Triangle wise Mapping Technique to Transform one Face Image into Another Face Image

Rustam Ali Ahmed
Department Information Technology
Sikkim Manipal Institute of Technology
Majitar, Sikkim- 737136, INDIA

Bhogeswar Borah, Ph.D
Department of Computer Science and Engineering
Tezpur University
Napaam, Assam- 784028, INDIA

ABSTRACT

This paper describes a triangle based algorithm to transform a source image into a target image. In this paper, a digital source face image is mapped to a target face image to transform the shape of source into the target. This work is done with six steps. The initial step of the stated algorithm takes source and destination face images as input from the specified location or through the webcam. The second step deals with finding the 68 landmark points of input images for tracking the features such as eyes, mouth, nose, lips, ears and face. The third step generates proposed 116 nos of non-overlapping triangles from 68 landmark points (which has been found in the second step) for both the input face images. In forth step one mapping link is established between the each pair of corresponding proposed 116 triangles of both the images. Then these pairs of triangles are divided on the basis of given threshold value of the in-radius of the triangles pairs. In this step a set of smallest subtriangles are found in the last label for the triangle pairs. In fifth step each pair of smallest sub-triangles are mapped with pixels from source face image to destination face image and then generate intermediate image with color interpolation. The sixth step is the process of assembling the 116 nos of triangles to generate the resultant face image in the shape of target face image. The results show that the proposed approach is simple and takes less time to transform the source image.

Keywords:

Image morphing, warping, triangle wise mapping, mesh deformation, triangular mesh deformation

1. INTRODUCTION

Shape transforming of a given image into a target or template image is an important task in image processing. Image transformation has also proven to be a very important and powerful tool for visual effects in computer graphics and animation. Basically this process of image transforming is known as image morphing[22] and has received much attention in recent years. At present there are many interesting instances in film and television[13] [21] of the smooth transformation of one digital image into another. In general this method is called as morphing and is recognized by joining images by warping with color interpolation. In the process of image warping 2D geometric transformations are done with the images. In this process geometric alignment between the features of the input images are retained while color interpolation blends their colors.

The morphing sequence quality depends on the solution of three problems[24]: feature specification, warp generation and transition control. Choosing the correct correspondence between pairs of feature specification can be performed more precisely. In actual morphing algorithms, meshes[4] [8] [22], line segments[2] [10] [11], or points[1] [9] are used to determine feature positions in the images. In the process of morphing or image transformation some selective feature points such as points of different face parts e.g., eye, nose, mouth etc. of each input images are selected. These feature points are represented by some particular spatial co-ordinate points, we call these as landmark[16] points. So group of some nearer landmark points represents the different parts of the given images. Image transformation between two images begins with these pairs of landmark points. Then some mapping functions are defined as the spatial relationship between all the corresponding landmark points in both images. These relative mapping functions are used to interpolate the positions of the features across the transform sequence. After the mapping of intermediate feature positions are established both images have been warped into alignment for intermediate feature positions. Once both images have been warped into alignment for intermediate feature positions, ordinary color interpolation (cross-dissolve[22]) is performed to generate image morphing. The rate of warping and color blending across the morph sequence is determined by the transition control.

Urvashi[3] et al. recently implemented image morphing using mesh warping algorithm combined with the cross dissolving technique. The algorithms used were fast and intuitive, which efficiently computed the mapping of each pixel from the source image to the destination image. The mesh was formed of triangles obtained from the manually specified control points.

Y Guo[7] et al. have presented an efficient algorithm in their paper *"Efficient view manipulation for cuboid-structured images"* for transforming the viewpoints of cuboid-structured images. In this algorithm the new image with high visual quality is generated by making the rest image region deform in accordance with the re-projected cuboid structure, via a triangular mesh deformation scheme and demonstrated an application of upright adjustment of photographs and a user interface which enables the user to watch the scene under new viewpoints on a viewing sphere interactively.

Feature specification is the most deadly aspect of morphing. In the maximum work in this regard the landmark points of an image are determined physically. A way to determine the landmarks automat-

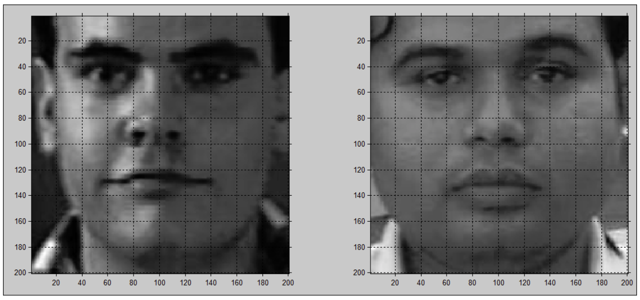


Fig. 1. Aligned Image Pair

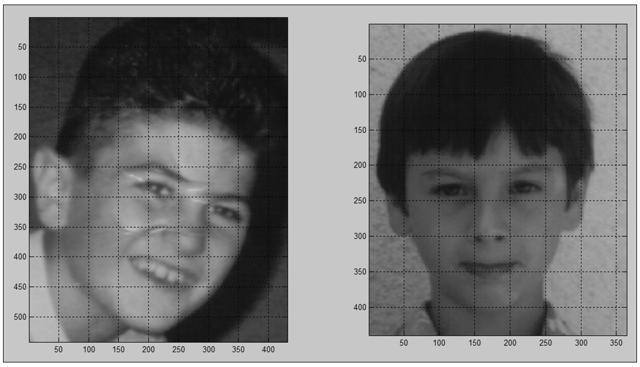


Fig. 2. Un-Aligned Image Pair

ically, without the participation of a human, would be desirable. P. Perakis and G. Passalis[16] have proposed a 3D facial landmark detection method under large yaw and expression variations. In that method the main contribution is the automatic and pose-invariant detection of landmarks on 3D facial scans under large yaw variations (that often result in missing facial data), and its robustness against large facial expressions. In this sense, we are also in the way to develop an efficient tool to solve this problem.
In this work, we have developed a triangle based algorithm to establish a spatial relationship between all points in both images and we call this as triangle based mapping between two images. We initially choose work with images of faces with uniform illumination, without glasses and facial hair to simplify the problem. And with a challenge we have chosen images with of any view not only in frontal view and also under pose and expression variations and without any restriction of shape and sizes.

2. PROBLEM DEFINITION

Given two objects S and D with known point-to-point correspondences, find a smooth isometric deformation from S to D.
Here each object S and D are represented by a fixed numbers of triangles.
Here in the Fig.1 a pair of aligned image is given where almost all the landmark points are in the same positions. But in the Fig.2 a pair of unaligned image is noticeable. In this figure the main face-part points of the face images are not in the same positions and the sizes of the images are also not same. The co-ordinate points that represent the various parts of the face image we call the landmark point of the face image. It is noticeable that the mapping of landmark point to landmark point is easy for the aligned pair of images, but for the unaligned pair of images mapping is not so easy.

(a) Pos- 0 (b) Pos- 08

Fig. 3. Unlike Posed Image Pair from ExtendedYaleB database

So our main objective is to map landmark points to landmark points of any pair of face images. It can be done by image transformation like rotating, shifting, scaling and shearing the images. But in some cases these techniques do not produce good results. In the Fig.3 two images are shown where the face appearances are different as for different poses (images are obtained from Extended YaleB database[6]). For this type of cases we are interested to transform one image into the other using our proposed triangle based mapping technique.

3. MAPPING OF FACE IMAGES

Mapping of face images of same person where poses are taken from different angle is an important task in image analysis techniques. In some cases like PCA algorithm we need face images where all images are having of unique angled face image. If the images are of different angled image then finding mean face in PCA algorithm will not give the correct result. So using our algorithm we can bring all required face images of any angled to a standard angled face image. Using this technique our output images can be used where same angled images are required.
We have proposed a technique to map one face image to another face image where images may be of any angled. Basically we have proposed a mapping technique which maps two triangle of any shape and establish a relation to the each pixel of one triangle to another. Then we apply this triangle based mapping to map the two face images by dividing the total faces images by our 116 numbers of proposed triangles. In the following section we have describe the triangle based mapping techniques.

4. TRIANGLE BASED MAPPING

We have implemented a triangle based mapping algorithm where one triangle is mapped with another one. Here both the triangle $T_1 = \Delta P_1 P_2 P_3$ and $T_2 = \Delta Q_1 Q_2 Q_3$ may be of any shape/type and our technique works for all the shapes/types.
We divide the first triangle into four sub triangles by the middle point of each side of original triangle. And applying same technique we get four equal corresponding sub triangles from second triangle also. Thus we get one to one mapping of four sub triangles of the given two main triangles in the first step. Now we give the name of these four sub triangles as *upper, left, right* and *middle* for both the main triangles. Thus we get four sub triangles $T_2^{upper}, T_2^{left}, T_2^{right}, T_2^{middle}$ of triangle ΔT_2 for corresponding sub triangles $T_1^{upper}, T_1^{left}, T_1^{right}, T_1^{middle}$ of triangle ΔT_1. Algorithm 3 named “Sub-triangle Algorithm” describes the procedure of dividing a triangle to a set of four sub-triangles.
Now we compute the in-radius of subtriangle T_2^{upper} of triangle T_2 and for corresponding subtriangle T_1^{upper} of triangle T_1 (Fig.4) and if either one is less than the given threshold value ϵ then

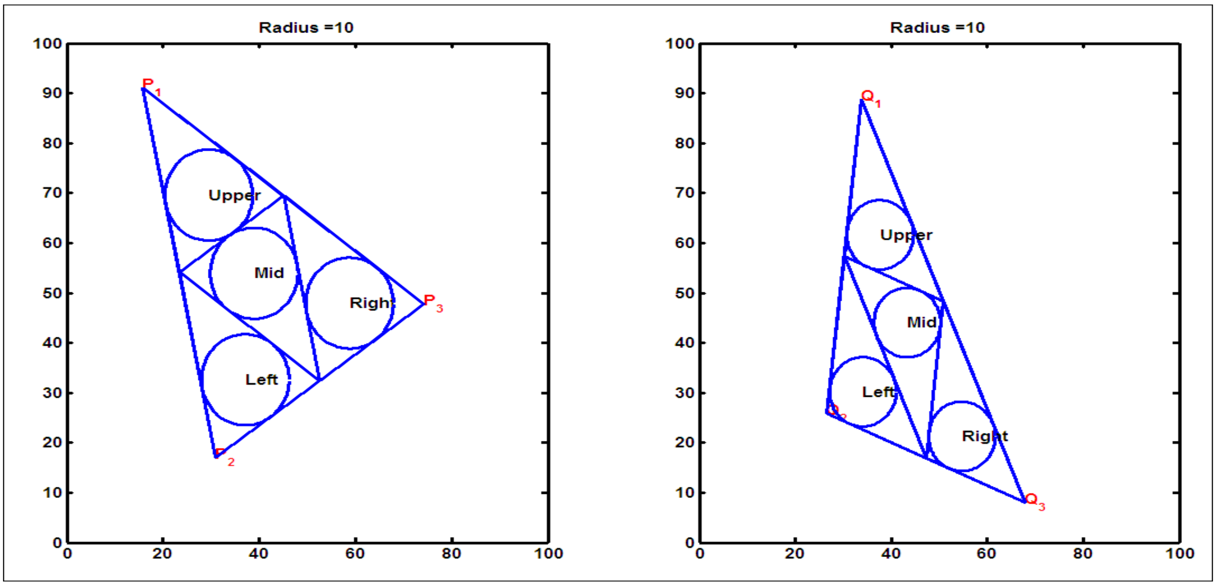
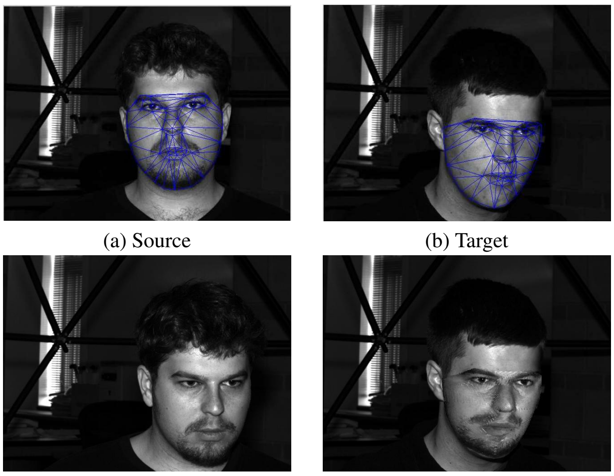


Fig. 4. Sub-triangles of two given triangles T_1 and T_2



(a) Source

(b) Target

(c) Original into the shape of target

(d) Output into the shape of target

Fig. 6. To transform the source image into the shape of target image

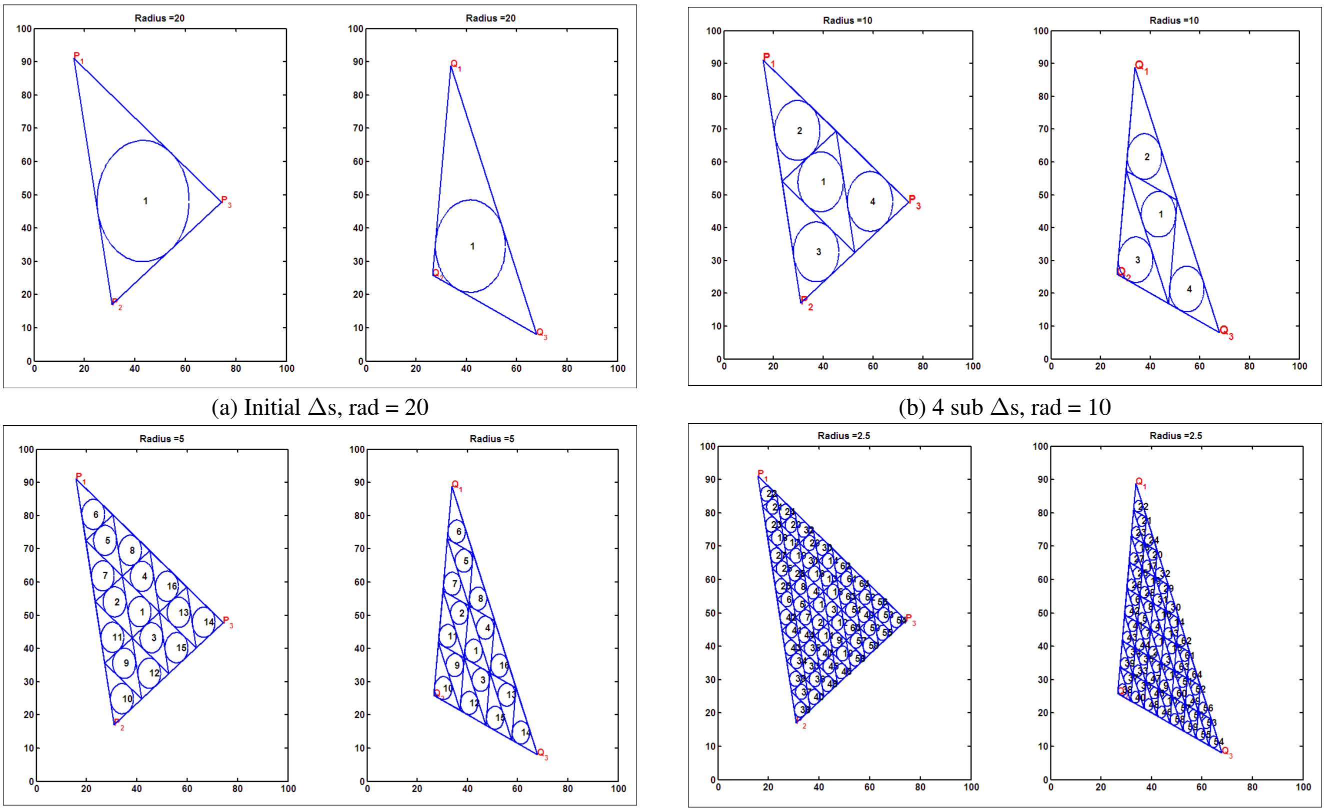
we record the one-one mapping point of the in-centre of T_2^{upper} and T_1^{upper}. Otherwise we consider T_2^{upper} and T_1^{upper} as our main triangles and repeat the same technique recursively. Similarly we apply this recursive technique for other pair of sub triangles (T_2^{left}, T_1^{left}), $(T_2^{right}, T_1^{right})$ and $(T_2^{middle}, T_1^{middle})$. And finally we get the record of one-one mapping of each smallest sub triangles of in-radius ϵ of main triangles T_1 and T_2. Algorithm 2 named “Triangle Mapping Algorithm” describes the procedure of mapping all the interior points of two given triangles. Some steps of the process of this algorithm is shown in Fig.5.

5. TRIANGLE BASED FACE IMAGE MAPPING

We take two face images as source and target image. These two images may be of any angled image, we mean the image may be without any restriction of shape, size, pos, angle or distance from cameras. Our target is to find the one-one map of each pixel of source and target image where each mapped pixel positions must represent the same face part of both the images. Fig.6 shows one example of our experiment to transform a source image into the shape of target image.
In the first step we relate the face image with the given landmark points. The details of the method of landmark point detection is described in the section-6. The obtained landmark points helps us to find all the important face parts of the given image automatically and it detects the required face portion from the input image. Then in the second step we crop the face portion of the image and rotate it keeping the two eye coordinate in the horizontal axis and scaled it to our standard size. After completing this step we have the normalized images for our work.
Now we divide the each normalized images by our proposed 116 numbers of triangles from the 68 numbers of landmark points found applying the landmark tool earlier. Now these 116 triangles are our 116 numbers of sub-problem of our main problem.
Now with 116 numbers of iteration we pick up 116 pairs of corresponding triangles from both the normalized images. And for each pair of triangle we apply our triangle based mapping algorithm to get record one-one mapping of each pair of triangle. And thus finally we get record of one-one mapping face images. The results are shown in Fig.5 & 9. The entire process is described in the algorithm-1 named “Triangle-wise Object Mapping Algorithm”.

6. FINDING LANDMARK POINTS

In our work basic needs are the landmark points of the face images. The landmark points can be acquired automatically or manually. In this regard various works have been carried out for detecting landmark points automatically. Lin[12] et al. introduced a coupled 2D and 3D feature extraction method to determine the positions of



(a) Initial Δs, rad = 20

(b) 4 sub Δs, rad = 10

(c) 16 sub Δs, rad = 5

(d) 64 sub Δs, rad = 2.5

Fig. 5. Steps in Triangle Mapping Algorithm-2, read Δ as triangle

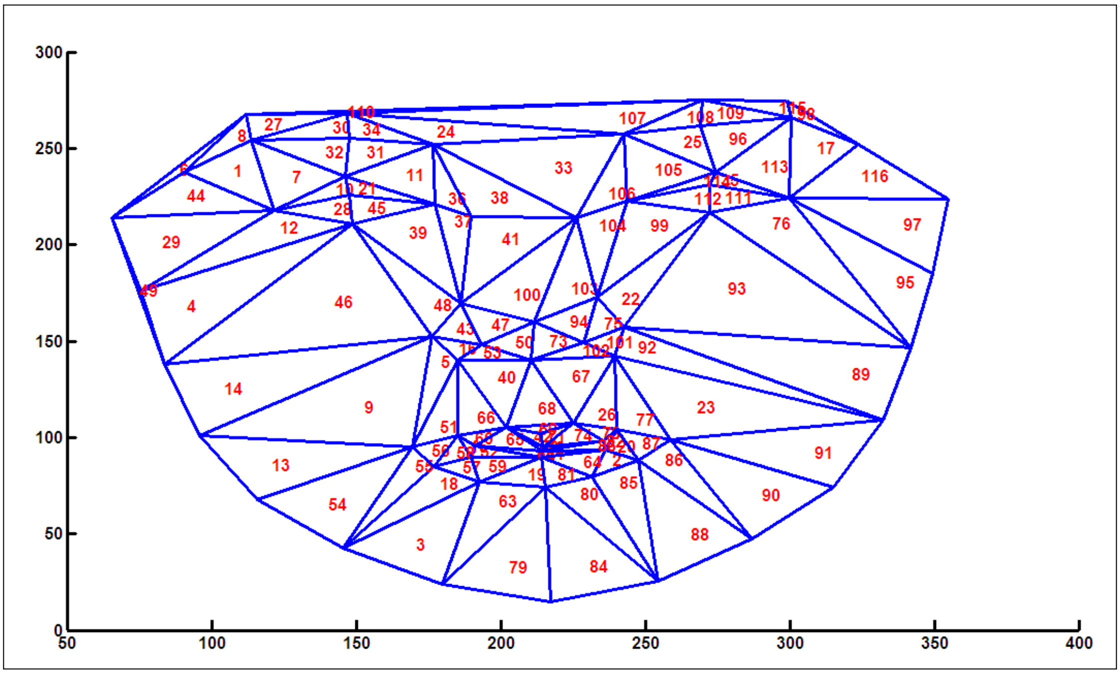


Fig. 7. Our proposed 116 Triangles of a face image

Table 1. Our proposed 116 triangles IDS represented by the 68 landmark points

Pts/ TrIDs	P_1	P_2	P_3	Pts/ TrIDs	P_1	P_2	P_3	Pts/ TrIDs	P_1	P_2	P_3	Pts/ TrIDs	P_1	P_2	P_3
1	28	22	27	30	24	26	27	59	61	62	59	88	10	9	56
2	57	63	56	31	26	25	29	60	66	50	51	89	44	13	12
3	59	7	6	32	27	26	29	61	62	67	63	90	11	10	55
4	3	2	31	33	25	19	46	62	62	66	67	91	55	12	11
5	49	40	41	34	24	25	26	63	59	58	7	92	43	44	12
6	23	22	1	35	37	34	33	64	62	63	57	93	13	44	36
7	27	29	28	36	38	30	25	65	66	51	67	94	48	68	45
8	23	27	22	37	38	39	30	66	50	41	51	95	13	33	14
9	4	40	49	38	46	38	25	67	43	53	42	96	34	20	21
10	29	32	28	39	39	31	30	68	53	51	42	97	33	15	14
11	29	25	30	40	42	51	41	69	53	52	51	98	21	17	16
12	28	31	2	41	39	38	46	70	65	67	51	99	36	45	35
13	49	5	4	42	65	51	52	71	53	65	52	100	68	39	46
14	3	40	4	43	47	40	39	72	64	67	65	101	43	48	44
15	50	66	61	44	22	28	1	73	68	48	42	102	43	42	48
16	41	40	47	45	30	31	32	74	64	65	53	103	46	45	68
17	33	21	16	46	40	3	31	75	45	44	48	104	46	35	45
18	60	59	6	47	47	39	68	76	36	33	13	105	34	35	19
19	59	62	58	48	40	31	39	77	55	54	43	106	35	46	19
20	56	63	54	49	1	2	3	78	64	53	54	107	24	18	19
21	30	32	29	50	68	42	47	79	58	8	7	108	19	18	20
22	44	45	36	51	50	49	41	80	58	57	9	109	20	18	21
23	12	55	43	52	61	66	62	81	58	62	57	110	24	23	18
24	19	25	24	53	42	41	47	82	63	64	54	111	36	37	33
25	34	19	20	54	49	6	5	83	63	67	64	112	36	35	37
26	53	43	54	55	49	60	6	84	58	9	8	113	33	34	21
27	24	27	23	56	49	50	60	85	57	56	9	114	37	35	34
28	31	28	32	57	60	61	59	86	56	55	10	115	18	17	21
29	2	1	28	58	60	50	61	87	55	56	54	116	33	16	15

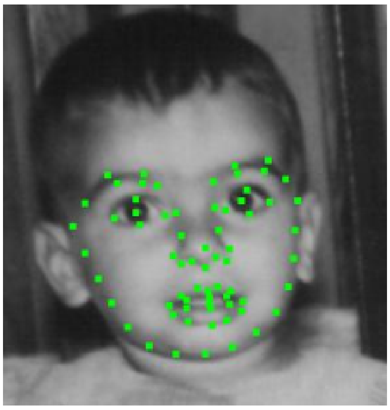


Fig. 8. Landmarks labeled in FGNET Aging Database

eye sockets by using curvature analysis. Segundo[23] et al. introduced a face and facial feature detection method by combining 2D face segmentation on depth images with surface curvature information in order to detect the eye corners, nose tip, nose base, and nose corners. Wei[20] et al. introduced a nose tip and nose bridge localization method, based on a Surface Normal Difference algorithm and shape index estimation in order to determine the facial pose in pose-variant systems. Mian[14] et al. introduced a heuristic method for nose tip detection that was based on a geometric analysis of the nose ridge contour. A recent work has been carried out by P Perakis[16] et al. on 3D Facial Landmark Detection under Large Yaw and Expression Variations. Y Ren[17] et al. presents a novel learning method for precise eye localization in order to improve the performance of face processing algorithms. Some other major methods are described in the papers[5] [19] [18] [15] [25].

Implementation of the landmark point detection is out of the scope of this work. We have experimented with the face images of the FG-NET aging database (section-10), where all the required 68 numbers of landmark points of each face images are available.

We have used a few numbers of face images beyond FG-NET aging database for which we manually marked the 68 landmark points in our experiments.

Standard 116 Nos of Triangles: In this work each face image is represented by our proposed non overlapping 116 triangles. Each triangle is constructed by 3 fixed landmark points. Fig.7 and table-5 describe our proposed 116 triangle IDs of given 68 numbers of landmark points of a face image. For example upper part of the left eye (Fig.7) is represented by the triangle ID 10, and the three corner of this triangle are the landmark point IDs 29, 32, 28 (table-5). The different triangle IDs that represent the face parts of a face image is described in table-6.

7. ALGORITHM

The following few notations are used in the procedures of our proposed algorithm 1, 2, 3, 4.

Table 2. Parts of face and corresponding triangle IDs

Parts of face	Our proposed triangle IDs
Left Eye	[10 21 28 45]
Right Eye	[112 111 35 114]
Left Eye Brow	[8 27 30 34]
Right Eye Brow	[98 115 109 108]
Nose	[41 100 103 43 47 94 75 16 53 50 73 102 101]
Mouth	[56 57 58 15 60 69 52 42 70 62 19 81 61 72 71 74 83 64 2 82 78 20 87]
Others	Remaining triangle IDs

Object $Obj = \{T_1, T_2, \ldots, T_n\}$ is a set of n numbers of non-overlapping triangles.
Triangle $T_1 = \Delta(P_1, P_2, P_3)$, where $P_i = (x_i, y_i), i = 1 \ldots 3$ are triangular vertices of Triangle T_1.
Triangle $T_2 = \Delta(Q_1, Q_2, Q_3)$, where $Q_i = (x_i, y_i), i = 1 \ldots 3$ are triangular vertices of Triangle T_2.
Algorithm 1 named "Triangle-wise Object Mapping Algorithm" does the work of mapping each points of two given objects Obj_1, Obj_2, where objects Obj_1 and Obj_2 are nothing but the set of finite numbers of disjoint triangles.

Algorithm 1 Triangle-wise Object Mapping Algorithm

procedure MAPPOBBJECT(Obj_1, Obj_2)
  for $i \leftarrow 1, n$ **do**
    $MappTriangle(T_1^i, T_2^i)$
    [Triangle $T_1^i \in Obj_1$ and $T_2^i \in Obj_2$]
  end for
end procedure

Algorithm 2 named "Triangle Mapping Algorithm" maps all the interior points of two given triangles T_1, T_2. Initially it checks the value of in-radius of the triangle pair and if it is greater than threshold value 1 for the both the triangles then it divides them into four equal subtriangles. Then recursively call the same function for these four pair of subtriangles. Otherwise in this case triangles are smallest and so it maps the incenters and boundary points of the incircles of the pair.

Algorithm 2 Triangle Mapping Algorithm

procedure MAPPTRIANGLE(T_1, T_2)
  if $Radius(T_1) < 1 \parallel Radius(T_2) < 1$ **then**
    $C_1 \leftarrow Center(T_1)$
    $C_2 \leftarrow Center(T_2)$
    $MappCircle(C_1, C_2)$
    for $i \leftarrow 1, 2, 3$ **do**
      $Mapp(P_i, Q_i)$
      [$P_i \in T_1$ and $Q_i \in T_2$]
    end for
    return
  end if
  $[T_1^{Upper}, T_1^{Left}, T_1^{Right}, T_1^{Middle}] \leftarrow Subtri(T_1)$
  $[T_2^{Upper}, T_2^{Left}, T_2^{Right}, T_2^{Middle}] \leftarrow Subtri(T_2)$
  for $Pos \leftarrow Upper, Left, Right, Middle$ **do**
    $MappTriangle(T_1^{Pos}, T_2^{Pos})$
  end for
end procedure

Algorithm 3 named "Sub-triangle Algorithm" divides a given triangle T into four numbers of sub-triangles named $T^{Upper}, T^{Left}, T^{Right}, T^{Middle}$. The experimental result is shown in Fig.4.

Algorithm 3 Sub-triangle Algorithm

procedure SUBTRI(T)
  for $i \leftarrow 1, 3$ **do**
    $P_i' \leftarrow \frac{P_i + P_{(i+1)\%3}}{2}$
  end for
  $T^{Upper} \leftarrow (P_1, P_1', P_3')$
  $T^{Left} \leftarrow (P_1', P_2, P_2')$
  $T^{Right} \leftarrow (P_2', P_3, P_3')$
  $T^{Middle} \leftarrow (P_1', P_2', P_3')$
  return $[T^{Upper}, T^{Left}, T^{Right}, T^{Middle}]$
end procedure

The algorithm 4 named "Circle Mapping Algorithm" maps all the boundary points and the center points of two given circles C_1, C_2.

Algorithm 4 Circle Mapping Algorithm

procedure MAPPCIRCLE(C_1, C_2)
  Radius $r = 1$
  for $\theta \leftarrow 45^o, 360^o$ **do**
    $x_1 \leftarrow x(C_1) + r\cos(\theta)$
    $y_1 \leftarrow y(C_1) + r\sin(\theta)$
    $x_2 \leftarrow x(C_2) + r\cos(\theta)$
    $y_2 \leftarrow y(C_2) + r\sin(\theta)$
    $P = (x_1, y_1)$
    $Q = (x_2, y_2)$
    $Mapp(P, Q)$
    $\theta \leftarrow \theta + 45^o$
  end for
  $Mapp(C_1, C_2)$
end procedure

8. EXPERIMENTS

We have experimented to transform a given image 'Fig.3.a' into the 8 different form of poses[6], where template poses are given in column-1, the outputs of our experiments are shown in column-2, the original images in the pos of template images are shown in column-3 and computed measurement of similarity by equation-6 $Sim(Image_1, Image_2)$ ranges from 1 to 0 of the outputs and the original images are shown in column-4 respectively in Fig.9.

9. SIMILARITY MEASURE OF TWO MAPPED FACE IMAGES

For any two given images A and B we found mapped vector as $V_A = (a_1, a_2, \ldots, a_n)$ and $V_B = (b_1, b_2, \ldots, b_n)$ using our proposed image mapping algorithm. Here the values of $a_1, a_2, \ldots, a_n$ are the intensity values of i^{th} mapping point of image A, $i = 1, 2, \ldots, n$, and similarly for image B. Thus we have

$$V_A = (a_1, a_2, \ldots, a_n), a_i = [0, 255] \quad (1)$$

$$V_B = (b_1, b_2, \ldots, b_n), b_i = [0, 255] \quad (2)$$

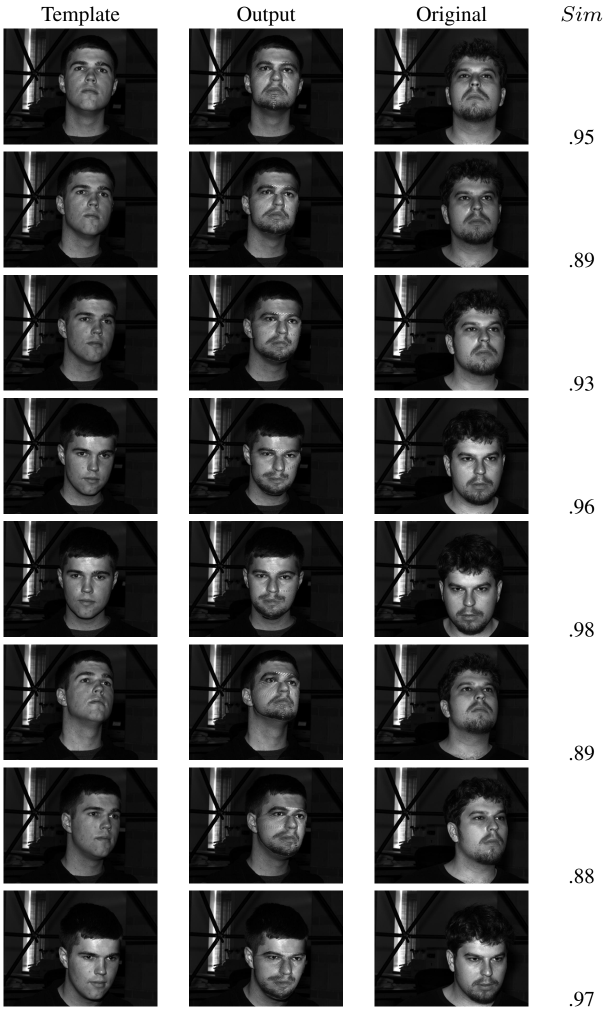


Fig. 9. Result of face transformation of a given image from Fig.3.a. 8 different poses are presented from row no 1 to 8. 1^{st} column represents the templates to transform, 2^{nd} column is the outputs of our experiment, 3^{rd} column is the actual real images for comparisons with our output, 4^{th} column represents $Sim(Output, Original)$ = similarity between output and original image. All these images are taken from ExtendedYaleB database

where $i = 1, 2, \ldots, n$. We divide the vectors V_A and V_B by 255 so that we can get euclidian distance between in these vectors in the range of $[0, 1]$, therefore from equation-1, 2 we get

$$V_A = (a_1, a_2, \ldots, a_n), a_i = [0, 1] \tag{3}$$

$$V_B = (b_1, b_2, \ldots, b_n), b_i = [0, 1] \tag{4}$$

where $i = 1, 2, \ldots, n$. Now Euclidean Distance δ for V_A and V_B is as given below

$$\delta(V_A, V_B) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2} = [0, \sqrt{n}] \tag{5}$$

Table 3. Specifications of the FG-NET aging database

Total number of images	1002	
Number of subjects	82	
Number of images per subject	6 to 18 (on average 12 images/subject)	
Minimum age in year	0	
Image Type	JPG images, color or grey scale images	
Image Resolution	Variable - approximately 400x500	
Conditions encountered on images from the database	Illumination	Varying
	Pose	Varying
	Expression	Varying
	Beards	Yes
	Moustaches	Yes
	Spectacles	Yes
	Hats	Yes

Now we calculate similarity between image A and B as $Sim(A, B)$

$$Sim(A, B) = 1 - \frac{\delta(V_A, V_B)}{\sqrt{n}} = [1, 0] \tag{6}$$

10. DATABASES

FGNET Aging database: The Face and Gesture Recognition Research Network (FG-NET) provides an image database containing face images showing a number of subjects at different ages. The database has been developed in an attempt to assist researchers who investigate the effects of aging on facial appearance. For each image in the database the corresponding points file is available. Point files contain the pixel coordinates of 68 landmarks points located on the face. The positions of the landmarks are shown as green spots in the Fig. 8. The specification of this database is described in table-10.
The Extended Yale Face Database B[6]: The extended Yale Face Database B contains 16128 images of 28 human subjects (Fig.10) under 9 poses and 64 illumination conditions.

11. CONCLUSION

We developed a method to perform the automatic transformation of one face images into another using our proposed triangle based mapping algorithm. In this work we have described the recent advances in the field of image morphing and image transformation. Here we do not need any training set of faces to obtain any model. This is a very simple technique that works for any type of face images, that means face image may be taken from any angle or may be of any posed. As a future work we plan to construct synthesis face image at older ages of given face image of a child. This work may be extended in animated movie field to act for a particular face by any given face. As a future work this method may be used in polymorphing also, polymorphing: morphing among multiple images involves a blend of several images at once.
In this work we got the scarcity of available tools to detecting the landmark points for a face image. We feel the importance of this type of tool and as our future work we are in the way of developing a useful tool to detect the landmark points.

12. ACKNOWLEDGMENTS

The authors would like to thank all the reviewers for their helpful and constructive comments to this work.

Fig. 10. Samples of 28 human subjects in Extended Yale Face Database B

13. REFERENCES

[1] Nur Arad and Daniel Reisfeld. Image warping using few anchor points and radial functions. *Computer Graphics Forum*, 14:35–46, 1995.

[2] Thaddeus Beier and Shawn Neely. Feature-based image metamorphosis. *SIGGRAPH Comput. Graph.*, 26(2):35–42, July 1992.

[3] Urvashi Bhushan, G. P. Saroha, and Disha Tiwari. An implementation of image morphing through mesh morphing algorithm. *International Journal of Advanced Research in Computer Science and Software Engineering*, 2(7), 2012.

[4] Martin Bichsel. Automatic interpolation and recognition of face images by morphing. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition, ICAFGR 96*, pages 128–135, 1996.

[5] Timothy C. Faltemier, Kevin W. Bowyer, and Patrick J. Flynn. Rotated profile signatures for robust 3d feature detection. In *FG*, pages 1–7. IEEE, 2008.

[6] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001.

[7] Yanwen Guo, Guiping Zhang, Zili Lan, and Wenping Wang. Efficient view manipulation for cuboid-structured images. *Computers Graphics*, (0):–, 2013.

[8] Aaron W. F. Lee, David Dobkin, Wim Sweldens, and Peter Schrder. Multiresolution mesh morphing. In *PROCEEDINGS OF SIGGRAPH 99*, pages 343–350, 1999.

[9] Seung-Yong Lee, Kyung-Yong Chwa, James Hahn, and Sung Yong Shin. Image morphing using deformation techniques. 7(1):3–24, January 1996.

[10] Seung-Yong Lee, Kyung-Yong Chwa, and Sung Yong Shin. Image metamorphosis using snakes and free-form deformations. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pages 439–448, New York, NY, USA, 1995. ACM.

[11] Seungyong Lee, George Wolberg, Kyung yong Chwa, and Sung Yong Shin. Image metamorphosis with scattered feature constraints. *IEEE Transactions on Visualization and Computer Graphics*, 2:337–354, 1996.

[12] T. H. Lin, W. P. Shih, W. C. Chen, and W. Y. Ho. 3d face authentication by mutual coupled 3d and 2d feature extraction. In *Proceedings of the 44th Annual Southeast Regional Conference*, ACM-SE 44, pages 423–427, New York, NY, USA, 2006. ACM.

[13] Peter Litwinowicz and Lance Williams. Animating images with drawings. In *SIGGRAPH*, pages 409–412. ACM, 1994.

[14] Ajmal S. Mian, Mohammed Bennamoun, and Robyn Owens. An efficient multimodal 2d-3d hybrid approach to automatic face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007:1584–1601, 2007.

[15] Prathap Nair and Andrea Cavallaro. 3-d face detection, landmark localization, and registration using a point distribution model. *IEEE Transactions on Multimedia*, 11(4):611–623, 2009.

[16] P. Perakis, G. Passalis, T. Theoharis, and I. A. Kakadiaris. 3d facial landmark detection under large yaw and expression variations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1552–1564, 2013.

[17] Yan Ren, Shuang Wang, Biao Hou, and Jingjing. Efficient view manipulation for cuboid-structured images. *IEEE Transactions On Image Processing*, 23(1), January 2014.

[18] Marcelo Romero and Nick Pears. Landmark localisation in 3d face data. In *Proceedings of the 2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, AVSS '09, pages 73–78, Washington, DC, USA, 2009. IEEE Computer Society.

[19] Albert Ali Salah and Lale Akarun. 3d facial landmarking under expression, pose, and occlusion variations, 2008.

[20] Xiaozhou Wei, Peter Longo, and Lijun Yin. Automatic facial pose determination of 3d range data for face model and expression identification. In Seong-Whan Lee and Stan Z. Li, editors, *ICB*, volume 4642 of *Lecture Notes in Computer Science*, pages 144–153. Springer, 2007.

[21] George Wolberg. *Digital Image Warping*. IEEE Computer Society Press, Los Alamitos, CA, USA, 1st edition, 1994.

[22] George Wolberg. Recent advances in image morphing. In *Proceedings of the 1996 Conference on Computer Graphics International*, CGI '96, pages 64–, Washington, DC, USA, 1996. IEEE Computer Society.

[23] T. Yu and Y. Moon. A novel genetic algorithm for 3d facial landmark localization. *Biometrics: Theory, Applications, and Systems*, 2008.

[24] Vittorio Zanella and Olac Fuentes. An approach to automatic morphing of face images in frontal view. In Raul Monroy, Gustavo Arroyo-Figueroa, Luis Enrique Sucar, and Juan Humberto Sossa Azuela, editors, *MICAI*, volume 2972 of *Lecture Notes in Computer Science*, pages 679–687. Springer, 2004.

[25] Xiangxin Zhu and Deva Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, pages 2879–2886. IEEE, 2012.