Digital Image Processing

Face Morphing

Done by:

G. Nanda Kishore (S20180010054)
S. Harsha Vardhan (S20180010151)
B. Abhinay (S20180010022)
P. Kalasagar (S20180010129)

Course instructor: Dr. Mrinmoy Ghorai

Face Morphing

Definition:

- **Morphing:** Morphing is a special effect that transforms one image or shape into another through seamless transitions. Traditionally, such depictions have been achieved by melting techniques on film.

- A Morphed facial images are synthetically created images that closely resemble the faces of two subjects, both of which can be used to validate against biometric systems. counteract image degradation during the creation of morphed facial images caused by blending.

Problem statement

- **Input:** Two images containing human faces (Image I_0 and Image I_1)
- **Output:** A fluid transformation video transitioning from I_0 to I_1 with intermediate image(M_t)
- **Goal:** The transition should be smooth and the intermediate frames should be as realistic as possible

Face Morphing

- Morphing is tweening between different images.
- Morphing derives from “metamorphosis”, an ancient greek word meaning “transformation”.

How’s face morphing done?

- Algorithm explains the extra feature of points on face and based on these feature points, images are partitioned and morphing is performed.
- Algorithm has been used to generate morphing between images of faces of different people as well as between images of images of individuals.

Morphing two images

Types of morphing algorithm

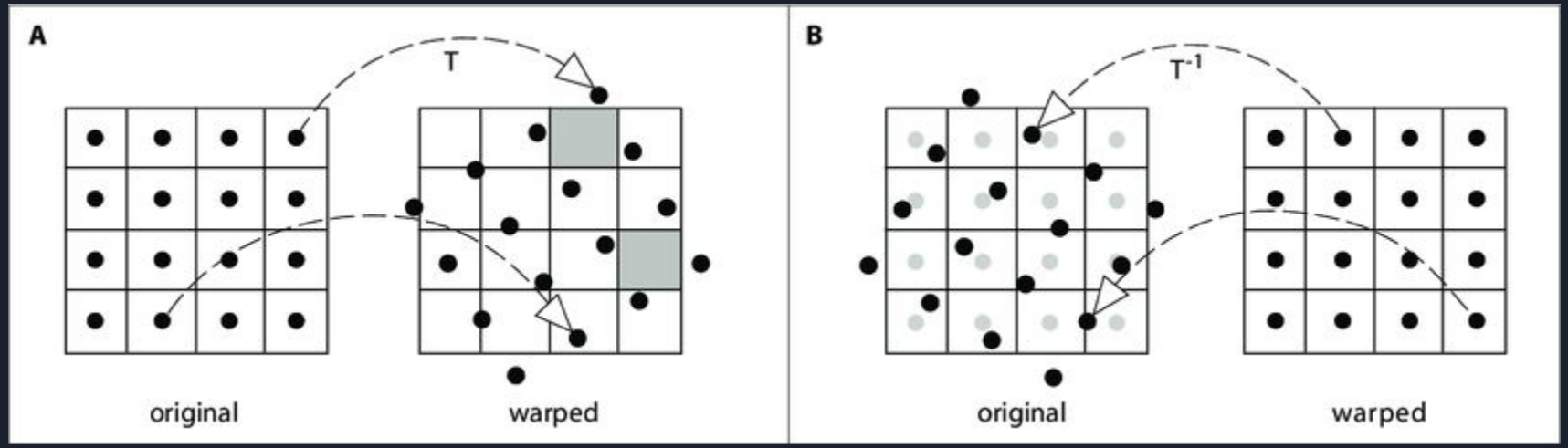
Morphing involves two types of algorithms

1. Dissolving (creates transition morphs)
2. Warping (creates distortion morphs)

Warping

- There are two ways to warp an image
- The first is forward warping, scan through the source image pixels to pixel and copied to appropriate place in the destination image
- The second one, reverse mapping goes through the destination image pixel by pixel and sample the correct pixel from the source image.
- The most important feature of inverse mapping is that every pixel in the destination image gets set something appropriate.
- In forward mapping case some pixels in the destination might not get painted and would have to be interpolated we calculate the image deformation as the reverse mapping.

Image warping



Forward-warping

Backward-warping

Working of face morphing algorithm

Image I_0 and Image I_1, Parameter t controls morphing from I_0 to I_1. The value of t is from 0 to 1. Both images I_0 and I_1 are assumed to be of the same dimension.

$$M(x_t, y_t) = (1-t) * I_0(x_j, y_j) + t * I_1(x_j, y_j)$$

Calculating morphing frames

Now, we have a high-level idea of how morphing frames are calculated.

Given the corresponding features pairs, for each frame at time t:

1. Compute the intermediate shape by linear interpolation of each feature pair:

$$x_t = (1-t) * x_j + t * x_j$$

$$y_t = (1-t) * y_j + t * y_j$$

2. Cross dissolve the color by interpolating two images:

$$M(x_t, y_t) = (1-t) * I_0(x_j, y_j) + t*I_1(x_j, y_j)$$

STEPS TO FOLLOW THE ALGORITHM

Step 1:

In this step, we have to mark the key points of both images I0 and I1 and we have to establish the one-to-one relation between the points in the image I0 and I1 correspondingly. The number of points marked in image I0 must be equal to the Number of points marked in image I1.

Step 2:

After marking all these points in the both images we have to calculate the weighted averages of the two sets of points based on a value of t.

If the two set of points are:-

SI0 = {m1, m2, m3,… .., mk}

SI1 = {n1, n2, n3,… ..,nk}

Let the other set of points for intermediate step is

SMt = {i1, i2, i3,… .., ik}

$ik = (1-\alpha)\, mk + \alpha nk$

Step 3:

Next we have Warping Triangular Mesh..

With the paired landmarks, we could define the same triangular mesh over two sets of points, which will give us triangle-to-triangle correspondences between two sets of facial features. Why would we use a triangle over other primitives? My understanding is that they are easy to work with and tessellate well so that we won't end up with cracks in the mesh where two edges join. Now we can break down the problem of defining the intermediate shape into many sub-problems — finding each intermediate triangles and then cross-dissolve the values into colors in the warped shape. Before we dive into how to assign the points into triangles, we will first show the feasibility of warping one triangle to another.

Now select the triangle T from the image I_0, select the corresponding triangle V in the intermediate image M_t, and calculate the affine transformation (explained below) that transforms the triangle T into V

Affine mapping

Affine mapping :-

U = ax+by+c

V = dx+ey+f

A square can be mapped to an arbitrary parallelogram

T A = B

ð T = B A-1

To calculate the transformation, apply the transformation matrix T to all the pixels in the triangle P. Let the point be X.

XT = [x y 1]

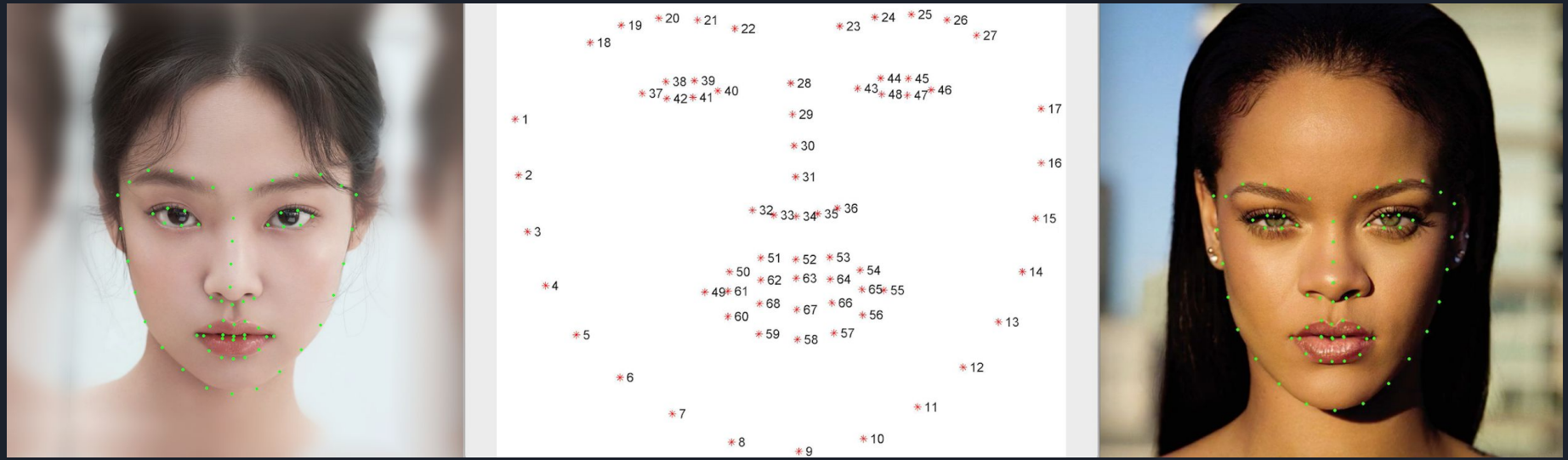
X' = T X,

Where X' is the position of the converted pixel.

FUNCTIONS WRITTEN

1. get_landmarks

Gives landmarks for origin and destination images.



2.) landmarks_to_points

It takes near set of landmarks to form a set of points.

3.) find_sub_pixel

Pixels will be updated by image matrix

4.) triangulate

It will form the triangulars using the warping triangulation with the nearby points.

5.) warp_face

It will perform warping on the face

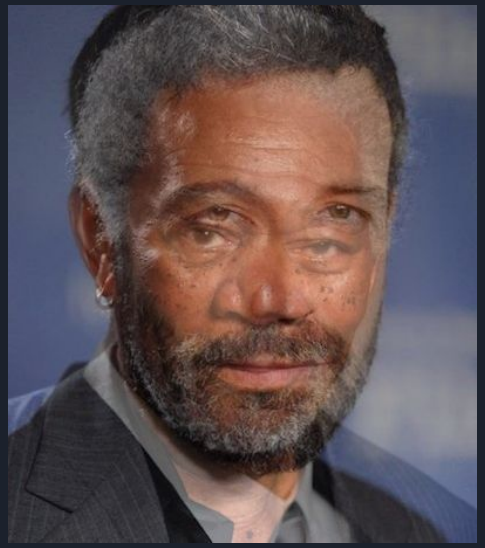
6.) morph_face

It will morph the face using the variable t between the two given input images.

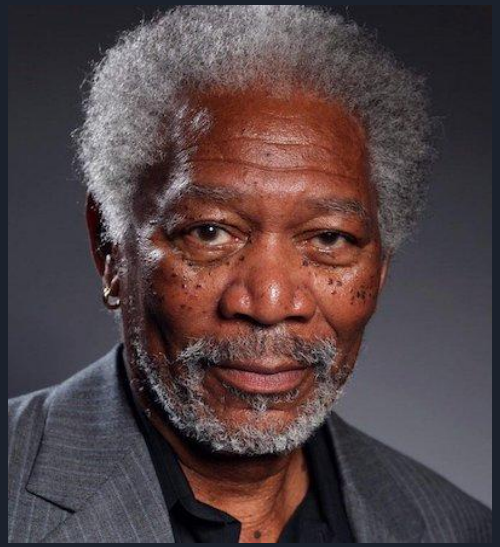
7.) run_face_morphing

Runs the face morphing code. You have to give origin and destination images that you want to morph.

Origin image



75th frame (Intermediate)



Destination image

Conclusion

A face morphing algorithm is proposed. The algorithm consists of a feature finder followed by a face-morpher that utilizes affine and coordinate transforms.

We believe that feature extraction is the key technique toward building entirely automatic face morphing algorithms. We demonstrated that a image of two human faces can be generated by morphing, and the hybrid face we generated resembles each of the two "parent" faces. Also, we demonstrated that face morphing algorithms can help generate animation.

Ideally speaking, the more feature points we can specify on the faces, the better morphing results we can obtain. If we can specify all the important facial features such as the eyes, the eyebrows, the nose, the edge points of the mouth, the ears, and some specific points of the hair, we are confident that we can generate very smooth and realistically looking morphing from one image to another.

Thank you