Analysis of Camera Pose Estimation Using 2D Scene Features for Augmented Reality Applications

Mémoire

Shabnam Meshkat Alsadat

Maîtrise en génie électrique
Maître ès sciences (M.Sc.)

Québec, Canada

Analysis of Camera Pose Estimation Using 2D Scene Features for Augmented Reality Applications

Mémoire

Shabnam Meshkat Alsadat

Sous la direction de :

Denis Laurendeau, directeur de recherche

Résumé

La réalité augmentée (RA) a récemment eu un impact énorme sur les ingénieurs civils et les travailleurs de l'industrie de la construction, ainsi que sur leur interaction avec les plans architecturaux. La RA introduit une superposition du modèle 3D d'un bâtiment sur une image 2D non seulement comme une image globale, mais aussi potentiellement comme une représentation complexe de ce qui va être construit et qui peut être visualisée par l'utilisateur. Pour insérer un modèle 3D, la caméra doit être localisée par rapport à son environnement. La localisation de la caméra consiste à trouver les paramètres extérieurs de la caméra (i.e. sa position et son orientation) par rapport à la scène observée et ses caractéristiques.

Dans ce mémoire, des méthodes d'estimation de la pose de la caméra (position et orientation) par rapport à la scène utilisant des correspondances cercle-ellipse et lignes droites-lignes droites sont explorées. Les cercles et les lignes sont deux des caractéristiques géométriques qui sont principalement présentes dans les structures et les bâtiments. En fonction de la relation entre les caractéristiques 3D et leurs images 2D correspondantes détectées dans l'image, la position et l'orientation de la caméra sont estimées.

Abstract

Augmented reality (AR) had recently made a huge impact on field engineers and workers in construction industry, as well as the way they interact with architectural plans. AR brings in a superimposition of the 3D model of a building onto the 2D image not only as the big picture, but also as an intricate representation of what is going to be built. In order to insert a 3D model, the camera has to be localized regarding its surroundings. Camera localization consists of finding the exterior parameters (i.e. its position and orientation) of the camera with respect to the viewed scene and its characteristics.

In this thesis, camera pose estimation methods using circle-ellipse and straight line correspondences has been investigated. Circles and lines are two of the geometrical features that are mostly present in structures and buildings. Based on the relationship between the 3D features and their corresponding 2D data detected in the image, the position and orientation of the camera is estimated.

Table of contents

List of Figures

List of Tables

Acknowledgments

I would like to express my sincere gratitude to

- Prof. Denis Laurendeau for his encouragement and guidance during the recent years and for giving me the opportunity to learn and explore.
- Maman and Baba for their unconditional love and support.
- Shadi for always being there for me.

Introduction

Augmented reality (AR) has been a breakthrough technology in the construction industry in recent years. It allows civil engineers and architects to visualise a representation of the structure that is going to be built at different stages of the construction by simply using a tablet or a hand-held device. Moreover, AR enables engineers and construction workers to evaluate how an architectural model will fit and look like in the construction site by inserting a 3D model onto the viewed scene as well. In fact, AR finds the viewpoint of the user and provides a depiction of the 3D model from the same viewing angle.

Indeed, the main task in AR is to localize the camera with respect to the scene that is being viewed. Positioning a camera requires tracking the features in the captured images. Based on the position of the features in the real environment and in the image, the exterior pose of the camera can be obtained and augmentation can be achieved.

The goal of this project is to localize the camera as well as its position and orientation by considering geometrical features that are present in the acquired images. Since points have been thoroughly investigated in other researches, we decided to proceed with straight lines and circles as more sophisticated geometrical primitives that are mostly visible in buildings and architectural environments. Having these features detected in 2D images, we estimate the camera pose by establishing the relationship between them and their corresponding 3D data.

Chapter 1 includes a literature review of previous studies on pose estimation from straight lines and circles, as well as a brief review on the methods using points. The camera pose estimation methods that are used in this project are presented in chapter 2. Line and ellipse detection methods adopted in feature detection step are presented in chapter 3. In chapter 4, we present and discuss the experimental results. Lastly, we sum up the thesis with the conclusion.

Chapter 1

1. Literature Review

In this chapter, a brief review on pose estimation approaches using point features is presented. Furthermore, we continue by providing a review on previous works on pose estimation method using circles and straight lines.

1.1 Pose Estimation from Point Correspondences

The problem of finding the pose of a camera with point features is a well-studied topic in computer vision. The general case, referred to as Perspective-n-Point problem (PnP), investigates the solution to pose estimation using n point correspondences between the world and the camera frames. However, some researches are dedicated to solve the pose problem for a specific number of points. For instance, [1] and [2] provide solutions for P3P problem. Some approaches to solve for P4P are presented by [3], [4] and [5]. In [6], the authors deal with P5P case with a closed-form numerical method.

Among all methods aiming at achieving real-time performance, those trying to eliminate the initialisation step and those that present a non-iterative method have been of more interest. Authors in [7] propose a method to estimate the pose directly by solving a nonlinear least-squares system of equations with the points expressed in spherical coordinates. Following [7], [8] attempts to reduce the computational complexity of the approach by presenting OPnP and [9] improves its accuracy with a method called UPnP. The closed-form solution of [10], also known as EPnP, forms a system of equations that are formulated as a weighted sum of four virtual control points. Although their method is less accurate than the iterative approach of [11], it is faster due to its complexity that is linear with respect to the number of points [10].

The fact that the pose of a camera is a matter of projective transformation, leads to a simple yet useful geometric constraint known as collinearity equation. An object point and its image on the image plane lie on a straight line passing through the optical center. Based on the distances between the points in object space and their projections on image plane, the method presented in [12] builds a set of quadratic equations. Having the depth scalings recovered, it becomes an absolute orientation problem which seeks rotation and then translation from the relationships between the centroids of the points in both frames [13]. Considering the collinearity of the point sets, the approach in [14] solves for pose by minimising a linear least-squares system consisting of the distances between back-projected and scaled points in object space. Geometrical constraints, particularly the collinearity constraint, are the core idea of many approaches such as [11], [15], [16] and [17].

1.2 Pose Estimation from Circle-Ellipse Correspondences

In a general configuration, the shape of a circle is not preserved under perspective projection. The only situation in which the image of such features remains circular is when the object's surface is parallel to the image plane [18]. In most cases, the circle transforms into an ellipse. Several papers used circles and circular targets as an improvement to the camera calibration process. Mostly the studies with circles are on estimating intrinsic camera parameters, e.g. [19], [20], [21] and [22]. Indeed, camera pose estimation using circular features has not been targeted in recent works. There are a few researches dedicated to extrinsic calibration with circle-ellipse correspondences that will be discussed in the following.

As a preliminary step in pose estimation from circle-ellipse correspondences, most approaches recover the normal vector and the center of detected ellipses on the image plane. The method proposed in [23] considers a single view of two coplanar circles with unknown radii. Firstly, they proceed with finding the center and normal of each ellipse on the image plane by defining an oblique elliptical cone. This cone is formed assuming straight lines starting from the camera frame origin to the circumference of the ellipse in the image, as shown in Figure 1. Rotating the base plane of the oblique elliptical cone, in a way that its normal vector has the same direction as the Z-axis of camera frame, the oblique elliptical cone transforms to an oblique circular cone. The normal vector to the ellipse is computed

based on this rotation. Since both circles should have equal normal vectors regarding their coplanarity. The focal length is recovered by minimizing the dot product of both normal vectors. To find the pose of the camera, they assume a reference frame with its origin placed at the center of the detected ellipse. Despite the description of the process given in the paper, the authors do not explain clearly how to deal with the ambiguity of solutions to the normal vector and the center of ellipses. Recovery of the camera pose is vaguely explained as well. In addition, [24] mentions that the method presented in [23] results in two possible solutions while they handle the ambiguity by using a specific design for the marker to distinguish the right resulting pose.

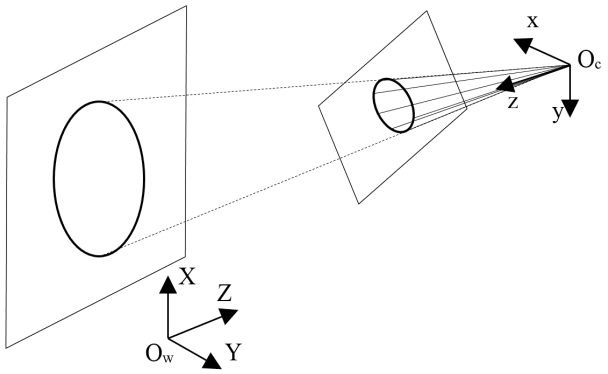


Figure 1 - Oblique elliptical cone defined by straight lines from the optical center to the ellipse on the image. Ow and Oc indicate the world and the camera reference frame respectively.

Another approach that makes use of two coplanar circles is the one from [25]. Their work is based on the calibration procedure explained in [26] using the image of absolute conic of two coplanar circles with known radii. They derive the rotation matrix from the constraints between the projected circle centers and the vanishing line of the plane that both object circles lie on, as in the pole-polar constraints explained in [26]. According to degenerate conic equations, they solve for translation in closed-form, which can be used as an initialization for other iterative methods as well as the previously mentioned method of [23]. Another research based on the concept of absolute dual conic and vanishing line relationships is proposed by [27], but it does not present well-elaborated results.

Equiform geometry also known as similarity geometry is based on the relationships of object features that remain unchanged between the object in the world and its image [28]. The methods in [29] and [30] provide more or less the same approach toward circular features considering two coplanar conics by employing equiform transformations[1]. They suggest dividing the problem into two parts by introducing an induced frame between world and camera frames. The induced frame is the transformed world frame with a rotation and translation in a way that its XY plane is placed parallel to the image plane and its origin placed at $(0,0,r)$. The first part of the problem concentrates on finding parameters of that rotation and translation by solving fourth-order polynomials iteratively. The second part includes an isotropic scaling that maps the object on the XY plane of induced frame to the image plane. The scaling parameter is computed in least-squares form. The authors also recommend applying an iterative method to improve the estimation.

Most researches are conducted using two or more circles in different configurations. However, there is no proven method to solve for pose from one circle in a single view without using other information from the scene [31]. In addition, most approaches deal with the problem from a geometrical point of view. Hence, it is difficult to evaluate systematically which method may outperform other methods based on objective criteria.

1.3 Pose Estimation from Line Correspondences

Exterior calibration from lines, frequently addressed as PnL (Perspective-n-Line), includes finding the position and orientation of a camera using straight lines in the image plane and in the object frame. Some of the methods in this field are inspired by pose estimation methods from points, PnP, such as [32]. While a large number of studies are based on the geometrical aspects of lines and their characteristics in perspective geometry, there is a key principle behind almost every solution to the pose estimation problem from line features. In perspective projection, for each line there exists a plane passing through the optical center, the line's image on the image plane and the 3D line in the object frame, see Figure 2. In other words, the line on the image plane and the object line are coplanar. Therefore, geometric constraints

[1] An affine transformation consisting of a rotation and a translation is called an equiform transformation. [57]

retrieved from this plane and the directions of the line in both 2D and 3D play an important role in solving for position and orientation of the camera.

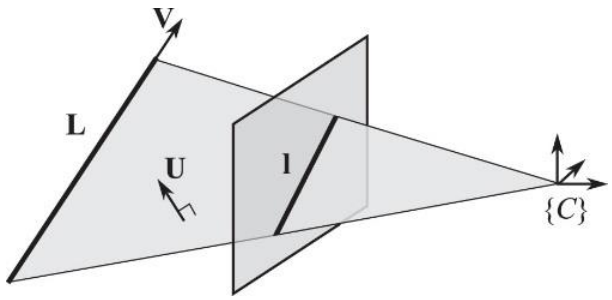


Figure 2 - The line in the object frame and its projection on the image plane lie on a plane passing through the optical center *[33]*.

The authors of [12] approach the pose estimation problem from the geometrical point of view. They construct a system of quadratic equations derived from equation (1) and equation (2) to be solved with SVD, see Figure 3. Equation (1) says that the direction of the line in object frame, w, equals the direction of the line in the camera frame, v, rotated by the rotation matrix R. In addition, the position of the point q in the world frame can be calculated by rotating and then translating its corresponding point p in the camera frame. They consider the orthogonality constraint [34] of the rotation matrix, Equation (3), in constructing the linear system of equations as well.

$$w = Rv \tag{1}$$

$$q = Rp + t \tag{2}$$

$$R^T R = RR^T = I \tag{3}$$

Recovering the rotation, they find the translation by applying SVD to another system of linear equations from the back-projected lines. A major improvement of this method is that it overcomes the ambiguity problem and yields a unique solution for extrinsic calibration.

An algebraic method that solves the pose problem in a least-squares sense is the one from [35]. The following condition must be fulfilled for each line:

$$n^T R\, l = 0 \tag{4}$$

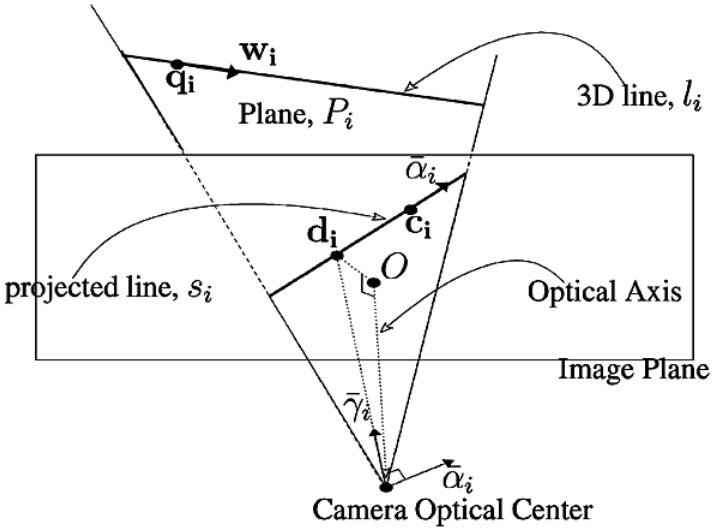


Figure 3 - The corresponding line on the image plane and in the world frame *[12]*.

where n is the normal vector of the plane in the camera frame and l is the line in world coordinates. In addition, R indicates the rotation of the camera in the world frame. They parameterize the rotation matrix with Cayley-Gibbs-Rodriguez representation [36] in order to reduce the computational cost. Thus, equation (4) is minimized along with the orthogonality constraints to find the rotation matrix. The translation vector is then computed based on the method provided in [37].

The approach proposed in [38] is a non-iterative method called Robust PnL (RPnL). It assumes the line with the longest projection on the image plane to be an intermediate coordinate system between the world and the camera frames, see Figure 4. All the lines are divided into sets of triplets to help with reducing the complexity of the algorithm. They minimize a sixteenth order polynomial based on equation (4), which was also used in [35], for each triplet of lines. In real experiments, they recommend to normalize the result of the minimization. In order to normalize the resulting rotation matrix, every object point is aligned with its back-projected correspondence. To overcome the ambiguity of possible solutions, those with large

orthogonality errors are eliminated in the first place. Among the remaining solutions, the one with the minimum reprojection error is chosen.

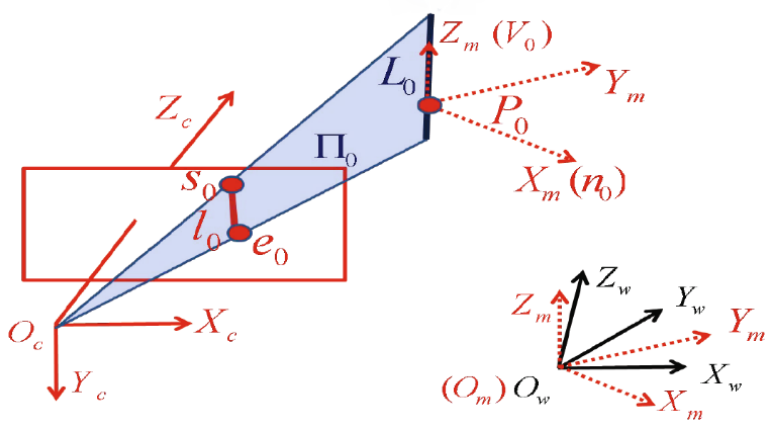


Figure 4 - The line with the longest projection (circumscribed in the red rectangle) is defined as the model co-ordinate frame *[38]*.

The results they provide in real experiments show more accuracy than the methods of [12] and [35]. Based on RPnL, the authors of Accurate Subset-based PnL (ASPnL) [39] go one-step further. In addition to choosing the line with the longest projection as the model coordinate frame, they define the auxiliary axis as the line with the second longest projection to make the complexity of the method linear with respect to the number of lines. This leads to minimizing eighth order polynomials instead of sixteenth order polynomials for each triplet of lines. The same process of eliminating incorrect solutions is done to find the right pose as well. Finally, the best solution is chosen based on minimum orthogonality error. Although ASPnL outperforms all the previously mentioned methods in situations with no or small amount of noise, it is quite sensitive to noise and outliers in larger sets of lines.

In addition, [40] provides an error function to be minimized for pose estimation consisting of two parts. A part based on point features and another for lines. The latter is solved as follows, which is similar to the approach proposed by [41]. The normal vector to the plane containing each line in 2D and 3D is found by calculating the cross product of vectors starting at the optical center to the both endings of the line, as in Figure 5. As a result, each point on the object must be perpendicular to that normal vector after being rotated and translated.

Hence, equation (5) is to be minimized. To make the minimization more robust to outliers, particularly for partially occluded scenes, a weight variable is assigned to each point or line indicating how much reliable that feature can be as described in [42].

$$e = N.(Rp + t) \tag{5}$$

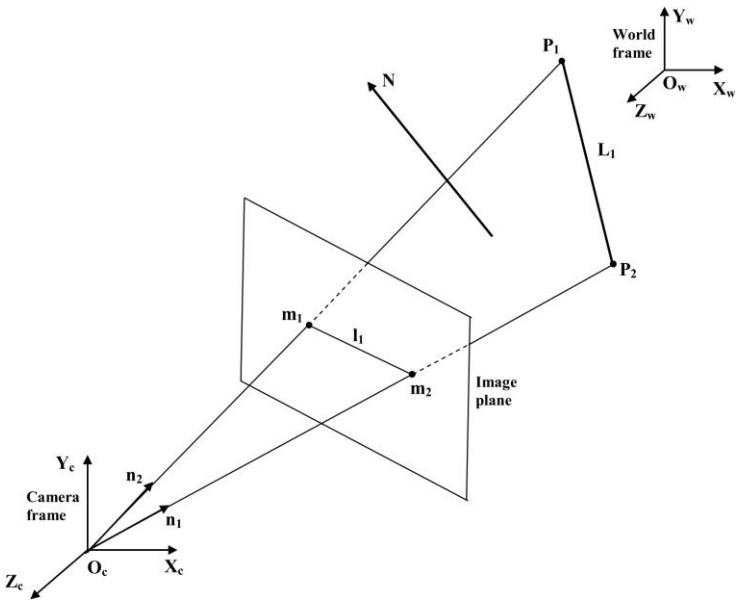


Figure 5 - Two vectors from the optical center O_c to the two endings of the line P_1 and P_2 *[40]*.

A recent study in [33], finds the projection matrix based on direct linear transformation [26] in the first place. Then the approach proceeds by decomposing the resulting matrix into sub-matrices to extract rotation and translation parameters. The redundancy of data is increased by expressing the correspondences in two coordinate systems, Euclidean and Plücker, to have more accuracy. However, there is no significant improvement reported in comparison with the results from [38] and [39].

Moreover, many papers present methods that include pose estimation from a combination of features such as points and lines to improve accuracy and robustness. Some of the methods with this approach are found in [41], [15], [12] and [43]. In [32], the approaches of OPnPL and EPnPL are presented for both point and line features based on their point-only counter-parts, OPnP [8] and EPnP [10] respectively.

Chapter 2

2. Pose Estimation

This chapter presents the approaches used in the experiments in chapter 4.

2.1 Concept

Perspective geometry describes how an object in the world coordinate system, projects on the image plane. In order to map each 3D point into its corresponding 2D point, it is required to have the camera calibration matrix. This matrix carries the information about the internal characteristics of the camera such as its focal length, and the external information on how it is related to the world outside. The work presented in this thesis is focused on estimating the position and orientation of a camera with respect to the scene that the camera is viewing. Consider a pinhole camera and a 3D point $X = (X, Y, Z)$ that is mapped to the point $x = (x, y)$ on the image plane in homogeneous coordinates as in equation (6).

$$x = K\,X \tag{6}$$

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{7}$$

The intrinsic parameters of the camera are expressed in matrix K that transforms the object points onto the image plane in a pinhole camera, see Figure 6. These parameters come from the characteristics of the camera. As in equation (7), the focal length is indicated as f_x and f_y in two dimensions. s is the skew parameter, it is used in the cases that the camera reference frame axes are not perpendicular to each other. (c_x, c_y) is the coordinates of the center of the image plane [44].

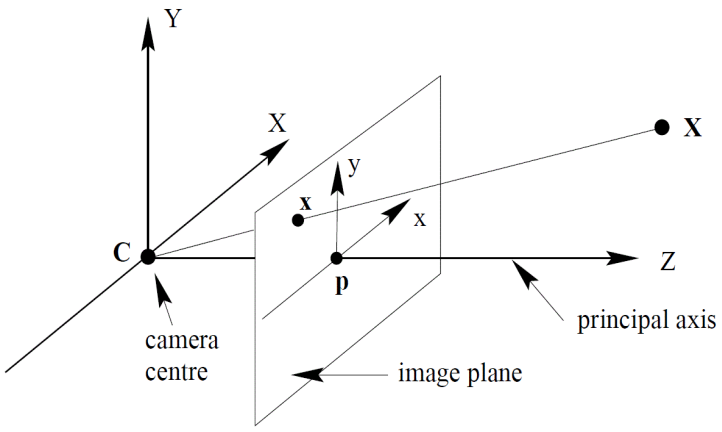


Figure 6 - Pinhole camera *[26]*.

By introducing another coordinate system, called the world frame or the object frame, to express object coordinates, it is required to take into account another transformation to project every point from one frame to another. This affine transformation consists of a rotation matrix and a translation vector. Consequently, in a general setting we will have:

$$x = K\,[R \quad t]\,X \tag{8}$$

Where R is the rotation matrix and t is the translation vector. They both contain extrinsic parameters in camera calibration that illustrate how the world coordinate frame is transformed to the camera frame, shown in Figure 7.

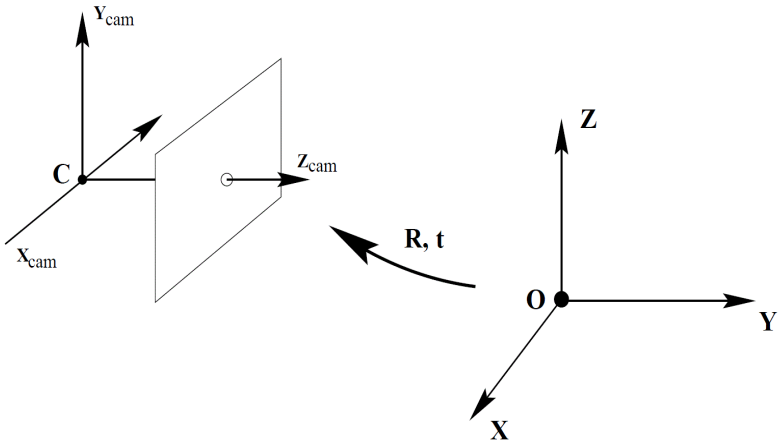


Figure 7 - The transformation from the world coordinate system to the camera coordinate system *[26]*.

Pose estimation is the process in which we determine the position and orientation of the world frame with respect to the camera frame. In other words, it includes extrinsic calibration and finding the extrinsic parameters. In this project, we assume that the intrinsic camera parameters are known. The method of intrinsic camera calibration used in this work will be explained in chapter 4.

2.2 Pose Estimation from Circle-Ellipse Correspondences

What follows in this section summarizes the concepts in [23] and [15] that are used in the experiments reported in section 4.2.

Let us consider a circle with known center coordinates and normal vector to its supporting plane as C_w and N_w respectively. In perspective projection, the image of this circle on the image plane will be an ellipse in a general configuration, see Figure 8. According to [15], the pose can be estimated from the following equations:

$$N_c = RN_w \tag{9}$$

$$C_c = RC_w + t \tag{10}$$

where C_c is the center and N_c is the normal vector to the supporting plane of the resulting circle in the camera coordinate frame. R is the rotation matrix and t is the translation vector.

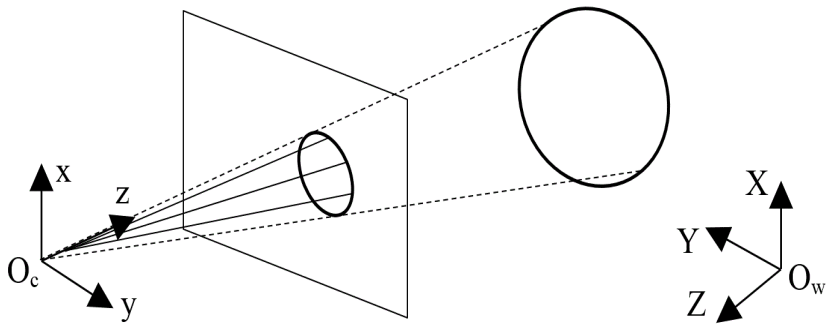


Figure 8 - Circle-ellipse correspondence

Primarily, in order to solve for R and t, we need to find the normal vector and the center of the image circle in the camera frame. As in [23], considering a point on the ellipse as (x_e, y_e), the ellipse equation in the image plane in quadratic form is:

$$Ax_e^2 + 2Bx_e y_e + Cy_e^2 + 2Dx_e + 2Ey_e + F = 0 \tag{11}$$

Alternatively, in matrix form, it can be written as:

$$x^T \begin{bmatrix} A & B & D \\ B & C & E \\ D & E & F \end{bmatrix} x = 0 \tag{12}$$

Let the ellipse be the base of an oblique elliptical cone and the optical center to be its apex, as shown in Figure 8. With the image plane placed at $z = f$, and f being the focal length, every point on the oblique elliptical cone can be expressed as:

$$P^T = k(x_e, y_e, f)^T \tag{13}$$

where k is a scale factor indicating the distance of each point of the cone to the origin of the camera frame. As a result, from the equation (12) and equation (13), the oblique elliptical cone is:

$$Q = \begin{bmatrix} A & B & \frac{D}{f} \\ B & C & \frac{E}{f} \\ \frac{D}{f} & \frac{E}{f} & \frac{F}{f^2} \end{bmatrix} \tag{14}$$

By decomposing Q, as in equation (15), we will have the eigenvalues and the eigenvectors of the oblique elliptical cone in equation (16) and equation (17) respectively.

$$Q = VDV^T \tag{15}$$

Where D and V are:

$$D = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \tag{16}$$

$$V = \begin{bmatrix} v_{11} & v_{21} & v_{31} \\ v_{12} & v_{22} & v_{32} \\ v_{13} & v_{23} & v_{33} \end{bmatrix} \tag{17}$$

Considering the orthogonality constraint of the rotation matrix, the normal vector and the center of the circle in the camera coordinate frame are calculated as in equation (18) and equation (19). The details on how to get these equations from equation (15) is provided in [23].

$$C = \left(S_3 \frac{\lambda_2 r}{\sqrt{-\lambda_1 \lambda_3}} \right) V \begin{bmatrix} S_2 \frac{\lambda_3}{\lambda_2} \sqrt{\frac{\lambda_1 - \lambda_2}{\lambda_1 - \lambda_3}} \\ 0 \\ -S_1 \frac{\lambda_1}{\lambda_2} \sqrt{\frac{\lambda_2 - \lambda_3}{\lambda_1 - \lambda_3}} \end{bmatrix} \tag{18}$$

$$N = V \begin{bmatrix} S_2 \sqrt{\frac{\lambda_1 - \lambda_2}{\lambda_1 - \lambda_3}} \\ 0 \\ -S_1 \sqrt{\frac{\lambda_2 - \lambda_3}{\lambda_1 - \lambda_3}} \end{bmatrix} \tag{19}$$

S_1, S_2, S_3 are either $+1$ or -1, giving eight possible set of solutions for N and C. Parameter r is the radius of the circle. Based on the coordinate system configurations, the incorrect solutions are eliminated under conditions in equation (20) and equation (21). These conditions ensure that the circle is faced towards the camera.

$$C.[0 \quad 0 \quad 1]^T < 0 \tag{20}$$

$$N.[0 \quad 0 \quad 1]^T > 0 \tag{21}$$

Therefore, we can put the normal vector and center of the circle in the camera frame and its 3D coordinates in the world frame into equation (9) and equation (10), and construct a system of linear equations to solve for pose parameters. Each of the equation (9) and equation (10), give three sets of equations including the information from a single circle-ellipse correspondence. Since we have twelve unknown parameters, nine for rotation matrix and three for translation vector, this system requires at least two circle-ellipse correspondences to be solved.

2.3 Pose Estimation from Line Correspondences

What follows summarizes the concepts in [41] that are used in the experiments reported in section 4.3.

In the world coordinate frame, we assume the line L with direction vector V and a point P on this line. The corresponding line l on the image plane has the direction vector v and a point on it, p, in the camera coordinate system as in Figure 9. Based on [41], with the rotation matrix R and the translation vector t, the relationship between the two frames can be formulated as:

$$v = RV \tag{22}$$

$$p = RP + t \tag{23}$$

Considering the object line and its image on the image plane, we define a plane containing the two lines and optical center of the camera. The coplanarity of the corresponding lines is one of the geometrical constraints commonly exploited in this type of pose estimation problems. We have discussed some of them in chapter 1. Because of the coplanarity, the normal vector to this plane is perpendicular to any vector in this plane including the direction vectors of the corresponding lines and the points on them. Therefore, we find the normal vector to the plane by normalizing the cross product of v and p, so that it is expressed in the camera reference frame as well.

$$n.v = 0 \tag{24}$$

$$n.p = 0 \tag{25}$$

By substituting equation (22) and equation (23) into equation (24) and equation (25), we have two equations to minimize.

$$n.(RV) = 0 \tag{26}$$

$$n.(RP + t) = 0 \tag{27}$$

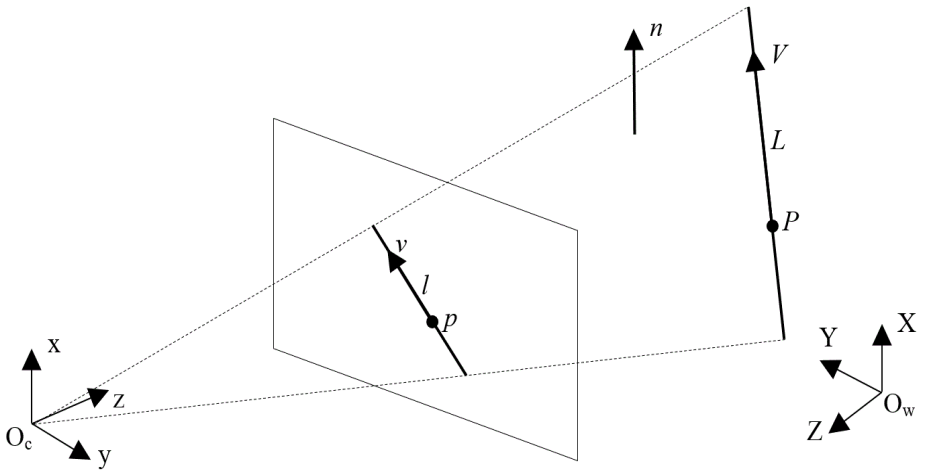


Figure 9 - 2D-3D line correspondence

We proceed to the solution of pose estimation in two steps. First, to solve for rotation, we minimize equation (26) by employing the Trust-Region algorithm [45] iteratively. The Trust-Region method is a nonlinear iterative optimization. In each iteration, it searches for the local minima of an approximated function in a specific interval and increases the interval when the local minima is less than the function value at the center of the interval. The search continues until it has converged to the global minimum [46]. Due to the rotation ambiguity, it may not converge to the optimum rotation in the first run. Since it is rapid in converging to the minima, we recommend using Trust-Region method in iteration, and choosing the best estimate to find the global minimum. The best estimate is chosen based on coordinate systems configurations. In order to avoid the optimization to converge to incorrect minima and to decrease the search time, we generate an arbitrary initial guess from the range of possible rotation angles for each axes. For rotation around X and Z axes, we assume the range to be $[-\pi, \pi]$ and $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ for Y axis. Having recovered the rotation, the translation vector is simply computed by solving the system of linear equations in equation (28) that is derived from equation (27).

$$t = n^{-1}(n.RP) \tag{28}$$

There are six degrees of freedom, three for the angles of rotation and three for the translation vector. Since each step is done separately, the pose estimation requires at least three line correspondences.

The methods described above were used in our experiments for the pose estimation problem from circle-ellipse and line-line correspondences. Before providing experimental results, we will discuss the methods employed for feature detection (i.e. ellipses and lines) in images in the next chapter.

Chapter 3

3. Feature Detection

In pose estimation, the corresponding features on the image plane and in the world coordinate system are taken into account. In our project, these features are ellipses as images of circular objects and straight lines. Before presenting the experimental results, we will discuss the methods that are employed in our experiments for collecting the 2D information on these features.

3.1 Ellipse Detection

The method in [47] proposes a numerical least-squares method for ellipse detection inspired by [48]. By applying the quadratic elliptical constraints, equation (30), they directly solve for ellipse coefficients in a least-squares sense based on algebraic distances of the points to the ellipse. The elliptical constraint ensures that a conic in the form of equation (29) satisfies the constraint in equation (30) [48]. Although this method runs very fast, it does not yield an accurate and reliable ellipse equation. Moreover, it is highly sensitive to noise and outliers. Particularly, in cases with irregular point distribution.

$$C(x, y) = Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \tag{29}$$

$$4AC - B^2 = 0 \tag{30}$$

An accurate and robust yet fast approach to ellipse detection is presented in [49]. It converts the conic equation to the dual conic representation. This new representation is based on lines instead of points. The dual conic C^* is defined as a set of lines such as l, that are tangent to the surface of the conic, see Figure 10, and satisfies the following equation:

$$l^T C^* l = 0 \tag{31}$$

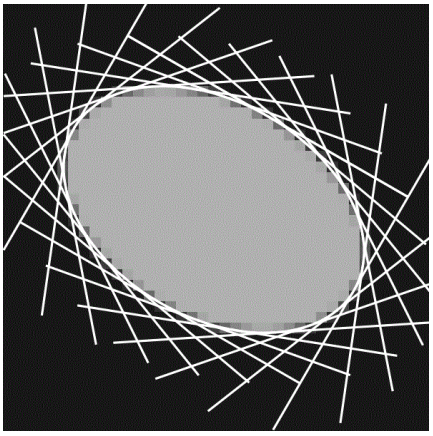


Figure 10 - Dual conic with the lines tangent to its surface *[49]*.

The authors show that the ellipse constraint in Equation (30) is valid for a dual ellipse in the form of $F^* = 1$ which is defined in equation (32). Each line is expressed in homogeneous coordinates and is obtained from the gradient magnitude at the pixel x, $I = [I_u, I_v]$, within a contour around the ellipse, equation (33). This contour is estimated by setting a threshold on the gradient magnitude map of the image.

$$F^* = \frac{1}{4|C|}\,(4AC - B^2) \tag{32}$$

$$l = \left[\frac{I_u}{|I|}, \frac{I_v}{|I|}, -\frac{I}{|I|}x\right] \tag{33}$$

Consequently, the ellipse is detected in dual representation from a linear least-squares minimization of equation (31) under dual ellipse constraint. The resulting dual ellipse coefficients can be converted to normal ellipse representation afterwards.

3.2 Line Detection

There are a number of line detection methods currently proposed in the literature. Line Segment Detector (LSD) is presented in [50]. In this approach, a list of pixels is made based on the gradient values, with the highest value as the first element on the list. The first pixel in the list is taken as the seed of the first support region to find. The reason is that there is a higher chance that the pixels with highest gradient belong to edges. The support region for the possible line is created by searching the neighboring pixels and choosing those with close

angles to the seed within a threshold. The angle of the possible line at each pixel is computed, being orthogonal to its gradient. The search ends when there is no more neighboring pixel with a close angle to be added to the current line segment. The next unvisited pixel in the list is chosen as the new seed. The support regions are approximated by rectangular line segments, as in Figure 11. At last, the Number of False Alarms (NFA) is computed for each line segment. The line segments with a NFA less than a specified threshold are regarded as the meaningful segments, i.e. the detected lines.

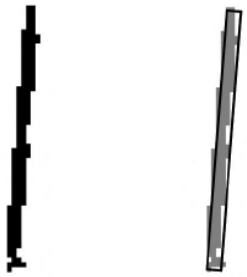


Figure 11 - Support region (left), rectangular line segment (right) *[50]*.

Number of False Alarms (NFA) is a probabilistic function employed to validate the line segments. It determines whether a line segment is meaningful or not. A meaningful segment is defined as a segment for which the structure has a low chance of occurrence in the structure of the whole image [51]. NFA rejects wrong detections caused by misleading background structure or image noise such as white noise. The NFA of a meaningful line segment is nearly zero.

LSD performs more accurate on small images than high-resolution images. However, for images that are larger in size, the main problem of this approach is over-segmentation. Lines with longer lengths are prone to over-segmentation in higher resolutions. The method proposed in [52] that is called MLSD (Multiscale LSD), is a multiscale approach based on LSD. The goal of the authors of MLSD is to deal with the problem of over-segmentation. At first, they begin the line detection by applying LSD at the coarsest scale. This step gives a rough estimate of the location of longer lines in the image. Then by gradually going to next scales, the fusion score is measured for the segments detected in those locations. This score is computed based on the comparison of the meaningfulness, i.e. NFA, of each individual segment

to a greater segment, that is the merged segments in that location. The fusion score determines if the segment is a part of the larger segment and should be merged or not. At the end, those segments with acceptable NFA are kept as detected lines.

Although MLSD almost handles the over-segmentation problem, due to its multiscale nature, its execution time is not negligible. Another line detection method called EDLines from [53] runs faster than both previously mentioned approaches. In our experiments, the resulting detected lines with EDLines are comparable to both methods as well. EDLines exploits the Edge Drawing [54] algorithm to find edge segments, since edges are considered the best candidates for lines. They apply least-squares line fitting on the pixels for each edge segment. Therefore, edge segments are divided into line segments. When all the edge pixels are processed, the detected lines are obtained from a validation step based on the length and the NFA of each line segment.

According to the above explanations, the ellipse detection approach using dual conics from [49] and EDLines [53] algorithm for straight line detection are exploited in our experiments. The experimental results will be discussed in the next chapter.

Chapter 4

4. Experimental Results

In this chapter, we present and discuss the experimental results obtained from the approaches mentioned in chapter 3. We used a Basler acA2040-25gm GigE color camera along with a Kowa lens with a focal length of 25mm and the aperture range of F1.4 to F16, as demonstrated in Figure 12. Furthermore, the pixel size is $5.5\ \mu m\ \times 5.5\ \mu m$. We have implemented the camera pose estimation in Matlab R2016a (9.0.0.341360). The experiments are executed on a computer in Windows 7 with the specifications of Intel® Core(TM) i7-5820K, CPU @ 3.30 GHz and RAM of 16 GB.

Figure 12 - The camera in the setup

4.1 Intrinsic Calibration

In order to find the camera matrix, *Camera Calibration Toolbox for Matlab* [2] by Jean-Yves Bouguet was used. This toolbox requires several images of a checkerboard from various orientations and positions. It computes the calibration information by extracting the checkerboard pattern corners and based on the Zhang's [55] calibration method. We used 40 images of a checkerboard pattern with 9×11 grid and the square size of 23×23 mm. Figure 13 shows a sample image of the checkerboard.

Figure 13 - The checkerboard used for the intrinsic calibration

4.2 Experiments Using Circles

This experiment includes estimating the pose of the camera using two circles attached to a surface with different radius. The radius of the larger circle is $100\ mm$ and the smaller one is $75\ mm$. A checkerboard is also placed in the image to serve as ground truth for the estimated pose. In order to detect circles and retrieve their equations, the code from [49] in Matlab has been used. The detected ellipses on the acquired image are marked in red in Figure 14(b). Based on Section 3.2, the normal vector and center of the circles are derived from the detected ellipse equations to estimate the pose.

[2] http://www.vision.caltech.edu/bouguetj/calib_doc/

(a) (b)

Figure 14 - (a) Image of two circles, (b) Detected ellipses marked in red.

Table 1 - Estimated and ground truth pose using the two circles in the image.

	Rotation	Translation
Ground Truth	$\begin{bmatrix} 0.02 & 0.99 & 0.00 \\ 0.62 & -0.02 & 0.00 \\ -0.77 & 0.01 & -0.62 \end{bmatrix}$	$\begin{bmatrix} -502.3 \\ -331.4 \\ 3466.5 \end{bmatrix}$
Estimation	$\begin{bmatrix} 0.00 & 1.24 & 0.18 \\ 0.76 & 0.40 & 0.89 \\ 1.51 & 2.43 & 0.40 \end{bmatrix}$	$\begin{bmatrix} 0.0 \\ 0.0 \\ 0.0 \end{bmatrix}$

The resulting estimated pose is unsatisfactory as seen in Table 1. This method of pose estimation does not result in a reasonable and acceptable rotation and translation matrices. In addition, the approach was supposed to enforce the orthogonality constraints on the rotation matrix, while we observe that the estimated transformation does not satisfy these constraints. A factor influencing this poor performance could be the normal vector and the center of the circles that are computed from the ellipse equations. These parameters cannot be validated since there is no information available for their actual value as ground truth. We conducted many other experiments on circles with various radius values and several orientations and

positions. In fact, the estimated pose in all the experiments that are performed using different circles are out of an acceptable range and unreliable.

Although circles are not among the most common features seen in the urban areas, buildings and construction sites, circle-ellipse correspondences could have contributed to the accuracy and robustness of the pose estimation as an extra information along with the line correspondences. However, the experiments that we have conducted are not convincing enough to use such an approach based on circles.

4.3 Experiments Using Straight Lines

Since buildings and urban environments mostly include straight lines, a metal structure has been used in our experiments to resemble such a view. The structure in the setup is shown in Figure 15 with the black background for better contrast and more accuracy in the detection. Line correspondences are chosen manually by pairing detected lines and the lines on the structure. The straight lines were detected using the code provided by [53] in C++[3]. Figure 16 shows all the detected lines in an image of the structure as an example. Those lines that belong to the structure and that are measurable are chosen for the estimation.

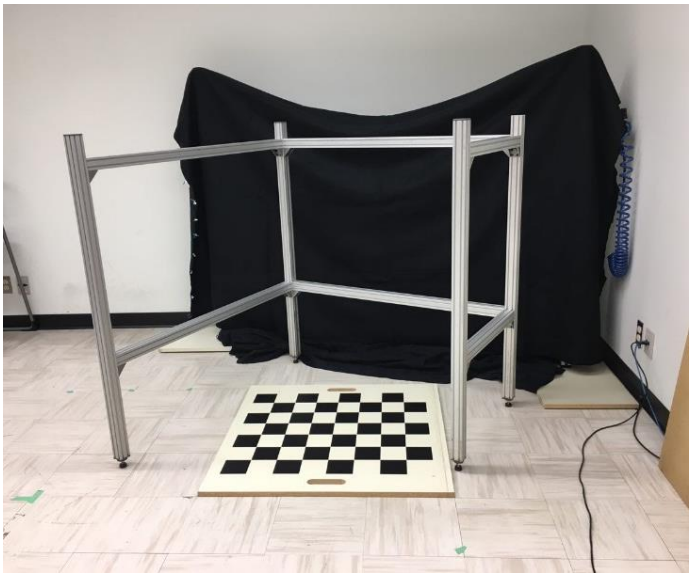


Figure 15 - The metal structure with black background and the checkerboard for the ground truth

[3] http://ceng.anadolu.edu.tr/CV/EDLines/

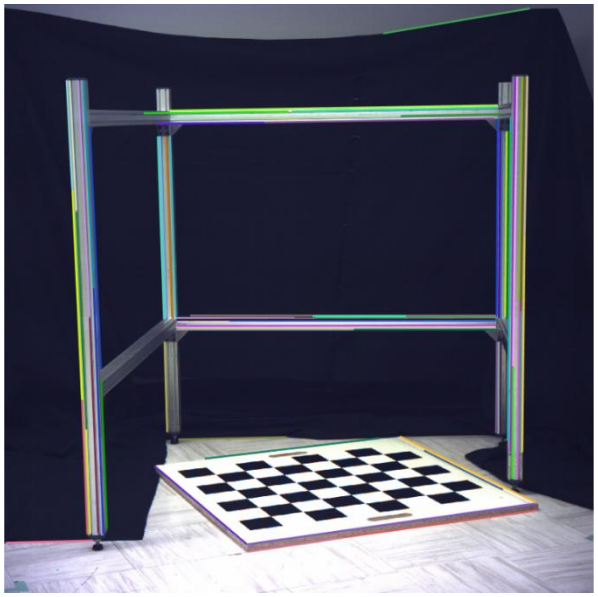


Figure 16 - All the detected lines are marked in colors

This experiment is conducted with 21 different poses of the camera that are illustrated in Figure 17. In order to evaluate the impact of the number of lines on the accuracy of the pose estimation, we performed the experiment with three different sets of lines, including 5, 10 and 15 to 25 lines (depending on the availability of the detected lines in each image). The point cloud of a bunny is reprojected on the image with the ground truth and the estimated pose. All the images with the bunny reprojections can be found in Figure 20, Figure 21 and Figure 22 at the end of this chapter. We define the reprojection error ε as the normalized sum of distances between the two reprojections:

$$\varepsilon = \frac{1}{n}\sum_{i=1}^{n}\sqrt{(P_{i\,Est} - P_{i\,GT})^2} \tag{34}$$

where n is the number of points in the point cloud. P_{Est} indicates the reprojected points using the estimated pose and the reprojected points using the ground truth are expressed as P_{GT}.

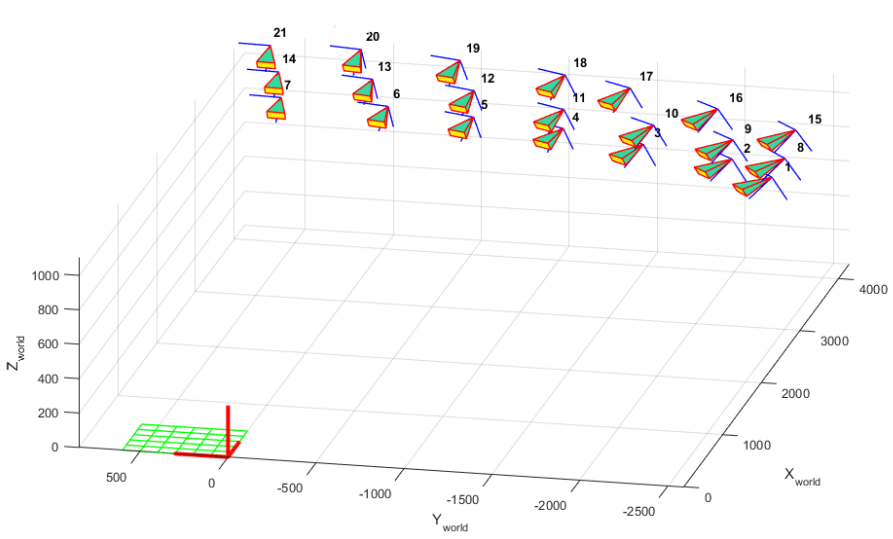


Figure 17 - Camera positions and orientations with respect to the object reference frame.

The average reprojection error is $0.0015\ px$ when using 5 or 10 lines, and $0.0016\ px$ when using 15 to 25 lines. In addition, it is believed that the slightly less accurate results of line set 3 may be caused by the fact that the level of noise increases while using more lines. The noise is due to the inaccuracy in the manual measurements or line detections in the image. Table 2 provides the reprojection error for each image. The total average reprojection error for the 21 images with the poses in Figure 17 in three different line sets is $0.0015\ px$.

Table 2 - Reprojection error of the images with the three line sets with the factor of 10^{-4} in pixels.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
5	26	26	7	7	24	11	18	23	28	37	4	19	8	7	15	16	9	9	7	8	15
10	20	20	15	11	9	19	17	13	19	29	12	13	9	22	16	21	15	8	9	8	15
15-25	10	20	21	11	12	22	20	15	21	25	12	14	19	36	15	15	19	12	8	8	20

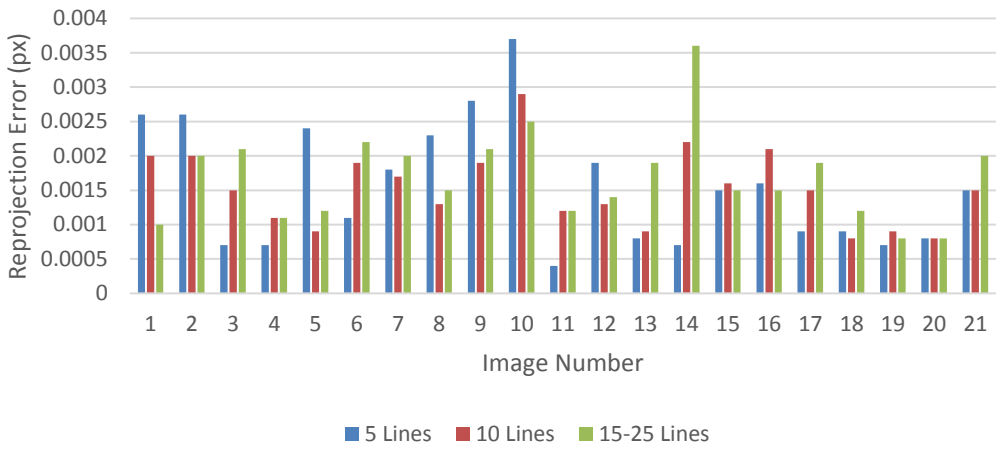


Figure 18 - Reprojection error comparison between three line sets.

Table 3 displays the execution time for each image and line set. The average execution time is 0.27 s for every line set signifying that this method is efficient with a quite consistent run time with no regard to the number of lines or position of the camera.

Table 3 - Execution time of all images in seconds.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
5	0.22	0.23	0.26	0.27	0.26	0.30	0.31	0.24	0.29	0.27	0.26	0.18	0.28	0.25	0.27	0.30	0.30	0.29	0.26	0.36	0.37
10	0.20	0.21	0.28	0.26	0.28	0.22	0.29	0.21	0.31	0.30	0.31	0.20	0.30	0.32	0.26	0.25	0.25	0.27	0.32	0.32	0.40
15-25	0.25	0.23	0.27	0.27	0.37	0.26	0.31	0.22	0.27	0.31	0.30	0.21	0.26	0.30	0.28	0.27	0.26	0.29	0.28	0.26	0.39

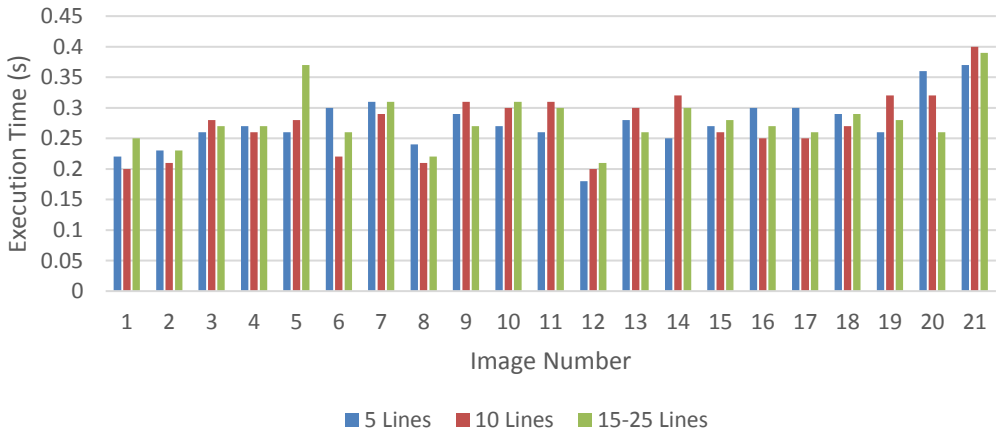
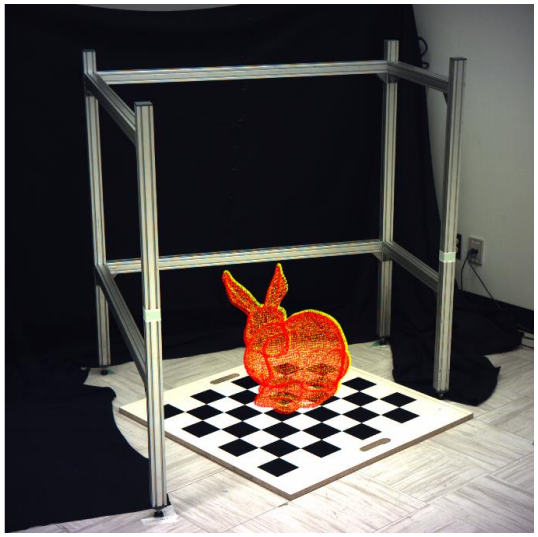


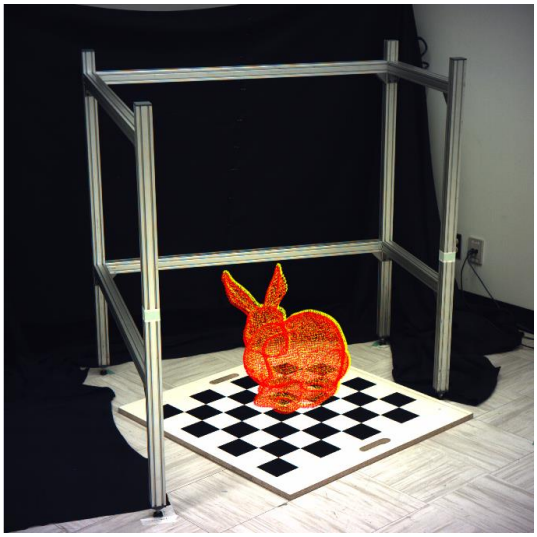
Figure 19 - Execution time comparison between three line sets.

The estimated pose using circle correspondences did not lead to promising results. However, the results explained in this chapter confirm that the presented approach for pose estimation using straight lines is accurate and efficient, while demonstrating reasonable execution time.

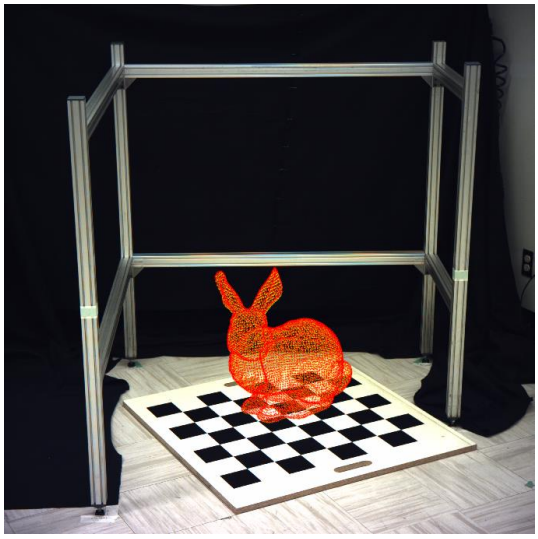
The superimposition of the 3D point cloud of the bunny using the estimated camera pose in red is compared to the bunny in yellow, which is superimposed using the ground truth pose in Figure 20 to Figure 22. Based on qualitative assessment of the results presented in these figures, it is safe to say that this approach provides sufficient accuracy to be used in augmented reality applications.



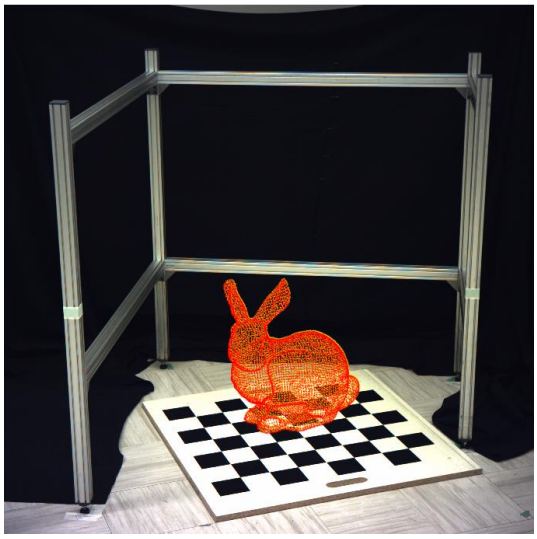
(1)



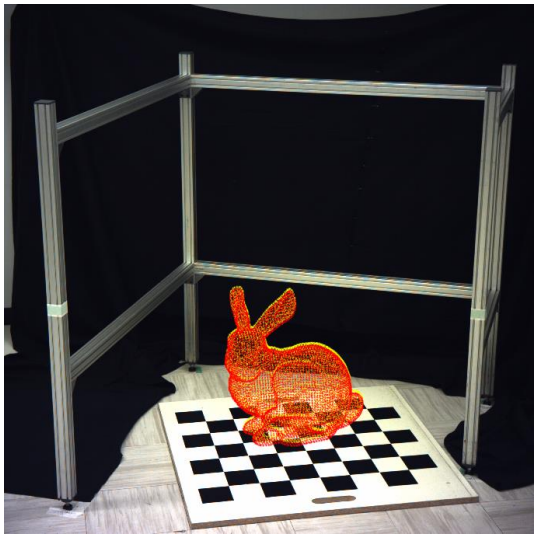
(2)



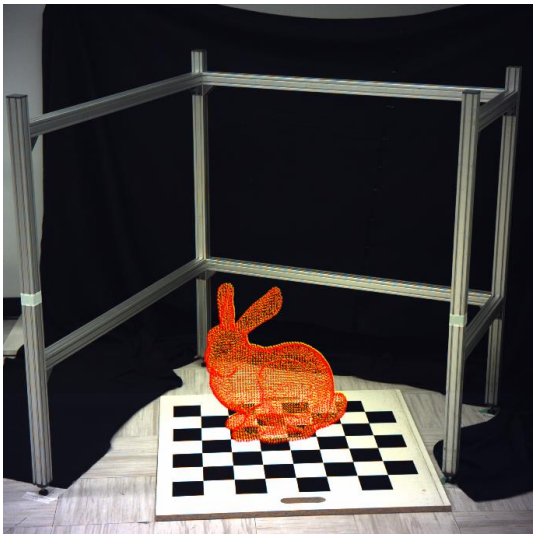
(3)



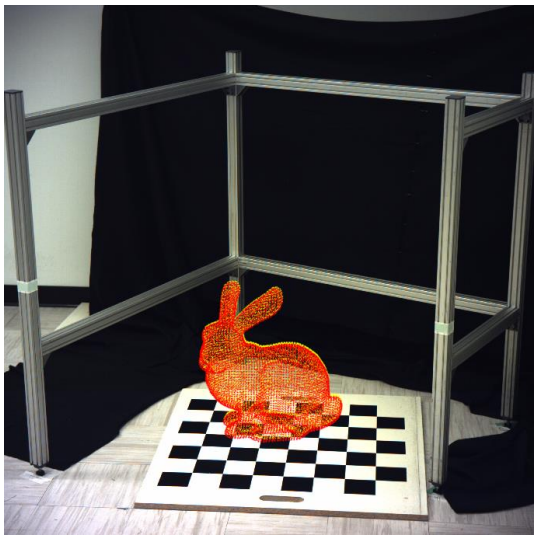
(4)



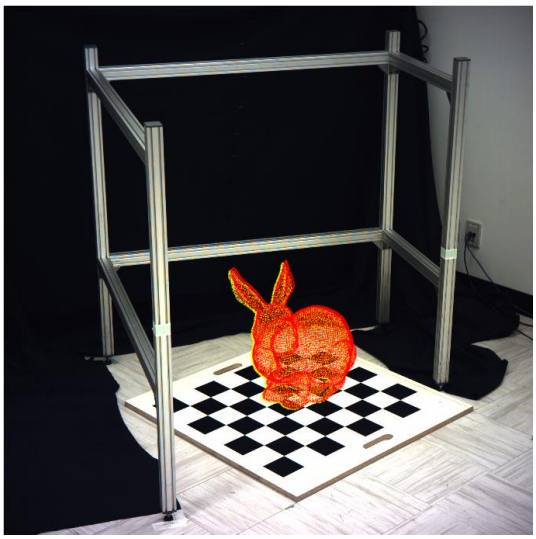
(5)



(6)

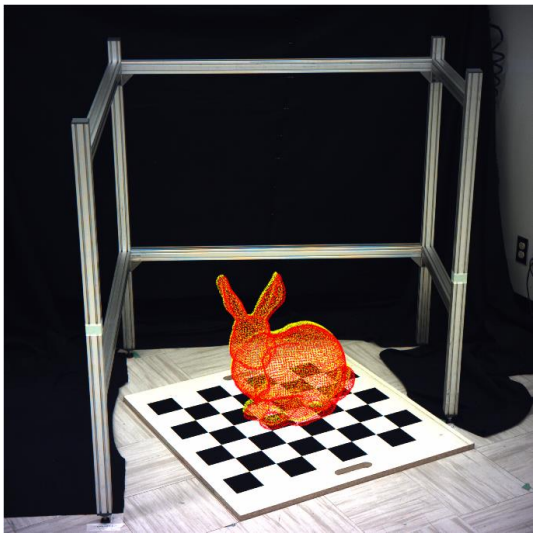


(7)

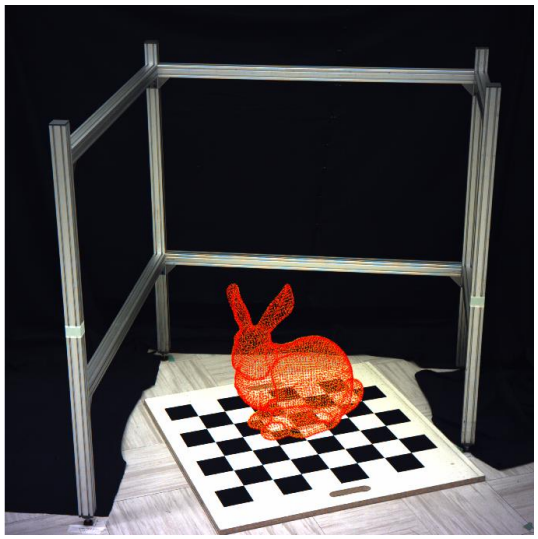


(8)

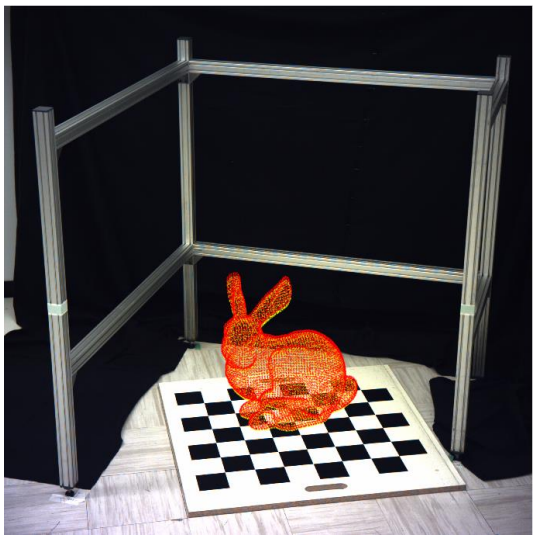
(9)



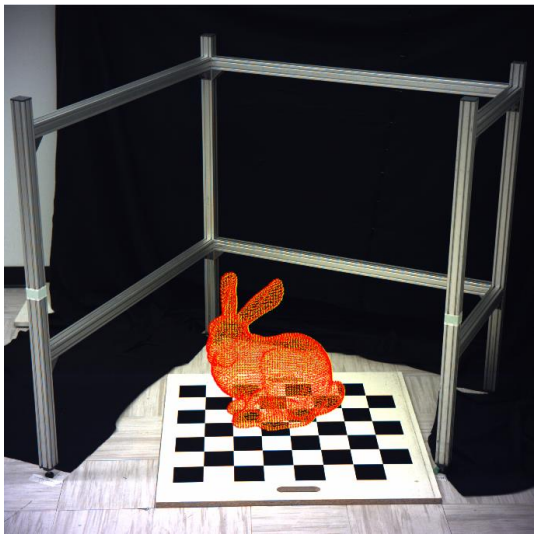
(10)



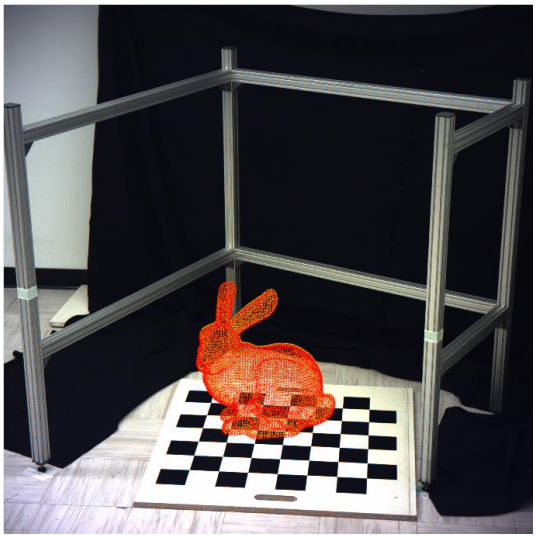
(11)



(12)

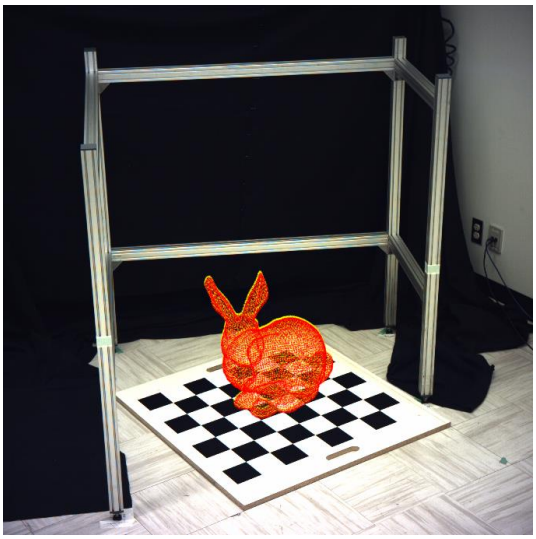


(13)

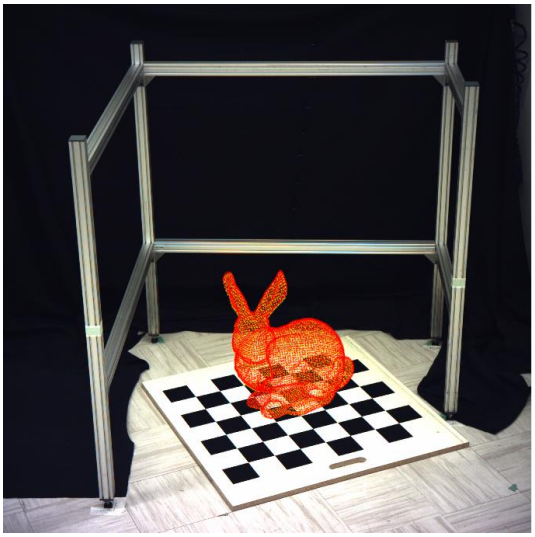


(14)

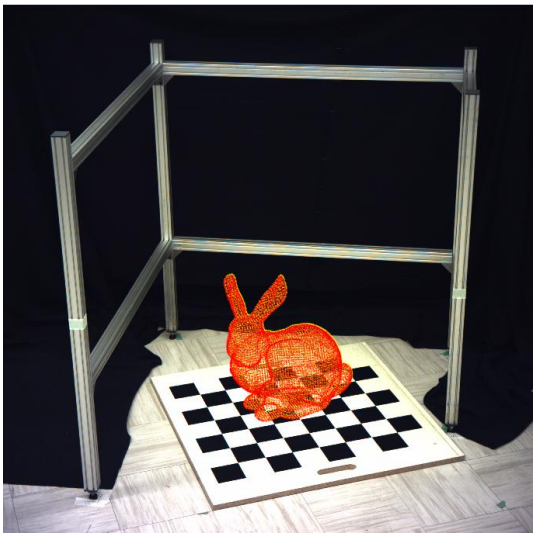
(15)



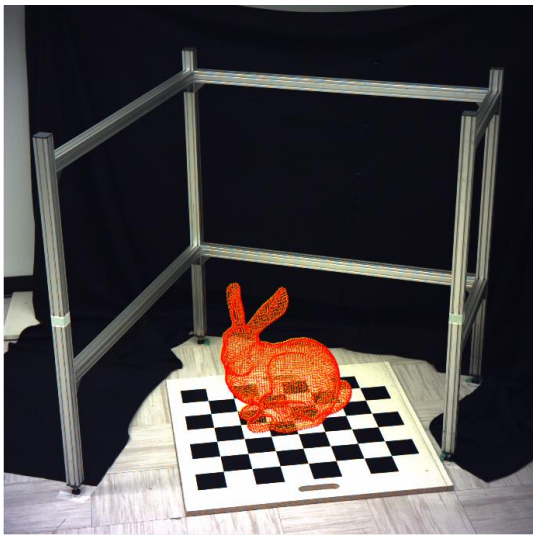
(16)



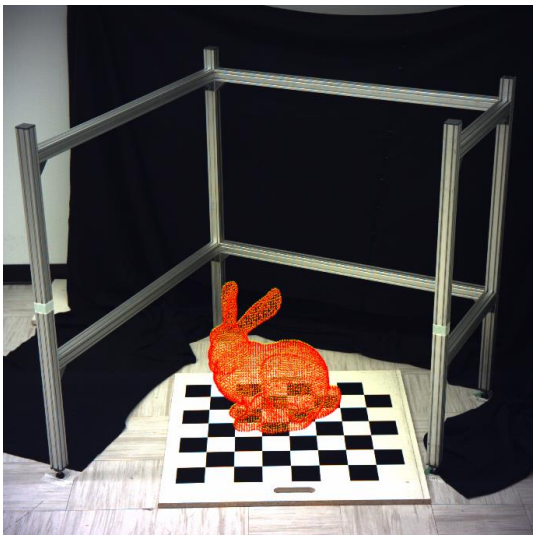
(17)



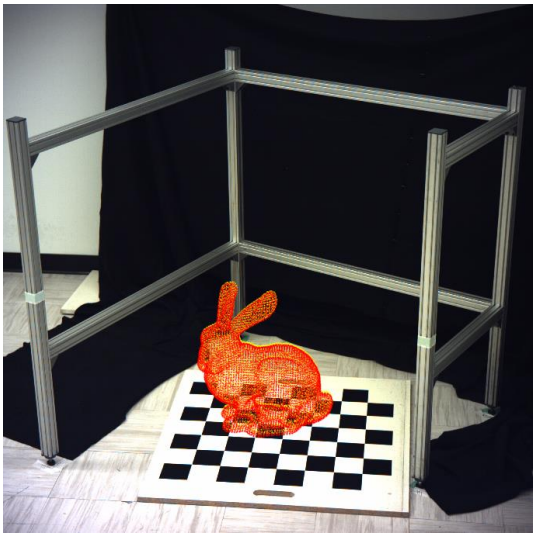
(18)



(19)

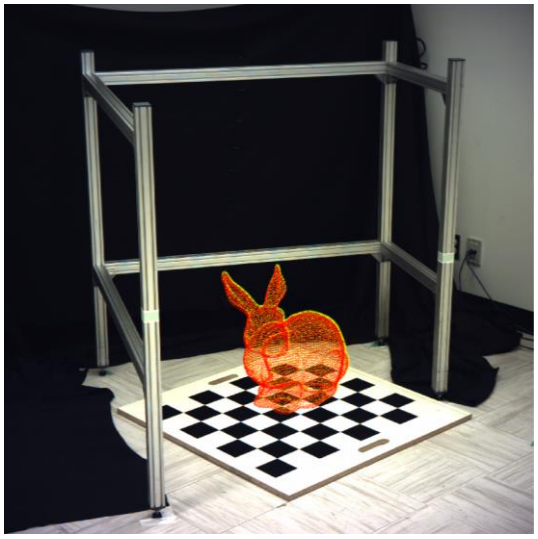


(20)

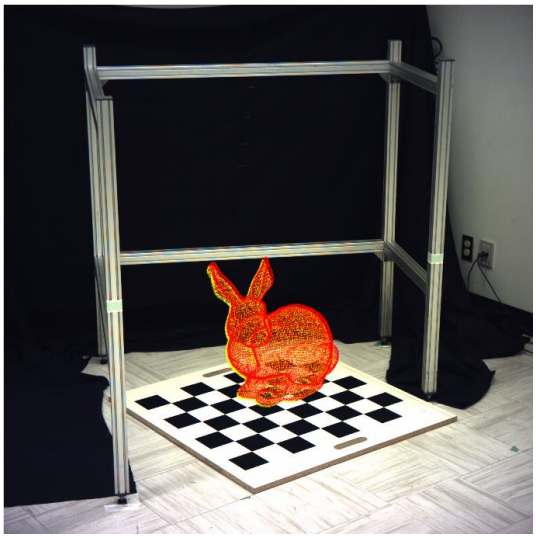


(21)

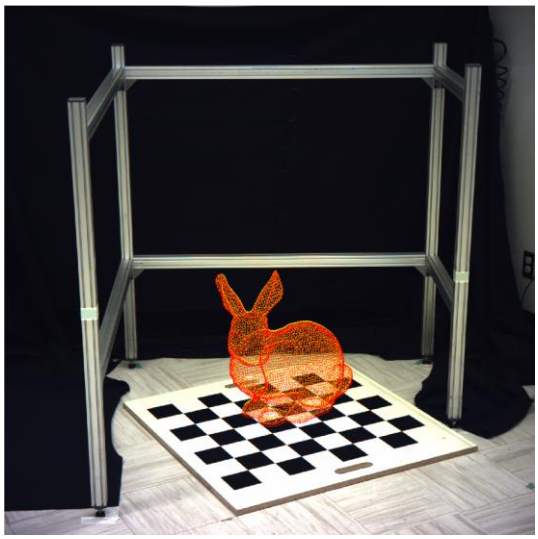
Figure 20 - Bunny reprojection. Red bunny is reprojected with the estimated pose using 5 lines, and the yellow bunny is the ground truth reprojection.



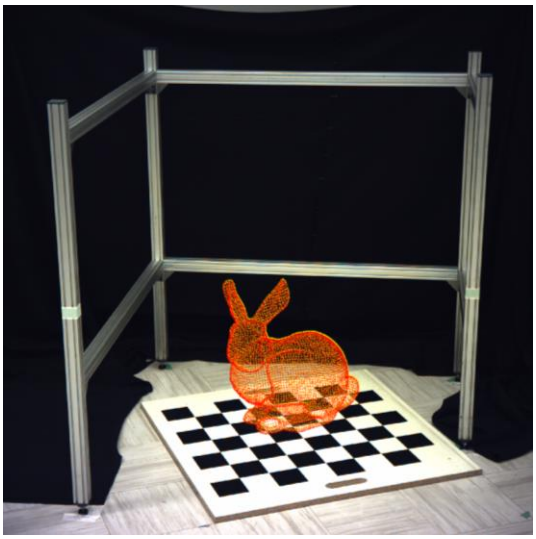
(1)



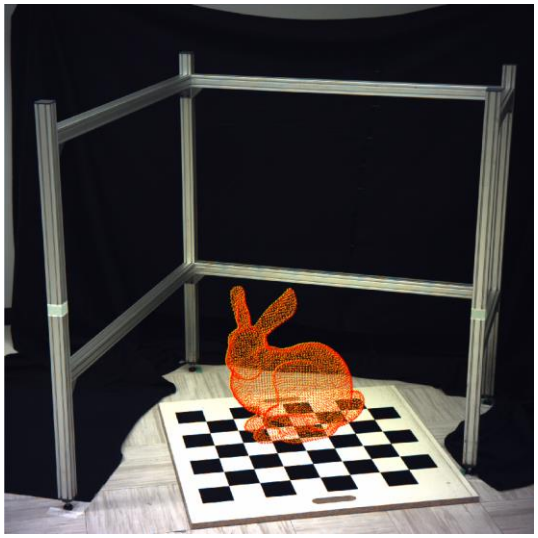
(2)



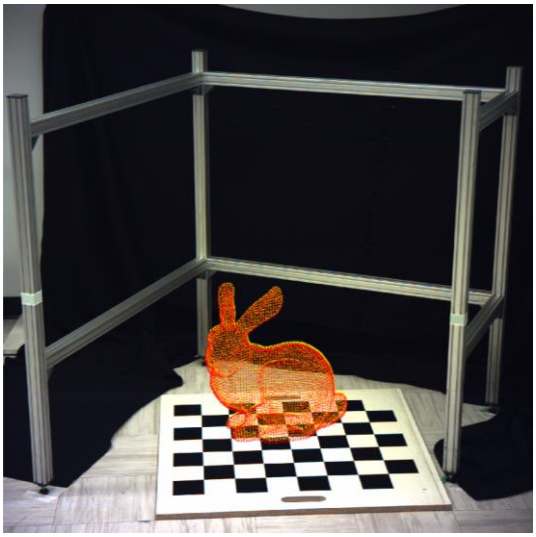
(3)



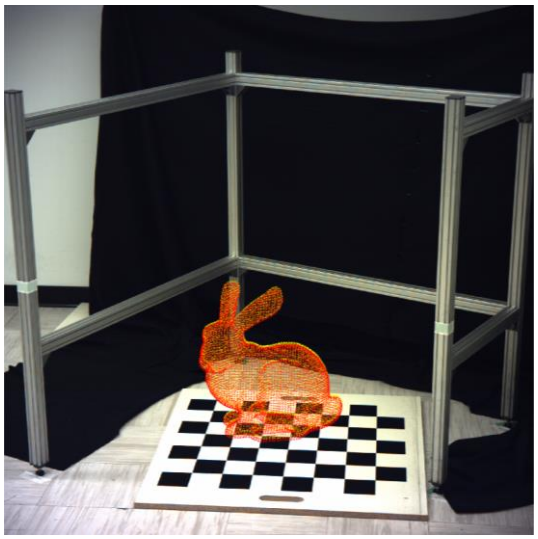
(4)



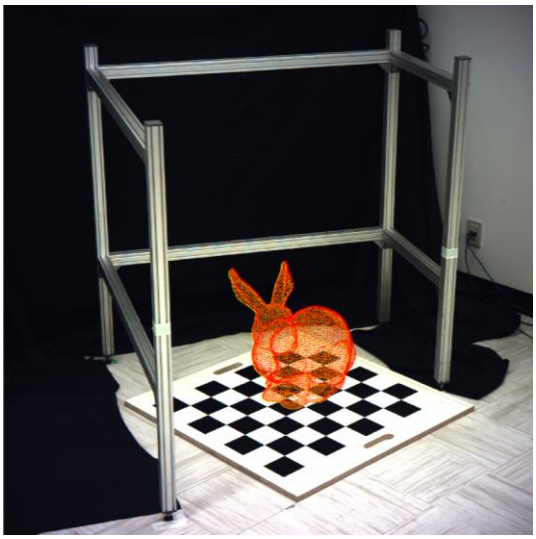
(5)



(6)

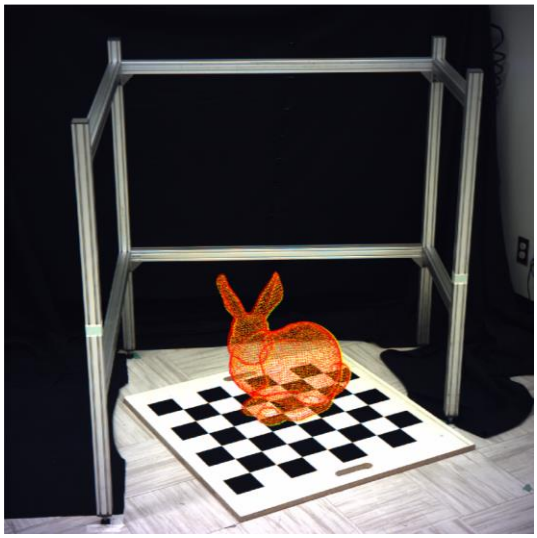


(7)

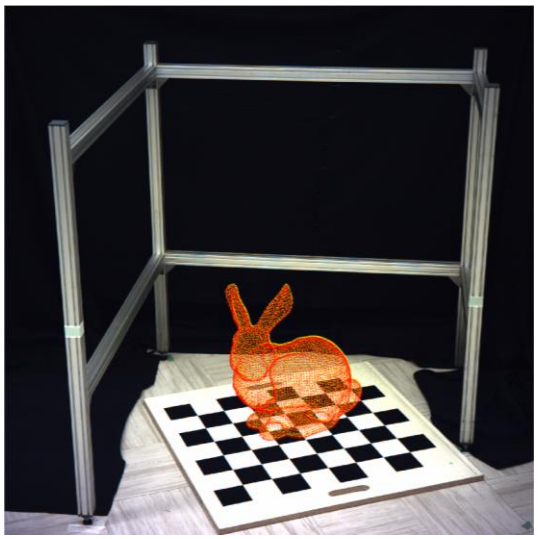


(8)

(9)

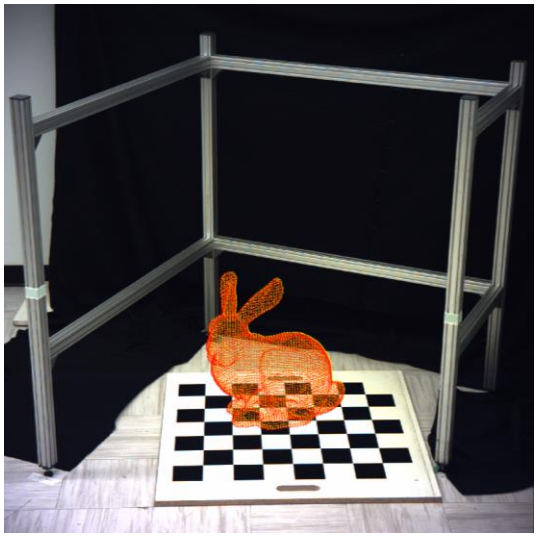


(10)

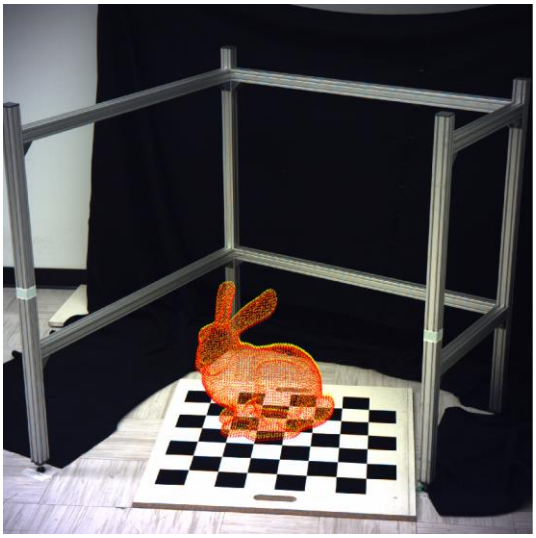


(11)

(12)

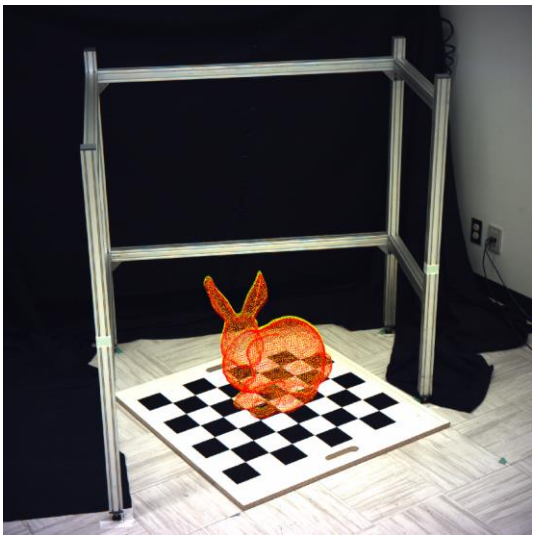


(13)

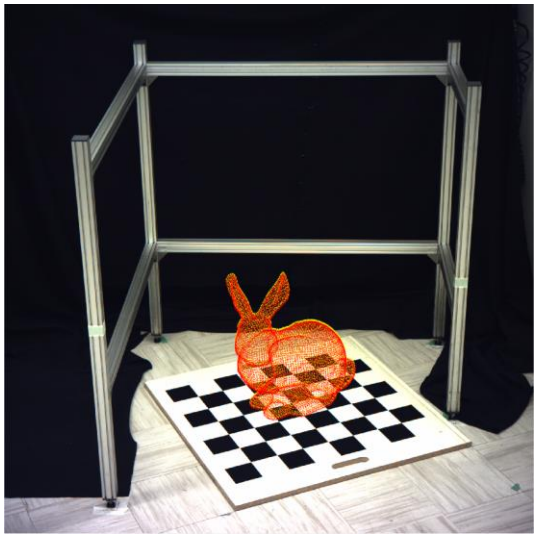


(14)

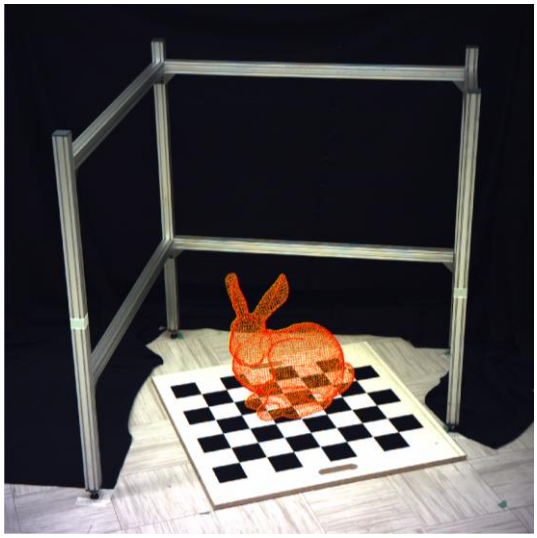
(15)



(16)

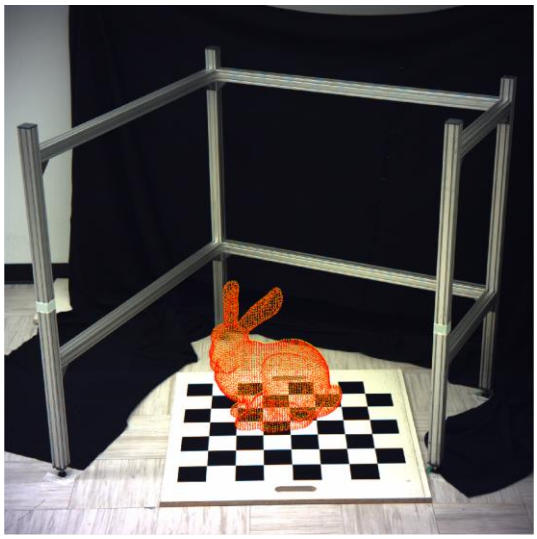


(17)

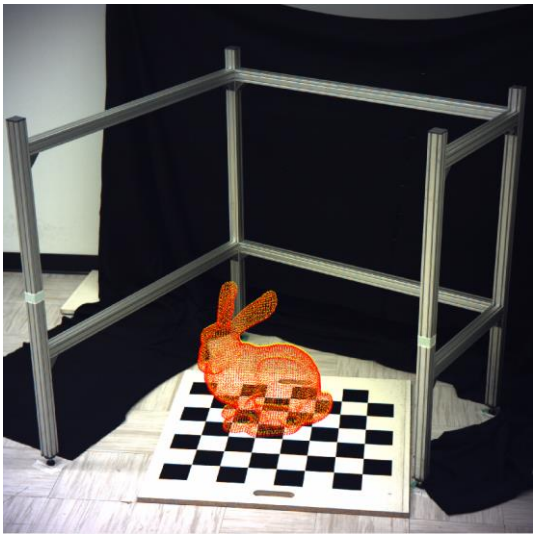


(18)

(19)

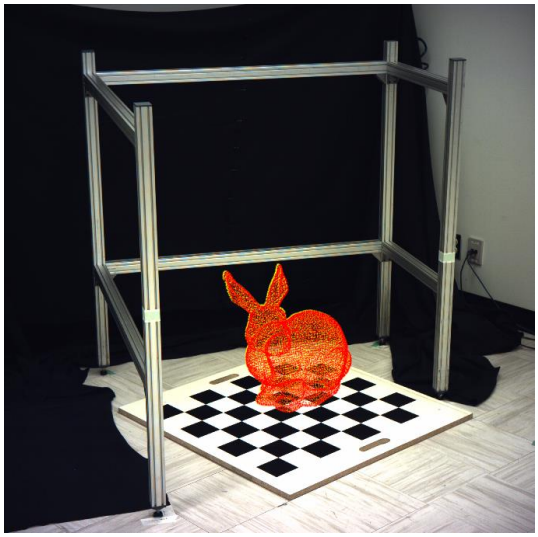


(20)



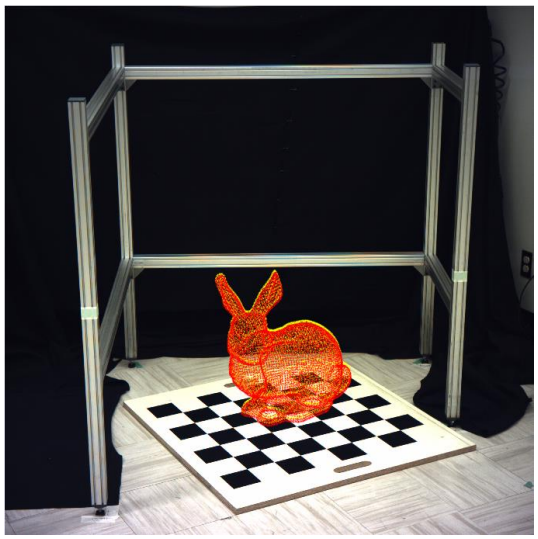
(21)

Figure 21 - Bunny reprojection. Red bunny is reprojected with the estimated pose using 10 lines, and the yellow bunny is the ground truth reprojection.

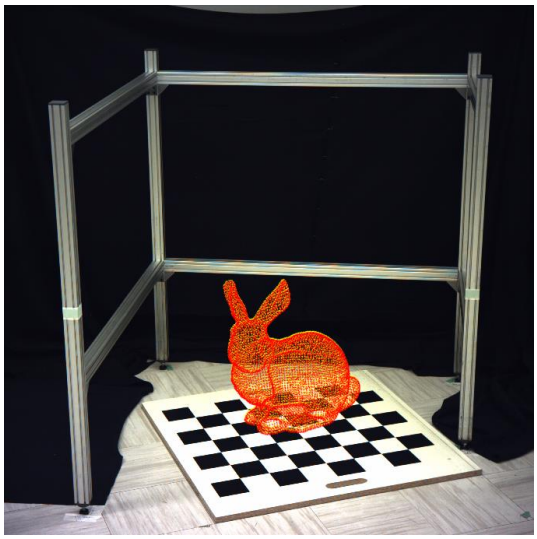


(1)

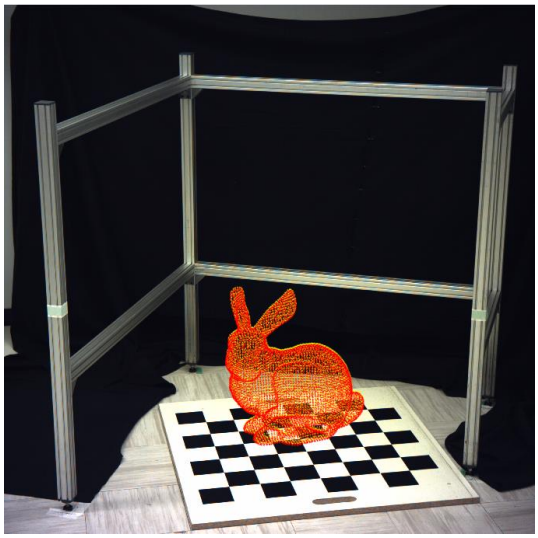
(2)



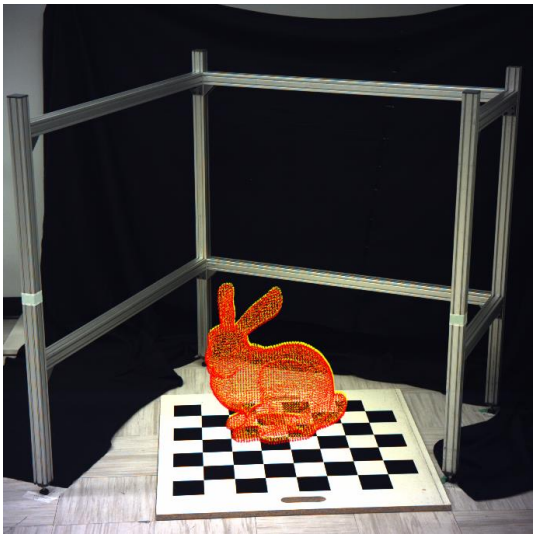
(3)



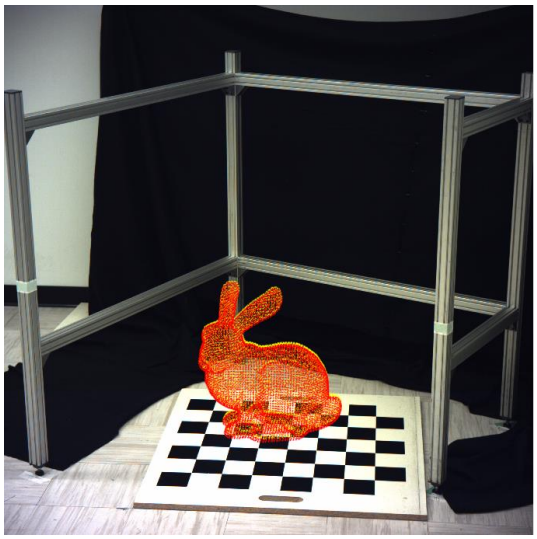
(4)



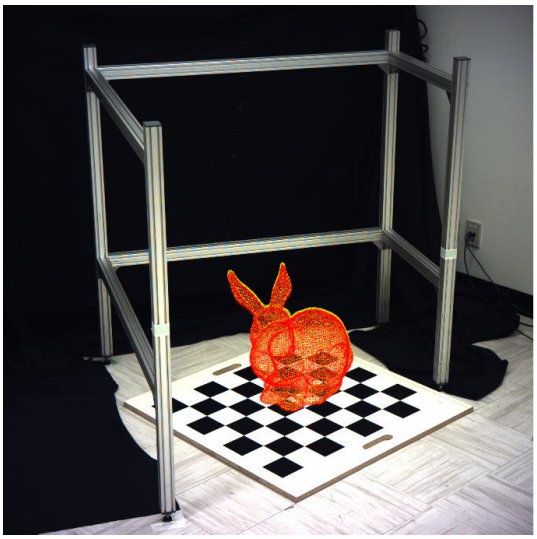
(5)



(6)

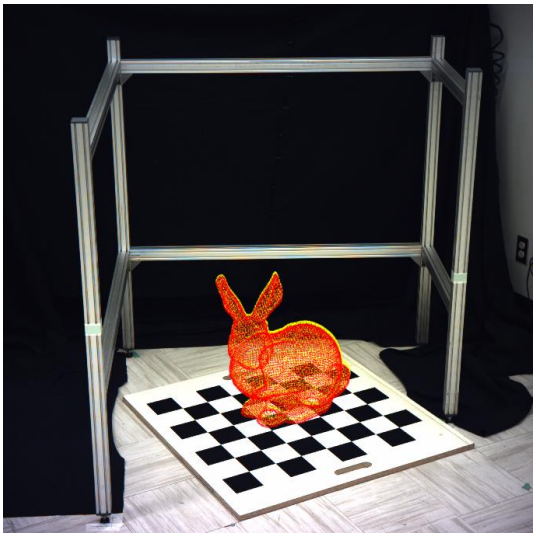


(7)

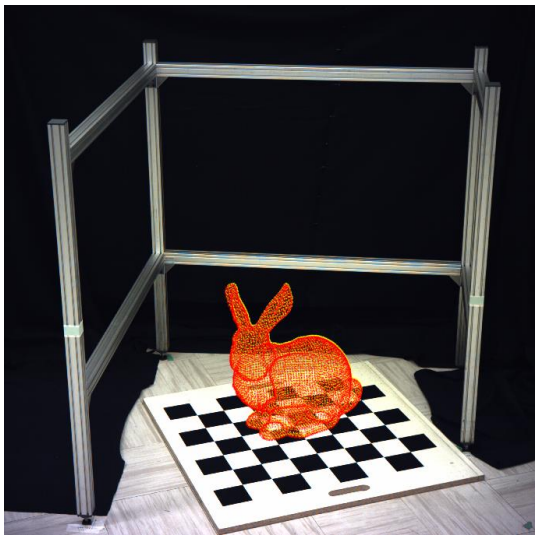


(8)

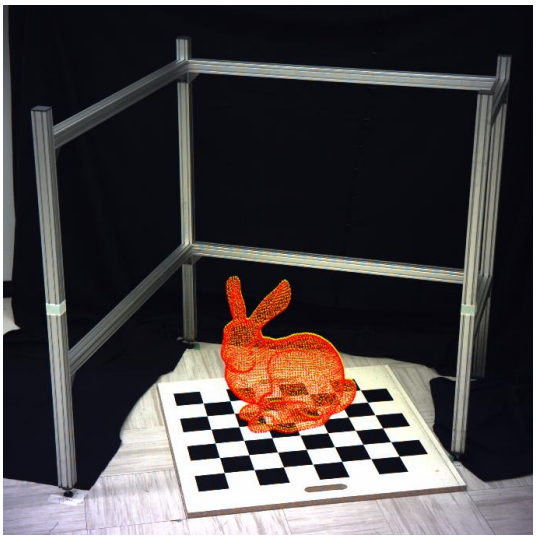
(9)



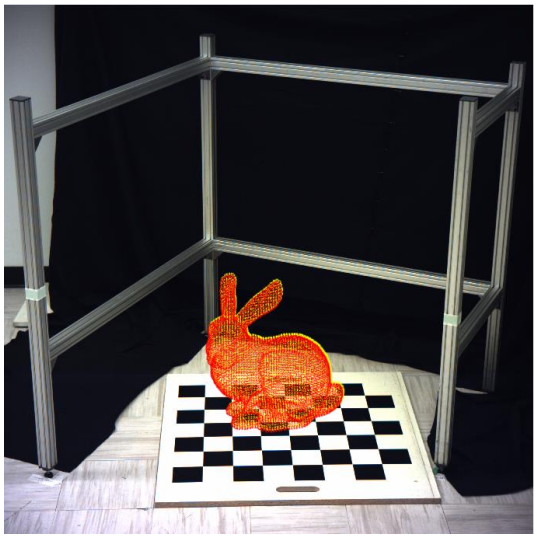
(10)



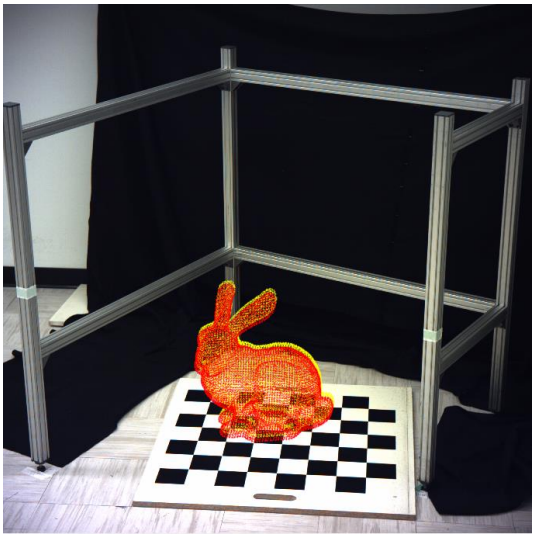
(11)



(12)

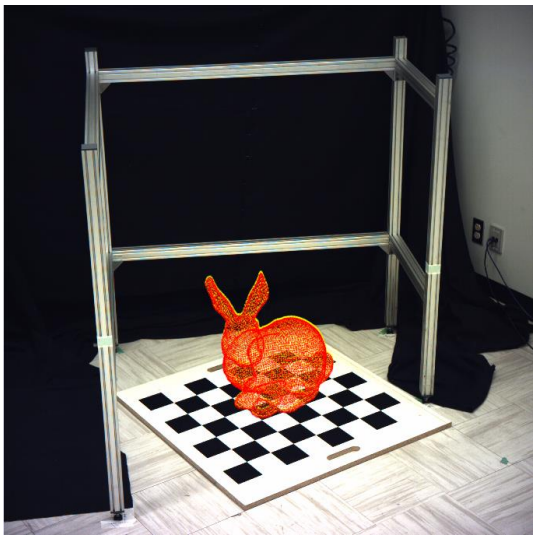


(13)

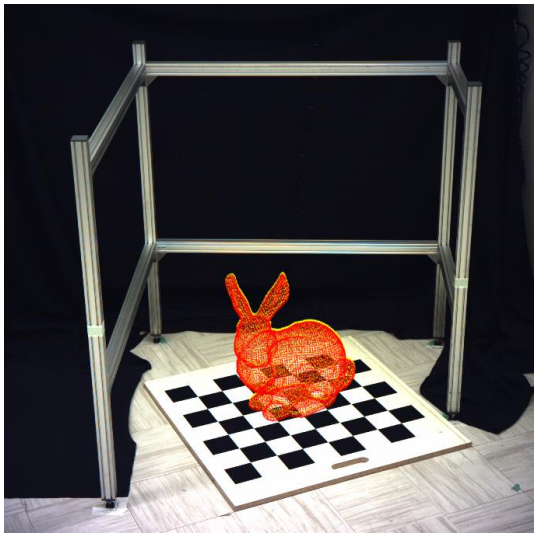


(14)

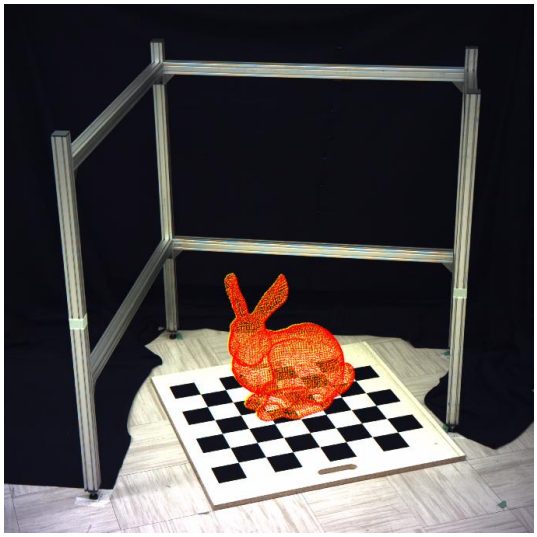
(15)



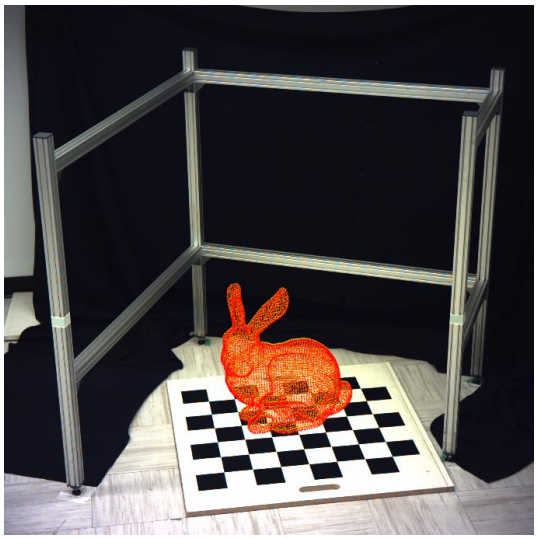
(16)



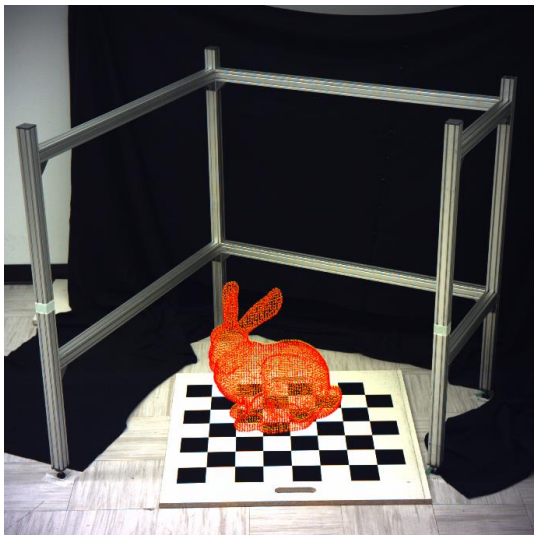
(17)



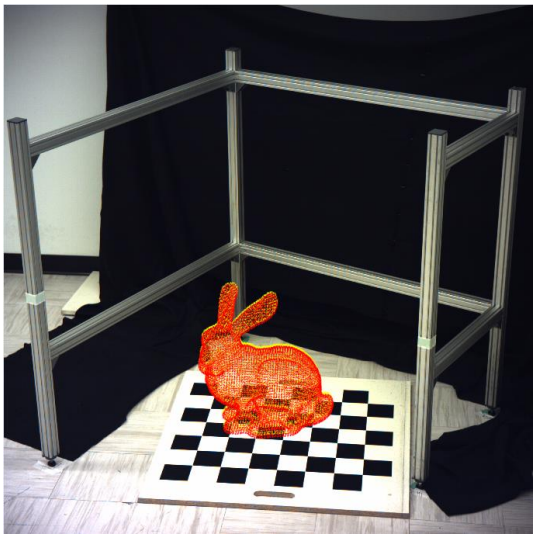
(18)



(19)



(20)



(21)

Figure 22 - Bunny reprojection. Red bunny is reprojected with the estimated pose using 15 to 25 lines, and the yellow bunny is the ground truth reprojection.

Conclusion

In the construction industry, the role of augmented reality is to perform as a platform for visualising the ideas from imagination and paper to concrete and perceptible models. Overlaying a 3D model onto an image requires localizing the camera in its environment. We intended to provide a solution to camera pose estimation in this thesis with two of the common geometrical features visible in architectural structures.

We proceed to solve for camera pose using circles and straight lines by taking into account the relationship between the coordinates of the features in the 2D image and their corresponding positions in the 3D world.

In general, a circle projects as an ellipse on the image plane. By detecting the ellipse in the image plane, the normal vector and the center of the circle in the camera frame (resulting from rotating the ellipse in a way that it becomes a circle) are computed. These two parameters are in fact equal to the normal vector and center of the circle in the world frame after being rotated and translated by the camera's exterior orientation and position. However, the experimental results on pose estimation from circle-ellipse correspondences demonstrate that this approach does not provide reliable solutions to this problem.

Straight lines are detected on the image plane and the correspondences are defined with a vector to indicate the line's direction and a point lying on that line. Since there exists a plane containing both lines in the world and camera frame, passing through the optical center, the pose of the camera is recovered from the fact that the direction vectors and the points on both lines must be perpendicular to the plane's normal. In terms of straight lines, the reprojection error between the estimated and the ground truth pose confirms that this approach is accurate, and its execution time is fast as well. In addition, it is suitable for pose estimation from various number of lines.

Finding correspondences is one of the challenges that requires more investigation. Specifically, an approach that pairs an object with its image automatically. This will be helpful in

real time applications and in video sequences as well. Furthermore, employing features that are more complicated may improve the robustness of the estimation.

Bibliography

[1] D. DeMenthon and L. S. Davis, "Exact and approximate solutions of perspective-three-point problem," *Pattern Analysis and Machine Intelligence*, vol. 14, no. 11, pp. 1100-1105, 1992.

[2] R. M. Haralick, C.-N. Lee, K. Ottenberg and M. Nolle, "Review and analysis of solutions of the three point perspective pose estimation problem," *International Journal of Computer Vision,* vol. 13, no. 3, pp. 334-356, 1994.

[3] R. Horaud, B. Conio and O. Leboulleux, "An Analytic Solution for the Perspective 4-Point Problem," *COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING,* vol. 47, pp. 34-44, 1989.

[4] Z. Y. Hu and F. C. Wu, "A note on the number of solutions of the concoplanar P4P problem," *IEEE Transactions on Pattern Analysis and Machine Interlligence ,* vol. 24, no. 4, pp. 550-555, 2002.

[5] M. Bujnak, Z. Kukelova and T. Pajdla, "A general solution to the P4P problem for camera with unknown focal length," in *IEEE Computer Vision and Pattern Recognition*, Anchorage, AK, 2008.

[6] D. Nistér, "An Efficient Solution to the Five-Point Relative Pose Problem," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE,* vol. 26, no. 6, pp. 756-770, 2004.

[7] J. A. Hesch and S. I. Roumeliotis, "A Direct Least-Squares (DLS) method for PnP," in *International Conference on Computer Vision*, Barcelona, 2011.

[8] Y. Zheng, Y. Kuang, S. Sugimoto, K. Åström and M. Okutomi, "Revisiting the PnP Problem: A Fast, General and Optimal Solution," in *IEEE International Conference on Computer Vision*, Sydney, NSW, 2013.

[9] L. Kneip, H. Li and Y. Seo, "UPnP: An Optimal O(n) Solution to the Absolute Pose Problem with Universal Applicability," in *Computer Vision – ECCV*, 2014.

[10] V. Lepetit, F. Moreno-Noguer and P. Fua, "EPnP: An Accurate O(n) Solution to the PnP Problem," *International Journal of Computer Vision,* 2009, 81:155.

[11] C.-P. Lu, G. D. Hager and E. Mjolsness, "Fast and globally convergent pose estimation from video images," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 22, no. 6, pp. 610-622, 2000.

[12] A. Ansar and K. Daniilidis, "Linear Pose Estimation from Points or Lines," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 25, no. 5, pp. 578-589, 2003.

[13] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America A*, vol. 4, no. 4, pp. 629-642, 1987.

[14] A. Ess, E. Neubeck and L. Van Gool, "Generalised Linear Pose Estimation," in *British Machine Vision Conference*, 2007.

[15] Q. Ji, M. S. Costa, R. M. Haralick and L. G. Shapiro, "A robust linear least-squares estimation of camera exterior orientation using multiple geometric features," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 55, no. 2, pp. 75-93, 2000.

[16] L. Quan and Z. Lan, "Linear N-point camera pose determination," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 8, pp. 774-780, 1999.

[17] S. Li, C. Xu and M. Xie, "A Robust O(n) Solution to the Perspective-n-Point problem," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE,*, vol. 34, no. 7, pp. 1444-1450, 2012.

[18] J. Heikkila and O. Silven, "A four-step camera calibration procedure with implicit image correction," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Juan, 1997.

[19] X. Meng and Z. Hu, "A new easy camera calibration technique based on circular points," *Pattern Recognition*, vol. 36, no. 5, pp. 1155-1164, 2003.

[20] V. Fremont and R. Chellali, "Direct camera calibration using two concentric circles from a single view," in *International Conference on Artificial Reality and Telexistence*, 2002.

[21] Z. Zheng, W. Zhenzhong and Z. Guangjun, "Estimation of projected circle centers from array circles and its application in camera calibration," in *Asia-Pacific Conference on Computational Intelligence and Industrial Applications (PACIIA)*, Wuhan, 2009.

[22] J.-S. Kim, H.-W. Kim and I. S. Kweon, "A Camera Calibration Method using Concentric Circles for Vision Applications," in *Asian Conference on Computer Vision (ACCV)*, Melbourne, 2002.

[23] Q. Chen, H. Wu and T. Wada, "Camera Calibration with Two Arbitrary Coplanar Circles," in *European Conference on Computer Vision (ECCV)*, 2004.

[24] A. Pagani, J. Kohler and D. Stricker, "Circular markers for camera pose estimation," in *Proceedings of the International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, 2011.

[25] Y. Zheng and Y. Liu, "The projective equation of a circle and its application in camera calibration," in *International Conference on Pattern Recognition*, Tampa, FL, 2008.

[26] R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2004.

[27] G. Wang, J. Wu and Z. Ji, "Single view based pose estimation from circle or parallel lines," *Pattern Recognition Letters*, vol. 29, no. 7, pp. 977-985, 2008.

[28] J. Odelstad, "Invariance and Structural Dependence," in *The Concept of Dependence in Applied Mathematics; a First Account*, Springer, 1992.

[29] D. Forsyth, J. L. Mundy, A. Zisserman, C. Coelho, A. Heller and C. Rothwell, "Invariant Descriptors for 3D Object Recognition and Pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 10, pp. 971-991, 1991.

[30] C. A. Rothwell, A. Zisserman, C. I. Marinos, D. A. Forsyth and J. L. Mundy, "Relative motion and pose from arbitrary plane curves," *Image and Vision Computing*, vol. 10, no. 4, pp. 250-262, 1992.

[31] H. Bin, S. Yongrong, Z. Yunfeng, X. Zhi and L. Jianye, "Vision pose estimation from planar dual circles in a single image," *Optik*, vol. 127, no. 10, pp. 4275-4280, 2016.

[32] A. Vakhitov, J. Funke and F. Moreno-Noguer, "Accurate and Linear Time Pose Estimation from Points and Lines," in *European Conference on Computer Vision*, 2016.

[33] B. Pribyl, P. Zerncik and M. Cadik, "Absolute pose estimation from line correspondences using direct linear transformation," *Computer Vision and Image Understanding*, vol. 161, pp. 130-144, 2017.

[34] F. L. Markley and J. L. Crassidis, Fundamentals of Spacecraft Attitude Determination and Control, New York, NY: Springer, 2014.

[35] F. M. Mirzaei and S. I. Roumeliotis, "Globally optimal pose estimation from line correspondences," in *IEEE International Conference on Robotics and Automation*, Shanghai, 2011.

[36] M. D. Shuster, "A servey of attitude representations," *The Journal of the Astronautical Sciences*, vol. 41, no. 4, pp. 439-517, 1993.

[37] H. H. Chen, "Pose determination from line-to-plane correspondences: existence condition and closed-form solutions," in *International Conference on Computer Vision* , Osaka, Japan, 1990.

[38] L. Zhang, C. Xu, K.-M. Lee and R. Koch, "Robust and Efficient Pose Estimation from Line Correspondences," in *Asian Conference on Computer Vision (ACCV) 2012*, Berlin, Heidelberg, 2013.

[39] C. Xu, L. Zhang, L. Cheng and R. Koch, "Pose Estimation from Line Correspondences: A Complete Analysis and a Series of Solutions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1209-1222, 2017.

[40] F. Ababsa and M. Mallem, "Robust camera pose estimation combining 2D/3D points and lines tracking," in *IEEE International Symposium on Industrial Electronics*, Cambridge, UK, 2008.

[41] T. Q. Phong, R. Horaud, A. Yassine and P. D. Tao, "Object pose from 2-D to 3-D point and line correspondences," *International Journal of Computer Vision,* vol. 15, no. 3, pp. 225-243, 1995.

[42] R. Kumar and A. R. Hanson, "Robust Methods for Estimating Pose and a Sensitivity Analysis," *CVGIP: Image Understanding,* vol. 60, no. 3, pp. 313-342, 1994.

[43] S. Ramalingam, S. Bouaziz and P. Strum, "Pose estimation using both points and lines for geo-localization," in *IEEE International Conference on Robotics and Automation*, Shanghai, 2011.

[44] R. Szeliski, Computer Vision, London: Springer-Verlag, 2011.

[45] Y.-x. Yuan, "A review of trust region algorithms for optimization," *ICIAM,* vol. 99, pp. 271-282, 2000.

[46] P.-A. Absil, R. Mahony and R. Sepulchre, Optimization Algorithms on Matrix Manifolds, Princeton, NJ: Princeton University Press, 2008.

[47] R. Halir and J. Flusser, "Numerically stable direct least squares fitting of ellipses," in *Proceedings of the 6th International Conference in Central Europe on Computer Graphics and Visualization*, Plzen, Czech Republic, 1998.

[48] A. Fitzgibbon , M. Pilu and R. B. Fisher, "Direct Least Squares Fitting of Ellipses," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 21, no. 5, pp. 476-480, 1999.

[49] J.-N. Ouellet and P. Hebert, "A Simple Operator for Very Precise Estimation of Ellipses," in *Computer and Robot Vision (CRV)*, Montreal, 2007.

[50] R. Grompone von Gioi, J. Jakubowicz, J.-M. Morel and G. Randall, "LSD: A Fast Line Segment Detector with a False Detection Control," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE,* vol. 32, no. 4, pp. 722-732, 2010.

[51] A. Desolneux, L. Moisan and J.-M. Morel, "Meaningful Alignments," *International Journal of Computer Vision,* vol. 40, pp. 7-23, 2000.

[52] Y. Salaun, R. Marlet and P. Monasse, "Multiscale line segment detector for robust and accurate SfM," in *Proceedings of the 23rd International Conference on Pattern Recognition (ICPR)*, Cancun, Mexico, 2016.

[53] C. Akinlar and C. Topal, "EDLines: A real-time line segment detector with a false detection control," *Pattern Recognition Letters,* vol. 32, pp. 1633-1642, 2011.

[54] C. Topal, C. Akinlar and Y. Genc, "Edge Drawing: A Heuristic Approach to Robust Real-Time Edge Detection," in *International Conference on Pattern Recognition (ICPR)*, Istanbul, Turkey, 2010.

[55] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Kerkyra, 1999.

[56] Y. Zheng and Y. Liu, "Closed-form solution for circle pose estimation using binocular stereo vision," *Electronics Letters,* vol. 44, no. 21, pp. 1246-1247, 2008.

[57] N. H. Abdel-All, H. N. Abd-Ellah, F. M. Hamdoon and M. A. Abd-Rabo, "THREE DIMENSIONAL SURFACES OBTAINED BY THE EQUIFORM MOTION OF A SURFACE OF REVOLUTION," 2015.

[58] Y. Zheng and Y. Liu, "The projective equation of a circle and its application in camera calibration," in *International Conference on Pattern Recognition*, Tampa, FL, 2008.