CS6790 Project Report

R Krishnan CS12B016 Sanjay Ganapathy CS12B027

11 May 2015

1 Introduction

Our goal in this project was to reconstruct the 3D model of the face given certain views of the face and a existing 3D reference model for the face. Given any view in terms of the camera matrix P, we reconstruct the image of the face from that view.

2 Approach

Many papers describe ways to extract the 3D reconstruction of a shape from multiple views of the face (like http://iris.usc.edu/outlines/papers/2010/ch-me-li-si-re-pa-fa-icpr10.pdf). The paper roughly follows the approach of first detecting the landmarks in a face, performing bundle adjustment and locating the 3D coordinates of the landmark points, (upto a homography). Then it uses a generic model of a face, finds the components along which meaningful changes to the face shape are possible, uses these reduced number of components to find a mapping from the reference 3D model to the actual face (using the triangulated 3D facial points) and finally deforms every point in the reference model using the mapping to get an approximate 3D model of the current face.
There are other approaches of 3D reconstruction where instead of deforming the 3D model, they use it as it is and rely on the lighting of the face images to mould the face (like https://homes.cs.washington.edu/ kemelmi/ProjectPages/Faces/paper.pdf) and other approaches like frontalization which just use the reference model itself as an approximation to the 3D model of the face to find the frontal view of an image from a single face image (like http://arxiv.org/pdf/1411.7964v1.pdf).
What we do in the project is a hybrid approach. We first find the camera matrices and the location of landmarks using an iterative resectioning-triangulation procedure, find a homography between the reference 3D model's landmark points and the triangulated landmark points, and use the 3D model and all the reference images to synthesize a view of the face with respect to a given camera matrix. Due to time constraints, we do not deform the 3D model to fit the face better.
We take 51 landmark points in the face (removing the points from the boundary) and use a dataset with images of people taken in different poses. We use a landmark detector detecting 68 points shown below to detect landmarks in the face images and store these prior to 3D reconstruction.
We also reconstruct only the central part of the face since we are using only the landmarks present there and the boundary points are inconsistent across images.

3 Resectioning-triangulation

The algorithm we use is as follows:

```
Initialize P (or X)
Loop till convergence
    Find X from P and img landmark coordinates (or P from X and img landmark coordinates)
    Normalize X (or P)
    Find P from X and img landmark coordinates (or X from P and img landmark coordinates)
    Normalize P (or X)
Find Homography between X and reference coordinates.
Use the homography and the P matrices along with the reference face model coordinates to
    predict the new face's appearance
```

Many papers solve a minimization problem to compute P and X after triangulation to refine the landmark coordinates. But we just stick with the above alone since we almost obtained a similar value of back-projection error as claimed in the first paper.
Whether to initialize P or X was a problem. We tried doing it both ways (the method we used to initialize P is described below, for X, we just used the coordinates of the landmarks in the reference 3D image). We found out that we had to do solve a minimization problem after the above algorithm in case we did it using P's initialization. So, we stuck with X's initialization which provided decent results.
Once we have the 3D points, P_i's and the homography in place, finding the new image with respect to a given P matrix involves using every point X in the 3D model and projecting it to every image to obtain the average colour of the point and projecting using the input camera matrix to obtain the corresponding pixel's colour.

Let $P_1, P_2, ...P_k$ denote all the camera matrices calculated.
Let P be the new input matrix.
We take a point X on the reference model,
To find the colour of X's projection on P, we do

$$C(PX) = avg(C(P_1X), C(P_2X), C(P_3X), ..., C(P_kX))$$

Now, we'll look at the different steps used in our project

4 Initializing P

We first assign one image as the principal image and denote it's projective camera matrix, call it P1 = [I | O]. Then we find the fundamental matrix between each image and the image corresponding to P1 using the landmark point correspondences and estimate P of the other camera from this. Let's say P' is another camera with image I' corresponding to it. Then we compute F, which is the fundamental matrix between I and I'. From F, we can get an initialisation for P' with a 3-fold ambiguity.

4.1 Computing F matrix

We compute F matrix between two points using the 8-point algorithm. Using the F matrix and assuming one of the cameras is P1, we can get an expression for the other camera, which we use as an initialization for bundle adjustment. Using these corresponding points, we build the constraint matrix and solve for F using DLT.

F is of the form

$$\begin{bmatrix} F11 & F12 & F13 \\ F21 & F22 & F23 \\ F31 & F32 & 1 \end{bmatrix}$$

For corresponding points x = (x1 x2 1) and x' = (x1' x2' 1), we get the following row in matrix A :

[x1 * x1' x1 * x2' x1 x2 * x1' x2 * x2' x2 x1' x2' 1]

We stack this up for every pair to get an equation of the form Af = 0, where f contains the paramters of F, is f = [F11 F12 F13 F21 F22 F23 F31 F32 1 $]^T$

We solve for f by doing singular value decompostion of A and taking the eigenvector corresponding to the smallest singular value. We reshape this to obtain F.

In order to enforce the rank 2 constraint on F, we perform SVD on F.

$$F = U * D * V^T$$

In this we make D(3,3) which is the smallest singular value zero and use this modified D to recompute F. Finally, we recompute F for the unnormalised points.

4.2 Normalisation

To do this however, first we normalise both sets of corresponding points. We do normalisation of points because in theory, the constraint matrix, A in the equation Af = 0 should have one zero singular value and other non-zero singular values. However, in practice this is not true. However, it may happen that A is ill-conditioned and has several singular values quite small relative to the larger ones. This arises due to the fact that the first two coordinates in any point vary over a much larger range than the third corrdinate which we take to be 1 for all the points. We overcome this by first normalising the points. We do this by :

$$\bar{x_{new}} = (\bar{x} - \bar{\mu}) * (\sqrt{2} * n / (\sum \|\bar{x_i} - \bar{\mu}\|))$$

where $\bar{\mu}$ is the centroid of the points. So, we can see that we are first shifting the origin to the centroid of the data and then normalising the points so that the mean distance of normalised points from the origin is $\sqrt(2)$.

This can be represented at a linear transform of the point, say T, a 3x3 matrix. Therefore, the new point $\bar{x_{new}}$ = T*x.

Therefore, the new equation becomes

$$x'^T_{new} * F' * x_{new} = 0.$$

This is equivalent to

$$x'^T * T'^T * \text{F' * T * x} = 0.$$

Therefore, once we solve fo F', we obtain the F we are interested in by F = T'^T * F' * T.

4.3 Computing P from F

We know that

$$P2 = [e']_X e'$$

We can obtain e' from the transpose of the left null vector of F by doing singular value decompostion on F. We can find this P's value with a 3-fold ambiguity.

5 Camera Resectioning

We first initialise the 3D points using the reference 3D model. Then we compute P_j by camera resectioning. We do this by stacking up equations of the form

$$P^j * X_i = x_i^j$$

We use a 11x1 matrix for P, i.e, $P' = [p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}]^T$
where P is

$$\begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & 1 \end{bmatrix}$$

$$X = [X_1\ X_2\ X_3\ X_4]^T$$

$$x = [x_1\ x_2\ x_3]$$

The constraints are :

$$[X_1\ X_2\ X_3\ X_4\ 0\ 0\ 0\ 0\ -x_1X_1/x_3\ -x_1X_2/x_3\ -x_1X_3/x_3]P' = x_1/x_3$$

$$[0\ 0\ 0\ 0\ X_1\ X_2\ X_3\ X_4\ -x_2X_1/x_3\ -x_2X_2/x_3\ -x_2X_3/x_3]P' = x_2/x_3$$

We stack all such equations and update the value of P and then normalize it so that it converges and does not grow arbitrarily.

We get 2 constrains from the above equation. We use all the points and stack 51*2 such equations.

6 Triangulation of points

We have equations of the form

$$PX = x$$

$$P = \begin{bmatrix} R_1 & p_4 \\ R_2 & p_8 \\ R_3 & 1 \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & 1 \end{bmatrix}$$

$$X = [X_1\ X_2\ X_3\ X_4]^T = [X'^T\ X_4]^T$$

$$x = [x_1\ x_2\ x_3]$$

X is unknown. We assume $X_4 = 1$. So we need to solve for 3 values. The equations we get are

$$\frac{R_1 X' + p_4}{R_3 X' + 1} = x_1/x_3$$

$$(R_1 - (x_1/x_3)R_3)X' = x_1/x_3 - p_4$$

and

$$\frac{R_2 X' + p_8}{R_3 X' + 1} = x_2/x_3$$

$$(R_2 - (x_2/x_3)R_3)X' = x_2/x_3 - p_8$$

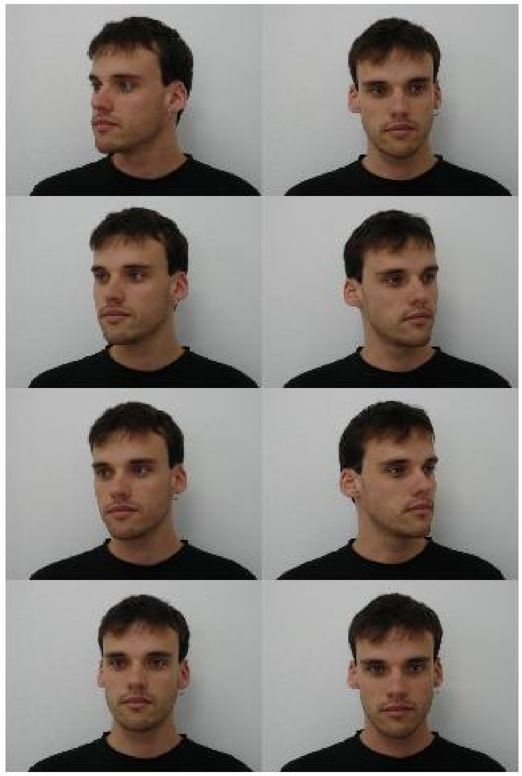
So we have 2 equations from one P. If we use all P's, then we'll have sufficient number of equations to solve for X'.

7 Finding Homography

Having now computed the 3D landmark points, we use these to find a 3D homography from the landmark points on the 3D model of the face to the reconstucted 3D points. We do this using DLT.

8 Results

We take the following as input images



For a few random Ps we obtain by taking one of the image's P and modifying it by adding some rotation and tranlation, we get the outputs to be