APG4011F Bundle Adjustment Report

Tim Marsh

21 April 2015

Contents

List of Figures

1 Introduction

There are four parts to this assignment:

1. **Simulating a camera, image points and object points** Creating the Cameras with made up parameters, Xo, Yo, Zo, κ, ϕ, ω. Creating 30 image points and calculating the corresponding ground coordinates for those image points.

2. **Intersection** After creating a new camera, with its own orientation parameters, calculate the corresponding image point coordinates.

   Then using least squares calculate the best fit ground coordinates for both images.

3. **Resection** Using the object coordinates from part 1 along with the corresponding image points, set up a least squares solution solution to determine the exterior orientation parameters of the image. do this for both images

4. **Bundle adjustment** Treat 25 of the 30 points as control points and the remaining 5 as tie points. Use the collinearity equations to setup a least squares solution to simultaneously determine the exterior orientation parameters of the two images and the object coordinates of the tie points.

These 4 parts are done as one single program written in python that runs from start to finish. the first two cameras are defined with orientation parameters, object points and image points. Each time some action needs to be preformed on a camera a new camera instance is created and the things that need to be calculated are cleared, calculated and saved in the new instance so that they can be compared to the original values.

2 Program structure

im order to store a lot of information in a program without losing any, and being able to access it whenever is necessary. This is done with dictionaries within dictionaries.

The main dictionary is Cameras, Cameras holds the image size as well as the focal length of the camera (the focal length is constant through all images). in the Cameras dictionary are Images there are multiple images (in this case 2) the Images class holds the exterior orientation parameters of that Image(photograph).

Each image has image points and object points, to keep the image points and object points paired up in created and saved them in a ray class. It holds no information other than keeping the image and object points paired up.

Figure 1 shows a very basic representation of the structure.

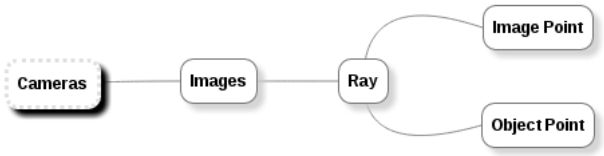

Figure 1: Structure of Dictionaries

3 Background to the Problem

Knowledge of how photogrammetry works is necessary for this assignment. Each image has its image center $Xo\ Yo\ Zo$ as well as rotations around the x y and z axes, $\kappa\ \phi\ \omega$.

4 Method

In the first part ground coordinates are calculated using the collinearity equations. the basic form of the collinearity equations take object points or image points and calculates the other.

$$\begin{bmatrix} X - Xo \\ Y - Yo \\ Z - Zo \end{bmatrix} = kR^T \begin{bmatrix} y \\ x \\ -c \end{bmatrix} \text{ where R} = R\kappa R\phi R\omega \text{ and } k = \text{scale}$$

In the second part this formula is reversed so that image coordinates are being calculated from the object points calculated in part 1.

These are easy steps and involve going through each point in the images and calculating its corresponding point in either the image or object space. Part 3 and 4 are more complex and involve least squares to solve.

Part 3 consists of calculating the exterior orientation parameters for an image having only image and object points. this is done through least

squares.

There are 6 unknowns; Xo, Yo, Zo, κ, ϕ, ω as well as a scale (s_i) for each point (You can see the new scale being brought in inn the last term of each equation below). This means that for every point you add in one more unknown must be added. Each point brings 3 equations:

$$U_i - \tilde{X}_i = \tilde{Y}_i dk + \tilde{Z}_i d\phi + X_o + \tilde{X}_i ds_i$$
$$V_i - \tilde{Y}_i = \tilde{Y}_i dk - \tilde{Z}_i d\omega + Y_o + \tilde{Y}_i ds_i$$
$$W_i - \tilde{Z}_i = -\tilde{Y}_i dk + \tilde{Y}_i d\omega + Z_o + \tilde{Z}_i ds_i$$

There needs to be a minimum of 9 equations to solve the 6 unknowns as well as the scale being brought in. This is no problem for us because we have 30 points, which are more than enough to solve this.

From here we solve the seytem of equations using least squares as we would normal. create the A matrix and ℓ matrix then solve for X.

$$X = (A^T A)^{-1} A^T \ell$$

The X matrix will be our 6 unknowns followed by a scale for each point.

Part 4 is what we call relative orientation, we are taking 2 camera with unknown Xo, Yo, Zo, κ, ϕ, ω and orientating them relative to each other to create a stereo-model. So you would expect there to be 12 unknowns along with the point scales. But this is not solvable because there are to many unknowns. But this is still a solvable problem. We do not need all 12 variables to relativity orientate the 2 images. we only need 5. 5 Rotations or 2 translations and 3 rotations. once the two images are relatively orientated we must transfer them into the object coordinates.

This problem is once again solved using least squares.

5 Results

The results of this are hard to showcase with out some graphical representation. Starting off with a single camera and its image points then creating its object points results in a situation as shown in Figure 2 below. Figure 2 shows 30 points from a single camera as calculated in part 1. This image was generated using the code after part 1.

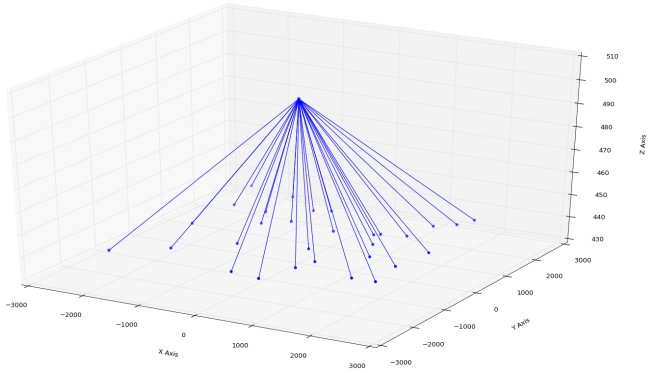

Figure 2: Single Camera View

After part 2 and 3 a second camera had been created and displayed as in Figure 3.

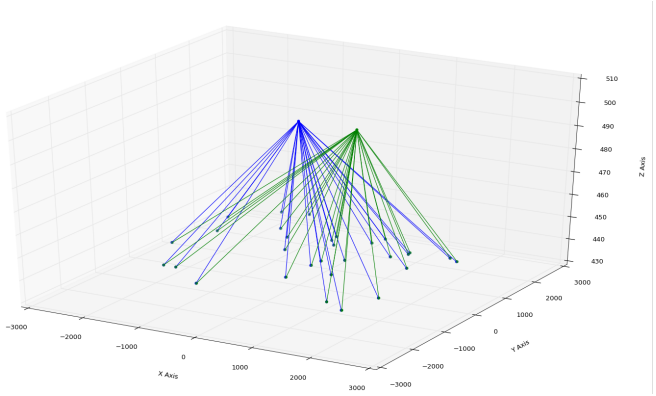

Figure 3: Two Cameras View

After part 4 the image produces was very much the same as in Figure 3.