alttan 2\r\n";
                }

            }
        }

        private void tabControl_SelectedIndexChanged(object sender, EventArgs 
e)
        {

        }
//////////////////////////////////////////////Find_Places_End/////////////////
/////////////////////////////////////////////////

    }
}
```

APPENDIX II - CV

PERSONAL INFORMATION

Surname, Name	**:**	**YAPICI, ZAFER**
Nationality	**:**	**T.C.**
Birth Date and Place	**:**	**1987 HATAY**
Marital Status	**:**	**SINGLE**
Phone	**:**	**0533 922 1121**
E-Mail	**:**	**zafer.yapici@hotmail.com**

EDUCATION

DEGREE	INSTITUTION	GRADUATION DATE
UNDERGRADUATE	Atılım University	2009
HIGH SCHOOL	Selim Nevzat Şahin Anatolian High School	2004

WORK EXPERIENCE

YIL	YER	POZİSYON
2009 - ...	Informascope Publishing Information Technologies	IT Systems Manager

FOREIGN LANGUAGE

English – Advanced Level

Arabic - In speaking Level