Computer Vision Assignment № 2

Metehan Seyran, 150170903, Istanbul Technical University 29/11/2020

Part 3 - Face Morphing

A.

This part calculates the coefficients for affine transformations for triangle1(source) to triangle2(target).

B.

This part loops from 0 to 1 with step size 0.02. The iterator t is weight for second image, and $1 - t$ is weight for the first image. By using the weights, it decides how much should first and second image should appear in that specific frame.

C.

The function *make_homogeneous* creates array that is used in Affine Transformation. This function creates the x for the equation $x' = Ax$.

D.

This part creates the matrix M in the equation $q = Ma$, which is a modified version of the equation $x' = Ax$ where the coefficient function A is flattened, and the matrix containing old points x is converted to the matrix M.

E.

This part calculates the coefficients for the affine transform by inverting the matrix M, and multiplying with the new triangle points, which defined as q in the equation $a = M^{-1}q$.

F.

This part converts the flattened coefficients to a 2d matrix and adding $[0, 0, 1]$ row at the end.

G.

This part calculates the bi-linear interpolation using vectorized way.

H.

This part rounds the result value computed in part G.

I.

This path weights the locations of source and target triangles by using the t variable. It gives $(1-t)$ weight for the source image, and t for the target image. As t increases, the new image will look more similar to the target image from source image.

J.

This will create a mask for the specific triangle, and fill it with values 255, and the locations that is not in region of interest are filled with 0.

K.

This will return points x and y located inside the specified triangle mask.

L.

This part will stack points and form $[x, y, 1]^T$ matrix of image1(source).

M.

This part will flatten the intermediate triangle points from the weighted matrix calculated in I.

N.

This part multiply the the mask with the new coefficients calculated by using *calc_transform* of transformation from first image's triangle, to intermediate triangle points and by using the coefficients, the matrix $[x', y', 1]^T$ will be calculated by multiplying the triangle region and coefficients.

O.

Calculate the new transformed images of source and target using bi-linear interpolation, and save it to *inter_image_1*, *inter_image_2*.

P.

Weight the final source and target image with $1-t$ and t. Finally, return the created image.