UID : 115458437
(ENPM 808 Report)

Independent Study Report

Rohith Jayarajan

10 December 2018

Abstract Calibration is the first step towards computer vision tasks that fuse data from multiple sensors to display an intelligent behavior. The information from this sensor network is useful only when each sensor has been calibrated with the others. This independent study conducted in the domain of computer vision focused on the calibration of the VICON Motion Capture System with the Dynamic Vision Sensor. The work in this independent study involved intrinsic calibration for both equidistant and radial-tangential model cameras, development of methods for extrinsic calibration of VICON Motion Capture System with the Dynamic Vision Sensor, and finally reconstruction of a 3D model from known four points on its surface.

Different approaches were taken for each of the aforementioned tasks. The description of the approaches, its reasoning, proof of concept, and limitations have been discussed in this report.

Keywords VICON Motion Capture System · Dynamic Vision Sensor · Calibration

Rohith Jayarajan
UID: 115458437
University of Maryland, College Park
E-mail: rohith23@terpmail.umd.edu

Contents

List of Figures

1 Acknowledgement

I thank Professor Yiannis Aloimonos and Professor Cornelia Fermller for their constant support and insights in computer vision methods that assisted the work done in this independent study. I thank Chethan Parameshwara for the assistance and guidance throughout the course of the independent study. I'm also grateful to the advice and help that Nitin J. Sanket, Snehesh Shrestha, Chahat Deep Singh, Kanishka Ganguly, and other colleagues in the Perception and Robotics Group provided me with. Finally I'd like to thank the department for providing me access to the lab and its cutting edge computer vision equipment without which the difficulties of crossing the reality gap from simulations to the real world applications wouldn't have been studied and worked on.

2 Introduction

The intrinsic parameters of a camera encompasses its focal length, the optical center also referred to as the camera center or the principal point, and the skew coefficient.

- Focal length in optics, is a unit of measure to denote the strength of convergence of divergence of light by the system. f_x and f_y represent the focal length of the camera in pixels.
- The values c_x and c_y represent the principal point in pixels. This denotes the center of the camera in the image plane in pixel units.
- s is the skew coefficient which attains a zero value if the image axes are perpendicular

Computing the intrinsic parameters also involve estimating the distortion parameters of the camera to account for the lens distortion. These are not included in the camera model as the cannot be included in the camera matrix. The distortion depends on the camera model e.g. the radial-tangential model, equidistant model, etc. Real lenses have a mostly radial and some amount of tangential distortion. The parameters k_1, k_2, k_3, k_4, k_5, and k_6 are the radial distortion coefficients whereas p_1, p_2 are the tangential distortion coefficients. The distorted points x_d and y_d are given by the following equation with radial distortion:

$$x_d = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{1}$$

$$y_d = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \tag{2}$$

Here x and y are the in the normalized image coordinates.

In the case of a lens with barrel distortion, the value of $k_1 > 0$ and for lenses with a pincushion distortion, $k_1 < 0$.

Tangential distortions arise when the plane of the camera lens is not parallel to the image plane. The distorted points x_d and y_d are given by the following equation with tangential distortion:

$$x_d = x + [2p_1 xy + p2(r^2 + 2x^2)] \tag{3}$$

$$y_d = y + [p_1(r^2 + 2y^2) + 2p_2 xy] \tag{4}$$

Here x and y are the in the normalized image coordinates and $r^2 = x^2 + y^2$.

The estimation of these camera parameters are through the steps of initialization and non-linear optimization.

Estimating the extrinsics is about finding the transformation that gives a relation between the origin of a sensor to the origin of other sensor(s).

The task of 3D reconstruction is the process of building up a known model in the 3D space given the knowledge of certain points on the object and its 3D point cloud data.

3 DVS Monochrome Camera Instrinsic Calibration

In this task, the process and the results of the intrinsic calibration of the Dynamic Vision Sensor's monochrome camera is discussed. Of various targets available for calibration, the April Grid was chosen over the other targets and the section 3.2 explains the reason for doing so.

3.1 The Intrinsic Calibration Specifications

The monochrome camera in the Dynamic Vision Sensor used had adjustable focal lengths and adjustable exposure settings. Initially the focal length was adjusted to be a wide angle camera focus. This wide angle camera thus obtained was under the category of a fisheye lens as it had high amounts of radial distortion. Later on, the focal length was reduced so as to incorporate a radial-tangential model on to the camera. The Kalibr visual-inertial calibration toolbox developed by the Autonomous Sytems Lab at ETH Zurich was used[2]. MATLAB's Camera Calibration Tool also proved to be useful especially in the case of fisheye effect lens due to their use of Omnidirectional Camera Calibration developed by Davide Scaramuzza.

3.2 The Setup and Process

For the camera intrinsic calibration process, AprilGrid was used as the camera calibration target because of the large number of corners present on it which improve the results of the non linear optimization in the calibration procedure. The figure 1 shows the representation of an April Grid.

The specifications of the April Grid used were:

- The number of AprilTag columns (tagCols): 6
- The number of AprilTag rows (tagCols): 9 The size of the AprilTag, edge to edge (tagSize): 0.0878 meters
- The ratio of space between the tags to edge to edge distance (tagSpacing): 0.297266515

The specifications of the camera model used were:

- Projection model: Pinhole camera model with the returned intrinsic vector values f_x, f_y, c_x, and c_y.
- Distortion model: Radial Tangential distortion model with the returned distortion coefficients k_1, k_2, p_1, and p_2.

Firstly, the images of the target; April Grid board in our case had to be collected. A ROS bag of the raw image data was prepared by recording the */dvs/image_raw* topic from the DVS sensor. A bag file of the image stream was collected for around 40 seconds with an average frame rate of 41 Hz. The camera is moved continuously in front of the April Grid with the target in sight

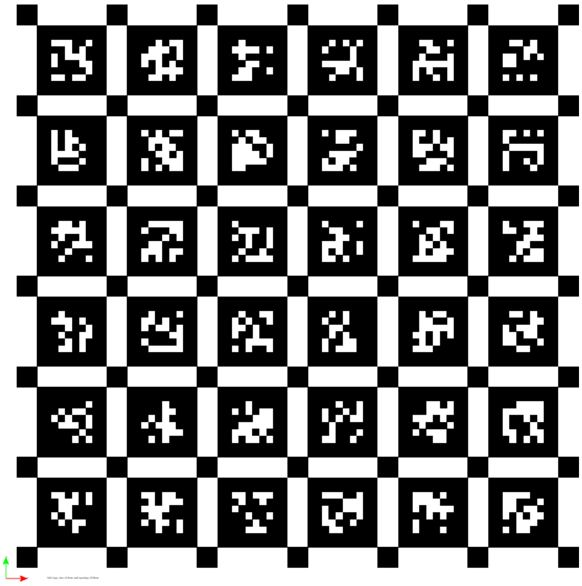

Fig. 1 AprilGrid: The Camera Calibration Target

at all times. The camera is rotated and translated so that enough data points are available for a good convergence of the calibration. Using a lower frequency of the camera stream may help lower the runtime of the optimization of the calibration process. Figure 2 gives an example of one pose of the camera for collecting the image of the caibration target.

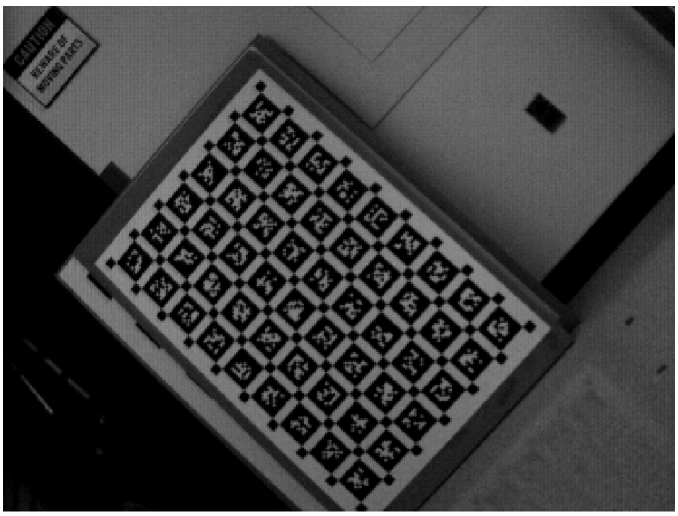

Fig. 2 One of the Images Recorded from the DVS Camera.

Once the bag file of the image stream has been prepared, in one terminal *roscore* is turned on, in another inside the sourced working directory of the Kalibr package, the following command is used:

kalibr_calibrate_cameras –target aprilgrid.yaml –bag recorded_image.bag –models pinhole-radtan –topics /cam0/image_raw

The calibration process takes some time as an initialization step is performed followed by multiple optimization procedures.

3.3 The Results

The results obtained from the intrinsic calibration of the DVS camera were:

– camera_model: pinhole
– distortion_coeffs: [-0.3242162686575333, 0.18902392288580785, 0.001956087069885271, 0.002927723754485499]
– distortion_model: radtan
– intrinsics: [487.3643905959518, 485.8619326463498, 188.4079565262138, 128.66643667290577]
– resolution: [346, 260]
– rostopic: /dvs/image_raw
– projection: [487.3643906, 485.86193265, 188.40795653, 128.66643667] +- [4.85793244, 4.76947933, 2.54408222, 3.16044953]
– reprojection error: [-0.000000, -0.000000] +- [0.095395, 0.098285]

The reprojection error can be visualized in the figure 3.

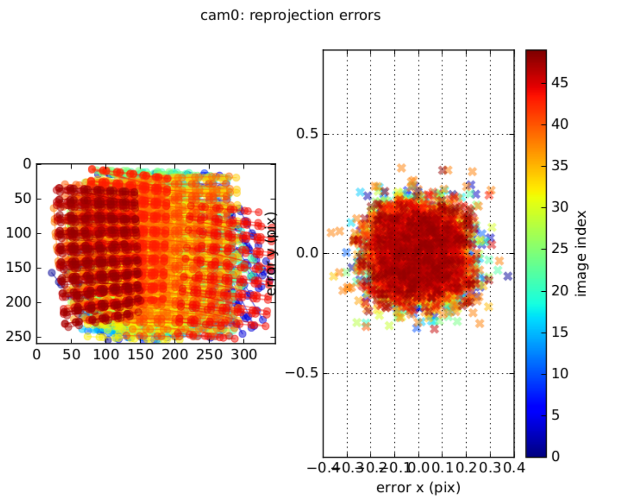

Fig. 3 Reprojection Error Plot

4 VICON Motion Capture System - DVS Camera Calibration

This part deals with finding the extrinsic parameters. The desired unknown we want to calculate for the extrinsics is the transformation from the origin of the VICON Motion Capture System frame to the camera center of the DVS camera. The first task for this process is to calibrate the VICON Motion Capture System and it was done by following the steps mentioned in the VICON documentation[3]. This has to be done so that we get accurate values of the coordinates reported by the Motion Capture System.

To perform this extrinsic camera calibration, two methods have been proposed.

4.1 Calibration using ARUCO Marker

The first approach makes use of the advantage of the ARUCO Markers. The pose reported by the ARUCO markers is used to estimate the extrinsics.

4.1.1 Approach

In this approach the following bodies are taken in consideration for estimation of extrinsics:

- VICON Center (W): This is the center or origin of the VICON Motion Capture System. This point in the 3D space will be considered as the World Frame and is denoted by W.
- Dynamic Vision Sensor Camera Center (CC): This is the center of the Dynamic Vision Sensor Camera is denoted by CC.
- ARUCO Marker Center (ARUCO): The origin/center of the ARUCO marker is denoted by ARUCO.
- VICON ARUCO Marker (ARMKR): VICON markers placed on the ARUCO marker is denoted by ARMKR.
- VICON Camera Marker (MKR): VICON markers placed on the DVS system is denoted by MKR.

In this setup, we know the origin of the VICON Motion Capture System by following the calibration of the VICON Motion Capture System [3]. But the position of the ARUCO Marker and the DVS Camera is unknown. To get these unknowns, we place VICON markers at known positions on them. We can extract the pose of these markers in the VICON frame as the Motion Capture System detects it. Now as we have all the necessary poses at this stage, we can proceed to estimation of the extrinsics.

The following steps give us all the known transformations in this system. The notation T_x^y or T_x_y denotes the transformation that will take the body from pose x to pose y.

1. Get the value of T_W^{MKR}, the transformation from the World Frame to the VICON marker on the Dynamic Vision Sensor.

2. Retrieve T_W^{ARMKR}, the transformation from the World Frame to the VICON marker on the ARUCO marker.
3. Since we have already calculated the intrinsics of the DVS Camera, compute the pose of the ARUCO marker with respect to the Camera Center. For this process, we already know the edge length of the square shaped ARUCO marker, and this pose estimation is reduced down to a PnP problem on the detected corners of the ARUCO marker. This in turn gives us the value of T_{CC}^{ARUCO} which is the transformation from the camera center to the ARUCO marker center.
4. We know that the center of the ARUCO marker is at the center of the square, hence computing the transformation of the ARUCO marker center with respect to the ARUCO VICON marker we get T_{ARMKR}^{ARUCO}

Given all these values, we need to estimate the value of T_W^{CC}. This can be broken down to a linear algebra problem pertaining to transformation matrices.

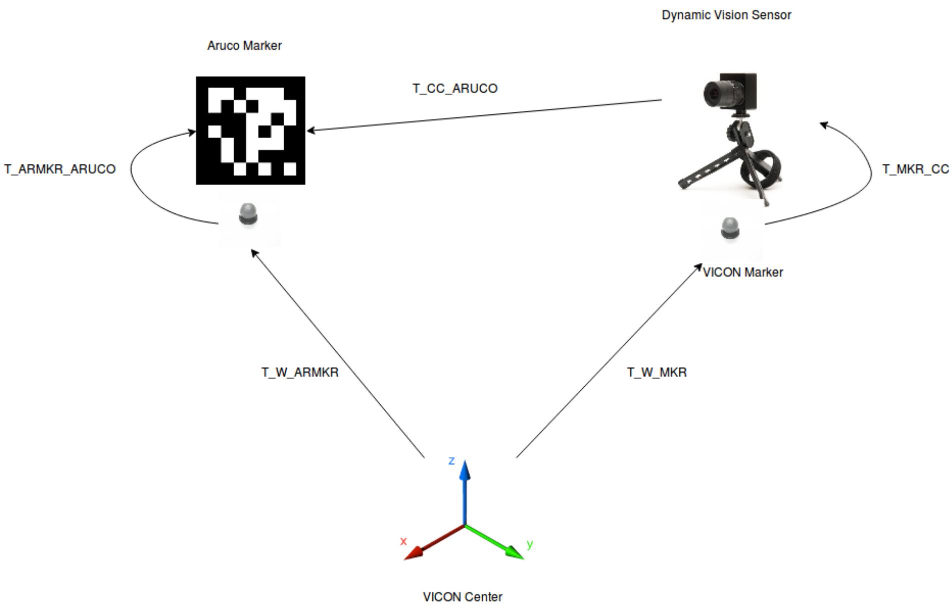

Fig. 4 Extrinsic Calibration Using ARUCO Marker [4][5][6].

$$T_W^{CC} = T_W^{MKR} * T_{MKR}^{CC} \tag{5}$$

But we know that

$$T_W^{MKR} * T_{MKR}^{CC} * T_{CC}^{ARUCO} = T_W^{ARMKR} * T_{ARMKR}^{ARUCO} \tag{6}$$

Hence solving for the unknown, we get

$$T_{MKR}^{CC} = T_W^{MKR^{-1}} * T_W^{ARMKR} * T_{ARMKR}^{ARUCO} * T_{CC}^{ARUCO^{-1}} \tag{7}$$

Finally we get the extrinsics between the vicon center and the camera center i.e. T_W^{CC}

$$T_W^{CC} = T_W^{ARMKR} * T_{ARMKR}^{ARUCO} * T_{CC}^{ARUCO^{-1}} \tag{8}$$

4.1.2 Limitations

The use of ARUCO markers for the extrinsics estimation wasn't fruitful due to the below mentioned reasons:

1. There are only four corners detected in the ARUCO marker, hence there are not enough points to refine the pose value.
2. Because of the symmetry of the four corners in the ARUCO marker, one axis tends of flip erratically , thereby giving a wrong pose estimate of the camera center.

To overcome these problems, we need to come up with a marker board that has a large number of corner feature points that will give a refined value of the camera pose. The following section discusses another approach to overcome this problem.

4.2 Calibration using Chessboard

Moving on from the limitations of the first approach using ARUCO Markers, the pose of the camera is computed by using a Chessboard pattern.

4.2.1 Approach

In this approach the following bodies are taken in consideration for estimation of extrinsics:

- VICON Center (W): This is the center or origin of the VICON Motion Capture System. This point in the 3D space will be considered as the World Frame and is denoted by W.
- Dynamic Vision Sensor Camera Center (CC): This is the center of the Dynamic Vision Sensor Camera is denoted by CC.
- Chessboard Center (CH): The origin/center of the Chessboard is denoted by CH.
- VICON Chessboard Marker (CHMKR): VICON markers placed on the Chessboard is denoted by CHMKR.
- VICON Camera Marker (MKR): VICON markers placed on the DVS system is denoted by MKR.

In this setup, we know the origin of the VICON Motion Capture System by following the calibration of the VICON Motion Capture System [3]. But the position of the Chessboard and the DVS Camera is unknown. To get these unknowns, we place VICON markers at known positions on them. We can extract the pose of these markers in the VICON frame as the Motion Capture System detects it. Now as we have all the necessary poses at this stage, we can proceed to estimation of the extrinsics.

1. Get the value of T_W^{MKR}, the transformation from the World Frame to the VICON marker on the Dynamic Vision Sensor.
2. Retrieve T_W^{CHMKR}, the transformation from the World Frame to the VICON marker on the Chessboard marker.
3. Since we have already calculated the intrinsics of the DVS Camera, compute the pose of the Chessboard marker with respect to the Camera Center. For this process, we already know the edge length of the square on the Chessboard marker, and this pose estimation is reduced down to a PnP problem on the detected corners of the Chessboard marker. This in turn gives us the value of T_{CC}^{CH} which is the transformation from the camera center to the Chessboard marker center. The advantage of using a Chessboard marker is that we have more number of corner points which improves the estimate of the pose.
4. We know that the center of the Chessboard marker is at the bottom left inner corner of the Chessboard, hence computing the transformation of the Chessboard marker center with respect to the Chessboard VICON marker we get T_{CHMKR}^{CH}

Given all these values, we need to estimate the value of T_W^{CC}. This can be broken down to a linear algebra problem pertaining to transformation matrices.

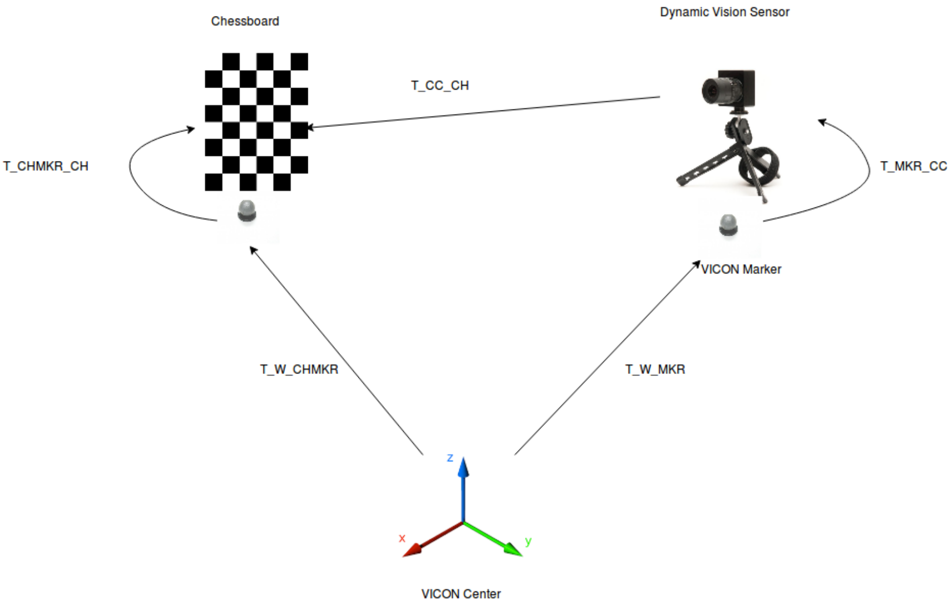

Fig. 5 Extrinsic Calibration Using Chessboard Markers [4][5][6].

$$T_W^{CC} = T_W^{MKR} * T_{MKR}^{CC} \tag{9}$$

But we know that

$$T_W^{MKR} * T_{MKR}^{CC} * T_{CC}^{CH} = T_W^{CHMKR} * T_{CHMKR}^{CH} \tag{10}$$

Hence solving for the unknown, we get

$$T_{MKR}^{CC} = T_W^{MKR^{-1}} * T_W^{CHMKR} * T_{CHMKR}^{CH} * T_{CC}^{CH^{-1}} \tag{11}$$

Finally we get the extrinsics between the vicon center and the camera center i.e. T_W^{CC}

$$T_W^{CC} = T_W^{CHMKR} * T_{CHMKR}^{CH} * T_{CC}^{CH^{-1}} \tag{12}$$

5 3D Model Image Reconstruction

Another task that was attempted in this independent study was to reconstruct a 3D object and project it onto the image plane. The problem, its description and method used to approach this problem are discussed in this section.

5.1 Problem Statement

Given the 3D point cloud model of the object of interest and coordinates of a few (four) fixed points on the 3D object from VICON markers, reconstruct the 3D object with pose reflecting the current readings from the VICON Motion Capture System and reproject it onto the image plane.

So, the point cloud file (.ply format) is available and the real time coordinates of four points where the VICON markers are placed on the 3D object is available to us. So we need to perform an estimation of the transformation from the 3D model file and the real world 3D model to reconstruct the model.

5.2 The Approach

The following algorithm can be used to the problem statement described above:

Algorithm 1 3D Reconstruction Algorithm

procedure INITIALIZE
    objectPoints ← coordinates of 4 points corresponding to VICON markers on object
    markerPoints ← coordinates of 4 points from VICON system
    objectPointsPlane ← plane using 3 points of *objectPoints*
    markerPointsPlane ← plane using 3 points of *markerPoints*
    Rotation, Translation ← align objectPointsPlane with markerPointsPlane **return** Rotation, Translation

procedure RECONSTRUCT 3D OBJECT(INPUTCLOUD, ROTATION, TRANSLATION)
    ReconstructedCloud ← Transform *InputCloud* using *Rotation, Translation*

procedure RECONSTRUCT AND PROJECT
    Rotation, Translation ← INITIALIZE()
    OutputCloud ← *RECONSTRUCT 3D OBJECT(InputCloud, Rotation, Translation)*
    Store OutputCloud
    Reproject OutputCloud to Image Plane
    For successive iterations, use ICP on current cloud and OutputCloud for Transformation

Firstly, we need to have the exact coordinates of the points on the 3D model of the object generated where the VICON markers are present, or at least offsetted by a constant value uniform across all points. If this is not the case, then it is observed that a correct rotation and translation matrix is not established.

Iterative Closest Point (ICP) is an algorithm that is widely used to align two point clouds by minimizing the aligning error between two sets of points cloud. Using ICP on two identical point clouds, will give us the rotation and translation matrices that align the two.

For the first iteration, or initialization, we can not use the ICP method as since the point clouds might be far away, the ICP will not converge and hence produce undesired, incorrect outputs. One solution is to possibly use the Global ICP algorithm to get the initial transformation matrix [7]. But that is not a tractable solution for this case as something more simpler, a naive solution could be presented. This is done by noting the coordinates of the 3D object where the markers are placed and registering the VICON marker coordinates from the published topics. Using three points each, we can describe the equation of the plane of the markers on the 3D object and the plane of the object in the real world. Once we align their surface , we get a rotation matrix between the two planes. For estimating the translation, we take the offsets in each axis of the aligned versions of the points. To further remove ambiguities, we can perform a chirality check by verifying the fourth point aligns perfectly with the fourth marker reading using the four combinations of rotation and translation matrices.

Once the initialization step has been performed, we can use this transformation to transform all the point in the point cloud file to reflect the current position and orientation of the 3D object. Reprojecting these points on the camera image plane will give us a visualization on how the image looks from the perspective of the camera. For every other successive value of the point cloud information, comparing it with the previous point cloud and using ICP will yield a correct value of transformation matrix as the change in pose between two successive readings from the VICON Motion Capture System will be small that the ICP algorithm can handle.

6 Conclusion

The aim of the independent study course ENPM 808 was to learn the fundamentals of calibration for the sensors used in computer vision tasks, study the process of 3D reconstruction, apply this knowledge and have a hands on experience in the domain of computer vision.

In the due course of the independent study, firstly, camera calibration was studied. The process of calibrating a camera for its intrinsics and distortion parameters was studied. This knowledge was applied by calibrating camera of namely two models- a pinhole-radial tangential distortion model camera and a fisheye camera of pinhole model with equidistant distortion. The results of the calibration with intrinsics and reprojection errors have been reported in section 3.3.

After this task, the process of extrinsic calibration was studied to understand the problem associated with calibrating multiple sensors. The problem at hand required to device an approach to find the extrinsic between the VICON Motion Capture System and the Dynamic Vision Sensor. Two approaches were developed and deployed to find the extrinsics, the specifics of which are mentioned in section 4.

Finally, an algorithm had to be developed to reconstruct and reproject a 3D object from the coordinates of four points reported from the VICON Motion Capture System. For this principles of linear algebra were exploited along with the use of point cloud and computer vision libraries. The algorithm for reconstruction of the 3D object using the VICON marker readings, and the method to reproject this reconstructed point cloud on to the image plane has been described in section 5.2.

This independent study course not only offered the opportunity to work on the above mentioned projects but also to study the LiDAR (Light Detection and Ranging) sensor, functioning of the event based camera, and simultaneous localization and mapping (SLAM) using LiDAR sensor. Various ROS packages were looked into and used to set up the LiDAR, calibrate the LiDAR with a camera, and map an environment.

References

1. OpenCV Documentation, https://docs.opencv.org/
2. The Kalibr Visual-Inertial Calibration Toolbox, https://github.com/ethz-asl/kalibr
3. VICON Documentation, https://docs.vicon.com/display/Nexus25/Calibrate+a+Vicon+system
4. XYZ Axes Image, https://goo.gl/images/s54bJM
5. DVS Image, https://goo.gl/images/29cX2U
6. VICON Marker Image, https://goo.gl/images/dxwV8F
7. Jiaolong Yang, Hongdong Li, Dylan Campbell, Yunde Jia, Go-ICP: A Globally Optimal Solution to 3D ICP Point-Set Registration