Face Image Fitting*

Kangcheng Hou[†] Shuqi Wang[‡] Tianhao Wei [§]

June 21, 2018

Contents

1 Introduction

A large proportion of images people uploaded is facial images. Motivated by this, in this course project, we want to develop a software that ease the process of editing facial image. The core part of our course project is face fitting, i.e. given a image, find the corresponding 3D model of the image. We also build two applications based upon the face fitting module.

The first one is to substitute the mouth in one image to another. The second one is to provide accessible way to edit face.

1.1 Structure of the report

In the following parts of the report, we will first introduce different basic components related to facial animation we have implemented. Then we will talk about different interesting facial animation application using some of basic components we have implemented. At the end of the report, we will discuss further possible impromments of our project.

2 Model fitting

2.1 Introduction

Given a photo, we want to find a facial model fitted to the image. The typical method previously used is to first find the 2d feature points of the face and fit the 3d model to the image.

[*] This is the first time we try to write the course project report in English. Your understanding is much appreciated.

[†] kangchenghou@gmail.com

[‡] wangshuqi.cn@gmail.com

[§] phi.wth@gmail.com

0.5

(a)

t

0.5

(b)

t

Figure 1: The image, feature points and the fitted model

We first use state-of-the-art commercial software provided by face++[1] to track the feature points $L^{(1),\cdots,(K)}$ which can be seen from the figure. Note that we include the points on the contour. As we will see later, feature points on the contour will be very helpful for fitting the model. Then we will fit a 3D model based on these points.

2.2 Parametric Face Model

Parametric face model use a low dimensional vector to represent different face model in the database. Previously methods use PCA or tensor decomposition to represent face model using low dimensional vectors. We use the Basel Face Model available online[2]. The basel face model provides us the mean face model $\overline{M}$, the identity principle components $\mathbf{V}_{\text{id}}$, the corresponding identity variations σ^{id}, the expression principle componets $\mathbf{V}^{\text{expr}}$, the corresponding expression variations σ^{expr}. The generating morphable model we use is

$$M = \overline{M} + \mathbf{V}^{\text{id}} \times \mathbf{w}^{\text{id}} + \mathbf{V}^{\text{expr}} \times \mathbf{w}^{\text{expr}} \quad [\mathbf{w}^{\text{id}}, \mathbf{w}^{\text{expr}}] \sim \mathcal{N}(0, \text{diag}([\Sigma^{\text{id}}, \Sigma^{\text{expr}}]))$$

Given the generating process of the 3D model, to fit the model to the 2D landmarks, we define the energy function to be optimized as follows.

$$E_{\text{fit}}(\mathbf{w}) = \sum_i ||L_i - s \cdot (R \cdot M_{c(i)} + t)||_2^2 + \lambda \frac{1}{2} \mathbf{w}^\top \Sigma \mathbf{w}$$

where s, R, t is camera parameters which we will introduce later, the parameter λ balances the fitness to the data and the prior constraint and $c(i)$ find the corresponding index in the model w.r.t the ith landmark.

add "for convenience, we will denote $\mathbf{w}$ as the concatenation of $\mathbf{w}^{\text{id}}$ and $\mathbf{w}^{\text{expr}}$

2.3 The General Procedure

There is no closed form solution to the energy function mentioned above. We adopt the coordinate descent method to optimize this energy function. Coordinate descent works in similar way of Gibbs Sampling, i.e., iteratively fixing other variables except one and optimize the energy function w.r.t that variable.

[1] https://www.faceplusplus.com.cn/

[2] https://faces.cs.unibas.ch/bfm/

Our algorithm can be summerized as follows:

```
Algorithm 1: Fit the model to a single image
  Input: facial landmarks $L^{(1),\ldots,(K)}$ and the shape PCA model
  Output: shape $M$ that best fits the landmarks and the corresponding
          camera parameters $s, R, t$
  Set $\mathbf{w} = 0$;
  repeat
      Set $\hat{M} = \bar{M} + \mathbf{V} \times \mathbf{w}$;
      Find the camera parameters $s, R, t$ from $\hat{M}$ and $L^{(1),\ldots,(K)}$ by using
        the least squares method;
      Project all vertices of $\hat{M}$ onto the image plane: $\hat{M}' = \Pi_{R,s,t}(\hat{M})$;
      Find the convex hull of $\hat{M}'$ as $H(\hat{M}')$;
      For contour landmarks $L_i$, find correspondence using $H(\hat{M}')$;
      Solve $\mathbf{w}$ using the energy function defined above.
  until $\mathbf{w}$ converges;
```

Knowing this general procedure, we now introduce what algorithm we use for seperate procedure.

2.4 The Camera Model

Before introducing the method of finding the camera parameter s, R, t. We first introduce the camera model we use. Like many previous work, we use the weak perspective camera model.In weak perspective camera model, the transformation between 2D points and 3D points is specified by three parameters: scale s, rotation R, translation s. The transformation is specified as follows:

Figure out this later

$$l = \Pi_{s,\mathbf{R},t}(m) = s\mathbf{R}m + t$$

This is based on the assumption that the object is far enough and no camera distortion is involved which is reasonable for modern camera setting and most applications.

2.5 Finding the camera parameters

Given the vertices on the 3D model $M_1, \ldots, M_N$ and the corresponding landmarks $L_1, \ldots, L_N$, we want to find the camera parameters s, R, t that minimizes the fitting error.

$$s, R, t = \arg\min_{s,R,t} \sum_{i=1}^{N} ||L_i - \Pi_{s,R,t}(M_i)||_2^2$$

SVD decomposition

2.6 Solve the w

Given the corrspondence, the camera parameters s, R, t, we can formulate the problem of solving $\mathbf{w}$ as a least square problem and solve it easily.

3 Joint fitting of the model

In some scenerios where we want to fit models for the same person that appeared in multiple images at the same time. We will need to fit the same identity weights $\mathbf{w}^{\mathrm{id}}$ and fit the expression weights $\mathbf{w}^{\mathrm{expr}}$ for different images. We adapt the previous algorithm as follows.

4 Introduction of the applications

Now we can fit a face model for a single image or fit multiple face models of the same person in multiple images, now we can do some interesting things. In the following sections, we will introduce three applications related to face model fitting and show the versatility of face fitting results.

5 Face Image Editing

Do some introduction of face image editing.

5.1 Lapalacian Mesh Editing

5.2 Face IK

5.3 Results

6 Mesh Deformation

To enable easy facial animation, we tried out different methods of mesh deformation. We have tried the following methods:

- Face IK
- Laplacian Mesh Editing
- ARAP Mesh Deformation

The setting of our application is as follows, we specify some constraint pairs on the face mesh. Then we want to find a new mesh that satisfies the constraint while **looks similar** to the original mesh. Vertices on the original mesh is $\hat{\mathbf{x}}$ and the unknown vertices position of the deformed mesh is $\mathbf{x}$. And we can write the deformation field as $\hat{\mathbf{x}} - \mathbf{x} =: d$. We can think of the deformation process in two complementary way:

- Think about $d = \hat{\mathbf{x}} - \mathbf{x}$. Scattered data interpolation problem, where the constraint points specify constraints on the field d. Say $d_1 = u_1, \cdots, d_n = u_n$.
- Think about the unknown vertices $\mathbf{x}$. The reconstructed mesh should satisfies the constraint while preserving some kind of details.

6.1 Face IK

6.2 Biharmonic mesh deformation

6.3 ARAP deformation