Mohit M.Tech AI 2022AIM1003

DIP Lab Assignment 2

Image Morphing

- **INPUT**

Two RGB images of identical sizes, image1 and image2.

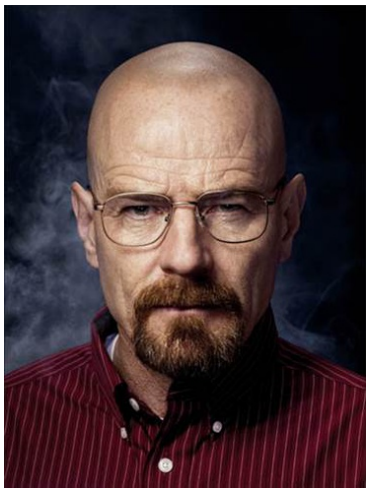


image1

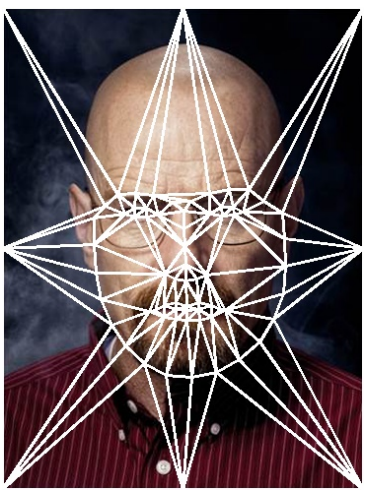
Image2

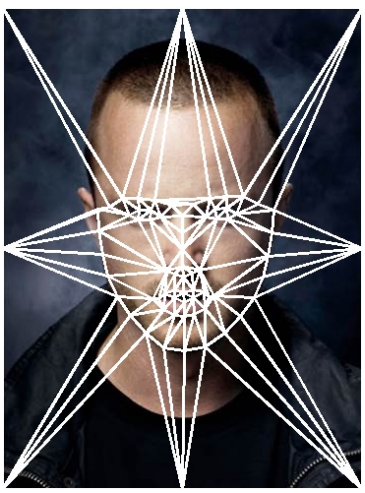
- **IMAGE MORPHING PROCESS**

1. **Feature Points:** Find feature points of each image such as the eyes, nose, lips, face boundaries etc. To extract the feature points '**dlib'** library is used. 'dlib' gives the coordinates of the 68 feature points in an array.

2. **Additional Feature Points:** Apart from 68 feature points provided by dlib library. Some additional feature points are also added manually. For example, for corner points and middle point of each side of the image

3. **Triangulation:** Once the feature points of both the images are ready, Delaunay Triangulation is done. Below shown is the Delaunay triangles of image1 and image 2. To achieve this, Delaunay function from scimitar learn library is utilised.





- Initialise value of alpha with 0.

4. **Finding Corresponding location of Feature points in intermediate image:** Take feature points from both the images (i.e. input and output images) and finding the corresponding location of that feature point in the intermediate image using below formula.

$$x\_inter = (1-alpha) * x\_inp + alpha * x\_out$$
$$y\_inter = (1-alpha) * y\_inp + alpha * y\_out$$

   Repeat above process for all the feature points.

5. **Triangulation in Intermediate image:** As we have used the Delaunay function from the scikit-learn library, we do not need to do the triangulation again for intermediate image. Because Delaunay function returns the Nx3 array in which each row has the index of three points in feature points array which forms the triangle. Since the order of the points is same for all the three images so we can use the same indexes to get the triangles for both output image as well as intermediate image.

- Initialise an empty image with all pixel values as zero and size same as the input and output images.

6. **Elementary Process:** Pick the triangles from the intermediate image and perform the following operations:

   i. Find a smallest axis parallel rectangle that encloses this triangle
   ii. Do the same for the corresponding input and output triangles
   iii. Find the relative position of the enclosed triangle in the enclosing rectangle (i.e. take reference from the top left corner of the enclosing rectangle and find the relative coordinates of the enclosed triangle), for each of the three triangles (I.e. input, intermediate and output)
   iv. Now find the transformation matrix (**M_in**) which transforms the relative input triangle (found in previous step) to the relative intermediate triangle (found in previous step) and similarly find for the output side (**M_out**)
   v. Transform the bounding rectangle of input triangle with **M_in** and bounding rectangle of output triangle with **M_out**
   vi. Find the corresponding resultant rectangle in the intermediate image using below formula:

$$\mathbf{r\_rect = (1 - alpha) * inp\_rect + alpha * out\_rect}$$

   vii. Create a mask (**M**) to only take the enclosed triangle from the enclosing rectangle of the intermediate image. Also create a reverse mask of this mask (**M_inv = 1 - M**)
   viii. Now replace the intermediate rectangle with below formula:

$$\mathbf{int\_rect = int\_rect * (1-M\_inv) + r\_rect * M}$$

   Note: Above replacement needs to done in the intermediate image using slicing

7. Repeat step 6 for all the triangles in the intermediate image and save the image in a list
8. Increment alpha according to number of intermediate images required (keep alpha between 0 to 1) and repeat from step 4 to 7.
9. Once all the intermediate images are saved in a list, then create a gif with all the images in the list.

- **OUTPUT**