Politecnico di Milano

Industrial Engineering Faculty

Master of Science in Mechanical Engineering

Stereoscopic measurements uncertainty analysis: a numerical and experimental approach

Supervisor: Prof. Emanuele ZAPPA

M.Sc Thesis of:

ÖZGÜR ÇELİK Matr. 764133

ILKER ERDEN Matr. 764417

Academic Year 2012-2013

Table of Contents

ABSTRACT

The stereoscopic system is settled in the laboratory and used for measurement of 3D coordinates. By using experimental data the uncertainty of the measurements described by means of discrepancies. A numerical approach developed by using Monte Carlo method in order to investigate the effect of parameters on the uncertainties. As a result; the effect of investigated parameters described including comparison for the certain experiment setup and the developed numerical approach is applicable for any other stereoscopic system with different baseline.

Keywords: stereoscopy, uncertainty, Monte Carlo, 3D measurement.

List of Figures

List of Tables

1. Introduction to Camera Calibration

Taking an image of an object by camera cause some defects on it during transferring from 3-D reference frame to 2-D image frame due to the camera components like lenses and sensors. The aim of the camera calibration is making correction on the image and gathering the image as close as to the real world.

There are some softwares to calibrate the cameras and to correct the images. In order to calibrate the cameras we used Camera Calibration Toolbox for MatLab Software designed by California Institute of Technology.

Before start to explain calibration procedure, we have to be familiar with some parameters and theory of the camera calibration and pinhole camera model.

1.1. Pinhole Camera Model

The pinhole camera model defines the mathematical relations through projection from 3D coordinates to image plane by a pinhole camera that no lenses are used to focus light as aperture .The camera aperture is just defined as a point. The pinhole aperture of the camera where the whole projection lines pass is considered to be extremely small point. In the literature this point in 3D space is referred to as the optical (or lens or camera) center.[1]

Light from a point travels along a single straight path through a pinhole onto the view plane. The object is imaged upside-down on the image plane.[2]

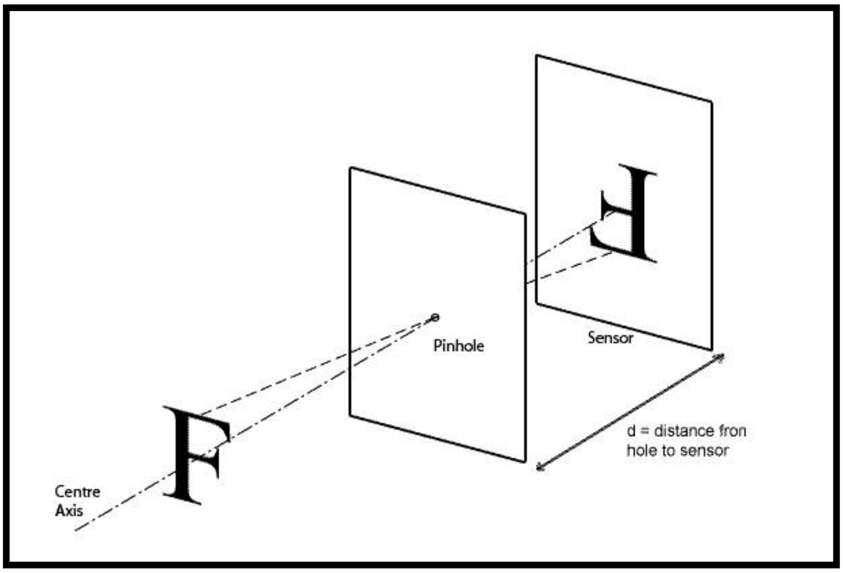


Figure 1.1 Geometry of a Pinhole Camera

There is another way of thinking about the pinhole cameras. Let’s suppose to view a scene with one eye looking through a square window, and draw a picture of what we see through the window . The image we would get corresponds to drawing a ray from the eye position and intersecting it with the window. This is equivalent to the pinhole camera model, except that the view plane is in front of the eye instead of behind it, and the image appears rightside-up, rather than upside down. (The eye point here replaces the pinhole). [2]

1.1.1 Geometrical and Mathematical Relations of the Pinhole Camera

By looking at figure-1.2 ,**C** is the origin of 3D orthogonal coordinate system also where the camera aperture is positioned. X, Y, Z are pointed as three axes of the coordinate system. Z axis is directed to the field of view of the camera and it is introduced as the optical axis or principal axis.

The image plane is parallel to axes X and Y and it is positioned at a distance from the origin **C** where the 3D world is projected through the aperture of the camera. The distance between image plane and the camera center gives the focal length **f** of the pinhole camera. The image plane is perpendicular to the Z axis.

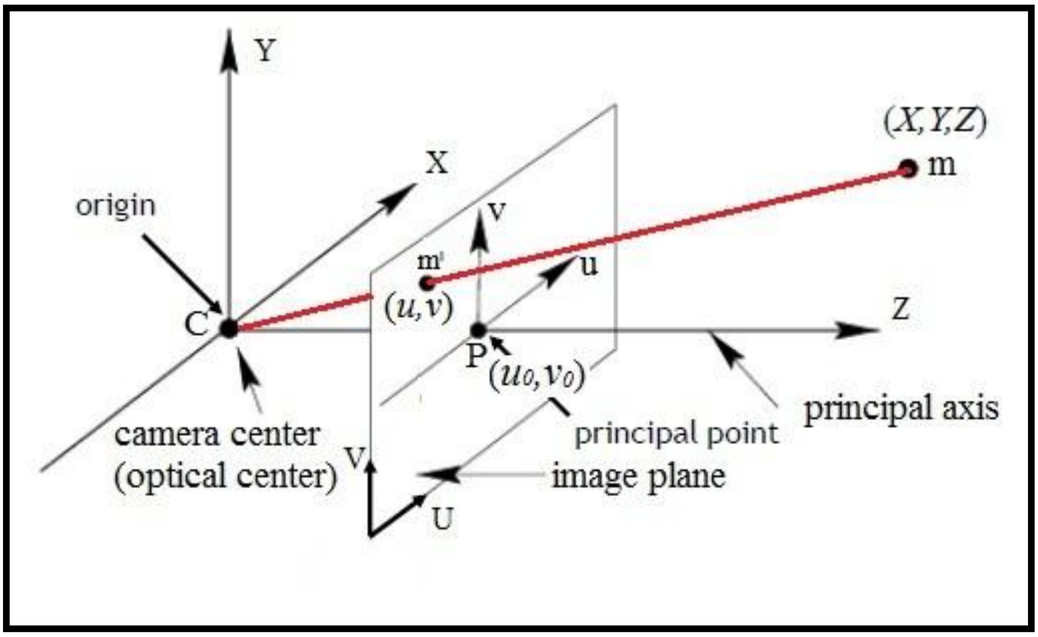


Figure 1.2 Geometry of a Pinhole Camera with Coordinates

The point **P** is cited as the principal point or image center where the intersection of the image plane and the optical axis and the coordinates of the principal point is (u_0, v_0) . The point **m** into the real world with coordinates related to the axes X,Y, Z is projected to the image plane denoted as **m**' through the red projection line. The point **m**' is the intersection point of the projection line(red) and the image plane.

Moreover, there is a 2D coordinate system on the image plane, with origin at **P** and with axes u and v which are parallel to X and Y, properly. The coordinates of point **m**' corresponding to this coordinate system is (*u*, *v*).

For the next step, we need to understand the relations between the coordinates of **m**' (*u*, *v*) and coordinates of point **m** (*X*, *Y*, *Z*) according to camera reference frame. We can obtain the equation (1.1) by figure-1.3.

$$u = f\frac{X}{Z} \ \ and \ \ v = f\frac{Y}{Z} \qquad (1.1)$$

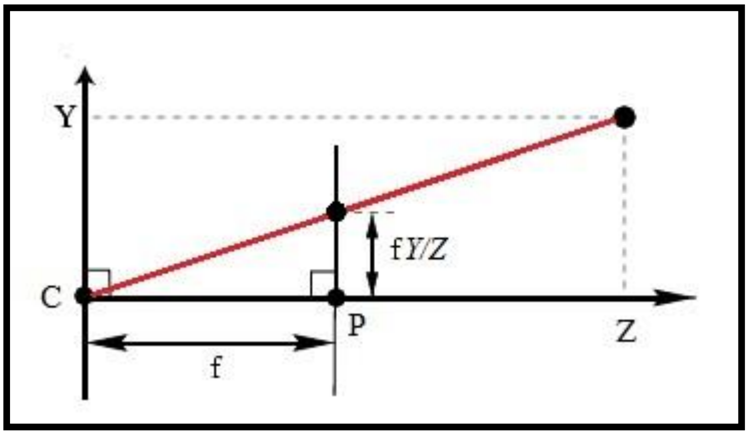


Figure 1.3. Geometry of a pinhole camera as seen from the X axis

1.1.2. Central Projection by Using Homogenous Coordinates

If the world and image points are represented by homogenous vectors, then central projection is very simply expressed as a linear mapping between their homogenous coordinates. It may be written in terms of matrix multiplication as in the equation 1.2. [3]

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \qquad (1.2)$$

In practice, the origin of coordinates in the image plane is not coincident with the principal point. So, the expression on equation 1.2 is modified as equation 1.3.

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \sim \begin{pmatrix} fX + Zu_0 \\ fY + Zv_0 \\ Z \end{pmatrix} = \begin{bmatrix} f & & u_0 & 0 \\ & f & v_0 & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \qquad (1.3)$$

$$\mathrm{K} = \begin{bmatrix} f & & u_0 \\ & f & v_0 \\ & & 1 \end{bmatrix} \qquad (1.4)$$

$$\mathbf{m}' = \mathbf{K}\ [\ \mathrm{I} \mid 0]\ \mathbf{m} \qquad (1.5)$$

And, **K** is called the camera calibration matrix. In (1.5) If we write $(u, v, 1)^T$ as **m**' and $(X, Y, Z, 1)^T$ as **m** and the camera is assumed to be located at the origin of Euclidian coordinate system with the principal axis of the camera pointing straight down the Z-axis and the point **m** is expressed in this coordinate system. This coordinate system may be called the camera coordinate frame.[3]

1.2. Description of the calibration parameters

In this section the calibration parameters will be explained in detail with the contribution of visual sources.

1.2.1. Intrinsic parameters

In the program shortenings are used to identify the internal parameters.

- **Focal length:** The focal length of an optical system is a measure of how strongly the system converges or diverges light. For an optical system in air, it is the distance over which initially collimated rays are brought to a focus.

  A system with a shorter focal length has greater optical power than one with a long focal length; that it bends the rays more strongly, bringing them to a focus in a shorter distance.

Longer focal length (lower optical power) leads to higher magnification and a narrower angle of view; conversely, shorter focal length or higher optical power is associated with a wider angle of view. As it can be seen on figure 1.4, if the focal length decreases, the angle of view increases.

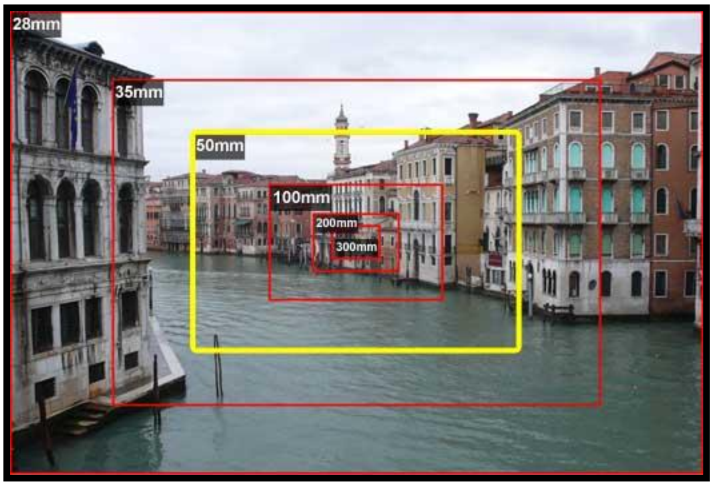


Figure 1.4 Visual demonstration of the relation between angle of view and focal length

In the calibration tool the focal length in pixels is stored in the 2x1 vector as **fc**, the first row of the matrix is the focal length on **u** direction and the second row of the matrix is the focal length on **v** direction. The ratio of **fc**(2) / **fc**(1) is called aspect ratio and it equals to 1 when the CCD sensor array is square but camera sensors generally are non-square, it is the reason of why we use two different focal length. Considering the focal length on **u** and **v** axis , we have to update the **K** matrix .

$$\mathbf{K} = \begin{bmatrix} \mathbf{fc}(1) & & u_0 \\ & \mathbf{fc}(2) & v_0 \\ & & 1 \end{bmatrix} \tag{1.6}$$

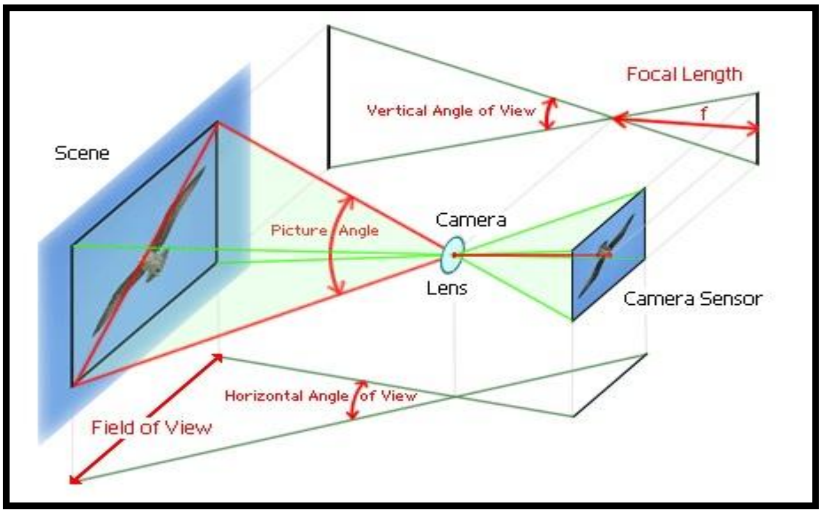


Figure 1.5. Focal Length on u and v axes

- **Principal point:** The principal point coordinates are the center of coordinates on the image plane (at the same time, sensor plane), **P** (u_0, v_0) and these coordinates stored in the 2x1 vector. First row of the vector demonstrates the pixel coordinates on **u** direction and the second row of the vector demonstrates the pixel coordinate on **v** direction. Principal point is shown in the tool as **cc** and it is fixed as $\binom{0}{0}$ during calibration.

- Skew coefficient: The skew coefficient defines the angle between the **u** and **v** pixel axes. It's optimal value is 90 ^{o} and shown as **alpha_c**. If we update the matrix **K**

$$\mathbf{K} = \begin{bmatrix} \mathbf{fc}(1) & \mathbf{alpha\_c} & u_0 \\ & \mathbf{fc}(2) & v_0 \\ & & 1 \end{bmatrix} \qquad (1.7)$$

- Distortions: The image distortion coefficients (radial and tangential distortions) are stored in the 5x1 vector **kc**.

  Radial distortions are symmetric and ideal image points are distorted along radial directions from the distortion center. This is caused by imperfect lens shape and deviations are most noticeable for rays that pass through the edge of the lens. Decentering distortions are usually caused by improper lens assembly and ideal image points are distorted in both radial and tangential directions.[4]

Let's consider the projection of point m on the camera reference frame and let's take $\mathbf{X_n}$ as the normalized (pinhole) image projection.

$$\mathbf{X_n} = \begin{bmatrix} X/Z \\ Y/Z \end{bmatrix} = \begin{bmatrix} u_n \\ v_n \end{bmatrix} \qquad (1.8)$$

And let $$r^2 = u_n{}^2 + v_n{}^2 \qquad (1.9)^5$$

$\mathbf{X_d}$ is the distorted coordinates and it is derived from undistorted image coordinates $\mathbf{X_n}$ in (1.10), where **dx** is the tangential distortion vector that may be found from (1.11)[5]

$$\mathbf{X_d} = \begin{bmatrix}\mathbf{X_d}(1)\\ \mathbf{X_d}(2)\end{bmatrix} = (1 + \mathbf{kc}(1)\,\mathrm{r}^2 + \mathbf{kc}(2)\,\mathrm{r}^4 + \mathbf{kc}(5)\,\mathrm{r}^6)\,\mathbf{X_n} + \mathbf{dx} \qquad (1.10)^5$$

$$\mathbf{dx} = \begin{bmatrix} 2\,\mathbf{kc}(3)\,u_n v_n + \mathbf{kc}(4)\left(\mathrm{r}^2 + 2\,{u_n}^2\right) \\ \mathbf{kc}(3)\left(\mathrm{r}^2 + 2\,{v_n}^2\right) + 2\,\mathbf{kc}(4)\,u_n v_n \end{bmatrix} \qquad (1.11)^5$$

kc vector consists of both tangential and radial distortion coefficients. After distortion is assigned, the final pixel coordinates c_pixel = [*u;v*] of the projection of **m** on the image plane is:

$$u = \mathbf{fc}(1)\big(\mathbf{X_d}(1) + \mathbf{alpha\_c} * \mathbf{X_d}(2)\big) + \mathbf{cc}(1)$$

$$v = \mathbf{fc}(2)\mathbf{X_d}(2) + \mathbf{cc}(2) \qquad (1.12)^5$$

The pixel coordinate vector c_pixel and distorted coordinate vector $\mathbf{X_d}$ are realted to each other by linearly.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathrm{K} \begin{bmatrix} \mathbf{X_d}(1) \\ \mathbf{X_d}(2) \\ 1 \end{bmatrix} \qquad (1.13)^6$$

Finally, the camera calibration matrix K is defined as:

$$\mathbf{K} = \begin{bmatrix} \mathbf{fc}(1) & \mathbf{alpha_c} * \mathbf{fc}(1) & u_0 \\ 0 & \mathbf{fc}(2) & v_0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (1.14)^6$$

Also, this tool calculates the uncertainties of **fc, cc,kc** and **alpha_c** and those vectors are 3 times of the standard deviation of error estimation. Pixel coordinates are expressed as [0;0] is the left upper pixel's center. [nx-1;0] is the right upper pixel's center, [0;ny-1] is the lower left corner pixel's center, [nx-1;ny-1] is the lower right pixel's center.

In our experiment nx=3872 , ny=2592 . In the toolbox there is a function that computes direct projection map.This function takes the 3-D coordinates of the set of points in world or camera reference frame and intrinsic camera parameters and returns the pixel projections of the points on image plane. [6]

1.2.2. Extrinsic Parameters

Extrinsic parameters are related with the camera's location and the orientation of it with the world. Rotations and translations vectors represent the extrinsic parameters and give the relations between the world reference frame $(X_{world}, Y_{world}, Z_{world})^T$and the camera reference frame coordinates $(X, Y, Z)^T$.

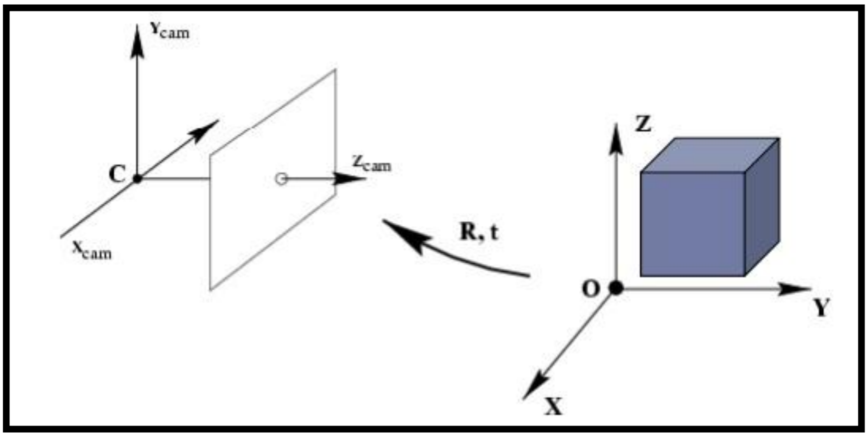


Figure 1.6. Rotation and translation from world coordinates to camera coordinates

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K}_{3\mathrm{x}3}({}^{\mathrm{c}}_{\mathrm{w}}\mathbf{R}_{3\mathrm{x}3}\ {}^{\mathrm{c}}_{\mathrm{w}}\mathbf{O}\ {}_{3\mathrm{x}1})_{3\mathrm{x}4}\begin{pmatrix} {}^{w}\mathbf{m} \\ 1 \end{pmatrix} = \mathbf{K}[r_1\ \ r_2\ \ r_3\ \ t] \qquad (1.15)^{7}$$

$$\mathbf{m'} = \boldsymbol{P}\ \mathbf{m} \qquad (1.16)$$

${}^{\mathrm{c}}_{\mathrm{w}}\mathbf{R}_{3\mathrm{x}3}$ is the rotation matrix which rotates the coordinates from the world reference frame to the camera reference frame and can be seen in (1.17). ${}^{\mathrm{c}}_{\mathrm{w}}\mathbf{O}\ {}_{3\mathrm{x}1}$ is the translation matrix that translates the coordinates from the world reference frame to the camera reference frame. ***P*** is the projection matrix that equals at the same time $\mathbf{K}_{3\mathrm{x}3}({}^{\mathrm{c}}_{\mathrm{w}}\mathbf{R}_{3\mathrm{x}3}\ {}^{\mathrm{c}}_{\mathrm{w}}\mathbf{O}\ {}_{3\mathrm{x}1})_{3\mathrm{x}4}$ and uses both the intrinsic (**K**) and the extrinsic parameters (**R,O**).

There are 11 free parameters into the Perspective Projection Matrix ***P***, 5 of them are intrinsic, 3 of them are rotation and 3 of them are translation parameter.[7]

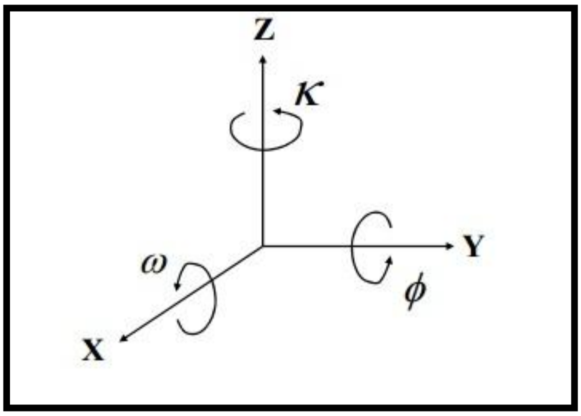


Figure 1.7. Rotation with Euler Angles[8]

$$\mathrm{R} = \begin{pmatrix} \cos K & \sin K & 0 \\ -\sin K & \cos K & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos ø & 0 & \sin ø \\ 0 & 1 & 0 \\ -\sin ø & 0 & \cos ø \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \omega & \sin \omega \\ 0 & -\sin \omega & \cos \omega \end{pmatrix} \qquad (1.17)^{8}$$

$$\mathrm{R} = \begin{pmatrix} \cos K \cos ø & \sin K \cos \omega - \cos K \sin ø \sin \omega & \sin K \sin \omega + \cos K \sin ø \cos \omega \\ -\sin K \cos ø & \cos K \cos \omega + \sin K \sin ø \sin \omega & \cos K \sin \omega - \sin K \sin ø \cos \omega \\ -\sin ø & -\cos ø \sin \omega & \cos ø \cos \omega \end{pmatrix} \qquad (1.18)$$

Finally, we may represent the relation between the world coordinates $(X_{world}\,, Y_{world}\,, Z_{world})^{\mathrm{T}}$ and the image coordinates (u, v) in (1.19)

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} \begin{pmatrix} . & . & . & T_x \\ . & R & . & T_y \\ . & . & . & T_z \end{pmatrix} \begin{pmatrix} X_{world} \\ Y_{world} \\ Z_{world} \end{pmatrix} \qquad (1.19)^{8}$$

In the notation of the Camera Calibration toolbox for Matlab, the reference frame of the calibration grid is (O,X,Y,Z) in the Figure1.8

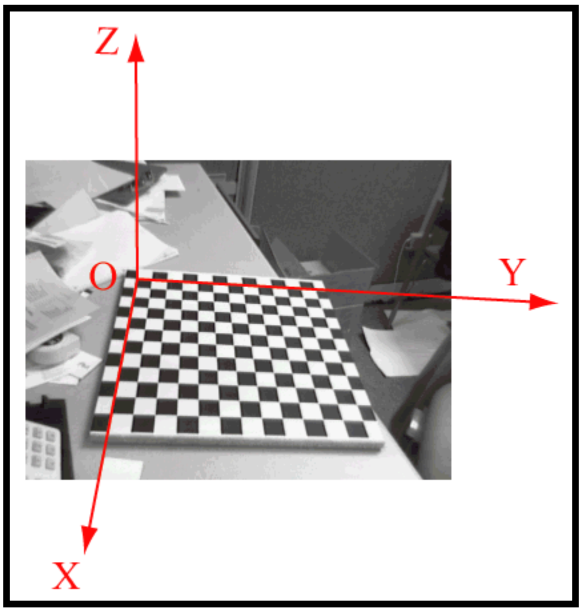


Figure 1.8. Reference frame of the calibration grid[6]

X is a point in space with coordinate vector XX = [X;Y;Z] on the grid reference frame (figure 1.8). XXc = [Xc;Yc;Zc] is the coordinate vector of point X on the camera reference frame . Then XX and XXc are related to each other with the equation (1.20)[6]

$$XXc = Rc\_1 * XX + Tc\_1 \qquad (1.20)^6$$

The translation vector Tc_1 is the coordinate vector of the origin of the grid pattern (O) in the camera reference frame and the third column of the matrix Rc_1 is the surface normal vector of the plane contains the planar grid in camera reference frame. This is valid for whole the extrinsic parameters (Rc_2,Tc_2), (Rc_3,Tc_3), ... , (Rc_25,Tc_25). The vectors omc_1, omc_1, ... , omc_25 are the rotation vectors related to the rotation matrices Rc_1, Rc_1, ... , Rc_25. These two are associated to the Rodrigues formula.

For example, Rc_1 = rodrigues(omc_1). As with in the intrinsic parameters, uncertainties related to the extrinsic parameters omc_i, Tc_i (i=1,...,n_ima) can be also calculated by toolbox. These uncertainties are represented approximately three times of the standard deviation of error estimation.[6]

1.3. Camera Calibration Procedure

In this section single camera calibration, stereo vision and the stereo camera calibration procedure will be explained.

1.3.1 Single Camera Calibration

A planar checkerboard grid is used in more than 2 different positions like Zhang's calibration method. [9] In this experiment the planar checkerboard grid is used in 25 different orientations in order to calibrate the camera. During calibration process there are several steps which are loading calibration images, extracting image corners, running the main calibration engine, displaying the results, controlling accuracies, adding and suppressing images, undistorting images, exporting calibration data to different formats.[10]

1.3.1.1. Loading Calibration Images

First step is to upload the images on the calibration tool for Matlab that are taken with the checkerboard grid with 25 different positions on the experiment. (See figure 1.9)

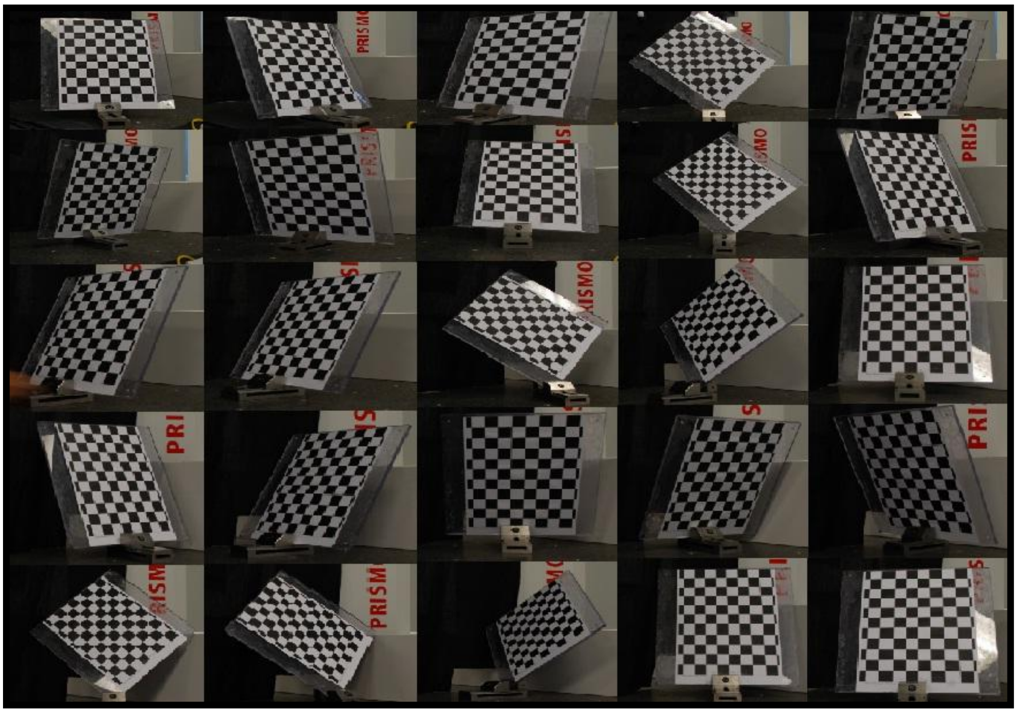


Figure 1.9. 25 Different orientation of the planar checkerboard grid

If the memory of the hardware that executes the MatLab tool is not enough, it is better to switch the program with memory efficient version.

1.3.1.2. Extracting Image Corners

By choosing extract grid corners option in the camera calibration tool we choose all the images. Then it is selected default window size of the corner finder and in this experiment it is chosen wintx and winty as 5. It means that automatic corner finder is going to search the corners in 11x11 pixel size. When the distortion values are high it may not be enough wintx and winty as 5. In order to catch the correct corners these optimum values have to be chosen. If the windows sizes are chosen too high, the program may find the wrong corners. [10]

Then, it is chosen 4 extreme corners of the rectangular checkerboard pattern. It should be clicked carefully at most 5 pixel away the corners. On the contrary, some corners might be missed.

The first clicking point is related to the origin point of the reference frame. The other 3 points of the rectangular grid can be chosen by any order. (See figure 1.10).

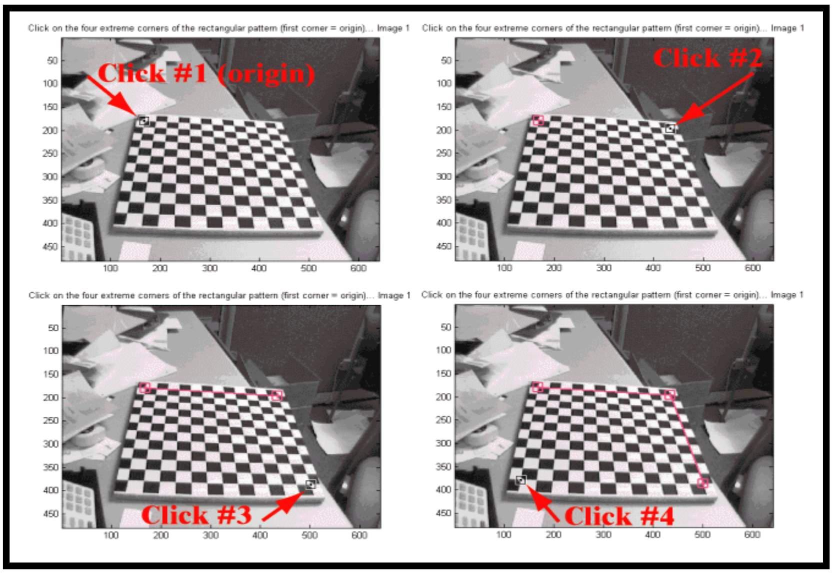

Click on the four extreme corners of the rectangular pattern (first corner = origin)... Image 1
Click #1 (origin)
Click #2
Click #3
Click #4
50
100
150
200
250
300
350
400
450
100
200
300
400
500
600

Figure 1.10. Extracting grid corners[10]

After choosing the corners of the grid, it is entered the x and y direction values of the squares of the checkerboard. It is 33 mm in this experiment. At the end of the value entries, the program counts the number of the squares in both dimensions and shows the approximated grid corners without distortion (figure 1.11).

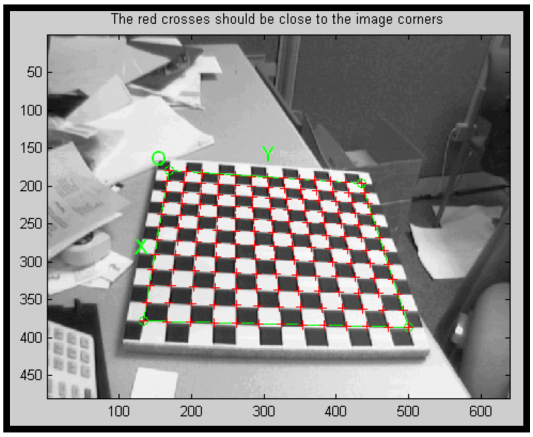


Figure 1.11. Checkerboard grid with counted squares

If the approximated corners are not satisfactory enough for the real corners, than the distortion cefficient should be predicted to compensate the distortion of the lense. kc (lense distortion coefficient) has to changed in order to obtain more realistic corner extraction. This process may be repeated with estimating kc values by trial and error till obtain reasonable corners.[10]

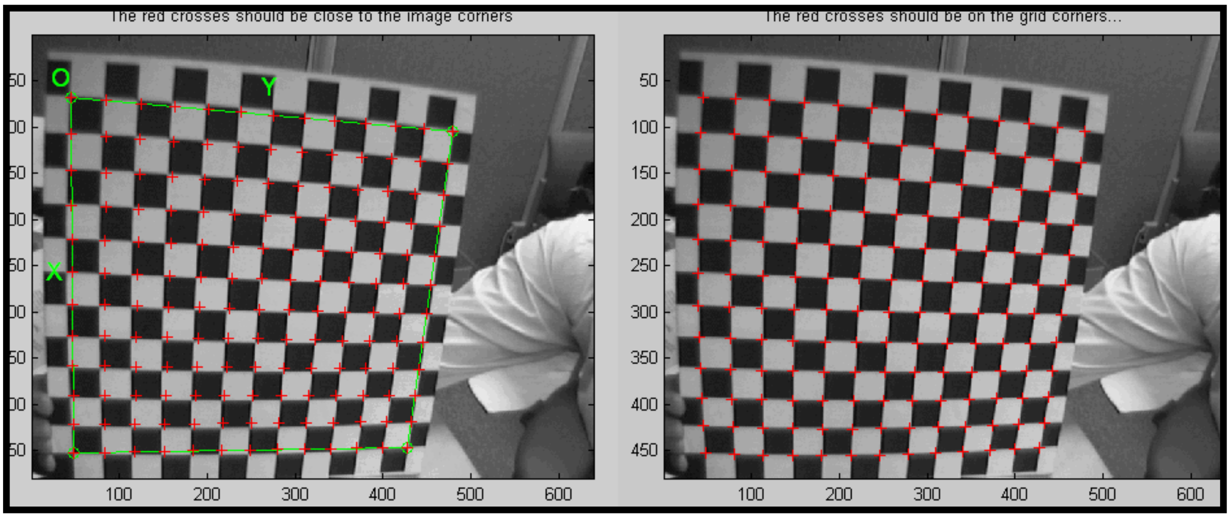

The red crosses should be close to the image corners
O
Y
X
100
200
300
400
500
600
The red crosses should be on the grid corners...
50
100
150
200
250
300
350
400
450
100
200
300
400
500
600

Figure 1.12. Extracting corners before and after the distortion coefficient is applied

The figure 1.12 shows the images of the corner extracting profcess before and after distortion coefficient is applied. The left image shows when the kc=0 , the right image shows kc = -0.3

After corner extraction, the matlab data file **calib_data.mat** is automatically generated. This file contains all the information gathered throughout the corner extraction stage (image coordinates, corresponding 3D grid coordinates, grid sizes, ...).

If the lens distortions are really too severe (for fisheye lenses for example), the simple guiding tool based on a single distortion coefficient kc may not be sufficient to provide good enough initial guesses for the corner locations. For those few difficult cases, a script program is included in the toolbox that allows for a completely manual corner extraction.[10]

1.3.1.3. Main Calibration Stage

After the corner extraction step is completed, the main camera calibration procedure is executed. Calibration occurs in two steps: first initialization, and then nonlinear optimization.

A closed-form solution is computed in the initialization step for the calibration parameters based not including any lens distortion. The total reprojection error is minimized by non-linear optimization step (in the least squares sense) over all the calibration parameters (9 DOF for intrinsic: focal, principal point, distortion coefficients, and 6*20 DOF for extrinsic, totally 129 parameters). The optimization is done by iterative gradient descent with an explicit (closed-form) computation of the Jacobian matrix.[10]

After these, the calibration parameters are stored in number of variables. If the skew coefficient alpha_c and the 6th order radial distortion coefficient (the last entry of kc) have not been estimated, the angle between u and v pixel axes is 90 degrees.If the reprojection error is still too large, it generally causes because of the grid corners were not very precisely extracted for a number of images.

Then, let’s click on Reproject on images in the Camera calibration tool to show the reprojections of the grids onto the original images. These projections are computed based on the current intrinsic and extrinsic parameters. Generally the reprojection error is too large across a large number of figures because of not doing a careful job when extracting corners with highly distorted images. A better job could have been done by using the predicted distortion option. We can correct for that now by recomputing the image corners on all images automatically.[10]

After that, we choose the mode of extraction: the automatic mode (auto) uses the re-projected grid as initial guess locations for the corner, the manual mode let's the user extract the corners manually (the traditional corner extraction method). In the present case, the reprojected grid points are very close to the actual image corners. Therefore, we can select the automatic mode. Only six iterations were necessary for convergence, and no initialization step was performed (the optimization started from the previous calibration results). After the optimization we can save the calibration results (intrinsic and extrinsic) in the matlab file Calib_Results.mat.[10]

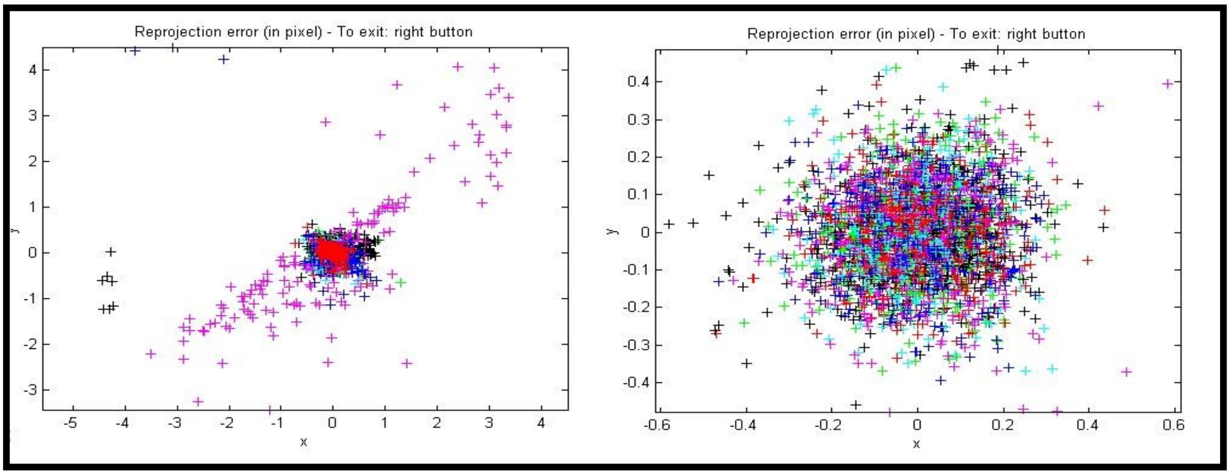


Figure 1.13. Projected (left) and Re-projected (right) error in pixels

In the left graph, projected error is more than the right graph; it means that after reprojection step, error value decreases.

The camera calibration tool can show the new 3D positions of the grids with respect to the camera by choosing the **Show Extrinsic** option.

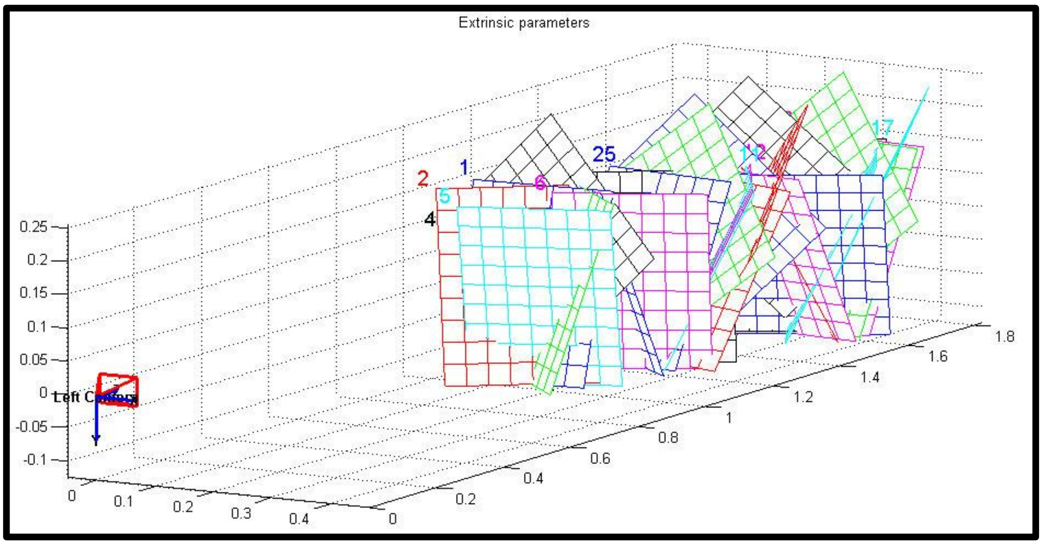


Figure 1.14. 3D positions of the checkerboard grids with respect to the camera

The tool **Analyse error** option allows to inspect which points match with the large errors. By clicking on the desired error region on (figure 1.14), the datas can be seen on the program.

The error inspection tool is very useful in cases where the corners are extracted badly into one or more images. In such a case, to recompute the corners of the specific images using a different window size is possible. Therefore, wintx and winty windows sizes can be choosen differently for all the images.

After recomputing the corners, we may recalibrate the camera by choosing calibration option in the tool. By this way, we can monitor a decrease on reprojection error compared to previous ones. In addition, we may recognize that the uncertainties on the calibration parameters are also smaller.

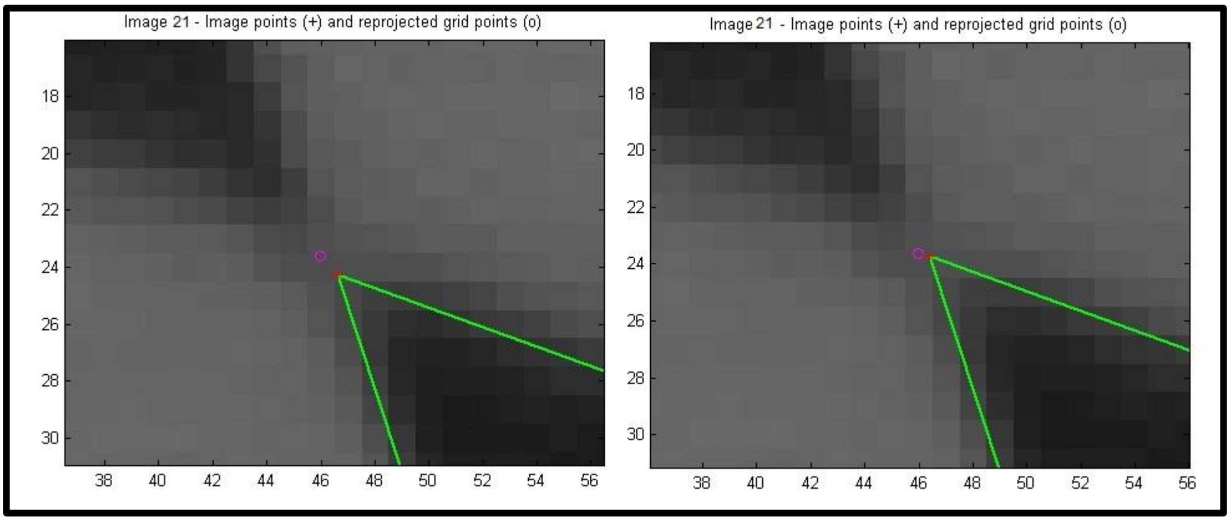


Figure 1.15. A close view to the corners of the grid on image 21 before and after recomputing corners

It can be observed on (figure 1.15) that changing windows sizes on related image is useful to contract the reprojection error. On the right image that is after recomputing corners step on (figure 1.15) corner extracting is more successful than the left image.

So it is possible to say that recomputing step serves properly. At the end of the process we may save the results of intrinsic and extrinsic parameters as **Calib_Results.mat**.

1.3.2 Stereo Vision Systems

Stereo vision is a technique that aims to infer the depth information from two or more cameras. With two or more camera it can be inferred depth by means of triangulation, if we may find the related points in the two images. In order to obtain the stereo vision we need to apply stereo camera calibration process.

1.3.2.1. Camera Calibration for Stereo Systems

While running the camera calibration tool for Matlab for stereo vision, first of all we have to calibrate the both two cameras separately like on the calibration procedure for one camera . After calibrating right and left cameras the calibration results will be saved as **Calib_Results_left.mat** for left camera and **Calib_Results_right.mat** for right camera. Then, within the folder including stereo data "Load left and right calibration files" option is chosen into the stereo toolbox and **Calib_Results_left.mat** , **Calib_Results_right.mat** is loaded.

At the end of the uploading data, intrinsic parameters of the right and left cameras are calculated and demonstrated. Moreover, extrinsic parameters **om**(rotation) and **T**(translation) vectors are calculated that characterize the relative location of the right camera with respect to the left camera.

The intrinsic parameters **fc_left, cc_left, alpha_c_left, kc_left, fc_right, cc_right, alpha_c_right and kc_right** are equivalent to the traditional parameters **fc, cc, alpha_c and kc** defined on the previous section.

The two pose parameters **om** and **T** are defined such that if we consider a point P in 3D space, its two coordinate vectors **XL** and **XR** in the left and right camera reference frames respectively are related to each other through the rigid motion transformation in equation (1.21).[11]

$$\mathbf{XR = R * XL + T} \qquad (1.21)^{11}$$

Where R is the 3x3 rotation matrix corresponding to the rotation vector **om**. The relation between **om** and **R** is given by the rodrigues formula **R = rodrigues(om)**. After executing “Run stereo calibration” in the stereo toolbox, we can derive the calibration data for the stereo system.

At the end of the procedure, all intrinsic and extrinsic parameters have been recalculated together and it can be observed that the uncertainties on intrinsic parameters for both cameras are smaller after stereo calibration. The stereo calibration results are saved as **Calib_Results_stereo.mat**. We can monitor the extrinsic parameters in a form of 3D plot.(see figure 1.16)

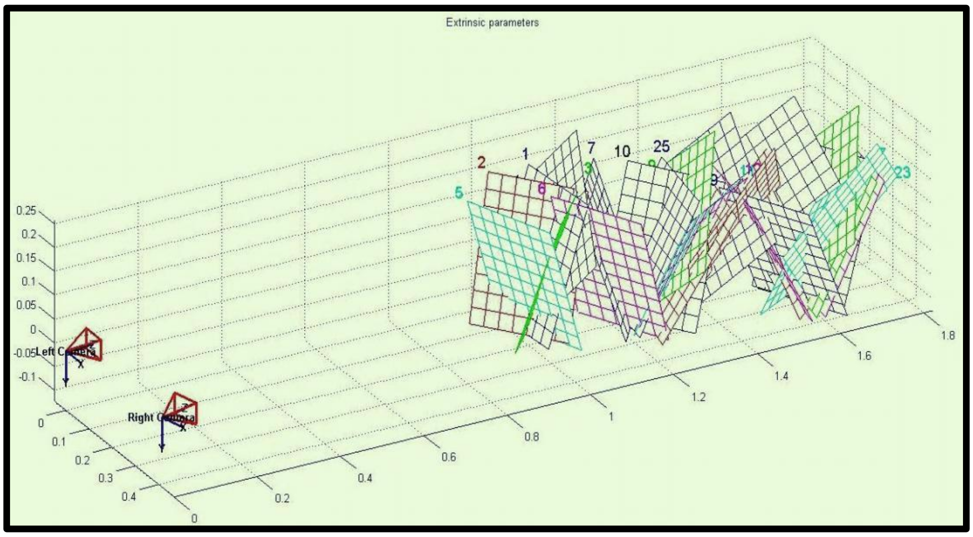


Figure 1.16. Extrinsic parameters of the stereo rig

It is very important to keep in mind for the corner extraction process that for each pair, the same set of points must be selected in the left and right images. This means that same grid of points and same origin point must be chosen. In this experiment the left upper corner has been chosen as the origin of the grid.

Finally, the stereo images can be rectified with the “Rectify the calibration images” option in the stereo toolbox. (With epipolar lines matching with the horizontal scanned lines). In addition to generating the rectified images, the script also saves the new set of calibration parameters under Calib_Results_stereo_rectified.mat (valid only for the rectified images).[11]

There are several advantages of using a stereo rig rather than a single camera. The epipolar geometry associated with the left and right images of the stereo rig remains unchanged over time and therefore it is possible to collect several image pairs and to compute a common fundamental matrix for all these image pairs rather than for individual pairs.

The tracking of image point-features over time is facilitated by the fact that the two cameras are rigidly attached to each other, and the problem of upgrading a 3D projective structure to an affine structure can be solved, either in closed form or using linear minimization methods which is a great advantage over nonlinear methods generally associated with single camera calibration.[12]

1.3.2.2. Stereo Triangulation

The stereo triangulation is a process that computes the 3D locations of a group of points given their left and right image projections. We can compute the 3D location of the grids points extracted on the one of the image pairs in the experiment.

For example, after running through the complete stereo calibration, the image projections of the grid points on the right and left images are available in the variables **x_left_1** and **x_right_1** for images (left01.jpg, right01.jpg).

In order to triangulate those points in space, we execute **stereo_triangulation.m** by inputting **x_left_1,x_right_1**, the extrinsic stereo parameters **om** and **T** and the left and right camera intrinsic parameters:

$$\left[\mathbf{Xc}_{\mathbf{1}_{\mathbf{left}}}, \mathbf{Xc}_{\mathbf{1}_{\mathbf{right}}}\right] = \text{stereo \_triangulation(1.22)}$$

$$(\mathbf{x\_left\_1}, \mathbf{x\_right\_1}, \mathbf{om}, \mathbf{T}, \mathbf{fc\_left}, \mathbf{cc\_left}, \mathbf{kc\_left}, \mathbf{alpha\_c\_left}, \mathbf{fc\_right}, \mathbf{cc\_right}, \mathbf{kc\_right}, \mathbf{alpha\_c\_right}) \quad (1.22)$$

The output variables **Xc_1_left** and **Xc_1_right** are the 3D coordinates of the points in the left and right camera reference frames respectively (observe that Xc_1_left and Xc_1_right are related to each other through the rigid motion equation(1.23)[11]

$$\mathbf{Xc\_1\_right = R * Xc\_1\_left + T} \qquad (1.23)^{11}$$

2. Experimentation in Laboratory

We can simply call "two sphere experiment" , basically there are 2 spheres which are placed with certain distance from each other .The idea is mount these two sphere on CMM machine and take photos in known position where we already generate single and stereo calibration for two cameras. Then by using the 2D coordinates of the center of the spheres as an input to the specific matlab code and gets the 3D coordinates of the center of the spheres in camera coordinate system. Then create a relation between known coordinates and out-put coordinates from the matlab code in order to obtain uncertainty map in the certain working volume which we are going to describe.

2.1. The Aim of the Experiment

The main target is to create an uncertainty map of certain volume where we have the stereo calibration. The point is which parameters have influence on the uncertainty and how much they are effective respect to other parameters.

2.2.The equipment's and initial parameters

For the experiment we need 2 cameras, 2 precise sphere mounted on bar , CMM machine , two tripod to hold the cameras , adjustable lights , remote control for the cameras.

One of the important thing is cameras should have same lenses and the initial settings such as ISO , focus distance , resolution should be same.

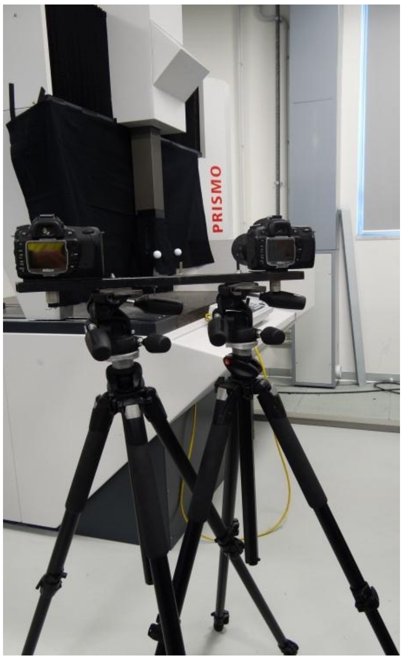


Figure.2.1. Fixed cameras on the tripods

As we can see on the figure.2.1 the cameras fixed on the tripods, then as we mentioned before cameras are settled to these settings:

- ISO 200
- Focus : Manual focus at a distance of 1.5 meter
- Format : Jpg
- Resolution : 3872 x 2592

The main reason to use same settings for both cameras is after the all process to be able to compare two cameras in a more reliable way.

2.3. Single Calibration of Cameras

After the settings the system is ready for the calibration step. First we calibrate the 2 cameras separately which calls single camera calibration. Then by using single camera calibration data’s we will be able to obtain stereo calibration step.

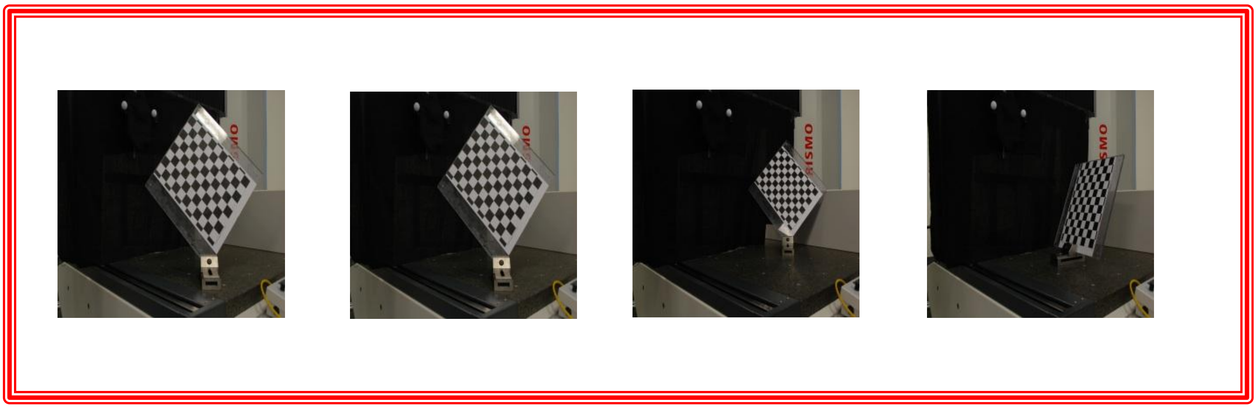


Figure.2.2. Positions of checker board

In the calibration step one of the most important point is our calibration volume should cover the working volume that where we are going to try to generate the uncertainty map. Because calibration data’s of cameras are valid in the certain volume where calibration is done, so we placed the checker board on 25 different positions. And the board took place close and far from the cameras and also photos taken for different angles.

2.3.1. Left Camera Calibration

Once we get the 25 photos of checker board from each camera then we will be ready to do calibration. As explained in the theory of the calibration we obtain the calibration data of left camera by following defined steps.

The Intrinsic parameters of left camera:

- **Focal length:** fc = [3929.5 ; 3943.2];
- **Principal point:** cc = [1936; 1296];
- **Skew coefficient:** alpha_c = 0;
- **Distortion coefficients:** kc = [-0.04 ; 0.06 ; - 0.0004; -0.004 ; 0];
- **Focal length uncertainty:** fc_error = [17.4 ; 17.2];
- **Principal point uncertainty:** cc_error = [0; 0];
- **Skew coefficient uncertainty:** alpha_c_error = 0;
- **Distortion coefficients uncertainty:** kc_error = [0.03 ; 0.14 ; 0.00 ; 0.003 ; 0];

By using matlab tool-box the intrinsic parameters which are well defined in the theoretic part are obtained.

2.3.2. Right Camera Calibration

By using same procedure of left camera calibration we obtained the right calibration data's.

The Intrinsic parameters of left camera:

- **Focal length:** fc = [3789.1 ; 3815.8];
- **Principal point:** cc = [1936; 1296];
- **Skew coefficient:** alpha_c = 0;
- **Distortion coefficients:** kc = [-0.072 ; 0.065 ; -0.0007 ; -0.0009; 0];
- **Focal length uncertainty:** fc_error = [15.2 ; 16.4];
- **Principal point uncertainty:** cc_error =[0; 0];
- **Skew coefficient uncertainty:** alpha_c_error = 0;
- **Distortion coefficients uncertainty:** kc_error = [0.02 ; 0.15 ; 0.0009; 0.001 ; 0];

2.4. Stereo Calibration of Two Cameras

Once we have the calibration data's of the single cameras, we can perform the stereo calibration. Again by using matlab tool-box we can easily perform this step. For stereo calibration tool-box has specific command which is "stereo_gui".

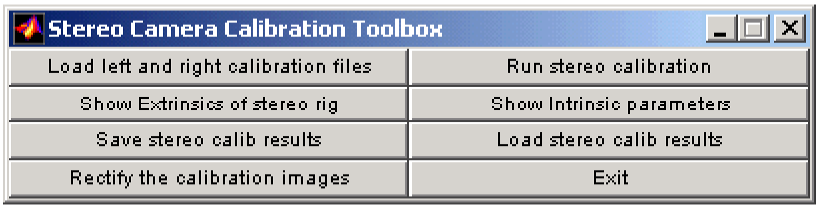

Stereo Camera Calibration Toolbox
Load left and right calibration files
Run stereo calibration
Show Extrinsics of stereo rig
Show Intrinsic parameters
Save stereo calib results
Load stereo calib results
Rectify the calibration images
Exit

Figure.2.3. Stereo Calibration Command Window

Once we write the "stereo_gui" command to matlab we see the menu which we can see above. This useful menu allows us to load the single calibration data's and by using them create the stereo calibration. After loading the single calibration data's we just simply use the "Run stereo calibration" command in order to obtain the stereo calibration data's.

Stereo calibration parameters:

Intrinsic parameters of left camera:

- **Focal Length:** fc_left = [3956.3 , 3976.3] ± [9.5 8]
- **Principal point:** cc_left = [1936, 1296] ± [0 0]
- **Skew:** alpha_c_left = [0] ± [0] => angle of pixel axes = 90± 0degrees
- **Distortion:** kc_left = [-0.096 0.29 -0.00026 -0.0021 0] ± [0.03 0.15 0.0005 0.003 0]

Intrinsic parameters of right camera:

- **Focal Length:** fc_right = [3826.9   3856.4] ± [8.3 8.3]
- **Principal point:** cc_right = [1936 1296] ± [0   0]
- **Skew:** alpha_c_right = [0] ± [0]   => angle of pixel axes = 90± 0degrees
- **Distortion:** kc_right = [-0.073   -0.016   0.0004   -0.00018 0] ± [0.02   0.14   0.00067 0.001 0]

Extrinsic parameters (position of right camera wrt left camera):

- **Rotation vector:** om = [0.01109   0.24241 0.00444] ± [0.00015   0.00046 0.00020]
- **Translation vector:** T = [-0.37076   -0.00069 0.06835] ± [0.00057   0.00016 0.00183]

The " stereo_gui window" includes the other useful commands such as " show extrinsic of stereo rig" which shows the 25 different positions of the checker board on the camera coordinate system and the position of the camera respect to each other .

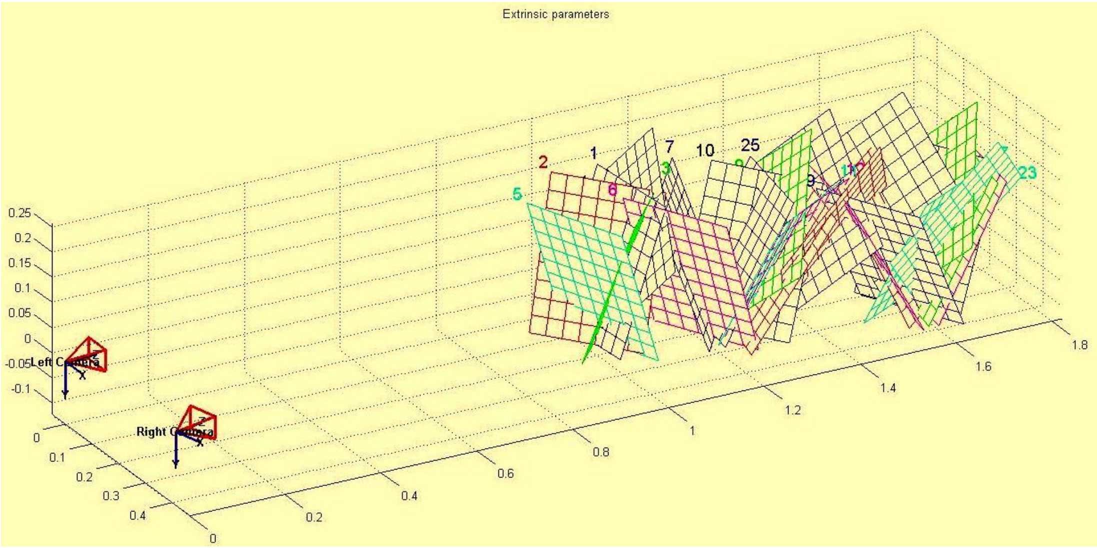


Figure.2.4. The position of cameras and positions of checker board on the camera coordinate system.

As we can see above the 25 different position of checker board and their distance to the cameras in camera reference system. We can say that calibration volume starts from 0.9 meter and finishes at 1.8 meter, so that means our calibration data's are useful in that range. That information will be helpful while we will be defining the working volume for the experiment; the working volume should be covered by the calibrated volume.

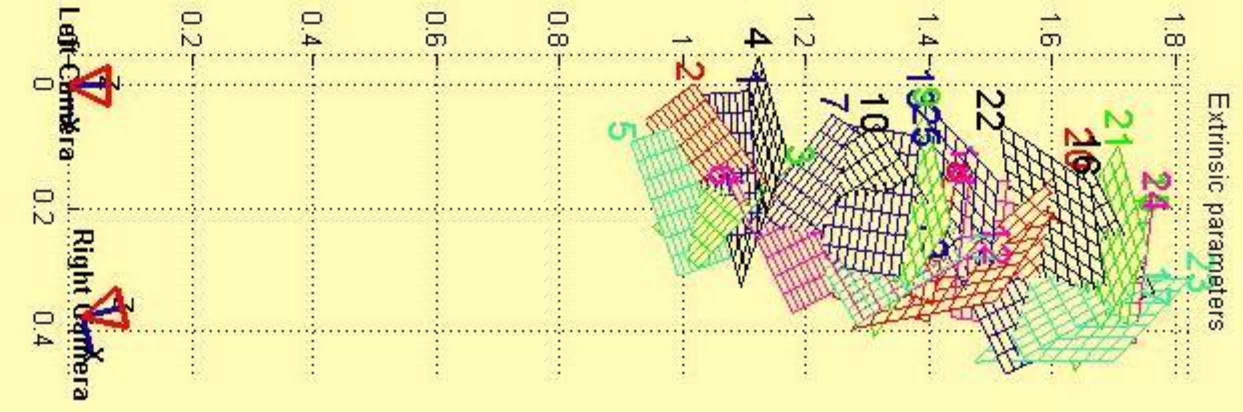


Figure.2.5. The distance between the cameras

The other important information is the distance between two cameras; according to stereo calibration data distance between cameras is 0.37076 meter, and we roughly estimated in advance this distance which is 0.38 meter. That information shows that the stereo calibration data's are sufficiently reliable.

2.5. Working Volume of Experiment

As it is mentioned before one of the important step is to define “working volume”, there is no such a rule for this definition. The important thing is that working volume should be placed in the intersection of the cameras view zone. Basically working volume should be visible from both cameras.

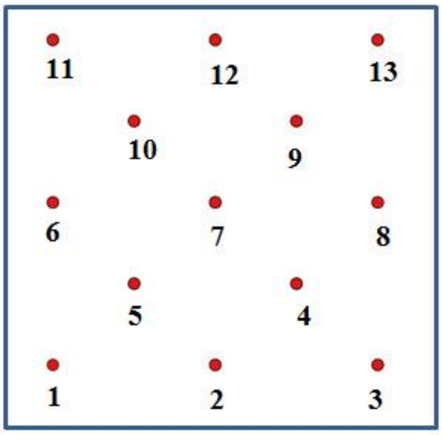


Figure.2.6. The order of points on the plane

First we define the orientation of the points on the plane, as we can see on figure.2.6 we have 13 points on one plane. Basically we can say that this is the first plane of our working volume.

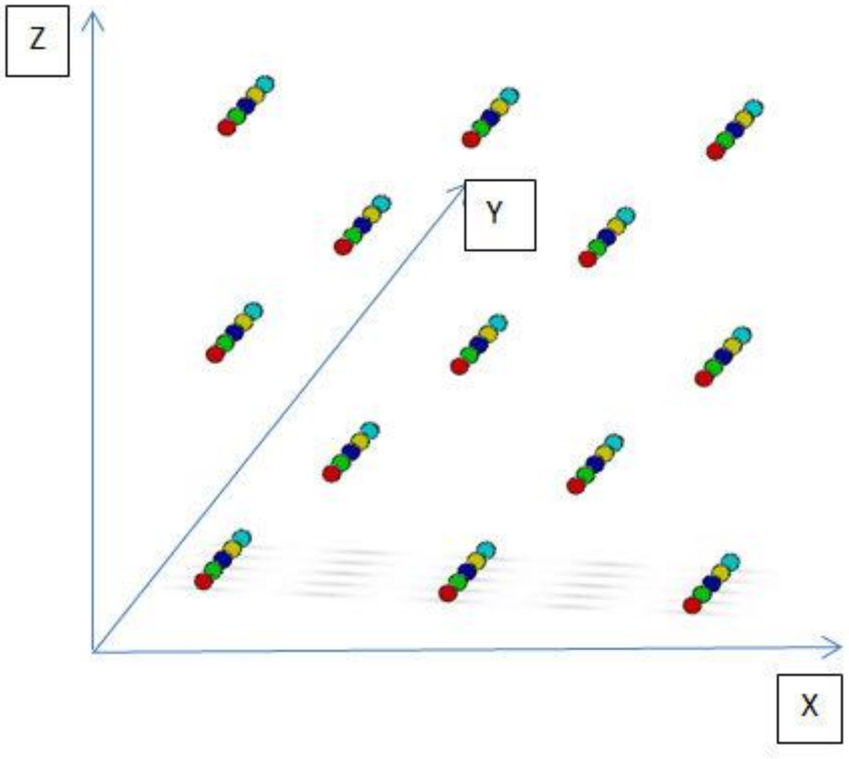


Figure.2.7. The order of planes

As we can see that after creating first plane, along the Y direction we have totally 5 planes which have also 13 same points on it. This coordinate system is CMM coordinate system, because we define these points and planes on the CMM machine. On the figure above different colors are representing the different planes, so now we can simply say that the working volume is defined. The overall working volume can fit in the cube shape which has 280 mm x 280 mm x 280 mm dimensions; basically we can the boundary of the working volume.

2.6. Two spheres and Known Positions

We have two spheres which are exactly have same dimensions; these spheres are mounted on the plastic bar with distance from each other 200 mm.

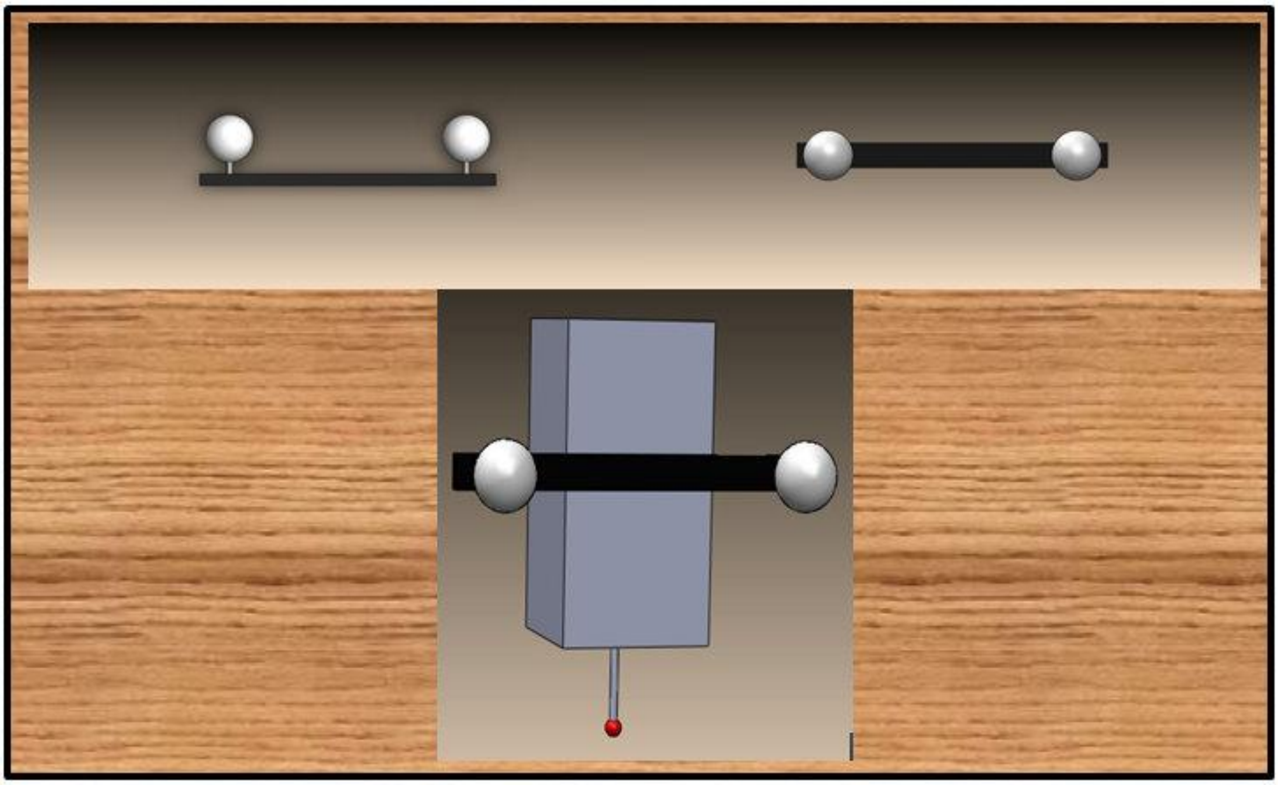


Figure.2.8. The two spheres on the plastic bar

On the figure above we can see the spheres position on the plastic bar on which spheres are mounted. Now the spheres are ready to mount on the CMM machine head which moves to the adjustable known position in the CMM coordinate system.

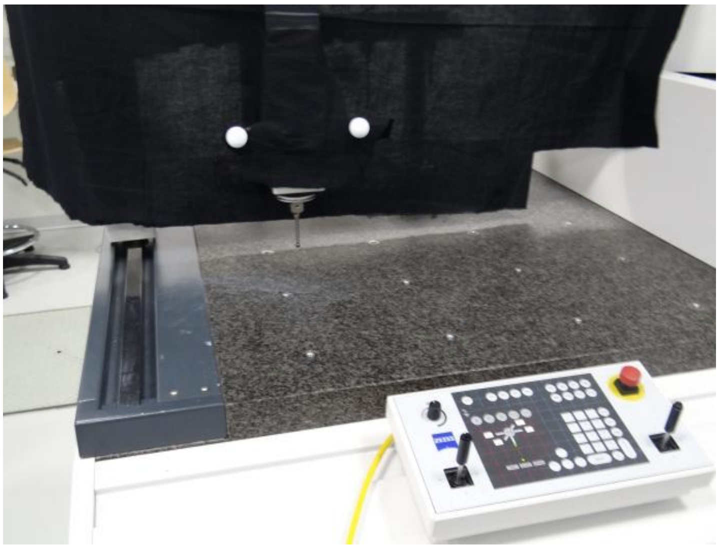


Figure.2.9. The spheres are mounted on the head of CMM machine

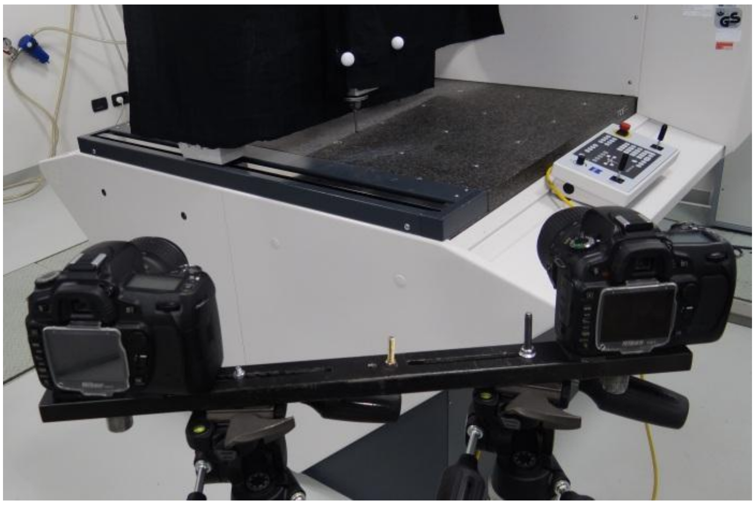


Figure.2.10. View angle from the cameras

On the figures above we can see that the two spheres are mounted on the head of CMM, also the cameras view angle to the spheres position is visible. And that view angle provides that the sphere on the left is always closer than the right sphere to the cameras , which is going to allow us to get two different data's for the same position of the CMM head.

2.6.1. The Known positions

The head of the CMM now is ready to move to the know positions which are on the planes that we already defined before.

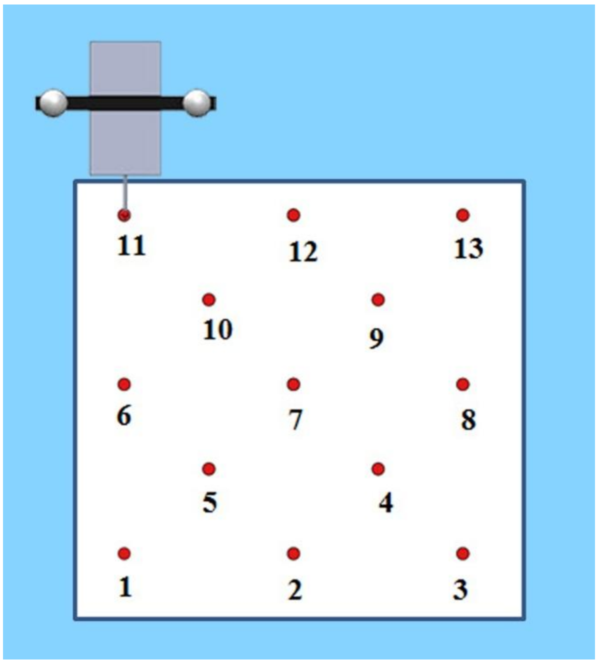


Figure.2.11. The head of CMM on the first plane

Figure.2.11 is helpful to show the move of head on the plane. As we can see that head coordinate point is on the 11th position and for that position by taking photo we will have 2 data set, one from the left sphere, one from the right sphere. All these 13 point on the planes are the known coordinates in the CMM coordinate system.

First Plane Coordinates													
	1	2	3	4	5	6	7	8	9	10	11	12	13
X	10	150	290	220	80	10	150	290	220	80	10	150	290
Y	26	26	26	26	26	26	26	26	26	26	26	26	26
Z	-55	-55	-55	15	15	85	85	85	155	155	225	225	225
Second Plane Coordinates													
	1	2	3	4	5	6	7	8	9	10	11	12	13
X	10	150	290	220	80	10	150	290	220	80	10	150	290
Y	96	96	96	96	96	96	96	96	96	96	96	96	96
Z	-55	-55	-55	15	15	85	85	85	155	155	225	225	225
Third Plane Coordinates													
	1	2	3	4	5	6	7	8	9	10	11	12	13
X	10	150	290	220	80	10	150	290	220	80	10	150	290
Y	166	166	166	166	166	166	166	166	166	166	166	166	166
Z	-55	-55	-55	15	15	85	85	85	155	155	225	225	225
Fourth Plane Coordinates													
	1	2	3	4	5	6	7	8	9	10	11	12	13
X	10	150	290	220	80	10	150	290	220	80	10	150	290
Y	236	236	236	236	236	236	236	236	236	236	236	236	236
Z	-55	-55	-55	15	15	85	85	85	155	155	225	225	225
Fifth Plane Coordinates													
	1	2	3	4	5	6	7	8	9	10	11	12	13
X	10	150	290	220	80	10	150	290	220	80	10	150	290
Y	306	306	306	306	306	306	306	306	306	306	306	306	306
Z	-55	-55	-55	15	15	85	85	85	155	155	225	225	225

Table.2.1 The known coordinates of 13 points on the all planes

After defining planes, points and the known coordinates of these points we are ready for the next step where we are going to take the photos.

2.6.2. Photos of Spheres on The Known Positions

This is one of the most important step of the experiment; because this step gives us the main data’s in order to understand the regime of uncertainties. Since we fixed the cameras on the tripods we did not move them, we kept them in the same position where we obtain the calibration data’s.

As we mentioned before now we move the head of CMM to the 13 points on the 5 different planes and take photo for each position. We took 5 photos for each position because in this way we can have more data for the same position and repetition makes them more reliable and meaningful. During this step we used remote control for taking photos in order to prevent the photos from the noise which can cause from the vibration of cameras.

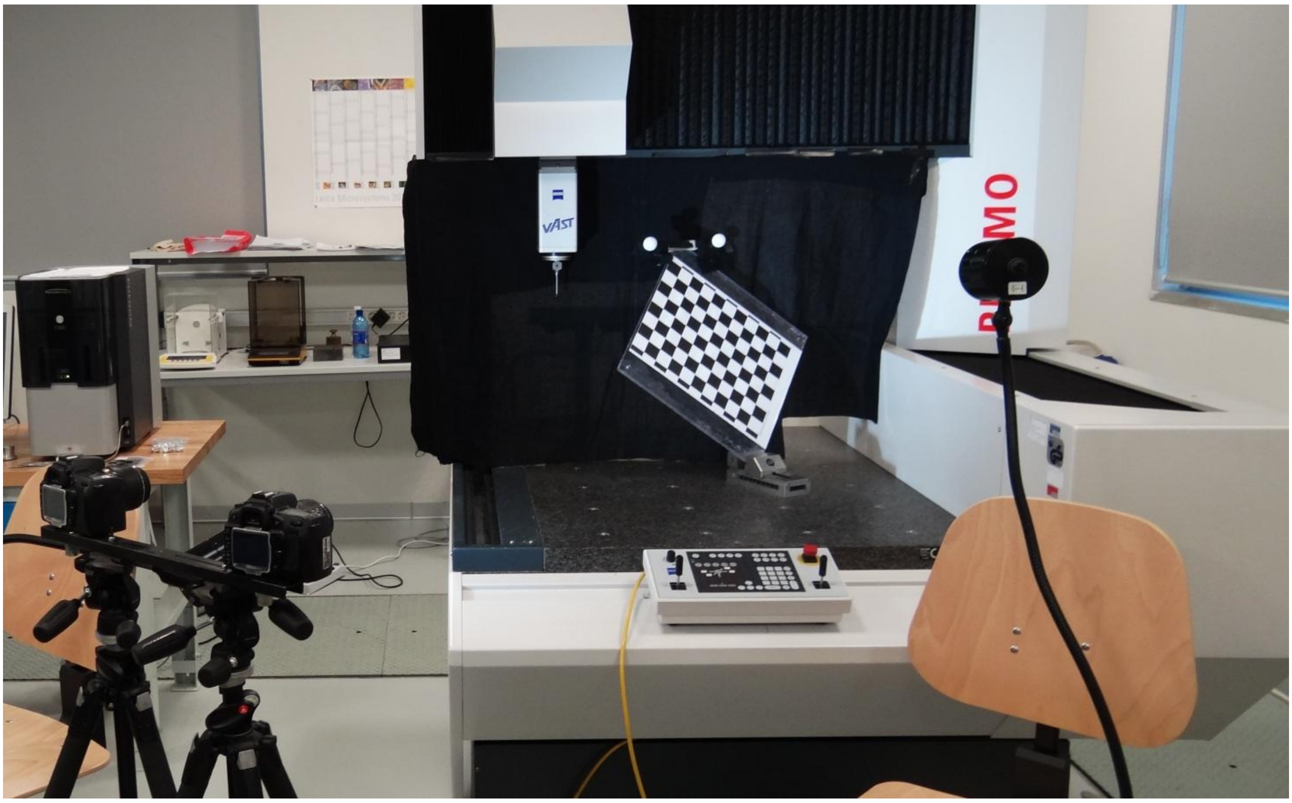


Figure.2.12. Lightening of the spheres by adjustable lights

The other important point of this step is lightening system, for this important parameter we used adjustable lights. While taking photos we change the position of the light and the quantity of the light for each position in order to get more reliable data's from the taken photos.

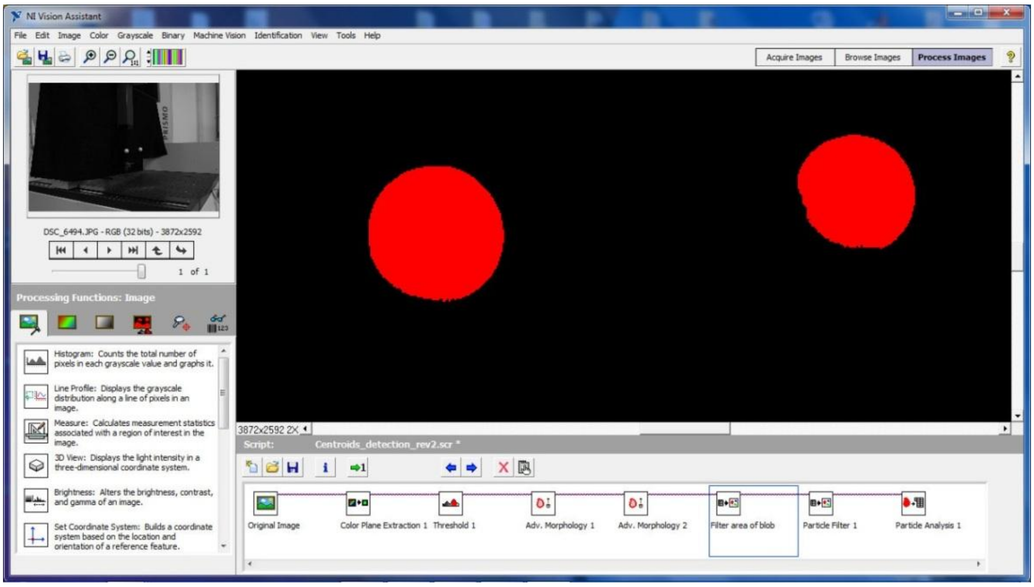
Since we complete this important step where we have taken photos on 13 different points with 5 repetition for 5 planes , we have 650 photos totally 325 of it from left camera other half from the right camera , that means we collect the data's which we need for the next step .

3. Data Analysis of Experiment

Since we have the photos which are the main input data for the analyzing process, we can start to analyze them and try to attribute a meaning to these data's then we will be able to reach final phase where we will have conclusions and comments.

3.1. Centroid Detection

As we mentioned that we have the 650 photos; we use this photos in order to get information such as the center coordinates of the spheres which is the most important data for our next process. In this step we used software which calls "National Instruments Vision Assistant 2009".Basically we upload the photos one by one to the software then by applying some filters and by using the tools of the small program we get the center coordinates of the spheres on the photo 2D pixel coordinate system.



Histogram: Counts the total number of pixels in each grayscale value and graphs it.

Line Profile: Displays the grayscale distribution along a line of pixels in an image.

Measure: Calculates measurement statistics associated with a region of interest in the image.

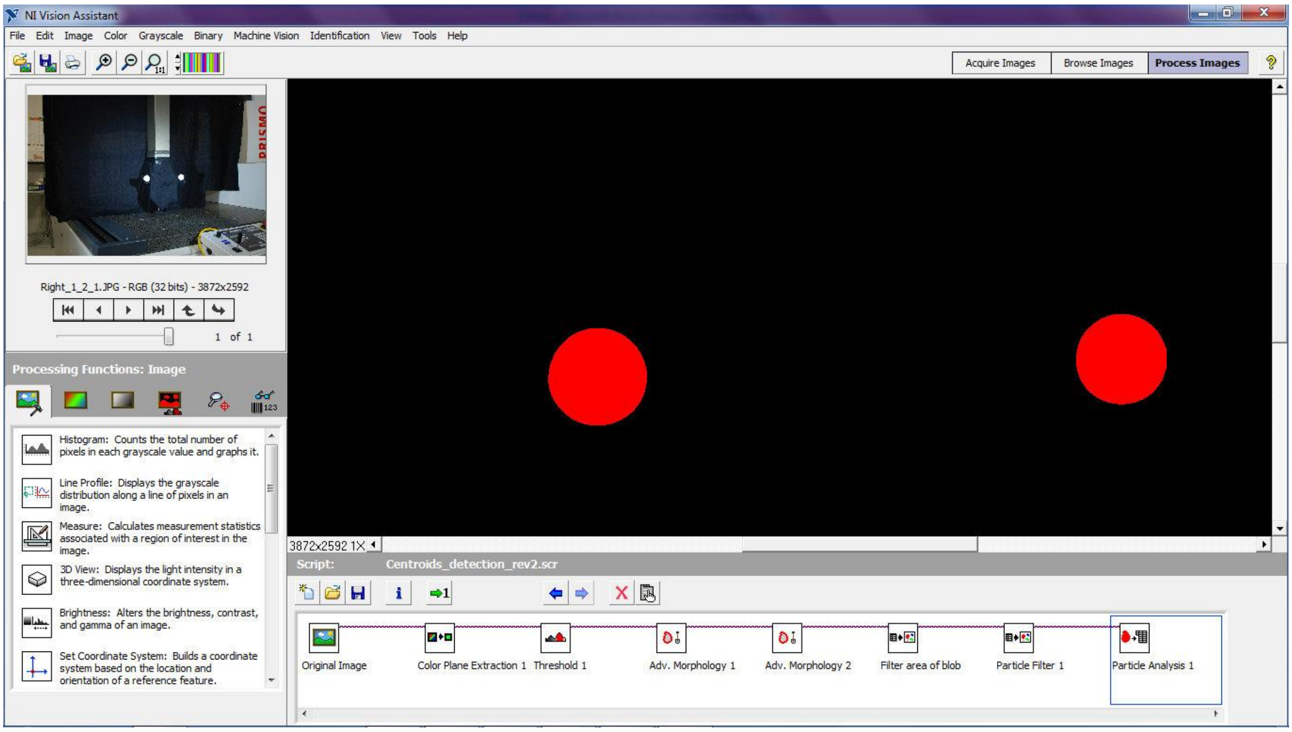
3D View: Displays the light intensity in a three-dimensional coordinate system.

Brightness: Alters the brightness, contrast, and gamma of an image.

Set Coordinate System: Builds a coordinate system based on the location and orientation of a reference feature.

Figure.3.1. Centroid detection of spheres without light

As we can see on figure.3.1 without adjustable lights after applying threshold, the shapes of the red areas are not like circle so these shapes are not able to give the real centroid of the spheres on the pixel coordinate system. As we mentioned before adjustable lights are really important in order to get more meaningful data. After that learning we repeated the experiment with adjustable lights which have led us to have more reliable and meaningful data's.



Histogram: Counts the total number of pixels in each grayscale value and graphs it.

Line Profile: Displays the grayscale distribution along a line of pixels in an image.

Measure: Calculates measurement statistics associated with a region of interest in the image.

3D View: Displays the light intensity in a three-dimensional coordinate system.

Brightness: Alters the brightness, contrast, and gamma of an image.

Set Coordinate System: Builds a coordinate system based on the location and orientation of a reference feature.

Figure.3.2. Centroid detection of spheres with light

On the figure above as we can see the original picture on the left upper part of the window, and also on the bottom it is possible to see the step by step filter and image tool that applied to the original photo. After applying this process we can see these two spheres which are represented in red color. Thanks to adjustable lights we have almost perfect circular shape after threshold.

Object	Sphere - 1	Sphere - 2
Center of Mass X	1924,19111	2480,6269
Center of Mass Y	1345,19897	1326,34845
Bounding Rect Width	105	96
Bounding Rect Height	103	96
Perimeter	328,66323	303,87509
Area	8524	7301
Holes' Area	0	0
Ratio of Equivalent Ellipse Axes	1,13861	1,12034
Compactness Factor	0,78816	0,79221
Heywood Circularity Factor	1,00421	1,00323

Table.3.1. Data's of two spheres from detection software

As we can on the table above software allows us to have the pixel coordinates of the two spheres ; and additionally we other useful information like circularity factor these number means that the spheres on the photos how much close to the perfect circle shape on the photo plane. This is an important point because if we have shape which is more close to the perfect circle shape that means we have pixel coordinates which are more close to the real centroid coordinates.

3.2. Triangulation

Once we complete the centroid detection step then we are ready for the triangulation step which is one of the most important step of data analysis. Before the analyzing of data's it is beneficial to know about triangulation or stereo triangulation. Once we have the stereo calibration data's it is possible to obtain 3D data of the any point which is visible by the two cameras by using triangulation.

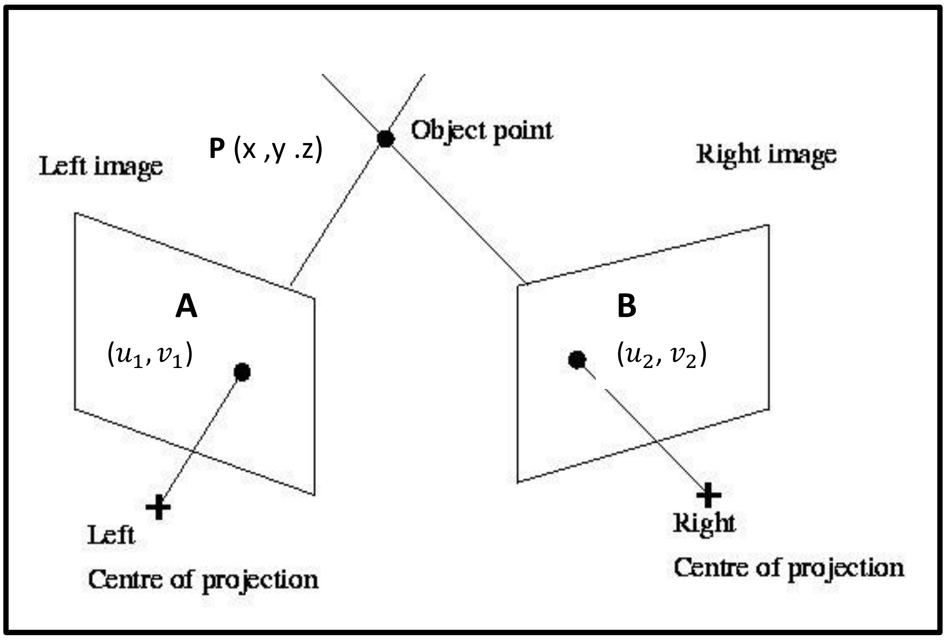


Figure.3.3. Triangulation principle

As we can see above there is 3D point which is point P and there are two projection of that point A on the left and B on the right camera image plane. The point A has the pixel coordinates u_1, v_1 and point B has the pixel coordinates u_2, v_2 which are 2D coordinates on the image plane .

```
[XL,XR]=stereo_triangulation(xL,xR,om,T,fc_left,cc_left,kc_left,alpha_c_left,fc_right,cc_right,kc_right,alpha_c_right)
```

Figure.3.4. Stereo Triangulation Code

We already know the terms like fc , cc , kc , alpha_c, om and T from the calibration step. xL and xR represent the 2D pixel coordinates of the same point from the left and right camera. For example in the figure 3.3 xL= (u_1, v_1) and xR= $(u_2, v_2\ )$. In our experiment case these values are the center 2D coordinates of the two spheres.

Once we have the all the data as input then “stereo_triangulation” code allows to get XL and XR which are 3D coordinates:

- XL: 3xN matrix of coordinates of the points in the left camera reference frame(1)
- XR: 3xN matrix of coordinates of the points in the right camera reference frame(1)

XL and XR are the output of the triangulation; here N represents the number of the points that is triangulated.

3.2.1. Triangulation of Experiment Data's and 3D coordinates

As a result of centroid detection we have the all 2D pixel coordinates of each sphere; we detected 325 images from left camera 325 images from right camera. Also we have the stereo calibration data's, now all inputs are ready for applying to triangulation.

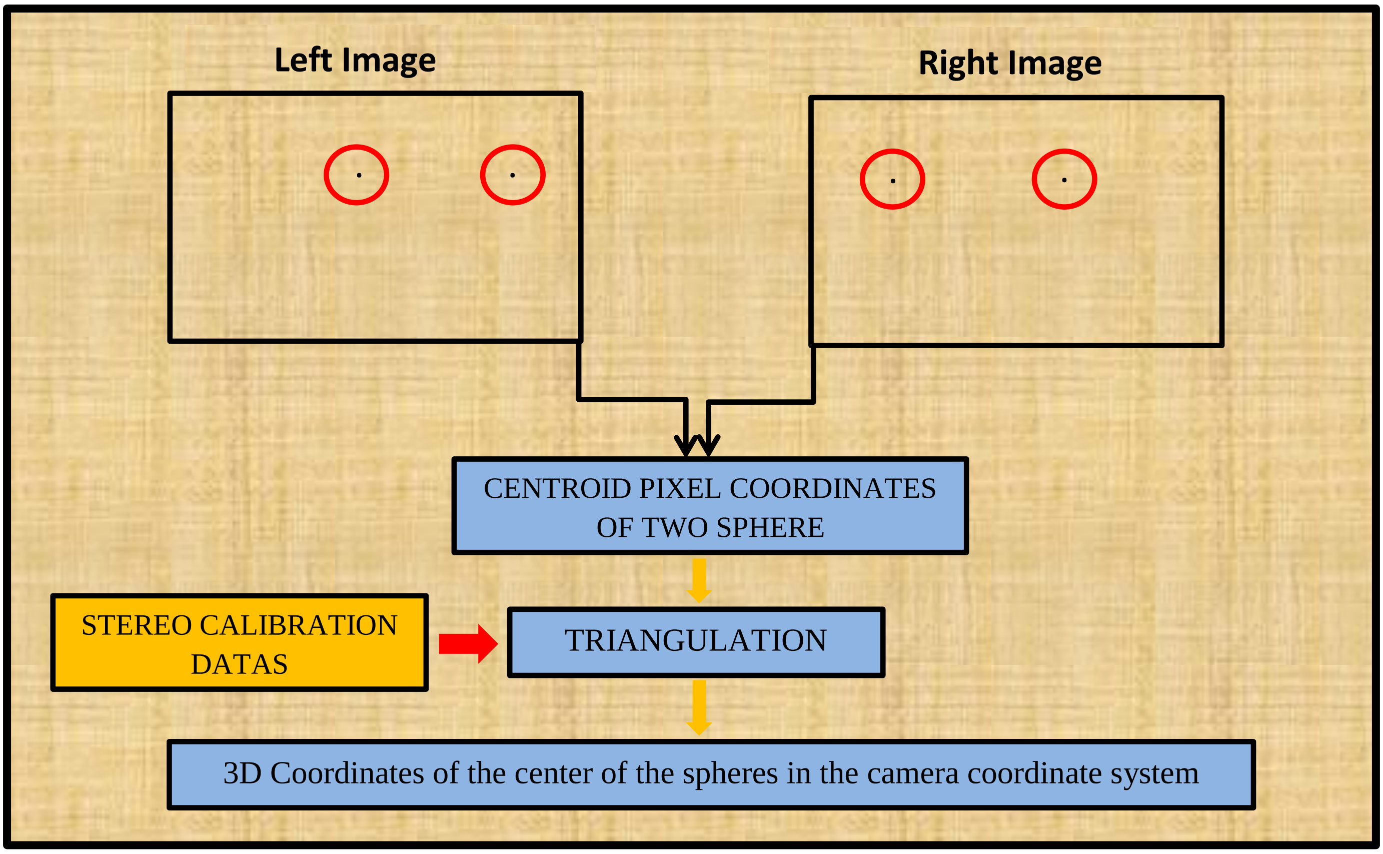

Left Image
Right Image
CENTROID PIXEL COORDINATES OF TWO SPHERE
STEREO CALIBRATION DATAS
TRIANGULATION
3D Coordinates of the center of the spheres in the camera coordinate system

Figure.3.5. Data flow and steps from 2D pixel coordinates to 3D coordinates.

Basically after triangulation we completed the passing from 2D pixel coordinates to 3D coordinates. As a result now we have new data's which are the 3D coordinates of the two spheres in the camera reference system.

First Plane 3D coordinates							
	X	Y	Z		X	Y	Z
1.Point	-0,036	0,027	1,136	8.Point	0,240	-0,113	1,096
	-0,036	0,027	1,136		0,240	-0,113	1,096
	-0,036	0,027	1,136		0,240	-0,113	1,096
	-0,036	0,027	1,136		0,240	-0,113	1,096
	-0,036	0,027	1,136		0,240	-0,113	1,096
2.Point	0,102	0,026	1,117	9.Point	0,170	-0,183	1,105
	0,102	0,026	1,118		0,170	-0,183	1,105
	0,102	0,026	1,117		0,170	-0,183	1,105
	0,102	0,026	1,117		0,170	-0,183	1,105
	0,102	0,026	1,118		0,170	-0,183	1,105
3.Point	0,241	0,026	1,098	10.Point	0,032	-0,182	1,124
	0,241	0,026	1,098		0,032	-0,182	1,124
	0,241	0,026	1,098		0,032	-0,182	1,125
	0,241	0,026	1,098		0,032	-0,182	1,125
	0,241	0,026	1,098		0,032	-0,182	1,124
4.Point	0,171	-0,044	1,107	11.Point	-0,038	-0,251	1,133
	0,171	-0,044	1,107		-0,038	-0,251	1,133
	0,171	-0,044	1,107		-0,038	-0,251	1,133
	0,171	-0,044	1,107		-0,038	-0,251	1,133
	0,171	-0,044	1,107		-0,038	-0,251	1,133
5.Point	0,032	-0,043	1,126	12.Point	0,101	-0,252	1,114
	0,032	-0,043	1,126		0,101	-0,252	1,114
	0,032	-0,043	1,126		0,101	-0,252	1,114
	0,032	-0,043	1,126		0,100	-0,252	1,114
	0,032	-0,043	1,126		0,101	-0,252	1,114
6.Point	-0,037	-0,112	1,135	13.Point	0,239	-0,253	1,095
	-0,037	-0,112	1,135		0,239	-0,253	1,095
	-0,037	-0,112	1,135		0,239	-0,253	1,095
	-0,037	-0,112	1,135		0,239	-0,253	1,095
	-0,037	-0,112	1,135		0,239	-0,253	1,095
7.Point	0,101	-0,113	1,116				
	0,101	-0,113	1,116				
	0,101	-0,113	1,116				
	0,101	-0,113	1,116				
	0,101	-0,113	1,116				

Table.3.2. 3D Coordinates of 13 points on the first plane (meter)

We can see table above the result of triangulation for the points on the first plane. We already mentioned about 5 repetitions for each position so here it is obvious that the coordinates are almost same for the repetitions, just few point has very small difference.

3.3. Roto Translation

As we know that as a result of our previous step we obtain 3D coordinates of the two spheres and CMM machine coordinates which are defined as a known positions. So that mean we have 3 information cloud, we will perform roto-translation between CMM coordinates and sphere coordinates which are output of triangulation. The main point is we actually need 2 CMM coordinate information cloud in order to roto-translate the other two information cloud of two spheres. Exactly at this point we tried two different cases.

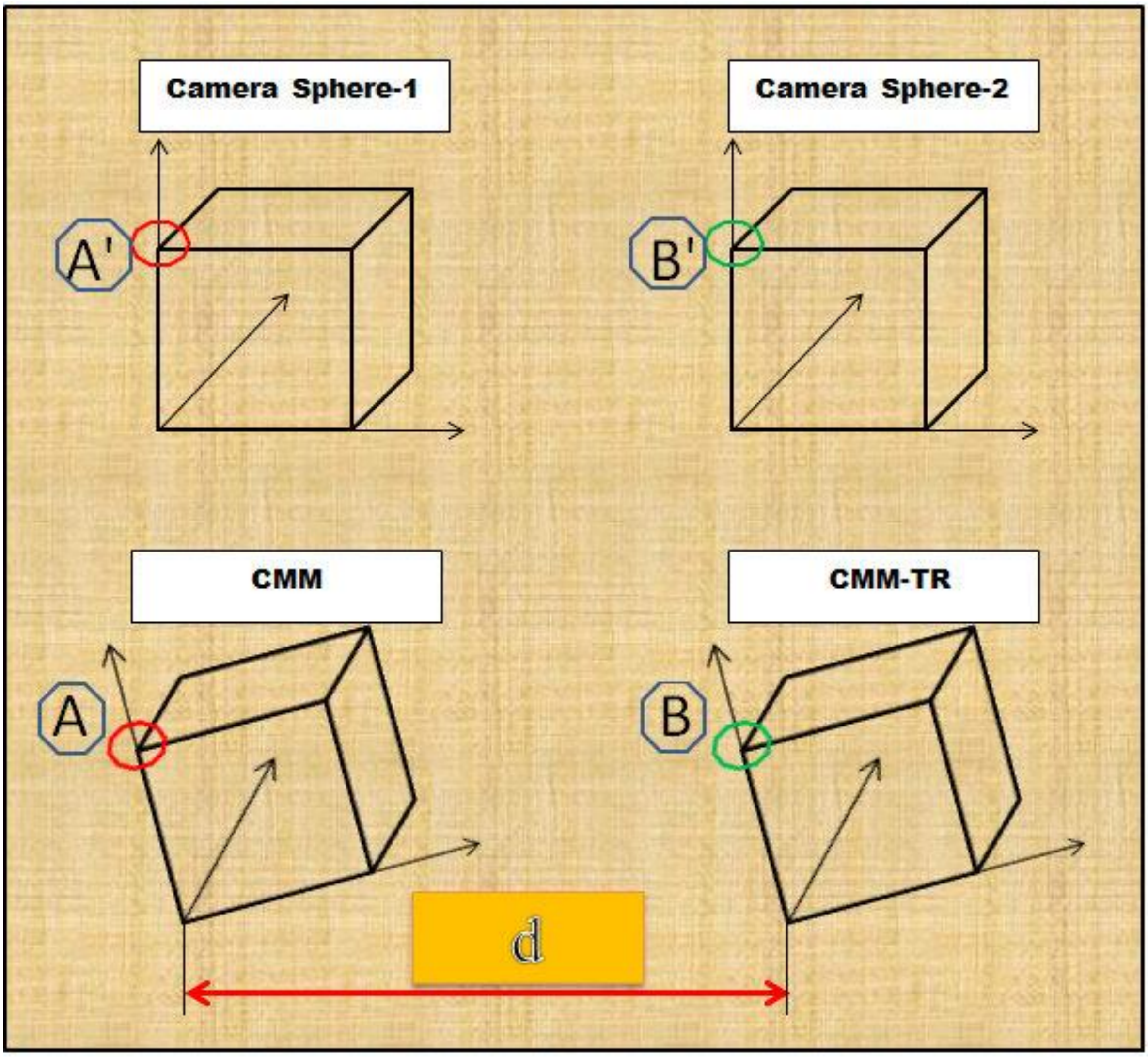


Figure.3.6. Roto-translation between coordinate systems.

On the sketch above Camera Sphere-1 and Camera Sphere-2 represent the xyz coordinates of the two spheres on camera coordinate systems. CMM represents the xyz coordinates on the CMM machine coordinate system, CMM-TR is shifted coordinates system of CMM according to “d” distance. The “d” is defined as the 3D distance between each same single point both on CMM and CMM-TR coordinate system.

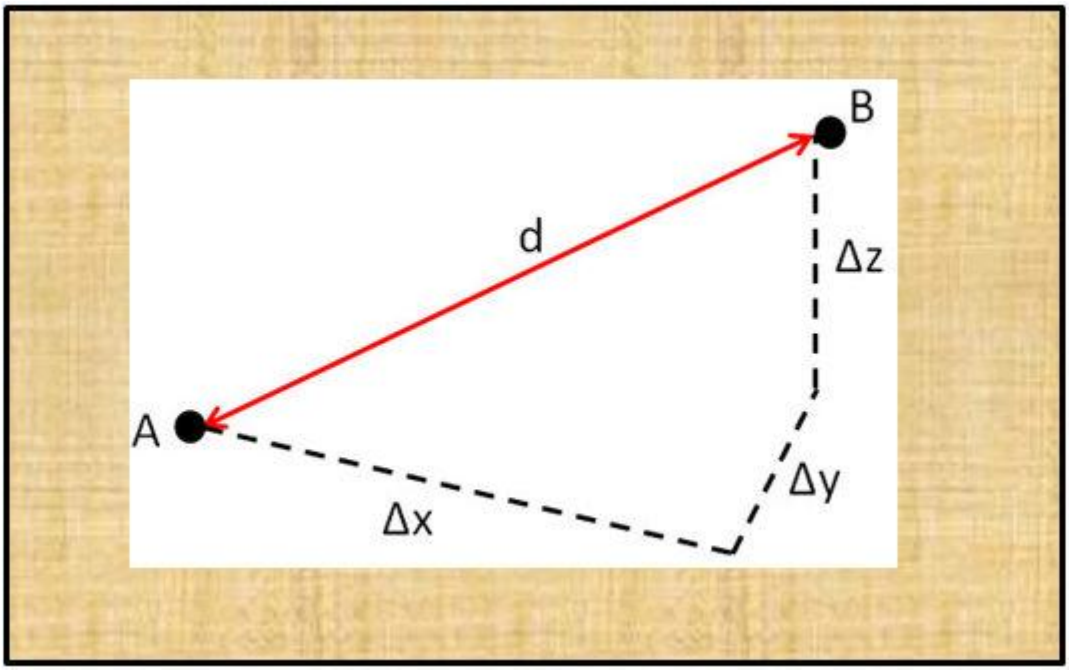


Figure.3.7. The distance between CMM and CMM-TR

For the first case we assume the “d” is not constrain but it is same between each same single point both on the CMM and CMM-TR , for example distance between A and B is fixed but B can move any place for “d” is stable so we can define CMM-TR is shifted form of CMM. Then we performed the roto-translation both between CMM – Camera Sphere-1 and CMM-TR - Camera Sphere-2 as an example A is roto-translated to A' and B is roto-translated to B’.

The figure above shows the distance “d” between CMM and CMM-TR and when d is not imposed we get these values:

- Δx= 0.131 m
- Δy=0.152 m
- Δz=0.0021 m

Then by using Euclidean formula d=$\sqrt{\Delta x^2 + \Delta y^2 + \Delta z\text{^}2}$; it is calculated d=0.2006 m. This is an important point to highlight because we knew in advance d is measured d=0.200 m and the result of calculation is close to the actual one.

For the second case d is imposed as 0.2 m; the program that is used for this we set the Δz as a free component that we calculated from Euclidean formula we get these values:

- Δx= 0.130 m
- Δy=0.152 m
- Δz=0.002 m

Then the numbers are close, we can say the data’s are collected from the process are reliable both to have good overlap by using roto-translation and to shift the CMM coordinate system.

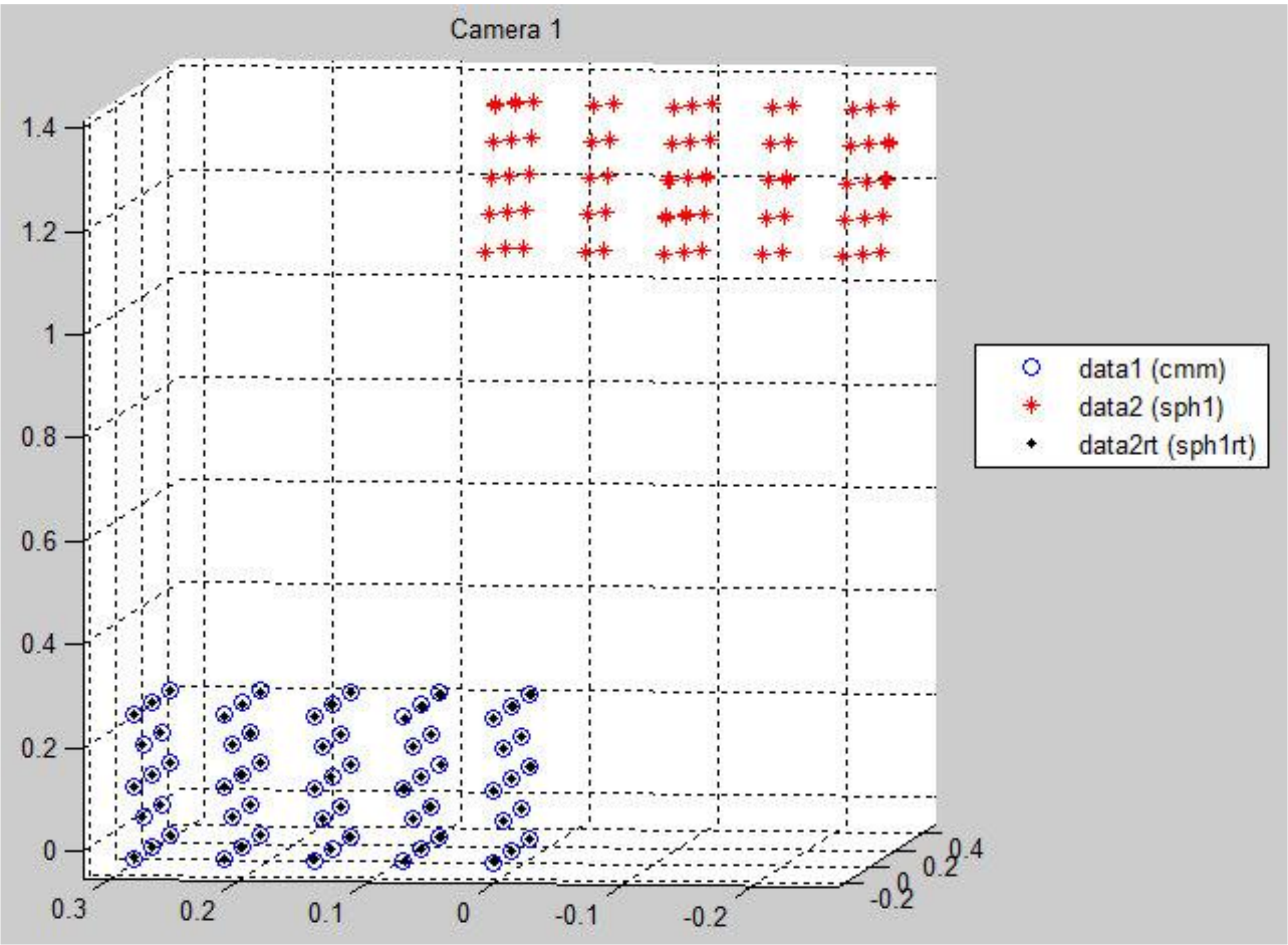


Figure.3.8. Roto translation of first sphere data's

Figure above represents the roto-translation between xyz coordinates of center of the sphere-1 result from triangulation and CMM xyz coordinates. It is possible to see that two information clouds are overlapped.

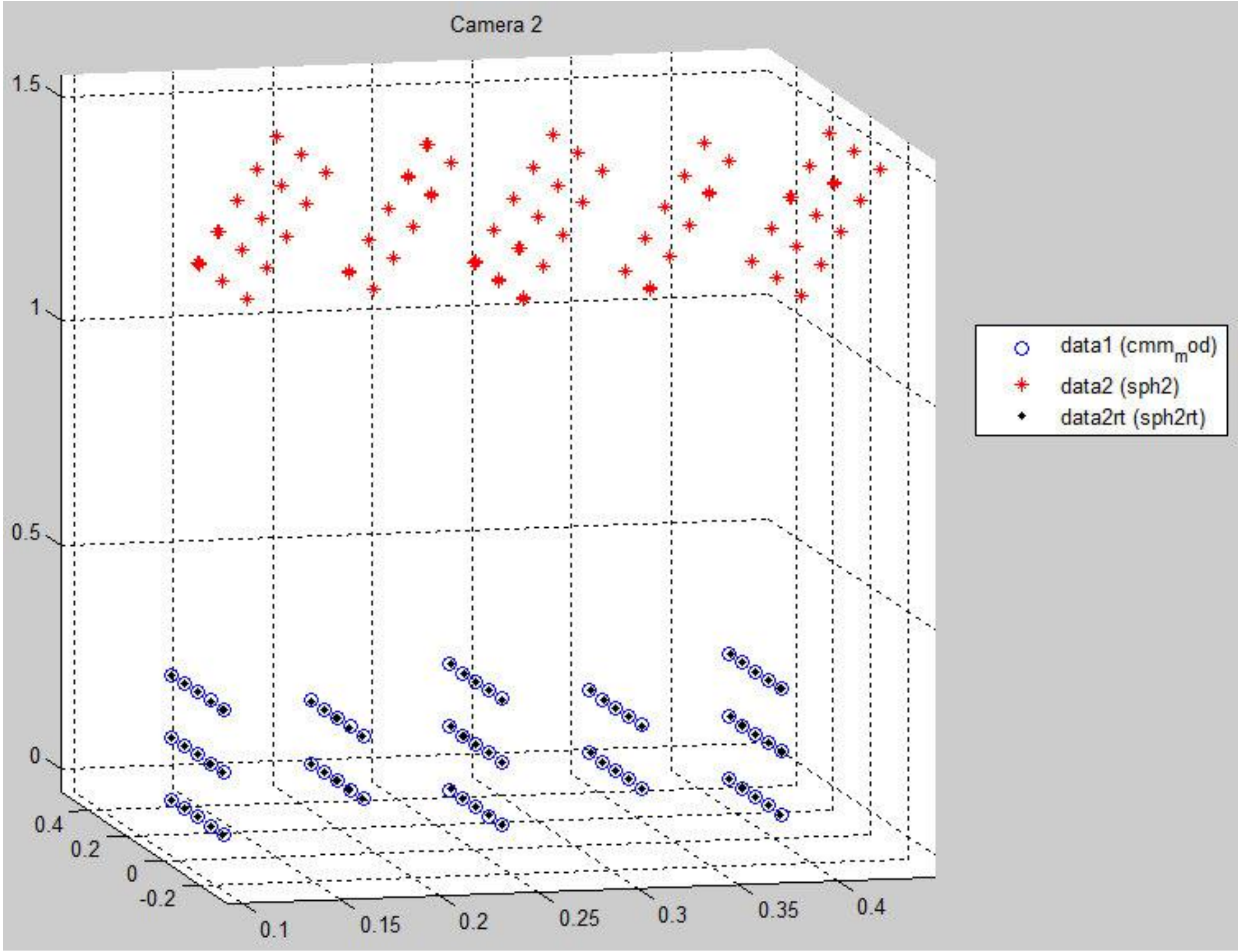


Figure.3.9. Roto translation of second sphere data's

Figure above represents the roto-translation between xyz coordinates of center of the sphere-2 result from triangulation and CMM xyz coordinates. It is possible to see that two information clouds are overlapped.

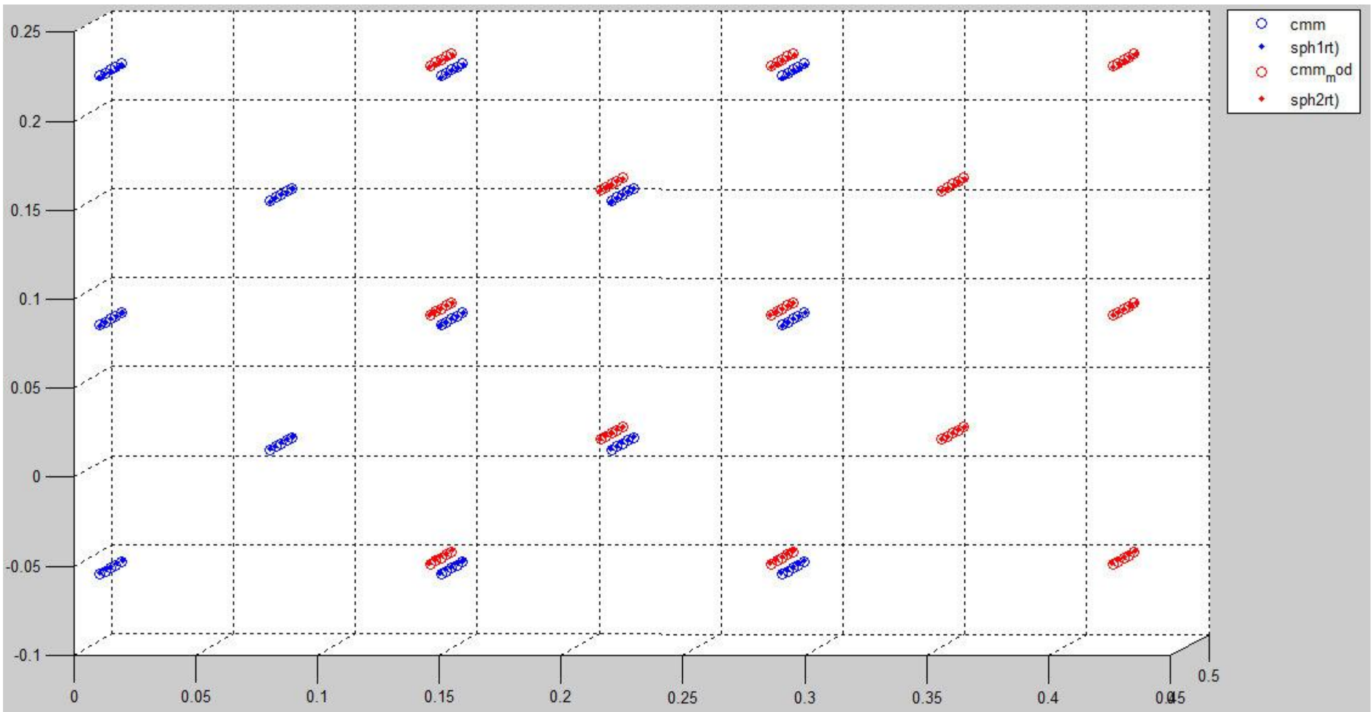


Figure.3.10. Roto translation of spheres.

Figure above shows the both sphere-1 and sphere-2 xyz coordinates roto translation to the cmm coordinates, here it is possible to see overlapped data's with the orientation of points and planes which are defined as working volume.

3.4. Uncertainties and Discrepancies

It is beneficial to know about terms like uncertainty, discrepancy before the explanation of experiment data analyze.

At the outset, it is important to recognize that all measurements are wrong in that the measured value (the result) and the right answer (the 'true' value) are different. The difference between these two is the measurement error, which is meant in the sense of a discrepancy, rather than an avoidable mistake (2).

Unfortunately the true value is never precisely known and, by the same token, neither is the measurement error. Uncertainty, on the other hand, takes the form of a range, and, if estimated for an analytical procedure and defined sample type, may apply to all determinations so described. In general, the value of the uncertainty cannot be used to correct a measurement result. (3)

For instance:

$$T = Tm \pm UT$$

Where; T is the true value, Tm is the measured value, and the range ±UT is the measurement uncertainty.

This statement can be taken to mean that the true value, T, probably lies somewhere between Tm - UT and Tm + UT. (2).

From the experimental data's there information data's to allow us to understand the uncertainty by means of discrepancy. We simply calculate discrepancy between xyz Cmm coordinates which we defined as known positions and xyz coordinates which are results of triangulation.

Discrepancy for the spheres:

D = $\sqrt{\Delta x^2 + \Delta y^2 + \Delta z\text{^}2}$

Where: Δx= (x_cmm) – (x_trg)

Δy= (y_cmm) – (y_trg)

Δz= (z_cmm) – (z_trg)

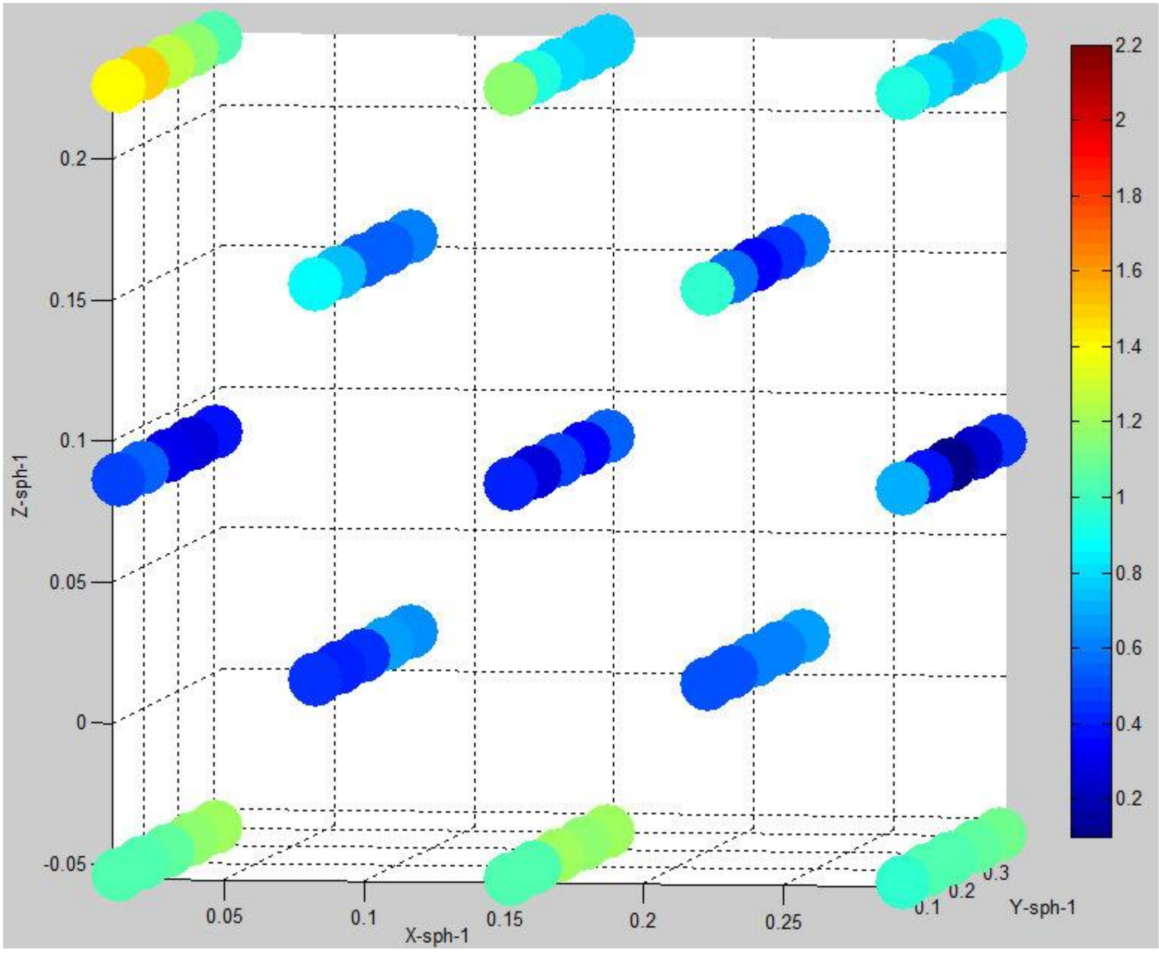


Figure.3.11. Discrepancy of sphere one.

On the figure3.11 it is possible to see the discrepancy for sphere one in Cmm coordinate system. The graph has some regime for uncertainty in the sense of discrepancy. The uncertainties are higher on the lower parts which are first, second and third positions of working volume. Uncertainty of upper part, which include positions 11, 12, 13 in the working volume, it is high as well but obvious that higher than the lower part of working volume. The middle part of working volume includes position with number 4, 5, 6, 7, 8, 9 and 10 where uncertainties are lower respect to other parts of working volume and in the middle part it is visible that uncertainty distribution is more homogeneous then the other parts.

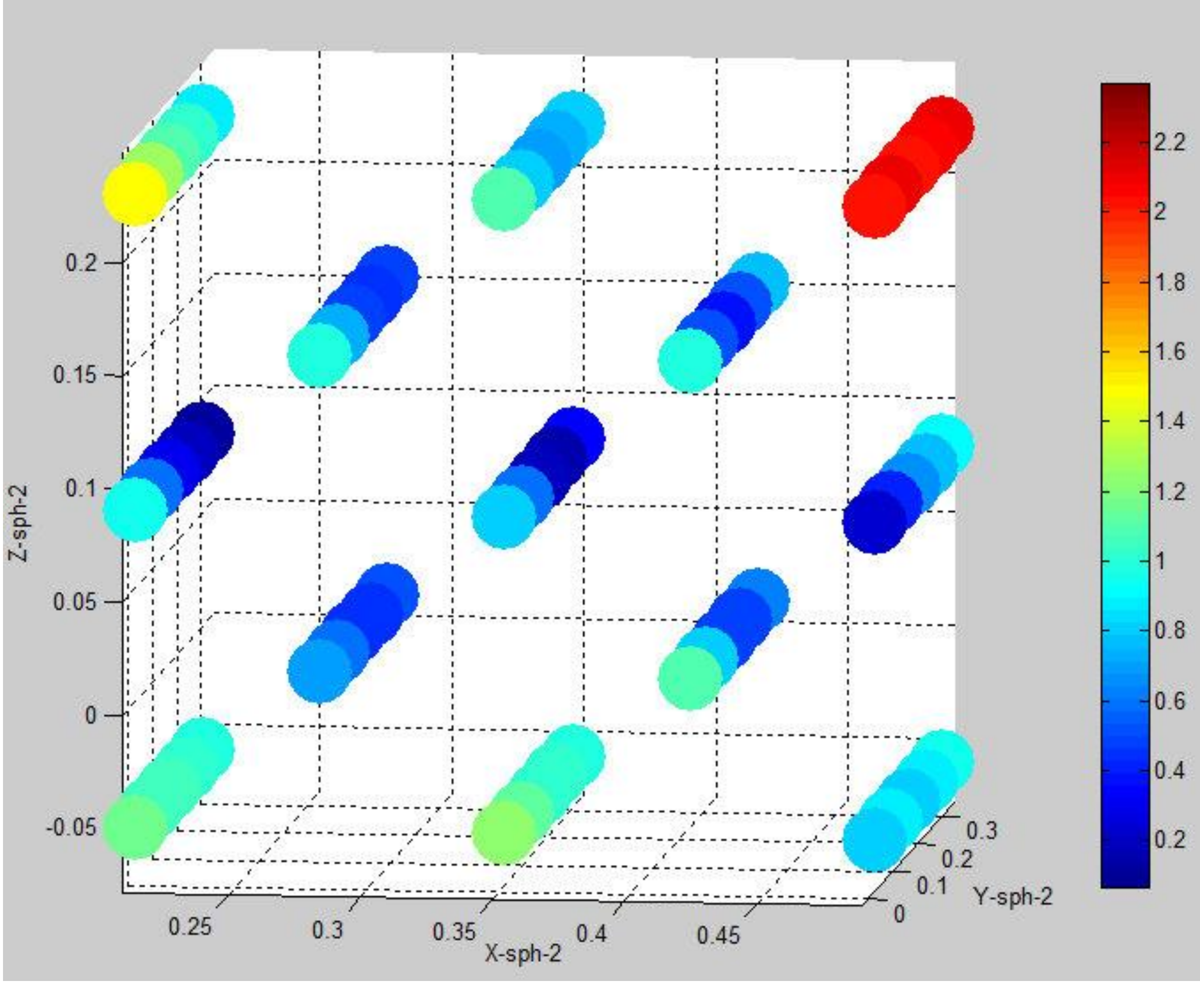


Figure.3.12. Discrepancy for sphere two.

Figure above shows the discrepancies of sphere two , and as we can see that position number 1 , 2, 3 which stays lower part of working volume and the position number 11 , 12 , 13 which are on the upper part of working volume have higher discrepancy than the middle part of the working volume. The other interesting point is that uncertainties are higher on the upper and lower part respect to the sphere one. On the upper part we can see the red color which represents the high uncertainty there. Also sphere two has regime in the middle part of working volume, there is similar color tone distribution.

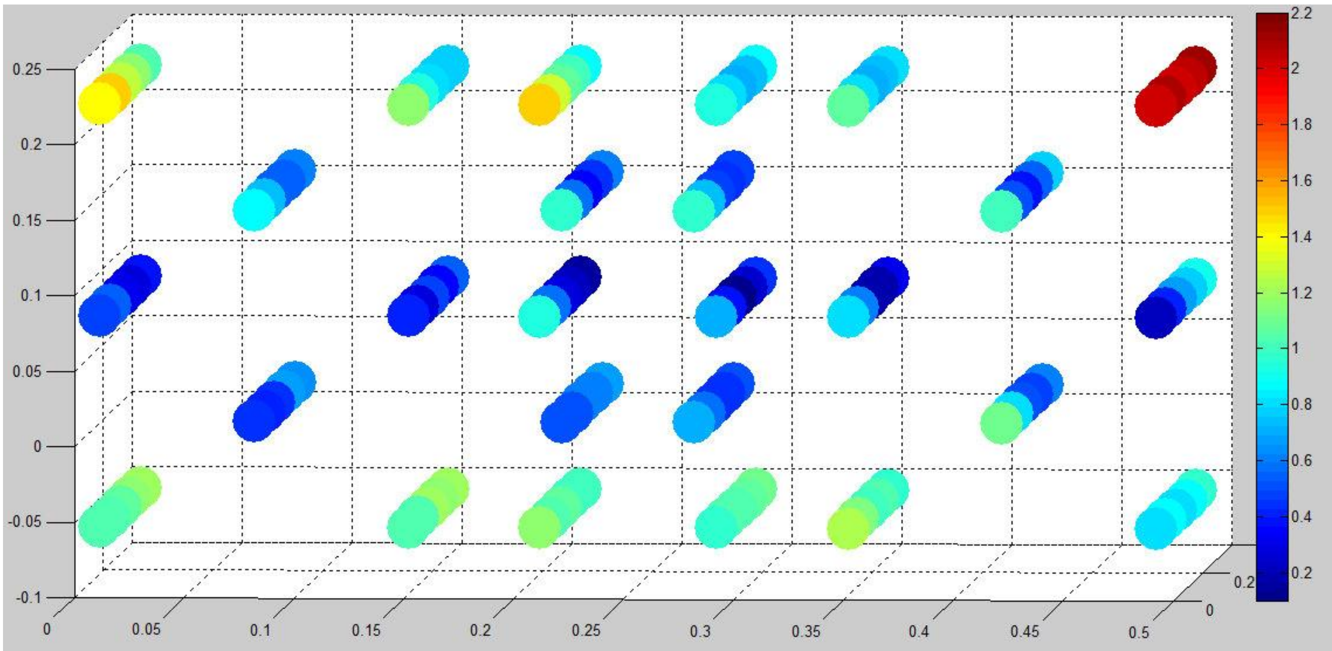


Figure.3.13. Discrepancy of two sphere together.

. On the figure above we can see the discrepancy distribution of two spheres together in the Cmm coordinate system. In that graph it is easier to see and understand the regime of the uncertainties in the working volume of both spheres. The general color tone distribution allows saying lower and upper part uncertainties are higher than middle part. On the position number 11 and 13 there are the highest discrepancies; especially on the right upper point which is number 13 has the highest discrepancy.

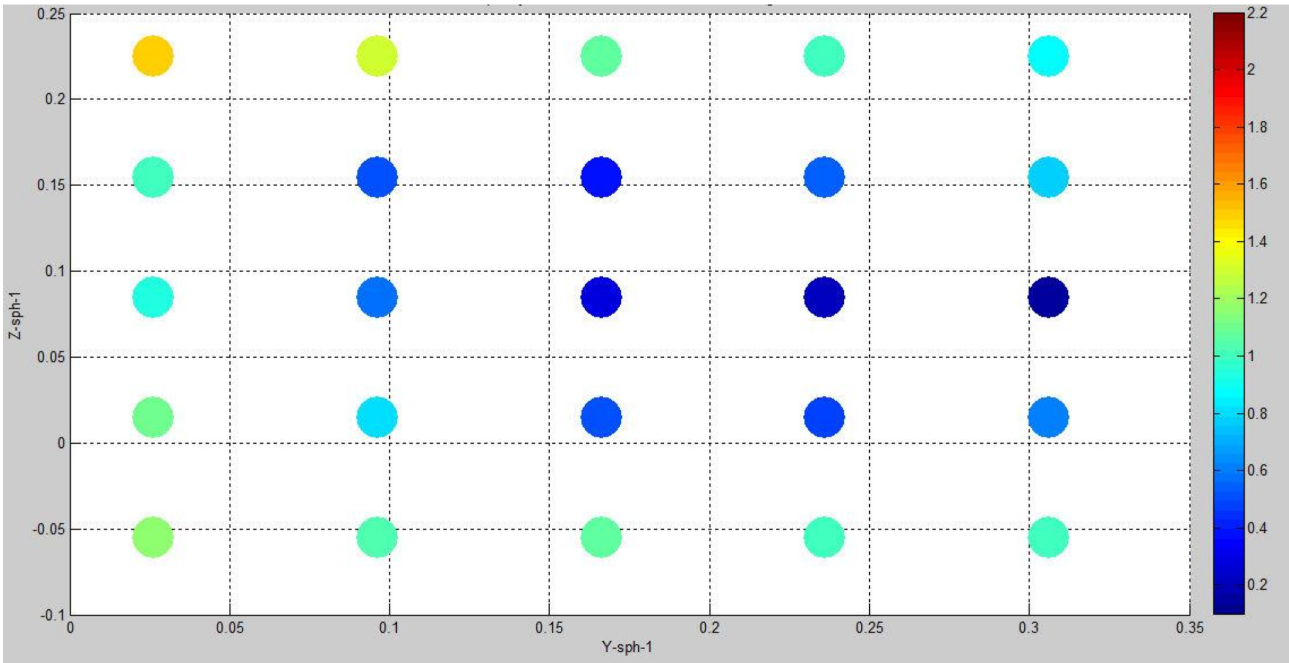


Figure.3.14. Discrepancy of two spheres on the zy plane.

Graph above also is helpful to show 5 planes along the Y coordinate direction and discrepancies have closer values in the middle part. Color tone distribution is easily visible in this perspective of graph.

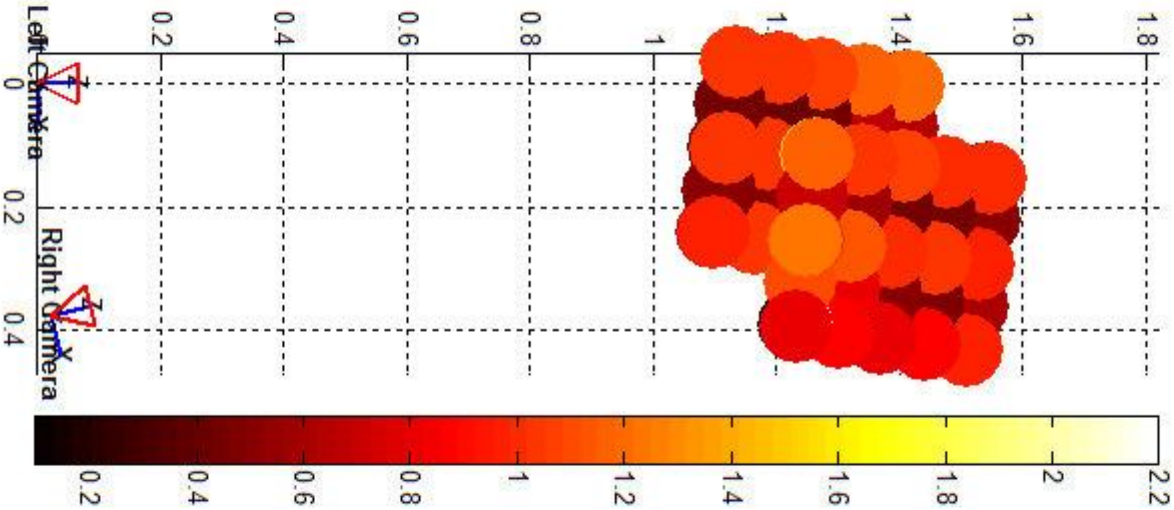


Figure.3.15. The discrepancy of spheres in camera coordinate system

The graph above shows the spheres discrepancy information in the camera coordinate system. As we can see the spheres position respect cameras.

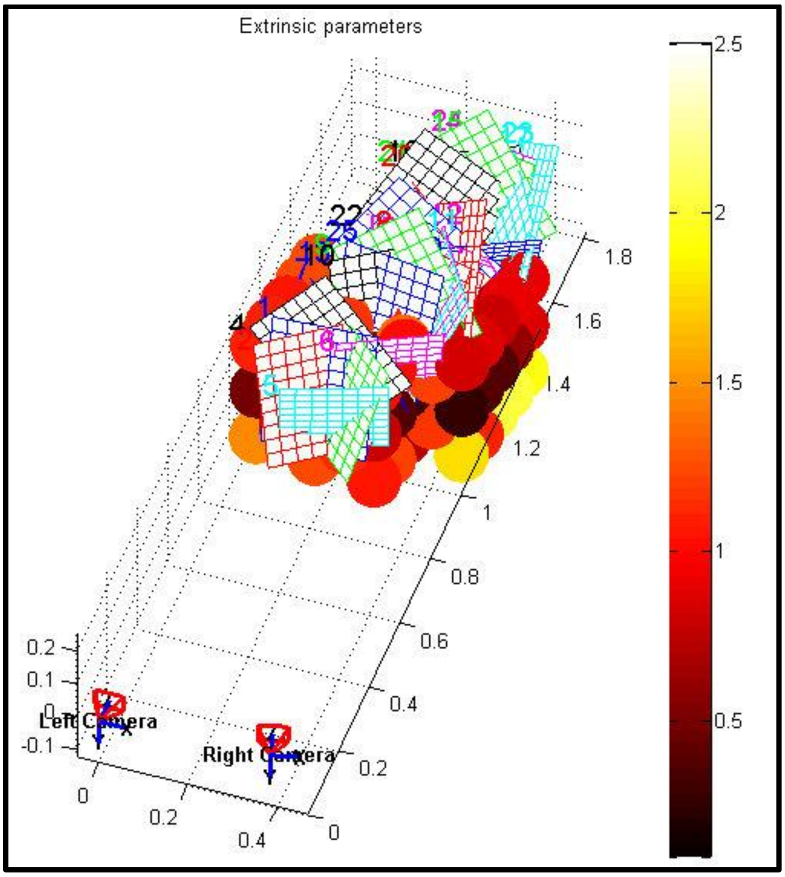


Figure.3.16. Position of spheres in the camera calibration area.

The figure above shows the where exactly place of working volumes in the calibration area of the two cameras in camera coordinates system. And it gives useful point of view about discrepancy distribution. The points are placed out of the calibration area has higher uncertainty. The middle part of the discrepancy graph which we mentioned before is placed more or less in the middle of calibration area where discrepancies are lower.

4. Monte Carlo Simulation

In this section the main idea is adding noise to the parameters such as; u v pixels coordinates, focal length, principal point (cc) and understand their effect on the discrepancies. Monte Carlo method is used for noise adding step.

4.1. Monte Carlo Method

The Monte Carlo method is an application of the laws of probability and statistics to the natural sciences. The essence of the method is to use various distributions of random numbers, each distribution reflecting a particular process in a sequence of processes such as the diffusion of neutrons in various materials, to calculate samples that approximate the real diffusion history. Statistical sampling had been known for some time, but without computers the process of making the calculations was so laborious that the method was seldom used unless the need was compelling. The computer made the approach extremely useful for many physics problems.[13]

The Monte Carlo method is very useful tool for measurement, especially when there are large amount of measurement which are result of repeated measuring operation. For example with this method you can add random noise to the inputs in order to simulate high variety of scenario.

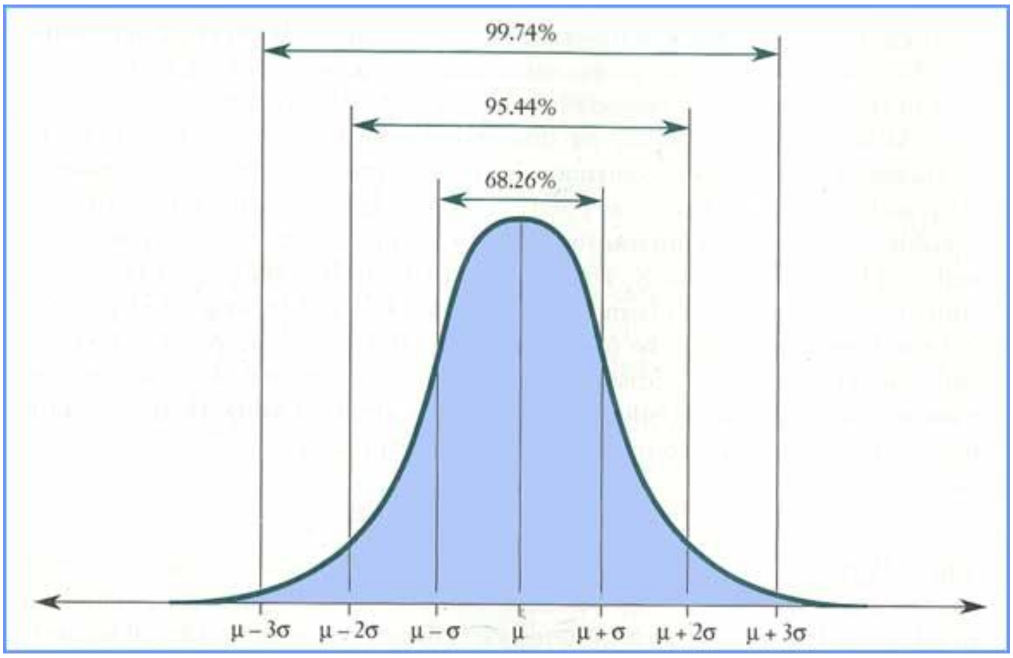


Figure.4.1. Gaussian distribution

The figure above shows the areas of the Gaussian distribution. The 68.26 % of the random numbers are generated ± σ one standard deviation from the mean value, 95.44% of the random numbers are generated ± 2σ two standard deviation from the mean value, 99.74% of the random number are generated ± 3σ three standard deviation from the mean value.

The parameter μ is the mean (location of the peak) and σ^2 is the variance. σ is defined as the standard deviation. The distribution with $\mu = 0$ and $\sigma^2 = 1$ is called the standard normal distribution.

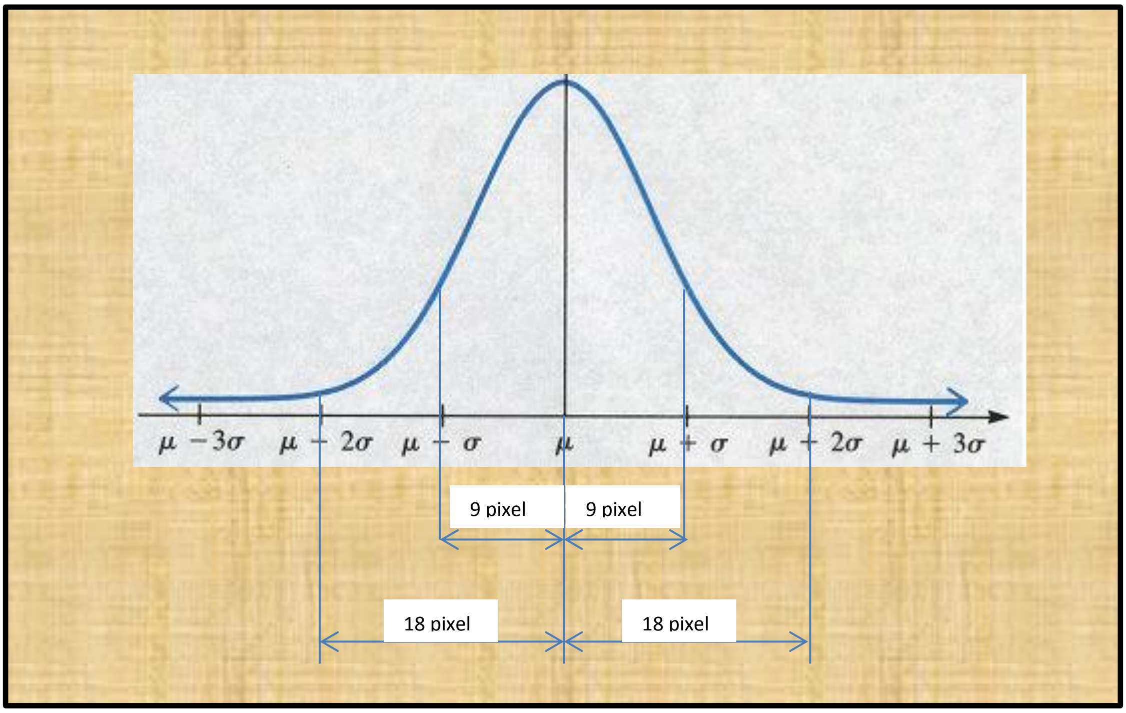


Figure.4.2. Example of noise adding by means of Gaussian distribution

Figure above valid for the case where noise added to principal point (cc) ; with 0.005 noise level Monte Carlo simulation creates random noise 68.26 % of them has the deviation ±9 pixel on the other side it is possible that by changing noise level we can adjust the deviation amount.

4.2. Point Projection and Noised Pixel Coordinates

In this part we project the 3D coordinates which are results of triangulation to 2D pixel coordinates. The matlab tool-box that we used for previous steps has a very useful command calls “project_points2”, this command allows project the 3D coordinates to 2D pixel coordinates we can say that it operates opposite process of triangulation.

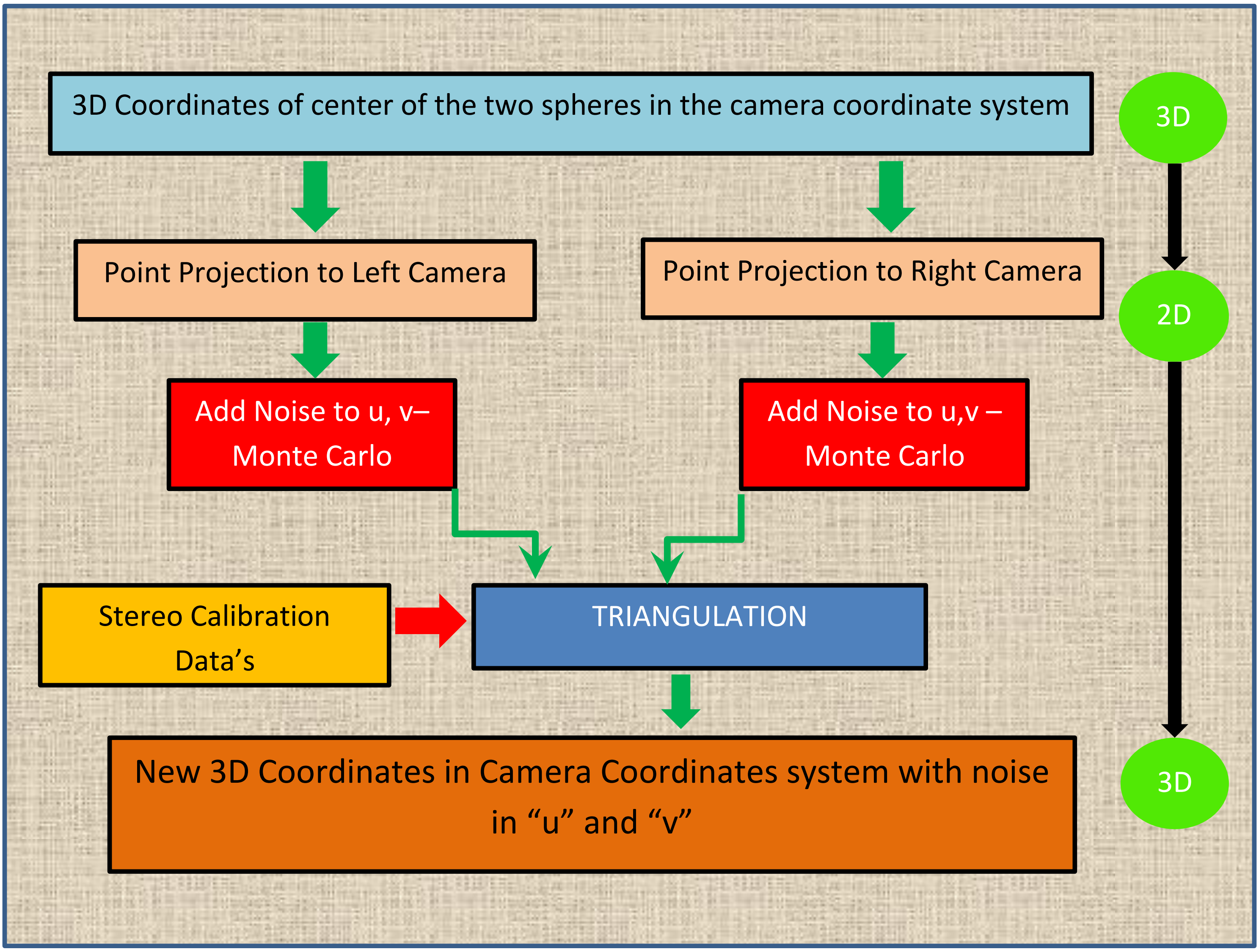

3D Coordinates of center of the two spheres in the camera coordinate system
3D
Point Projection to Left Camera
Point Projection to Right Camera
2D
Add Noise to u, v– Monte Carlo
Add Noise to u,v – Monte Carlo
Stereo Calibration Data’s
TRIANGULATION
New 3D Coordinates in Camera Coordinates system with noise in “u” and “v”
3D

Figure.4.3. Point projection steps

On the figure.4.3 as we can see that the process starts with 3D coordinates which are the results of triangulation. Then by using “project_points2” command we project 3D coordinates to 2D pixel coordinates.

Next step is vital point of the process which is adding noise by using Monte Carlo method; in this step we add random noise by means of Gaussian Distribution. But we defined noise coefficient in order to narrow down the limits of noise. After adding noise to the pixel coordinates we again triangulate them in order to obtain 3D coordinates.

By changing the noise level we try to understand the effect of parameter on the discrepancy.

Number of tests	**Noise Level**		
	Case 1	**Case 2**	**Case 3**
	0.5px	0.75px	1px
1000	Min. Std = 0.6 mm Max. Std = 1.2 mm	Min. Std = 0.9 mm Max. Std =1.8 mm	Min. Std = 1.2 mm Max. Std =2.4 mm

Table.4.1. Cases of noise levels on the u and v (noise level corresponds to the standard deviation of the Gaussian distribution which used to generate the noise)

The table above explains the minimum and maximum standard deviation of 3D coordinates for three different noise level cases. As we can see that even if we increase the noise level, deviation does not change too much.

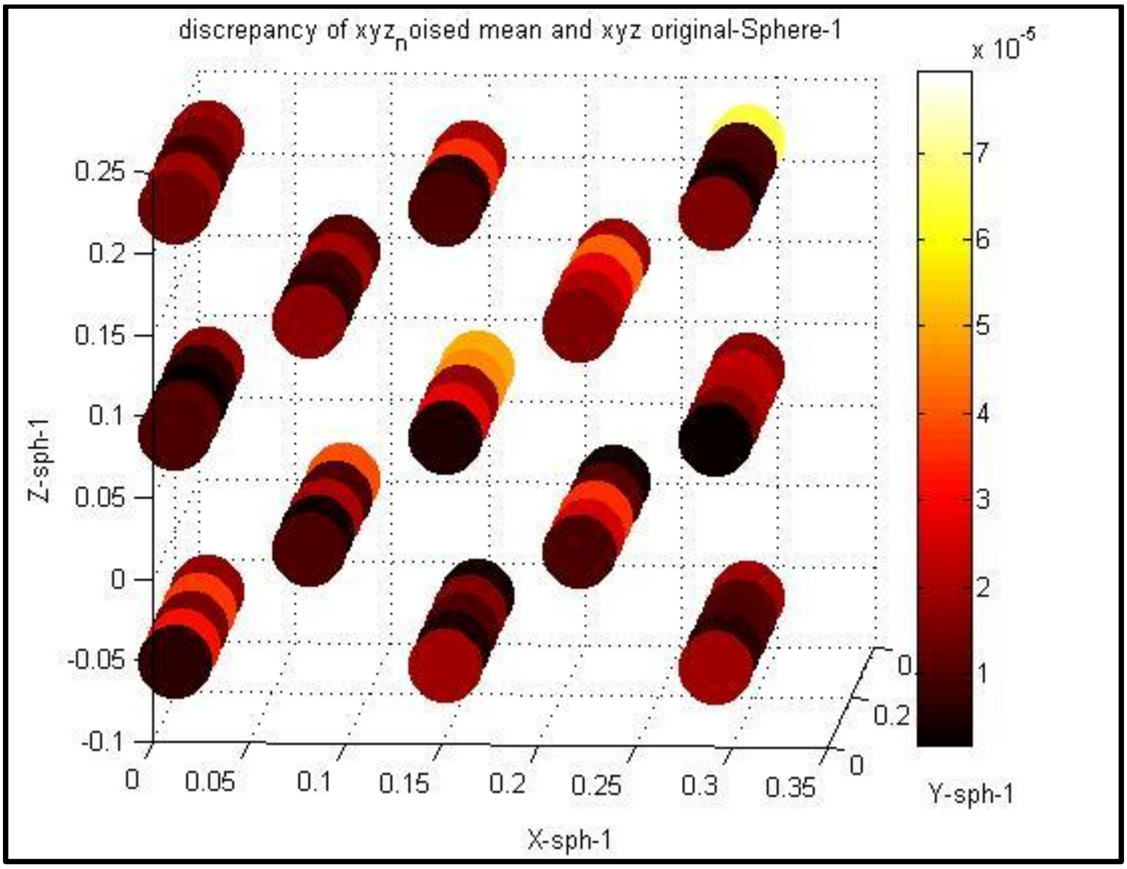


Figure.4.4. Discrepancy between 3D (with 0.5 pixel) mean of 1000 repeat and 3D without noise for sphere 1

After 1000 times repetition mean value of 3D coordinates is calculated then we compared them with 3D coordinates without noise. The graph above shows the discrepancy between them for sphere one and the discrepancies are quiet small. The discrepancy tends to be zero .This situation valid both sphere one and sphere two.

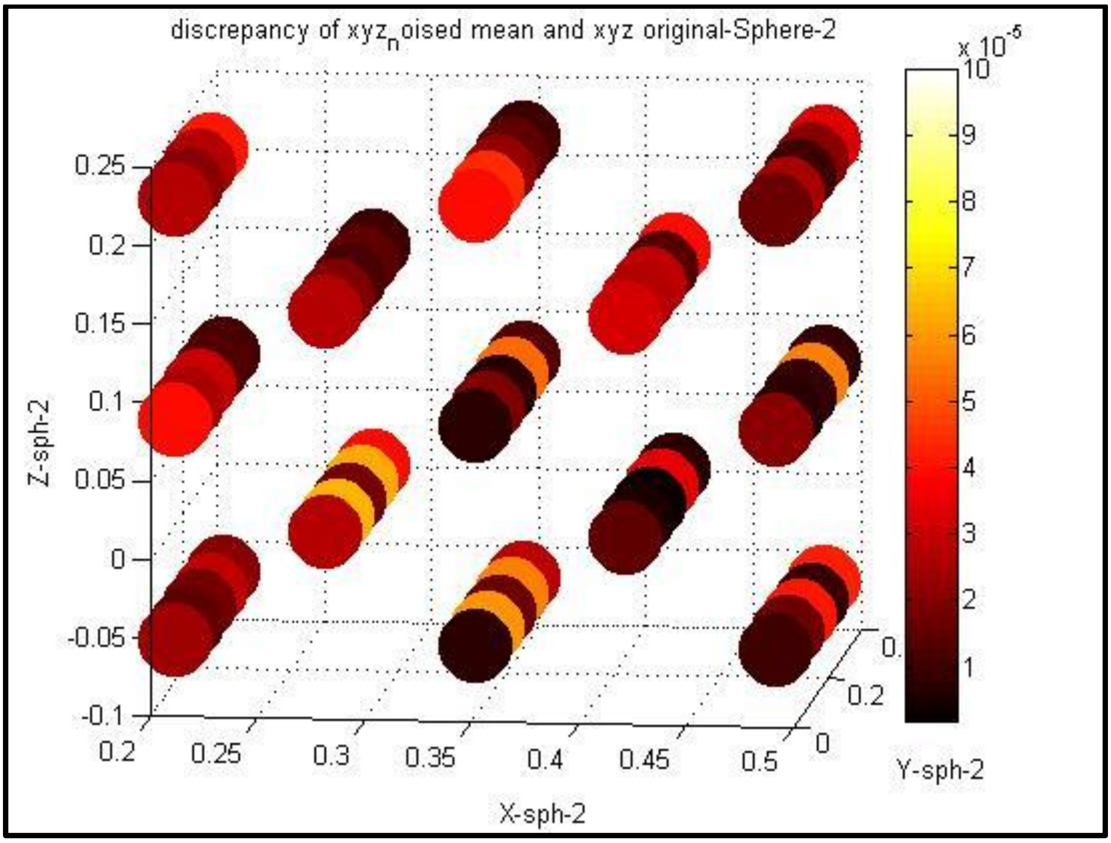


Figure.4.5. Discrepancy between 3D (with 0.5 pixel) mean of 1000 repeat and 3D without noise for sphere 2

Also for sphere two discrepancies are lower just a little bit higher respect to the sphere one. And also there is no certain regime for this discrepancy both for sphere one and sphere two. Also for sphere two it tends to be zero.

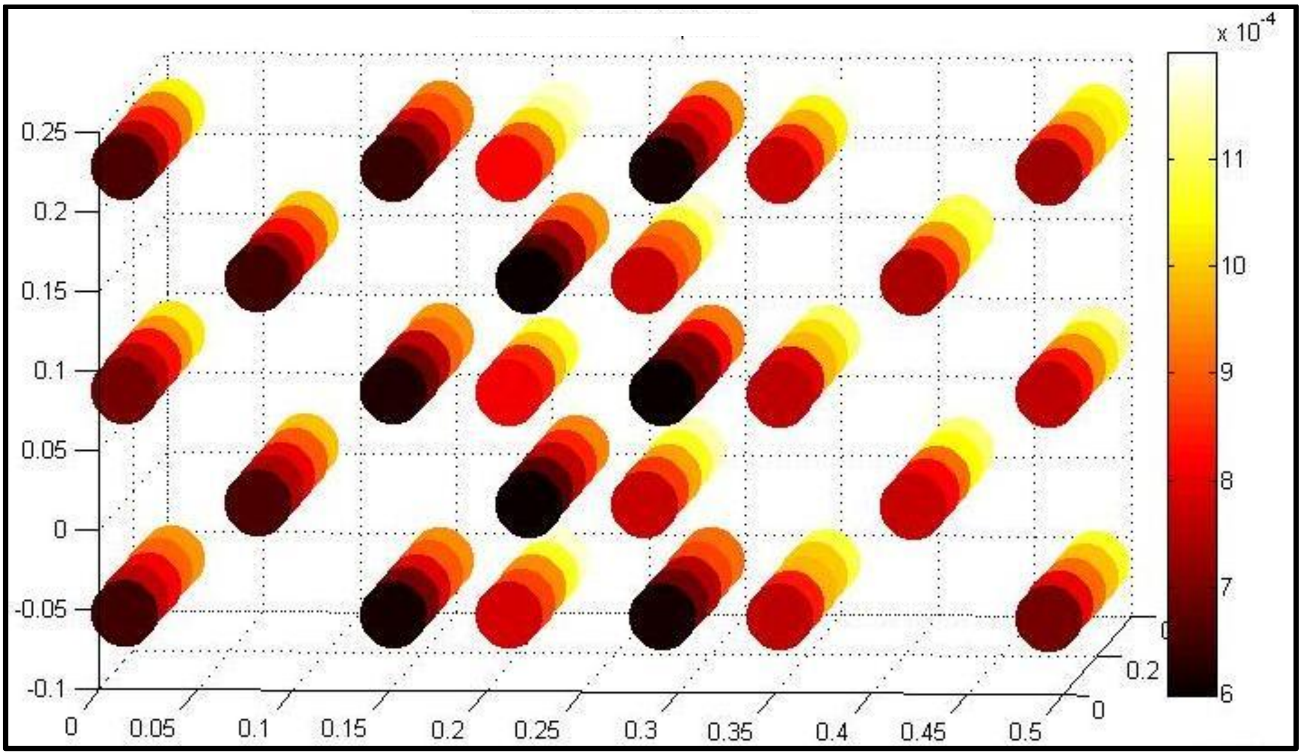


Figure.4.6. Standard deviations of 3D coordinate of sphere 1 and 2 for case one (with noise level 0.5 pixel)

The graphs above and below shows standard deviation result from noised repetitions and there is a regime from first plane to fifth plane, it is increasing from first plane to the fifth plane. But deviations are small numbers.

And also from first sphere to second sphere in the direction of x coordinates deviations are higher but still number are too small.

4.3. Focal Length and Noise Addition

As it is mentioned before triangulation uses data's which are coming from the stereo calibration, in this section we choose focal length parameter in order to understand its effect on the discrepancies or uncertainties. Focal length is an important parameter in the triangulation process and it is property of cameras.

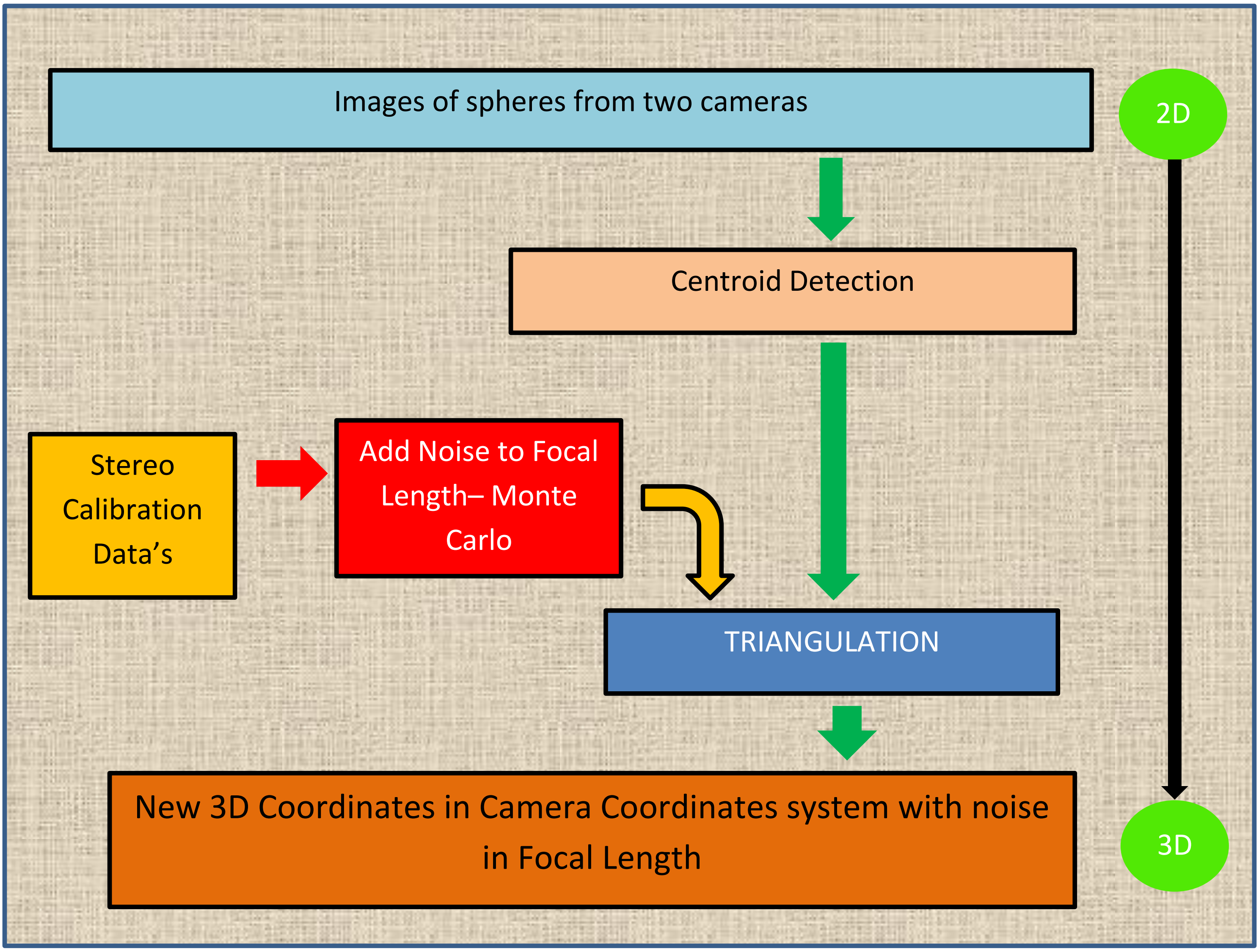

Images of spheres from two cameras
2D
Centroid Detection
Stereo Calibration Data's
Add Noise to Focal Length– Monte Carlo
TRIANGULATION
New 3D Coordinates in Camera Coordinates system with noise in Focal Length
3D

Figure.4.7. Adding noise to focal length

Figure 4.7 shows the noise adding to the focal lengths of two cameras, there two cameras and we have two focal length data’s for each of them in the stereo calibration data’s. As it is mentioned focal length has two dimensions.

$$\text{Noise Index } = \frac{\text{Mean of Left Camera Focal Length} + \text{Mean of Right Camera Focal Length}}{2}\ x\ coefficient$$

Figure.4.8. Noise equation for focal length

By using equation above we defined the noise index which is used to multiply with the random noises or number resulting from the Monte Carlo simulation and the result of this multiplication added to focal length of both right and left cameras. And by changing coefficient it is possible to adjust the boundary of the random noises.

Number of tests	Noise Level		
	Case 1	Case 2	Case 3
	0.1%	0.5%	1%
1000	Min. Std = 0.2 mm Max. Std = 2.5 mm	Min. Std = 1 mm Max. Std =11 mm	Min. Std = 2 mm Max. Std =24 mm

Table.4.2. Cases of noise levels on the focal length (noise level is the percentage of nominal focal length value and corresponds to the standard deviation of the Gaussian distribution which used to generate the noise)

The table.4.2 shows the three cases with different noise level and we can see the minimum and maximum standard deviations for three cases. It is obvious that while noise level increasing the deviations are increasing as well. The deviations are increasing linearly respect to the noise level.

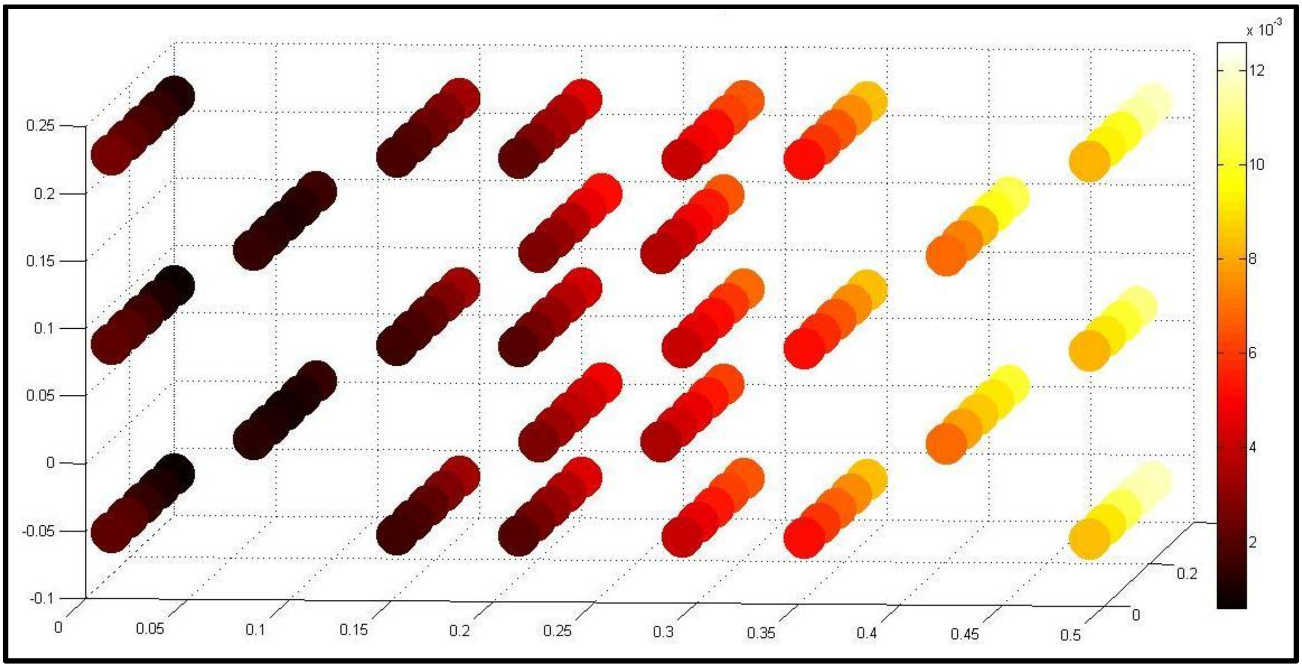


Figure.4.9. Standard deviations of spheres in the case 2 (standard deviation results from noise corresponds to the 0.5% of the nominal focal length value)

The figure above shows the standard deviations for the case 2 for sphere one and sphere two. There is a regime from first sphere to second sphere deviations are increasing. And also from first plane to the fifth plane deviations are increasing.

The discrepancies between mean of 3D coordinates (with noise in focal length) and 3D coordinates without noise is too small , it tends to be zero for huge number of repetition.

4.4. Principal Point and Noise Addition

The other important parameter is principal point which we already defined in theory part. In this section noise will be added to the principal point and principal point has two components (u, v) which are noised separately.

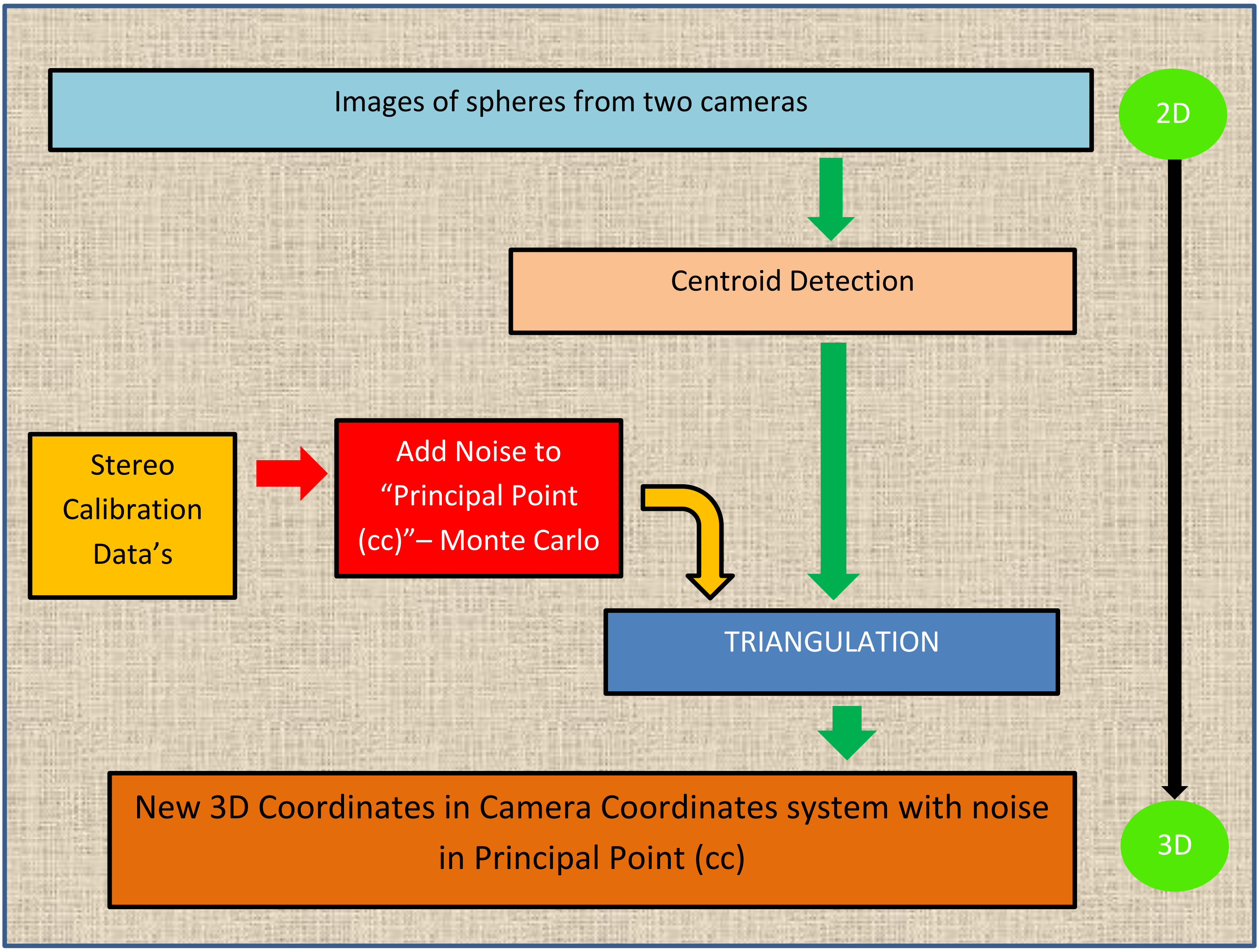

Images of spheres from two cameras
2D
Centroid Detection
Stereo Calibration Data’s
Add Noise to “Principal Point (cc)”– Monte Carlo
TRIANGULATION
New 3D Coordinates in Camera Coordinates system with noise in Principal Point (cc)
3D

Figure.4.10. Adding noise to principal point (cc) length

The principal point is coming from the stereo calibration data’s, as we explained before passing from 2D to 3D same process performed. But this time noise is added to the principal point (while other parameters without noise) in order to understand its effect on the discrepancies.

$$\text{Noise Index of horizontal coordinate} = \text{Principal point position along the horizontal direction } x\ coefficient$$

$$\text{Noise Index of vertical coordinate} = \text{Principal point position along the vertical direction } x\ coefficient$$

Figure.4.11. Noise equation for principal point

Number of tests	**Noise Level**		
	Case 1	**Case 2**	**Case 3**
	0.05%	0.1%	0.5%
1000	Min. Std = 1.1 mm Max. Std = 2.3 mm	Min. Std = 2.2 mm Max. Std =4.6 mm	Min. Std = 11.1 mm Max. Std =23.4 mm

Table.4.3. Cases of noise levels on the principal point position (noise level is the percentage of nominal principal point position and corresponds to the standard deviation of the Gaussian distribution which used to generate the noise)

Table.4.3 shows the three noised cases of principal point , as we can see that noise is added to principal point effected the deviations and deviations are big numbers for the case one and two . The process from 2d to 3d is get affected a lot by principal point.

For example for the case 1 the noise is 0.05 percent of the nominal value of the principal point position and we have min. deviation 1.1 millimeter and max deviation is 2.3 millimeter. For case 2, noise is 0.1 percent of the nominal value and minimum deviation is 2.2 mm while maximum deviation is 4.6 mm. Also for the case 3 the noise is 0.5 percent of the nominal value but deviations are unacceptably big numbers. Due to having this comparison of these 3 cases, it is obvious that principal point has non-negligible effect on this process.

4.5. Comparison of Parameters

First of all comparisons among the parameters is valid only for this certain setup (certain baseline) which is done in the laboratory of the Politecnico di Milano but the numerical simulation (which used to analyze the parameters) is applicable for any other stereoscopic system with different baseline.

Previously we run Monte Carlo simulation for focal length, principal point and u v center pixel coordinates. They showed different reaction to noise addition and each parameter had affected the process (2d to 3d) in the different way. Finally we can compare them in order the have idea about their effectiveness on the process respect each other.

Parameters	Noise													
	0.05%		0.1%		0.5%		1%		0.5px		0.75px		1.0px	
	Min.	Max	Min.	Max	Min.	Max	Min.	Max	Min.	Max	Min.	Max	Min.	Max
U,V	–	–	–	–	–	–	–	–	0.6	1.2	0.9	1.8	1.2	2.4
FC	–	–	0.2	2.5	1	11	2	24	–	–	–	–	–	–
CC	1.1	2.3	2.2	4.6	11.1	23.4	–	–	–	–	–	–	–	–

Table.4.4. The results for all parameters

Table.4.4 shows the deviations for all parameter and cases, the minimum and maximum deviations are in millimeter unit. The noise ranges; 0.05 % - 0.5% of nominal value for principal point, 0.1 % - 1% of nominal value for focal length. The u v coordinates has no nominal value because it is different value for all different position of the spheres that’s why noise defined as a pixel amount.

When there is 0.1% noise in focal length, the maximum deviation is 2.5 mm; for the same percentage of the noise in the principal point the maximum deviation is 4.6 mm. The principal point has more affect than focal length in that certain setup for the same percentage of noise.

The u v parameters effects also the deviation for the case 1 where the noise is 0.5 pixel minimum deviation is 0.6 mm , maximum deviation is 1.2 . While the noise level increasing the minimum and maximum deviations are increasing linearly.

5. CONCLISIONS

After performing the experiment and analyzing the data's that we have as a result of experiment, there are important points to highlight both related to procedure and analysis of data's. First, during the experiment it is mentioned that lighting system is really important in order to get reliable data's to analyses. Other than the working volume has to be placed in the calibration volume in order to be able to understand regimes during analyzing step.

The simulations play important role in analyzing step where focal length, principal point and u v center pixel coordinates are investigated. Comparison between parameters is performed by the help of Monte Carlo method. The idea is to obtain order of importance among the parameters, importance in the sense of effectiveness on the discrepancies. At the end of the analyzing process, the principal point has much more effect than focal length. The u v center pixel coordinates parameter has the linear effect respect to the noise level on the discrepancies. The numerical simulation (which used to analyze the parameters) is applicable for any other stereoscopic system with different baseline.

For future works, the next beneficial step would be investigating other parameters with different baselines in order to understand uncertainty in stereoscopic measurement systems.

References

1. Homepages.inf.ed.ac.uk (n.d.) Elements of Geometric Computer Vision - Pin-hole Camera Geometry. [online] Available at: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/FUSIELLO4/tutorial.html#x1-30003 [Accessed: 15 Nov 2012].

2. Fleet , D. and Hertzmann, A. (2006) Computer Graphics Lecture Notes. [e-book] Toronto: p.38. http://www.dgp.toronto.edu/~hertzman/418notes.pdf [Accessed: 15.11.2012].

3. HARTLEY, R. and Zisserman, A. (2003) Multiple view geometry in computer vision. 2nd ed. Cambridge: Cambridge University Press, p.154-157.

4. Zhang, Z. (n.d.) CAMERA CALIBRATION. [e-book] p.12. Available through: cronos.rutgers.edu http://cronos.rutgers.edu/~meer/TEACHTOO/PAPERS/zhang.pdf [Accessed: 21.11.2012].

5. Heikkilä, J. and Silvén, O., (1997), "A Four-step Camera Calibration Procedure with Implicit Image Correction", Computer Vision and Pattern Recognition, Proceedings of IEEE Computer Society Conference, Washington, DC, June, 17-19 , 1997, pp. 1107.

6. Vision.caltech.edu (2010) Camera Calibration Toolbox for Matlab - Description of the calibration parameters. [online] Available at: http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/parameters.html [Accessed: 20 Nov 2012].

7. Elgammal, A. (n.d.) CS 534: Computer Vision Camera Calibration. [e-book] New Jersey: Dept of Computer Science, Rutgers University. p.23. Available through: www.rutgers.edu http://www.cs.rutgers.edu/~elgammal/classes/cs534/old_lectures/Lecture4_5_6_CameraGeometry_Calibration.pdf [Accessed: 21.11.2012].

8.Modeling the Pinhole Camera Department of Mathematics, University of Central Florida, (2011) Modeling the Pinhole Camera. [online] Available at: http://www.math.ucf.edu/~xli/Pinhole%20Camera2011.pdf [Accessed: 18 Nov 2012].

9. Fraser, C. and Remondino, F., (2006), "DIGITAL CAMERA CALIBRATION METHODS: CONSIDERATIONS AND COMPARISONS", 'Image Engineering and Vision Metrology', Proceedings of the ISPRS Commission V Symposium, Dresden, September, 25-27, 2006, pp. 267.

10. Vision.caltech.edu (2010) Camera Calibration Toolbox for Matlab - First calibration example - Corner extraction, calibration, additional tools. [online] Available at: http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/example.html [Accessed: 20 Nov 2012].

11. Vision.caltech.edu (2010) Camera Calibration Toolbox for Matlab - Fifth calibration example - Calibrating a stereo system, stereo image rectification and 3D stereo triangulation. [online] Available at: http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/example5.html [Accessed: 28 Nov 2012].

12. Horaud , R. et al. (2012) Stereo Calibration from Rigid Motions. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 22 (12), p.1446. Available at: http://hal.archives-ouvertes.fr/docs/00/59/01/27/PDF/HoraudCsurkaDemirdjian-pami2000.pdf [Accessed: 28.11.2012].

13. Anderson, H. (1986) Metropolis, Monte Carlo and the MANIAC. LOS ALAMOS SCIENCE, Iss. Fall p.96.