COL783: Digital Image Processing

Assignment 2

Tushar Verma (2022AIY7514) & **Naimish Pintukumar Machchha** (2022AIY7512)

Part 1: Face Expression Transfer

Task 1: Warp matrix based expression transfer for one triangle

Assumption: Our program expects input as 3 channel RGB images only

Methodology:

We have created a GUI for taking corresponding anchor points for I1, I2 and I3 all at the same time. This ensures that point k of I1 semantically corresponds to the same point in I2 and I3.

As given in the procedure we use these anchor points to Find the warp matrix (H1) between the corresponding triangles of I1 and I2 and use it to warp I1 to I1' and I3 to I3' and (H3) between the corresponding triangles of I1' and I3'.

To find H1 and H3 we have created two functions one is using cv2's **getAffineTransform** method while the other one that we have created is solving simultaneous equations and finding the unknowns, both works seamlessly.

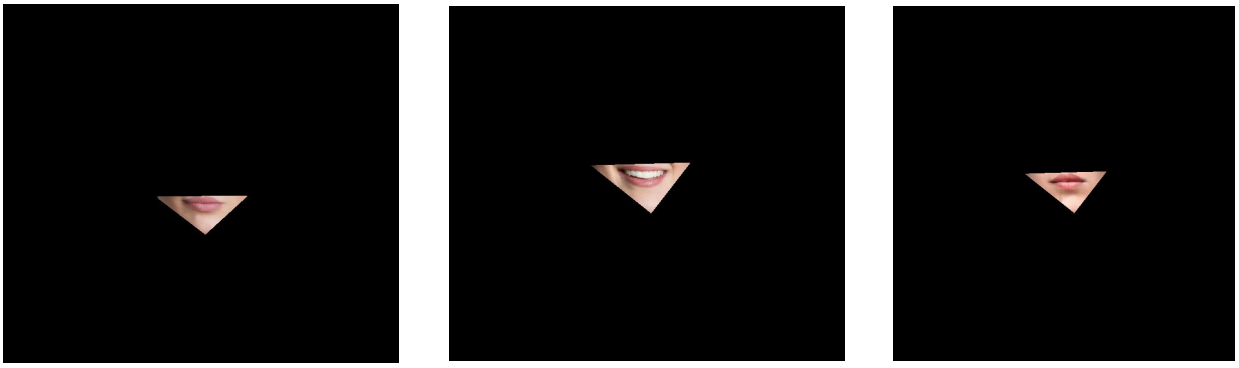
For warping we have implemented an **inverse warping** method which considers the target triangle and uses the inverse H1 and H3 matrix to get the mapped point in the source triangle and take the pixel from there and put it into the considered point inside the target triangle.

To find if a given point is inside a triangle or not we have used an area based method where we see if three supposedly smaller triangles having two vertices from the original triangle and the one vertex is the considered point. If the sum of

area of these smaller triangle becomes equal to the outer bigger triangle then the point is present inside the triangle
Finally as mentioned we used the inverse of the H2 matrix to get I4 from I2

Results:
GUI to select anchor points
I1, I2, I3 are lined one after another leftwords

I1’ I3’ I4



Conclusion: As given in the task, only one triangle is warped.

Task 2: Barycentric coordinate-based expression transfer for one triangle

Assumption: Our program expects input as 3 channel RGB images only

Methodology:

We have used the same anchor points given to task1 using the GUI program

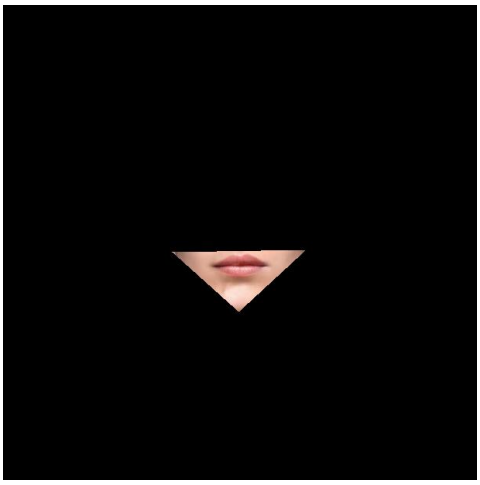
We have implemented a reverse warping function similar to that in task1 but using barycentric coordinates and its related equation.

As It was further discussed in class we first warped I2 to the triangle of I1 and finally warped this triangle to the triangle of I3 to finally get I4

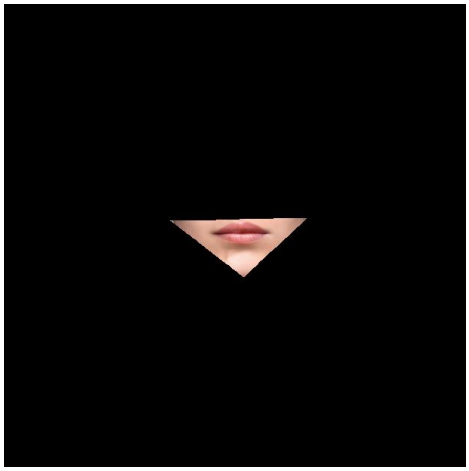
To find if a given point is inside a triangle or not we have used an area based method where we see if three supposedly smaller triangles having two vertices from the original triangle and the one vertex is the considered point. If the sum of area of these smaller triangle becomes equal to the outer bigger triangle then the point is present inside the triangle

Results:

I2’ = I2 to I1 triangle warp



I4=I2’ to I3 warp triangle



Conclusion: As given in the task, only one triangle is warped.

Task 3: Expression transfer over the entire image

Assumption: Our program expects input as 3 channel RGB images only

Methodology:

We have used the same GUI application we used in task1 for taking corresponding anchor points for I1, I2 and I3 all at the same time. This ensures that point k of I1 semantically corresponds to the same point in I2 and I3.

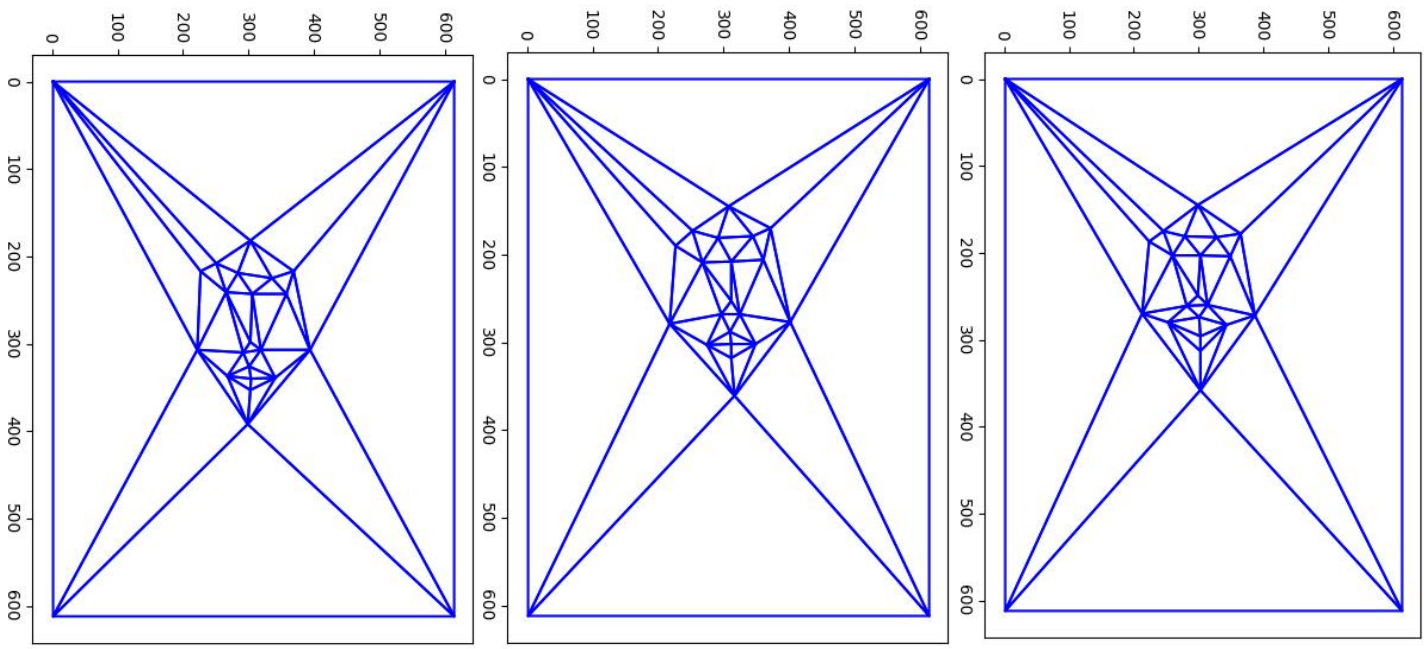
As asked in the task post anchor selection we have also included the corner points in the set of points for triangulation.

We have implemented **Delaunay triangulation** using **Bowyer Watson** algorithm from scratch and used it to find triangles in any one of the three images.
Then used the set of triangles to find the corresponding triangle in the other two images by making ono-one corresponding to the anchor points.

After getting the triangles for all the three images we perform task1 and task2 **recursively** on the triangles to obtain I4 in both the ways using barycentric coordinates and using affine warp matrix.

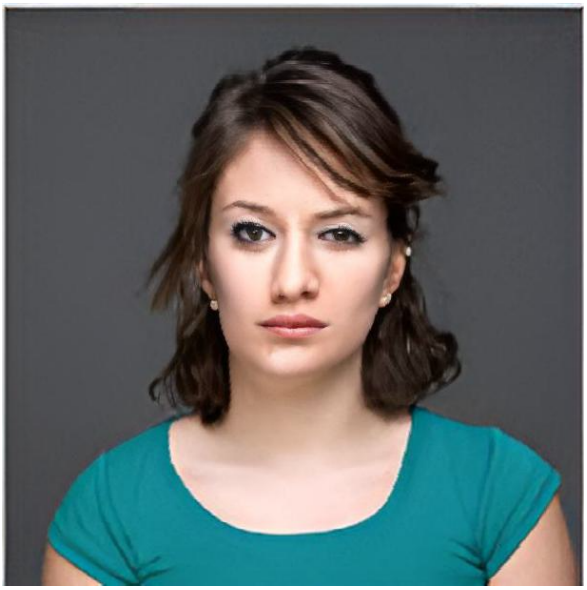
Results:
GUI to select anchor points (20 points example shown below)
I1, I2, I3 are lined one after another leftwords

Delaunay Triangulation of I1, I2, I3



Barycentric coordinate-based result:

I2’ = I2 to I1 triangle warp



I4=I2’ to I3 warp triangle

Affine Warp matrix-based result:

I1’ I3’ I4



Conclusion: From the results of task3 it is evident that task1 and task2 are working properly since we are using the same function.
It is also visible that warping using barycentric coordinates approach is giving much better result than affine warp matrix approach This is because when we select anchor points for the images I1,I2,I3 the transformation between corresponding triangles might be non-linear, such as with facial expressions or complex deformations, an affine warp matrix may not capture these deformations accurately. Barycentric coordinates provide a more flexible way to handle nonlinear deformations. Also since we are warping I3 to I3’ using warp matrix determined using I1 and I2, the image space is changed, maybe this is the cause for poor results.

Other Results:

Task 3:
For anchor points: 1

Result for barycentric approach

Result for warp matrix approach

For anchor points: 5

Result for barycentric approach

Result for warp matrix approach

For anchor points: 10

Result for barycentric approach

Result for warp matrix approach

Part 2: Face Swapping

Assumption: Our program expects input as 3 channel RGB images only also ensure that point k of source semantically corresponds to the same point in target

Methodology:

We have used the modified GUI application we used in task1 for taking corresponding anchor points for two images only source and target. In this part we decided to take 30 anchor points

Then similar to task3 we have performed delaunay triangulation on the anchor points, as mentioned we have limited the triangulation to the face only.
Then we use barycentric coordinates to transfer source image triangle pixels to the target image triangle pixels.

While transferring the pixel we have also performed the color adjustment step. Initially we normalized the source image Luminance to the target image Luminance by converting the images to LAB color space and then performing the z normalization on the source image.
Now when we transfer a pixel from source to target image we only copy the Luminance value to leave the color value of the target as it is, This is done by converting the source and target pixels to LAB color space and only copy the luminance value to the source pixel to the target pixel and preserve the color values of the target pixel as it is. To make this work we need to make sure that while selecting anchor points we need to ensure that point k of source semantically corresponds to the same point in target which was one of the prerequisites.

Finally to blend the source face into target face we used a masked gaussian filter approach, in which we find the mask of the face being swapped and then on it we apply a gaussian filter of high degree. Finally we use the mask value to blend the source pixel with the target pixel.

Face alignment is automatically done by one-one correspondence of the delaunay triangles between source and destination

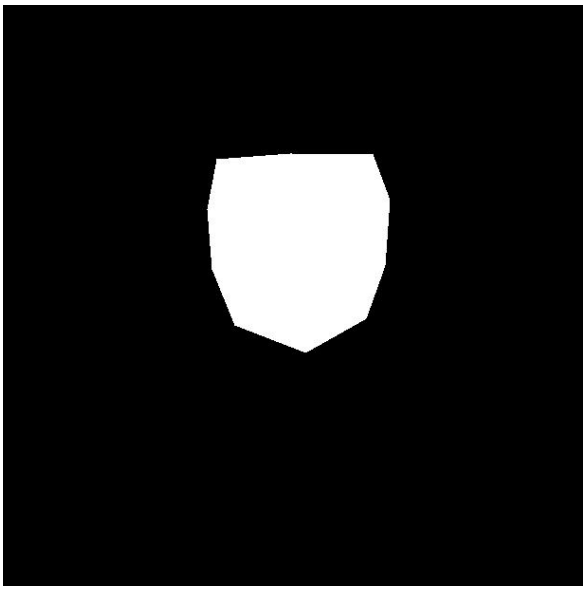
Results: 1

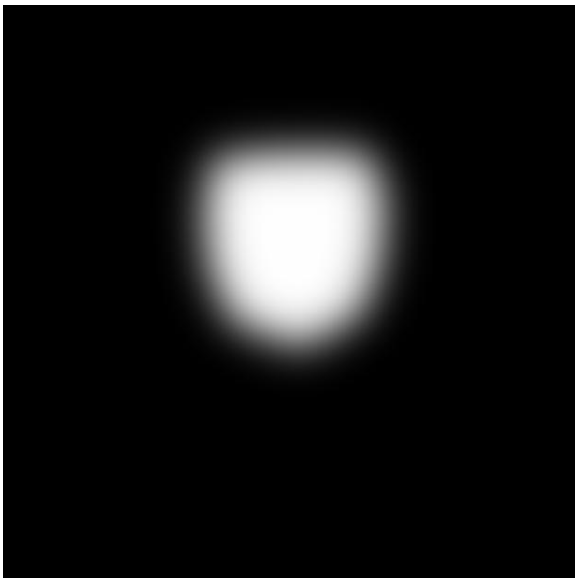
Source face

Target image

Transferring the delaunay triangles with and without color manipulation technique discussed above

Mask and Gaussian mask (kernel size: 151) for smoothing



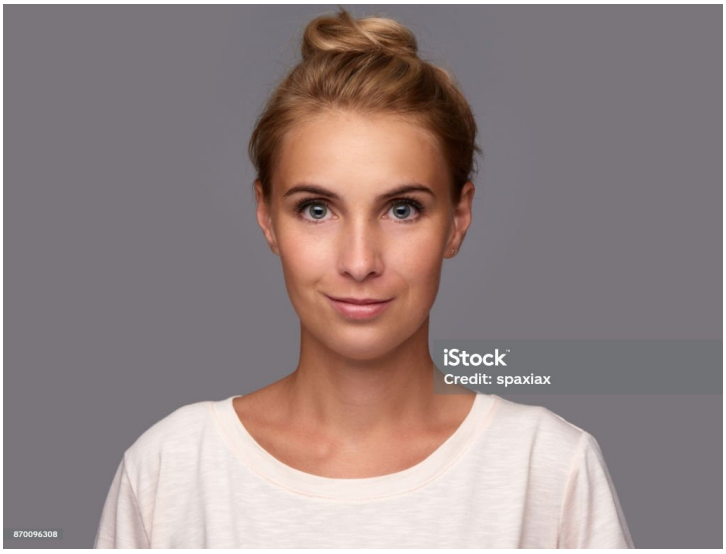


Final image

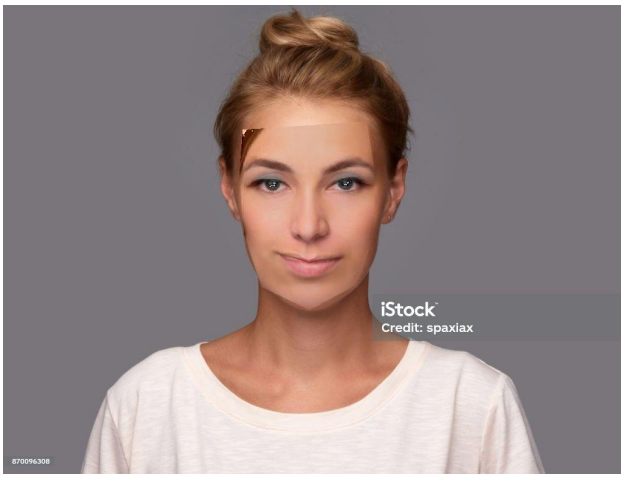
Results: 2

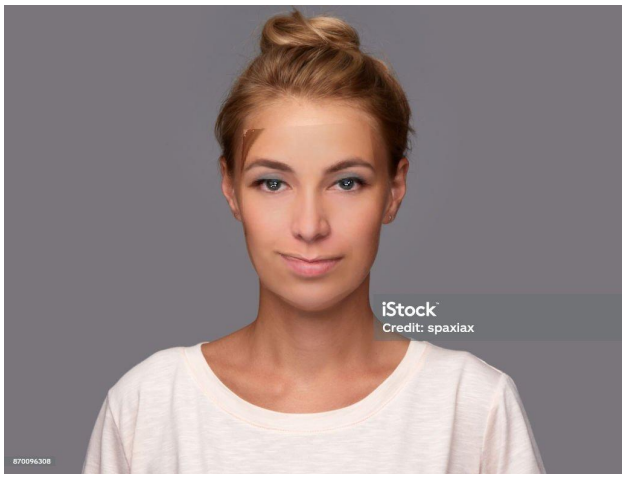
Source face

Target image



Final image with and without smoothing





Women end up smiling as well because of the choice of the anchor points

Conclusion: The performance of the algorithm visually speaking depends highly on the anchor selection, and the kernel size used for masking and blending the image.