SCHOOL OF DATA SCIENCE,FUDAN UNIVERSITY

Homework5 Deformation On Human Face

Linhangyu, Xiajinghui, Mayiqing

May 31, 2017

1 ABSTRACT

Homework 5 mainly focus on the conversion from the human-being's face to the baboon's face. To achieve this goal, we write a program principally applying three methods based on affine transformation. When a human's photo and a baboon's photo are given, the program can generate the new photo based on human's photo but have the feature of babnoon. We also realize an interactive interface for convenient practical application which can invoke these three algorithms and change the parameters' value in the algorithms.

In this report, we mainly use three different algorithms. The first is the Local Affine Transformation [3], in which we also tried different kinds of affine method from simple translation to straight lines transformation. The second is the Moving least square [2]. We applied this algorithm to compare the performance of different deformation algorithms . The third is called Image Warping Morphing. We hope to achieve a superior gradually-change visual effect by applying this algorithm. We also introduced several difficult problems we have overcome and some improvements we have made in the algorithms. Then comes the interactive interface, we design the GUI based on the PySide package which is convenient for users to change the photo they want. Last part is the conclusion of our work.

2 INTRODUCTION

Affine transformation is a common concept in geometry, which also called affine map or affinity. It means a function between affine spaces which preserves points, straight lines and planes. The main point in affine transformation is that sets of parallel lines remain parallel after an affine transformation. It does not preserve angles between lines or distances between points, though it does preserve ratios of distances between points lying on a straight line.Affine transformations include translation, scaling, similarity transformation, reflection, rotation, shear mapping, and compositions of them in any combination and sequence.

This affine transformation has a widely use in which the most common kind is digital image processing. Since the Digital Image Processing has wide applications and tremendous prospects in various fields such as "Image sharpening and restoration","Medical field","Remote sensing","Transmission and encoding","Machine/Robot vision". This affine transformation is of great significance too.

The affine transformations are analogous to printing on a sheet of rubber and stretching the sheet's edges parallel to the plane.This transform relocates pixels requiring intensity interpolation to approximate the value of moved pixels.

In this report ,we realize the face conversion based on affine transformation. With previous work like "A Registration-Based Propagation Framework for Automatic Whole Heart Segmentation of Cardiac MRI", We make an inner analysis and research on the algorithm. we optimize our algorithm to shorten the running time and to improve accuracy of the conversion.The human-computer interact interface provide a practical application for our work.This offers users an good mutual interface to reproduce our work.

3 ALGORITHM

3.1 LOCAL AFFINE TRANSFORMATION

The first algorithm we introduce is Local Affine Transformation method [3]. In the paper, the Local Affine Transformation is used for registration. But in our assignment, we will use this method to do deformation on the human face. As we know, a gloabl affine transformation is always challenged to give good deformation of substructures. So it is neccessary to introduce the local affine transformation to give a better deformation performance. Here, we give the details of the algorithms. We first define a set of predefined local regions (a region may be a point, line ,or polygons etc.) $\{V_i\}$ and let $\{G_i\}$ be the corresponding affine transformation function of each region V_i. Then we can have the global transformation, a direct fusion based on the distance weighting interpolation is used

$$T(x) = \begin{cases} G_i(x) & x \in V_i, i = 1,2,3...,n \\ \sum_{i=1}^{n} w_i(x) G_i(x) & x \notin \bigcup_{i=1}^{n} V_i. \end{cases}$$

where $w_i(x)$ is a normalized weighting factor related to the distance between point x and region V_i

$$w_i(x) = \frac{1/d_i(x)^e}{\sum_{j=1}^{n} 1/d_j(x)^e}$$

where e cantrol the locality of the affine transformations.

This is the Local Affine Transformation, and we will apply this method in our deformation process. First of all, we will choose the predefined local regions on both the reference image, i.e. babnoon image and the original image, i.e. the human face (Like the eye center, the left/right corners of mouths). Second, we will compute the transformed position $T(x)$ of every pixel of reference image. Third, we will use interpolation to compute the pixel value of each transformed position $T(x)$. Finally, we will assign every pixel value of x to a new image with the same shape of the reference image. Then, we will get the deformed image.

3.2 MOVING LEAST SQUARES DEFORMATION

The second method is Moving Least Squares Deformation [2]. In fact, the deformation process of this method is the same with the Local Affine Transformation method, the difference between them is the diverse of the transformation function $T(x)$.

The Moving Least Squares Deformation method employ Moving Least Squares [1] to construct the deformation function f. Let p be a set of control points and q the deformed positions of the control points p. (Not regions) .Given a point v in the image, we solve for the best affine transformation $l_v(x)$ that minimizes

$$\sum_i w_i |l_v(p_i) - q_i|^2 \tag{3.1}$$

where p_i and q_i are control point and the deformed point, and the weights w_i have the form

$$w_i = \frac{1}{|p_i - v|^{2\alpha}} \tag{3.2}$$

Because the weights w_i in this least squares problem are dependent on the point of evaluation v, we call this a Moving Least Squares minimization. Then we can define the deformation function $f(v) = l_v(v)$, it is clear when $v \in p_i$, $f(p_i) = q_i$. Since the $l_v(x)$ is affine transformation,it has the form of

$$l_v(x) = xM + T \tag{3.3}$$

Then by solving the Least Squares problem, we will get

$$T = q_* - p_* M \tag{3.4}$$

where $p_* = \frac{\sum_i w_i p_i}{\sum_i w_i}$,$q_* = \frac{\sum_i w_i q_i}{\sum_i w_i}$,and substitute the T we get

$$l_v(x) = (x - p_*)M + q_* \tag{3.5}$$

Define $\hat{p}_i = p_i - p_*$,$\hat{q}_i = q_i - q_*$. And the least square problem will be

$$\sum_i w_i |\hat{p}_i M - \hat{q}_i|^2 \tag{3.6}$$

Then solve it we will get the close form of M

$$M = (\sum_i \hat{p}_i{}^T w_i \hat{p}_i)^{-1} \sum_j w_j \hat{p}_j{}^T \hat{q}_j \tag{3.7}$$

Then we will get the deformation function $f(v)$

$$f(v) = \sum_i A_i \hat{q}_i + q_* \tag{3.8}$$

where $A_i = (v - p_*)(\sum_i \hat{p}_i{}^T w_i \hat{p}_i)^{-1} w_j \hat{p}_j{}^T$. So with this transformation function, we can do the deformation following the steps in Section 3.1.

3.3 Image Morphing

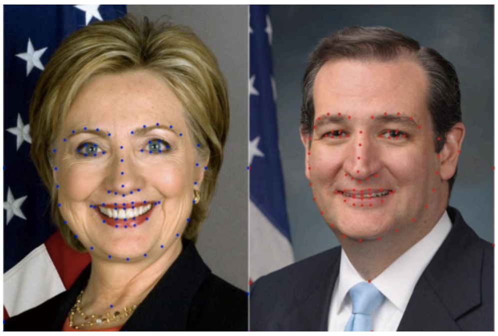
The third thing we want introduce is the Image Morphing.Morphing is a special effect in motion pictures and animations that changes (or morphs) one image or shape into another through a seamless transition.(wikipedia) In our report, we are faced with a deformation from human to babnoon, so it is natural to use the image morphing techniques to some extent. The idea behind the image morphing is simple. Given two images I and J we want to create an in-between image M by blending images I and J. To do the morphing, we will need to find some control points which is the same as the methods above. For example we have x_i in I and x_j in J, then we can get $x_m = (1-\alpha)x_i + \alpha x_j$ in M. Then we can compute the pixel value $M(x_m)$ by interpolation. We give more details by describe the method step by step

1. Finding the corresponding control points on image I and image J. For example, in Figure 1(a) , we give the corresponding points on Hillary Clinton and Senator Ted Cruz faces by facial keypoint detection.

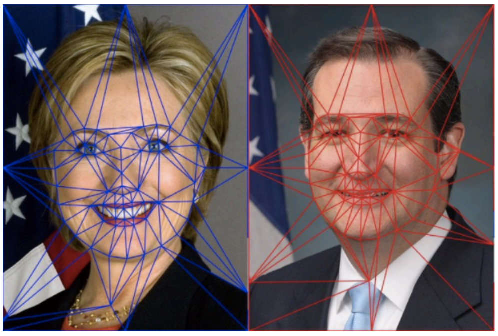
2. Then, given these corresponding points, compute the corresponding points in M, then divide the image into many triangle regions, here we will use the Delaunay Triangulation to do this. In Figure 1(b) we show the triangles.

3. Doing affine transformation on each corresponding triangles between I and M, J and M. Then doing the affine transformation on I and J and get two morphed image I_m and J_m. By blending them with $M = (1-\alpha)I_m + \alpha J_m$. We will get the morphed figure. The example is in Figure 1(c)

4 Some Study On Algorithms



(a) The facial keypoints



(b) The Delaunay Triangulation

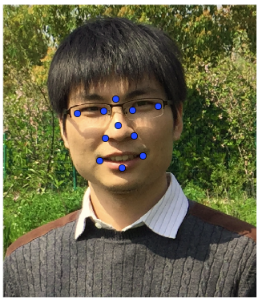
(c) The morphed image

Figure 3.1: The example plots from http://www.learnopencv.com/face-morph-using-opencv-cpp-python/

4.1 Control Regions (Points) Effect And Comparsion of Algorithms

In this section, we test how different choice of control regions will affect the accuracy of the deformation. Here, we test three different choice of control regions which will be shown in the plots. Here we will use the Local Affine Transformation method to analysis.

First, we use the simple control points set *left_eye_left_corner,left_eye_right_corner, right_eye_left_corner,rig* and I plot these points on both human face and babnoon face in Figure 1(a) and Figure 1(b), the deformed image in Figure 1(c). From this plot we can find that the nose and the eyes is



(a) The human face

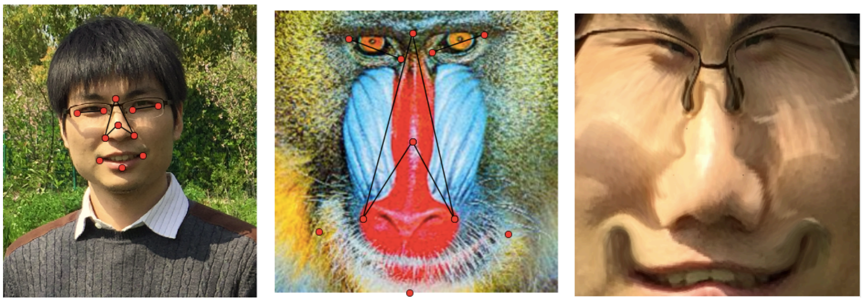
(b) The babnoon face

(c) The deformed face image

Figure 4.1: The deformation on simple control points

(a) The human face (b) The babnoon face (c) The deformed face image

Figure 4.2: The deformation on control points with control lines

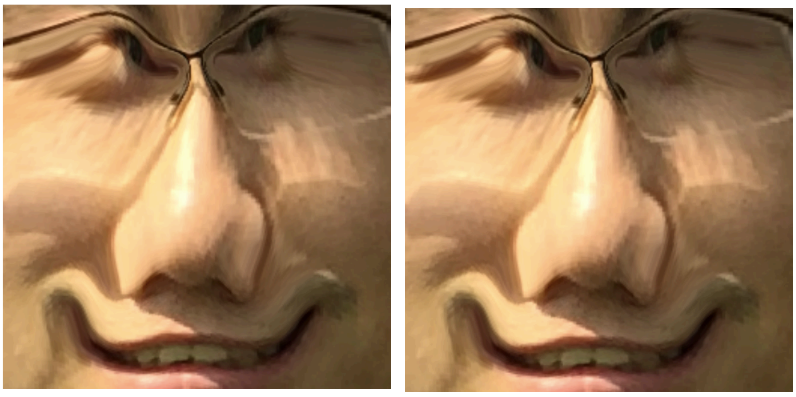


(a) The human face (b) The babnoon face (c) The deformed face image

Figure 4.3: The deformation on control points with control lines

not so accurate in the deformed plot. We want to solve the problem step by step, first we and lines between the nose center and nose left,nose right. Then we get a better result in Figure 4.2 In this case, we can find the nose looks like better than the face in Figure 1(c). The problem in Figure 1(c) is that the nose is bigger than we expect, so we just add the lines to make the shape of it more accurate which is shown in Figure 2(c). Then, we want to make the eyes look better , so we add lines inside every eye. By adjusting the control regions, we can find the final image is pretty good and accurate. From the process we change the control regions, we can find the choice of control regions actually affect the performance of our deformation.

Then we give the deformed picture generated by the MLS method in Figure 4.1. We can find that the results with simple control points set is almost the same by using these two methods. The MLS method run faster than the LAT method(no so sigficant). But, the LAT method can be generalized to using control regions while the MLS method can not do this directly.



(a) The LAT Method (b) The MLS Method

4.2 Distance Function and Parameter e

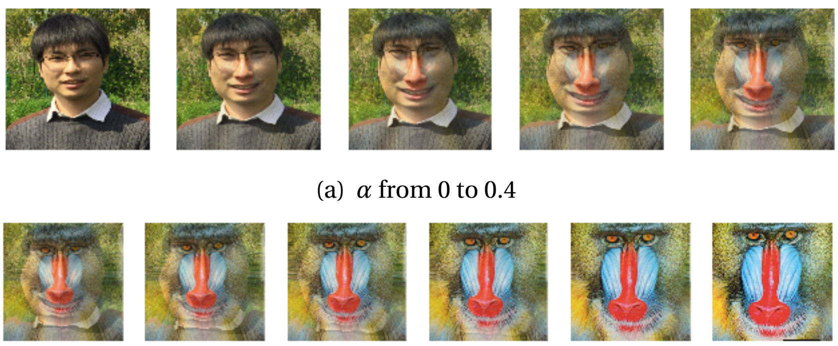
In this part, we discuss how distance function and the parameter e in w_i affect the deformation. We test two distance function using $L2$ and $L1$ norm. And we choose $e \in 0.5, 1, 2, 5$ From the result, we can clearly find the choice of e actually affect the performance significantly. As we mentioned above, e controls the locality of the affine transformations. From the results we can also prove it, when e is small(0.5,1) the deformed image is similar to the orginal image and just like doing some translation on the original face image, in another word, it use low locality of the transformations. But when the $e = 2$ is appropriate the deformed figure have the local property in reference image but still have the original face. And when the e is too big like 5, it will make the deformed image so deformative and lose the global feature.
So we can find, it is improtant to choose a good e for good deformation performance.

(c) $e = 0.5$ (d) $e = 1$ (e) $e = 2$ (f) $e = 5$

Figure 4.4: The plots with different e

4.3 Morphing process

In this part, we will show the Morphing process by altering the α value from 0 to 1. We show the morphing process in Figure 4.5, in fact, if we ignore the pixel value from the reference image when the $\alpha = 0.2$, the shape of the human face is prettry like babnoon but still with the human feature which can be seen as a goood deformation here. So to some extent this method can be seen as a good method for this deformation problem.(The algorithm is fastest.)



(a) α from 0 to 0.4

(b) α from 0.5 to 1

Figure 4.5: The morphing process

5 How to use the GUI

We move this part to the file readme.pdf

6 Conclusion

In this report, we implements three methods to solve the deformation problem. The first algorithm is Local Affine Transformation method, the second method is Moving Least Squares Deformation method, the third is image morphing. All of them have a good performance. And we did some study on these algorithms. We also design a GUI for user to do the deformation. But there are still somethings we can do such as how to choose the best control regions.

References

[1] David Levin. The approximation power of moving least-squares. *Mathematics of Computation of the American Mathematical Society*, 67(224):1517–1531, 1998.

[2] Scott Schaefer, Travis McPhail, and Joe Warren. Image deformation using moving least squares. In *ACM transactions on graphics (TOG)*, volume 25, pages 533–540. ACM, 2006.

[3] Xiahai Zhuang, Kawal S Rhode, Reza S Razavi, David J Hawkes, and Sebastien Ourselin. A registration-based propagation framework for automatic whole heart segmentation of cardiac mri. *IEEE transactions on medical imaging*, 29(9):1612–1625, 2010.