Design and Calibration of a Network of RGB-D Sensors for Robotic Applications over Large Workspaces

By:

Rizwan Macknojia

A thesis submitted to the

Faculty of Graduate and Postdoctoral Studies

In partial fulfillment of the requirements for the degree of

Master of Applied Science

In Electrical and Computer Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering

School of Electrical Engineering and Computer Science

University of Ottawa

Abstract

This thesis presents an approach for configuring and calibrating a network of RGB-D sensors used to guide a robotic arm to interact with objects that get rapidly modeled in 3D. The system is based on Microsoft Kinect sensors for 3D data acquisition. The work presented here also details an analysis and experimental study of the Kinect's depth sensor capabilities and performance. The study comprises examination of the resolution, quantization error, and random distribution of depth data. In addition, the effects of color and reflectance characteristics of an object are also analyzed. The study examines two versions of Kinect sensors, one dedicated to operate with the Xbox 360 video game console and the more recent Microsoft *Kinect for Windows* version.

The study of the Kinect sensor is extended to the design of a rapid acquisition system dedicated to large workspaces by the linkage of multiple Kinect units to collect 3D data over a large object, such as an automotive vehicle. A customized calibration method for this large workspace is proposed which takes advantage of the rapid 3D measurement technology embedded in the Kinect sensor and provides registration accuracy between local sections of point clouds that is within the range of the depth measurements accuracy permitted by the Kinect technology. The method is developed to calibrate all Kinect units with respect to a reference Kinect. The internal calibration of the sensor in between the color and depth measurements is also performed to optimize the alignment between the modalities. The calibration of the 3D vision system is also extended to formally estimate its configuration with respect to the base of a manipulator robot, therefore allowing for seamless integration between the proposed vision platform and the kinematic control of the robot. The resulting vision-robotic system defines the comprehensive calibration of reference Kinect with the robot. The latter can then be used to interact under visual guidance with large objects, such as vehicles, that are positioned within a significantly enlarged field of view created by the network of RGB-D sensors.

The proposed design and calibration method is validated in a real world scenario where five Kinect sensors operate collaboratively to rapidly and accurately reconstruct a 180 degrees coverage of the surface shape of various types of vehicles from a set of individual acquisitions performed in a semi-controlled environment, that is an underground parking garage. The vehicle geometrical properties generated from the acquired 3D data are compared with the original dimensions of the vehicle.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. Pierre Payeur, for his support and assistance throughout my studies and research. His guidance helped me in writing this thesis and my research papers. I would like to thank my colleague, Dr. Alberto Chavez, for giving productive ideas and assistance.

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

1.1 Motivation

A decade ago, most of the industrial companies were not equipped with vision guided robotic systems. They preferred discrete automation over robotic manufacturing because it was easy to implement and did not require complex programming and massive computing power. Work was distributed in series of interconnected modules, controlled by a programmable logic controller (PLC). Nowadays, the situation is changing rapidly due to the availability of various imaging technologies, range sensors and higher end computing devices. These devices are becoming cheaper and attract manufacturers to integrate them into their operation. Lots of research has been done in the area of image processing, camera calibration, features detection, tracking, and machine learning algorithms. Moreover, faster computing makes it possible to implement and get response in real time.

Machine vision is an automated technology where images are captured and processed for a wide variety of applications such as quality control, inspection, reporting, motion capture, modeling of human gestures and many more. These systems are easy to implement and become increasingly more powerful. Recent advances in machine vision technology and robotics have revolutionized industrial automation by merging both technologies into a single solution. This enlarges the scope of the machine vision market for a wider range of applications in the industrial and non-industrial sectors.

Machine vision with 3D guidance is recognized to provide advantages over 2D vision for numerous robotic applications. However 3D vision is more critical in terms of the quality and the quantity of data that must be processed. System setup plays an important role in data collection, which usually involves a lot of performance testing. In the context of service robotics, the use of 3D sensors not only provides enough information to guide the robot in particular areas, but also supports the analysis and understanding of the shape of objects to enhance the execution of complex tasks.

An essential component of the development of a vision-guided robotic system is calibration. This includes the calibration of every vision sensor, as well as calibration in between the imaging sensors and the robot. The errors that might remain in the calibration between sensors inevitably result in mismatches in the resulting 3D point cloud, which is impacted either by scaling errors (the reconstructed model has different size as compared to the original dimension in the real world) or displacement errors (the location of 3D points do not refer to the same real-world coordinates). Scaling error can result from inaccurate estimation of the value of the focal length during intrinsic calibration. The lack of corresponding points during extrinsic calibration generates displacement errors and misalignment in between separate point clouds. In the process, the selection of the calibrating target to be used for calibration is crucial for achieving accurate correspondence between scenes.

This thesis presents the design and implementation of a vision-guided robotic system for automated and rapid vehicle inspection. However, the developed framework finds application in any RGB-D imaging scenarios over enlarged field of view, like navigation of a mobile robot around objects or public places surveillance. This work focuses on the requirements imposed by security screening of vehicles at checkpoints, which must be performed fairly rapidly but also reliably to ensure the relevance of the operation, the safety of operators and of vehicles passengers. The main objective of this research is to achieve quick acquisition of a 3D model of an automotive vehicle, or large objects, which will then provide accurate enough data to a robot for performing interactions with the surface of the vehicle, or objects, in an autonomous way. In particular, the work must deal with the complexity of the design environment, the selection of vision sensors, the optimal localization and calibration of the sensors to cover a large workspace that can contain a typical automotive vehicle, and the calibration of the whole system with a robotic manipulator, to achieve a fully integrated vision-guided robotic system. To address the requirements of large workspace coverage and rapid acquisition of a model, various 3D sensors were considered. The Microsoft Kinect sensor technology that provides simultaneous color and range imaging (RGB-D) was retained, which involved a study and analysis of its performance in the context of a practical industrial application. This study is therefore also an integral part of the research reported in this thesis.

1.2 Objectives

To achieve the general objective described in the previous section, the work was conducted as a sequence of specific objectives. These include:

- A complete analysis of the Microsoft Kinect sensor technology, under the first two generations, which includes an experimental evaluation of its performance in different scenarios, based on depth quality measurement, actual field of view of the sensor, and its color response;
- The design and implementation of a reconfigurable multi-camera vision system, which is capable of covering a large imaging volume while remaining easy to calibrate;
- The development of a methodology for intrinsic calibration of each sensor, which gives low reprojection error;
- The development of a procedure to accurately and efficiently merge depth and color data, acquired from several viewpoints;
- The design and implementation of an extrinsic calibration method in between pairs of sensors configured in a network, while ensuring fast and easy execution of calibration on the field and with best possible accuracy;
- The design and implementation of a method for calibration between a robotic arm's base reference frame and the multi-camera vision system;
- An experimental testing and validation of the multi-camera vision system in the real-world application considered.

The main contribution of this thesis resides in chapter 5, where an original method developed to calibrate a network of Kinect RGB-D sensors is presented. Given the relatively recent introduction of this sensing technology on the market and the fact that its primary field of application remains computer gaming, there are not yet many techniques available to calibrate a network of Kinect sensors for industrial applications over an extended workspace. The techniques currently available in the literature are limited to smaller workspaces than those of interest in this work, as imposed by the application considered for RGB-D imaging over

automotive size objects, or do not formally estimate all internal and inter-sensor calibration parameters involved.

1.3 Thesis Organization

The thesis organization is as follows. Chapter 2 presents the review of 3D imaging technologies and their implementation in a multi-camera system design. The review also covers calibration techniques of multi-camera systems and the registration methods for 3D data. Chapter 3 presents a complete analysis of the Microsoft Kinect sensors which examines its data quality, the range of operation and the effects of color and reflective characteristics of the object. In chapter 4, the design of a multi-camera system using Kinect sensors is presented. The system designed offers coverage of a large workspace. In addition the specific issues related multi-camera systems are also discussed, including the hardware requirements to connect all sensors to a central computer. In chapter 5, the calibration of the proposed multi-camera system is developed. The latter includes the internal calibration of all Kinect sensors individually, the extrinsic calibration of the sensors over the network, and finally the calibration of the network with the robotic manipulator. The implementation of the multi-camera vision system in a practical application for automotive vehicle 3D imaging is presented in Chapter 6. Finally, chapter 7 provides the conclusions and explores future work.

Chapter 2. Literature Review

Industrial robots usually work without any embedded sensors. They depend on the known position and orientation of an object for performing a task. However, integrating 3D imaging around a robot can be extremely useful to enhance robot navigation and create more autonomous systems. A lot of research has been conducted in this direction and several solutions have been proposed. This chapter presents a review of the latest advances in the field of 3D imaging technologies and examines methods that can be used to calibrate a network of RGB-D sensors. The last section describes point clouds registration methods that can be used to further refine the overall calibration accuracy.

2.1 3D Imaging Technologies

3D imaging technologies measure the distance to some objects rather than the light intensity reflected or diffused over their surface. The output of the sensor is known as a range image, where the value of each pixel represents a distance between a frame of reference and a surface point on the scene. Some sensors provide RGB-D (color + range) data, where the color associated with each pixel in a range image is also measured, along with the distance. However, most sensors do not provide the complete 3D model of a scene but rather a single side of the 3D objects, which actually makes them 2.5D sensors. Many different technologies are available that provide 3D (or 2.5D) information, but each comes with its own limitations, advantages and related cost. In principle, there are two main types of sensors to acquire 3D data; passive range sensors and active range sensors.

2.1.1 Passive Range Sensors

Passive range sensors rely on intensity images to reconstruct a representation of the surface. Structure from motion (SfM) and stereovision systems work on this principle. A stereovision system consists of two cameras, while a SfM system relies only on a single camera. These systems find the correspondences between two separate images taken simultaneously using stereo cameras [1, 2], or taken by a single camera at different times or places [3]. Hartley and Zisserman [1], define the procedure to calculate the 3D sparse structure of the scene by

using a linear triangulation, provided that the features of correspondence between two images and a camera matrix are given. It is not always the case that every pixel in the first image has a corresponding point in the other image. Therefore, passive range sensors often provide only sparse 3D information. The process of finding all corresponding points into another image is also challenging and can take a long time. Therefore, epipolar geometry [1] defines the relation between 3D points and their 2D projections on the image plane, when captured by two cameras at different viewpoints. This relationship reduces the search for corresponding points from the 2D image plane to a 1D line in an image plane. Another problem with stereovision comes from occlusions, that occur when a point in the 3D space in front of the cameras gets depicted in one of the images but is blocked from being depicted in the other one because not in direct sight of the second camera. As such, stereovision introduces physical restrictions because of the camera separation.

One well known commercial example of a stereovision system is developed by PointGrey Research, the Bumblebee stereo vision camera. It provides the complete hardware and software to process pairs of images and estimate sparse 3D point clouds with color mapping. Figure 2.1 shows the Bumblebee stereovision camera system.

Figure 2.1 : Bumblebee®2 Point Grey Research's IEEE-1394 (FireWire) Stereo Vision camera system.

On the other hand, structure from motion techniques requires a single camera to be moved between two positions to capture the images. However, the SfM technique is better suited for static scenes where objects are not moving. As images are taken at different times and positions, the location of moving objects becomes uncertain with the SfM technique. Both approaches typically produce a sparse and unevenly distributed datasets.

2.1.2 Active Range Sensors

Active range scanners emit some form of energy toward the surface to be imaged and detect its reflection in order to measure the distance. The types of radiation used include light, laser, sound, X-Rays and electromagnetic waves. These sensors can be further divided into two categories, those that project light over the surface to generate a pattern or a dot and then use triangulation in order to calculate the disparity; and others that project radiation toward the surface and calculate the time of flight for the wave to go and come back to the sensor.

2.1.2.1 Time-of-Flight (ToF) Sensors

A time-of-flight (ToF) camera measures the distance based on the known speed of light or sound. When using light (Lidar technology), the camera generates particular frequencies of light and calculate the amount of time this beam energy takes to reach the surface of an object and return, in part, to the sensor. Since the speed of light, c, is known for different media (e.g. air, empty space), the roundtrip time corresponds to twice the distance between the scanner and the object. If t is the roundtrip time, then the distance is equal to $(c * t)/2$. Some ToF technologies are based on phase measurement of the modulated light and are implemented into a standard CMOS or CCD technology [4, 5]. The other approach is using optical shutter technology to capture the light, an approach that has been used for studio camera initially [6] and then developed into miniaturized cameras such as the Zcam [7]. ToF sensors are largely independent of the scene texture and full frame real-time depth estimates are possible. Unfortunately, the data is noticeably contaminated with random and systematic measurement errors. In addition, the X-Y resolution of the sensors is often limited to 320x240 pixels or fewer, which is far below the resolution of modern cameras [8].

2.1.2.2 Active Triangulation Sensors

This sensor technology measures the distance of a target by illuminating the target with a light source, typically laser light. The laser shines on the surface of the object and the camera identifies the location of the corresponding laser dot [9], often through frequency band filtering. The projection of the laser on the image plane appears at a different location,

depending on how far the laser dot appears. The laser dot, T, the camera center of projection, O, and the laser emitter, L, form a triangle as shown in Figure 2.2. The distance OL between the camera and the laser emitter, and the angle $< TLO$ at the laser projector are known. The angle $< LOT$ is determined by projection of the laser reflection on the image plane. This projection appears at different locations depending on the distance of the object. This information is then used to determine the location of the laser dot over an object at a certain distance. If the angle $< TLO$ remains constant and object *1* moves toward object *2* in a straight line, then the projection of the object also move towards the optical center in a straight line. The difference, dz, in the image plane increases the angle $< LOT$, forming a new triangle. This triangle allows determine the new distance of the object which is $LT + Dz$.

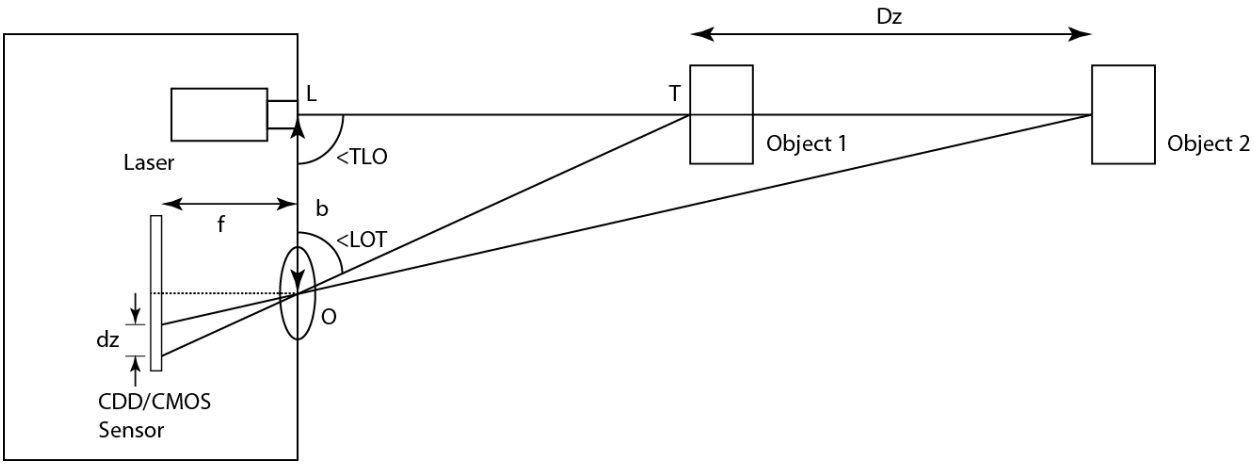


Figure 2.2 : Principle of active triangulation illustrated over two object distances.

In some devices the laser can scan the object by moving the laser dot over the entire field of view of the sensor. The triangulation is sequentially performed on every point to reconstruct the object. The Jupiter range sensor developed by National Research Council of Canada (NRC) is an example of an active triangulation based device. This device scans the line by swapping a laser point over a line using an oscillating mirror. The scanning process of active triangulation usually provides far better accuracy and resolution than any other 3D scanning technology. However, the major disadvantages of active triangulation scanners are the use of mechanical components, the fact that they do not deliver 3D range data over a full image map in a single capture, and their high sensitivity to surface texture and reflectance characteristics.

Furthermore, most of them are unable to provide the color information on the object along with its surface shape.

In some sensors a laser stripe is used instead of a single laser dot to scan the object. Such line scanners represent an extension to the single laser point range sensor and allow to speed up the acquisition process. A line scanner is very helpful for profiling an object. The deformation in the scan line explicitly provides the range information. However, it does not provide a detailed reconstruction of each point in a line because there is no means to differentiate in between points that are highlighted within the same linear laser projection. The Vivid 910 system [10] developed by Konica Minolta and ShapeGrabber systems [11] developed by ShapeGrabber Inc. are examples of line scanner devices. These devices scan the object by moving the laser stripe on the object surface. They are generally used in modeling complex objects, inspection and reverse engineering processes. The major advantage of line scan profilers is the scanning speed as compared to single laser dot scanners but still they do not provide real-time depth measurements.

2.1.2.3 Structured Light Sensors

Structured light sensors adopt a different approach that removes the need for scanning a beam of energy over the entire surface. Structured light sensors rather project a 2D pattern over the entire surface of the scene. A camera captures an image of the surface over which a predefined visual pattern is artificially created and the deformation of the imaged pattern, which varies with the shape and distance of the object, is analyzed. An algorithm is used to calculate the distance at each point created by the pattern. The patterns are selected to be easily detectable. These patterns are either projected in multiple shots of different sequences or as a single shot with a complete pattern.

Valkenburg and McIvor [12] used the sequence of binary coded black and white stripes as a pattern. These patterns are projected in sequence on the object surface. Figure 2.3 shows a 5 bit projection pattern. These patterns generate $2^5 = 32$ unique binary coded areas on the object. The 3D coordinates of 32 binary coded areas along each horizontal line are measured on

the basis of the triangulation principle. The spatial resolution depends on the number of binary patterns used to ensure that each pixel is coded with a unique binary number. The sequential projection is only applicable on a static environment and requires time for generating the sequence of binary patterns.

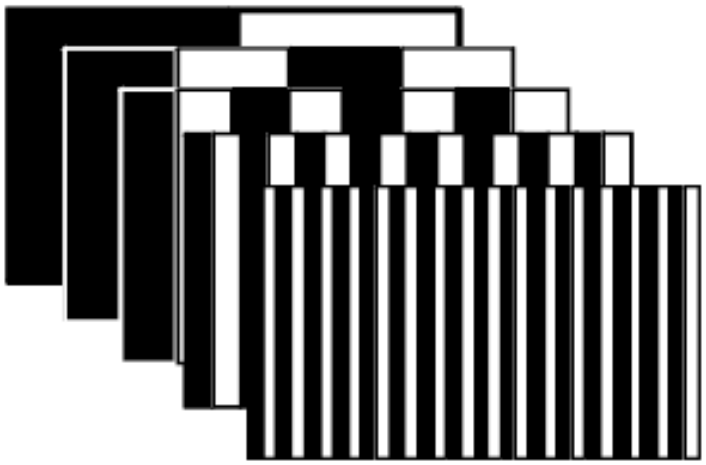


Figure 2.3 : 5 bit binary coded pattern.

The binary coding only supports two levels of intensity and requires a large number of patterns to encode the object surface. This problem is overcome by introducing gray level patterns [13] which add different intensities of gray color in the pattern. These additional levels increase the number of unique codes and require fewer patterns to be projected. This greatly increases the system speed but is still not fast enough for real-time depth measurement. Moreover the system must distinguish between two adjacent levels of gray for reliable measurement, which is prone to errors.

Boyer and Kak [14] introduced the color encoded stripes to illuminate the object with only one encoded pattern. The color representation can provide 2^{24} different intensities for three 8 bit (RGB) channels. This range can be utilized to create a large number of sets in which each color has a maximum distance from any other similar color. This encoding removes the cross talk between two adjacent colors during detection. Durdle *et al.* [15] determined the correspondence between two images in a stereo camera setup by projecting a pattern that consists of repeated group of gray level intensities (black, gray and white). The group contains

the combination of (black, gray and white) intensities as, BWG, WBG, WGB, GWB, GBW, and BGW. The matching process first locates the group and then identifies the sub gray level pattern such as BGW.

2D grid pattern techniques further refine the concept of a single indexing strip projection. Each part of a 2D grid pattern is uniquely identifiable in the 2D space unlike the single dimension indexing scheme where every horizontal line shares the same pattern. These techniques overcome the correspondence problem in the stereoscopic images feature matching by projecting artificial features over the object. Payeur and Desjardins [16] used a bi-dimensional colored pseudo-random pattern for creating such artificial features. Each artificially created feature point is uniquely defined to increase reliability of correspondence between pair of points. The system used the pair of stereo cameras to detect the pattern projected by a projector to avoid the calibration between camera and a projector. Freedman *et al.* [17] present the active light system similar to stereoscopic vision but with one camera replaced by the projector. Unlike Payeur and Desjardins' [16] system, the calibration between the projector and the camera is required. The random pattern is projected on a surface and the deformation created by the objects surface is observed by a camera. The stereo matching processes is performed between the known reference pattern projected on a surface and the observed pattern to estimate the depth.

Some structured light sensors are capable to provide real-time depth measurements, which reduces the problem of distortion from motion. Some of the existing systems are capable of scanning moving objects in real time, such as Microsoft Kinect and Asus Xtion Pro. These sensors project a fixed 2D infrared pattern on the object and an IR camera present in the system captures the pattern to produce a depth image. The Kinect sensor contains one color (RGB) camera, one infrared (IR) camera and an infrared pattern optical projection mechanism. The Asus Xtion Pro is similar to the Kinect sensor, except that it does not include an RGB camera. The Kinect sensor is capable of collecting or estimating depth information for each pixel in a color image in real time, which opens the door to a great variety of applications. Kinect provides depth data with a resolution of 640x480 at 30 fps, which is better than that of

many ToF cameras. In spite of its recent introduction, a lot of research work has already been conducted with the Kinect sensor to investigate the technology behind the device, analyze its properties, performance, and perform comparison with other structured lighting devices. This has been motivated by the fact that originally the sensor was launched as a peripheral for the Xbox 360™ video game console and no information about the sensor technology and performance was provided by the manufacturer. Due to the popularity that the device earned for 3D imaging research, Microsoft introduced a commercial version of the sensor named *Kinect for Windows*, in an attempt to address the needs of the research community. This thesis also focuses, in part, on the Kinect sensor's technology and examines its operating range, data quality and its limitations for industrial and robotic 3D imaging applications. The characteristics of the device are more extensively discussed in Chapter 3.

2.2 Multi-Sensor System Design

The goal of this research project is the development of a 3D vision system for vehicle screening. The 3D data collected is used as an input for a robotic path planner to move a manipulator robotic arm in close proximity of a vehicle to perform some inspection tasks. The design of an adequate multi-camera system is therefore important because it can impact on the quality and completeness of the data. During system design a number of components must be considered. The system must consist of a minimum number of range sensors to ensure complete data acquisition over a volume large enough to contain a typical vehicle. This brings important considerations in terms of speed and workload on the system to process the data. The sensors must also be positioned in a way that eases the calibration process between all sensors. An accurate calibration is essential to achieve registration between the point clouds individually collected by every sensor. Furthermore the object must lie between the acceptable ranges of the sensor to ensure the quality of the data.

The systems presented by Cheung *et al.* [18] and Doubek *et al.* [19] are examples of vision systems that operate over large volumes and in real-time. The processing is distributed over several computer nodes. Each computer node captures a video frame with its respective camera and extracts a silhouette. The latter is sent to a master computer to perform 3D data

reconstruction. Parallel computing is used to increase the overall system speed. Parallel computing requires good synchronization between frames and is prone to inaccuracy due to communication latencies between the multiple nodes.

The above system uses color cameras which add to the workload on the computer to extract and process silhouettes for reconstruction. Tong e*t al.* [20] use the Microsoft Kinect sensor, which directly provides color and depth data, and reduces the complexity of the reconstruction process from the computer perspective. The system is designed to perform 3D scanning of full human body. They use three Kinect sensors. Two of them are positioned to capture the top and bottom part of the body from one direction and the third sensor is placed in the opposite direction to capture the middle part. These arrangements avoid overlapping regions between sensors, which may cause noise in the depth reading due to Kinect sensor's IR projector. A turntable is placed in the center of the system, which rotates the human while the system captures the 3D data in each frame. This provides a compromise between the use of few sensors for complete coverage of the subject while increasing the acquisition speed. Their system takes about 6 min to generate a complete model of the human body.

Maimone and Fuchs [21] present a real-time telepresence system. They also use Kinect sensors for capturing the human body. The major difference between Tong e*t al.'s* system [20] and this one is the use of a larger number of sensors to cover the complete field of view around the body, such that there is no need to rotate the subject. This brings the scanning to be performed in real time but also introduces interference problems between the overlapping regions of the sensors. The depth generated in the overlapping regions contains holes with no depth information. They fix this problem by filling the holes [22] while making the assumption that they are part of a continuous surface. They perform the experiments using five Kinect sensors. Four of them are surrounding the scanning area and one is placed on top. The system scans in real time and also performs eye tracking but with reduced speed using a single computer. A hole filling approach works fine between the surface areas but generates irregular boundaries on the subject, which cannot be corrected with this technique.

2.3 Camera calibration

Camera calibration is an important step in the development of many computer vision systems. Complete calibration includes the estimation of internal parameters of the cameras and the determination of their respective position and orientation with respect to a global reference frame, often associated with one of the camera. The latter stage is extremely important in multi-camera systems, where knowledge about the exact camera position and orientation is required to generate a reliable and accurate 3D model of a scene. In this section an in depth look into the camera calibration process is proposed, followed by a discussion about some of the modifications that other researchers have proposed to calibrate Microsoft Kinect sensors. Finally, extrinsic calibration methods are discussed.

2.3.1 Intrinsic Parameters

Intrinsic camera parameters describe the relationship between a 3D world point and its 2D projection on the image plane. This relationship is defined by a pinhole camera model and based on a focal length, f (the distance between the center of projection and the image plane), and a principal point, O (the intersection of the image plane and the optical axis), as shown in Figure 2.4.

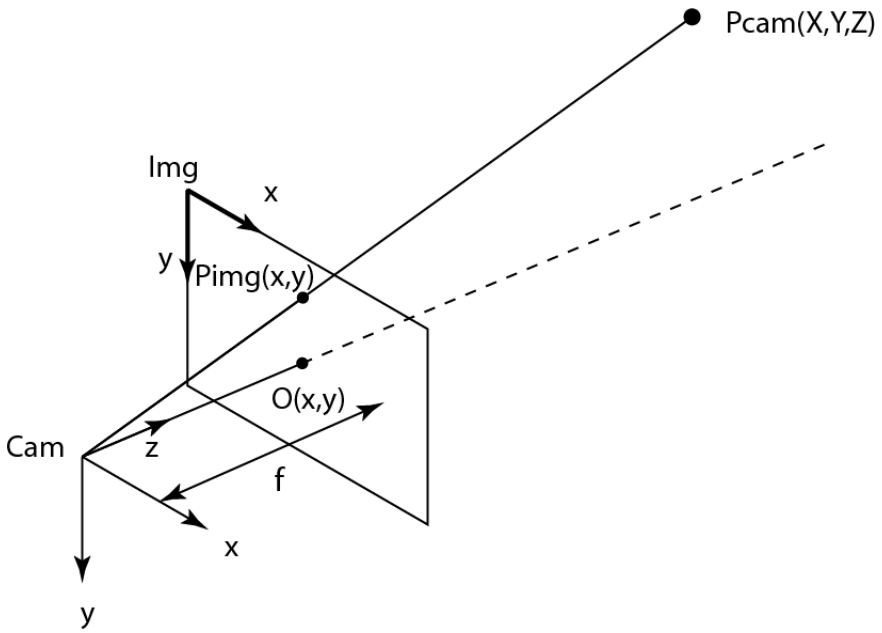


Figure 2.4 : Pinhole camera model.

Cam is the camera reference frame where the location of an object is defined in terms of real world units. *Img* is the image reference frame where the location of a pixel is defined in terms of pixel units. *Pcam* is the point defined in the world with respect to the camera frame, *Cam*. *Pimg* is the location of the point on the image plane with respect to the image frame. Tsai [23] defined the pinhole camera model, k, which relates all 3D points, *Pcam*, with respect to the camera coordinate in real world units, and their 2D projections, *Pimg*, on the image plane in pixel units. This relationship is defined in Equation (2.1) as a 3x3 matrix.

$$k = \begin{bmatrix} \frac{f}{s_x} & 0 & O_x \\ 0 & \frac{f}{s_y} & O_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

Here, f is the lens focal length, (s_x, s_y) are the size of a pixel, and (O_x, O_y) is the optical center on the image plane.

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = k \begin{bmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \end{bmatrix} \tag{2.2}$$

$$P_{img} = \begin{bmatrix} x_{img} \\ y_{img} \end{bmatrix} = \begin{bmatrix} \frac{x_1}{x_3} \\ \frac{x_1}{x_3} \end{bmatrix} \tag{2.3}$$

This simple model usually fails to establish perfect correspondence between the 2D image plane and the 3D world coordinates due to some distortions contributed by the lens (radial distortion) and/or misalignment of the optical axis (tangential distortion). These distortions parameters convert the perspective projection relationship into a nonlinear form. Tsai [23] further introduced calibration methods to compensate for radial distortion. Zhang [24], and Heikkila and Silven [25], extended the Tsai model [23] to deal with radial and tangential distortions. Zhang [24] also introduced the skew coefficient defining the angle between the x and y pixel axes. Bouguet [26] reported that the skew is almost zero and does not affect the camera model as most cameras currently manufactured do not have centering imperfections.

2.3.2 Extrinsic Parameters

The relationship between the world reference frame and the camera coordinate system is defined by extrinsic parameters. The exact location of the camera is usually defined from the optical center of the camera which is physically inside the camera and not exactly known. As such, the location of any point with respect to the camera frame is very difficult to define. Therefore the point is first defined in the world coordinate system and then transformed to the camera frame. The transformation between world coordinates and the camera coordinate system is defined using Equation (2.4)

$$P_{cam} = R_{3x3}.P_{world} + T_{3x1} \tag{2.4}$$

where R is a rotation matrix of size 3x3 and T is a translation vector of size 3x1. A point in the world coordinate system, P_{world}, is first transformed into the camera coordinate system, P_{cam}, using the equation above and then the relationship between this projected point and its pixel coordinates, P_{img}, is defined by Equations (2.1) to (2.3).

2.3.3 Calibration Methods

Camera calibration requires a number of points in the real world to be defined with respect to the world coordinate system and the location of their respective projections on the image plane. Assuming known correspondence between every pair of points (world coordinate system to image plane coordinate system) permits to solve for the intrinsic and extrinsic parameters. A point in the real world is usually extracted from a predefined target. The target points can be coplanar or non-coplanar. Non-coplanar target points result in more accurate calibration than with a coplanar points distribution due to the lack of correspondence between the points over the target itself [23]. However non-coplanar targets, i.e. 3D calibration objects, are more complex to design. A planar target with coplanar points is easy to create but usually requires more than one view to generate a sufficiently large number of points to perform robust calibration. Tsai [23] and Zhang [24] used a planar target with disconnected black squares over a white background. Corners of the square pattern are easily detectable as calibration points. However the disconnected square pattern introduced localization error due

to the lenses as the corners are not always sharp enough to get an accurate pixel location. Heikkila and Silven [25] used a circular pattern instead and considered the center of a circle as a target point, which can be detected with sub-pixel accuracy. Lucchese and Mitra [27] used a regular checkerboard pattern and proposed a method to extract the corners with sub-pixel accuracy.

The Microsoft Kinect depth sensor embeds a color and an IR camera in the same device. The device is capable of providing color, IR and depth images. In order to combine color and depth data accurately, proper calibration is required between the two sensors within a Kinect. Various approaches have been proposed for simultaneous calibration of Kinect sensors. The color camera can be calibrated easily while more challenges remain in calibrating the depth sensor. There are two approaches that can be considered to calibrate the depth sensor, that is, either to calibrate it using the depth image or to calibrate it using the IR image. The depth image is generated from the IR image, therefore calibrating with the depth image is the same as with the IR image. Burrus [28] proposes to use traditional techniques for calibrating the Kinect color camera, and involves manual selection of the four corners of a checkerboard for calibrating the depth sensor. The black and white squares on a checkerboard calibration target cannot be distinguished in the depth image because they lie on the same plane, therefore manual selection is necessary. However, the error introduced in selecting the points leads to inaccurate results. Zhang *et al*. [29] automatically samples the planner target to collect the point for calibration of depth sensor and used manual selection of corresponding points between color and depth images for establishing extrinsic relationship. Gaffney [30] describes a technique to calibrate the depth sensor by using 3D printouts of cuboids to generate different levels in the depth image. These levels can be easily identified in the depth image from their different depth measurements. However this solution requires an elaborate process to construct the target. Berger *et al*. [31] use a checkerboard where black boxes are replaced with mirroring aluminum foil. The Kinect sensor does not generate depth data on the reflective surface patches because they tend to disperse the IR pattern in different directions, resulting in the sensor not being able to estimate the depth at those points. The mirroring aluminum foil appears as black regions (no data) in the depth image which makes the checkerboard target

visible to the IR camera. However for this procedure to work properly, care must be taken for the Kinect not to be parallel to the target during calibration.

Another way to calibrate the depth sensor consists of using the IR image. But the IR camera captures the IR projected pattern mapped over the surface to calculate the depth. This pattern is present in every IR image which makes it difficult to calibrate based on the IR image when the IR projector is on. A compromise consists of blocking the IR projector during the calibration by placing a mask in front of the projector [32]. However, the IR camera only detects IR intensities. Therefore, if there is not enough IR illumination present in the environment, an external IR source is required to illuminate the checkerboard such that its features can be detected. The benefits of this approach lies in the fact that there is no need to create an elaborate calibration target.

2.3.4 Multiple Cameras Calibration

The case where multiple cameras are used and must be calibrated all together can leverage the existing approaches for single camera calibration, provided that two camera views share substantial enough overlap. Many approaches rely on a planar calibration target of an appropriate size, which fits between two camera views, and calibrate the network of cameras in pairs. Zhang's calibration method [24] which is based on the correspondence between 2D image and 3D real-world points can be applied to estimate the pose of the planar target with respect to one camera. If two cameras are seeing the same target then the spatial relationship between the two cameras can be established mathematically. Hartley and Zisserman [1] derived a technique to estimate the extrinsic relationship between two cameras using epipolar geometry and prior knowledge of the cameras intrinsic parameters. This technique consists of decomposing a fundamental matrix into a stereo rotation matrix and translation vector. The main advantage of this technique is that the fundamental matrix between two views is easy to compute as it only requires image correspondences rather than matches between 3D world points and 2D image points. The selected features are typically corner points, such as Harris corners [33] and scale-invariant feature transform (SIFT) points [34]. They are matched by local descriptors which characterize texture or shape of their neighborhoods to establish

correspondences. The matching needs to be robust to variations of viewpoints and lighting between camera views. The automatically obtained pairwise correspondences between feature points may include a significant amount of false matches. RANSAC [35] is used to find the homography that brings the largest number of feature points into match.

All these approaches rely on coplanar points. Some authors also proposed methods to overcome the coplanarity constraint. Rander [36] proposed to use calibration bars mounted on tripods. These bars are translated vertically and horizontally in order to obtain non-coplanar points. This approach requires a lot of efforts to gather points. Drouin *et al*. [37] use a planar target and collect points in multiple views rather than a single view. Svoboda *et al*. [38] suggested to use a single point target. The target is comprised of a single point light source, like a LED, mounted on a bar. The target is waved within the multi-camera system field of view to collect non-coplanar points. A large number of points are collected and false matches are filtered out by RANSAC [35]. They also perform bundle adjustment [39] for final optimization. Bériault [40] calibrates a network of 8 color cameras by waving a LED stick within the network of cameras. Each location of the LED is recorded with timestamps and matched between different pairs of cameras for extrinsic calibration. The approach further solves the global scale factor estimation within the network of cameras. The recorded points using a single LED do not provide an absolute scale factor, therefore a pair of LEDs is eventually used where the distance between two LEDs is known and used to define the absolute scale factor. All the pairs are then rescaled for the global scale factor with respect to the first pair.

2.4 Registration

Registration is the process of combining several datasets into a global coordinate system. The registration of point clouds extracted from range images is a complex task. The calibration of multiple cameras provides estimates of the relative position and orientation between cameras, but range images registered only on the basis of experimentally estimated extrinsic calibration parameters tend to still contain inaccuracies. Range image registration methods provide a way to improve the alignment between 3D point clouds, and therefore to further

refine calibration parameter estimates. Several methods are available to perform registration between range images.

The most classical registration technique used in the literature is an iterative approach. Besl and McKay [41] introduced the iterative closest point (ICP) algorithm, which describes a method for registering a set of 3D target points with a set of 3D reference points. The method calculates the closest points in a target and reference set and iteratively estimates the best transformation for registering those two groups of points. The method operates in cycles, where every iteration consists of estimating a transformation and calculating the remaining mean square error between the target and the reference points. The mean square error is compared against a predefined threshold to stop the process. The ICP algorithm requires an initial estimate that relates the two datasets. This requirement is necessary for the convergence of the algorithm towards a global minimum, although the iterative solution tends to converge to local minima, especially in the presence of point correspondence error. The ICP algorithm is often adequate to provide an estimate of the registration parameters but heavily depends on a close enough initial estimation. The algorithm also tends to introduce false matches, as it depends only on a distance constraint to determine the positions of the correspondences.

A number of techniques have been proposed to improve over the original ICP approach, by eliminating false matches among other things. Rodrigues and Liu [42] improved the point correspondence by generating an error histogram which removes false mappings. Masuda and Yokoya [43] integrated random sampling of a dataset with the ICP algorithm and defined the threshold as residuals of a least median square. This reduces the outliers in the dataset and improves the ICP algorithm to converge toward a global minimum solution. Benjemaa and Schmitt [44] describe a method for global registration of several overlapping 3D surfaces. The overlapping regions are segmented into the optimized set of z-buffers. This technique arranges data in multiple z-buffers and allows to apply 2D image processing techniques on 3D data. This technique rapidly searches the nearest correspondence point between overlapping surfaces.

2.5 Summary

This chapter presented a review of 3D imaging technologies used in the market with an introduction to the Microsoft Kinect sensor. The use of imaging technology in multi-camera system design with color cameras and Kinect sensors was also examined. The review was extended to the calibration of such multi-camera systems, along with a discussion about intrinsic and extrinsic calibration methods and, more specifically, about calibration strategies that have been dedicated to Kinect sensors. Finally, optimization techniques to improve the extrinsic calibration estimates were presented.

Chapter 3. Imaging Technology Selection and Experimental Study of Kinect Sensors

3.1 Introduction

In the previous chapter the state-of-the-art in the field of 3D imaging was presented. The focus was on active and passive devices which are non-contact based 3D measurement techniques. Particular attention was given to active devices which are most popular on the market. The focus of this thesis being to provide 3D measurements over a large workspace for robotic applications, the reconstruction process should therefore be fast and relatively accurate, such that a robot can quickly start working over an object modeled with sufficient details. In order to ensure coherence between sensing units in the vision stage, all 3D imaging devices should be positioned at fixed locations to reconstruct a large size object, such as an automotive vehicle. Among passive sensor approaches available, structure-from motion (SfM) is not suitable because it requires a camera to be moved to reconstruct a given scene. On the other hand, stereo cameras remove this restriction by imaging an object from two different perspectives and generating 3D data without moving the device. A stereo camera like the Bumblebee2 can provide depth data with a spatial resolution of 640x480 at 48 frames per second (fps), which is fast enough, but the quality of the depth data largely depends on the structure of the scene. In addition, the resolution of the range data tends to significantly degrade for larger distances. In that, the Bumblebee2 camera only provides 30 disparity levels between 0.5 and 1m of range, which is not suitable to cover large workspaces.

ToF sensors are largely independent of the scene texture and can provide real-time depth estimation but the data is often contaminated with random and systematic measurement errors. In addition the spatial resolution of the sensors is usually limited to 320x240 pixels which is not suitable for guidance of a robotic application.

Single point laser scanners provide far better accuracy and resolution than any other 3D scanning devices, but also take a large amount of time to generate the 3D point cloud. This type of sensor is suitable in industries to generate precise models of parts but hardly find their place

for real-time 3D imaging. Line scanners can achieve higher acquisition speed than single point scanners but still cannot provide real-time depth data over a large surface.

2D structured light range sensors project a pattern on the surface of the object to create artificial feature points. These sensors can provide a 3D point cloud of an entire surface in a single capture and therefore generate depth data in real-time. While their depth accuracy is not comparable with that of single point or line based laser scanners, these devices overcome the correspondence problem of stereoscopic imaging by projecting a known map of artificial features on the object which usually lead to better depth resolution than stereovision and avoid surface texture dependencies.

The final 3D data is used by the robot as general guidance for navigation and interaction with the vehicle's surface to perform the task. While the acquisition of the surface shape that can be achieved with the Kinect technology is not a priori accurate enough to drive fine and precise interaction of the robot with the vehicle, in the current context of application, the precision of the robotic system is meant to be further enhanced by embedding proximity and touch sensing devices on the end effector of the robot. As a result, a compromise between speed of acquisition and depth accuracy can be made.

The Kinect sensor works on the principle of a structured light range sensor and has the potential to be used in a robotic inspection station for large volume. The extreme acquisition speed of the Kinect technology with the medium quality depth data that it can generate has been the major selection criteria for this sensor to be used for efficiently acquiring 3D data over large volumes, such as that of automotive vehicles. Moreover, the projected pattern is in the infrared spectrum and remains invisible to the human eye. The quality of the Kinect depth data is good enough for the general guidance of robot navigation towards the vehicle surface, given that the robot is equipped with proximity and touch sensing devices for fine and precise interaction. The rest of the chapter examines the mathematical model and quality of depth data provided by the *Kinect for Xbox 360* sensor and the more recently introduced Microsoft *Kinect for Windows* version. This formal comparison was performed to evaluate the suitability

to replace the original *Kinect for Xbox 360* with the new Microsoft *Kinect for Windows* in the experimental implementation.

The analysis of the two Kinect sensor versions is subdivided into four main sections. The first section discusses the major components present in the Kinect sensor. In the second section, the operation of the depth sensor is discussed and a mathematical model is provided to convert the depth data into real world measurement units. The third section examines the theoretical depth resolution and quantization step of the depth data. The last section provides the experimental study of the depth sensor and compares it with the theoretical depth resolution and quantization error. Moreover, the sensitivity to object's color and reflectance characteristics is also analyzed.

3.2 Kinect Sensor Components

In 2010, Microsoft introduced the *Kinect for Xbox 360* sensor as an affordable and real-time source for medium quality textured 3D data dedicated to gesture detection and recognition in its *Xbox 360* game controller. The sensor consists of one color camera, one infrared camera, one infrared projector, a motorized tilt mechanism, a multi-array microphone to provide interaction on voice commands, and an accelerometer. Figure 3.1 shows the Kinect sensor and the location of its main components.

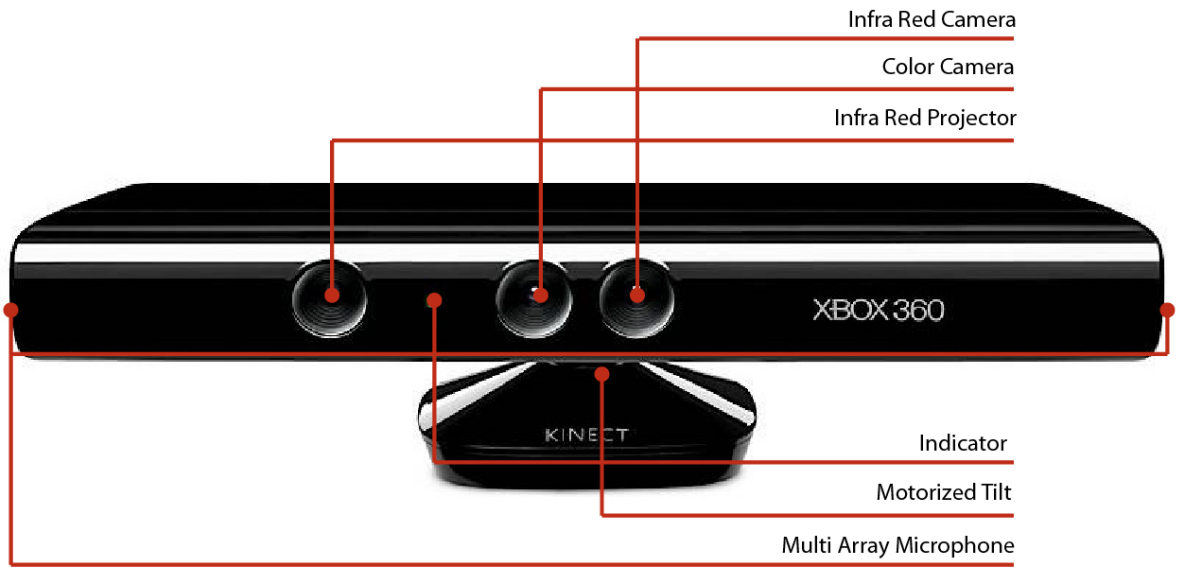


Figure 3.1 : Microsoft Kinect sensor and its components.

3.2.1 Color Image

The color camera present in the Kinect device has a resolution of 1280x1024 pixels with the horizontal field of view covering 63 degrees and the vertical field of view 50 degrees. The Kinect device is capable to stream color images at a framerate of 10 frames/sec (fps) with the maximum resolution of 1280x1024. The frame rate goes up to 30 fps with a resolution of 640x480. The output color image is composed of three 8-bit channels, red (R), green (G), and blue (B).

3.2.2 Infrared Image

The IR camera has the same resolution as the color camera that is 1280x1024 pixels. However, the horizontal field of view is reduced to 57 degrees and the vertical field of view to 43 degrees. The color camera images over a larger surface than the IR camera at any given depth. The output of the IR camera is a 10-bit single channel image. This camera is dedicated to detect the IR projected pattern for the estimation of depth (discussed in section 3.3). The infrared projector illuminates the scene with the help of a 830 nm laser diode. This wavelength is out of the visible spectrum of the human eye, and as a result the pattern created over the scene is not readily visible. The IR camera includes an optical filter with minimal sensitivity for wavelengths that differ from that of the laser diode, resulting in a crisp pattern of the projection over the IR image plane.

3.2.3 Depth Image

The depth image is a representation of a scene where the intensity values of the pixels are replaced by the depth of the closest surface mapped to a given pixel with respect to the camera optical center. The maximum depth image resolution supported by the Kinect device is 640x480 pixels. The field of view in the depth image is the same as that of the IR camera because the depth map is generated from the IR camera along with the infrared projector. The Kinect outputs 11-bit disparity values for each pixel, which are mapped in the depth image.

3.3 Operation of the Kinect Depth Sensor

The Kinect sensor operates on the principle of structured light range sensors. The pattern is predefined and fixed. It is generated by projecting optical radiation (830 nm laser diode) through a transparency micro-grid containing the engraved pattern [17]. The infrared camera contained in the Kinect captures the reflected light pattern on the surface of the objects and compares it against the predefined reference pattern [17]. Figure 3.2 shows the reference pattern created on a flat surface, as captured by the Kinect's IR camera. As can be noticed, the intensity of the IR radiation is greater in the center of the structured light map created on the surface of an object when compared to the peripheral areas due to the non-uniform emission of the laser diode in all directions.

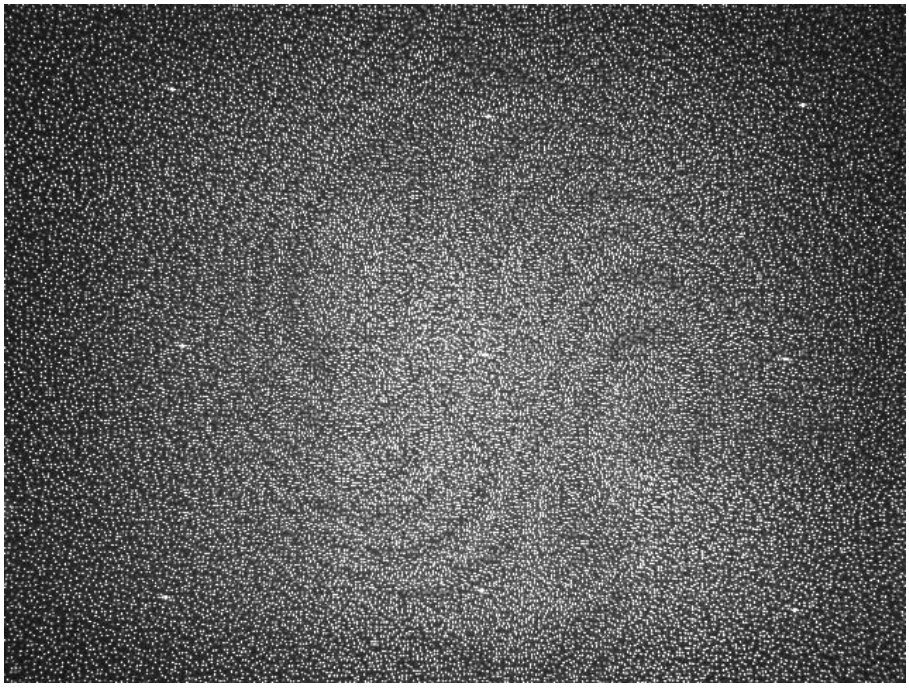


Figure 3.2 : Structured light pattern projected on an object.

The depth calculation is performed through a triangulation process illustrated in Figure 3.3. The depth, Z, of a point, P, on an object is expressed from the optical center of the infrared camera. The z axis of the IR camera is orthogonal to the image plane and passes through the optical center in the direction of the object. The IR projector location is aligned with the x axis of the IR camera. The distance between the optical center of the IR camera and that of the IR

projector is defined by the baseline, b. The reference plane in the schematic diagram of Figure 3.3 corresponds to the projected IR image shown in Figure 3.2 when no object is present in the scene. This reference image contains a predefined pattern which consists of tiny dots imaged in the IR spectrum and its reference distribution is stored in the firmware of the sensor during the manufacturing process. The introduction of an object in the field of view of the Kinect depth sensor deforms the reference pattern captured by the IR camera. The resulting shift of the dots in the pattern from the reference image is measured as a disparity in the image plane of the IR camera. The shift essentially takes place along the x axis because the IR camera and the IR projector are parallel and only translated along the x axis, with a separation defined by the baseline, b. Equation (3.1) defines the relationship between the distance to the object, Z, and the raw disparity, d.

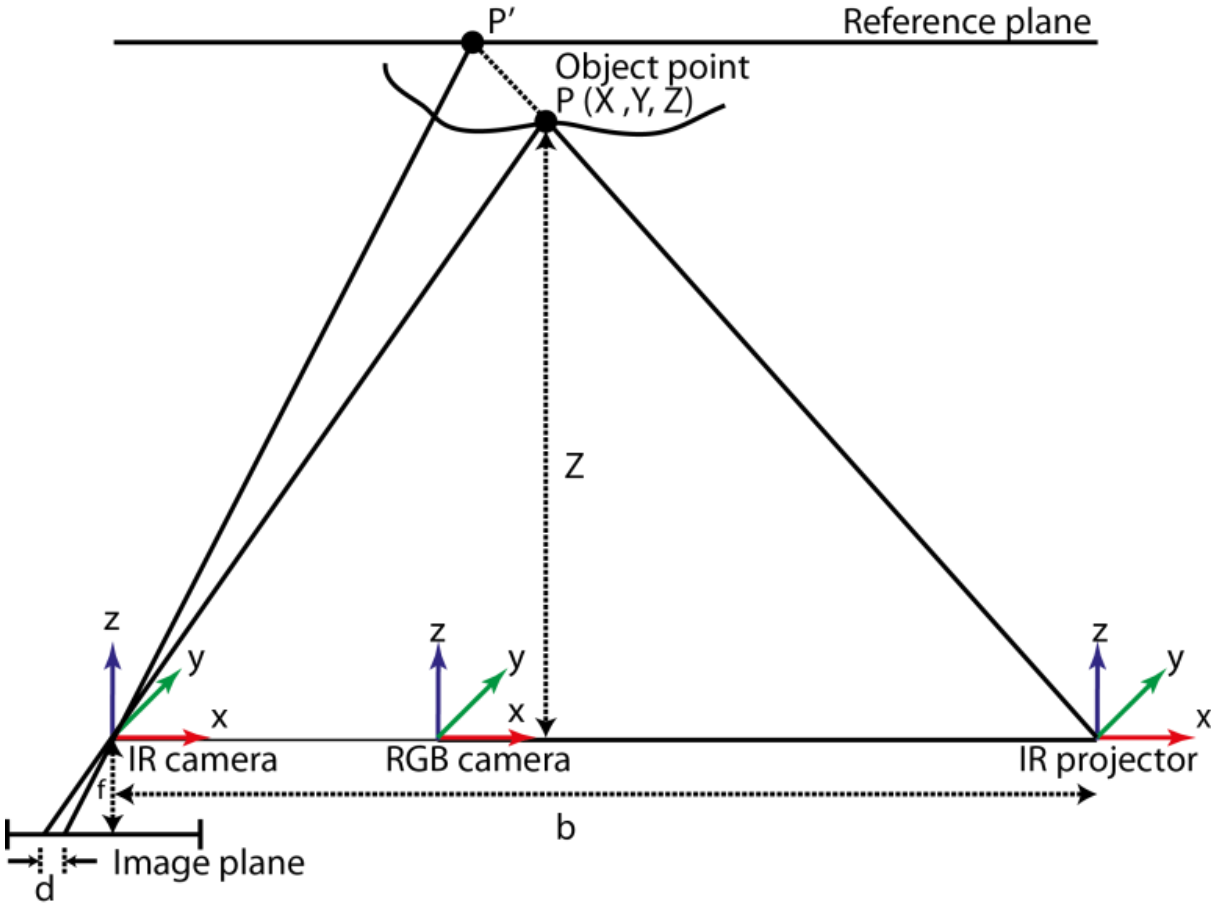


Figure 3.3 : Depth and disparity relationship used for depth estimation.

$$Z = \frac{fb}{d} \tag{3.1}$$

where Z is the depth of the object from the optical center of the IR camera in the direction of the z axis (in meters), b is the horizontal baseline between the IR camera and the IR projector (in meters), f is the focal length of the IR camera (in pixels), and d is the raw disparity (in pixels).

In order to estimate disparity, the Kinect sensor calculates the correlation between every point on the reference pattern, P', and the corresponding point on the deformed pattern, P, produced over the surface of the object in the scene, under the constraint of a known predefined pattern. The correlation is performed using a window of size 9x9 or 9x7 pixels [45, 46]. The results of the matches found by correlation are stored as a raw disparity image. However, Kinect does not return the raw disparity image. Instead, it provides the normalized disparity, d', between 0 and 2047 (11-bit integer). Equation (3.2) defines the relationship between the raw disparity, d, and the normalized disparity, d', which is finally transposed in Equation (3.1) to estimate the depth of a given point marked by the IR pattern over the surface of the object, as shown in Equation(3.3).

$$d = md' + n \tag{3.2}$$

$$Z = \frac{fb}{md' + n} \tag{3.3}$$

where m and n are factors for denormalization. The typical values reported in the literature for those parameters are: $m = 0.125, n = 135.25, b = 0.075m$ and $f = 580$ pixels [46].

3.4 Theoretical Depth Resolution and Quantization

For both versions of the Kinect sensor, the depth resolution depends on the resolution of the disparity image. Kinect returns an 11-bit disparity output. Therefore, it contains 2048 levels of disparity in principle. Nevertheless, *Kinect for Xbox 360* mostly returns disparity values between 400 and 1050, which restrict the output between 0.5m and 8m. Below the minimum range the IR pattern looks like a bright spot with no regular pattern that is sharp enough to establish correlation between the reference pattern and the actual IR pattern. The disparity

value of 2047 is returned where no disparity can be estimated because the object is out of the sensor's depth of field or no correlation between the reference and the actual IR marks can be established in some areas of the scene. The actual range of disparity values for *Kinect for Windows* is different based on the mode of operation. In the *default mode*, the output range is between 0.8m and 8m while in the *near mode* the output range is reduced from 0.4m to 3m. These hardcoded limitations are applied on the final output by Microsoft's Software Development Kit (SDK), which considerably reduces the maximum distance that objects can be imaged from.

The depth resolution of a Kinect sensor decreases non-linearly as the distance between the object and the measuring device increases. The relationship between depth resolution and the distance can be calculated using Equation (3.3), where Z is the actual distance in the real world and d' provides the normalized disparity in the range (0-2047). The relationship between those variables is shown in Figure 3.4, using the parameters reported in the literature, that is $m = 0.125, n = 135.25, b = 0.075m$ and $f = 580$ pixels. Kinect returns 350 disparity levels between 0.5 m and 1 m (closest operational range), while only 23 disparity levels are available to quantify depth between 2.5 m and 3 m (furthest operational range).

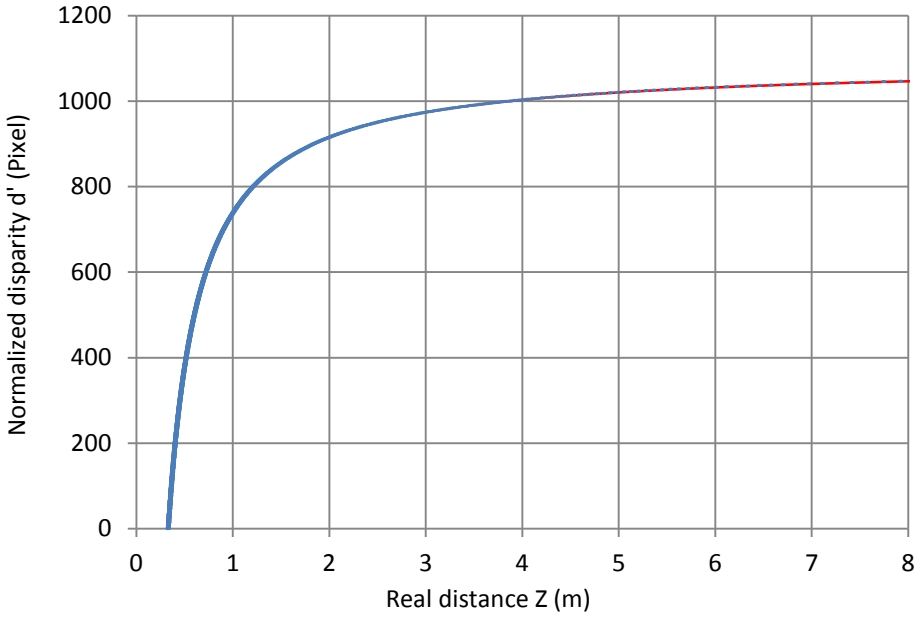


Figure 3.4 : Kinect normalized disparity for various distances (adapted from [45]).

The distance between two possible consecutive depth values determines the quantization error. It is clear from Figure 3.4 that the quantization step size increases with distance because of the reduction in the depth resolution. The difference between each consecutive value is plotted in Figure 3.5. These values demonstrate the quadratic relationship between the distance and the quantization step. A best fit quadratic curve, *Poly*, is also shown in Figure 3.5. Under this approximation, the quantization step, q, as a function of distance, Z, can be defined as in Equation (3.4), which is obtained by fitting a quadratic polynomial on the data.

$$q(Z) = 0.3021Z^2 - 0.056Z + 0.0307 \tag{3.4}$$

where Z is in meters and returns the quantization step, q, in cm. Since the valid operating region is typically between 0.5m and 3.0m, the corresponding quantization error on depth measurements ranges between 0.07cm and 2.58cm.

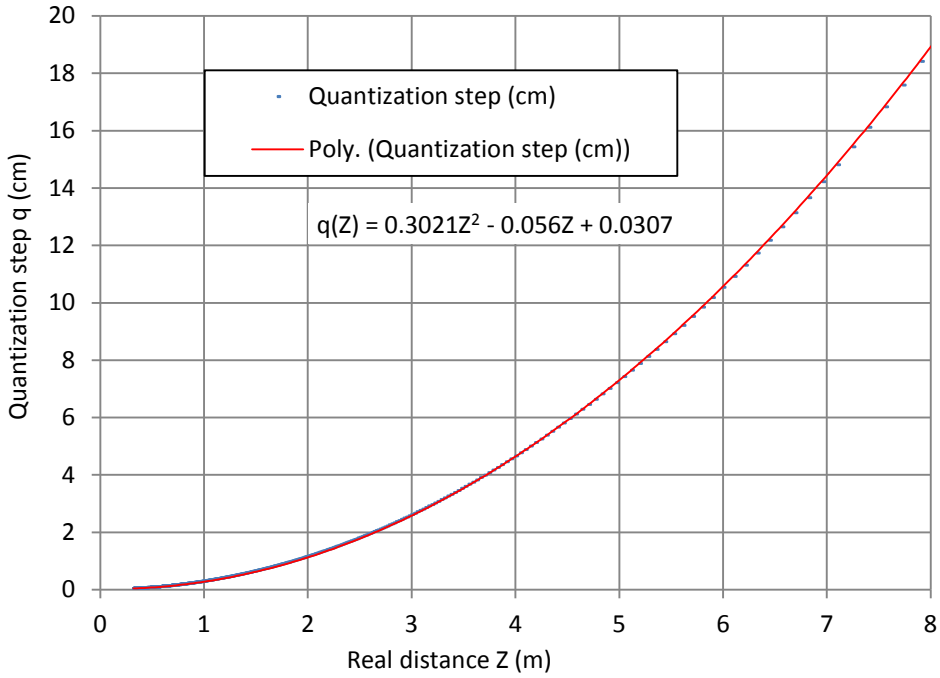


Figure 3.5 : Quantization step size for various distances and best quadratic curve fit (adapted from [45]).

The XY resolution (related to the density of depth measurements over the depth map) depends on the resolution of the depth image. The depth image has a fixed size of 640x480 pixels, therefore the spatial resolution of the points projected on the XY plane also depends on

the distance between the object and the Kinect sensor. The distance between two consecutive pixels in the real world is plotted in Figure 3.6 for a depth range between 0.5m and 8m. These values demonstrate the linear relationship between the quantization step size in (X, Y) and the depth, which can be defined as follows:

$$q_x(Z) = q_y(Z) = \frac{Z}{f} \tag{3.5}$$

where Z is the depth of a pixel and f is the focal length of the IR camera. The distance between two pixels, when projected into the real world increases with respect to the distance from the image plane. This phenomenon can be observed in Figure 3.7, where the projection of few pixels from the image plane to the real world is shown. The distance between pixels at depth Z2 is greater than the distance between pixels at depth Z1.

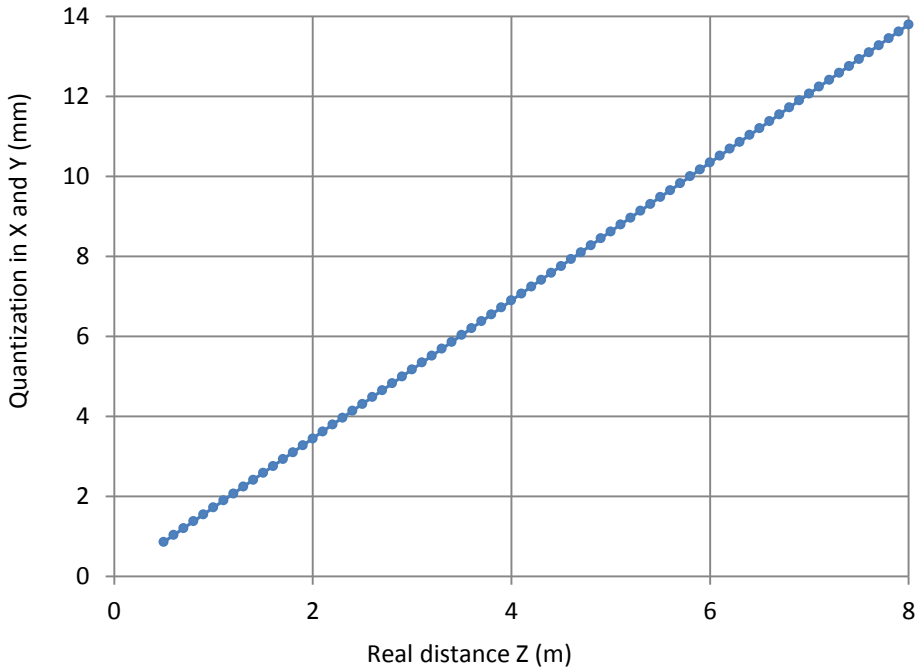


Figure 3.6 : Quantization step size in X and Y for various distances.

For Kinect sensor, the distance between two consecutive pixels in the real world is around 1.03mm for an object at 0.6m while the distance increases to 5.17mm for an object at 3m. Therefore, if an object of size 400x400mm is placed at 0.6m in front of the Kinect sensor, it will

be mapped by (400/1.03)*(400/1.03) = 150815 points, while the same object supports only (400/5.17)*(400/5.17) = 5986 points if located at 3m.

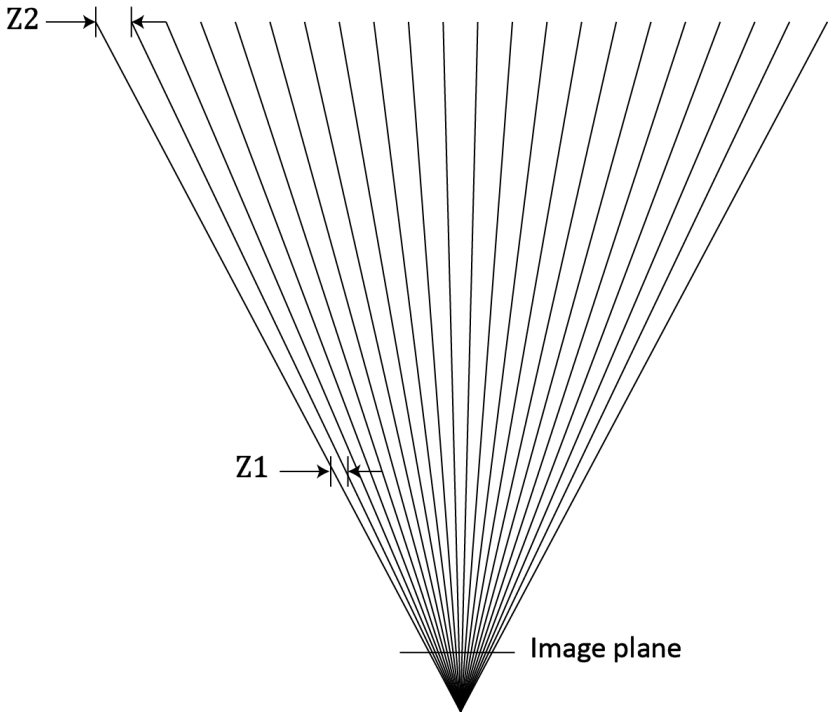


Figure 3.7 : Projection of the pixels in a real world.

3.5 Experimental Evaluation of Kinect Depth Sensor

Experiments were performed respectively with a *Kinect for Windows* and a *Kinect for Xbox360* in order to evaluate the performance of both generations of the RGB-D sensor and to compare with the theoretical expectations detailed in the previous section. The Microsoft SDK is used to capture the data from the *Kinect for Windows*, while OpenNI [47] is used with *Kinect for Xbox360*. There are two operating modes for *Kinect for Windows*, i.e. the default mode and the near mode. *Kinect for Windows* works exactly as *Kinect for Xbox360* in the default mode, but in the near mode it provides access to depth data as close as 0.4m. However, the first version of Microsoft SDK also limits the range of available data. With this software package, the Kinect sensor range in the default mode is between 0.8m and 8m, while in the near mode the range is between 0.4m and 3m. Therefore the data is only available over a bounded range for the experiments. Recently, OpenNI added preliminary support for the *Kinect for Windows*, but

only in the default mode without imposing any limit on depth and therefore provides data between 0.5m and 8m. As such as, OpenNI provides the same range of data from both devices.

3.5.1 Evaluation of Depth Estimation

To perform an evaluation of the depth estimation for the two generations of Kinect sensors, each Kinect sensor is placed on a planar surface in front of a flat wall with close to lambertian reflectance characteristics. The Kinect IR camera is positioned with its optical axis perpendicular to the wall. Then 3D points are captured over the wall. The Kinect is moved repeatedly between 0.4m to 3.5m away from the wall, with an interval of 10cm. At each interval 100 depth images are captured and the whole process is repeated 5 times. The setup of the experiment is shown in Figure 3.8.

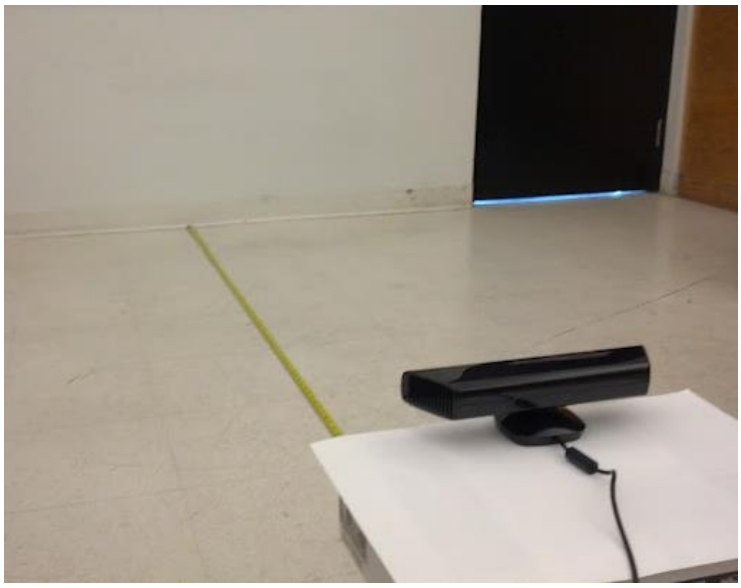
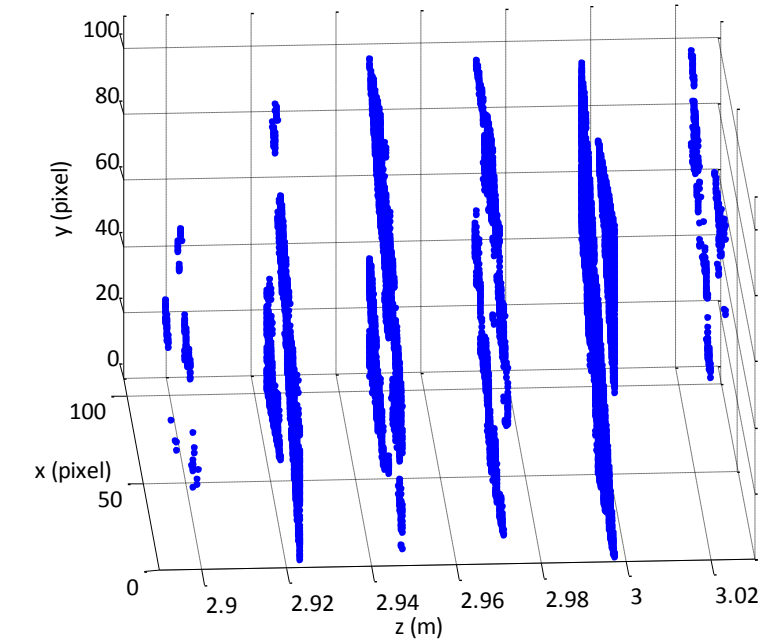
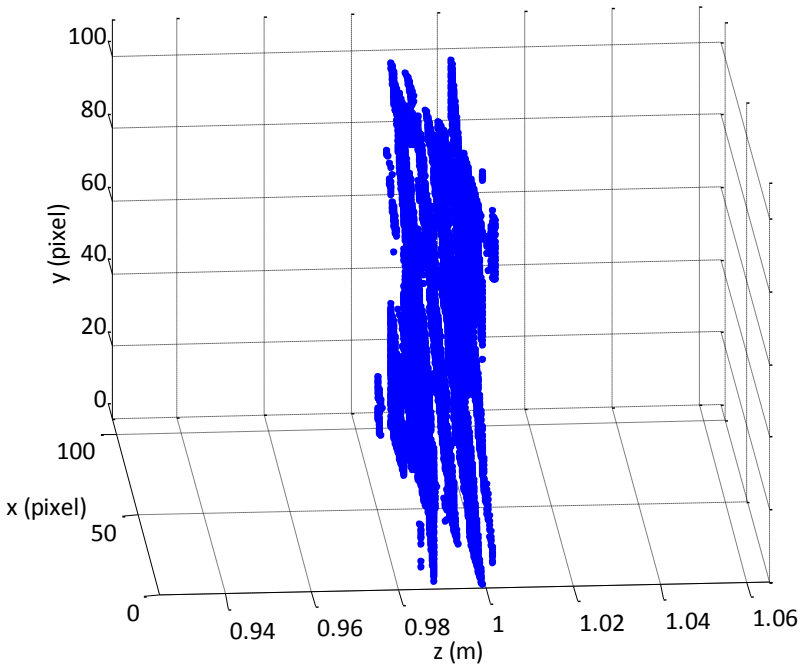


Figure 3.8 : Setup for experimental evaluation of depth estimation.

The recorded depth measurements at each distance not only contain the exact depth of the planar surface but also some depth values with slight errors. Figure 3.9 shows the depth measurements distribution for a wall located respectively at 3m and 1m in front of the *Kinect for Xbox360*. Although the Kinect's IR camera image plane is kept parallel to the planar target to minimize the error, the depth exhibits significant variations. The distribution of points over specific slices clearly demonstrates the impact of quantization errors, which is estimated at about 2.5cm at 3m.



(a)



(b)

Figure 3.9 : Distribution of depth values from Kinect for Xbox 360 over a plane at (a) 3m, (b) 1m.

The quantization step size observed on every recorded value is plotted in Figure 3.10 for both generations of Kinect sensors. The quantization error on both Kinect sensors approximately follows the same quadratic curve defined in Equation (3.4) and depicted in

Figure 3.5, with a slight overestimation of the theoretical quantization error being perceptible for the *Kinect for Xbox360* version.

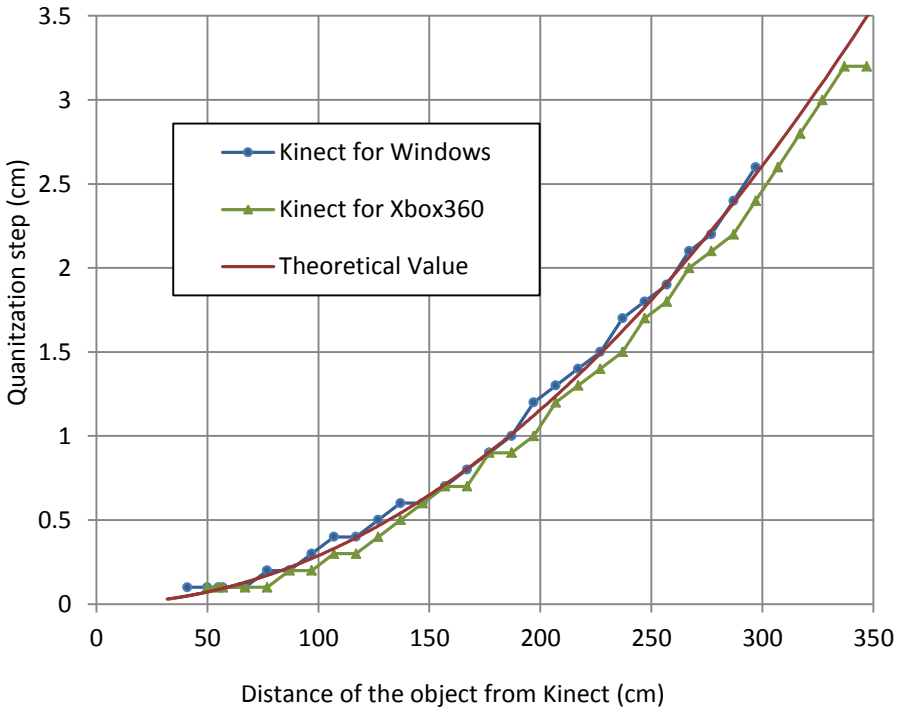


Figure 3.10 : Experimental quantization step size as a function of distance.

In the above experiment, 500 depth images were captured for each interval over 5 successive experimental runs with both Kinect devices. The standard deviation of the distribution is calculated for each interval and for each device. The results are plotted in Figure 3.11. The standard deviation of both Kinect sensors increases fairly linearly within 1.0m and becomes more erratic beyond that distance. Furthermore, *Kinect for Windows* does not compromise the quality of data below 50cm in the near mode, which allows for slightly closer acquisition starting from 40 cm, but systematically cuts any depth estimate beyond 3m. These experiments demonstrate that it is preferable to operate both types of Kinect sensors within a 2m distance from the object to ensure coherence of measurements with a maximum standard deviation of around 1cm.

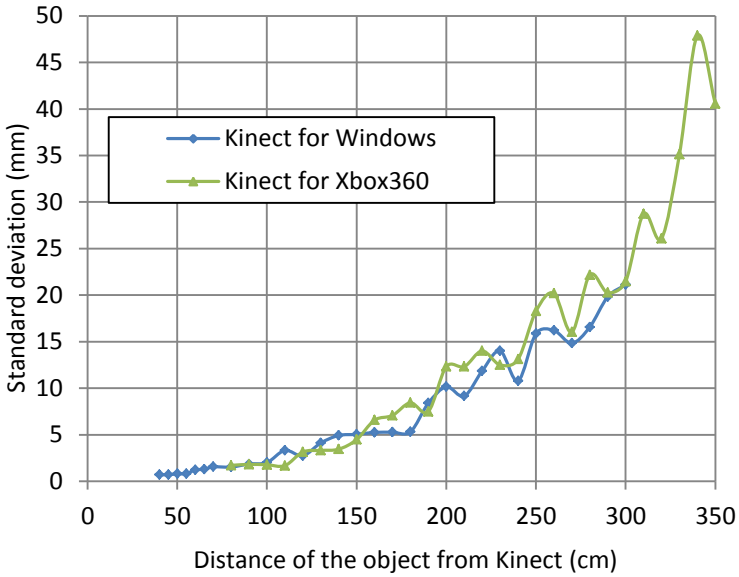


Figure 3.11 : Standard deviation of depth measurement with respect to distance.

3.5.2 Sensitivity to Object's Color and Reflectance Characteristics

The way the Kinect sensor responds to various colors and reflectance characteristics when creating depth maps is also a recognized issue with this technology. The second set of experiments conducted aims at determining the range of capabilities of the sensor when operating on objects with various colors and reflectance characteristics. Since Kinect uses a structured pattern of IR light that needs to reflect back into the IR camera, the amount of IR energy reflected toward the IR camera, and the sharpness of the imaged IR pattern, directly influence the density and the accuracy of the 3D reconstruction.

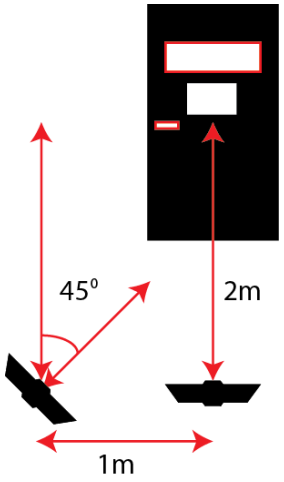
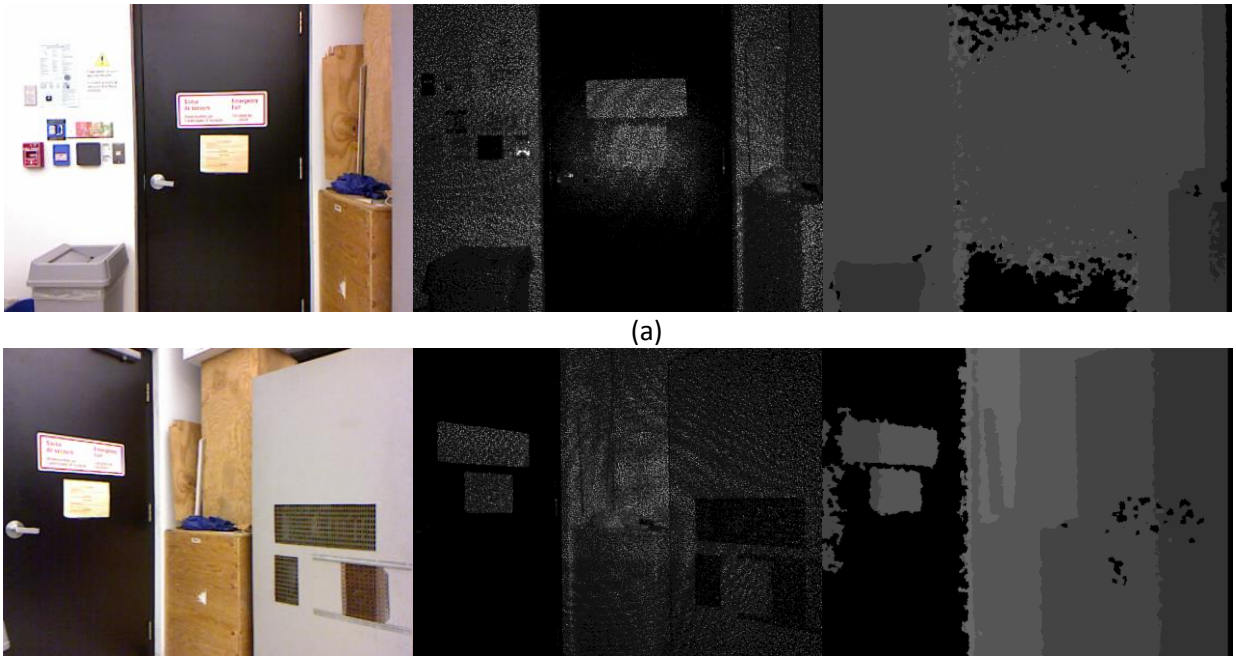


Figure 3.12 : Setup for experimental evaluation of color and reflectance characteristics.

The setup of the experiment is shown in Figure 3.12 where *Kinect for Xbox360* is first placed 2m away from a door and the sensor's principal axis is positioned perpendicular to the door. Figure 3.13 illustrates the response of the IR projection on a scene containing a white wall, a black door and various other objects with different colors and textures. The whole scene is illuminated by the IR projector pattern under standard fluorescent indoor lighting. We notice that the top and bottom parts of the door are not properly imaged in the depth map, as shown by black pixels (missing points) in the lower part of the rightmost image Figure 3.13(a). These regions correspond to areas over which most of the IR energy is absorbed by the surface of the object, as shown in the IR image in the middle part of Figure 3.13(a).



(a)

(b)

Figure 3.13 : Response of Kinect depth camera on a black door and other objects of various colors and textures. (a) Color, IR and depth images respectively, when Kinect is perpendicular to the door, (b) color, IR and depth images respectively, when Kinect's pose is modified.

In order to test with a different configuration, shown in Figure 3.13(b), the Kinect sensor is shifted 1m to the side and rotated 45 degrees, such that the door is off centered in the image plane as shown in Figure 3.12. The black door is therefore imaged over a section of the IR pattern where the IR radiation is not as intense as in the center of the projected pattern, as shown in Figure 3.2. In this case the entire door is missing in the range image, as shown in the

rightmost image of Figure 3.13(b), except for the white stickers attached on the door which are more reflective and make the IR projected pattern visible to the IR camera.

The evaluation was further refined by studying the impact of objects' color and reflectance characteristics over the operating field of view of the Kinect sensor. To further evaluate the operational field of view of the sensor with respect to the distance of the object from the Kinect sensor, black and white objects were used. The black color is known to absorb more radiation, while white objects tend to reflect a larger fraction of the energy that they receive, since many of them are closer to lambertian surface characteristics.

An experiment was performed using both generations of Kinect sensors. As before, in each case the Kinect sensor was placed on a planar surface and aligned with its principal axis perpendicular to a white wall and a black door. Range data was collected at different distances by moving the Kinect sensors between 0.4m and 5m as shown in Figure 3.14. The *Kinect for Xbox360* was moved between 0.5m and 5m and data was collected by both Microsoft SDK and OpenNI. Both software packages gave identical depth estimation. Since the *Kinect for Windows* can be operated in two modes, data was collected between 0.4m and 3m in near mode, and between 0.8m and 5m in default mode, using Microsoft SDK. OpenNI was also used to collect data in default mode between 0.5m and 5. The measurements from both modes were essentially identical. Therefore they were combined to form a comprehensive set of images for *Kinect for Windows* between 0.4m and 5m.

Figure 3.14 : Setup for experimental evaluation of sensitivity to color with respect to distance.

In this experiment, it was found that the top and bottom portions of the door in the depth image were not completely visible. The middle portion of the door is visible but further reduced when the distance between the sensor and the door increases. The distance limits the perception of the black door, which results in a reduction of the effective field of view for depth perception over dark IR absorbing surfaces, as a function of distance. The Kinect depth sensor's field of view typically covers 57 degrees horizontally and 43 degrees vertically. The horizontal angular field of view within which a portion of the black door appears in the depth image is calculated for each interval and is plotted in Figure 3.15. Conversely, white objects keep being imaged over the complete field of view of 57 degrees for both Kinect sensors independently from the distance. As black objects absorb the IR energy and its intensity is further reduced as objects move farther away from the sensor, it results in some clipping over such dark and absorbing objects. This clipping of the view can be seen in the depth image of Figure 3.13(a), where the depth of top and bottom of the black door is not estimated. Therefore the horizontal viewing angle is progressively reduced, as shown in Figure 3.15. A similar characteristic is observed in the vertical direction.

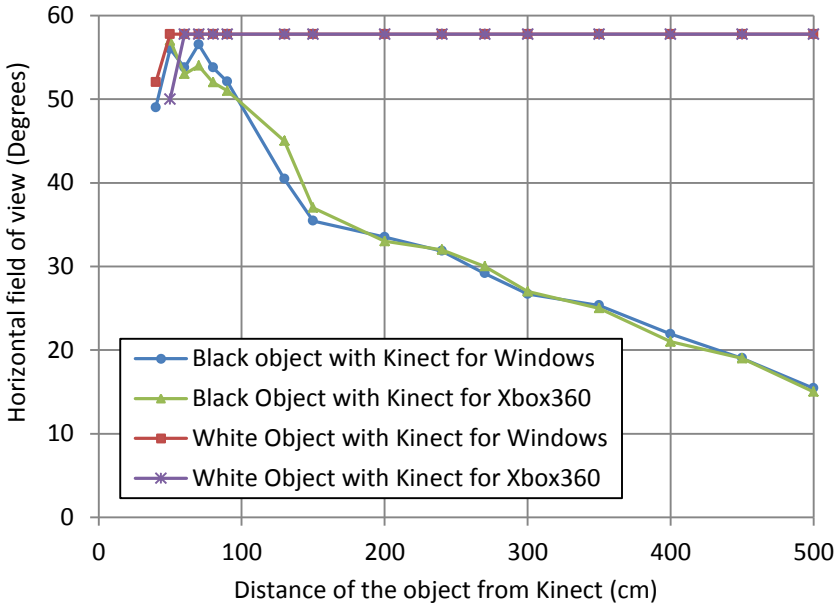


Figure 3.15 : Impact on horizontal field of view of the Kinect sensor over surfaces with different colors and reflectance characteristics. Angles represent the effective portion of initial field of view perceived by the depth sensor as distance varies.

Finally, to investigate the response of the Kinect sensor over different types of materials with complex reflectance characteristics, depth maps were captured over automotive vehicles in two different scenarios. In the first scenario, the experiment was conducted in a semi-outdoor parking garage over day time. Natural light was present in the scene via peripheral openings in the garage (open outside walls) while the sensors were protected from direct sunlight by means of the concrete ceiling. Figure 3.16 shows the view of a vehicle with the corresponding 3D reconstruction achieved from the fusion of three separate point clouds collected side by side. The acquisition was performed over the winter season in Ottawa, Canada, resulting in the vehicle's side panels being covered with dirt and salt deposits from the road conditions, which created various shades of green paint, gray dirty areas, as well as specular reflection spots from the overhead lighting present in the installation. This experimentation demonstrates that without direct sunlight radiation, indirect daylight does not negatively affect the performance of Kinect sensor. The sensor still fully captured the lateral view of the vehicle including the black rubber tires, and the completeness of the 3D reconstruction was not impacted by the shades of green and salt deposits that greatly reduce the shininess of the surface. However, lateral windows of the vehicle are entirely missing in the depth map because the IR energy passes through the transparent surfaces. As a result, parts of the inside seats in the back of the vehicle appear in the 3D reconstruction.

(a) (b)

Figure 3.16 : Vehicle in a semi-outdoor parking garage: a) color image, b) 3D reconstruction.

In the second scenario, the experiment was conducted in an underground garage with artificial ambient lighting from fluorescent tubes. Figure 3.17 presents different views of two vehicles with the corresponding 3D reconstructions. Similar to the previous experience, the windshield and lateral windows of the vehicles are entirely missing in the depth map because

the IR energy passes through the transparent surfaces or is deflected in other directions. But some sections in the interior of the vehicles, that are visible through the glass, are accurately reconstructed. However, the rear window of the red vehicle, which is made of tinted glass, is partially captured. The transparent sections of headlamps are not captured while the rear lamp reflectors, which partially return the IR pattern back to the sensor, are partially reconstructed in the model. The front wind deflector on the first vehicle, made of black shiny plastic also gets partially reconstructed in the 3D model, depending on the relative orientation of the sensor to the sections of the curved deflector. Finally, all of the main areas of the vehicle body and wheels, including dark rubber tires, are very accurately reconstructed, even over narrow roof supporting beams and highly curved bumpers areas. It is also noticeable that the sensor handles well the shiny painted surface of the vehicle in spite of some glare from the overhead lighting. The summary of conclusions from the experiments is shown in Table 3.1.

Figure 3.17 : Different views of a two vehicles and corresponding 3D reconstructions with missing depth areas depicted in white.

Table 3.1 : Summary of analysis on the sensitivity to object's color and reflectance characteristics.

Vehicle parts	Material	3D reconstruction performance
Lateral panels	Painted curved metal and plastic with dirt and salt deposits	Complete
Lateral panels	Painted curved metal and plastic	Complete
Tires	Dark rubber	Complete
Wheel caps	Grey plastic	Complete
Windshield and lateral windows	Transparent glass	Failed, but see through
Rear window	Tinted glass	Partial
Headlamps	Clear plastic	Failed
Rear lamp reflectors	Reflective, colored	Partial
Wind deflector	Dark shiny plastic	Partial

3.6 Summary

This chapter presented an experimental study of the Kinect's depth sensor. A major contribution of these experiments is the formal comparison between the *Kinect for Xbox 360* and the *Kinect for Windows* generations of the sensor. Moreover, the characterization of *Kinect for Xbox 360*, which includes depth resolution and quantization error, experimentally validates the values that are reported in the literature [45]. The investigation in this work was however extended to a wider range of experiments performed in real world scenarios, with objects of different colors and materials' reflectance in order to further evaluate the sensor's operational characteristics and response, beyond what has been reported in the literature until now.

The depth quantization error is found to be similar for both devices and increases quadratically with the distance of the object from the sensor. The standard deviation of depth measurements shows that the accuracy of data collected with the *Kinect for Windows* version operated in the near mode does not deteriorate between 40cm and 50cm, a range that is not accessible with the *Kinect for Xbox 360* version. These experiments also confirm that the best functional operating range for both devices can be up to 2m with standard deviation and quantization error bounded within about 1cm. Finally, both Kinect generations tend to severely degrade their performance by not providing substantial depth information over black or dark

color objects with low reflectance characteristics. Proper perception of depth over low reflectance objects is confined to the center of the field of view and rapidly degrades with an increase in the distance from the object. Furthermore Kinect sensors should be positioned with close to perpendicular orientation to such absorbing object surfaces to provide reliable depth measurements. Finally, an experimental demonstration was made that the Kinect technology can perform fairly well to acquire relatively accurate 3D reconstructions over objects with a large volume, non-transparent but shiny surfaces, and highly curved shapes. Overall, this experimental investigation demonstrated the important influence of the location of objects within the IR pattern projection area, and that of the texture, roughness and color of the surface of the object that influence its reflectance characteristics, as well as of the incident angle of the projection and imaging viewpoint with respect to the normal to the object's surface. In that, Kinect RGB-D sensors exhibit a behavior that is very similar to many other active range sensing technologies.

Discussions in the next chapters only focus on *Kinect for Xbox 360*, which is supported by a growing collection of open source software and development toolkits like OpenNI, Libfreenect, Microsoft, etc. On the other hand, at the present time, the recently introduced Microsoft *Kinect for Windows* is only supported by Microsoft's SDK. However, OpenNI is beginning to provide primarily support for *Kinect for Windows* version, a direction that is expected to continue to grow.

Chapter 4. Multi-Camera System Design

This chapter discusses the design of the multi-camera system which is used in the inspection of vehicles. Designing such a system is not a trivial task because many requirements need to be satisfied. Improper design can potentially lead to a sub-optimal multi-camera acquisition system. This system is based on a number of Kinect devices, which were examined in Chapter 3, and are now grouped into a network of RGB-D sensors to enlarge the field of view to cover a large workspace, as required to accommodate a typical automotive vehicle. The proper design of the vision system is important especially to cover the whole working environment while preserving the minimum setup dimension to constrain the overall system volume and related computational load. The aim of the project is to rapidly collect data over the surface of a vehicle in order to efficiently navigate a robotic manipulator toward regions of interest that are separately identified. Moreover, the system design must ensure sufficient overlapping regions which are essential for the calibration between the RGB-D imaging devices, which will be extensively discussed in Chapter 5. The present chapter specifically addresses the multi-camera system design and proposes an appropriate acquisition framework, with hardware and software considerations.

4.1 Proposed Acquisition Framework

The acquisition system presented here permits efficient and accurate integration of information from multiple RGB-D sensors to achieve fully automated and rapid 3D profiling of automotive vehicles. A set of Kinect sensors are placed conveniently to interact as a collaborative network of imagers in order to collect colored texture and 3D shape information over a vehicle within a short response time. The final goal of the system being developed is to support the navigation of a robotic arm in proximity of the vehicle in order to perform a series of tasks (e.g. cleaning, maintenance, inspection) while it is interacting with the vehicle surface. The specifications initially set for the task were that the entire procedure of acquisition, modeling, and robotic interaction is completed within 2 to 3 minutes.

The proposed layout of the vehicle scanning station is shown in Figure 4.1. The layout is designed to allow a vehicle to pass easily in front of the imaging system, without creating any occlusion. At the beginning, the robotic arm which can be moved on a motorized linear track is positioned at the extremity of its workspace outside the field of view of the sensors. Then, the vehicle enters the scanning area between some guiding lines and stops in the designated space. The yellow lines delimit the area where the vehicle stops while the depth and color information is collected. Five Kinect sensors collect the color and depth information over a 180 degrees view of the vehicle (one side). The same setup can be deployed on the other side to cover the complete 360 degrees view of the vehicle. The information is then processed in order to construct a 3D model of the vehicle.

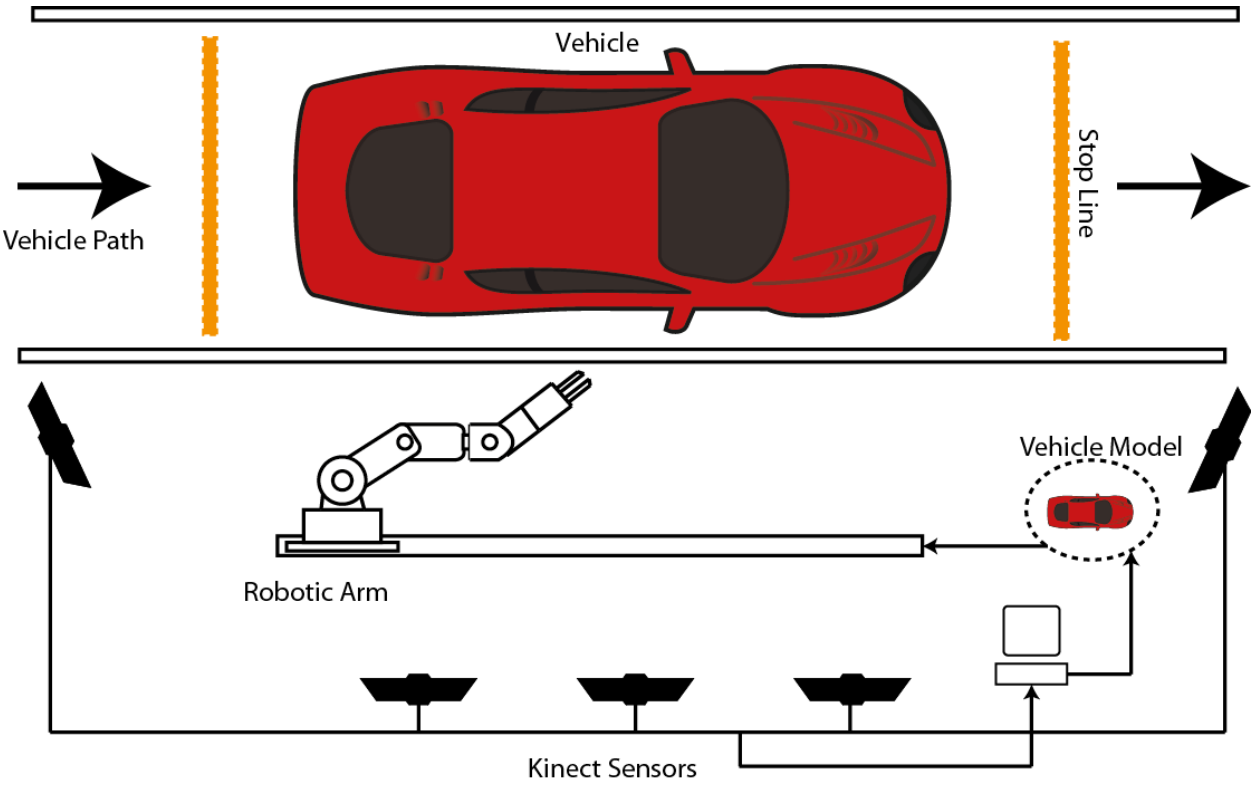


Figure 4.1 : System layout for the proposed scanning system.

The whole scanning and modeling process is meant to be fast, in order to support high inspection cadence. This criterion was the main factor to support the adoption of the Kinect technology in this application in spite of its limited depth resolution and sensitivity to ambient lighting conditions. The final 3D data is used by the robot as general guidance for navigation

and interaction with the vehicle's surface to perform the task. While the acquisition of the surface shape that can be achieved with the Kinect technology is not a priori accurate enough to drive fine and precise interaction of the robot with the vehicle, the precision of the robotic system is meant to be further enhanced by embedding proximity and touch sensing devices on the end effector of the robot. While this aspect of the project is beyond the scope of this thesis, which focuses on the rapid overall prototyping of the vehicle shape, the proximity and touch sensors that will assist the robotic task and increase its accuracy will intervene only over limited regions of the vehicle (where screening is required). The adoption of a rapid visual 3D acquisition technology, as selected in this work, alleviates the time constraints imposed by the large areas that need to be modeled initially before a higher resolution representation is developed only over selected regions of interest, as will be briefly described in section 4.4.1.

4.2 Complete System Overview

The complete range of components for the proposed vision-guided robotic platform for vehicle inspection is detailed in Figure 4.2. The registration module performs registration between color and depth data within each Kinect sensor as well as registration of every 3D point cloud collected with respect to the base of the robot. The methods for experimentally estimating internal and external calibration parameters and for calibration of the robot with the network of Kinect sensors are discussed in chapter 5. All these calibration parameters are used to register the data collected by all Kinect sensors with respect to the robot base. The alignment in between the registered point clouds is further refined in the next step, using an ICP algorithm. Given that the ICP algorithm is fairly slow due to its iterative nature, it is applied only once during the calibration process to provide slight corrections over the marker-based and experimentally determined intra- and inter-calibration parameters between the sensors and the robot. Later, during operation, only the resulting sets of refined calibration parameters are used to perform actual registration between the piecewise datasets. The following stages in the system shown in Figure 4.2 are beyond the specific scope of this thesis but are exposed here for completeness of the framework definition. A surface mesh is created from the registered point clouds using accelerated meshing technique that takes advantage of structured

organization of depth readings generated by Kinect sensors. The meshing technique was developed by collaborators on the project and is beyond the scope of this thesis. It generates the mesh for the point cloud from one Kinect in about 0.1 second. The mesh is then sent to a path planning algorithm for the robot end effector to follow the contour of the object surface with which it needs to interact. At this point and in the specific case where automotive vehicles are considered, the system is assisted by another input to the path planner that helps automatically define, over the generated surface mesh, the location of different vehicle parts that are of interest (e.g. door handles, mirrors, etc.) to drive the inspection. The location of vehicle parts is performed by a custom visual detector of vehicle parts (VDVP) algorithm [48], briefly described in section 4.4.1, which was developed by collaborators on this project. This algorithm requires a 2D color image of a lateral view of a vehicle. Given that the registered point cloud acquired via the calibrated network of Kinect sensors does not provide accurate 2D mapping of the images due to missing depth values over certain areas of the vehicle, original color images, also acquired from the network of Kinect sensors, are merged while taking advantage of the previously estimated calibration parameters to generate a complete lateral color view of the vehicle to support the VDVP algorithm. Finally, because Kinect sensors remain only medium quality depth measurement estimators, the robotic path planning and navigation is further assisted by proximity and touch sensors mounted on the robot end effector. These devices are meant to compensate for any error due to the relatively low accuracy of the 3D model achieved with the Kinect sensors only, and to ensure more precise interaction between the robot and the object under inspection without compromising the rapid rate of acquisition. The development of this proximity/touch sensing layer is also beyond the scope of this thesis. In the following sections and chapters, the blocks shown in green in Figure 4.2, which represent the contributions of this research, will be examined in details.

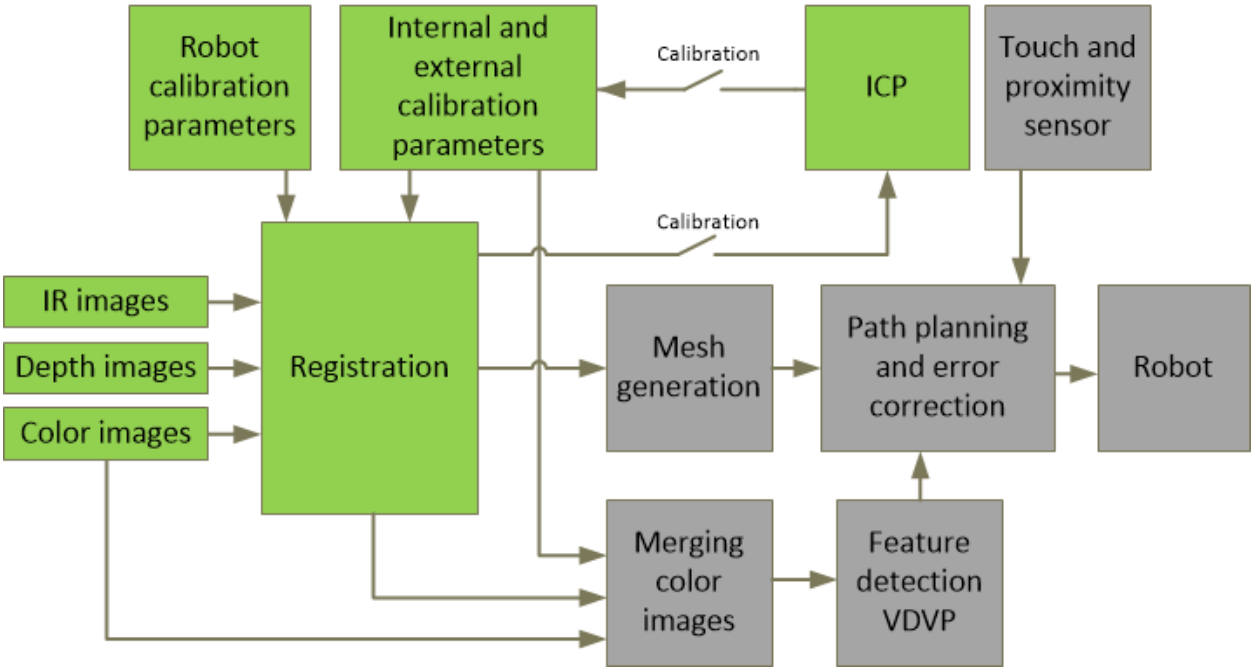


Figure 4.2 : Block diagram of the complete vision-guided robotic system for vehicle inspection.

4.3 Hardware

The acquisition system shown in Figure 4.1 is composed of an Intel corei5 PC with 8GB of RAM, five *Kinect for Xbox 360* sensors and a Thermo CRS F3 robot manipulator. All Kinect sensors are mounted to a reconfigurable structure which is portable and easy to install. This structure consists of three separate pieces. Three Kinect sensors are mounted on one stand that covers the lateral part of a vehicle and the two other sections of the structure cover respectively the partial front and back sides. The structure is designed and installed within the minimum working area possible which allows the vehicle to pass in front of the scanning system while remaining within the active field of view and depth of field of the Kinect sensors. The structure permits the scanning area to accommodate vehicles up to a SUV or a minivan, but no truck or bus.

All Kinect sensors are connected via a USB interface to the same computer. One Kinect is connected to the built-in USB port and the other four are attached to a PCI express USB 2.0 adapter card which adds four extra USB 2.0 ports in the existing mainboard. USB extension cables are used to connect furthest devices. In principle, the average allowable length for USB

2.0 extension is 5m. Within this distance the Kinect devices should operate without any signal degradation issue. The USB cable attached to a Kinect sensor is 3.3m long by default. We increased the length of the cable with a USB extension to determine how far the Kinect sensor can be positioned from the PC. Our experimentation revealed that the device is working fine with a 4m extension cable, which increases the distance between the Kinect sensor and the host computer to about 7.3 m. Beyond that length Kinect sensors were unable to provide the data. This imposed that the computer is located somewhere in the middle of the acquisition system, where all cables are running from proper routing. As an alternative, active extension cables, which boost the signal and increase the possible reach of USB connections, can also be used. Such cables are needed to extend to acquisition system for complete 360 degrees coverage of a vehicle with a single host computer. Another solution consists of duplicating the structure with two separate host computers (left and right sides of the vehicle) and performs data transfer to a central server before merging data and computing a full 3D reconstruction of the vehicle model from a network of 10 RGB-D sensors. However, our experimental investigation did not address this component.

The acquisition system also requires proper lighting for clear color and bright IR images to be generated. The illumination is provided by incandescent lamps. These lamps generate light over the visible to deep infra-red spectrum, which provides good lighting conditions for both the color and IR cameras.

4.4 Architecture Design

The previous section described the general overview of the system and its main components. The current section describes how all those components are connected to the main computer and interact with each other. The diagram of Figure 4.3 shows the complete system architecture. Usually a computer has two built-in host controllers on the motherboard. Each controller is capable to handle multiple USB ports. However, *Kinect for Xbox 360* devices cannot be connected to USB ports that share the same USB host controller. *Kinect for Xbox360* sensors must rather connect to USB ports which are operated by separate USB host controllers into the computer. Therefore, a maximum of two *Kinect for Xbox 360* devices can typically be

connected to each computer. On the other hand, two *Kinect for Windows* devices can operate on a single USB host controller. Therefore, up to four of these devices can be connected to a single computer, if four USB ports are available, while capitalizing on its two inherent host controllers. Fortunately, USB 2.0 adaptor cards are available on the market to increase the number of host controllers. Our implementation uses one PCI express to USB 2.0 adapter card, which is capable to handle four more *Kinect for Xbox 360* devices through four separate host controllers that it creates in the computer. Kinect *K0*, is attached to the default system USB port, while the other four Kinect devices, i.e. *K1*, *K2*, *K3*, and *K4*, are connected to the USB 2.0 ports of the adapter card.

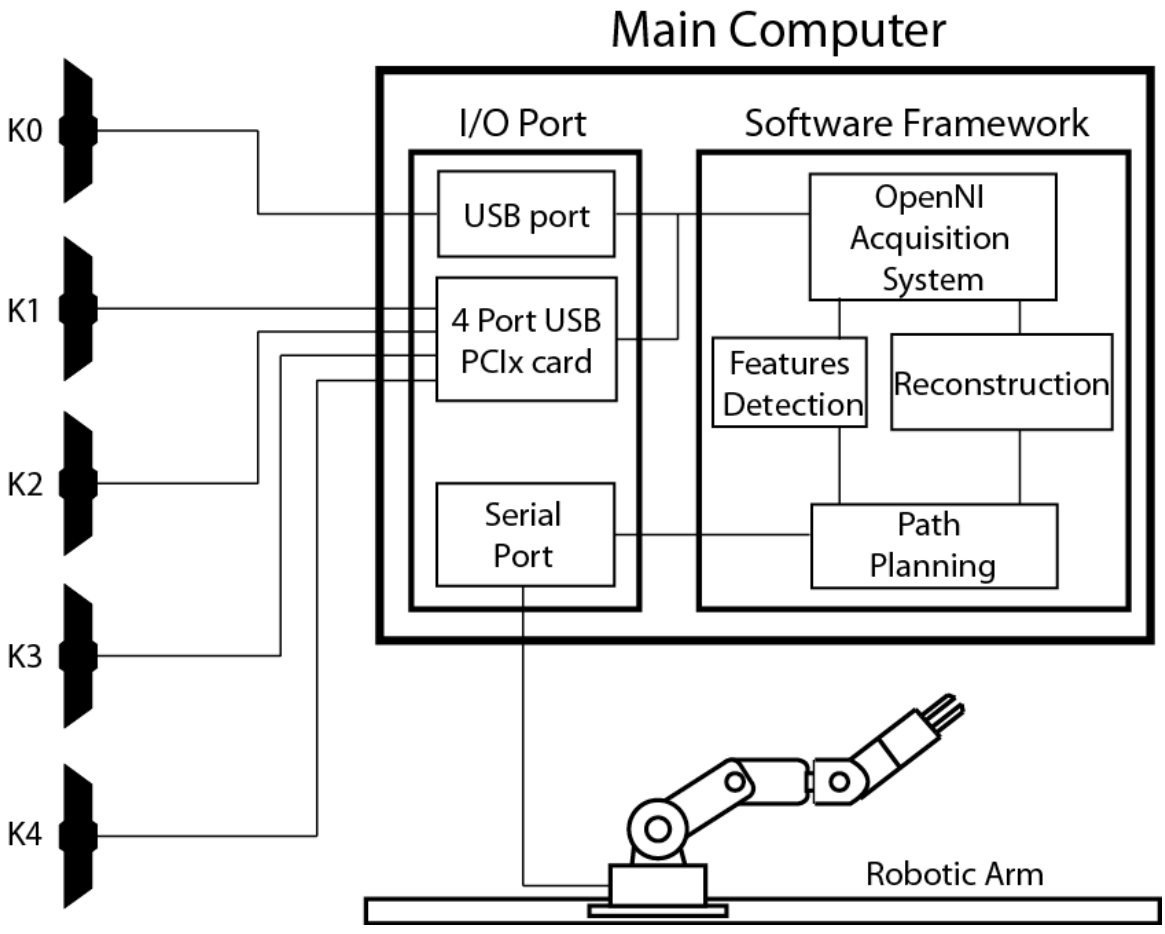


Figure 4.3 : Robot-vision architecture design.

All Kinect sensors are operated using the OpenNI framework [47] which consists of drivers and software commands to communicate with a Kinect device. The data provided by the

devices and collected using OpenNI is then registered and used to reconstruct a model of the object. The data provided by the devices is also used to detect regions of particular interest in the model (discussed in section 4.3.1). The complete reconstructed model and the areas of interest in the model are then utilized by path planning algorithms and software tools to guide the robot towards the particular features over the object where the task is performed. The robot is connected via a serial cable, which connects the robot controller box to the same computer where the RGB-D data is acquired and the model is generated.

4.4.1 Feature Detection and Path Planning

The purpose of the features detection process in the proposed framework is to specify areas of interest over a large object such as a vehicle in order to speed up the modeling process that is needed for the guidance of the robot arm that will eventually interact with the vehicle to perform either inspection or maintenance tasks. Acquiring knowledge about the location of dominant features over a vehicle reduces the amount of time spent on scanning at higher resolution to accurately drive the manipulator by focusing the scanning operation only over selected areas of a limited size. It also allows for the robot to rapidly determine where to operate, as it is very unlikely that the robotic operation will be required over the entire vehicle.

For the efficient and reliable detection and localization of characteristic areas over a limited accuracy 3D reconstruction of a vehicle, as provided by Kinect sensors, a visual detector of vehicle parts (VDVP) was introduced by Chávez-Aragón *et al.* [48], as a complementary work to this research. The VDVP receives as an input a color image of a lateral view of the vehicle to determine the location of up to 14 vehicle parts. The method works with images of different types of vehicles such as: 4-door sedan, 2-door sedan, 3-door hatchback, 5-door hatchback, SUV and pickup trucks. Figure 4.4 illustrates the detection that can be achieved by applying the method over a test image. Round areas indicate features detected by the classifier; square regions mean that the locations of the features were inferred based on other known features.

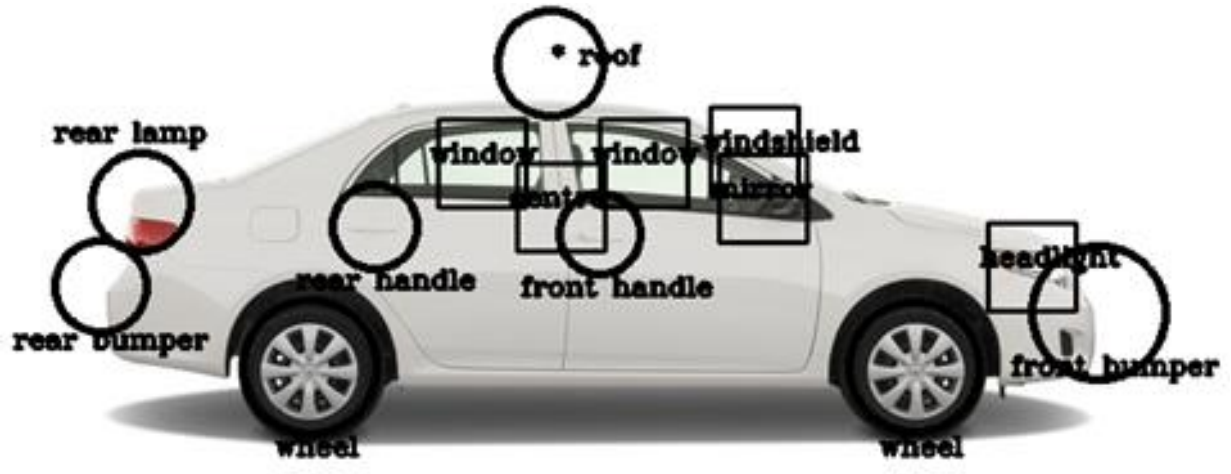


Figure 4.4 : Automatic detection of parts of interest over a side view of a car (reproduced from [48]).

4.5 System Integration

The integration of the numerous components of the proposed acquisition framework is performed in four steps described in Figure 4.5. This first step consists of the intrinsic calibration of every camera used in the system. The proposed system is based on Kinect sensors, therefore, each camera, color and infrared, within every Kinect unit is first calibrated individually. The second step consists of the extrinsic calibration between the color and the IR cameras contained within each Kinect unit. This calibration is used to accurately merge color and depth data generated by a given Kinect sensor. The first two steps are independent from the placement of the Kinect sensors in the overall system. Therefore, these steps are only applied once and called *internal intrinsic* and *internal extrinsic* calibration, respectively. Before the next two steps are performed, all the Kinect sensors and the robot are properly installed into the acquisition structure. The third integration step consists of estimating the position and orientation of each Kinect sensor with respect to one reference Kinect sensor. This procedure represents the *external extrinsic* calibration stage. The selected reference sensor is used in the fourth and final step to calibrate the vision system with the robot reference frame attached to its base. These procedures will be further detailed in Chapter 5.

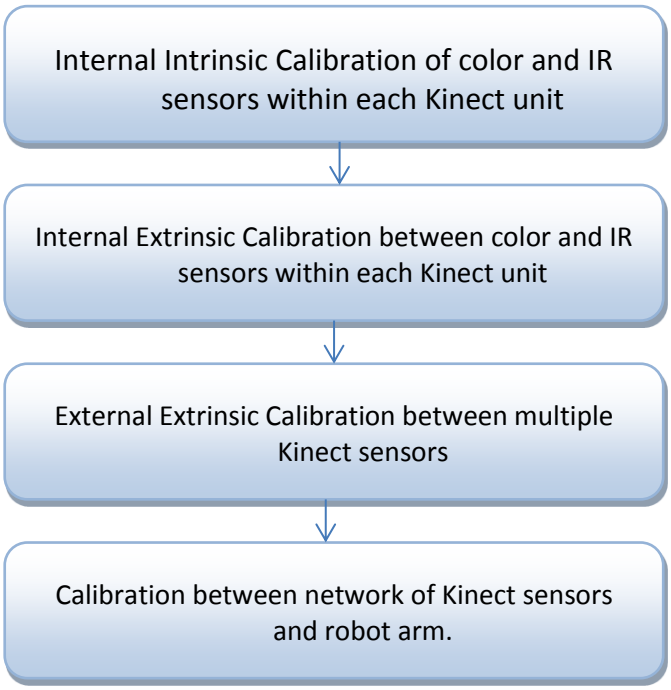


Figure 4.5 : System integration steps.

4.6 Cameras Configuration

The previous sections described the general model of the acquisition system and the hardware requirements. The purpose of this section is to analyze where to locate Kinect cameras in order to obtain complete scanning capabilities over the volume occupied by a vehicle with a good quality of data. Chapter 3 detailed the operational characteristics of Kinect devices. It was found that the valid operating region is typically between 0.5m and 3.0m, but the best functional operating range for both versions of the device is between 0.5m and 2m since the standard deviation and quantization error are bounded to about 1cm within that range. Moreover the behavior of the sensor over dark surfaces further reduces the horizontal viewing angle by about 40% at 2m. These limitations provide important guidelines to determine that the maximum distance to the vehicle should be around 2m to ensure sufficiently accurate measurements to safely navigate the manipulator around the vehicle. The vertical field of view of Kinect sensor is 43 degrees, therefore, at 2m it almost covers a height of 1.6m, which is slightly under the nominal height of a standard vehicle. In case of a complete dark surface, the field of view may be further reduced and the sensor focus only on the middle portion of the vehicle.

The proposed acquisition system that was developed for rapidly collecting color and depth information over vehicles is depicted in Figure 4.6. The five Kinect sensors are positioned to cover the complete side and partial front and back sections of the vehicle. The setup covers a 180 degrees view of a vehicle and can be replicated to the other side for a 360 degrees view, if necessary. The sensors are positioned 1.0m above the ground and kept parallel to the floor. The Kinect sensors *K1*, *K2* and *K3* are positioned to provide the whole lateral view of the vehicle, while the Kinect sensors *K3* and *K4* are rotated towards the vehicle by about 65 degrees with respect to sensors *K1*, *K2* and *K3* and provide partial front and back views, respectively. The distance between Kinects *K0*, *K1*, and *K2* is 1.3m, which provides a significant overlap of about 0.85m between views at 2m distance. Furthermore, the position of *K4* and *K3* also provides significant overlap with *K0* and *K2* respectively. Such overlap between contiguous sensors is implemented to ensure accurate point clouds alignment and to support the calibration process that will be detailed in Chapter 5. Moreover, it also helps filling missing depth data in the model.

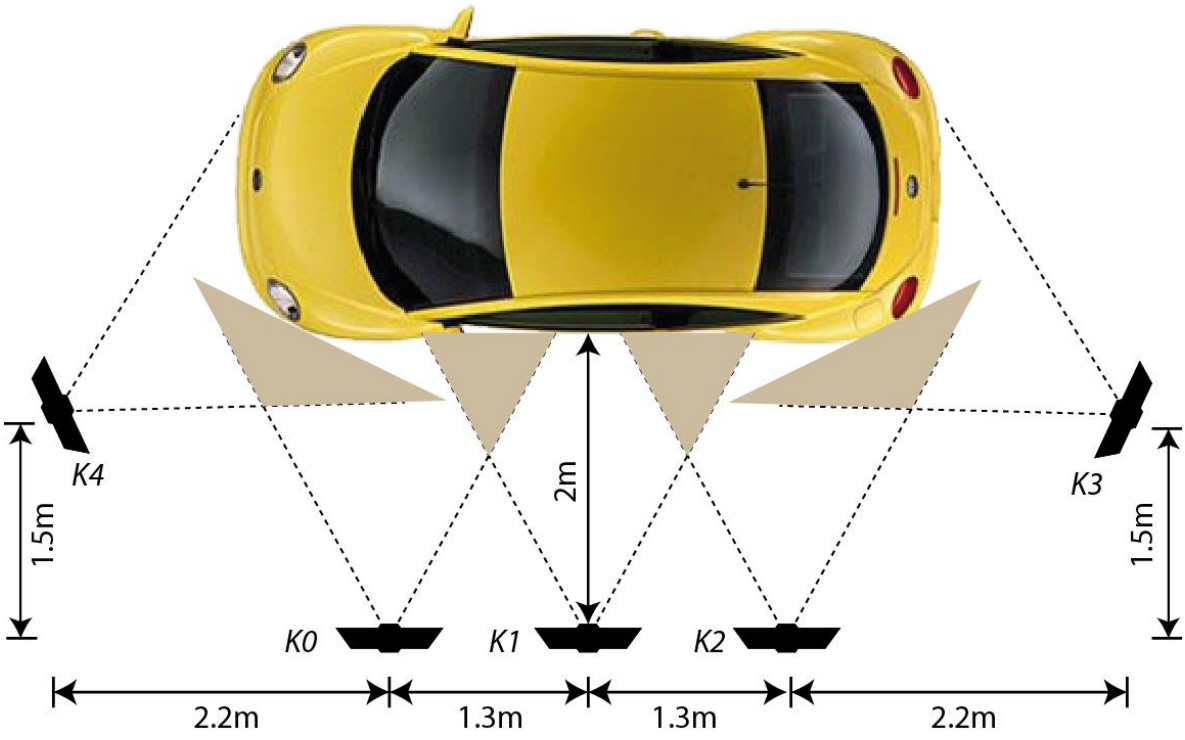


Figure 4.6 : Experimental configuration of acquisition stage for scanning a vehicle.

This configuration permits to meet the following requirements to cover the entire side of a vehicle: 1) a minimum coverage area of 4.8 *m x 1.6 m*, which is the typical size of a vehicle; 2) collection of depth readings within the range of *0.8 m* to *3* m, which corresponds to the range within which Kinect sensors perform well; 3) an overlapping area in the range of *0.5* m to *1* m, between contiguous sensors to ensure accurate point clouds alignment and external extrinsic calibration process. Table 4.1 summarizes the parameters of the proposed scanning system. It is worth mentioning that the acquisition stage can be easily adapted for larger vehicles by including extra sensors along the line of Kinects *K0*, *K1* and *K2*.

Table 4.1 : Parameters of the proposed scanning system.

	IR camera	**RGB camera**
Horizontal field of View	57^{o}	61^{o}
Vertical field of view	43^{o}	47^{o}
Distance between sensors (K0,K1),(K1,K2)	1.3 m	1.3 m
Distance between sensors (K0,K4),(K2,K3)	2.66 m	2.66 m
Height of the sensors over the ground	1 m	1 m
Distance between (K0,K1,K2) sensors and vehicle	2 m	2 m
Horizontal overlapping area between two sensors	0.85 m	1.15 m
Coverage area for each sensor	2.2 m x 1.6 m	2.35 m x 1.75 m
Total coverage area for the (K1, K2 and K3) depth sensors	4.8 m x 1.6 m	4.95 m x 1.75 m

4.7 Interference Issues

As discussed in Chapter 3, Kinect sensors operate on the principle of structured lighting for depth measurements. Each Kinect sensor projects an IR pattern over the scene and compares the IR image of this projected pattern with an image of the predefined pattern stored in internal memory to calculate the depth. Multiple Kinect devices working simultaneously in the same environment create interference one to each other within the overlapping regions between contiguous Kinect sensors since all Kinect devices project the same pattern of infrared points at the same wavelength to create their respective depth map. As a result of interference, a given Kinect sensor is unable to determine the proper correspondences for parts of the pattern. This produces holes on the depth maps of overlapping sensors. The effect is shown in Figure 4.7(b) where a depth map of a mockup car door is generated by one Kinect sensor

operating alongside a second unit that simultaneously projects its own IR pattern over the same surface. The corresponding depth map when only one Kinect device operates on the object is shown in Figure 4.7(a). The interference created by multiple and simultaneous IR pattern projections over a same scene is demonstrated by the appearance of multiple holes over all sections of the objects.

(a) (b)

Figure 4.7 : Comparison of depth maps: a) without interference from another IR projector, b) with interference from another IR sensor.

To prevent this problem, the solution is to collect data sequentially over different time slots. Unfortunately, Kinect sensors do not have any software switch to only shutdown the IR projector. Therefore, Kinect devices should be completely shut down in alternance and then re-initialized sequentially to image the entire automotive body panels. This process introduces minor delays between the shots, where the amount of time needed to shut down the device and re-initialize the next device impacts the speed of acquisition. The delay slightly varies depending on the number of devices attached to the system. In our setup, the delay is between 1 and 2 seconds to perform data collection over the entire vehicle, when 5 Kinect devices are connected to the acquisition system. The OpenNI framework is used to control the Kinect sensors. In a first time slot, sensors *K1*, *K3* and *K4* simultaneously collect their respective information. No interference happens since their respective fields of view do not overlap. Next, sensors *K0* and *K1* complete the scan of the vehicle.

4.8 Summary

This chapter examined the design considerations of a multi-camera system based on a network of Kinect sensors. The layout of the proposed system was presented with a discussion about the hardware requirements. The hardware and software communication details were discussed next. Then the four steps required for system integration were detailed in terms of calibration components. Some limitations related to Kinect sensors were taken into the consideration to determine the proper positioning of the sensors around a standard size vehicle. Attention was paid in the design to properly support the calibration procedure that will be detailed in the next chapter, while also ensuring the quality of the color and depth data collected by the acquisition system. Finally, interference issues appearing when using multiple Kinect devices simultaneously were investigated and a solution was proposed.

Chapter 5. Multi-Camera System Calibration

A premise to almost all multi-camera computer vision applications is the accurate calibration of all camera sensors. Chapter 2 introduced some methods to achieve such calibration that are either based on a classical 2D planar calibration target [23] [24] [25] [27] or a non-planar target [30] [36] [38] [40]. The use of points extracted over a non-coplanar target result in more accurate calibration than with a coplanar target due to the lack of correspondence between the points over the target itself [23]. However non-coplanar targets, i.e. 3D calibration objects, are more complex to design. A planar target with coplanar points is easy to create but requires more than one view to generate a sufficiently large number of points to perform robust calibration.

This chapter proposes a method to calibrate multi-camera systems. The approach is implemented and validated, more particularly, using the vision system designed in the previous chapter. The proposed method combines the strengths of approaches based on classical 2D calibration targets to model the internal behavior of each camera sensor, and also uses the target to generate a cloud of 3D points, to achieve precise and complete inter camera registration. Of particular interest in the current imaging framework, advantage is taken of depth sensors to generate the 3D point clouds and that information is used to estimate the registration parameters between sensors. The method is suitable for RGB-D cameras, like the Microsoft Kinect. It is also further extended to support the calibration between a robotic manipulator and the multi-camera network using a 2D calibration target mounted on the robot.

5.1 Camera Calibration Overview

The calibration procedure of the network of RGB-D sensors is divided into two stages, i.e. internal and external calibration. The internal calibration procedure estimates the intrinsic parameters of each camera within every device as well as the extrinsic parameters between the RGB and the IR cameras inside a given Kinect unit. The external calibration estimates the extrinsic parameters in between any respective pair of Kinect devices. In the first stage, the intrinsic parameters of every camera are estimated. This also includes the estimation of lens

distortion parameters for each. Since these parameters are fixed and completely independent from the positioning of the cameras, it is convenient to estimate these parameters separately for each camera. Each Kinect sensor consists of a color and an IR camera. Therefore, the internal intrinsic parameters are estimated for both cameras, separately. The relative position and orientation of the color and IR cameras into a given Kinect unit is fixed, but only roughly defined by parameters reported in the literature and not very precisely reproduced in every Kinect device, as our experimentation with several Kinect units demonstrated. Therefore, the extrinsic parameters between these two cameras are also estimated individually for every Kinect used in the network, as part of the calibration procedure. These internal extrinsic parameters help in more accurately merging the color and depth data collected by one Kinect unit, as will be discussed in section 5.2.4.

All of the fixed parameters of the RGB-D sensors are estimated in a first calibration stage. In a second stage, the relative position and orientation between pairs of Kinect units are estimated. Prior to performing the second phase of the calibration procedure, all Kinect units involved in the network are positioned according to the system configuration developed in Chapter 4. The relative position and orientation between Kinect units is estimated with respect to their respective IR camera. This choice is made because the depth data, which is used for calibration, is readily generated with respect to the IR sensor. The second calibration stage provides the external extrinsic parameters. It needs to be repeated every time there is a change in the configuration of one or more Kinect units in the network of sensors, while the internal intrinsic and extrinsic parameters can be preserved. The latter remain independent of the multi-camera network configuration, as long as the same Kinect units remain in operation. The overall proposed calibration procedure is illustrated in Figure 5.1. The following two sections detail the proposed calibration procedure for the network of Kinect sensors.

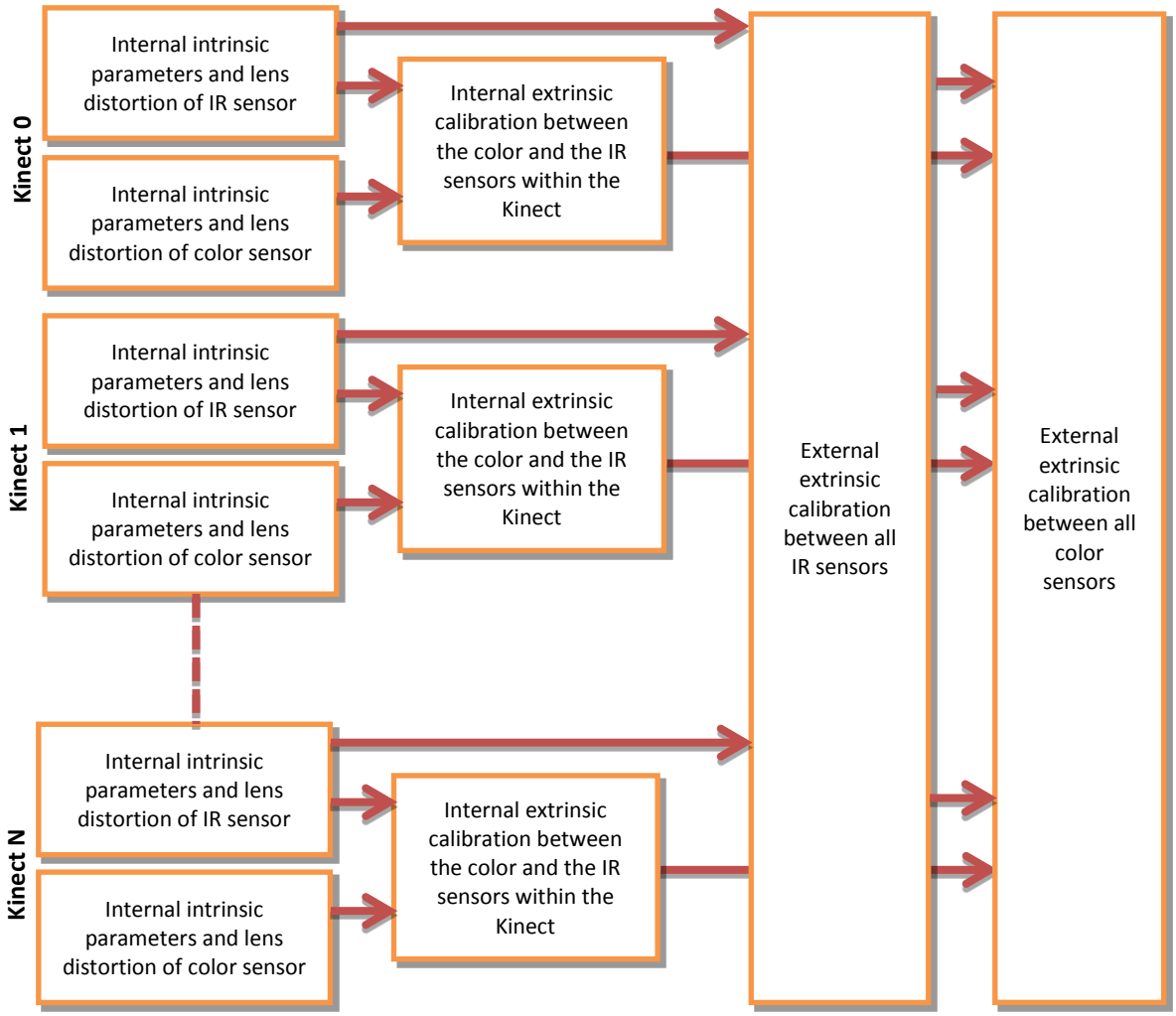


Figure 5.1 : Overview of the proposed calibration procedure for a network of Kinect RGB-D sensors.

5.2 Internal Calibration

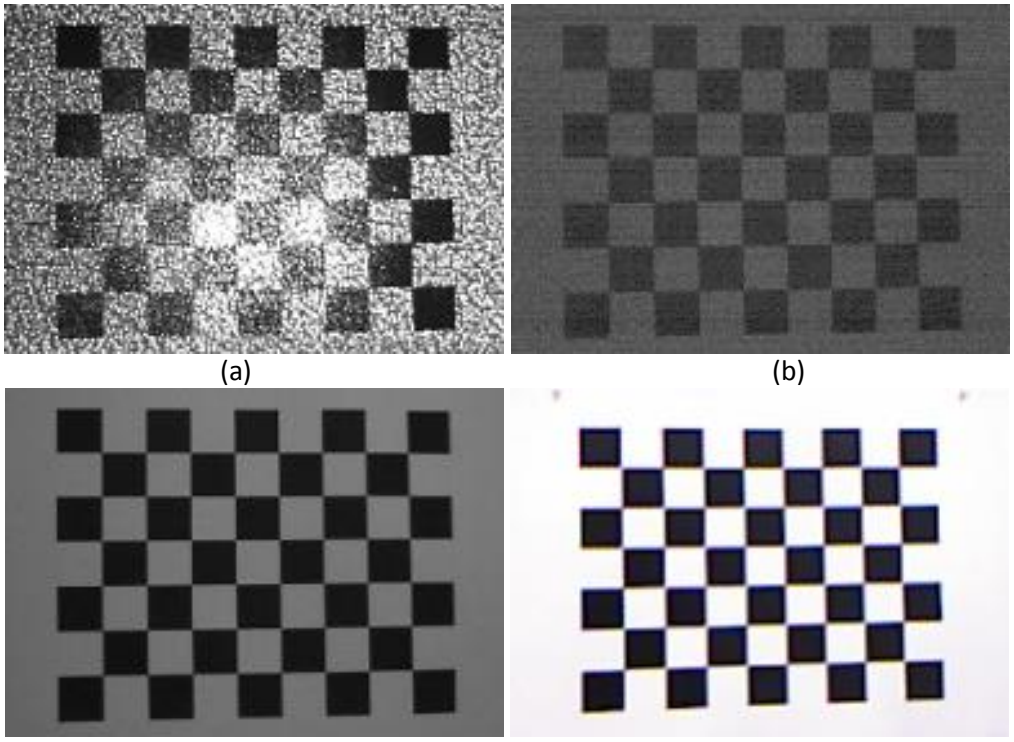
5.2.1 Intrinsic Parameters Estimation for Built-in Kinect Cameras

The internal calibration procedure includes the estimation of the respective intrinsic parameters for the color and the IR sensors, which are: the focal length(f_x, f_y), the principal point(O_x, O_y), and the lens distortion coefficients $(k_1, k_2, p_1, p_2, k_3)$ [24]. Because the RGB and IR cameras exhibit different color responses, the proposed calibration technique uses a regular checkerboard target of size 9x7, as shown in Figure 5.3 that is visible in both sensors' spectra. During internal calibration the Kinect's IR projector is blocked by overlapping a mask on the projector window as shown in Figure 5.2, since it cannot be turned off by the driver software.

The IR projector otherwise introduces noise over the IR image as shown in Figure 5.3(a). Without projection, the IR image becomes too dark to accurately retrieve features points from it, as shown in Figure 5.3(b). Therefore, standard external incandescent lamps are added to the setup to illuminate the checkerboard target, which improves the quality of the IR image, as shown in Figure 5.3(c). The color image is not affected by the IR projection and creates a clear pattern under these lighting conditions, as seen in Figure 5.3(d).

(a) (b)

Figure 5.2 : Covering projector's window during internal calibration: a) without mask b) with mask.



(a) (b)

(c) (d)

Figure 5.3 : Views of the checkerboard in different configurations: a) IR image with IR projector, b) IR image without IR projector, c) IR image with incandescent lighting and without IR projector, and d) color image with incandescent lighting and without IR projector.

The checkerboard is printed on a regular A3 size paper of 420 x 297 mm. Regular mat paper is preferable to glossy photo paper because it does not reflect back the bright blobs that

can be created by the external incandescent lamps in the IR image plane. To ensure the best calibration results, the calibration target is moved within the entire field of view of the Kinect sensor and 100 images of the calibration board are collected from both the color and the IR cameras. Both images are synchronized in each frame, such that they can be used for extrinsic calibration between the cameras (next section). These images contain the projection of the 3D object points, which are the corners of the checkerboard target in the pattern coordinate system. The 2D corresponding points of each view of the checkerboard are extracted by OpenCV *findChessboardCorners* function [49]. This function gives the location of the corners on the image plane. All the 2D projection points along with the 3D object points are then used to calculate the intrinsic parameters of camera using OpenCV *calibrateCamera* method. This method estimates the intrinsic and extrinsic parameters for each view of checkerboard using Zhang's camera calibration method [24]. The method also calculates and minimizes the reprojection error that is the sum of the square distance between 2D image points (obtained from corners extraction) and projected 2D points using the current estimation of intrinsic and extrinsic parameters.

$$Reprojection\ error = \sum_{i} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \tag{5.1}$$

Where (x_i, y_i) and $(\hat{x}_i, \hat{y}_i)$ denote the image points and the projected points respectively.

The method is applied on 10 groups of 30 images randomly selected among the 100 captured images. Each group contains random samples of the images which helps achieve accurate calibration over the entire workspace of the cameras since during image capture the checkerboard is moved in a series of steps from left to right and top to bottom over the complete field of view of the cameras. The calibration method returns the reprojection error and the estimated intrinsic parameters for each group of images. Among the results, the intrinsic parameters with the least reprojection error are selected. The results of the intrinsic calibration with the least reprojection error are shown in Table 5.1 for five Kinect sensors involved in the network.

Table 5.1 : Internal intrinsic calibration of embedded sensors.

Intrinsic Parameters of IR camera in pixels					
sensor	f_{x_IR}	f_{y_IR}	O_{x_IR}	O_{y_IR}	**Reprojection Error**
K0	584.2	582.6	326.7	233.5	0.136
K1	585.9	583.8	325.2	242.3	0.148
K2	597.7	595.7	322.2	232.1	0.131
K3	599.0	597.1	331.5	240.3	0.157
K4	581.7	579.5	319.6	246.3	0.145

Intrinsic Parameters of RGB camera in pixels					
sensor	f_{x_RGB}	f_{y_RGB}	O_{x_RGB}	O_{y_RGB}	**ReprojectionError**
K0	517.9	516.7	321.0	245.6	0.127
K1	518.8	517.0	331.1	261.4	0.124
K2	535.7	537.3	336.2	252.8	0.129
K3	525.1	523.0	322.1	255.1	0.153
K4	517.2	515.2	319.7	254.8	0.146

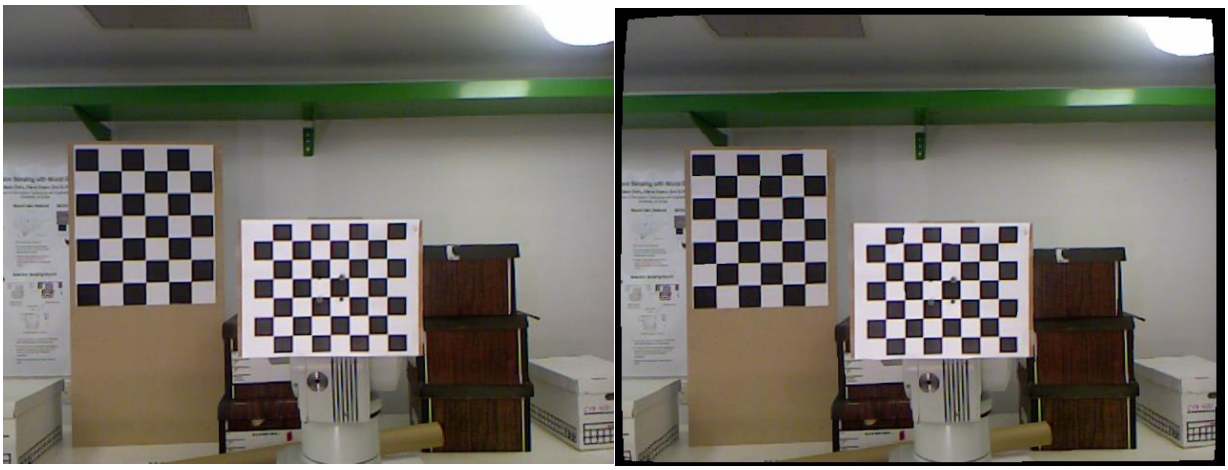
Distortion Parameters of IR camera						
sensor	k_1	k_2	p_1	p_2	k_3	**Reprojection Error**
K0	-0.1193	0.5768	0.0011	0.0037	-0.8692	0.136
K1	-0.1323	0.6297	-0.0004	0.0028	-0.9595	0.148
K2	-0.1279	0.7134	0.0003	0.0014	-1.2258	0.131
K3	-0.1505	0.6235	0.0004	0.0033	-0.9402	0.157
K4	-0.1394	0.7395	0.0019	0.0018	-1.2704	0.145

Distortion Parameters of color camera						
sensor	k_1	k_2	p_1	p_2	k_3	**Reprojection Error**
K0	0.2663	-0.8656	0.0015	-0.0053	1.0156	0.127
K1	0.2918	-1.0374	-0.0012	-0.0056	1.4310	0.124
K2	0.2914	-1.1027	-0.0002	-0.0009	1.5614	0.129
K3	0.2516	-0.9045	-0.0015	0.0017	1.1420	0.153
K4	0.2380	-0.8270	-0.0010	0.0020	1.0251	0.146

The focal length of the IR camera is larger than that of the color camera, i.e. the color camera has a larger field of view. It is also apparent that every Kinect sensor has slightly different intrinsic parameters. This confirms the need for a formal intrinsic calibration to be

performed on every device to support accurate data registration. The typical focal length value of the IR camera is 580. According to the calibration results, the maximum deviation from this nominal value is 3.25 percent in all five sensors.

The radial distortion in the Kinect sensor is not very significant and hardly noticed in practice. Figure 5.4(a) shows one original image provided by the Kinect color camera. The long green wire holder appears relatively straight, as well as the lines on the checkerboard targets which are more centred in the image. An undistorted image produced with the estimated distortion parameters for the color camera reported in Table 5.1 is shown in Figure 5.4(b). Little transformation can be seen at the edges of the image, but no major effect is perceived over the straight components of the scene. It can therefore be concluded that the lens used in the Kinect color camera performs well within the field of view that it supports, and that distortion estimation and correction might not be required for most applications being supported by this imaging technology.



(a) (b)

Figure 5.4 : Effect of lens distortion in Kinect color camera: a) original image b) undistorted image.

5.2.2 Evaluation of Intrinsic Parameters

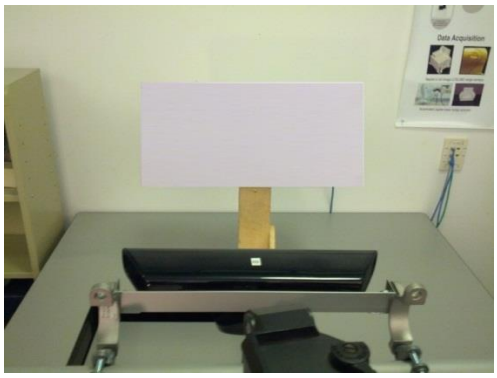
After calibration, both the RGB and IR cameras achieve reprojection error between 0.12 and 0.16 pixel, which is better than the original performance given by the Kinect sensor. The reprojection error without calibration of the IR camera is greater than 0.3 pixel and that of the color camera is greater than 0.5 pixel [46]. Through calibration of the devices, the reprojection

error improves by a factor of two in the case of the IR camera, while for the color camera it improves by a factor of three. The summary of comparison is presented in Table 5.2.

Table 5.2 : Comparison of Kinect camera performance with calibration and without calibration.

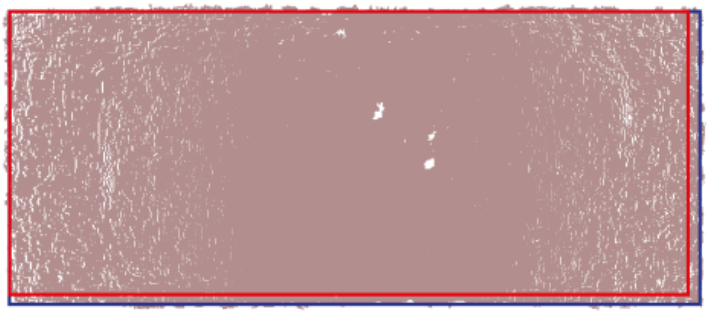
Camera	Average Reprojection Error Without Calibration (Pixel)	Average Reprojection Error After Calibration (Pixel)	Improvement (%)
IR	0.312	0.143	54.2
Color	0.523	0.135	74.2

The quality of the intrinsic calibration method, as discussed above, is evaluated by the reprojection error. Some experiments are also performed to further observe the effect of inaccurate intrinsic parameter estimates during reconstruction. The Kinect sensor provides the depth of each pixel captured via the IR image, but the exact location of the pixel in the X and Y directions depends on the intrinsic parameters. For these experiments, the Kinect sensor is placed in front of a rectangular and planar object of size 60x25cm. The object is kept parallel to the Kinect IR camera image plane. As discussed in chapter 3, the Kinect depth data is more accurate in close range, therefore the object is placed at a distance of 60cm where the quantization step size is about 1mm. First the planar object is imaged using the experimentally obtained intrinsic calibration parameters. The reconstructed model of the object is then projected into the world coordinates. The reconstructed object is shown in Figure 5.5(b), where the red silhouette defines the actual size of the object. We can observe that under these acquisition conditions with fine internal calibration of the IR camera, the reconstructed object is completely surrounded by the silhouette, which reveals reconstruction with an accurate scale. The same experiment is also performed using the default intrinsic parameters encoded in OpenNI [47] and the result is shown in Figure 5.5(c). In this case the reconstructed object is significantly enlarged as compared to the red silhouette defining the size of the original object. The blue silhouette highlights the scaled dimensions, which are increased by 8.6mm in width and 6.2mm in height with the default intrinsic parameters used by OpenNI [47], as detailed in Figure 5.5(d). Therefore, a formal estimation of the intrinsic parameters within any Kinect sensor helps improve the accuracy on the scale of the reconstruction.

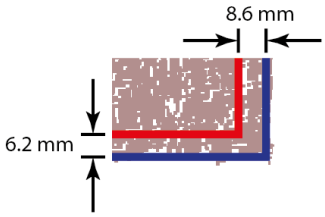


(a)

(b)



(c)



(d)

Figure 5.5 : Reconstruction of a planar target. Red silhouette shows the actual size of the object. a) experimental setup, b) reconstruction using experimental calibration parameters, c) reconstruction using OpenNI default parameters, blue silhouette highlights the extended dimensions, d) difference in the size.

The two evaluations of the estimated intrinsic parameters discussed above, one based on the reprojection error and one based on the dimensions of the reconstruction of an object,

demonstrate the improvements achieved with a formal calibration performed to estimate the intrinsic parameters for every Kinect unit.

5.2.3 Extrinsic Parameters Estimation Between Built-in Kinect Cameras

The respective position and orientation of the color and IR cameras in each Kinect unit can be determined by stereo calibration. The camera calibration method proposed by Zhang's [24] also provides the location of the checkerboard target with respect to a camera coordinate system as shown in Figure 5.6. If the target remains fixed for both cameras, then the geometrical transformation between the cameras is defined by Equation (5.2).

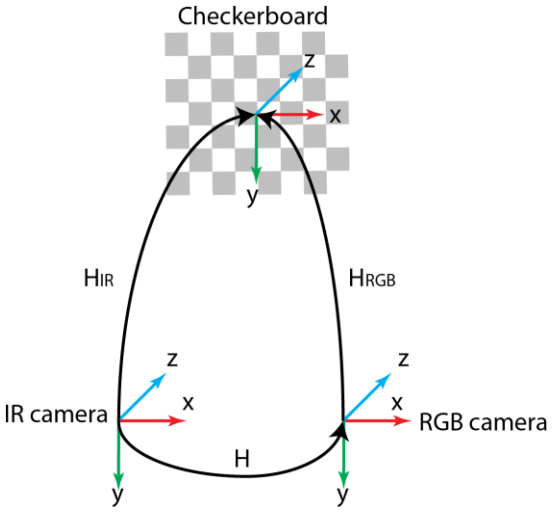


Figure 5.6 : Stereo calibration

$$H = H_{IR}H_{RGB}^{-1} \tag{5.2}$$

where H is the homogenous transformation matrix (consists of a 3x3 rotation matrix, R, and a 3x1 translation vector, T) from the IR camera to the RGB camera, H_{IR} is the homogenous transformation matrix from the IR camera to the checkerboard target, and H_{RGB} is the homogenous transformation from the RGB camera to the checkerboard target. Applying the classical calibration procedure on feature points extracted over the checkerboard calibration target respectively in the color and IR images, the experimentally estimated translation and rotation parameters for the internal extrinsic calibration between the IR and RGB sensors are shown in Table 5.3 for the five Kinect sensors, all *Kinect for Xbox360* type, used in our experiments.

Table 5.3 : Extrinsic calibration of embedded sensors

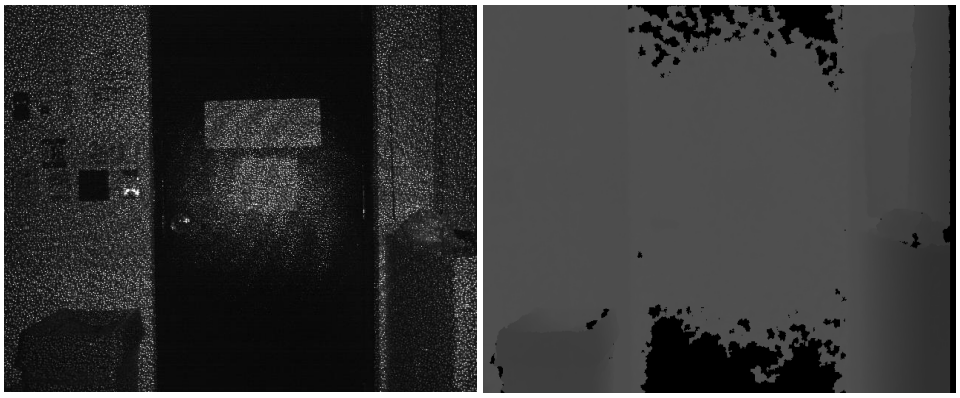
Translation (cm) and Rotation (degree) between IR and RGB						
sensor	T_x	T_y	T_z	R_x	R_y	R_z
K0	2.50	0.0231	0.3423	0.097	0.103	-0.469
K1	2.46	-0.0168	0.1426	0.280	0.183	0.441
K2	2.41	-0.0426	0.3729	0.154	0.372	-0.429
K3	2.49	0.0153	0.2572	-0.263	0.423	0.200
K4	2.47	0.0374	0.3120	0.297	0.200	0.257

The physical separation between the IR and the RGB sensor in every Kinect is about 2.5 cm, which can be validated by measuring the distance between the two cameras on the Kinect sensor. The displacement in y and z directions is difficult to physically measure because it is fairly small, but the experimental results demonstrate only tiny variation in the y (vertical) direction. A slightly larger translation is however noticed along the principal axis of the cameras (z direction). This is because each camera has slightly different focal lengths and the location of the optical center is different inside the Kinect unit for each camera. The typical focal length of the IR camera is 580 pixels and the size of each pixel on the image sensor in real-world units found in the datasheet is 5.2µmx5.2µm. This gives the location of the optical center to be 580*5.2µm = 3.01mm behind the lens of the IR camera. The typical focal length of the color camera is 525 pixels with a pixel size of 2.8µmx2.8µm. This gives the location of the optical center to be 525*2.8µm = 1.47mm behind the lens. The theoretical difference between optical centers in the z direction is therefore 1.54mm, assuming that both lenses are exactly aligned. The results of the calibration are between 1.4 mm and 3.7 mm in the z direction. With respect to orientation, the experiments demonstrate that both sensors are almost perfectly parallel one to each other, as all rotation parameters are inferior to 0.5 degree with respect to each axes. The external calibration parameters allow to accurately relating the color and depth data collected by a single Kinect unit.

5.2.4 Registration of Color and Depth Within a Given Kinect Device

The Kinect sensor does not readily provide registered color and depth images. Once the internal intrinsic and extrinsic parameters are determined for a given Kinect unit, the procedure to merge the color and depth based on the estimated registration parameters is performed as

follows. The first step is to properly relate the IR image and the depth image. The depth image is generated from the IR image but there is a small offset between the two, which is introduced as a result of the correlation performed internally during depth calculation. Figure 5.7(b) shows the null band in the horizontal direction on the right side of the depth image. The width of the null band is about 8 pixels. In the vertical direction the null band is not visible. This is explained by the fact that the IR camera has a resolution of 1280x1024, which comes down to 640x512 when the resolution is reduced by half. The Kinect actually returns a 640x480 raw disparity image, therefore, the band in the vertical direction is clipped off [45] [46] but still exists.



(a) (b)

Figure 5.7 : Shift between the IR and the depth images: a) IR image, b) depth image with black border in the rightmost columns showing the magnitude of the shift.

The offsets in the depth image can be removed by using Equation (5.3), which implements the experimentally obtained offset parameters. The actual offsets are calculated by fitting the IR and depth images on top of each other and by displacing the depth image until it exactly fits with the IR image. The test is performed on a few sets of images and it provides some constant offsets of 5 pixels in the horizontal direction and 4 pixels in the vertical direction. As a result, each pixel of the depth image can exactly map the corresponding pixel in the IR image. Therefore, all the intrinsic calibration parameters estimated for the IR camera can be applied on the depth image.

$$depth(x, y) = depth\_o(x - 5, y - 4) \tag{5.3}$$

where x and y are the pixel location, $depth\_o(x, y)$ is the offseted depth map generated by the Kinect depth sensor, and $depth(x, y)$ is the corrected depth map.

The second step to register color and depth information is to transform both the color and the depth images to compensate for radial and tangential lens distortion using OpenCV [49]. This function estimates the geometric transformation on the images using the distortion parameters and provides the undistorted color image and the undistorted depth image $depth\_ud(x, y)$. Even though it was demonstrated in section 5.2.1 that distortion does not play a major role in the acquisition of images with Kinect sensors, the distortion compensation procedure is preserved in our implementation, in order to achieve the best possible accuracy on the textured 3D reconstruction.

The following step consists in determining the 3D coordinates corresponding to each point mapped in the undistorted depth image. Back projection in the real Cartesian world is achieved using Equations (5.4) to (5.6) and the experimentally estimated intrinsic parameters of the IR camera.

$$X_{IR} = \frac{\left(x - O_{x\_IR}\right) depth\_ud(x, y)}{f_{x\_IR}} \tag{5.4}$$

$$Y_{IR} = \frac{\left(y - O_{y\_IR}\right) depth\_ud(x, y)}{f_{y\_IR}} \tag{5.5}$$

$$Z_{IR} = depth\_ud(x, y) \tag{5.6}$$

where (X_{IR}, Y_{IR}, Z_{IR}) are the 3D point coordinates of a depth image with respect to the IR camera reference frame, (x, y) are the pixel location in the undistorted depth image, (f_{x_IR}, f_{y_IR}) represent the focal length of the IR camera, (O_{x_IR}, O_{y_IR}) is the optical center of the IR camera, and $depth\_ud(x, y)$ is the depth of a given pixel (x, y) in the undistorted depth image.

Next, the corresponding color is assigned from the RGB image to each reconstructed 3D point $P_{IR}(X_{IR}, Y_{IR}, Z_{IR})$. The color is mapped by transforming the 3D point P_{IR} into the color camera reference frame using the internal extrinsic camera parameters, and then reprojecting that point on the image plane of the RGB camera using its intrinsic parameters to find the pixel location in the undistorted color image. This is achieved using Equations (5.7) to (5.9).

$$P_{RGB}(X_{RGB}, Y_{RGB}, Z_{RGB}) = R \,.\, P_{IR} + T \tag{5.7}$$

$$x = \left(\frac{X_{RGB} f_{x\_RGB}}{Z_{RGB}}\right) + O_{x\_RGB} \tag{5.8}$$

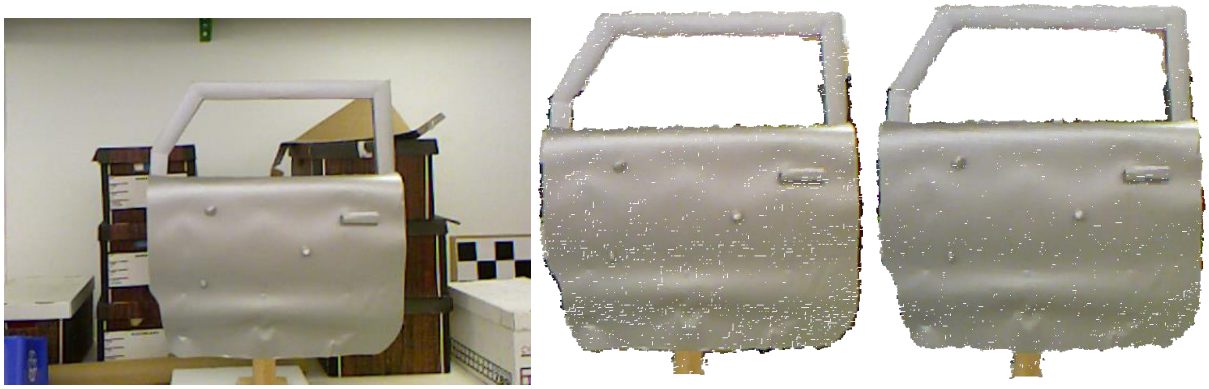
$$y = \left(\frac{Y_{RGB} f_{y\_RGB}}{Z_{RGB}}\right) + O_{y\_RGB} \tag{5.9}$$

where P_{RGB} is the 3D point with respect to the color camera reference frame, R and T are the rotation and translation parameters from the color camera to the IR camera estimated from internal extrinsic calibration, H^{-1} in Equation (5.2), and (x, y) is the location of color information in the undistorted color image.

5.2.5 Evaluation of Registration between Color and Depth

The accuracy of color and depth registration achieved after calibration with the proposed procedure is calculated by the following experiment and it is compared against the color and depth registration that is achieved using the parameters by default encoded in the OpenNI framework. A mockup door is positioned in front of the Kinect sensor at 1.5m. The color of the door and the background wall appeared to be fairly similar in the color image as shown in Figure 5.8(a). Therefore, a scattered scene is created by placing some dark brown color boxes behind the door to better observe the mismatch in the color on the door after registration. First, the door is reconstructed with the registration method defined in OpenNI and the result is shown in Figure 5.8(b). The brown color from some background objects is clearly seen around the 3D textured door model. The maximum size of the mismatch between color and depth along the border of the door is 6.6mm. Second, the door is reconstructed using the method described in section 5.2.4, which makes use of the refined internal intrinsic and extrinsic calibration parameters obtained experimentally. The 3D textured reconstruction generated with the proposed calibration and registration methods is shown in Figure 5.8(c). The improvement on the sharpness of the alignment between color and depth is qualitatively visible. The size of the maximum mismatch is also reduced to 2.3mm, and only appears sporadically over some sections of the contour of the door. The proposed technique therefore demonstrates superior performance to that achieved with hardcoded, device independent, calibration parameters in the OpenNI framework for the Kinect sensor. The random noise

appearing as white dots over the reconstructed door model is an artifact resulting from the capture of 2D images from the 3D point cloud to produce these figures. This noise is not related with the color and depth merging process.



(a) (b) (c)

Figure 5.8 : Evaluation of color and depth registration and fusion: a) color image, b) registration with OpenNI method and default parameters, c) registration with proposed method and experimentally estimated calibration parameters.

Another experiment is conducted to validate the performance of the color and depth registration method, with formal calibration, on a real car which covers a wider scene, as required in the context of this research. Figure 5.9 shows the colored depth information in the range 0-2.5m from the slightly different points of view of the color and IR cameras contained in a same Kinect device. The difference in position and orientation between the two cameras contained in the Kinect unit is accurately compensated for by the estimated extrinsic parameters obtained from internal calibration. The accuracy of the color and depth merging can be seen around the wheel where the color of the wheel and that of the car are clearly separated.

(a) (b)

Figure 5.9 : Registration of color and depth images: a) color image, b) colored depth image.

5.3 External Calibration

The parameters estimated in this phase of the calibration process are the extrinsic parameters, position and orientation, of the IR cameras belonging to every Kinect sensor. The external calibration is performed between pairs of IR cameras for efficiency and accuracy considerations because depth information is generated by default with respect to the IR cameras when working with Kinect RGB-D sensors. However, since internal extrinsic calibration between the color and IR cameras within any Kinect unit has been previously defined, equivalent calibration parameters between the color cameras of different Kinect units can be computed. The concept behind the proposed external extrinsic calibration method consists in determining, for every pair of sensors, the position and orientation of a fixed planar checkerboard in real-world coordinates. Knowing the orientation and center point of the planar target from two different points of view (i.e. two sensors), it is possible to estimate the relative orientation and position change between the sensors.

5.3.1 External Calibration Procedure

The procedure developed for external calibration consists in positioning a standard planar checkerboard target within the overlapping fields of view of any two Kinect sensors. Because of limited overlapping space due to large baseline between the sensors that are combined in a network to ensure coverage over a wide workspace, the target should be placed as close as possible to Kinect sensors within that overlapping space, but not less than 100cm, where the IR projector noise is strong enough to distort the target image. The Kinect provides more accurate data in closer range, therefore setting a target in close range also gives more accurate target points. The result of the method is a rigid body transformation that best aligns the data collected by a pair of RGB-D sensors. The calibration method is suitable for Kinect sensors as it takes advantage of the rapid 3D measurement technology embedded in the sensor and provides registration accuracy within the range of the depth measurements accuracy provided by this technology. Another important advantage of this method is the fact that it is unnecessary to cover the Kinect infrared projector to perform this phase of the calibration, which facilitates manipulations when dealing with the network of Kinect devices.

The choice of target is important to collect a sufficiently large number of points. The overlapping region between two sensors is limited and the size of the target cannot be bigger than the overlapping region. A special target was created that resembles the traditional checkerboard calibration target. The only difference was that the black boxes of the checkerboard target were replaced with augmented reality (AR) tags, as shown in Figure 5.10. This target was experimented with in an attempt to collect a larger number of feature points within the limited overlapping region. Indeed, each AR tag contains four outer corners points and up to eight inner points which allowed to multiply the number of feature points by two or three. However, our experiments revealed that the tags cannot always be detected with the imaging resolution supported by Kinect sensors and under the external noise created by the IR projector. Therefore, this calibration target provided unreliable results. In conclusion, the classical checkerboard target was selected as it ensures reliable and easily detectable features points, even though they appear in smaller number. The format of a grid of 9x7 black-and-white square boxes printed over a regular A3 size paper of 420 x 297 mm, as shown in Figure 5.3, was adopted as it represented the best compromise between the overall size of the calibration target and the number of features points that can be reliably extracted.

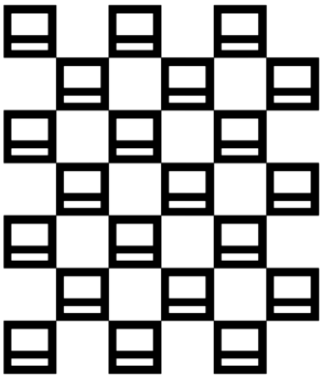
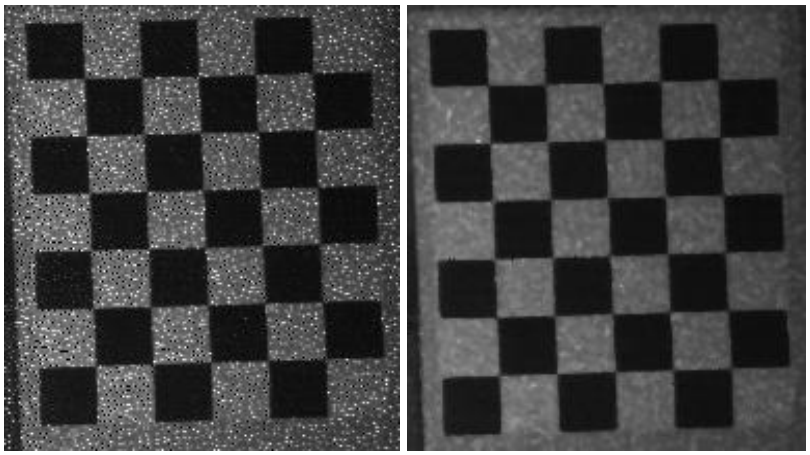


Figure 5.10 : Alternative checkerboard with augmented reality (AR) tags.

The method consists in finding a coordinate frame of the checkerboard target that is located in the center of a target with respect to RGD-B sensors. If the coordinate frame of the checkerboard is known with respect to two Kinect sensors for a fixed target, then these information can be utilized to find the homogenous transformation between both Kinect sensors. The center of the checkerboard gives the translation of the checkerboard frame with

respect to the RGD-B sensors and unit vectors of the coordinate axes provide information about the rotation matrix in between the sensors. The first step is to find the 3D coordinates of the corners of the checkerboard pattern with respect to the IR camera reference frame, using Equations (5.4) to (5.6). When the checkerboard target is positioned in front of a Kinect sensor, the IR projector pattern appears on the checkerboard target as shown in Figure 5.11(a). This pattern creates noise and makes it difficult to extract the exact corners using OpenCV [49]. This noise is similar to salt and pepper noise. A median filter of size 3x3 provides a substantial reduction in the level of noise without blurring the image, as shown in Figure 5.11(b).



(a) (b)

Figure 5.11 : Checkerboard target for external calibration: a) affected by the IR projected pattern, b) filtered IR image using a Median filter of size 3x3.

Moreover, the extracted points are not entirely mapped over a single plane because of the quantization effects in the Kinect depth sensor. Therefore, the corner points of the visible checkerboard are used to estimate a three dimensional plane, Equation (5.10), that minimizes the orthogonal distance between that plane and the set of 3D points. The equation of the plane then permits to estimate the orientation in 3D space of the target with respect to the IR camera.

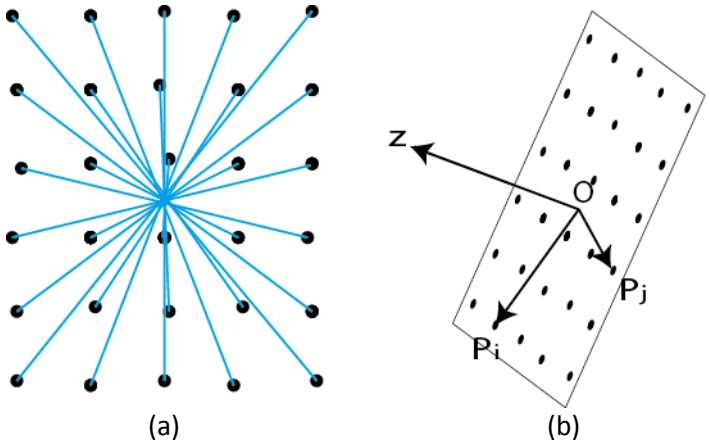
$$z = Ax + By + C \tag{5.10}$$

Let the 3D coordinates of the corners extracted from the checkerboard target be $S_1(x_1, y_1, z_1)$, $S_2(x_2, y_2, z_2)$, …, $S_n(x_n, y_n, z_n)$, then the system of equations for solving the

plane equation are $Ax_1 + By_1 + C = z_1$, $Ax_2 + By_2 + C = z_2$, …, $Ax_n + By_n + C = z_n$. These equations can be formulated into a matrix problem.

$$\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} \tag{5.11}$$

This over-determined system is solved for the values of A, B, and C by well-studied geometrical problem known as the orthogonal distance regression plane problem [50], which provides the best fit plane on those points. All the 3D points, S_n, are projected on the best fit plane as P_n. These points now serve to define the center and the three unit vectors of the coordinate frame of the checkerboard. However, the projected points, P_n, do not represent the exact corners of the checkerboard. Therefore the center cannot be defined by the intersection of only two straight lines, generated from pairs of opposite feature points over the checkerboard pattern, and passing close to the center of the target. Figure 5.12(a) shows the entire set of possible straight lines passing close to the center. The closest point to all the possible intersections between these lines is selected as a center point O.



(a) (b)

Figure 5.12 : a) Possible combinations of feature pairs straight lines passing through the center of the checkerboard, b) the resulting normal vector and the center of a checkerboard target.

Two points P_i and P_j are selected on the plane to define vectors $\overrightarrow{OP_i}$ and $\overrightarrow{OP_j}$. The normal to the plane is then defined by the cross product:

$$z = \frac{\overrightarrow{OP_i} \times \overrightarrow{OP_j}}{\left|\overrightarrow{OP_i} \times \overrightarrow{OP_j}\right|} \tag{5.12}$$

This normal is the unit vector of the z-axis of the checkerboard frame with respect to the RGB-D sensor. The unit vector of the y-axis of the checkerboard can be found by any two vertical points in the checkerboard. Let's P_i and P_j be the two vertical points where P_i is the top end and P_j is the bottom end of a vertical line. N is the total number of possible combinations of vertical lines. The average unit directional vector then can be defined as:

$$y = \frac{1}{N}\sum \frac{P_i - P_j}{\left|P_i - P_j\right|} \tag{5.13}$$

This unit vector is the unit vector of the y-axis of the checkerboard frame with respect to the RGB-D sensor. The last unit vector for the x-axis can be found by a cross product, defined as:

$$x = \mathrm{y} \times \mathrm{z} \tag{5.14}$$

All the unit vectors of the coordinate frame of the checkerboard target can be combined to define the rotation matrix between the RGB-D sensor and the checkerboard frame as:

$$R = \begin{bmatrix} x_x & y_x & z_x \\ x_y & y_y & z_y \\ x_z & y_z & z_z \end{bmatrix} \tag{5.15}$$

The translation is the same as the location of the center of the checkerboard frame.

$$\mathrm{T} = [O_x \quad O_y \quad O_z] \tag{5.16}$$

R and T are the rotation and the translation of the checkerboard frame with respect to the Kinect IR sensor. The position and orientation between two Kinect sensors can be determined by the following procedure. Let H_1 and H_2 be the homogenous transformations between Kinect 1 and the checkerboard and between Kinect 2 and checkerboard respectively, as shown in Figure 5.13. If the target remains fixed for both Kinect sensors, then the geometrical transformation between the sensors is defined as follows:

$$H = H_2 H_1^{-1} \tag{5.17}$$

where H is the homogenous transformation matrix from the Kinect 2 to the Kinect 1 sensor.

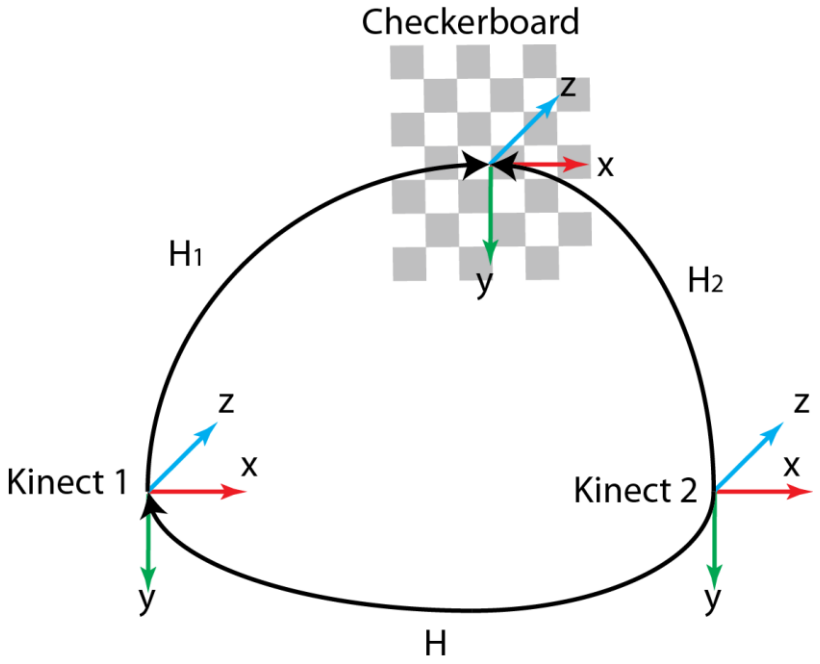


Figure 5.13 : Extrinsic calibration

5.3.2 External Calibration Method Implementation

The external calibration method described in the previous section requires depth information from the sensor and is only applicable if the depth of each point over the target is known. This method is suitable for RGB-D sensors like the Kinect. The only important limitation to the method is that the target must be located sufficiently far from the camera. When this condition is not satisfied, two problems can be observed. When the calibration target is closer than the minimum depth of field of the sensor, that is within 50 cm for the Kinect sensor, then no depth data can be collected. The second problematic situation observed is when the calibration target is placed in close proximity to the minimum depth of field of the sensor that is between 50cm and 100cm from the Kinect. Then the IR projector noise tends to be strong enough for the median filter to fail at generating a sharp enough IR image for the corners detector to reliably extract the feature points required to perform extrinsic calibration.

Fortunately, these cases do not appear with the multi-camera system designed here to operate over a large volume.

This method is tested in the laboratory on a network comprised of four Kinect sensors. Three sensors, *K0*, *K1* and *K3*, are separated with large baselines and placed at similar height of 1.65m above the ground, as shown in Figure 5.14. The baseline between Kinects *K1* and *K0* is 1.3m, and between *K0* and K3 is 2.1m. The viewing angle is almost parallel to the ground for Kinects *K0*, *K1* and *K3*. To monitor the performance of the calibration method, the fourth Kinect, *K2*, is placed close to Kinect *K1* but at a different height of 1.2m above the ground. The baseline between *K1* and *K2* is 0.46m and the viewing angle is not parallel to the floor. The setup for the experiment is shown in Figure 5.14 from a lateral and a back view, respectively, with the checkerboard target also visible in the view from the back of the setup.

(a) (b)

Figure 5.14 : Setup for validating the proposed extrinsic calibration method: a) side view showing four Kinect sensors, b) back view showing three Kinect sensors and the calibration target.

The number of blocks in the checkerboard is 6x7 and the size of each block is 9x9cm. The Kinect K1 is set as the base reference frame and the position and orientation of the other Kinect units is estimated with respect to *K1*. The external calibration procedure is divided in three steps. In first step the calibration is performed between Kinects *K1* and *K0* by placing the target in front of both cameras at a distance of 1.8m. External incandescent lamps are used to illuminate the target for better visibility in the IR camera. The calibration method described in section 5.3.1 is applied. The results give the transformation from Kinect *K1* to *K0*. Next, the calibration between Kinects *K1* and *K2* is estimated in the same manner. The target is not moved because it is also visible from Kinect *K2* in its initial location. Finally, the target is moved

to be visible simultaneously by Kinects *K0* and *K3* and the calibration procedure is repeated over that pair of sensors to provide the transformation from *K0* to *K3*.

Since Kinect *K1* serves as a reference base for the multi-camera vision system, the calibration is further computed with respect to that base. Kinects *K0* and *K2* have a direct relationship with *K1*, immediately defined by the extrinsic calibration parameters obtained in each case, but *K3* needs to be related to *K1* through an intermediate node, *K0*. The relation (*K1*, *K3*) is given by the following equation:

$$H_{K1\leftarrow K3} = H_{K1\leftarrow K0}H_{K0\leftarrow K3} \tag{5.18}$$

The physical separation between the devices is already known, as it is measured during setup using a measuring tape. Although these are not absolutely exact measurements, they can be used as a reference to compare the results from extrinsic calibration. The comparison is shown in Table 5.4. The experimentally estimated distance is close to the reference distance with an error lower than 2% in all cases. The Kinects *K0* and *K3* are almost at the same height with their optical axis direction parallel to the floor, while *K3* is rotated around 60 degrees with respect to Kinect *K1*'s y-axis. This reference value for the rotation can be compared to the experimental result where the rotation around y is estimated at 63.55 degrees, with minor rotations around other two axes.

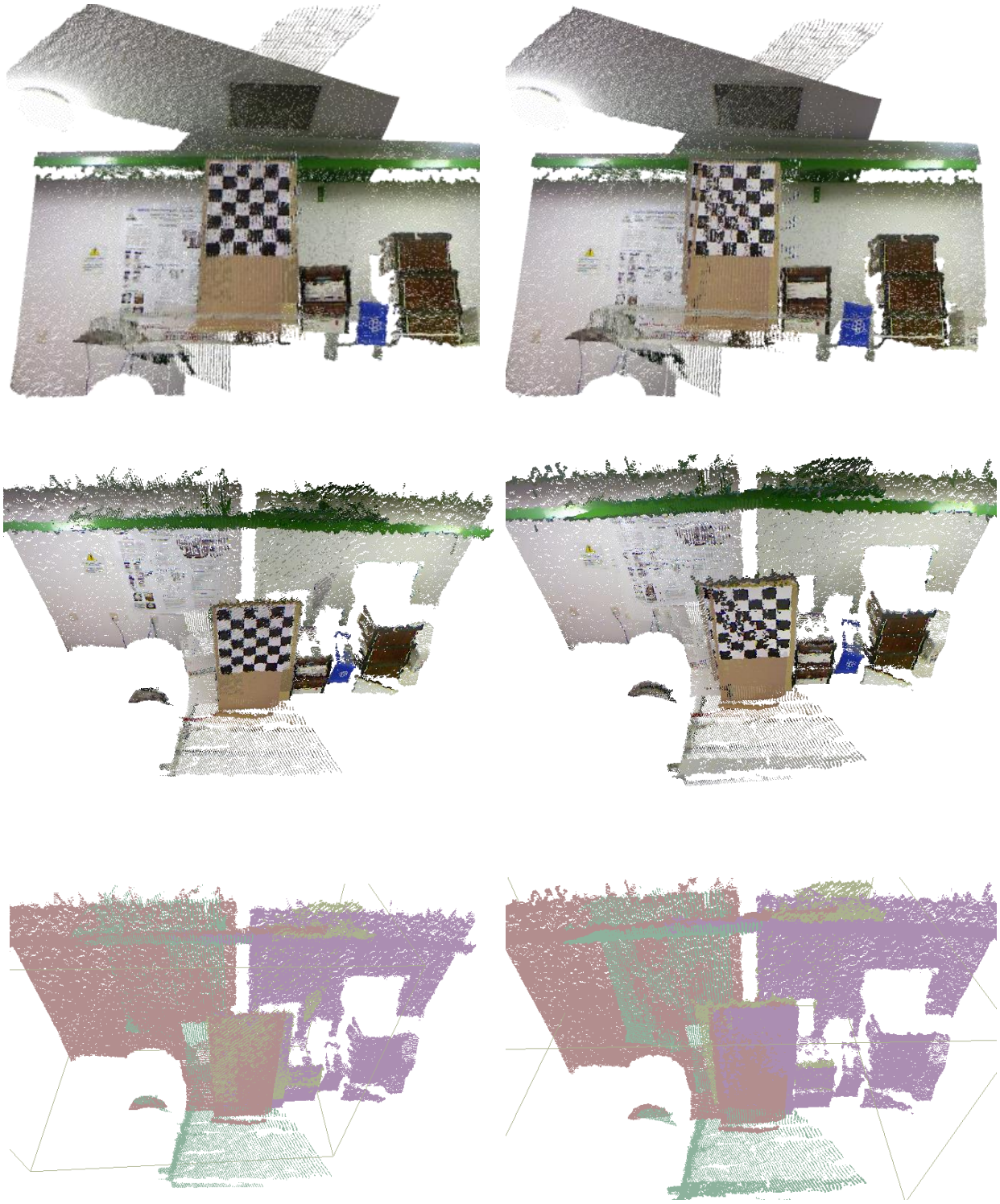
Table 5.4 : Extrinsic calibration of the network of RGB-D sensors.

Translation (cm) and Rotation (degree) between two Kinect IR sensors									
sensor	***T_x***	***T_y***	***T_z***	***R_x***	***R_y***	***R_z***	***Distance estimated (cm)***	***Distance measured (cm)***	***Difference in %***
K0->K1	-131.23	6.73	0.76	-0.51	3.40	-1.74	131.41	130.0	1.08
K3->K0	-189.39	0.61	95.99	2.39	63.55	2.46	212.33	210.0	1.11
K2->K1	-2.62	43.41	-17.28	19.45	-22.77	4.24	46.80	46.0	1.73

During calibration the same checkerboard target is also used to determine the extrinsic calibration parameters between the IR cameras of the different Kinect sensors, but using Zhang's camera calibration method. The Zhang's method only relies on the IR image without

considering any depth data. The results from Zhang's method are used to compare against the results obtained with the proposed method which also uses the depth data. The whole scene is reconstructed as a textured 3D model using the extrinsic parameters of proposed method on one side, and using the Zhang's method on the other hand. The four different views of the Kinect color cameras are shown in Figure 5.15. The reconstruction of the scene in Figure 5.16(a) is achieved with the proposed method, while that in Figure 5.16(b) is obtained using Zhang's calibration scheme. In Figure 5.16, three different views of the model of the same scene are presented. The first two views show the reconstruction with the color mapping and the last one shows the output in four different colors. Each color represents one Kinect. The reconstruction obtained with Zhang's method does not provide as good an alignment as the proposed procedure between the four outputs of the sensors. The limited alignment is clearly seen from the long green wire holder shown in the second picture of Figure 5.16(b). The reconstructed wire holder is better aligned and straighter with the proposed method as shown in Figure 5.16(a). A similar impact is visible on the checkerboard object that appears duplicated when textured 3D models are merged following an external calibration based on Zhang's approach, as shown in Figure 5.16(b). This effect is compensated for with the proposed procedure, as seen in Figure 5.16(a).

Figure 5.15 : Color images captured by Kinect sensors K0, K1, K2 and K3 respectively.



(a) (b)

Figure 5.16 : Comparison of the extrinsic calibration results from 3D textured reconstructions of the scene. Three different views of the scene are presented: a) with proposed method, b) with Zhang's calibration method.

5.3.3 Evaluation of Possible Refinement on the External Calibration

A common method to evaluate extrinsic calibration is to compare some ground truth depth data with the reconstructed data obtained with a calibrated network of cameras. Because of the unavailability of such ground truth data, this is often difficult to perform. The evaluation presented here rather capitalizes on the bases of the ICP algorithm [9]. The ICP algorithm iteratively registers point clouds, but also requires user control to properly align those point clouds. To handle this requirement, the point clouds shown in Figure 5.16, which are already registered with either the proposed method or Zhang's technique, are further treated with an implementation of the ICP algorithm in MeshLab [51]. The improvement provided by the ICP algorithm to refine the alignment between the point clouds can be seen as a representation of the remaining error after the formal calibration is used to register the point clouds. The output of the ICP algorithm represents the transformation matrix between two point clouds. These improvements on the translation and rotation angles over the alignment already provided by extrinsic calibration parameters are reported in Table 5.5. The translation correction performed by ICP on point clouds previously registered with Zhang's camera calibration method is between 6cm and 12cm, while when the parameters obtained with the proposed method are used as seed values to the ICP algorithm, the correction on translations reduces to values between 1cm and 3cm for all pairs of Kinect sensors. The rotational error correction is lower than 0.56 degree for the proposed method as compared to values up to 4.05 degrees with Zhang's camera calibration method.

Table 5.5 : Comparison of corrections on calibration parameters estimated with ICP algorithm for two calibration methods.

	ICP-based corrections to Zhang's calibration parameters			**ICP-based corrections to proposed calibration parameters**		
	Corrections on translation (cm) [x y z]	**Distance (cm)**	**Corrections on rotation (degree) [Rx Ry Rz]**	**Corrections on translation (cm) [x y z]**	**Distance (cm)**	**Corrections on rotation (degree) [Rx Ry Rz]**
K0->K1	[-5.4 1.0 -3.3]	6.41	[0.94 1.39 -1.87]	[-1.1 0.3 -0.5]	1.25	[0.19 0.33 -0.54]
K3->K0	[-9.3 4.1 1.7]	10.31	[1.5 0.86 -0.38]	[1.5 2.4 0.2]	2.84	[0.36 -0.30 0.56]
K2->K1	[-9.7 -4.2 -4.8]	11.61	[-1.87 4.05 -1.02]	[-1.6 1.3 -0.9]	2.25	[0.39 0.4 -0.04]

5.4 Calibration of a Network of RGB-D Sensors with a Robotic Manipulator

The final stage in the design and implementation of the multi-camera system consists of registering the robotic manipulator with the network of RGB-D cameras in order to achieve accurate robotic operation under visual guidance. In the previous sections, the calibration of the multi-camera system was presented, where every RGD-B camera is calibrated with respect to a global frame of reference. This global frame of reference is set within one of the RGB-D cameras and the respective position and orientation of all other cameras are defined with respect to that reference camera. The introduction of the robot into the system brings the global reference frame into the base of the robotic manipulator, since all operations must be defined with respect to that base for the robot to understand and perform the desired movement. The only calibration required between the robot and the peripheral vision system is therefore between the robot base reference frame and the global reference frame of the RGB-D cameras network. The estimation of these extra calibration parameters will support the transformation of all the visual data into the robot frame of reference, which will then be used by path planning and obstacle avoidance algorithms to guide the robot while performing tasks under visual guidance.

5.4.1 Setup for Calibration

The system framework for robot calibration with the RGD-B sensors is shown in Figure 5.17. The global reference frame for the entire system is set at the base of the robot. The robot takes an input defined with respect to its base which specifies the position and orientation of the tool plate through forward kinematics. In order to determine the needed location of the robot, the position and orientation of the tool plate needs to be estimated with respect to the global camera reference frame. As described in section 5.3.1, the position of a calibration checkerboard can be found with respect to a camera frame. Therefore, a checkerboard calibration target is precisely attached on the tool plate of the robot, such that the tool plate and the checkerboard target share a common frame of reference. This completes the transformation graph in between the robot base and the tool plate reference frames (defined

by robot's kinematic model), and in between the tool plate and the camera reference frames (defined by a vision system calibration procedure), as shown in Figure 5.17. The position and orientation of the reference Kinect sensor can be then calculated with respect to the base of robot, which provides calibration between the vision stage and the robotic platform.

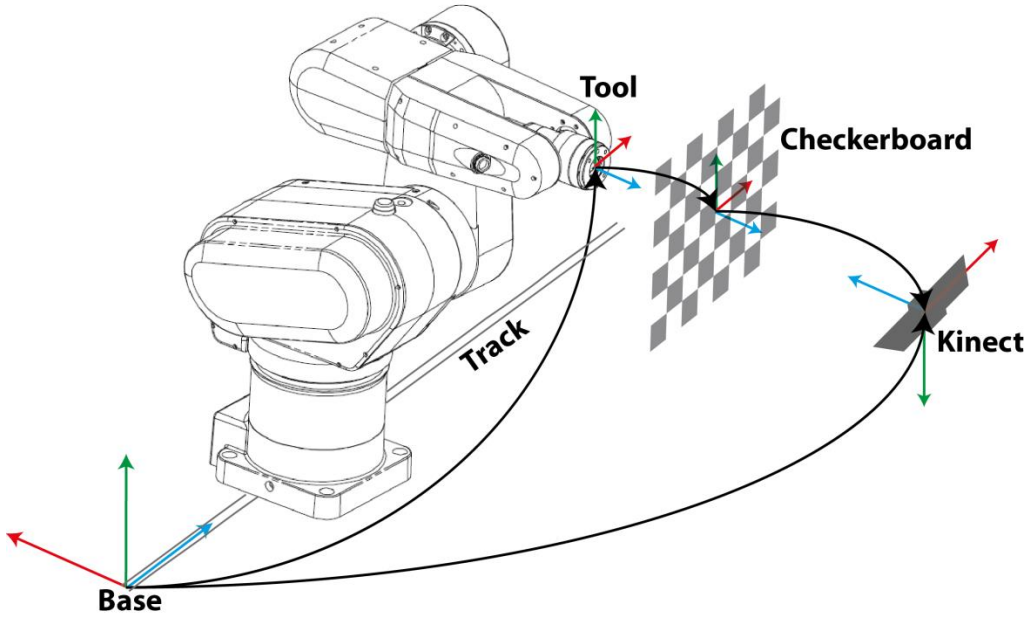


Figure 5.17 : Frame transformations between robot, calibration target, and reference Kinect sensor.

The calibration process is divided into three steps. The first step is to align the checkerboard calibration pattern on the tool plate, which reduces the transformation between the tool reference frame and the checkerboard target. Practically, it is not possible to exactly align the target. Therefore the transformation between the tool and the checkerboard can be defined as:

$$H_{Tool \leftarrow Checkerboard} = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \tag{5.19}$$

Where $H_{Tool \leftarrow Checkerboard}$ is the transformation from the end effector of the robot to the checkerboard, R is the rotation sub-matrix, and T is the translation vector which compensates for the thickness of the target.

In the second step, the transformation from the base of the robot to the tool plate is calculated using the forward kinematic model of the robot. The last step consists of computing

the position of the checkerboard calibration target with respect to the Kinect sensor, which is achieved with the calibration procedure presented in section 5.3. The complete transformation between the robot base and the reference Kinect sensor can therefore be defined as:

$$H_{Base \leftarrow Kinect} = H_{Base \leftarrow Tool} H_{Tool \leftarrow Checkerboard} H_{Checkerboad \leftarrow Kinect} \tag{5.20}$$

where $H_{Base \leftarrow Kinect}$ is the frame transformation from the robot base to the Kinect's IR camera reference frame, $H_{Base \leftarrow Tool}$ is the transformation from the robot base to its end effector, $H_{Tool \leftarrow Checkerboard}$ is the transformation from the robot end effector to the checkerboard reference frame, and $H_{Checkerboard \leftarrow Kinect}$ is the transformation from the checkerboard target to the Kinect's IR sensor reference frame.

5.4.2 Checkerboard Target Design and Alignment on Tool Plate

A meticulous alignment of the checkerboard on the tool plate of the robot contributes to reduce the error introduced in the robot-vision calibration procedure. Figure 5.18(a) shows the CAD diagram of the robot's end effector tool plate. Four threated holes are available on the end effector to attach the target. The CAD diagram is first printed on letter size paper, while keeping the center of the diagram exactly in the center of the paper. This way, the XY coordinates of the end effector get aligned with the center of the paper. The latter is then attached on a hard wooden sheet, where holes are drilled according to the diagram. These holes help to attach the checkerboard target on the tool plate of the robot. The size of the wooden sheet preferably matches the size of the paper, such that the center of the sheet is centered with the end effector. The calibration target with a 8x7 checkerboard pattern is printed on another piece of letter size paper. The size of each block in the checkerboard is set to 3cm. The center of the checkerboard, as shown in Figure 5.18(b), is printed to align with the center of the paper, as this is done for the CAD diagram of the end effector. This paper is then directly attached on top of the wooden sheet, where the CAD paper is already attached. This ensures alignment between the center of the checkerboard calibration target and the center of the end effector tool plate.

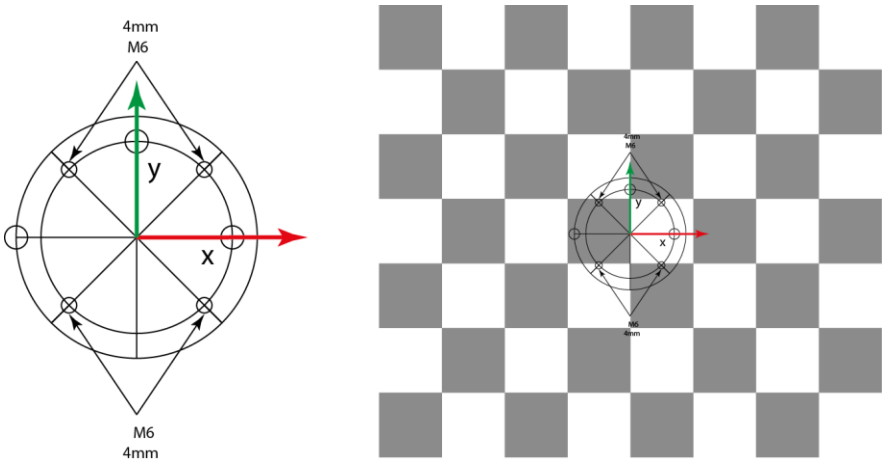


Figure 5.18 : Alignment of the checkerboard on the robot end effector.

The four threated holes in the wooden sheet help keep the calibration target rigidly in place and eliminate any rotation between the two reference frames, as well as any translation in the XY plane. The only translation present is due to the thickness of the calibration target, *w*. This transformation can be defined as:

$$H_{tool \leftarrow checkerboard} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & w \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.21}$$

where $H_{tool \leftarrow checkerboard}$ is the transformation from the end effector to the checkerboard target, and *w* is the thickness of the board that is 0.5cm. Figure 5.19 shows the checkerboard attached to the end effector.

Figure 5.19 : Robot carrying a checkerboard target during vision-robot calibration.

5.4.3 Calibration of Robot with Kinect

In order to perform the vision-robot calibration procedure, the robot end effector is brought in front of the Kinect sensor and the distance between the sensor and the end effector is set to at least 50cm, which represents the minimum depth of field of the Kinect depth sensor. After positioning the robot, the transformation from the robot base to the end effector is determined using the robot forward kinematics. Figure 5.20 shows the robot with all its joints. The robot is mounted on a 2-meter linear track to translate in the workspace. In the home position, the base, which is located at one end of the track, and joint1 (J1) are on the same location. The complete forward kinematic from the base to the tool is defined in Equations (5.22) to (5.29).

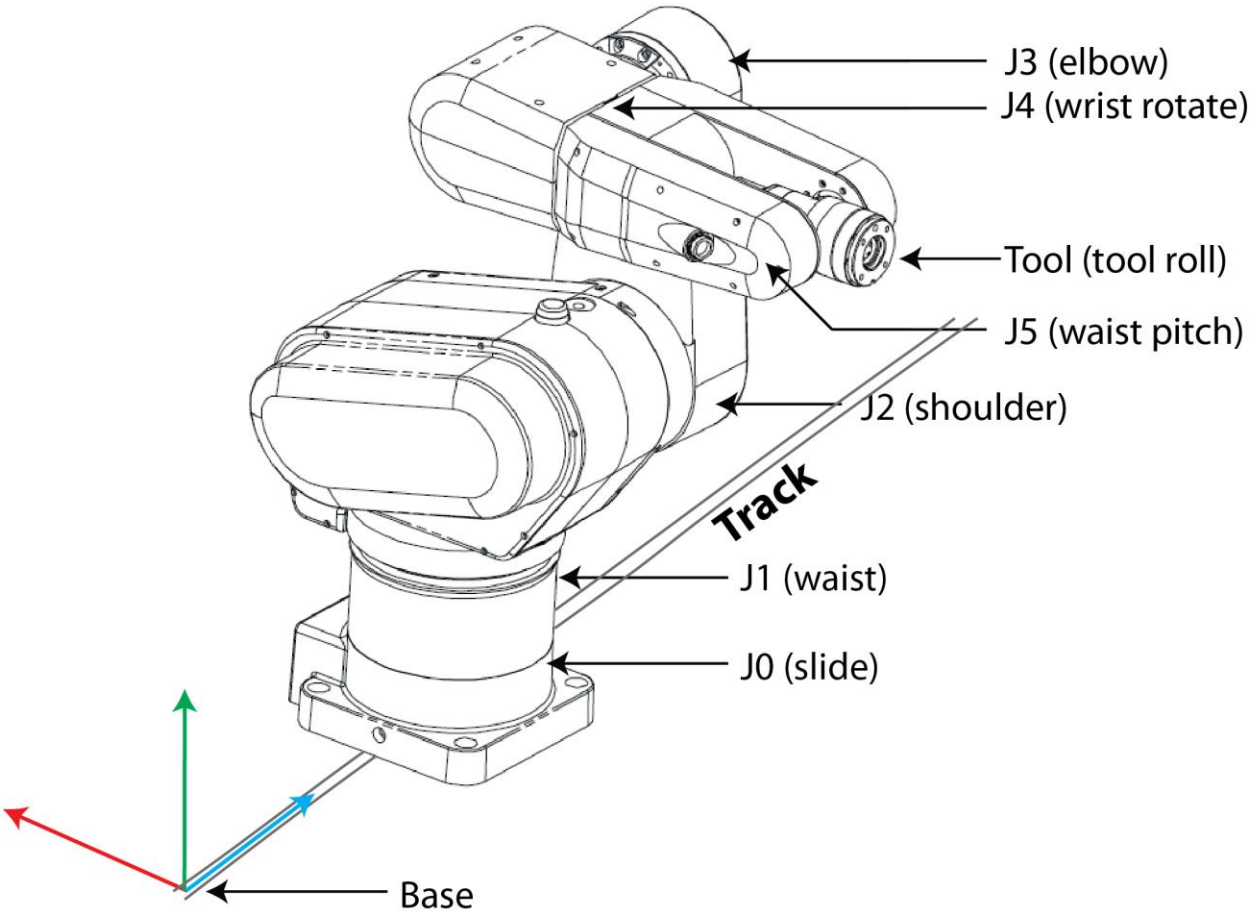


Figure 5.20 : CRS-F3 robotic system.

$$H_{Base\leftarrow J0} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & D \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.22}$$

$$H_{J0\leftarrow J1} = \begin{bmatrix} cos(\theta_1) & 0 & sin(\theta_1) & 100 * cos(\theta_1) \\ sin(\theta_1) & 0 & -cos(\theta_1) & 100 * sin(\theta_1) \\ 0 & 1 & 0 & 350 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.23}$$

$$H_{J1\leftarrow J2} = \begin{bmatrix} cos(\theta_2) & -sin(\theta_2) & 0 & 270 * cos(\theta_2) \\ sin(\theta_2) & cos(\theta_2) & 0 & 270 * sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.24}$$

$$H_{J2\leftarrow J3} = \begin{bmatrix} cos(\theta_3) & 0 & sin(\theta_3) & 0 \\ sin(\theta_3) & 0 & -cos(\theta_3) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.25}$$

$$H_{J3\leftarrow J4} = \begin{bmatrix} cos(\theta_4) & 0 & sin(\theta_4) & 0 \\ sin(\theta_4) & 0 & -cos(\theta_4) & 0 \\ 0 & 1 & 0 & 265 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.26}$$

$$H_{J4\leftarrow J5} = \begin{bmatrix} cos(\theta_5) & 0 & sin(\theta_5) & 0 \\ sin(\theta_5) & 0 & -cos(\theta_5) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.27}$$

$$H_{J5\leftarrow Tool} = \begin{bmatrix} cos(\theta_6) & -sin(\theta_6) & 0 & 0 \\ sin(\theta_6) & cos(\theta_6) & 0 & 0 \\ 0 & 0 & 1 & 75 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.28}$$

$$H_{Base\leftarrow Tool} = H_{Base\leftarrow J0} H_{J0\leftarrow J1} H_{J1\leftarrow J2} H_{J2\leftarrow J3} H_{J3\leftarrow J4} H_{J4\leftarrow J5} H_{J5\leftarrow Tool} \tag{5.29}$$

where D is the translation between the base reference and joint *J0,* which is the length of translation of the robot along the linear track. θ_1 to θ_6 are the joint angles of the joints, J1 to J6, respectively.

This transformation is then multiplied by the transformation between the tool and the checkerboard which compensates for the thickness of the checkerboard, as defined in Equation

(5.21). Finally the result is multiplied by the transformation between the checkerboard and the camera, which is estimated by the proposed extrinsic calibration method detailed in section 5.3. The transformation provides the position of the reference Kinect IR camera with respect to the robot base reference frame. The transformation between the base of the robot and the camera frame is independent from the location of the checkerboard, therefore, the checkerboard target can be moved over the entire field of view of the reference Kinect sensor. The target is moved sequentially to 15 different locations between 50cm and 80cm from the sensor. The size of the target is small, therefore the corners of the checkerboard are not reliably detectable for larger distances. For each view the location of the base of the robot is estimated with respect to the reference Kinect sensor. Averaging is performed over the 15 different transformations to estimate the final transformation. As a result, this extra calibration procedure formally links the reference Kinect sensor to the robot base. Given the entire intrinsic and extrinsic calibration conducted previously on the network of Kinect sensors, it is then possible to navigate the robot from the visual and depth information collected from any Kinect sensor in the network.

5.5 Evaluation of the Calibration between RGB-D Sensors and Robot

To evaluate the calibration between the vision stage and the robot achieved with the proposed method, the robot is operated to move and point with a stick to certain locations initially measured by the Kinect sensors in the network. The robot movement being more precise than the depth and spatial resolution of the Kinect sensors, the error originating from the robot itself in the pointed position is considered negligible for this evaluation.

In this experiment a few points are selected in the vicinity of the robot workspace. Selected points are shown in Figure 5.21. Points 1 and 2, in Figure 5.21(c), are selected from the depth map generated by the reference Kinect *K1*, which is calibrated with the base of the robot. Points 3 and 4, in Figure 5.21(b), are selected from the depth map of Kinect *K0*, which relates to the base frame through an intermediate node, *K1*. Lastly, points 5 and 6, in Figure 5.21(a), are selected from Kinect *K3*, which relates to the base frame through Kinects *K0* and *K1*. All the points are around 2m away from their respective Kinect sensor.

(a) (b) (c)

Figure 5.21 : Feature points locations for the robot to reach: a) points defined from Kinect K3, b) points defined from Kinect K0, and c) points defined from Kinect K1.

For this experiment, the checkerboard mounted on the robot tool plate during the vision-robot calibration procedure is replaced by a pointer stick of length 26cm. This length is taken into account in the robot control model in a similar way to the adjustment performed for the thickness of the calibration checkerboard. Once the scene feature points are transformed in the base reference frame of the robot, series of commands are sent to the robot to move the pointer stick toward the selected features, one at a time. For simplicity, during the process, the orientation of the pointer is set parallel to the ground. Figure 5.22(a) shows the robot pointing towards point 3. The final location of the end effector does not exactly points on the features in the real world. This is due in part to the quantization error, which is around 1cm at 2m distance from the Kinect sensor, and to residuals in the calibration process. The magnitude of the error in the pointing location is estimated by sending some incremental commands to move the robot on the exact location in the real world as shown in Figure 5.22(b). These incremental commands provide an estimate of the residual distance between the end effector and the real-world feature location. The errors for all six features are presented in Table 5.6.

(a) (b)

Figure 5.22 : Moving robot towards selected points with respect to robot base: a) reconstructed point b) exact point.

Points 1 and 2 are directly related to the base of the robot and are therefore more precise than the other points that also depend on intermediate transformations among the network of RGB-D sensors. As expected, residual errors on the transformations contribute to the error on the estimation of the location of the actual points. However, the maximum error is overall less than 2.6cm which is coherent with the relative quality of the data provided by the Kinect sensor. The maximum error is observed in the x direction for Points 1 to 4 with respect to base of the robot, which is approximately the z direction for Kinects *K0* and *K1*. Here the quantization of the Kinect depth sensor is contributing towards the z direction.

The goal of the RGB-D imaging system is to provide a rapid 3D vision acquisition platform to guide the general movement of a robotic inspection system toward selected regions of interest over large objects, and given that the robotic platform will be further assisted by the extra proximity and touch sensing stages for fine interaction of the robot with those objects, as described in section 4.2, the magnitude of the errors reported here is considered tolerable. The robot end effector is equipped with a proximity/touch sensing layer that uses SHARP GP2D120X proximity sensors with a maximum depth of field of up to 40cm. Therefore a tolerance of 2.6cm in the vision stage can easily be accommodated.

Table 5.6 : Evaluation of the vision-robot calibration.

Point	Residual distance (cm) [x y z]	Magnitude (cm)
1	[0.8 0.3 0.6]	1.04
2	[1.0 0.3 0.5]	1.15
3	[2.1 0.5 1.4]	2.57
4	[1.9 0.6 1.1]	2.27
5	[0.7 0.7 1.8]	2.05
6	[0.9 0.7 1.9]	2.21

5.6 Summary

This chapter presented the procedure developed and implemented to calibrate a network composed of multiple RGB-D sensors along with a robotic arm. The intrinsic parameters were first computed using a checkerboard calibration target for every color and IR camera of the Kinect sensors in the network. The intrinsic calibration improved the reprojection error of the color cameras by a factor of two and that of the IR cameras by a factor of three when compared with data processing performed using default parameters encoded in the OpenNI framework. It also demonstrated the variability of those parameters among different Kinect units of the same model. A procedure for internal extrinsic calibration and merge of color and depth data was also introduced, which demonstrated improved alignment of the measurements collected within a single Kinect unit, when compared to OpenNI methods. A method for extrinsic calibration in between separate Kinect units distributed over a network in which some partial overlap exists in between the respective fields of view of the sensors was also presented. It takes advantage of the 3D measurement technology embedded in the sensors and provides registration accuracy within the range of the depth measurements accuracy provided by the Kinect technology. Finally, a complete integration of the multi-sensor vision stage with an industrial robotic manipulator was performed and experimentally evaluated for its relative accuracy when operating the robot under visual guidance provided by the RGB-D sensors.

Chapter 6. Experimental Validation in Field Operation

This chapter presents the implementation of the multi-camera vision system in a practical application for automotive vehicle imaging where it covers a large working volume within which it serves to capture textured 3D data, taking advantage of the integration and registration between multiple RGB-D sensors distributed in a network and operating collaboratively. It supports an experimental evaluation of performance in the targeted field operation.

6.1 Setup

The acquisition stage was installed in an underground parking garage, where direct or indirect sunlight cannot impact on the sensors. The motivation for this testing environment is that the Kinect sensor cannot acquire depth data under bright ambient light, because the IR pattern that it projects to create the depth image gets mixed up with the daylight and the pattern becomes unrecognizable. The setup of the system is shown in Figure 6.1.

Figure 6.1 : Multi-camera acquisition platform setup in real operating environment.

All Kinect cameras are positioned as proposed in chapter 4. Upon initial installation, there may be some error in the positioning of the sensors, which result in clipping of some parts of the vehicle. Therefore, after the initial setup and before actual calibration is performed, the vehicle is parked inside the working volume and the sensors position and orientation are slightly adjusted to provide a complete view the parts of interest, and ensuring that the entire height of the vehicle is covered. In this configuration, Kinects *K0*, *K1* and *K2* are capturing the

lateral part of the vehicle and Kinects *K3* and *K4* are covering the partial front and back regions of the vehicle, respectively. Kinect *K1* is set as the base reference frame for the vision system, because it is located in the center of the acquisition platform and no more than two intermediate transformations are required to reach any other Kinect's frame of reference, which makes data registration efficient.

6.2 Network Calibration

All the Kinect sensors used in the system can be pre-calibrated for their internal parameters, which include the intrinsic parameters for the IR and the color cameras respectively, and the internal extrinsic calibration between the IR and the color cameras. After the vision stage is properly configured to cover the entire vehicle, ideally the largest model to be inspected, all the RGB-D cameras are calibrated in pairs for their external extrinsic parameters between respective IR sensors. The checkerboard target is successively placed in the overlapping regions of the pairs of cameras. The size of the checkerboard target used in this process is 56x65cm with a pattern of 6x8 blocks measuring 9x9cm each. The approximate distance between the calibration target and the RGB-D sensor during calibration is 1.8m. The external calibration is estimated between the IR sensors of every pair of Kinect sensors using the proposed method discussed in section 5.3. Figure 6.2 shows the calibration target placed inside the overlapping region between Kinects *K0* and *K4* during calibration. The calibration time between a pair of sensors is around 5 sec, which includes the collection time of data sequentially by both sensors due to interference issues discussed in section 4.6.

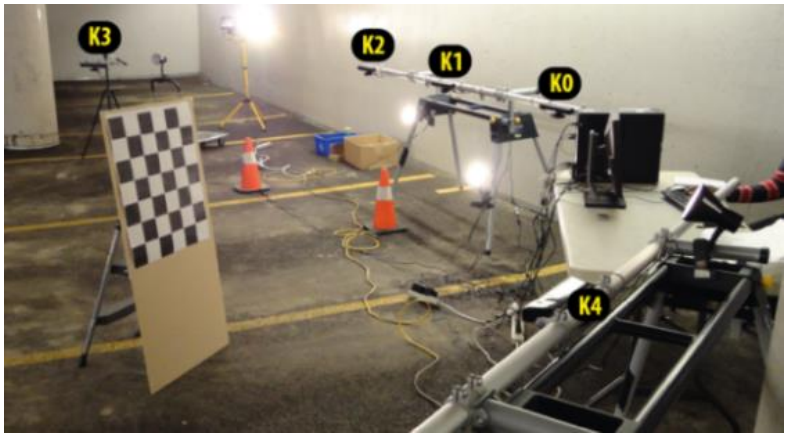


Figure 6.2 : Placement of calibration target during calibration of Kinects K0 and K4.

The external calibration process is divided into four steps. In each step the calibration between a pair of Kinect sensors is estimated. The calibration pairs are (*K0*, *K4*), (*K1*, *K0*), (*K1*, *K2*) and (*K2*, *K3*). The calibration flow is shown in Figure 6.3. Since Kinect *K1* serves as the base reference frame for the multi-camera vision system, the calibration is further computed with respect to that base. Kinects *K0* and *K2* have a direct relationship with *K1*, immediately defined by the extrinsic calibration parameters obtained in each case, but *K4* and *K3* need to be related to *K1* through an intermediate node, respectively *K0* and *K2*. The relations between (*K1*, *K4*) and (*K1*, *K3*) are given by the following equations.

$$H_{K1\leftarrow K4} = H_{K1\leftarrow K0}H_{K0\leftarrow K4} \tag{6.1}$$

$$H_{K1\leftarrow K3} = H_{K1\leftarrow K2}H_{K2\leftarrow K3} \tag{6.2}$$

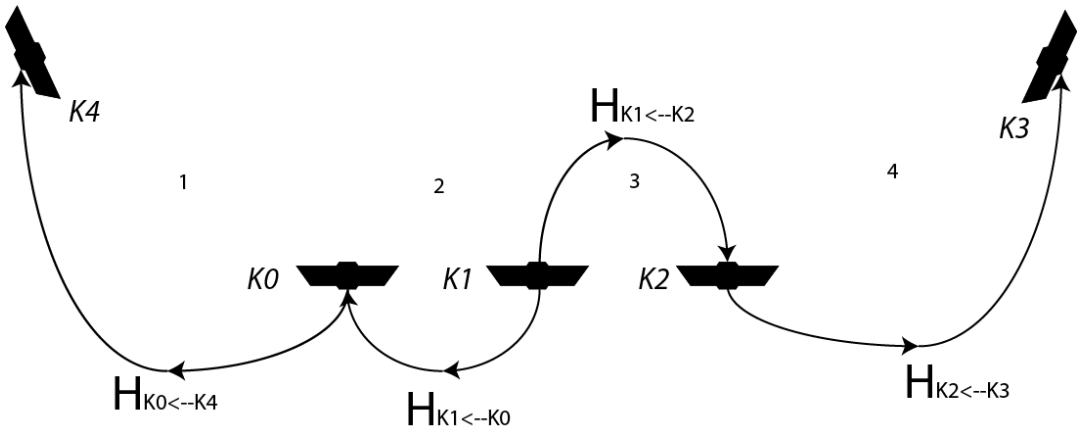
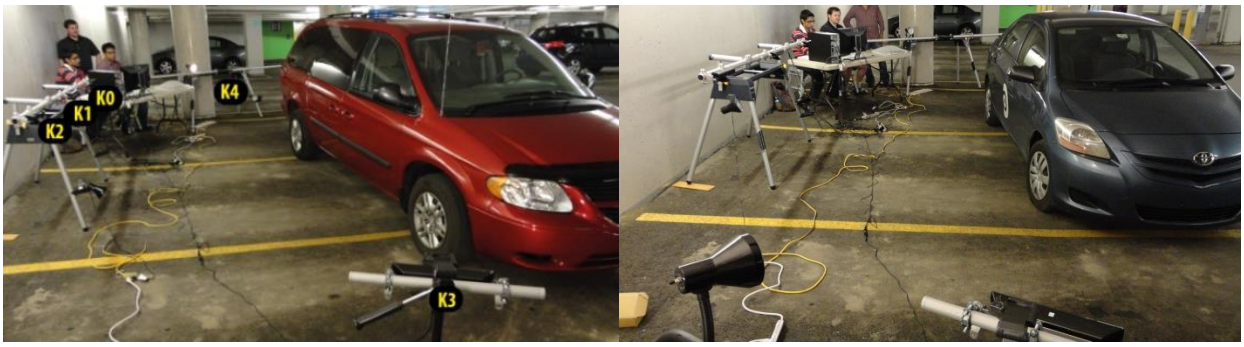


Figure 6.3 : Registration between Kinect K1 and all others Kinects in the network.

6.3 Data Collection and Results

After calibration, the data collection with the system is performed in a sequence. The overlapping regions between two contiguous Kinect sensors might contain interference, since all Kinect devices project a pattern of infrared points at the same wavelength to create their respective depth map. This produces small holes over the depth maps of overlapping sensors, as demonstrated in section 4.6. To prevent this problem, the data is collected sequentially over different time slots. During the first time slot, sensors *K0* and *K2* simultaneously collect their respective information. Then, sensors *K1*, *K3* and *K4* scan the corresponding regions over the vehicle. The delay between the shots is the time needed to shut down the devices and initialize

the next devices. This process is performed by the Kinect driver from the OpenNI framework [47]. It takes between 1 and 2 seconds to initialize each device and less than a second to collect data and reconstruct the 3D model. Figure 6.4 shows two different types of vehicle standing in front of the network of sensors for rapid 3D modeling that will eventually drive a robotic inspection via the manipulator arm that is integrated and calibrated with the peripheral vision stage, as detailed in section 5.4.



(a) **(b)**

Figure 6.4 : 3D textured data collection over different categories of vehicles for performance validation of the calibrated network of RGB-D sensors.

Reconstruction of the vehicle in Figure 6.4(a) is shown in Figure 6.5. The complete data collection and reconstruction takes around 15 sec for the entire network of sensors. The top and the lateral views of the vehicle are shown to analyze error in the calibration and registration processes. Figure 6.5(a) shows the reconstruction using the proposed method with only minor mismatches in the point clouds. The error in the profile of the vehicle can be noticed, especially on the back side of the vehicle where the calibration of Kinect *K4* propagates from *K0* to *K1*, which leads to cumulative error components on the calibration parameter estimates. Moreover the mismatch on the front narrow beam of the roof can be noticed. These minor errors are then refined with the ICP algorithm in MeshLab [51], starting with the calibration parameters as initial estimates, and the results are shown in Figure 6.5(b). Some improvement in the profile of the vehicle and on the roof of the vehicle can be noticed.

(a) (b)

Figure 6.5 : Reconstruction of the vehicle using the proposed method a) without using ICP b) with ICP refinement.

Figure 6.6 shows six different views of the reconstructed minivan refined with the ICP algorithm. All of the main areas of the vehicle body which are made of dark shiny red surfaces and wheels, including dark rubber tires, are accurately reconstructed and sections of the model acquired from the five viewpoints are correctly aligned. In these reconstructions, the windshield, lateral windows, and part of headlamps and rear lamps are missing in the depth map because the IR energy generated by the Kinect devices passes through these transparent surfaces or is deflected in other directions. However, the rear window of the minivan, which is made of tinted glass, is partially captured. The wind deflector, which is the dark shiny plate in front of the vehicle, is partially captured.

Figure 6.6 : Six different views of the reconstructed minivan.

Figure 6.7 shows six different views of the reconstructed car. Unlike the minivan, this vehicle does not contain tinted glass and therefore windows completely disappeared from the reconstruction. This vehicle is 0.8m smaller in length than the minivan but has similar shiny surfaces with a different color. All the surface areas of the vehicle and the tires are properly reconstructed. Moreover the circular logo on the side of the vehicle is also reconstructed fairly with minor errors. Similar to the case of the minivan, the windshield, lateral windows, and part of headlamps and rear lamps are missing in the depth map.

Figure 6.7 : Six different views of the reconstructed car.

Table 6.1 presents a comparison between the characteristics of the reconstructed vehicles and their actual dimensions (obtained from manufacturer's websites). The Kinect depth quantization introduces scaling errors of about 1cm in the height and the width, and a depth error of about 2.5 cm at 3m distance. Each sensor is installed to cover the full height of the vehicle and the average error on height achieved with these tests is under 1% of the total vehicle height. The estimation of the length of the vehicles and the length of the wheels separation (i.e. the distance between the centers of the front and back wheels) involves all the calibration parameters. These metrics are then representative of the overall accuracy that can be achieved in the reconstruction with the network of RGB-D sensors. The error on the wheels separation which involves Kinects *K0*, *K1* and *K2*, and the error on the total length of the vehicles, which involves all five sensors, is under 2.35% of the measured length, which is relatively minor given the medium quality of data provided by Kinect sensors at a depth of 2m,

as proposed for the vehicle location in section 4.5, and in proportion to the large working volume that is achieved.

Table 6.1 : Dimensions estimated from reconstructions compared with ground truth values.

		Height	Length	Wheels separation
Car	**Actual (mm)**	1460	4300	2550
	Model (mm)	1471	4391	2603
	Error (%)	0.75	2.11	2.07
Van	**Actual (mm)**	1748	5093	3030
	Model (mm)	1764	5206	3101
	Error (%)	0.91	2.21	2.34

These results can be compared to the implementation of the telepresence system by Maimone and Fuchs [21]. They used five Kinect sensors within an office cubicle of size 1.9x2.4m to capture the human body. The Kinect sensors were setup to view the rear side of the cubicle which is 1.9m wide. Within that small area they achieved a maximum error of 3.63cm, which is around 1.91%. The error in the proposed system is around 2.35% while covering the larger length of 5.2m.

Tong e*t al.* [20] used three Kinect sensors to scan the full human body, while the body is rotating on the turntable between the system. The two Kinect sensors are positioned to capture the top and bottom part of the body from one direction and the third sensor is placed at 2m in the opposite direction to capture the middle part. After merging the data, the acquired model dimensions were compared with the original dimension. The maximum error they achieved is 3cm while covering the height of the human body which is approximately 1.8m. The maximum error in their system is around 1.67% while covering the small work area and the distance between object and the sensors is less than 1m.

The error observed on the visual guidance system that is developed is tolerable given that the multi-camera acquisition platform is to be incorporated into the robotic system that will

embed proximity and touch sensing devices on the end effector of the robot. The 3D models achieved very rapidly therefore provide enough information to move the robot in proximity of the vehicle and then have to robot rely on the other sensors mounted on the end effector to finely control the task performed on the vehicle.

Figure 6.8 shows the reconstruction of other models of vehicles along with that of some garbage bins, acquired with the exact same setup, to evaluate the generalization capabilities of the proposed calibrated RGB-D acquisition framework. A wide range of vehicles was covered during experiments, in terms of colors and size. The white color vehicle appears more integrally than the vehicles with dark gray color, where missing depth data are noticed over the front part on the right of the vehicles where the density of points in the acquisition varies to a greater extent given the significant change of alignment between Kinects *K2* and *K3*. The dark green garbage bins are also correctly reconstructed with proper alignment between the piecewise RGB-D models. Unlike the vehicles, the garbage bins do not present any transparent surfaces, and as a result they become entirely visible in the reconstruction, except for the cover parts where the relative normal orientation of the top panels is too far away from the Kinect sensors principal axes.

Figure 6.8 : Reconstruction of various vehicles and garbage bins.

6.4 System Improvement

This system is implemented in the underground parking garage to analyze the maximum efficiency of the sensor without the presence of any direct or indirect day light. In section 3.5.2 the effect of indirect day light is also presented where the 3D data around the vehicle is gather in a semi outdoor parking during day time, which brings indirect natural lighting on the scene. The vehicle is also covered with dirt which makes apparent shades of paint on the vehicle. In such conditions, the Kinect depth sensor still provides good depth data over the complete vehicle. Therefore, it is allowed to expect that the proposed system can be extended to operate in semi-outdoor environments.

The current system only provides coverage over one half of the vehicle and can therefore easily be extended to a complete 360 degrees view by replicating the same structure on the other size of the vehicle, if working space permits. Acquiring a complete peripheral view can bring some improvements in the overall registration of the system. As five Kinect sensors provide half of the views, if the other half was available then a complete loop would be available for registration and reconstruction. Using such a closed loop, the error on calibration parameters at each stage can be further minimized when applying the ICP algorithm.

The inherent limitations of the Kinect sensor are clear in the reconstructions achieved, especially on complex surfaces with different reflectance characteristics as found on automotive vehicles, which lead to the incapability to capture transparent surfaces. However, these areas can be artificially filled, for safety reasons among others, during mesh generation by locating the contour of the windows with the assistance of the visual detector of vehicle parts (VDVP) algorithm introduced by Chávez-Aragón *et al.* [48], and described in section 4.4.1. Similarly, some of the missing depth values around the front and head lamps can be approximated by interpolation [52]. Finally, some parts of the vehicle, like the dark shiny wind deflector in Figure 6.6, could also be reconstructed more extensively if the second half of the setup was included and provide significant overlap between sensors to cover those regions.

Several techniques are being developed nowadays to improve the performance of the Kinect sensors and for expanding its use to different scenarios. In chapter 3, it has been discussed that the Kinect sensor offers an advantageous choice of technology for application scenarios that require fast acquisition of data with relatively good quality. This work contributes towards the development of such applications for the Kinect sensor in robotic systems.

6.5 Summary

The chapter presented an experimental validation of the proposed configuration of the multi-camera vision system in a practical application for automotive vehicle. To measure the full efficiency of the system the acquisition stage was installed in an underground parking garage to avoid any interference from any direct or indirect sunlight. The setup covered a large working volume using five Kinect sensors. A method for extrinsic calibration in between separate Kinect units is applied using the proposed method which takes advantage of the 3D measurement technology embedded in the sensors and provides registration accuracy within the range of the depth measurements accuracy provided by the Kinect technology.

3D data is collected over two different kinds of vehicles, such as a minivan and a car. In both cases the reconstructed model covered the entire 180 degrees of a vehicle, which includes the main areas of the vehicle body and wheels. Some areas are partially reconstructed such as headlamps and rear lamps. Most transparent surfaces are completely missing because the IR energy passes through these surfaces expect for tinted glass on the minivan that is partially captured.

The performance of the system is measured by comparing the reconstructed vehicle with its original dimensions. For such a large system the deviation of the measurement is less than 2.35% with respect to the original dimensions of a vehicle. This error is due to the medium quality of depth data and the residual error of the calibration process. However, this level of error is tolerable because the robotic system that will be incorporated into the system embeds proximity and the touch sensing devices on the end effector.

Finally, some further improvements on the system are proposed to overcome the limitations of the Kinect sensor to fill the missing areas in the reconstructed model of a vehicle. These improvements include the incorporation of more sensors to cover the complete vehicle and fill the missing parts using interpolation on problematic areas detected the Visual detector of vehicle parts (VDVP) algorithm introduced by Chávez-Aragón *et al.* [48].

Chapter 7. Conclusion and Future Work

This thesis focused on the design and calibration of a network of RGB-D sensors distributed to provide fast color and 3D imaging over a large workspace. The work included the design of a custom multi-camera system and its calibration. In addition, the calibration of a robotic manipulator with the proposed multi-camera system was also addressed. This final chapter summarizes the research work and highlights the major contributions. The last section discusses possible future extensions.

7.1 Summary

This thesis began with a review of 3D imaging technologies and their working principles. The pros and cons of each technology were also presented. The review was then extended toward the use of 3D imaging technologies in the design of systems that include more than one sensor. A few examples of multi-camera systems were presented. Finally the review ended with a discussion on the calibration of the sensors and the multi-sensor system integration with registration methods.

Chapter 3 proceeded with the selection of a 3D imaging technology suitable for the scope of this reach work among the reviewed technologies. The Microsoft Kinect sensor was selected as a 3D imaging device for this work because of its speed and as a decent compromise between rapidity and quality. The remaining of the chapter analyzed the characteristics of the Kinect sensor, its performance, usable operating range and response under some lighting conditions.

All characteristics of the Kinect sensor were taken into consideration for the multi-camera system design in chapter 4. The proposed acquisition system was designed to collect textured 3D data from the surface of an automotive vehicle of size up to that of a minivan. The design also properly supports a comprehensive calibration procedure between several Kinect sensors. The interference issue between multiple Kinect sensors was also investigated.

A customized calibration procedure for the system defined in chapter 4 was detailed in chapter 5. The first phase of the calibration addresses the internal calibration of all Kinect

sensors individually. This calibration relates the color and depth data together. The next phase of the calibration supports the external extrinsic calibration between all Kinect sensors in the network. This calibration was achieved by taking advantage of the 3D measurement technology of the Kinect sensor to provide registration accuracy within the range of the depth sensor accuracy. Finally a robot manipulator base reference frame is calibrated with the network of Kinect sensors using a checkerboard target mounted on the end effector of the robot.

In order to validate the proposed acquisition framework, the implementation of the proposed multi-camera system in a real-world application scenario was presented in Chapter 6. In this experiment, the acquisition system was installed in an underground parking garage and data was collected over vehicles of different sizes. An examination of the accuracy of the reconstruction achieved, in the context of the task considered for this research work, was conducted and demonstrated adequate performance both in time and resolution of the size and shape of the models.

7.2 Contributions

This thesis proposes a multi-camera system design and calibration, integrated with a robotic system, for the latter to perform inspection tasks on a vehicle in an autonomous manner under visual guidance. The main contributions of this work are:

- An experimental study of the Microsoft Kinect sensor technology, under its first two generations. The analysis confirms and extends knowledge about expected performance metrics in different scenarios, based on the quality of depth measurements, actual field of view of the sensor, and its color response.
- The design and implementation of a reconfigurable multi-camera RGB-D vision system, which is capable to cover a large imaging volume while remaining easy to calibrate.
- The development of a methodology for internal intrinsic and extrinsic calibration of Kinect sensors, which furtherreduces reprojection errors over default manufacturer`s calibration.

- The development of a procedure to accurately and efficiently merge depth and color data, acquired from Kinect sensors, based on their internal calibration.
- The design and implementation of an extrinsic calibration method in between pairs of RGB-D sensors configured in a network, while ensuring fast and easy execution of the procedure on the field with best possible accuracy.
- The design and implementation of a method for calibration between a robotic arm base reference frame and Kinect based multi-camera vision system.
- An experimental testing and validation of the multi-camera vision system in a real-world application.

The main contribution of this thesis is the method developed to calibrate a network of Kinect RGB-D sensors. There are not yet many techniques available to calibrate a network of Kinect sensors for industrial applications over an extended workspace. The techniques currently available in the literature are limited to smaller workspaces, like the telepresence system by Maimone and Fuchs [21] or full body scanner by Tong e*t al.* [20]. The method developed here can be used for imaging large objects and it is validated in this thesis for imaging automotive vehicles. Parts of the work presented in this thesis have been published in [53] [54].

7.3 Future Work

This thesis presented the development of a vision system to acquire textured 3D data over a vehicle to guide robotic work. The current system is limited to cover one half of the vehicle and could advantageously be extended to complete 360 degrees coverage by replicating the same structure to the other side of the vehicle. This acquisition platform can also be fully integrated with the work of Chávez-Aragón *et al.* [48] to efficiently and automatically detect and localize a number of characteristic areas over a vehicle. Such integration between the frameworks would provide direct means to mark regions of interest over the 3D reconstruction surface where the robot needs to perform its operations.

Further improvement in the data provided by the Kinect sensors could be achieved by post-processing the data to recover the missing depth area using filters or interpolation. The

color values may vary between all color cameras of Kinect units due to different exposure to the light. The color matching process can give a uniform color all over the point cloud. Fast mesh generation methods applied on the final reconstructed vehicle can also be included as part of future work to improve the visual appearance of the models, that would be needed as an interface with human operators.

References

[1] R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision, UK: Cambridge University Press, 2000.

[2] O. Faugeras, Three-Dimensional Computer Vision: A Geometric Viewpoint, Massachusetts: MIT Press, Cambridge, 1993.

[3] J. Oliensis, "A Critique of Structure-from-Motion Algorithms," *Computer Vision and Image Understanding,* vol. 80, pp. 172-214, 2000.

[4] R. Schwarte, Z. Xu, H.-G. Heinol, J. Olk, R. Klein, B. Buxbaum, H. Fischer and J. Schulte, "New Electro-Optical Mixing and Correlating Sensor: Facilities and Applications of the Photonic Mixer Device (PMD)," *in Proceedings of the SPIE 3100,* 1997, doi:10.1117/12.287751.

[5] T. Oggier, B. Büttgen, F. Lustenberger, G. Becker, B. Rüegg and A. Hodac, "Swissranger SR3000 and First Experiences Based on Miniaturized 3D ToF Cameras," *in Proceedings of the First Range Imaging Research Day at ETH Zurich,* pp. 97-108, 2005.

[6] G. J. Iddan and G. Yahav, "3D Imaging in the Studio and Elsewhere," *in Proceedings of SPIE,* vol. 4298, pp. 48-56, 2001.

[7] G. Yahav, G. Iddan and D. Mandelboum, "3D Imaging Camera for Gaming Application," *in Proceedings of the Intl Conference on Consumer Electronics,* pp. 1-2, 2007.

[8] S. Schuon, C. Theobalt, J. Davis and S. Thrun, "High-Quality Scanning Using Time-Of-Flight Depth Superresolution," *in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops,* pp. 1-7, 2008.

[9] P. J. Besl, "Active, Optical Range Imaging Sensors," *Machine Vision and Applications, Springer-Verlag,* vol. 1, no. 2, pp. 127-152, 1988.

[10] "VIVID 910 3D Laser Scanner," [Online]. Available: http://sensing.konicaminolta.asia/products/vivid-910-3d-laser-scanner/.

[11] "ShapeGrabber," [Online]. Available: http://shapegrabber.com/.

[12] R. J. Valkenburg and A. M. McIvor, "Accurate 3D Measurement using a Structured Light System," *Image and Vision Computing,* vol. 16, pp. 99-110, 1996.

[13] J. L. Posdamer and M. D. Altschuler, "Surface Measurement by Space-Encoded Projected Beam Systems," *Computer Graphics and Image Processing,* vol. 18, no. 1, pp. 1-17, 1982.

[14] K. L. Boyer and A. C. Kak, "Color-Encoded Structured Light for Rapid Active Ranging," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 9, no. 1, pp. 14-28, 1987.

[15] N. G. Durdle, J. Thayyoor and V. J. Raso, "An Improved Structured Light Technique for Surface Reconstruction of the Human Trunk," *in Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering,* vol. 2, pp. 874-877, 1998.

[16] P. Payeur and D. Desjardins, "Structured Light Stereoscopic Imaging with Dynamic Pseudo-Random Patterns," *in Proceedings of the Intl Conference on Image Analysis and Recognition,* vol. 5627, pp. 687-696, 2009.

[17] B. Freedman, A. Shpunt, M. Machline and Y. Arieli, "Depth Maping Using Projected Patterns". United States Patent US 2010/0118123 A1, 13 May 2010.

[18] G. Cheung, T. Kanade, J. Bouguet and M. Holler, "A Real Time System for Robust 3D Voxel Reconstruction of Human Motions," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* vol. 2, pp. 714-720, June 2000.

[19] P. Doubek, T. Svoboda and L. V. Gool, "Monkeys – a Software Architecture for ViRoom – Low-Cost Multicamera System," *in Proceedings of the 3rd Intl Conference on Computer Vision Systems,* pp. 386-395, April 2003.

[20] J. Tong, J. Zhou, L. Liu, Z. Pan and H. Yan, "Scanning 3D Full Human Bodies Using Kinects," *IEEE Transactions on Visualization and Computer Graphics,* vol. 18, no. 4, pp. 643-650, April 2012.

[21] A. Maimone and H. Fuchs, "Encumbrance-free Telepresence System with Real-time 3D Capture and Display using Commodity Depth Cameras," *IEEE International Symposium on Mixed and Augmented Reality,* pp. 137-146, 2011.

[22] P. Merrell, A. Akbarzadeh, L. Wang, P. Mordohai, J.-M. Frahm, R. Yang, D. Nister and M. Pollefeys, "Real-Time Visibility-Based Fusion of Depth Maps," *in Proceedings of the 11th IEEE International Conference on Computer Vision,* pp. 1-8, 2007.

[23] R. Y. Tsai, "A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-shelf TV Cameras and Lenses," *IEEE Journal of Robotics and Automation,* vol. 3, no. 4, pp. 323-344, August 1987.

[24] Z. Zhang, "A Flexible New Technique for Camera Calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 22, no. 11, pp. 1330-1334, November 2000.

[25] J. Heikkila and O. Silven, "A Four-Step Camera Calibration Procedure with Implicit Image Correction," *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pp. 1106-1112, 1997.

[26] J. Bouguet, "Camera Calibration Toolbox for Matlab," [Online]. Available: http://vision.caltech.edu/bouguetj/calib_doc.

[27] L. Lucchese and S. K. Mitra, "Using Saddle Points for Subpixel Feature Detection in Camera Calibration Targets," *in Proceedings of the Asia-Pacific Conference on Circuits and Systems,* vol. 2, pp. 191-195, 2002.

[28] N. Burrus, "RGBDemo," [Online]. Available: http://labs.manctl.com/rgbdemo/.

[29] C. Zhang and Z. Zhang, "Calibration betweeen Depth and Color Sensors for Commodity Depth Cameras," *IEEE International Conference on Multimedia and Expo,* pp. 1-6,

2011.

[30] M. Gaffney, "Kinect/3D Scanner Calibration Pattern," [Online]. Available: http://www.thingiverse.com/thing:7793.

[31] K. Berger, K. Ruhl, Y. Schroeder, C. Bruemmer, A. Scholzk and M. Magnor, "Markerless Motion Capture using Multiple Color-Depth Sensors," *in Proceedings of Vision, Modeling and Visualization,* p. 317–324, 2011.

[32] L. Wonwoo, "Kinect Color - Detph Camera Calibration," 2011. [Online]. Available: http://cv4mar.blogspot.ca/2011/03/kinect-color-detph-camera-calibration.html.

[33] C. Harris and M. Stephens, "A Combined Corner and Edge Detector," *in Proceedings of the 4th Alvey Vision Conference,* pp. 147-151, 1988.

[34] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision,* vol. 60, no. 2, pp. 91-110, 2004.

[35] M. Fischler and R. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography," *Communications of the ACM,* vol. 24, pp. 381-395, 1981.

[36] P. Rander, "A Multi-Camera Method for 3D Digitization of Dynamic, Real-World Events," *PhD Thesis, Robotics Institute, Carnegie Mellon University,* May 1998.

[37] S. Drouin, R. Poulin, P. Hébert and M. Parizeau, "Monitoring Flexible Calibration of a Wide Area System of Synchronized Cameras," *in Proceedings of the 16th Intl Conference on Vision Interface,* pp. 49-56, June 2003.

[38] T. Svoboda, D. Martinec and T. Pajdla, "A Convenient Multi-Camera Self-Calibration for Virtual Environments," *Presence: Teleoperators and Virtual Environments,* vol. 14, no. 4, pp. 407-722, 2005.

[39] B. Triggs, P. McLauchlan, R. Hartley and A. Fitzgibbon, "Bundle Adjustment: A Modern

Synthesis," *Vision Algorithms: Theory and Practice, LNCS,* vol. 1883, pp. 298-372, 2000.

[40] S. Bériault, "Multi-Camera System Design, Calibration and 3D Reconstruction for Markerless Motion Capture," *Master Thesis, University of Ottawa,* 2008.

[41] P. J. Besl and N. D. McKay, "A Method for Registration of 3-D Shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 14, pp. 239-256, Feb 1992.

[42] M. Rodrigues and Y. Liu, "Registering Two Overlapping Range Images Using A Relative Registration Error Histogram," *in Proceedings of the IEEE Intl Conference on Image Processing,* vol. 3, pp. 841-844, 2002.

[43] T. Masuda and N. Yokoya, "A Robust Method for Registration and Segmentation of Multiple Range Images," *in Proceedings of the IEEE Second CAD Based Vision Workshop,* pp. 106-113, 1994.

[44] R. Benjemaa and F. Schmitt, "Fast Global Registration of 3D Sampled Surfaces Using a Multi-Z-Buffer Technique," *in Proceedings of the IEEE International Conference on Recent Advances in 3-D Digital Imaging and Modeling,* pp. 113-120, 1997.

[45] J. Smisek, M. Jancosek and T. Pajdla, "3D with Kinect," *in Proceedings of the IEEE Intl Conference on Computer Vision Workshops,* pp. 1154-1160, 2011.

[46] K. Konolige and P. Mihelich, "Technical Description of Kinect Calibration," [Online]. Available: http://www.ros.org/wiki/kinect_calibration/technical.

[47] "OpenNI," [Online]. Available: http://openni.org/.

[48] A. Chávez-Aragón, R. Laganière and P. Payeur, "Vision-Based Detection and Labelling of Multiple Vehicle Parts," *in Proceedings of the IEEE International Conference on Intelligent Transportation Systems,* pp. 1273-1278, 2011.

[49] "OpenCV," [Online]. Available: http://opencv.willowgarage.com/wiki/.

[50] "Real-Time Computer Graphics and Physics, Mathematics, Geometry, Numerical Analysis,

and Image Analysis. Geometric Tools," [Online]. Available: http://www.geometrictools.com/LibMathematics/ Approximation/Approximation.html.

[51] "MeshLab," [Online]. Available: http://meshlab.sourceforge.net/.

[52] S. Matyunin, D. Vatolin, Y. Berdnikov and M. Smirnov, "Temporal Filtering for Depth Maps Generated by Kinect Depth Camera," *in Proceedings of the 3DTV Conference on The True Vision - Capture, Transmission and Display of 3D Video,* pp. 1-4, 2011.

[53] R. Macknojia, A. Chávez-Aragón, P. Payeur and R. Laganière, "Experimental Characterization of Two Generations of Kinect's Depth Sensors," *in Proceedings of the IEEE Intl Symposium on Robotic and Sensors Environments (ROSE 2012),* pp. 150-155, Magdeburg, Germany, 16-18 November 2012.

[54] R. Macknojia, A. Chávez-Aragón, P. Payeur and R. Laganière, "Calibration of a Network of Kinect Sensors for Robotic Inspection over a Large Workspace," *in Proceedings of the IEEE Workshop on Robot Vision (WoRV 2013),* Clearwater, FL, Jan. 2013.