Task 3&4 Photogrammetry - Algorithms and Libraries

Hemanth Balaji Dandi
hemanthdandi.aero2astro@gmail.com
19 May 2021 - 20 May 2021

1 Aerial Triangulation

Aerial Triangulation is the process of estimating the 3D points of a surface on the ground from a flying aircraft/UAV through photogrammetry.
It is solved by using Bundle Adjustment (which will be gone into with more detail in the below section) wherein the coordinates of the points on the ground, also known as new-points, are determined through triangulation of the orthoimages taken from the aircraft, with the help of reference points called Ground Control Points (GCPs) which help modulate the scale and accuracy of the predicted 3D coordinates.
Nowadays, most of the GCPs are replaced by taking the GPS coordinates/ Geo-tagging by the aircraft/drone itself, called Real-Time-Kinematic (RTK) and Post-Processing-Kinematic (PPK).
The difference between RTK and PPK is the proximity/connection between the drone and it's base-station-For PPK, the raw GPS data from the base-station is sent to the drone, and the drone combines it's own GPS information to geotag points on the image with great accuracy, hence, requiring there to be a strong connection between the drone and the base-station, whereas in the case of a PPK drone, the GPS information onboard the drone and the base-station are combined afterwards to get the accurate geo-tags of the surface. Aerial Triangulation therefore has the advantage of reducing time and manual risk by manual surveyors in remote and steep regions.

1.1 Requirement of Ground Control Points (GCPs)

- Improves the precision of the GPS coordinates and reduce errors
- Helps to compare the current map to existing maps for further analysis

1.2 Inputs

- Overlapping images
- Camera configuration
- GCPs, CPs, MTPs

1.3 Methodology

- Setup GPS and GCPs
  The GPS needs to be setup accurately on the drone to get the coordinates while mapping, Also, it needs to be setup accurately in the base-station so as to improve the location points during post-processing. (For PPK)
  Few GCPs needs to be kept to provide further correction to the GPS coordinates.

1.4 Algorithm

- Establish Ground Control Points (GCPs) and Check points
- Initialize Exterior Orientation elements ($\{X_p$,Y_p,$Z_p\}$ (Projection center coordinates) ,$\{\phi$, ω, $\kappa\}$ (Exterior Azimuth elements)) and Ground points (X,Y,Z):
  Determine the approximate initial value of the exterior orientation elements of all images and the ground coordinates of the object points. The next steps are of least squares block adjustment
- Construct the Error Equation:
  Establishing the error equation about the encryption point and the control point of each image according to the collinear condition.
- Construct and Solve the Normal Equation:
  Establishing the normal equation of error equation point by point and using the elimination and cyclic block method to solve the normal equation.
- Get the exterior orientation elements of each image.(Refined)
- Get the ground points by forward intersection. (Refined)
- Assess the accuracy through checkpoints and manual tie-points (MTP) (during reconstruction).

1.5 Software/Library

- DroneMapper: Gives Orthomaps, DEMs
- Pix4D : Gives Orthomosaic,3D Point Cloud, Topographic, DSM
  - Front overlap should be greater than 75% and side overlap, greater than 60%

2 Bundle Adjustment

The Bundle Adjustment (BA) problem is estimation of the 3D coordinates on the ground (called new points) and camera parameters which includes it's intrinsic values and position given the 2D points in the camera images (large number) taken by the drone.
The Mean Reprojection Error (MRE) is minimized between the observed 2D points and the predicted re projected 2D points estimates from the 3D coordinates so as to give a accurate 3D model map estimate.

The Equation of a BA optimization problem (in a calibrated/non-calibrated camera) is denoted by:

$$\mathbf{BA} = min\, dist\{kP_{image}\, X_{point}, x_{(point,image)}\} \tag{1}$$

where

- k is a Scale factor because of homogeneous coordinates (Dividing by Z)
- P_{image} is the Projection Matrix of the Camera (which needs to be estimated)
  It consists of:
  - Intrinsic and Extrinsic camera parameters

  - Non-linear corrections to distortions like optical,perspective
  - The above distortion parameters depend upon the input $x_{point,image}$ points

- X_{point} is the mapped 3D world coordinate to the input image through P_{image} *(which also needs to be estimated)*
- $x_{point,image}$ is the 2D point in each image (which also includes corrections introduced for precise mapping)
- *dist* Distance between the observed 2D points in the camera image and the reprojected 2D points from 3D by the Projection Matrix is minimized

2.1 Advantages of BA

- : Compatible
  BA can handle a diverse set of problems like different surfaces,cameras,scenes,curves,2D/3D features and error frameworks as well, while simultaneously handling missing data.
- : Precise results:
  Gives accurate results since it's derivation comes from statistical models and optimization.

2.2 Methods for BA

2.2.1 Constraints/Degrees of Freedom (DOF)

There are **(DOF of Camera) x (Number of Camera images) + (DOF 3D world point) x (Number of Points)** Degrees of Freedom in the equation where

- $DOF\ of\ Camera$ is the **Intrinsic + Extrinsic Parameters of the Camera**
  - **Intrinsic Parameters have 5 DOF** which are Focal length in (x,y), Principle point coordinates in pixels (x,y) and Skew (5) (remain the same)
  - **Extrinsic Parameters** have 6 DOF which are 3 Rotation and 3 Translation (could change for very image depending upon the condition of weather for example)
- $DOF\ of\ 3D\ coordinate$ is 3

So we need to take into account a huge number of equations from the matrix multiplication and optimize to find the solution of each of the variables, which would supplement computational time and decrease speed depending upon the number of images captured as well. (Even the scale factor and non-linear distortion parameters are additional DOF that would be needed to estimate in-order to have a lower MRE).

2.2.2 Ways to reduce DOF

- We can get rid of the scale factor by divi

2.2.3 Software

- Large-scale bundle adjustment
  - Language: Python
  - Package: Scipy
  - Link: `https://github.com/scipy/scipy-cookbook/blob/master/ipython/bundle_adjustment.ipynb`

2.3 Library for BA

- Link: `https://github.com/IshitaTakeshi/SBA`
- Language: Python

Method Flow/Functions:

- Initialize class **class sba.core.SBA(viewpoint_indices, point_indices, do_check_args=True)**
  where *point_indices* are 3D points and *viewpoint_indices* are those 3D points that are visible from all viewpoints
- Compute Gauss-Newton update by **compute(x_truepoints, x_predpoints, A, B, weights=None, mu=0.0)**
  where *x_truepoints* Actual 2D keypoints of the shape/structure, *x_predpoints* Reconstructed 2D keypoints by a projection matrix, A Jacobian Matrix based on Rotation and Translation parameters, B Jacobian w.r.t 3D points, *weights* Symmetric weight matrix for Gauss-Newton, mu damping factor for LM
  This returns the updated pose parameters (rotation and translation) thus getting an updated projection matrix, and updated 3D points.

3 Structure from motion

3.1 Definition

Structure from Motion is the process of create a 3D map/model of a scene/structure by combining multi-viewed successive 2D images. The difference between Structure from Motion (SfM) and Multi-View-Stereo (MVS) is that SfM deals with estimating the camera parameters,projection matrix, etc while MVS uses this information to construct a 3D model.

3.2 Algorithm

- Collect multiple images of the scene/object spanning different views such that there is significant overlap between them
- Perform triangulation to convert the images(from the camera attached to the UAV) to a 3D map/point cloud
- This is done by extracting the matching feature points from the successive images, thereby obtaining the (X,Y,Z) locations of each point in the 3D map/cloud.
- Improve alignment of 3D points and location through Bundle Adjustment and through GCPs

A alternative Algorithm instead of the direct approach above is given below:

- FInd correspondences/features (like edges) b/w images using keypoints/descriptors through various feature detectors like SIFT,SURB,ORB such that the features are scale and rotation invariant
- Perform matching between similar features amongst the overlapping images using the descriptors using KLT
- Filter bad matches them by using RANSAC, which helps to remove matches above a certain threshold which are poor.
- Then these are used an inputs to estimate the 3D point locations and camera parameters.
- Bundle adjustment to improve refinement of points.

3.3 SfM with OpenCV

- Feature Detection and Extraction: Extract key-points and descriptors for using a feature detector (say, SIFT) using **cv2.xfeatures2d.SIFT_create().detectAndCompute(each_image,None)**
- Perform Feature matching between the two images using **cv2.FlannBasedMatcher(index_parameters, search_parameters).knnMatch(descriptor1,descriptor2,k)**
- Filter out worse matches by a distance metric/ratio
- Get the Essential Matrix by **cv2.findEssentialMat(pointsmatching_left, pointsmatching_right, Camera Matrix, cv2.RANSAC, 0.999, 1.0)**
- Extract the Rotational and Translation matrix, ie, Pose, by **cv2.recoverPose(Essential Matrix, pointsmatching_left, pointsmatching_right)**
- Calculate the Projection Matrix for each camera through $[R|t]$
- Perform triangulation to extract 4D homogenous points by **cv2.triangulatePoints(ProjectionMatrixLeft, ProjectionMatrixRight, Transposed_Undistorted_pointsmatching_left, Transposed_Undistorted_pointsmatching_Right)**
- Divide the above vector by the Z axis to get the 3D point cloud.

3.4 Applications

- Othomosaics - Stitching Orthophotos to form a topographical map using Photogrammetry.
- 3-D Point Clouds - Used in Self-Driving cars, structure of trees, rocks, etc.
- DEM - Building high resolution topographical maps which can be used for keeping track of and analysing various surface and environmental variations like mountains,slopes,etc.

3.5 Recent Advances

4 Multi view stereo reconstruction

4.1 Definition

Multi-view stereo reconstruction is the process of constructing/estimating the 3D shape from a group of multi-view taken images under specific conditions using stereo correspondence.
In general this implies that images from two or more cameras are evaluated by an algorithm that tries to compute which pixels correspond to the same physical object.When this matching is done, it is known for each pixel how large it is shifted in other images. By knowing the characteristics of the cameras, these shifts (denoted as 'disparities'), can be converted to real xyz-coordinates. By using all image pixels together a 3D reconstruction of the scene can be obtained.

There are 2 main types of stereo reconstruction based on the optimization criteria used:
Block Matching and Dynamic Programming.

4.1.1 Block Matching

Block matching find strongly correlated pixels/points between two images capturing the same object.
It is further divided into two types:

- Block Matching/Winner-Takes-All,
  This focuses on high texture images (think a picture of a tree) and semi-global block matching will focus on sub pixel level matching and pictures with more smooth textures (think a picture of a hallway).
- Semi-Global Block Matching/Multi-Dimensional Optimization
  This focuses on smooth texture images through sub-pixel matching.
  Provides a tradeoff between Speed and Accuracy

4.1.2 Dynamic Programming/One-Dimensional Optimization

4.2 Algorithm

- Take a large number of images of the object to be 3D reconstructed across multiple angles to capture the entire shape of it's structure
- Calculate the Camera projection model of each image (Intrinsic and Extrinsic Parameters)
- Remove lens distortion for image
- Match similar pixel features b/w images by selecting good similar pixel candidates
- Reconstruct the 3D geometry of the scene/object using the set of images and corresponding camera projection model.
  - Use depth map to reproject onto a 3D space
  - If known camera parameters, then there will be correlation between pixels present in the set of images. Then to decide which pixel-candidates to select, a cost function relating photo-consistency is constructed that minimizes KLD (also knows as an MLE optimization problem) to get the best possible candidates for pixel matching.

4.3 Software/Tools

4.3.1 OpenCV

- Camera Calibration Step
  - Take multiple viewed photos of a chessboard on a white background.
  - Detect Chessboard Corners using **cv2.findChessboardCorners(inputimage, chessboarddimension, None)**
  - (Optional) Refine corner points using **cv2.cornerSubPix**
  - Get Camera calibration parameters (Of Projection Matrix P) : Camera intrinsic parameter matrix, Distortion coefficients, Rotation Vector, Translation Vector using **cv2.calibrateCamera(objectpoints, imagepoints,inputgrayimageshape, None, None)**
  - Get Focal length from Camera matrix obtained in the previous step

- Undistort images using **cv2.undistort(firstimage, Intrinsic Parameter, Distortion coefficients, None, Camera Matrix)**
- Downsample Image to increase computational speed and helps to tune the algorithm during mapping their disparity, which is reprrojecting the same pixel from different views and calculating it's displacement.
- Block Matching using either Stereo Block Matching/ Semi Block Matching:
  Create Block Matching object through (This is SGBM, if you want high frequency,use SBM)
  cv2.StereoSGBM_create(minDisparity,numDisparities, blockSize, uniquenessRatio, speckleWindowSize, speckleRange , disp12MaxDiff, parameter_smooth1,param_smooth2)
- Determine Disparity Map using **cv2.stereo.compute(downsampled_image_1, downsampled_img_2)**
- Calculate a Projection Matrix and use it to reproject image-points into 3D using
  cv2.reprojectImageTo3D(previous_disparitymap, ProjectionMatrix)
- Combine both the 3D reconstructed points and color points as an added dimension to give a 3D point cloud.

4.4 Application/Outputs from MVS

- Depthmaps
- Point Clouds
- Volume Scaler field
- Mesh

5 Multi view geometry

5.1 Definition

Multi-view Geometry is defined as the measures and concepts of geometry used to extract the 3D structure from different image-views through a combination of camera calibration and scene structure information. It is therefore vital in 3D reconstruction of the structure of an object and understanding how different views of a particular object can be mapped into a 3D point depending upon whether the camera is calibrated or uncalibrated.

Few Important concepts are given below

5.2 Intrinsic and Extrinsic Parameter Computation

In this section, both the camera intrinsics and extrinsics are estimated given the 3D points and 2-D points (of say, a Rubik cube given 28 3D points.

DLT Algorithm

Let X be the 3D points and x be the 2-D coordinates, then the 2D maps to 3D via the below equation,

$$x' \; PX' \tag{2}$$

where X' and x' are the homogeneous coordinates of the 3D and 2D points respectively. and P is the Projection Matrix which is of dimensions 3x4 . To remove the similarity, we use DLT.

First, we perform a cross product as follows,

$$x'xPX' = 0 \tag{3}$$

Solving the above equation, we get

$$\begin{bmatrix} 0^T & -X_i^T & y_i X_i^T \\ X_i^T & 0^T & -x_i X_i^T \\ -y_i X_i^T & X_i^T & 0^T \end{bmatrix} . \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} = 0 \tag{4}$$

where 0^T is a 1x4 dimension array of 0s; X is a 1x4 matrix containing $[X_i, Y_i, Z_i, 1]$, and it's multiplied with -1 because the third coordinate of the 2D coordinate vector $[x_i, y_i, 1]$ is set as 1.

Since in the Left Hand Side matrix , the first two equations are linearly independent, while the 3rd equation isn't, so we just use the first two equations to from our correspondences. Since we need minimum 6 points to solve for P. and since we are given with 28 points, so we form a 28-point correspondence set matrix,$'A'$, by stacking the 2x12 Matrix 28 times, to form a 56x12 matrix.

5.2.1 SVD of Matrix A

The equation after obtaining the A matrix is

$$A.P = 0 \tag{5}$$

In, order to solve for A, we decompose A into 3 matrices by SVD, by the following equation

$$A = U\Sigma V^T \tag{6}$$

where U, Σ, V are the left singular, Diagonal containing only the eigen values, and the right singular matrix respectively. The left and right singular matrices are orthogonal to each other and contain eigen vectors corresponding to AA^T and $A^T A$.

We are interested in the right singular matrix, V, the last column of which gives the P matrix, which is your calibration matrix, after reshaping into 3x4.

5.2.2 Minimum number of point correspondences needed for calculating P

The minimum number of point correspondences needed is 6.

5.2.3 K, R and T Parameters

From the obtained P, ie, projection matrix, it is decomposed into K, and $[Rt]$ by a variation of QR decomposition as KR is orthogonal and K is upper diagonal , as in the below equation,

$$P = K[Rt] = [KR][Kt] \tag{7}$$

where K is the calibration matrix, R is the Rotation Matrix, and [K R] has dimensions 3x3, and t is the Translation vector of dimensions 3x1.

5.2.4 Final Results

After obtaining P,K,R,t matrices, then the re-projected 2D points are calculated by

$$x\_reprojected = PX \tag{8}$$

Then, in order to view both mathematically and through vision, the RMS value of error between the re-projected points and the original 2-D points is calculated, to check the intrinsic and extrensic camera parameter matrices.

As we can see, the original(green) is very close to the re-projected 2D points(blue).

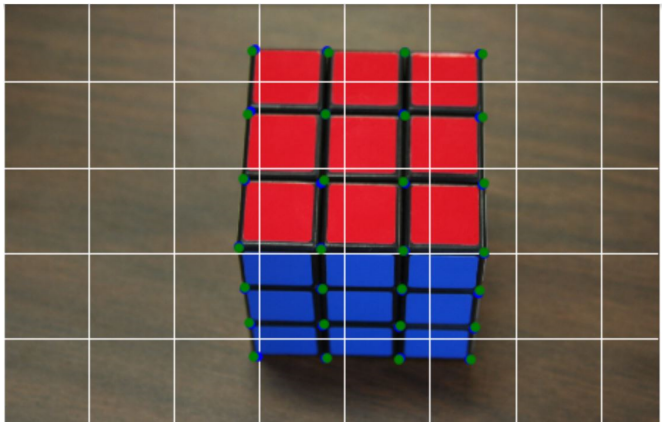

Figure 1: Rubik cube with given and re-projected 2D points on it

Multi-view geometry is broadly separated into two parts- Two-view and Multi-view geometry. Two-view geometry is different perspective views of the same 3D object by 2 images, it's also called epipolar geometry. There are two important terms-Fundamental and Essential Matrix

- Essential matrix requires the camera to be calibrated while the Fundamental matrix does not, the latter is a generalization to both calibrated and uncalibrated case
- Because the Essential matrix contains calibration, it has 5 DOF, while the Fundamental matrix has 8 DOF.
- Coordinates of Fundamental matrix are in pixels while essential matrix are normalized/dimensionless.

5.3 Some Application

5.3.1 Multiview Triangulation

Multiview Triangulation is the process of mapping the 3D structure of a series of images through triangulation by estimating the 3D world points and the projection matrix through minimization of the cost function using Bundle Adjustment (BA).

5.3.2 Structure and motion recovery

It is the process of extracting both the structure and motion from a sequence of images through extending the epipolar geometry between two views to multiple through a sequential algorithm

5.3.3 Motion Estimation

Motion estimation is the process of extracting the trajectory of a moving object in a sequence of continuous frames using Optical flow.

5.3.4 Affine reconstruction from affine cameras

Affine reconstruction is the conversion/modification of projective transform to euclidean transform reconstruction based on camera motion/scenes between two uncalibrated camera views.

5.3.5 3D – 3D registration and perspective matching

3D registration involves obtaining the 3D shape of an arbitrary object through 3D sensors which obtain the structure's 3D coordinates, then using multiple 3D images, obtain the shape of the object. It's useful for pinpointing the location of the 3D objects, matching as well as inspection for defects. 3D perspective matching is when the 3D shape/pose of a particular 2D region of an arbitrarily shaped object is extracted with a single camera. This can be used for 3D alignment, making a drone pick an object,etc.

6 Point clouds (PDAL, GDAL)

6.1 GDAL

- GDAL is Geospacial Abstraction Library
- Useful for raster and vectorized geospacial data

6.2 PDAL

- PDAL is Point Data Abstraction Library
- Used for Point Cloud estimation
- Doesn't have a friendly GUI

7 Useful GeoSpacial libraries (Python)

- Geopandas
- Shapely
- Rasterio
- pyproj

8 References

- `https://people.inf.ethz.ch/pomarc/pubs/HeydenPollefeysCVPR01.pdf`
- `https://www.cartesia.org/geodoc/isprs2004/comm3/papers/338.pdf`
- `https://www.3dflow.net/elementsCV/S4.xhtml`

- https://lear.inrialpes.fr/pubs/2000/TMHF00/Triggs-va99.pdf
- https://arxiv.org/pdf/1912.03858.pdf#:~:text=Bundle%20adjustment%20describes%20the%20sum,world%20frame)%20and%20camera%20parameters.
- file:///C:/Users/User%20Default/Downloads/Multi-View_Triangulation_Systematic_Comparison_and.pdf
- https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/ZISSERMAN/bundle/bundle.html
- https://www.youtube.com/watch?v=JJXwZiqe_Do&list=PLgnQpQtFTOGRsi5vzy9PiQpNWHjq-bKN1&index=39&ab_channel=CyrillStachniss
- file:///C:/Users/User%20Default/Downloads/forests-08-00068.pdf