3D RECONSTRUCTION OF SCENES AND STRUCTURES USING PHOTOGRAMMETRY

Report Submitted by **Swaha Swayamsiddha**
Department of Civil Engineering,
National Institute of Technology, Rourkela

Under the guidance of **Dr. Salil Goel**
Assistant Professor
Department of Civil Engineering
Indian Institute of Technology, Kanpur

ACKNOWLEDGEMENT

This project contains the results of the work done during the last 6 weeks (May 16 to June 30, 2018) in the Department of Civil Engineering, Indian Institute of Technology, Kanpur.

I wish to record my gratitude to Dr. Salil Goel, Assistant Professor, Department of Civil Engineering, IIT-Kanpur for his constant guidance and inspiration. The interest for Geoinformatics which he created even in this brief time drove me to complete this project. I will be forever grateful to him for trusting me with this project.

My thankful regards to the Director, IIT-K, the Dean of Academic Affairs, IIT-K, the Head of the Department for Civil Engineering, IIT-K and the research office for their constant encouragement towards research.

I am thankful to Prof. Mahabir Panda (HoD, Civil Engineering) and the Department of Civil Engineering, National Institute of Technology, Rourkela for allowing me to carry out this project.

OBJECTIVE

Having been introduced to the basics of surveying during the course of Engineering, this project appealed to me as a way of getting to know more about geomatics and its advanced methods.

The objective of this project is to obtain a 3-dimensional point cloud model of a scene from the multiple 2-dimensional images of the scene, taken from a single camera. It is necessary to have some degree of overlap between the photographs for this process to work. The only requirement in this project is that it must be possible to record the scene or the structure, photographically. This 3D reconstruction in digital form stores information, which can also be reassessed later.

Contents

1 Introduction

Geoinformatics is the branch of Civil Engineering that analyses problems related to geography, geosciences and other related branches. It is widely based on Information Science and technology and has helped ease the processes, accuracy and time involved in obtaining information on the aforementioned fields.

Photogrammetry is a branch of Geoinformatics and involves the concept and practice of obtaining measurements from photographs. It includes methods of image measurement and interpretation in order to derive the shape and location of an object from one or more photographs of the object. This helps in storing information which can be reassessed at a later time. However, the primary purpose of this method is the three dimensional reconstruction of the object in digital or graphical form. This is what we aimed to achieve through the undertaken project.

3D reconstruction of scenes and structures through photogrammetry has been carried out by different researchers over the years with many variations, i.e., usage of one camera or a pair of stereo cameras or multiple cameras, variation in the number and type of known and unknown parameters, usage of readily-available multiple images from various perspectives and cameras taken from the internet and so on. In this paper, **we intend to arrive at a 3D model of a scene by calibrating a single camera, and further using that camera, and its available parameters in the reconstruction, by taking multiple images of the required scene from various angles.**

Before going into what the emboldened statement means, let us first understand how reconstruction from various 2D photographs works. To understand this, one needs to have some basic knowledge about epipolar geometry, triangulation and bundle adjustment. The process involves calibrating the camera to be used, which will be discussed in the second section, then detection and matching of keypoints, sparse reconstruction, followed by dense reconstruction. These may be accompanied by steps like depth map inpainting and merging into the global point cloud to obtain the 3D model from the reconstructed point cloud.

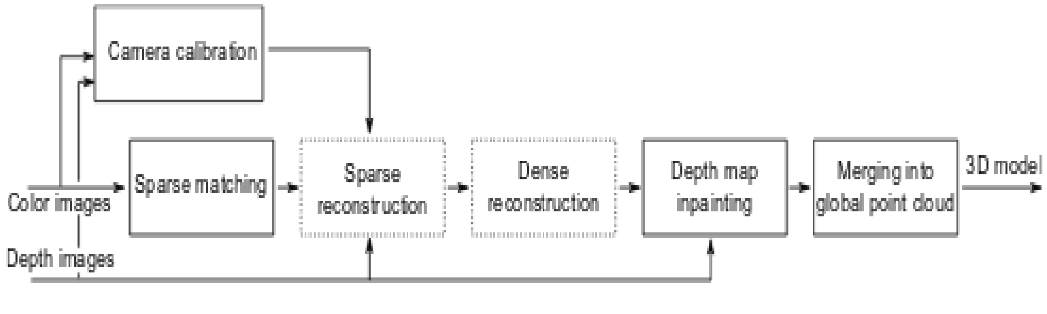

Image 1 - Steps involved in 3D Reconstruction

Epipolar Geometry passes some constraints on how one can determine the object point, after determining the image points and the pose and perspective center of the cameras. It also states that it is impossible to determine the position correctly, using only one image; one requires a minimum of two images to triangulate the required object point. The perspective center of the camera is joined to the image point,(of the point that one wants to reconstruct) on one of the image plane. Then the camera pose and the respective image plane changes, and the procedure is repeated. The second image should also have the point in question for this method to work. On joining and extending these lines, they are found to meet at the point, which gives the corresponding 3D point. This process of determination of the 3D point is called **Triangulation**. On applying this method to all detected and matched keypoints, across the images, one obtains a sparse point cloud reconstruction of the 3D object.

Bundle Adjustment takes these triangulated object points and refines and optimimally adjusts them, in the world coordinate system. It uses non-linear least squares method to minimize the error between the observed object points or image locations and the predicted object points. It takes into account the repetition or overlapping in the image planes to accurately restructure a scene.

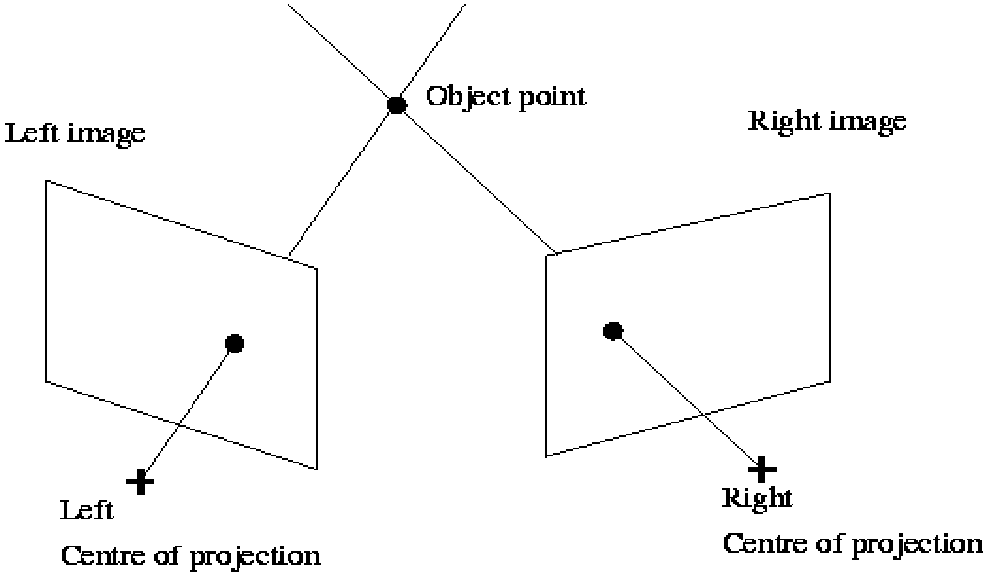

Image 2 - Application of Epipolar Geometry in Triangulation technique

This project was undertaken using a few prerequistites. Python was the computer language used and supporting libraries of Numpy and Matplotlib, and a few other modules were also used. The OpenCV library, which defines a few function to be used in camera calibration and 3D reconstruction, was also used.

In Section 2 of this article, we go through the process of calibrating a camera to obtain its intrinsic parameters.

2 Camera Calibration

Camera Calibration is the process of determining the intrinsic parameters of a camera using special patterns, also known as calibration patterns. Camera calibration provides us with details like the focal lengths, the coordinates of the perspective center, the scaling factor, the camera extrinsics; like the rotation and the translation vectors, and the distortion coefficients. Apart from using patterns to calibrate cameras, one can also use methods based on active vision or use self-calibration to calibrate a camera. In our project, we used a calibration pattern.

2.1 Parameters of the Camera

2.1.1 Camera Intrinsics

The intrinsic matrix of the camera is obtained through calibration. The intrinsic parameters include the focal length of the camera, the optical center (principal point) and the skew coefficient. The 3x3 camera intrinsic matrix, K, can be defined as:

$$\boldsymbol{K} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

where f is the focal length and c$_x, c_y$ are the coordinates for the optical center. Skew is taken as zero.

2.1.2 Camera Extrinsics

The extrinsic parameters of the camera include a rotation vector and a translation vector. They help define the orientation and the position of the camera.

$$[R \mid t] = \left[\begin{array}{ccc|c} r_{1,1} & r_{1,2} & r_{1,3} & t_1 \\ r_{2,1} & r_{2,2} & r_{2,3} & t_2 \\ r_{3,1} & r_{3,2} & r_{3,3} & t_3 \end{array}\right]$$

Here, R is the Rotation Matrix and t is the Translation Vector. They have been augmented to form the camera extrinsic matrix, as denoted by [R—t].

2.2 Distortion

There are two types of major distortion; radial distortion and tangential distortion. The camera model should take into account the distortion of the lens to accurately represent the camera. The distortion coefficients are five parameters, which need to be taken into account to arrive at the correct parameters of the camera.

$$Distortion\ coefficients = (k_1 \quad k_2 \quad p_1 \quad p_2 \quad k_3)$$

2.3 Setup

A calibration pattern of the checkerboard type was taken. Using the pre-built library, OpenCV, we sought to write a program which can provide us with the camera extrinsics and the intrinsics.

Atleast 10-20 images of the pattern are taken from various angles and orientations. One can also opt to fix the camera while moving the pattern to capture the required images. One needs to know the 3D world coordinates of the chessboard corners, in order to calibrate the camera. Hence we arbitrarily assume them to be (X,Y,Z). Now we can assume Z i.e., the pattern plane to be 0; the pattern is assumed to be in the XY plane. The side length of each chessboard square is measured, and the 3D points are passed accordingly. If the side length is not known, one can pass points as (0,0,0),(1,0,0),(2,0,0).. and so on depending on the number of corners in the pattern.

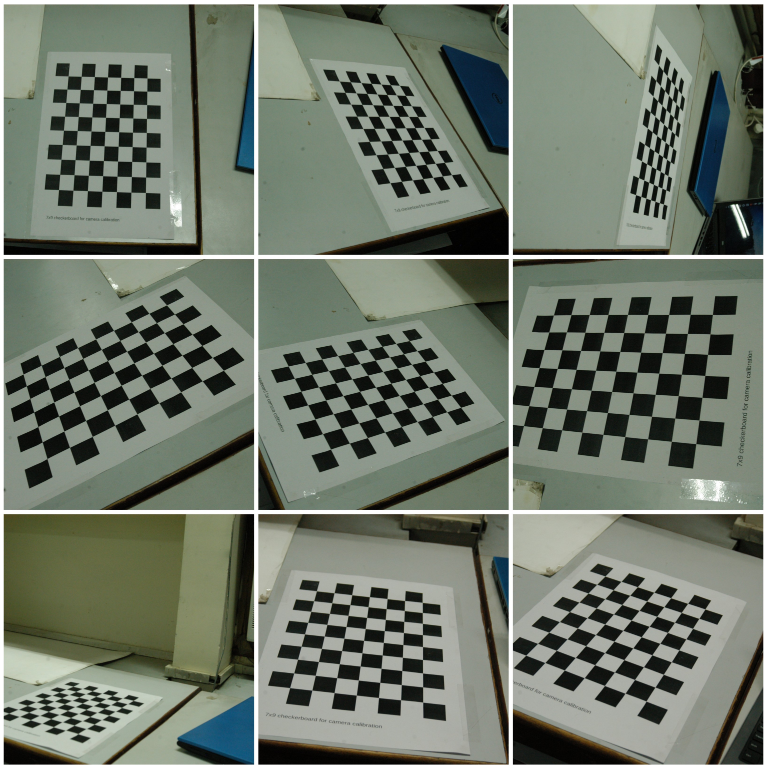

Image 3 - Images of the calibration pattern taken from various positions of the camera.

OpenCV uses Harris Corner Detector to detect the internal corners of the chessboard using the function findChessboardCorners(). It uses the detected image points and the assumed object points to determine the intrinsic matrix and the distortion coefficients. The images can further be undistorted after determining a new, optimal intrinsic matrix by taking the distortion coefficients into account and using the required functions.

The detected points and the actual points may not coincide; the deviation is very small. The reprojection error calculates the gives us an estimate of how large this deviation is, in general, by taking the arithmetic mean of all the observations.

A calibrated camera can further be used in the process of reconstructing a 3D scene. By using a calibrated camera to capture images of the scene, the obtained intrinsic matrix and distortion coefficients can be used to get the 3D points progressively. In section 3 of this article, we go through the process of Sparse Feature Matching.

3 Feature Detection and Sparse Matching

Features are geometric expressions such as points, lines and line segments, which can be matched between images, to establish feature correspondences amongst them. **Feature Detection** is the process of determining and detecting good-enough features for the further processes of feature description and matching. Good features are usually defined by the presence of gradients in at least two different orientations in a chosen patch of the image. The features are determined across two images, the first of which is called the Train image, and the second; the Query image.

After determining features, they are matched to determine which feature comes from which corresponding location across the different images. For this purpose, one uses various **descriptors** such as Multi-Scale Oriented PatcheS (MOPS), Scale Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF), Gradient Location Orientation Histogram (GLOH) and the likes. They are used to detect and compute the features.

On observing the time taken by SIFT and SURF in detecting and describing features, it was seen that SURF was faster, but only by a marginal amount. SIFT however, detected far more points than SURF, for which SIFT was chosen to be used.

3.1 Algorithms Used

1. **FLANN**: FLANN stands for Fast Library for Approximate Nearest Neighbour. It is basically a library that helps perform NN searches in high dimensional spaces. The FLANN matcher consists of many algorithms for matching. In this project, the KNN algorithm is used to match the features.

2. **BF Matcher**: BF Matcher stands for Brute Force Matcher. We start by comparing the first detected feature in the query image, with the first feature in the train image, then make our way through the detected features in the train image, until a match is found, using some distance function. Then this process is again repeated with the second detected feature in the query image.

Of these two, FLANN works faster for larger datasets.

Image 4-A pair of images, i.e., a train image and a query image, to be used in feature detection and matching.

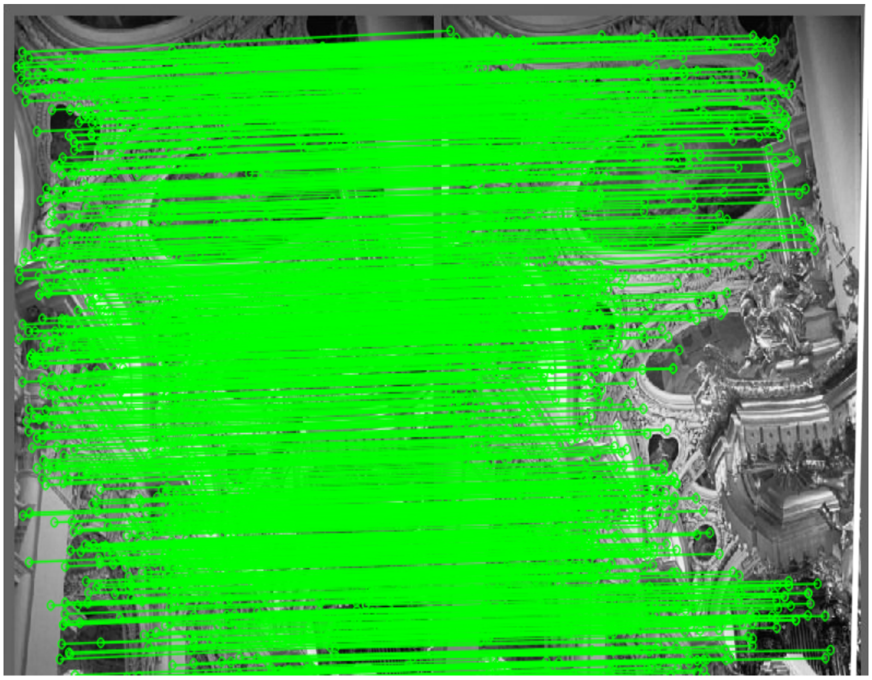

Image 5-Detected and matched features in the train and query images. The red circles denote the detected points while the green lines connect the matched features.

The detected points in the individual images were appended into arrays. The matched points were also entered into a separate set of individual arrays to be further used in the process of reconstruction. They are taken to be the 2D image points.

4 Sparse Reconstruction

Sparse Reconstruction gives us a sparse point cloud model of the scene under the lens. After the features have been detected and matched from the photographs, the feature correspondences need to be triangulated to get the 3D point coordinates of the matched 2D points. The process leading up to this triangulation of points requires many matrices for transformation, such as;

4.1 Fundamental Matrix

This is a 3x3 matrix which relates the corresponding points in both the images, i.e., it is the algebraic representation of epipolar geometry. Fundamental Matrix is calculated when neither the camera intrinsics, nor the camera extrinsics are known. Theoretically, with the help of cv2.findFundamentalMatrix() (this is a pre-defined function from the OpenCV library), it can be determined by passing the matched points in the two images, and by passing the method of determination from among the 7-point, 8-point, RANSAC, and LMedS algorithms.

$$\mathbf{x}'^{\top}\mathbf{F}\mathbf{x} = 0.$$

Here, x and x' denote the points (homogeneous) matched in first image, and their corresponding matches in the second image, respectively. F is the fundamental matrix, and this equation shows the relation between the matched points and the fundamental matrix.

NOTE: RANSAC stands for **Random Sample Consensus** and it is a method which estimates parameters of a mathematical model, iteratively. It helps in robust fitting of data in the presence of many outliers. It first selects the data items, and then estimates the parameters. It then finds out how many fit the model with the above parameter; if it is a large enough fit, then the parameter is accepted, else the process is repeated.

4.2 Essential Matrix

It is a 3x3 matrix which relates the normalized homogeneous matched points between the two images. This relation is given by,

$$(\mathbf{y}')^{\top}\,\mathbf{E}\,\mathbf{y} = 0$$

where y and y' are the homogeneous normalized points, and E is the essential matrix. The essential matrix is calculated after normalizing the coordinates, when the camera's intrinsic matrix is known. When one knows the camera's pose and orientation in two images, then the essential matrix can be calculated, using the formula,

$$\mathbf{E} = \mathbf{R}\ \mathbf{x}\ [\mathbf{t}]$$

where R is the rotation matrix, and t is the translation vector, obtained as and when the camera changes its pose and orientation.

The essential matrix can also be obtained from the fundamental matrix, given that the intrinsic matrix is known. The relation between the three is given by,

$$\mathbf{E} = \mathbf{K}'^{\top}\ \mathbf{F}\ \mathbf{K}$$

where K and K' are the intrinsic matrices of two cameras. When only one camera is in use, we can take them to be the same matrix, K. F and E denote the fundamental matrix and the essential matrix respectively.

NOTE: To obtain the Rotation Matrix and the Translation Vector from any given Essential Matrix, **Singular Value Decomposition**, i.e., SVD is used. SVD is the factorization of the essential matrix; with help from the intrinsic matrix this finally gives us the rotation and translation vectors.

4.3 Projection Matrix

These matrices can be obtained when both the intrinsic and the extrinsic parameters are known. Let, the two cameras used for clicking stereo images have intrinsic matrices K1 and K2, and let the first camera undergo a rotation and translation motion to reconfigure its orientation according to the second camera. This translation can be depicted in a vector, T, and the rotation can be depicted in a matrix R.

When all these parameters are known, the projection matrices P1 and P2 can be derived as;

$$\mathbf{P1 = K1.[I3—0]}$$
$$\mathbf{P2 = K2.[R—T]}$$

Here, — depicts concatenation, i.e., the second matrix gets appended to the first to form a combined matrix. I3 is the 3x3 identity matrix, while 0 is a 1x3 matrix of zeros.

When only a single camera is used, K1 and K2 are taken to be the same, i.e., K. Also, in this case, R and T depict the Rotation and Translation Matrices as the camera undergoes a pose change while clicking the two pictures.

4.4 Triangulation

As described in the Introduction section, triangulation is the process of determining a 3D point from its corresponding 2D points in two different images. For triangulation, we used a predefined function, cv2.triangulatePoints() from the OpenCV libraries, which takes the matched points in each of the matrices and the two, previously obtained, projection matrices, to give the 3D point coordinates in a 4xN array.This can then be reshaped and saved to a file, so as to give a sparse point cloud from the two images.

4.5 Writing Points to a File

The reshaped point coordinates need to be written to a file which can be accessed by a software which can display point clouds. The x,y and z coordinates of the points are separated from the array into separate arrays, for the file to be written, with a .ply or .stl extension.

4.6 Bundle Adjustment

When 3D reconstruction is done from a pair of images, triangulation itself gives the reconstructed features. However, in case of multiple images, bundle adjustment becomes necessary in order to obtain the reconstructed 3D features in world coordinates. Bundle adjustment is simply an optimization procedure which uses the least squares method, through the Levenberg Marquardt algorithm to get the refined parameters.

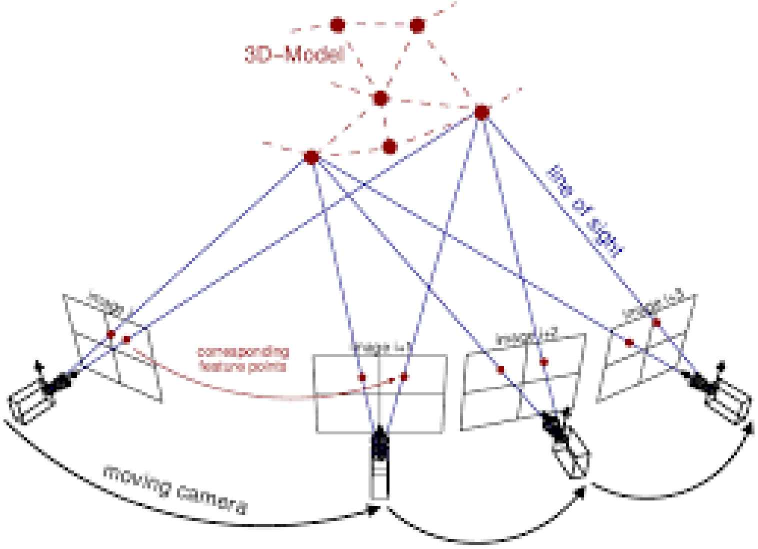

Image 6-The process of bundle adjustment; it takes advantage of the overlapping of features in many images to give accurately determined coordinates of the object points in the world frame.

5 Dense Reconstruction

Sparse Reconstruction gives one a 3D model point cloud model with less number of points. It is usually impossible to get the true geometry of the scene with this few points. Hence, addition of more points is necessary to get a dense point cloud model.

For this, we use KAZE, a algorithm which helps us detect significantly higher number of points, especially in spaces with a non-linear scale.

6 Further Processes

The dense point cloud is then converted into a mesh consisting of triangles. The points are joined together to form these triangular grids, which together form a mesh. Next, texture is rendered onto it with help of the disparity images obtained from the 2D pictures of the scene. Through depth map inpainting and other processes, one can finally obtain the 3D model of the scene.

7 Conclusion

The development of a 3D model will ease many difficult processes of surveying, and will provide the user with accurate and precise measurements, at one's fingertips. The process is also cheaper than the current methods being used, but takes considerably more time.

8 References

P1. Rapid 3D Reconstruction for Image Sequence Acquired from UAV Camera by Yufu Qu, Jianyu Huang and Xuan Zhang

P2. Unsupervised 3D Object Recognition and Reconstruction in Unordered Datasets by M. Brown and D. G. Lowe

P3. Comparision of Bundle Adjustment Formulations by Zach Moore, Daniel Wright, Dale E. Schinstock and Chris Lewis

P4. 3D Reconstruction from Multiple Images by Theo Moons, Maarten Vergauwen, Luc Van Gool

P5. Point Cloud Data from Photogrammetry Techniques to Generate 3D Geometry by David James, Juergen Eckermann, Fawzi Belblidia and J. Sienz

S1. OpenCV-Python Tutorials: http://opencv-python-tutroals.readthedocs.io/en/latest.html

S2. Feauture Matching and Reconstruction: http://nghiaho.com/?p=2379

S3. Essential and Fundamental Matrices: https://stackoverflow.com/questions/31688450/relative-pose-estimation-using-essential-matrix-wrong-r-and-t

S4. Projection Matrices: https://stackoverflow.com/questions/18018924/projection-matrix-from-fundamental-matrix

S5. http://www.cs.jhu.edu/ misha/ReadingSeminar/Papers/Triggs00.pdf

B1. Computer Vision: Algorithms and Applications by Richard Szeliski

B2. Close Range Photogrammetry: Principles, Techniques and Applications by Thomas Luhmann, Stuart Robson, Stephen Kyle, Ian Harley

B3. Multiple View Geometry in Computer Vision by Richard Hartley and Andrew Zisserman

B4.