

REALIZATION OF A SPATIAL AUGMENTED REALITY SYSTEM – A DIGITAL
WHITEBOARD USING A KINECT SENSOR AND A PC PROJECTOR

A Thesis

by

ANDREI A KOLOMENSKI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Approved by:

Chair of Committee,	John L. Junkins
Co-Chair of Committee,	James D. Turner
Committee Members,	Raktim Bhattacharya
	Hans A. Schuessler
Head of Department,	John E. Hurtado

May 2013

Major Subject: Aerospace Engineering

Copyright 2013 Andrei A Kolomenski

ABSTRACT

Recent rapid development of cost-effective, accurate digital imaging sensors, high-speed computational hardware, and tractable design software has given rise to the growing field of augmented reality in the computer vision realm. The system design of a 'Digital Whiteboard' system is presented with the intention of realizing a practical, cost-effective and publically available spatial augmented reality system.

A Microsoft Kinect sensor and a PC projector coupled with a desktop computer form a type of spatial augmented reality system that creates a projection based graphical user interface that can turn any wall or planar surface into a 'Digital Whiteboard'. The system supports two kinds of user inputs consisting of depth and infra-red information. An infra-red collimated light source, like that of a laser pointer pen, serves as a stylus for user input. The user can point and shine the infra-red stylus on the selected planar region and the reflection of the infra-red light source is registered by the system using the infra-red camera of the Kinect. Using the geometric transformation between the Kinect and the projector, obtained with system calibration, the projector displays contours corresponding to the movement of the stylus on the 'Digital Whiteboard' region, according to a smooth curve fitting algorithm. The described projector-based spatial augmented reality system provides new unique possibilities for user interaction with digital content.

ACKNOWLEDGEMENTS

I want to express my deepest appreciation to my advisor, Prof. John L. Junkins, for making this research possible. I am highly grateful to Prof. Junkins for giving me the opportunity to work at the Land Air and Space Robotics Lab at Texas A&M University and influencing me to pursue a specialization in computer vision as applied to aerospace engineering.

I am also greatly thankful to Prof. Majji for being my mentor during my first year of my graduate career. He supported a friendly research environment and provided me with the fundamental knowledge needed for conducting my research. I would also like to thank my committee co-chair Prof. Turner and committee members Prof. Bhattacharya, and Prof. Schuessler, for their guidance and continuing support throughout the course of this research.

Thanks also go to my friends and colleagues that I collaborated with at the Land Air and Space Robotics Lab throughout my graduate career at Texas A&M University. Also thanks go out to department faculty and staff for making my time at Texas A&M University an exciting and valuable experience.

Finally, I am grateful for the everlasting support and love from my dear mother and father. Their encouragement inspired me to pursue a degree of higher education in the field of aerospace engineering and their motivation allowed me to persevere through hard times.

TABLE OF CONTENTS

	Page
ABSTRACT	ii
ACKNOWLEDGEMENTS	iii
TABLE OF CONTENTS	iv
LIST OF FIGURES	vi
LIST OF EQUATIONS	viii
LIST OF TABLES	x
CHAPTER I INTRODUCTION	1
1.1 Augmented Reality	2
1.1.1 Overview of Current Augmented Reality Systems	3
1.2 Motivation and Goal	4
1.3 Thesis Structure	6
CHAPTER II SYSTEM DESIGN	7
2.1 Hardware Components	7
2.1.1 RGB Camera	8
2.1.2 IR Camera	9
2.1.3 IR Stylus	13
2.1.4 Digital Projector	14
2.2 Software Components	15
CHAPTER III SYSTEM CALIBRATION	16
3.1 Methodology	16
3.1.1 Zhang's Method	17
3.2 Camera Calibration	19
3.2.1 RGB Camera Calibration	19
3.2.2 IR Camera Calibration	23
3.2.3 Extrinsic RGB and IR Camera Calibration	28
3.3 Projector Calibration	29
3.3.1 Identification of Calibration Plane	30

3.3.2 Localization of 3D Points.....	33
3.3.3 Projector Calibration Results	36
CHAPTER IV USER INTERACTION WITH SAR SYSTEM	40
4.1 Projective Transformations	42
4.1.1 IR Camera to RGB Camera Transformation.....	43
4.1.2 RGB Camera to Projector Transformation.....	46
4.2 IR Stylus and Depth Detection.....	48
4.3 IR Stylus Input Visualization	49
4.3.1 Parametric Curve Fitting	50
CHAPTER V CONCLUSIONS	52
5.1 System Implementation.....	52
5.2 System Improvements	53
NOMENCLATURE.....	54
REFERENCES.....	56
APPENDIX COMPUTER VISION BACKGROUND.....	59
A.1 Projective Space	60
A.1.1 Projective Transformations	61
A.2 Pinhole Camera Model.....	61
A.2.1 Intrinsic Parameters.....	65
A.2.2 Extrinsic Parameters.....	70

LIST OF FIGURES

FIGURE		Page
1	Microsoft's Kinect sensor	2
2	Visual representation of the proposed SAR system	5
3	The proposed SAR system includes the Kinect sensor and a projector	8
4	'Skeleton' of Microsoft's Kinect sensor, showing its main functional units: structured light projector, RGB camera and IR camera	8
5	Experimental setup used to measure IR camera spectral sensitivity.....	10
6	Cropped IR image for a monochromator set wavelength of 820 nm.	11
7	Plot of spectral sensitivity versus wavelength of light for the IR camera.....	12
8	830 nm IR LED	13
9	IR stylus.....	14
10	Pixel re-projection error of all RGB calibration images	20
11	Pixel re-projection error of filtered RGB calibration images.....	21
12	Extrinsic parameters of calibration target for RGB camera calibration.....	23
13	Chessboard corner extraction applied to a low contrast IR image	24
14	Pixel re-projection error of all IR calibration images.....	25
15	Pixel re-projection error of filtered IR calibration images	26
16	RGB camera acquisition of real and projected calibration targets.....	30
17	Possible projector configurations that affect 'keystone' distortions	37
18	Kinect IR projector pattern.....	41
19	SAR system with visualized transformations.....	43
20	User interaction with developed SAR system.....	51

21	Pinhole camera model	62
22	Geometry for computing the x-coordinate on the image plane.....	63
23	Geometry for computing the y-coordinate on the image plane.....	64
24	Skewed CCD camera pixels	66

LIST OF EQUATIONS

EQUATION	Page
1	Difference norm of an IR Image with respect to a background IR image 11
2	Extrinsic relationship between RGB and IR cameras 28
3	Extrinsic parameter results between RGB and IR cameras..... 29
4	Rotation matrix and translation vector of RGB camera with respect to calibration plane 31
5	Definition of calibration plane normal vector 32
6	Representation of point \mathbf{q} that is located on the calibration plane 32
7	Vector equation of calibration plane that is satisfied by point \mathbf{q} 32
8	Scalar equation of calibration plane 33
9	RGB camera projection of 2D image point to corresponding 3D metric point..... 34
10	Intersection of 3D rays and calibration plane..... 34
11	Expression for scale parameter, λ , in RGB camera frame 35
12	Application of scale parameter, λ , to define 3D metric points in the RGB Camera reference frame 35
13	Extrinsic parameter result between RGB camera and projector 39
14	IR camera projection of 2D image point to corresponding 3D metric point... 44
15	Expression for scale parameter, λ , in IR camera frame..... 44
16	Application of scale parameter, λ , to define 3D metric point in IR camera reference frame..... 45
17	Projection of 3D metric point expressed in IR camera reference frame to the corresponding 2D image point in the image plane of the RGB camera.... 45

18	RGB camera projection of 2D image point to corresponding 3D metric point.....	46
19	Expression for scale parameter, λ , in RGB camera frame	47
20	Application of scale parameter, λ , to define 3D metric point in RGB camera reference frame	47
21	Projection of 3D metric point expressed in RGB reference frame to the corresponding 2D image point in the image plane of the Projector	48
22	3D relationship between Euclidian and projective space.....	60
23	Generalized relationship between Euclidian and projective space	60
24	Projection of a world 3D point to a 2D image point	64
25	Ideal pinhole camera projection	65
26	Pinhole camera projection.....	65
27	Skew factor relation	66
28	Aspect ratio relation	67
29	Perspective transformation using five intrinsic parameters	67
30	Intrinsic camera matrix.....	67
31	Radial and tangential distortion parameters	69
32	Projection mapping using both intrinsic and extrinsic camera parameters	70

LIST OF TABLES

TABLE		Page
1	RGB camera intrinsic parameters.....	22
2	IR camera intrinsic parameters.....	27
3	Projector camera intrinsic parameters	38

CHAPTER I

INTRODUCTION

Computer vision is a branch of computer science that deals with acquiring and analyzing digital visual input of the real world for the purpose of producing numerical or symbolic information. Since its inception, computer vision presented many challenges for the research community due to its unique hardware and software requirements and its computational intensity. In recent years this field has greatly evolved due to hardware and software advancements and wide availability of digital sensors and high-speed processors.

Initially, digital cameras produced two-dimensional digital images that captured the color or light intensity of a scene, however recent developments allowed digital sensors to obtain depth or three dimensional information of the captured scene. Before the invention of depth sensors, depth information could be obtained from two digital cameras by forming a stereo camera pair and applying triangulation. However, Microsoft's XBOX 360 Kinect sensor released in November of 2010 revolutionized the field of computer vision by offering a low cost and effective system that offers color and depth acquisition capabilities at a live frame rate of 30 Hz. The Kinect was hacked to operate with a PC through the USB port within a few hours of its release [1]. The Kinect uses a structured light approach to obtain depth that has many advantages over the traditional stereoscopic approach. The Kinect sensor is displayed in Figure 1. Over 8 million Kinect sensors were sold world-wide in the first sixty days since its release [2].

Kinect's commercial success is partly due to a growing scientific community that is interested in extending the sensing capabilities of computer vision systems.



Figure 1. Microsoft's Kinect sensor

1.1 AUGMENTED REALITY

An emerging field of computer vision that allows integration of digital content with the real world is called augmented reality. Augmented reality dates back to the early 1900's when the author L. Frank Baum introduces the idea of using electronic glasses that overlay digital data onto the real world [3]. The subsequent milestones in the development of this new field include:

- In 1962 a cinematographer Morton Heilig creates a motorcycle simulator called 'Sensorama' with 3D visual effects, sound, vibration, and smell [4].
- In 1966 Ivan Sutherland invents the head-mounted display during his research at Harvard University. With this head-mounted display simple wireframe drawings of digital content were overlaid over real world scenarios at real time frame rates [5].

- In 1990 Tom Caudell coins the term Augmented Reality; while at Boeing he developed software that could overlay the positions of cables in the building process helping workers assemble cables into aircraft [6].
- In 1997 Ronald T. Azuma publishes a survey paper which accurately defines the field of AR [7].
- In 2002 Steven Feiner publishes the first scientific paper describing an AR system prototype and its mechanics [8].

Substantial developments in this field arose in the early 1990's when new emerging hardware and software enabled the implementation of various augmented reality systems. Specifically, with the invention of 3D sensing systems, augmented reality has gained even greater attention as now it is easier to determine the important geometric relationship between a given scene and the acquisition sensors. A subcategory of augmented reality is spatial augmented reality (SAR) which uses a video-projector system to superimpose graphical information directly over a real world physical surface. This type of system is the focus of this research.

1.1.1 OVERVIEW OF APPLICATIONS OF SAR SYSTEMS

SAR systems find their applications in different fields due to several advantages that such systems offer. An image projection onto a screen makes it useable by several individuals simultaneously. Recent developments include applications for surgery, overlaying the image of internal organs on the outer surface, thus enabling visualization

of what is hidden under the skin [9]. SAR is also helpful in the training process providing necessary information and hints for the trainees [10]. Another application is in visualization of aerodynamics of objects: the aerodynamics flow lines can be directly imposed on the object, thus making the flow field apparent [11]. Also, SAR is found in industrial machinery operation by enhancing visibility of occluded tools and displaying the mechanical process itself in the 3D space [12]. Other applications exist in construction, architecture and product design. Recent suggestions include digital airbrushing of objects and employment of SAR for product design [13]. Such applications must take into account technical characteristics of the digital sensing systems and the projector as well as their integration with a control computer [14].

1.2 MOTIVATION AND GOAL

The goal of this research is to design an interactive SAR system that can turn any wall or planar surface into a 'Digital Whiteboard'. The proposed system consists of a Microsoft Kinect sensor coupled with a common PC external light projector, both connected to a desktop or laptop computer. The system is made mobile in the sense that it can be placed into different real world settings, and it is able to auto-calibrate itself to a given planar region that will serve as the calibration plane. The field of view of this calibration plane by the IR camera will determine the effective 'Digital Whiteboard' region that is initialized for IR stylus input.

User interaction is modeled by depth and infra-red (IR) information. Depth information is obtained by Kinect's structured light scanning system that utilizes

triangulation. User interaction through depth is performed by physically interacting with the system by moving or placing real world objects in front of the planar region, thereby displacing the depth of the region with respect to Kinect's depth sensor. IR information is acquired by the Kinect's IR sensor. An IR collimated light source, like that of laser pointer pen, serves as a stylus for user input on the planar region. The user will shine the IR stylus on to the 'Digital Whiteboard' and the scattered and reflected IR light will be registered by Kinect's IR camera. A tracking algorithm employed on the control computer will centroid and track the movement of the IR light source through-out the calibrated region. Using the geometric transformation between the Kinect sensor and the projector, obtained through system calibration, the projector will project the contour movement of the stylus onto the 'Digital Whiteboard' region. Figure 2 visualizes the proposed SAR system.

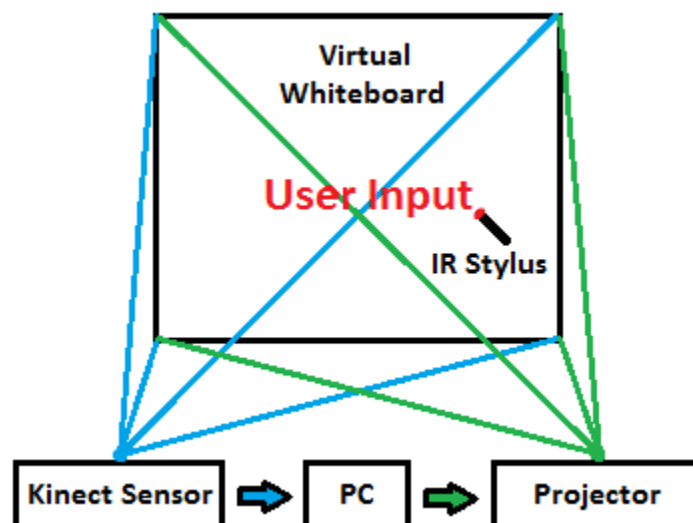


Figure 2. Visual representation of the proposed SAR system

The described SAR system will provide new unique possibilities for user interaction with digital content. Since the written information by the IR stylus is digitized it is possible to stream this information to other computers and devices for instantaneous exchange of written information. For example, this could be utilized in education to pass on lecture notes written by a teacher or professor, thereby replacing/complimenting the original chalk whiteboard.

1.3 THESIS STRUCTURE

The presented thesis is divided into five chapters: introduction, system design, system calibration, user interaction with SAR system, and conclusions. Each chapter is divided into subsections detailing the developments presented for the topic. Also, an appendix is added at the end of the thesis to provide required background in computer vision and image formation. A list of used nomenclature is placed after the appendix to define used acronyms and variables used in the mathematical development.

CHAPTER II

SYSTEM DESIGN

In order to realize the proposed SAR system various hardware components are required such as: IR camera, color camera, digital projector and a desktop or laptop computer. As the goal of the system is to digitize the input of an IR stylus so the projector can display its movement, the light reflection of the IR stylus must be observable by the IR camera. A usual color camera will not be able to detect the IR return. The advantage of using an IR stylus is that it will not interfere with the projected image displayed by the projector and IR input cannot be confused with projector output. Also such a system will be able to function under low lighting conditions providing more versatility with respect to its operational settings.

2.1 HARDWARE COMPONENTS

Microsoft's Kinect sensor provides a cheap and effective IR camera that is coupled with a color camera. Since the IR camera is part of a structured-light stereo pair with the IR projector, the Kinect can also obtain depth information. For the following SAR system development the Kinect sensor will provide both IR and color cameras. Also a common digital projector will be used for image projection. All of these digital devices will be connected to computer that supports the software controlling the SAR system. Figure 3 displays the hardware used to realize the proposed SAR system.



Figure 3. The proposed SAR system includes the Kinect sensor and a projector

2.1.1 RGB CAMERA

From this point on in the thesis, the color camera of the Kinect will be referred to as the RGB camera because it provides three color channels: red, green and blue for color visualization. By default the RGB camera supports an 8-bit VGA image resolution of 640 x 480 at a 30 Hz refresh rate using a Bayer color filter. Also it can support a 1280x1024 image resolution at a refresh rate of 15 Hz [15]. Figure 4 shows the 'skeleton' of the Kinect sensor with labeled hardware components.

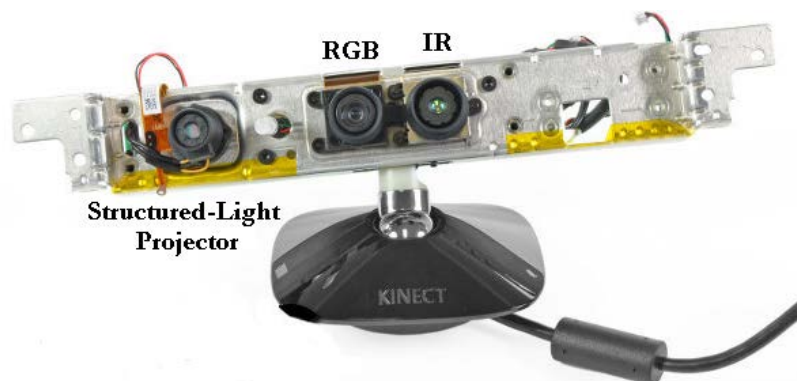


Figure 4. 'Skeleton' of Microsoft's Kinect sensor, showing its main functional units: structured light projector, RGB camera and IR camera

2.1.2 IR CAMERA

IR camera by default supports a 16 bit monochrome image resolution of 640 x 480 at a refresh rate of 30 Hz. Similar to the RGB camera, it can also support a 1280 x 1024 image resolution at a refresh rate of 15 Hz. The IR camera of the Kinect also serves as a depth sensor when coupled with Kinect's IR projector, as they form a stereo pair that uses a structured-light approach to obtain depth. The depth data stream consists of an 11 bit monochrome 640 x 480 image that can be converted to true metric depth [15]. However, the main focus of this thesis is on the IR camera characteristics.

An experimental analysis was conducted on the Kinect's IR camera to determine its operational wavelength spectrum and thus evaluate the spectral sensitivity of the camera. Determining the peak sensitivity wavelength of the IR camera allows to optimally select an appropriate external light source used in the IR stylus, that will be detected by the IR camera. This is important, since user interaction will be based on an IR stylus, so its light return must be clearly visible by the IR camera.

To measure the sensitivity spectrum of the IR camera an incandescent light lamp was positioned in front of a monochromator (H20 UV 100-999 nm) to isolate the individual wavelengths of the visible light source. Then the IR camera was setup to collect IR images of the light response from the monochromator. Extra care was taken to insure the Kinect IR sensor only collected the monochromator light response. The explained experimental setup is visualized in Figure 5.

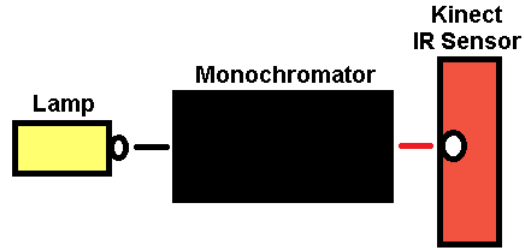


Figure 5. Experimental setup used to measure IR camera spectral sensitivity

An initial wavelength of 550 nm was set by the monochromator and 10 nm steps were taken to obtain IR camera images at each wavelength interval set by the monochromator, up to 890 nm. Also, a background IR image was taken with the lamp light source turned off to acquire the background light noise of the environment, for image subtraction. The experiment was conducted in a dark room with minimal unnecessary light sources to minimize the background light. In order to limit error and only measure the intensity of the monochromator response a constant region of interest is selected for each IR image centered on the light source. Figure 6 shows a sample IR image obtained for an 820 nm. wavelength.

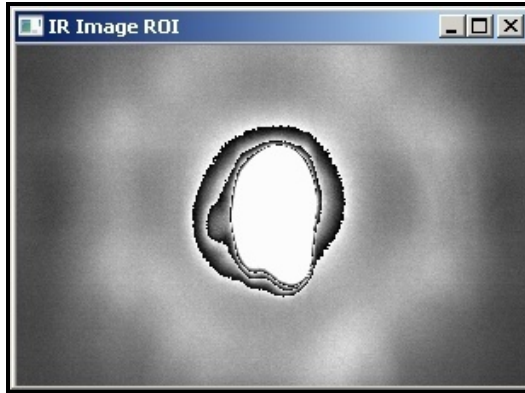


Figure 6. Cropped IR image for a monochromator set wavelength of 820 nm

A difference norm is computed between each IR image at a given wavelength and a common background IR image, to describe the change of the IR image pixel intensity with respect to the background noise. The difference norm is computed using Equation 1, where I_{img} is the input digital IR image associated with a given wavelength and I_{bg} is the digital IR image of the background light noise with the lamp turned off.

$$Norm(I_{img}) = \sum_{x,y} abs(I_{img_{x,y}} - I_{bg_{x,y}})$$

Equation 1. Difference norm of an IR Image with respect to a background IR image

Applying this formula to each IR image yields a normalized parameter that describes the change in pixel intensity for each wavelength with respect to the background image. This change in pixel intensity is proportional to sensitivity of the IR camera at a particular wavelength so it may be regarded as the spectral sensitivity of the

IR camera. The relationship between spectral sensitivity and wavelength of light for the IR camera is shown in Figure 7.

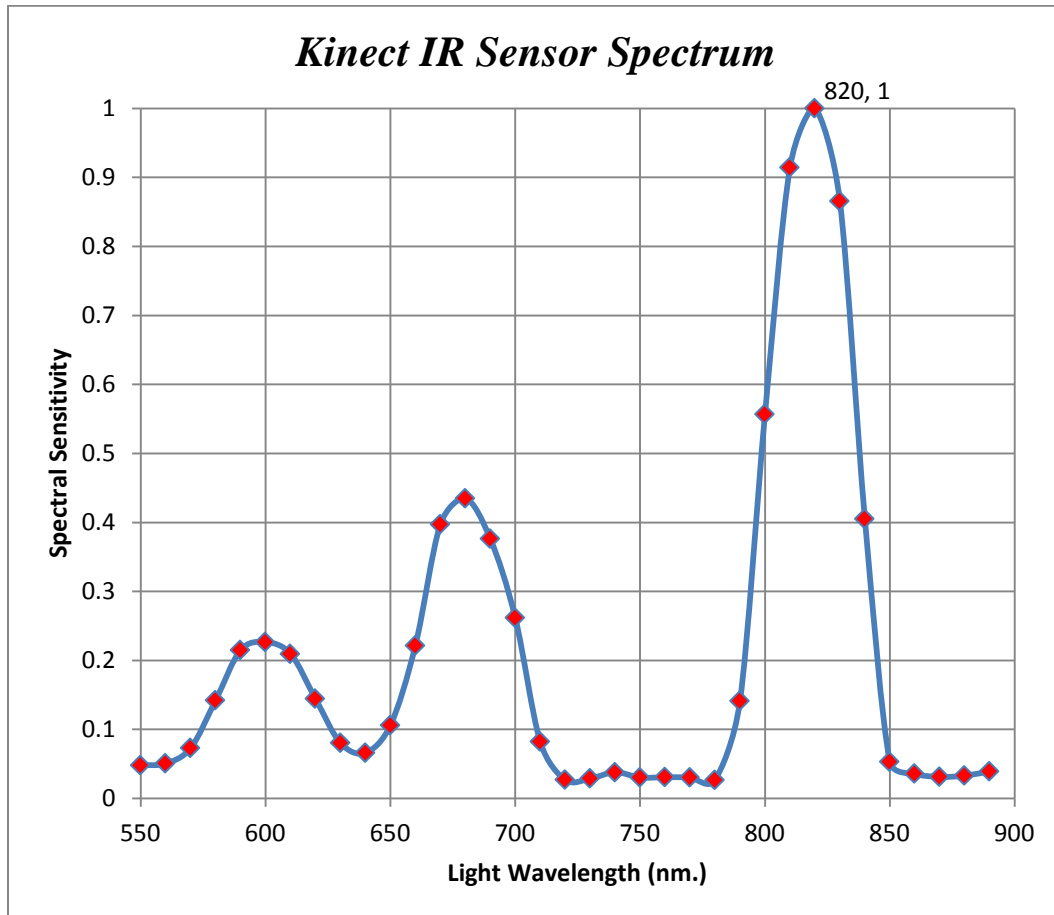


Figure 7. Plot of spectral sensitivity versus wavelength of light for the IR camera

It is interesting that there are three main peaks displayed in the spectral sensitivity, suggesting that the IR camera of the Kinect is equipped with an interference band-pass filter. Also, it may be noted that the peak IR camera sensitivity is obtained at

820 nm. wavelength, and the optimal operational wavelength range of the IR camera is approximately 800-840 nm.

The spectral trend in Figure 7 may be slightly skewed since the light bulb of the lamp emits mainly visible light. Visible light has a wavelength range of about 400 to 700 nm, so the lamp emits less light at 820 nm, the peak wavelength of the IR Sensor sensitivity. Due to this, the spectral sensitivity at 820 nm is probably greater than shown by the graphic, which further supports the conclusion that the Kinect's IR camera optimally performs in the 800-840 nm. wavelength range.

2.1.3 IR STYLUS

Knowing the optimal wavelength range for the IR camera an IR LED was obtained with a peak operational wavelength of 830 nm. This LED has a direct current of 100 mA and takes a 1.5 Volt power supply. It is capable of radiant intensity of 180 mW per steradian. The LED was obtained from Vishay Semiconductor. Figure 8 displays the utilized LED.

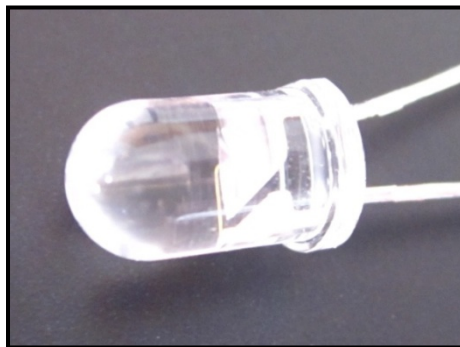


Figure 8. 830 nm IR LED

The pictured LED was placed inside a hollow pen with two 1.5 Volt AAA batteries so that when the stylus is clicked the LED turns on, and when the stylus is clicked again the LED turns off. Black cardboard material was placed in front of the LED with a very narrow opening to collimate the LED light source into a unidirectional light beam. This simple construction for the IR stylus is displayed in Figure 9.



Figure 9. IR stylus

2.1.4 DIGITAL PROJECTOR

The digital projector used for this SAR system is a simple Dell digital micro-projector (M109S) that connects to a PC using a VGA connection. It supports an 800 x 600 pixel resolution at a refresh rate of 60 Hz. It has a slider for focus adjustment. Also, it has an option to adjust for keystone distortion, which is caused by off centered projection. The intrinsic parameters of this projector will be determined during system calibration.

2.2 SOFTWARE COMPONENTS

The written software to control the SAR system is developed using the C++ programming language in a Visual Studio 2008 environment, on a Windows 7 64-bit operating system. Various open source third-party software libraries were used in the development of this software. The OpenNI library is used to facilitate communication with the Kinect sensor, as it provides the application program interface (API) to acquire data streams of the RGB, IR and depth sensors. The OpenNI library was developed in cooperation with PrimeSense, the company responsible for developing the Kinect sensor [16]. This is the primary reason for choosing this framework to work with the Kinect. The OpenCV library is also utilized to facilitate digital image processing operations employed by the written software. Also, a 3-D party add-on library to OpenCV called cvBlobsLib is used to facilitate detection of the IR stylus return in the IR images.

CHAPTER III

SYSTEM CALIBRATION

Vision system calibration is among the first and most important computer vision tasks to complete, as it allows extracting metric information from 2D images. Also, it provides a geometric relationship between the calibrated sensors for coherent sensor interaction. The objective of the implemented calibration procedure is to define all of the intrinsic and extrinsic parameters for the RGB camera, IR camera and projector. The intrinsic parameters describe the internal parameters of a camera such a focal length, principal point off-set, pixel skew and lens distortions. The extrinsic parameters describe the geometric relationship between the observed scene and the camera. For an introductory development of projective geometry, image formation and camera parameters refer to the appendix at the end of the document.

3.1 METHODOLOGY

The overall SAR system calibration involves the calibration of the RGB camera, the IR camera and the projector. The intrinsic parameters for the RGB camera and IR camera may be defined independently using the well-known and commonly used Zhang's calibration method. Once the intrinsic parameters are known for these two cameras, a stereo calibration procedure is utilized to determine the extrinsic parameters between the two cameras. Essentially, the stereo calibration involves both RGB and IR cameras observing a known geometrical pattern and solving for the extrinsic parameters

between the observed pattern and each camera. Since both cameras observe the same pattern and the rotation and translation is known for each camera with respect to the calibration target, it is possible to solve for the intermediate extrinsic parameters between the two cameras. Calibrating the intrinsic parameters of the projector and extrinsic parameters with respect to the two cameras is not as straightforward, because the projector does not acquire images, instead it projects images. Consequently, the projector may be thought of as an inverse of camera, since it takes 2D image points and projects them to 3D metric space. In this case, the same pinhole camera model may be applied to the projector as for the RGB and IR cameras. So as long as a proper point correspondences between 2D image points and 3D metric points are made for the projector, it can be calibrated by the same method as used for the RGB and IR cameras [17]. The following section explains the Zhang's calibration procedure that can be directly applied to RGB and IR cameras to obtain calibration parameters [18]. However, a modification of this method is used to calibrate the projector.

3.1.1 ZHANG'S METHOD

Single camera calibration is performed using Zhang's method. This technique requires that a given camera observes a geometrically known planar pattern at numerous different orientations (at least two). The planar pattern used for this calibration is a checkerboard pattern with known metric dimensions. This calibration pattern is attached to a rigid planar poster board and then placed in front of the camera. For every checkerboard orientation captured by the camera, correspondences are made between the

3D metric positions of the checkerboard corner points to the detected 2D image point positions of the corresponding corners in the image plane of the camera. Minimizing the pixel re-projection error in a least squares sense over the intrinsic and extrinsic parameters for every homography between the model plane and the image plane allows to solve for the camera's intrinsic and extrinsic parameters. Pixel re-projection error is defined as the geometric error measured by the Euclidian distance between a 2D projected image point obtained by the estimated calibration parameters and a true detected position of the same 2D point in the image plane.

The following outline summarizes the major steps of Zhang's calibration method:

1. Attach a checkerboard pattern onto a planar surface.
2. Capture a few (at least two) images of the calibration plane under different orientations. For this implementation, the calibration plane is moved while the camera is held stationary.
3. Detect the corner points of the checkerboard in each captured image by the camera.
4. Calculate the five intrinsic parameters and the six extrinsic parameters using a closed-form solution.
5. Estimate the coefficients of radial distortion by solving linear least squares.

3.2 CAMERA CALIBRATION

As previously mentioned, both the RGB and IR cameras are calibrated independently using Zhang's method to obtain their corresponding intrinsic parameters. Bouget's Matlab calibration toolbox is used to perform these independent intrinsic camera calibrations [19]. This calibration toolbox provides a convenient graphical user interface for camera calibration using Zhang's method.

For this calibration, the skew parameter is assumed to be zero since most modern cameras do not have centering imperfections due to a square pixel size. The distortion model developed by Brown is used to determine the radial and tangential distortion parameters [20]. Also, the higher order radial distortion parameters, k_4, k_5, k_6 , are not calculated since they usually have negligible effect on most camera lenses that do not have an extremely wide field of view. This is true for both RGB and IR cameras of the Kinect device.

3.2.1 RGB CAMERA CALIBRATION

RGB camera calibration was initially executed using twenty calibration images of the chessboard pattern. Some of these images introduced more error into the estimated calibration parameters than other images, causing a higher pixel re-projection error. This may be due to error in the computation of the 2D image corner positions or change in contrast or brightness within the image. Both of these effects add noise to the pixel positions of extracted corners, which degrades the quality of the calibration results. Figure 10 displays the pixel re-projection error for each image, where each colored cross

represents a single pixel re-projection error of a calibration corner for a given homography configuration.

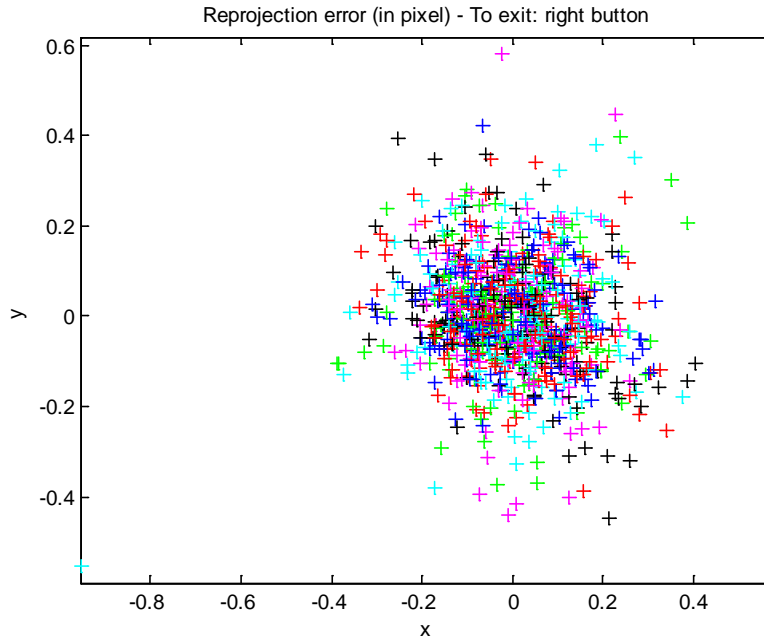


Figure 10. Pixel re-projection error of all RGB calibration images

Bouget's calibration toolbox has the useful option of suppressing bad images with high pixel re-projection error of calibration corners. This is done by excluding them from the camera calibration solution, thus filtering out the images that deviate from the optimized calibration parameters. This way, only good images are used for a calibration solution that insures a low pixel re-projection error. In this case, all calibration images that yielded a detected corner pixel re-projection error greater than 0.45 pixels were suppressed in the calibration solution. Figure 11 displays the pixel re-projection error for

the remaining images that were not excluded from calibration due to high re-projection error as compared to all other images in the set.

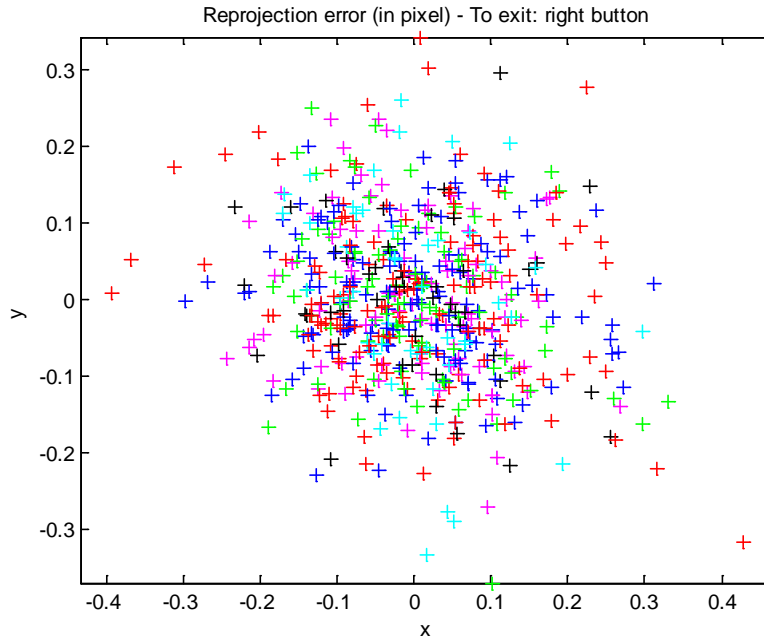


Figure 11. Pixel re-projection error of filtered RGB calibration images

Table 1 summarizes the intrinsic parameters obtained for the RGB camera using Bouget's calibration toolbox. A low root mean square pixel re-projection error of about a tenth of pixel was achieved for RGB camera calibration. This calibration result was achieved numerous times for the same RGB camera of the Kinect, to check for consistency.

Intrinsic Parameter	Parameter Value	Uncertainty
Focal Length	[515.33978 520.37584]	\pm [3.96014 3.87840]
Principal Point Offset	[312.73122 261.09161]	\pm [1.63181 1.87657]
Radial Distortion (k_1, k_2, k_3)	[0.17369 -0.38252 0.00000]	\pm [0.00966 0.02985 0.00000]
Tangential Distortion (p_1, p_2)	[-0.00230 0.00204]	\pm [0.00128 0.00132]
Pixel Re-projection Error	[0.11341 0.10237]	

Table 1. RGB camera intrinsic parameters

The computed uncertainties for each intrinsic parameter are within a tolerable range. Bouget's calibration toolbox has the option to visually re-project the calculated extrinsic parameters for each calibration pattern configuration. The visualization of the obtained extrinsic parameters is displayed in Figure 12. It can be confirmed that various calibration target poses were used for calibration, as this is essential to obtaining high fidelity calibration parameters [21].

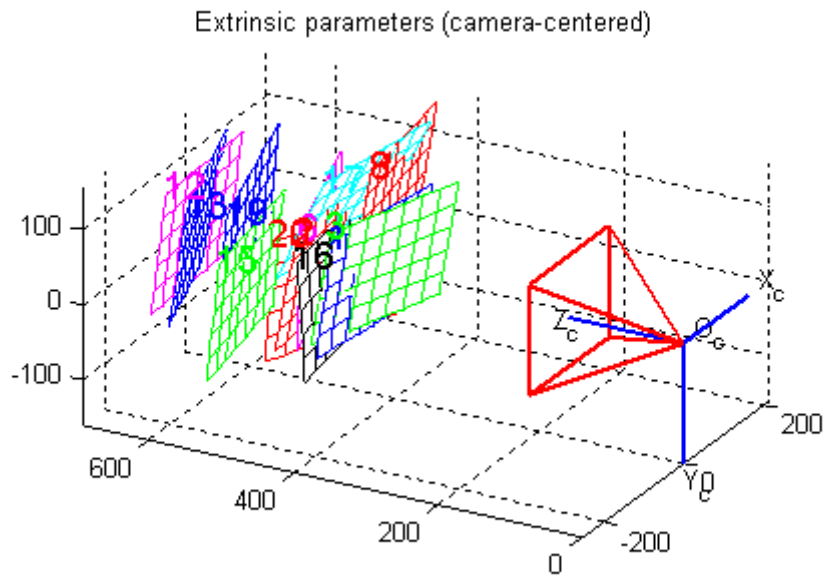


Figure 12. Extrinsic parameters of calibration target for RGB camera calibration

3.2.2 IR CAMERA CALIBRATION

The intrinsic camera parameters of the IR camera were also estimated using Bouget's calibration toolbox for a set of twenty IR images. Obtaining high accuracy intrinsic parameters for the IR camera is more difficult than the RGB camera since the IR image is monochrome and IR camera has little sensitivity to ambient light. Due this most IR images display poor contrast as evident in Figure 13. This decreases the accuracy of corner extraction in the IR image, resulting in higher pixel re-projection error. Nevertheless, corner detection can still be executed on the image.

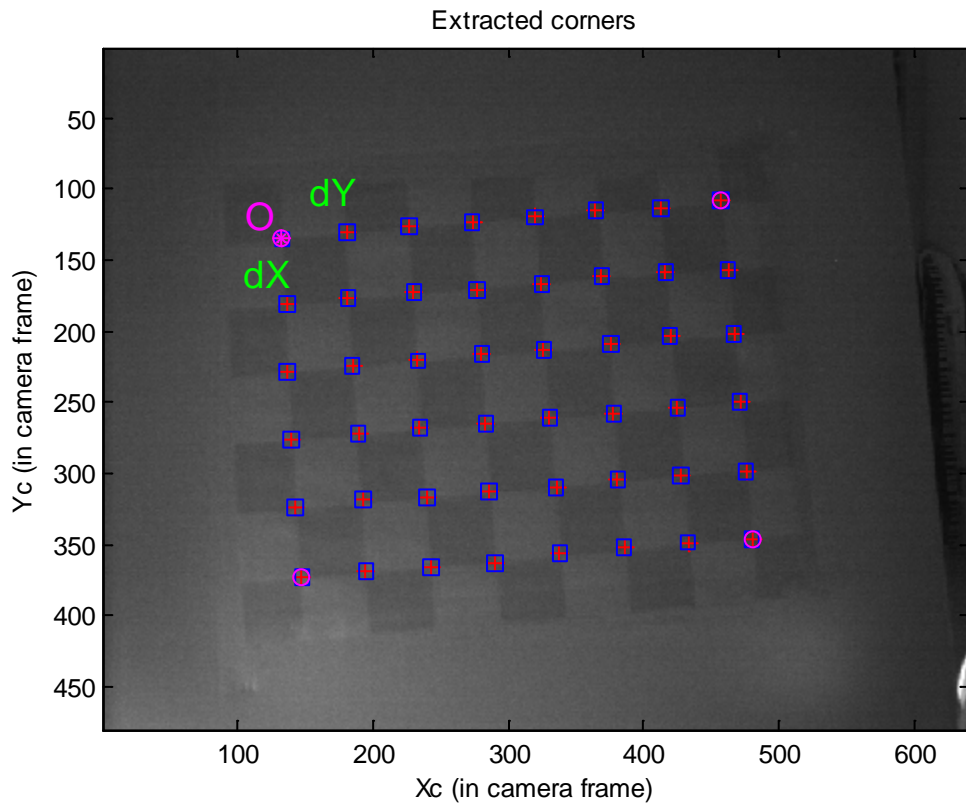


Figure 13. Chessboard corner extraction applied to a low contrast IR image

Similar to RGB camera calibration, some IR calibration images will exhibit more pixel re-projection error as compared to others in the set. Figure 14 displays the full set of IR calibration images without any suppressed images, where each colored cross represents a single pixel re-projection error of a calibration corner for a given homography configuration.

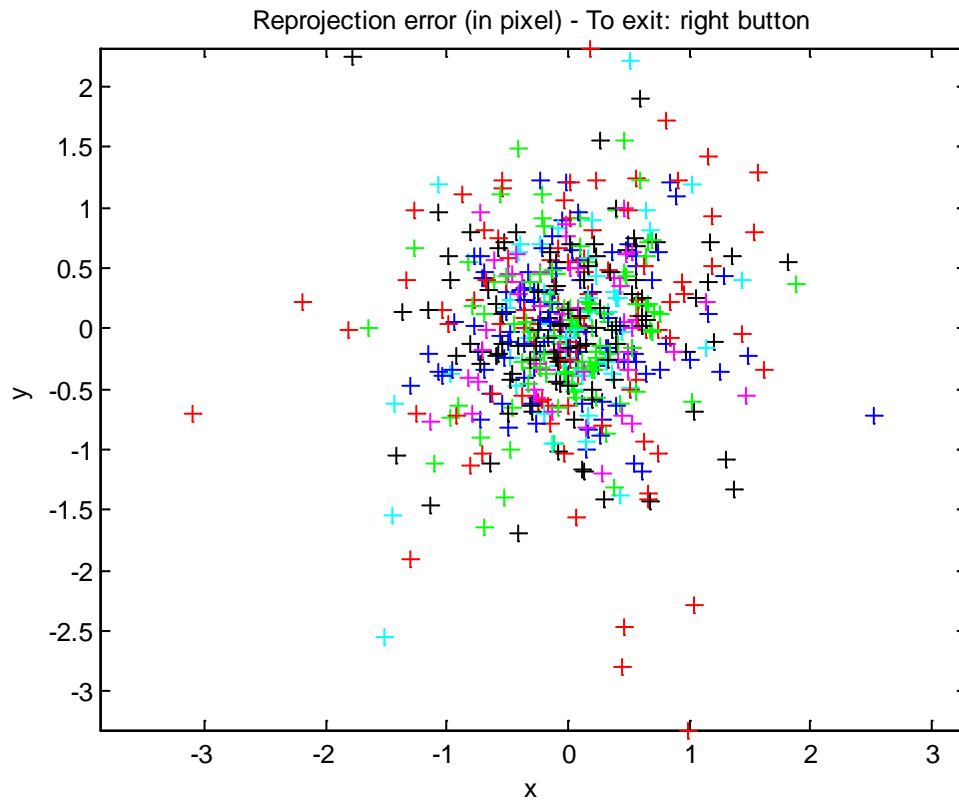


Figure 14. Pixel re-projection error of all IR calibration images

Again the IR images that exhibit a greater than average pixel re-projection error are excluded from the optimized calibration solution. In this case, all calibration images that yielded a detected corner pixel re-projection error greater than two pixels were suppressed from the calibration routine. Figure 15 displays the final set of images used for calibration and their respective pixel re-projection error.

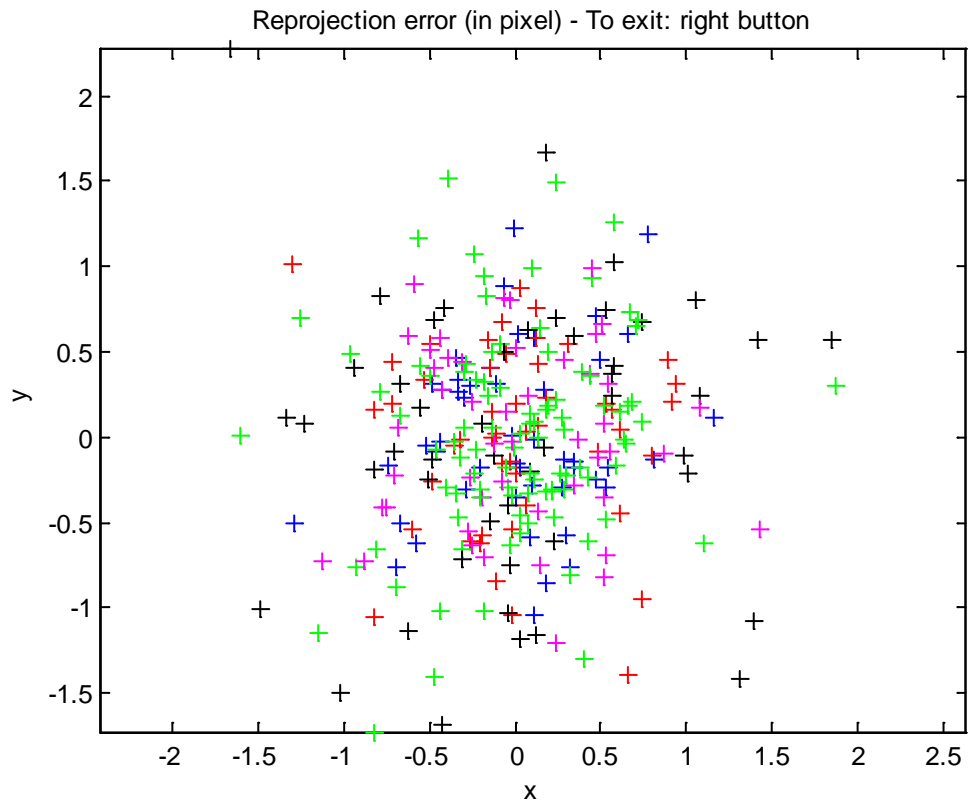


Figure 15. Pixel re-projection error of filtered IR calibration images

Using only the filtered IR calibration images that have relatively low pixel re-projection errors as compared to their counterpart images, a higher accuracy calibration solution was obtained. Table 2 summarizes the IR camera intrinsic parameters obtained from the optimized calibration solution.

Intrinsic Parameter	Parameter Value	Uncertainty
Focal Length	[585.34245 591.87060]	\pm [29.43746 30.53275]
Principal Point Offset	332.68929 252.19767]	\pm [11.86095 10.31408]
Radial Distortion (k_1, k_2, k_3)	[-0.15597 0.49463 0.00000]	\pm [0.08286 0.09225 0.00000]
Tangential Distortion (p_1, p_2)	[0.00038 0.00348]	\pm [0.00711 0.00925]
Pixel Re-projection Error	[0.56439 0.60367]	

Table 2. IR camera intrinsic parameters

The final root mean square pixel re-projection error was calculated to be about half a pixel. This result is worse than the one obtained for RGB camera calibration, however it is still reasonable for computer vision applications. This result can be improved if the contrast of the IR calibration image is increased, this may be possible by introducing an IR light sources that illuminates the calibration target. This could be a possible consideration for future calibration of IR cameras.

3.2.3 EXTRINSIC RGB AND IR CAMERA CALIBRATION

Having defined the intrinsic parameters of both the RGB and IR cameras it is now possible to obtain the extrinsic parameters between the two cameras. This extrinsic calibration is performed with the OpenCV library, using the 'StereoCalib()' subroutine that estimates the extrinsic parameters by minimizing the pixel re-projection error of both RGB and IR images for various calibration target poses. The subroutine takes as input: the intrinsic matrices of both RGB and IR cameras, the 3D metric coordinates of the corners of the calibration chessboard target, and 2D image coordinates of the observed corners of the chessboard calibration target for both the RGB and IR camera under different calibration target poses. The output of the function is a rotation matrix, $R_{RGB \rightarrow IR}$, and a translation vector, $\mathbf{t}_{IR \rightarrow RGB}$, that map the back-projected 3D metric point of the IR camera to the back-projected 3D metric points of the RGB camera [22]. Equation 2 defines the extrinsic transformation between the IR and RGB cameras.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{IR} = R_{RGB \rightarrow IR} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{RGB} + \mathbf{t}_{RGB \rightarrow IR}$$

Equation 2. Extrinsic relationship between RGB and IR cameras

Consequently, the rotation matrix and translation vector define the extrinsic parameters between the two cameras. The following rotation matrix and translation vector were estimated between the RGB camera and IR camera. These results are expressed in Equation 3.

$$R_{RGB \rightarrow IR} = \begin{bmatrix} 0.999 & 0.009 & -0.011 \\ -0.009 & 0.999 & -0.008 \\ 0.011 & 0.008 & 0.999 \end{bmatrix}$$

$$\mathbf{t}_{RGB \rightarrow IR} = [24.954 \quad -0.728 \quad -2.367]$$

Equation 3. Extrinsic parameter results between RGB and IR cameras

3.3 PROJECTOR CALIBRATION

As previously mentioned the projector can be regarded as an inverse of a camera, since it projects a digital image instead of capturing one. Consequently, if point correspondences are made between 2D points of the projector image and their 3D metric counterparts, the same camera calibration procedure (Zhang's method) may be applied to the projector to determine its intrinsic and extrinsic parameters.

To obtain the 2D points of the projected pattern, an image is taken by the RGB camera of the calibration plane that contains the projected calibration pattern displayed by the projector and a real calibration target attached to the calibration plane. This step is visualized in Figure 16.



Figure 16. RGB camera acquisition of real and projected calibration targets

Then a corner extraction function is executed on the captured RGB image, to define the 2D image locations of the real calibration target and the projected calibration target. Knowing the equation of the calibration plane in the RGB camera coordinate system allows for ray-plane intersection to define the projected 3D metric positions of the 2D image corner locations of the projected calibration pattern displayed by the projector. This allows making 2D image point to 3D metric point correspondences for the projector, which is essential for calibration using Zhang's method. The following sub-sections detail the steps needed to obtain the 3D metric point positions of their corresponding 2D point corner counterparts of the projected calibration image.

3.3.1 IDENTIFICATION OF CALIBRATION PLANE

Knowing the RGB camera intrinsic parameters, the extrinsic parameters between the RGB camera and the real calibration target can be computed with OpenCV's 'solvePnP()' subroutine that estimates the object pose from 2D image points and 3D

metric point correspondences, thus defining the rotation matrix, $R_{Plane \rightarrow RGB}$, and translation vector, $\mathbf{t}_{Plane \rightarrow RGB}$, between the RGB camera and the real calibration target attached to the calibration plane. Equation 4 relates the rotation matrix and the translation vector to its corresponding scalar components.

Now it is possible to compute the equation of the calibration plane in the RGB camera coordinate system. It is important to identify the orientation of this plane, as this will be used later in the calibration process to define the depth of a 3D point of the calibration plane. The 3D metric point positions of the projected pattern will depend on the intersection of the rays of each projected 2D image point and this calibration plane.

A plane in 3D space can be defined using a point, \mathbf{p} , that lies on the plane and a normal vector of the plane, $\hat{\mathbf{n}}$. Since, the extrinsic parameters obtained from OpenCV's 'solvePnP()' subroutine relate the optical center of the camera to a corner point on the calibration plane, the equation of the calibration plane may be recovered for each calibration plane pose.

The translation vector, $\mathbf{t}_{Plane \rightarrow RGB}$, provides the coordinates of a point detected on the calibration plane. Whereas, the third column of the rotation matrix, $R_{Plane \rightarrow RGB}$, provides the surface normal vector of the calibration plane containing the calibration pattern in the RGB camera coordinate frame, as expressed in Equation 4.

$$R_{RGB \rightarrow Plane} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad \mathbf{t}_{RGB \rightarrow Plane} = \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix}$$

Equation 4. Rotation matrix and translation vector of RGB camera with respect to calibration plane

$$\hat{\mathbf{n}} = \begin{pmatrix} r_{13} \\ r_{23} \\ r_{33} \end{pmatrix}_{RGB \rightarrow Plane} = \begin{pmatrix} a \\ b \\ c \end{pmatrix} = a\hat{x}_{RGB} + b\hat{y}_{RGB} + c\hat{z}_{RGB}$$

Equation 5. Definition of calibration plane normal vector

The normal vector is expressed in Equation 5, with respect to the RGB camera coordinate system. A point \mathbf{q} that lies on the calibration plane is expressed in Equation 6 with respect to the RGB camera coordinate system. Thus, an equation for the calibration plane is a set of all points that are located on the plane and satisfy Equation 7.

$$\mathbf{q} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}_{Plane} = X\hat{x}_{RGB} + Y\hat{y}_{RGB} + Z\hat{z}_{RGB}$$

Equation 6. Representation of point \mathbf{q} that is located on the calibration plane

$$\hat{\mathbf{n}} \cdot (\mathbf{q} - \mathbf{t}_{RGB \rightarrow Plane}) = 0$$

Equation 7. Vector equation of calibration plane that is satisfied by point \mathbf{q}

Note that the vector difference, $(\mathbf{q} - \mathbf{t}_{RGB \rightarrow Plane})$, defines a translation vector that lies on the calibration plane. Since the normal vector of the plane and the defined translation vector must be perpendicular, the plane equation may be defined by evaluating their dot product and setting it equal to zero due to their orthogonality. This result may be expanded and expressed as in Equation 8.

$$aX + bY + cZ - (at_x + bt_y + ct_z) = 0$$

Equation 8. Scalar equation of calibration plane

The calibration plane equation, Equation 8, provides a necessary constraint on the 3D metric points that will lie on the plane. This will later be used to define the scale parameter of the rays of the corresponding 3D metric points.

3.3.2 LOCALIZATION OF 3D POINTS

The 2D image locations of each projected calibration target corner are extracted from the RGB image using a corner extraction algorithm. So now it possible to define the 3D rays (Ray_x, Ray_y, Ray_z) that are projected from the RGB camera optical center through each 2D image detected corner (u_c, v_c) in the image plane. Equation 9 defines the RGB camera projection of a 2D image point to its corresponding 3D metric point counterpart. The corresponding 3D metric position $(\lambda_{RGB}Ray_x, \lambda_{RGB}Ray_y, \lambda_{RGB}Ray_z)$ of each projected corner on the actual calibration plane depend on the scale parameter, λ_{RGB} , which is unknown.

$$\lambda_{RGB} \begin{bmatrix} Ray_x \\ Ray_y \\ Ray_z \\ 1 \end{bmatrix}_{RGB} = [\tilde{K}_{RGB-Int.}]^{-1} \begin{bmatrix} u_c \\ v_c \\ 1 \end{bmatrix}_{RGB}$$

$$\text{where: } \tilde{K}_{RGB-Int.} = [K_{RGB-Int.} | \mathbf{0}_{3 \times 1}] = \begin{bmatrix} f_x & 0 & \sigma_x & 0 \\ 0 & f_y & \sigma_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Equation 9. RGB camera projection of 2D image point to its 3D metric point

Using the previous development of the calibration plane equation the scale parameter, λ , can be solved for, thus defining the true 3D metric locations of the projected corner points. Specifically, the intersection between the calibration plane and the 3D rays determines the 3D positions of the projected corners. So the correct scale parameter, λ , is the one that satisfies the equation of the calibration plane. Substituting the Cartesian components of 3D rays the equation of the calibration plane may expressed as Equation 10.

$$a(\lambda Ray_x) + b(\lambda Ray_y) + c(\lambda Ray_z) - \hat{\mathbf{n}} \cdot \mathbf{t}_{RGB \rightarrow Plane} = 0$$

Equation 10. Intersection of 3D rays and calibration plane

At this point all parameters of the plane ($\hat{\mathbf{n}}, \mathbf{t}_{RGB \rightarrow Plane}$) are known, the components of the 3D ray (Ray_x, Ray_y, Ray_z) are also known and the only unknown is scale parameter, λ . So we have one equation, Equation 11, and one unknown, thus allowing for a simple solution.

$$\lambda_{RGB} = \frac{\hat{\mathbf{n}} \cdot \mathbf{t}_{RGB \rightarrow Plane}}{\hat{\mathbf{n}} \cdot \begin{bmatrix} Ray_x \\ Ray_y \\ Ray_z \end{bmatrix}_{RGB}}$$

Equation 11. Expression for scale parameter, λ , in RGB camera frame

Knowing the scale parameter, λ , the 3D metric position of each projected corner can now be determined, according to Equation 12. This 3D position is computed for each detected projected corner found in the RGB image, and for each calibration plane configuration.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{RGB} = \lambda_{RGB} \begin{bmatrix} Ray_x \\ Ray_y \\ Ray_z \end{bmatrix}_{RGB}$$

Equation 12. Application of scale parameter, λ , to define 3D metric points in the RGB Camera reference frame

So now the 2D image point and 3D metric point correspondences are established for the projector digital image and the real world projected image displayed on the calibration plane. This allows using Zhang's method to perform calibration of the projector to determine its intrinsic parameters and the extrinsic parameters between the projector and calibration plane.

3.3.3 PROJECTOR CALIBRATION RESULTS

The previously detailed calibration procedure is implemented in the developed software to allow for auto-calibration for different SAR system settings. OpenCV's `calibrateCamera()` function is used to execute Zhang's method on the determined 2D image point and 3D metric point correspondences for the projector. The OpenCV calibration function does not report uncertainty associated with each intrinsic parameter. Also, the reported RMS pixel re-projection error is expressed as the square-root of the sum of squared RMS pixel re-projection errors for the two image axes $(\hat{u}, \hat{v})^T$.

Unlike a camera, the intrinsic parameters of the projector can change for various projector configurations. This is due to the fact the projector has an adjustable focal length that can be physically manipulated by the user, to adjust focus on the imaging plane. Also, the projector is prone to keystone distortion which occurs when the projector is aligned non-perpendicularly to the projection screen, or when the projection screen has an angled surface. The image that results from one of these misalignments will look trapezoidal rather than square. Figure 17 visualizes the possible configurations of the projector. Note the 'on-axis' case displays an ideal configuration where the principal axis of the projector is perpendicularly aligned with the projection screen. In this case there will be no keystone distortion and only projector lens distortions will affect the displayed image.

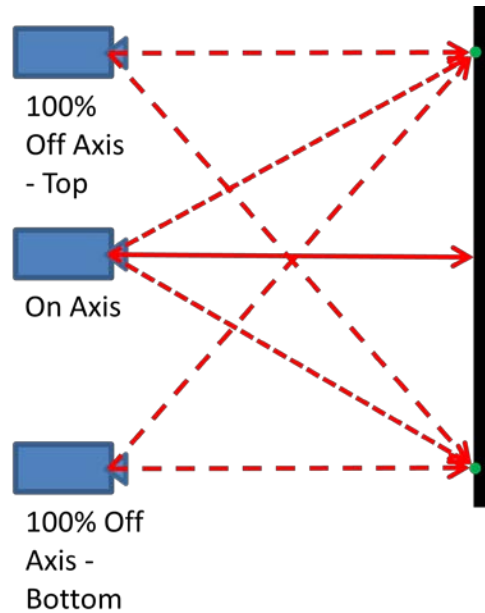


Figure 17. Possible projector configurations that affect keystone distortions

The utilized projector incorporates manual keystone distortion correction. If this setting is changed, the vertical principal point position of the projector is changed as well to account for the off axis projection onto the display screen. When the principal point of the projector is changed the image is projected through either the upper or lower part of the projector image acquisition lens. In result this also affects the intrinsic radial lens distortion parameters. However, the tangential lens distortion parameters are not affected because tangential distortion takes place when projector lens is not perfectly parallel to the imaging plane. During keystone adjustment, the projector lens orientation is not changed with respect to the imaging plane; only the vertical principal offset is adjusted.

Due to all of the previously mentioned changes in projector's intrinsic parameters, the projector has to be fully calibrated for each new SAR system setup

configuration. Table 3 summarizes the intrinsic calibration results obtained for a unique projector configuration using a set of ten images.

Intrinsic Parameter	Parameter Value
Focal Length	[1772.61112 1869.60390]
Principal Point Offset	[375.22251446.22731]
Radial Distortion (k_1, k_2, k_3)	[0.77017, -20.44847171.57636]
Tangential Distortion (p_1, p_2)	[-0.00230 0.00204]
Pixel Re-projection Error	[0.582298]

Table 3. Projector camera intrinsic parameters

For the given SAR system using greater than ten images per calibration set tends to only increase the pixel re-projection error and decrease the accuracy of both intrinsic and extrinsic parameters. The extrinsic parameters found for the following calibration setup are defined in Equation 13, where the translation vector is expressed in millimeter metric units.

$$R_{RGB \rightarrow Projector} = \begin{bmatrix} 0.999 & -0.016 & -0.039 \\ 0.013 & 0.996 & -0.082 \\ 0.040 & 0.081 & 0.995 \end{bmatrix}$$

$$t_{RGB \rightarrow Projector} = [-130.130 \quad -52.889 \quad 111.102]$$

Equation 13. Extrinsic parameter result between RGB camera and projector

The extrinsic parameters between the RGB camera and projector will change for every unique configuration of the Kinect sensor and projector. This means that if the Kinect sensor is moved with respect to the projector to a different position and orientation that pre-calibrated extrinsic results are no longer valid and need to be solved for again. If the focal length and keystone stone distortion of the projector are not changed during the re-positioning of the SAR system the intrinsic parameters will remain the same and only the extrinsic parameters will change.

CHAPTER IV

USER INTERACTION WITH SAR SYSTEM

The described SAR system is designed to primarily support IR stylus user input, however it can also obtain depth information due the Kinect's structured light stereo pair that consists of the IR Camera and an IR projector. Unfortunately, both inputs cannot be obtained at the same time due to Kinect hardware limitations that allow it to only stream IR or depth data at a single point in time. Even if the Kinect could stream both data streams at the same time, Kinect's IR projector would add its projection pattern to each IR image, thus greatly increasing image noise. This would hinder any computer vision application that is applied to the IR camera data stream due to additional IR noise.

Figure 18 displays the Kinect's IR projector pattern on a planar target.

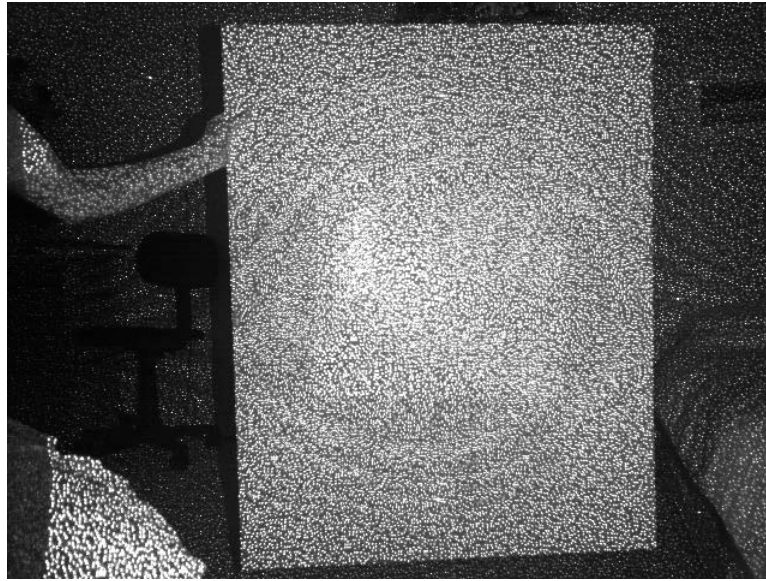


Figure 18. Kinect IR projector pattern

As previously stated, IR stylus input is the primary method for user and system interaction. The IR stylus is used to write or draw content on the 'digital whiteboard' region and the projector will display the traced movement of the stylus that characterizes the user defined contour. In case of depth interaction, the system will display an image of a button on the calibration plane using the projector and the user can physically displace depth in the button region by 'pressing' and activating the button.

For both of these types of user interaction, various transformations must be applied to the sensor inputs for the system to provide coherent feedback. The following sections explain the required transformations and their corresponding mathematical representations.

4.1 PROJECTIVE TRANSFORMATIONS

In order to display the IR stylus movement on the calibration plane using the projector, a correspondence problem must be solved that maps the detected IR stylus centroid position in the IR image to the corresponding 2D image point that will be displayed by the projector, resulting in a 3D metric point projected onto the calibration plane. Since the IR camera of the Kinect can also function as a depth sensor, the presented transformations apply to both IR and depth data streams.

The IR camera is mainly sensitive to light wavelengths above the visible range, so it does not register the light output of the projector. This restricts a direct extrinsic calibration between the two devices, and thus presents a problem for a direct mapping from the IR camera to the projector. Due to this the RGB camera is used as an intermediate sensor to which both the IR and projector are calibrated to. Consequently, the SAR system coordinate reference frame is placed at the optical center of the RGB camera to facilitate perspective transformations. This way a transformation may be defined from the IR camera to the RGB camera, and then another transformation from the RGB camera to the projector to solve the presented correspondence problem. The figure below visualizes these described transformations. The solid orange lines represent projection transformations of a 2D image point to a 3D metric point. Whereas, the dashed orange lines represent back-projection transformations from 3D metric point back to 2D image points in alternate reference frame. In essence, Figure 19 visualizes all the transformations applied to each acquired point by the IR camera.

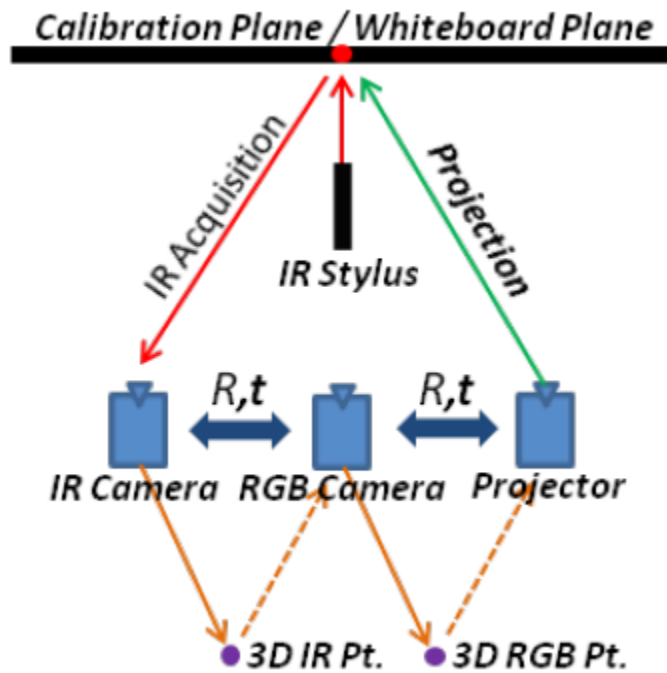


Figure 19. SAR system with visualized transformations

4.1.1 IR CAMERA TO RGB CAMERA TRANSFORMATION

The following transformations take a 2D image point of the IR camera and map it to the corresponding 2D image point of the RGB camera. In order to accomplish this, first the 2D image point of the IR camera must be back-projected to a metric 3D world point. This is accomplished by the following transformation expressed in Equation 14.

$$\lambda \begin{bmatrix} Ray_x \\ Ray_y \\ Ray_z \\ 1 \end{bmatrix}_{IR} = [\tilde{K}_{IR-Int.}]^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_{IR}$$

$$\tilde{K}_{IR-Int.} = [K_{IR-Int.} | \mathbf{0}_{3 \times 1}] = \begin{bmatrix} f_x & 0 & \sigma_x & 0 \\ 0 & f_y & \sigma_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Equation 14. IR camera projection of 2D image point to corresponding 3D metric point

The 3D metric point of the back-projected 2D point of the IR is image expressed in homogeneous coordinates, $[\lambda Ray_x \quad \lambda Ray_y \quad \lambda Ray_z \quad \lambda]_{IR}^T$, which are defined by a scale factor, λ . It can be obtained from the extrinsic parameters ($R_{IR-Plane}$, $\mathbf{t}_{IR-Plane}$) between the IR camera and the calibration plane, similar to how the RGB camera computed the 3D metric points of the detected projector pattern during projector calibration. The direct expression for the scale parameter is expressed in Equation 15.

$$\lambda_{IR} = \frac{\hat{\mathbf{n}} \cdot \mathbf{t}_{IR \rightarrow Plane}}{\hat{\mathbf{n}} \cdot \begin{bmatrix} Ray_x \\ Ray_y \\ Ray_z \end{bmatrix}_{IR}}$$

$$\hat{\mathbf{n}} = \begin{pmatrix} r_{13} \\ r_{23} \\ r_{33} \end{pmatrix}_{IR \rightarrow Plane} \quad R_{IR \rightarrow Plane} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

Equation 15. Expression for scale parameter, λ , in IR camera frame

Knowing the scale parameter, λ , the 3D metric position of each projected corner can now be determined by Equation 16. This 3D position is computed for each detected projected corner found in the IR image, and for each calibration plane configuration.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{IR} = \lambda_{IR} \begin{bmatrix} Ray_x \\ Ray_y \\ Ray_z \end{bmatrix}_{IR}$$

Equation 16. Application of scale parameter, λ , to define 3D metric point in IR camera reference frame

It is important to note that the obtained 3D metric points are represented in the IR camera coordinate system. In order to obtain the corresponding pixel positions on the image plane of the RGB camera, the 3D points need to be converted to the RGB Camera coordinate system using the extrinsic parameters ($R_{IR \rightarrow RGB}$, $t_{IR \rightarrow RGB}$) between the two cameras obtained from previous calibration. Then these transformed 3D points can be projected onto the image plane of the RGB camera using the intrinsic parameters ($K_{Color-Int.}$) of the RGB camera. This transformation is expressed in Equation 17.

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_{RGB} = [K_{RGB-Int.} | \mathbf{0}_{3 \times 1}] \begin{bmatrix} R_{IR \rightarrow RGB} & t_{IR \rightarrow RGB} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}_{IR}$$

$$\lambda \mathbf{W}_{RGB}' = [\hat{K}_{Color-Int.}] [\hat{K}_{IR \rightarrow RGB-Ext.}] \mathbf{W}_{IR}$$

Equation 17. Projection of 3D metric point expressed in IR camera reference frame to the corresponding 2D image point in the image plane of the RGB camera

4.1.2 RGB CAMERA TO PROJECTOR TRANSFORMATION

The following transformations take a 2D image point of the RGB camera and map it to the corresponding 2D image point of the projector. The methodology of this transformation is identical to the previous mapping of a 2D image point of the IR camera to a 2D image point of the RGB camera, except that different intrinsic and extrinsic parameters need to be used, that correspond to the RGB camera and projector. Accordingly, first the 2D image point of the RGB camera must be back-projected to a metric 3D world point. This is accomplished by the following transformation expressed in Equation 18.

$$\lambda \begin{bmatrix} Ray_x \\ Ray_y \\ Ray_z \\ 1 \end{bmatrix}_{RGB} = [\tilde{K}_{IR-Int.}]^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_{RGB}$$

$$\tilde{K}_{RGB-Int.} = [K_{IR-Int.} | \mathbf{0}_{3 \times 1}] = \begin{bmatrix} f_x & 0 & \sigma_x & 0 \\ 0 & f_y & \sigma_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Equation 18. RGB camera projection of 2D image point to corresponding 3D metric point

The 3D metric point of the back-projected 2D point of the RGB is image expressed in homogeneous coordinates, $[\lambda Ray_x \quad \lambda Ray_y \quad \lambda Ray_z \quad \lambda]_{RGB}^T$, which is defined by a scale factor, λ . It can be obtained from the extrinsic parameters $(R_{RGB-Plane}, \mathbf{t}_{RGB-Plane})$ between the IR camera and the calibration plane. The direct expression for the scale parameter is expressed in Equation 19.

$$\lambda_{RGB} = \frac{\hat{\mathbf{n}} \cdot \mathbf{t}_{RGB \rightarrow Plane}}{\hat{\mathbf{n}} \cdot \begin{bmatrix} Ray_x \\ Ray_y \\ Ray_z \end{bmatrix}_{RGB}}$$

$$\hat{\mathbf{n}} = \begin{pmatrix} r_{13} \\ r_{23} \\ r_{33} \end{pmatrix}_{RGB \rightarrow Plane} \quad R_{RGB \rightarrow Plane} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

Equation 19. Expression for scale parameter, λ , in RGB camera frame

Knowing the scale parameter, λ , the 3D metric position of each projected corner can now be determined by Equation 20. This 3D position is computed for each detected projected corner found in the RGB image, and for each calibration plane configuration.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{RGB} = \lambda_{RGB} \begin{bmatrix} Ray_x \\ Ray_y \\ Ray_z \end{bmatrix}_{RGB}$$

Equation 20. Application of scale parameter, λ , to define 3D metric point in RGB camera reference frame

It is important to note that the obtained 3D metric points are represented in the RGB camera coordinate system. In order to obtain the corresponding pixel positions on the image plane of the projector, the 3D points need to be converted to the projector coordinate system using the extrinsic parameters ($R_{RGB \rightarrow Projector}$, $\mathbf{t}_{RGB \rightarrow Projector}$) between the two cameras obtained from previous calibration. Then these transformed 3D points can be projected onto the image plane of the projector using the intrinsic

parameters ($K_{\text{Projector-Int.}}$) of the projector. This transformation is expressed in Equation 21.

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}_{\text{Projector}} = [K_{\text{Projector-Int.}} | \mathbf{0}_{3 \times 1}] \begin{bmatrix} R_{\text{RGB} \rightarrow \text{Projector}} & \mathbf{t}_{\text{RGB} \rightarrow \text{Projector}} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}_{\text{Projector}}$$

$$\lambda \mathbf{W}_{\text{Projector}}' = [\hat{K}_{\text{RGB} \rightarrow \text{Projector}}][\hat{K}_{\text{RGB} \rightarrow \text{Projector-Ext.}}] \mathbf{W}_{\text{Projector}}$$

Equation 21. Projection of 3D metric point expressed in RGB reference frame to the corresponding 2D image point in the image plane of the Projector

4.2 IR STYLUS AND DEPTH DETECTION

IR Stylus detection is performed by performing digital image processing on each acquired frame by the IR camera. This involves thresholding the IR digital image by a given high intensity value, like 250, since the IR stylus return will be very bright on the acquired image. All pixels below this threshold are zeroed out to the lowest intensity, and all pixels above or equal to this threshold are assigned the highest possible intensity. Performing this operation yields a binary black and white image. Then a blob detection algorithm is executed on the binary image according to the cvBlobsLib software library. This provides a contour of the detected IR stylus blob in the image of the IR camera. Afterward the detected blob is centroided to give its sub-pixel position. The result is the floating point coordinates of the center pixel of the detected blob corresponding to IR stylus return. This procedure defines the detection of the IR stylus.

Depth information is registered by the IR camera when coupled with the IR projector. Its detection is simply the displacement of depth. For the case of the projected button, a region of interest is selected in the IR camera that corresponds to the observed region of the projected depth button. The program monitors the change in depth of this region and once the depth has changed past a certain threshold the button is considered to be pressed by the user and activated.

4.3 IR STYLUS INPUT VISUALIZATION

Having detected the IR stylus sub-pixel position in a set of consecutive IR frames, it is possible to connect these points and estimate a contour that represents the user's input according to the IR stylus. This detected contour needs to be transformed to the appropriate image coordinates of the displayed image of the projector in order to coherently visualize the contour on the calibration/whiteboard surface.

Given a dense set of neighboring points, it possible to simply connect adjacent points with a small line; producing a contour defining the IR stylus movement. However, this can lead to noise contour detection, and it may be not possible to acquire these IR points in a dense enough manner due to the computational burden placed on the computer. For fast multi core central processing units (CPUs) that have a frequency of above 3.0 GHz this is not a problem, however it becomes apparent for older hardware that utilizes slower CPUs. In this case the resulting contour appears jagged and unpleasing to the human eye. This can be corrected for by applying the 'Smooth Irregular

Curves' algorithm developed by Junkins [23]. The following section details its application.

4.3.1 PARAMETRIC CURVE FITTING

The following algorithm called 'Smooth Irregular Curves' developed by Junkins considers a set of 2D points much like those attained from IR stylus detection and aims at estimating the best smooth and accurate contour passing through these input points by sequential processing. The arc length of the curve is chosen as an independent variable for interpolation because its length is monotonically increasing along the curve at every discrete point of the data set. Also, the method treats both x and y coordinates independently thus allowing for a parametric estimation of each individual variable. The method considers a local subset of six points and attempts to fit the best smooth curve passing through all of these points, by approximating the best fit cubic polynomial for both x and y coordinates in between each data point segment. This is performed by dividing the six point data set into a left, middle and right region, where data point position is considered along with the slope at each point. The contour solution is constrained at each data point end that defines a region. The point position and the slope at the point position are used as constrained parameters. As this is a sequential algorithm it marches from point to point reassigning the left, middle and right regions and using their constraint information provides the best smooth cubic polynomial approximation at the middle segment. Figure 20 visualizes the output of the algorithm applied to a discrete set of detected IR stylus points.

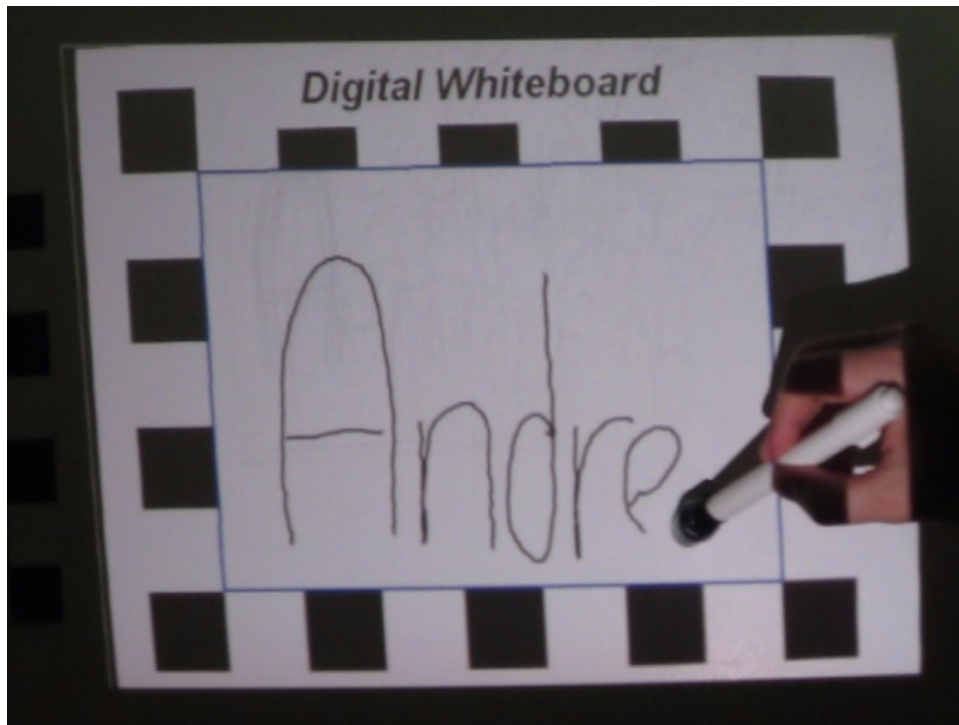


Figure 20. User interaction with developed SAR system

The interaction of a user with the SAR system is displayed in Figure 20. The displayed digital contour of the user's hand writing is refreshed at a frequency of approximately two Hz. At each update, a new six point approximated polynomial is added to the existing contour. This allows for near real-time user interaction with the developed SAR system.

CHAPTER V

CONCLUSIONS

The following key steps summarize the development of the described SAR system. The first step is preparing the hardware components and setting up the software interface for system control. Afterward, system calibration is performed to determine the intrinsic and extrinsic parameters that define the proper transformations between the utilized sensors. Finally, this enables IR stylus user input to be displayed digitally using the PC projector. Following these steps grants the realization of a SAR system.

5.1 SYSTEM IMPLEMENTATION

For the developed SAR system a C++ program was written to demonstrate its utility and validate the realization procedure. The program begins with auto-calibration, which determines all of the extrinsic parameters between the sensors given a specific SAR system configuration. This procedure also determines the intrinsic parameters of the projector for every configuration as these parameters can change from setup to setup. Once calibration is complete a 'Digital Blackboard' region is displayed on the calibration plane, then the user can begin writing with the IR stylus on the projected region designated by a blue rectangle. The projector will display the contours generated by the IR stylus on the projected region. The described user interaction with the developed SAR system is displayed in Figure 20.

5.2 SYSTEM IMPROVEMENTS

One disadvantage of the developed calibration procedure is that for projector calibration a known real world calibration pattern must be attached to the calibration plane. Also, since the RGB camera captures an image with both real and projected calibration patterns, the image is divided into two parts for calibrating the RGB camera to the calibration plane and the other for calibrating the Projector to the calibration plane. This reduces the utilized image resolution for each individual calibration step, thus slightly decreasing the accuracy of obtained intrinsic and extrinsic parameters. Using other IR and RGB cameras with greater resolution and faster acquisition rates can help improve the SAR system performance.

Another disadvantage of the developed SAR system is that the user can block some of the projected image displayed by the projector. This is seen in Figure 20, as the user's hand is blocking some of the projection on the right side of the image. If a transparent planar material is used for the projection region the projector can be placed on the opposite side from the user to prevent the user from blocking the image projection. This would improve the quality of the projected image and the accuracy of IR stylus movement re-projection.

NOMENCLATURE

$(u, v)^T$	Two dimensional image point (\mathbf{w}')
$(X, Y, Z)^T$	Three dimensional world point (\mathbf{W})
API	Application program interface
AR	Augmented reality
SAR	Spatial augmented reality
CCD	Charge coupled device imager
CMOS	Complementary metal–oxide–semiconductor imager
CPU	Central processing unit
IR	Infra-red wavelength
RGB	Color made up of red, green blue channels
$K_{Int.}$	Intrinsic matrix
$K_{Ext.}$	Extrinsic matrix
\mathbf{t}	Translation vector
$\hat{\mathbf{n}}$	Normal vector
R	Rotation matrix
RMS	Root mean square of a set value
$k_1, k_2, k_3, k_4, k_5, k_6$	Radial distortion coefficients
p_1, p_2	Tangential distortion coefficients
f_x	Focal length
f_y	Focal length

σ_x	Principal point offset
σ_y	Principal point offset
η	Focal length aspect ratio
τ	Pixel skew parameter
λ	Homogeneous scale parameter

REFERENCES

- [1] Finley, Klint. "Kinect Drivers Hacked – what Will YOU Build with it?" readwrite hack, accessed 1/10, 2013, <http://readwrite.com/2010/11/10/kinect-drivers-hacked--what-w>.
- [2] Leigh, Alexander. "Microsoft Kinect Hits 10 Million Units, 10 Million Games." Gamasutra, accessed 1/7, 2013, http://www.gamasutra.com/view/news/33430/Microsoft_Kinect_Hits_10_Million_Units_10_Million_Games.php
- [3] Johnson, Joel. "'The Master Key': L. Frank Baum Envisions Augmented Reality Glasses in 1901.", accessed 1/5, 2013, <http://moteandbeam.net/the-master-key-l-frank-baum-envisions-ar-glasses-in-1901>.
- [4] Laurel, Brenda. 1991. *Computers as theatre*, 40-65: Reading, Mass. : Addison-Wesley Pub.
- [5] Sutherland, Ivan. 1968. "A Head-Mounted Three Dimensional Display." *Proceeding AFIPS '68 (Fall, Part I) Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*: 757-764.
- [6] Caudell, Thomas and David Mizell. 1992. "Augmented Reality: An Application of Heads-Up Display Technology to Manual Manufacturing Processes." *Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences* 2: 659-669.
- [7] Azuma, Ronald. 1997. "A Survey of Augmented Reality." *In Presence: Teleoperators and Virtual Environments* 6 (4): 355-385.
- [8] Feiner, Steven, Blair Macintyre, and Dorée Seligmann. 1993. "Knowledge-Based Augmented Reality." *Communications of the ACM - Special Issue on Computer Augmented Environments: Back to the Real World* 36 (7): 53-62.
- [9] Sielhorst, Tobias, Marco Feuerstein, Joerg Traub, Oliver Kutter, and Nassir Navab. 2006. "CAMPAR: A Software Framework Guaranteeing Quality for Medical Augmented Reality." *International Journal of Computer Assisted Radiology and Surgery* 1 (1): 29-30.
- [10] Law, Alvin and Daniel Aliaga. 2012. "Spatial Augmented Reality for Environmentally-Lit Real-World Objects." *IEEE Virtual Reality*: 7-10.

- [11] Sheng, Yu, Theodore Yapo, and Barbara Cutler. 2010. "Global Illumination Compensation for Spatially Augmented Reality." *Computer Graphics Forum*: 387-396.
- [12] Olwal, Alex, Jonny Gustafsson, and Christoffer Lindfors. 2008. "Spatial Augmented Reality on Industrial CNC-Machines." *Proceedings of SPIE 2008 Electronic Imaging* 6804 (09).
- [13] Talaba, Doru, Imre Horvath, and Kwan Lee. 2010. "Special Issue of Computer-Aided Design on Virtual and Augmented Reality Technologies in Product Design." *Computer-Aided Design* 42 (5): 361-363.
- [14] Bimber, Oliver, Daisuke Iwai, Gordon Wetzstein, and Anselm Grundhöfer. 2008. "The Visual Computing of Projector-Camera Systems." *Computer Graphics Forum* 27 (8): 2219-2245.
- [15] OpenKinect. "Protocol Documentation.", accessed 01/14, 2013, http://openkinect.org/wiki/Protocol_Documentation#Control_Commands;a=summary.
- [16] PrimeSense, LTD. "Developers>openni.", accessed 1/13, 2013, <http://www.primesense.com/developers/open-ni/>.
- [17] Falcao, Gabriel, Natalia Hurtos, and Joan Massich. 2008. "Plane-Based Calibration of Projector-Camera System." Master, VIBOT - Erasmus Mundus Masters in Vision and Robotics.
- [18] Zhang, Zhengyou. 2000. "A Flexible New Technique for Camera Calibration." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (11): 1330-1334.
- [19] Bouguet, Jean-Yves. "Camera Calibration Toolbox for Matlab.", accessed 6/21, 2012, http://www.vision.caltech.edu/bouguetj/calib_doc/.
- [20] Brown, Duane. 1971. "Close-Range Camera Calibration." *Photogrammetric Engineering* 37 (8): 855-866.
- [21] Hartley, Richard and Andrew Zisserman. 2004. *Multiple View Geometry in Computer Vision*. 2nd ed. New York, NY, USA: Cambridge University Press.
- [22] Bradski, Gary and Adrian Kaehler. 2008. *Learning OpenCV: Computer Vision with the OpenCV Library*, edited by Mike Loukides. 2nd edition ed. Sebastopol, CA: O'Reilly Media.

- [23] Junkins, John and James Jancaitis. 1972. "Smooth Irregular Curves."
Photogrammetric Engineering 38 (6): 565-573.
- [24] Mohr, Roger and Triggs Bill. 1996. "A Tutorial Given at ISPRS." *XVIIIth International Symposium on Photogrammetry & Remote Sensing*.

APPENDIX

COMPUTER VISION BACKGROUND

Digital image formation of a real world scene consists of two unique components: geometric and spectroscopic [24]. The geometric component captures the shape of the scene observed, as a real world surface element in 3D space is projected to a 2D pixel element on the image plane. The spectroscopic component defines the image's intensity or color of a pixel that represents a surface element of the scene captured. The main focus of this thesis is on the geometric component of the image formation procedure.

Usually Euclidian geometry is used to model point, lines, planes and volumes. However, Euclidian geometry has a disadvantage that it can't easily represent points at infinity. For example, if two parallel lines are extended to infinity they will meet at a vanishing point, which is a special case of Euclidian geometry that presents difficulties in its expression. Also, when using Euclidian geometry the projection of a 3D point onto a 2D image plane requires the application of a perspective scaling operation, this involves division thus making it into a non-linear operation. Due to these disadvantages, Euclidian geometry is unfavorable and projective geometry is used to model the geometric relationship between 3D and 2D points. To utilize projective geometry a projective space needs to be defined that allows for projective transformations.

A.1 PROJECTIVE SPACE

A three dimensional point in Euclidian space is described using a three element vector, $(X, Y, Z)^T$. This point representation is described using inhomogeneous coordinates. In contrast, a three dimensional point in projective space is described using a four element vector, $(X_1, X_2, X_3, X_4)^T$, using homogeneous coordinates. A mathematical relationship allows for conversion between these two geometric spaces, as expressed in Equation 22.

$$X = \frac{X_1}{X_4}, Y = \frac{X_2}{X_4}, Z = \frac{X_3}{X_4} \text{ subject to } X_4 \neq 0$$

Equation 22. 3D relationship between Euclidian and projective space

Here the homogeneous coordinate, X_4 , is defined as the scale parameter, λ . This mapping can be generalized, as any n-dimensional Euclidian space can be represented by a (n+1)-dimensional projective space. This is expressed in Equation 23.

$$(X_1, X_2 \dots X_n)^T \rightarrow (\lambda X_1, \lambda X_2 \dots \lambda X_n, \lambda)^T$$

Equation 23. Generalized relationship between Euclidian and projective space

In a digital image each pixel represents an incoming line of sight of an incoming ray of light from a surface point in 3D space. Using two coordinates to describe an incoming ray on the image plane yields inhomogeneous coordinates. However, any 3D

point along this ray projects to the same digital image coordinate or pixel. So another way of representing this ray is done by arbitrarily choosing a 3D point along the ray's direction, and using three 'homogeneous' coordinates to define its position.

A.1.1 PROJECTIVE TRANSFORMATION

A projective transformation does not preserve parallelism, length, and angle. Although, it still preserves collinearity and incidence. Affine transformations are a unique subset of projective transformations. An affine transformation preserves collinearity, incidence and parallelism. Both affine and projective transformations are linear transformations that map one vector space into another by matrix multiplication. A linear transformation is one that preserves vector addition and scalar multiplication.

A.2 PINHOLE CAMERA MODEL

In order to model the image acquisition of a camera and the image projection of a projector the famous and commonly used pinhole camera model utilized. In essence, a camera observes real world 3D points and maps them to corresponding 2D digital points on the image plane. In contrast, the projector emits light rays from 2D digital image to corresponding projected 3D world points. So the projector may be regarded as the inverse of camera, and both can be modeled using the pinhole camera model.

To begin, consider a simplified pinhole camera model. This ideal pinhole camera model defines the relationship between a real world 3D point $(X, Y, Z)^T$ and its

corresponding 2D projection point, w' , on the image plane. This model is visualized below in Figure 21.

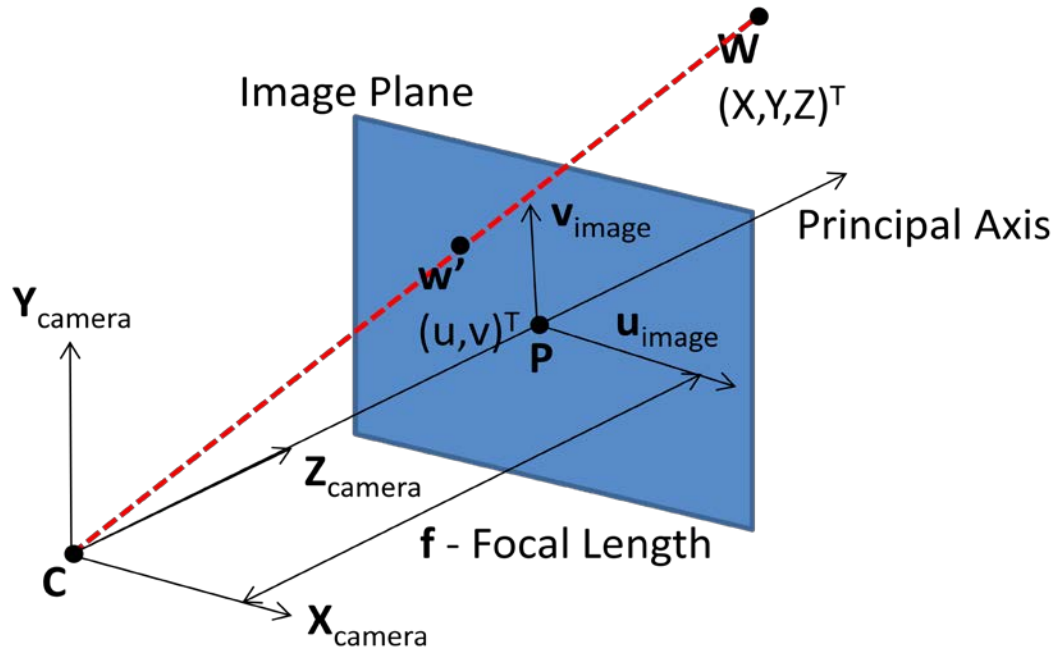


Figure 21. Pinhole camera model

The origin of a Euclidean coordinate system for the camera is placed at point C , defined as the camera's optical center where all the incoming light rays coalesce into a single point. The X_{camera} - Y_{camera} plane at C forms the principal plane. The principal axis also known as the optical axis, Z_{camera} , is projected from C , such that it is perpendicular to the image plane also known as the focal plane at a focal distance, f . The intersection of the principal axis with the image plane is defined as the principal point, P . Note, the principal plane is parallel to the image plane. Using the properties of similar triangles a

point, \mathbf{W} , in the 3D world, is mapped to a 2D point, \mathbf{w}' , in the image plane. The geometry for obtaining \mathbf{u} -component of the point, \mathbf{w}' , on the image plane is visualized in Figure 22.

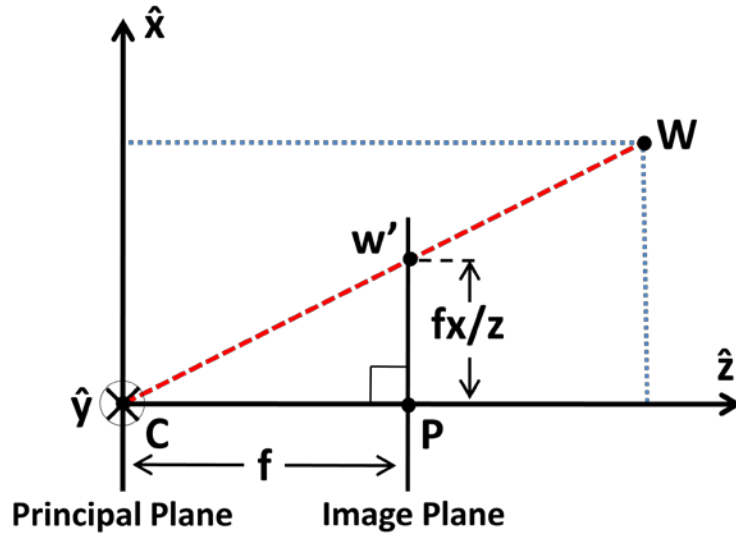


Figure 22. Geometry for computing the x-coordinate on the image plane

Likewise, the geometry for obtaining the \mathbf{v} -component of 2D image point, \mathbf{w}' , is shown in Figure 23.

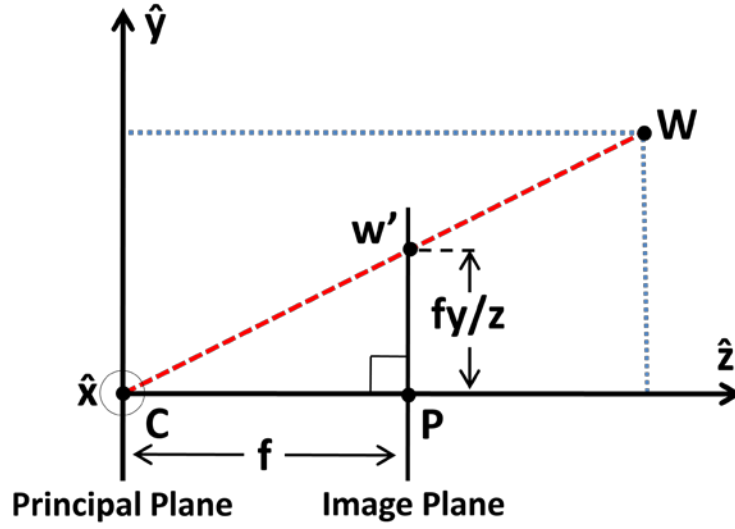


Figure 23. Geometry for computing the y-coordinate on the image plane

Using similar triangles, the projection of a 3D world point $\mathbf{W}(X, Y, Z)^T$ onto a corresponding 2D image point, \mathbf{w}' , can be expressed as Equation 24.

$$u = \frac{Xf}{Z} \quad v = \frac{Yf}{Z}$$

Equation 24. Projection of a world 3D point to a 2D image point

This relationship is determined using non-homogeneous coordinates expressed in the Euclidian framework and it requires a non-linear division operation. If the both the object-space and image-space points are expressed as homogeneous vectors using a projective space framework, it is possible to write a linear mapping relationship between the two points using matrix notation as expressed in Equation 25.

$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Equation 25. Ideal pinhole camera projection

A.2.1 INTRINSIC PARAMETERS

The previous relationship holds for the simplest ideal pinhole model that assumes the projection is made through the center of the camera, so that the origin of the image plane coordinates resides at the principal point. In reality, the CCD camera's principal point may be slightly offset (σ_x , σ_y) in both x and y directions from the center of the image plane due to manufacturing defects. This requires a translation of the image plane coordinate system to the true offset origin. This can be expressed by the modified intrinsic transformation as seen in Equation 26:

$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{bmatrix} f & 0 & \sigma_x & 0 \\ 0 & f & \sigma_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Equation 26. Pinhole camera projection

Also, it is initially assumed that the "image coordinates are Euclidean coordinates having equal scales in both axial directions", thus a pixel aspect ratio of 1:1. However, the image plane pixels may not be square, thus introducing a pixel skew factor, τ , and pixel aspect ratio, η . The skew factor, τ , is defined as the angle in radians between the y-

axis and the side of a neighboring pixel, α , times the x-component of the focal length, f_x , as expressed in Equation 27.

$$\tau = \alpha \cdot f_x$$

Equation 27. Skew factor relation

Thus when the pixels are not skewed the skew factor, τ , is equal to zero.

Figure 24 visualizes skewed CCD camera pixels. The Kinect cameras are assumed to have square pixels thus a zero skew factor.

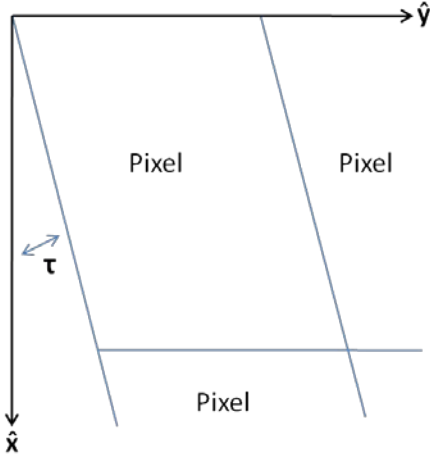


Figure 24. Skewed CCD camera pixels

The aspect ratio, η , is simply a ratio of the y-component of the camera focal length with the x-component of the focal length. Equation 28 defines the mathematical relation for aspect ratio.

$$\eta = \frac{f_y}{f_x}$$

Equation 28. Aspect ratio relation

Given the previous definitions it is possible to incorporate the principal point offset and a skewed pixel aspect ratio into a matrix central projection formula. The final pinhole camera mathematical model is expressed in Equation 29.

$$\lambda \mathbf{w}' = \lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{bmatrix} f & \tau & \sigma_x & 0 \\ 0 & \eta f & \sigma_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = [K_{Int.} | \mathbf{0}_{3 \times 1}] \mathbf{W}$$

Equation 29. Perspective transformation using five intrinsic parameters

Isolating the 3x3 matrix with the five intrinsic parameters defines the intrinsic camera matrix, $K_{Int.}$. The intrinsic camera matrix may be expressed as Equation 30.

$$K_{Int.} = \begin{bmatrix} f & \tau & \sigma_x \\ 0 & \eta f & \sigma_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} f_x & \alpha \cdot f_x & \sigma_x \\ 0 & f_y & \sigma_y \\ 0 & 0 & 1 \end{bmatrix}$$

Equation 30. Intrinsic camera matrix

So far in this development, the pinhole camera model did not account for the presence of lens distortions of a camera. Most cameras use a lens to focus the incoming light onto the camera imager (CCD or CMOS) located at the optical center of the camera. The presence of a lens introduces non-linear geometric distortions in the acquired image. Most lens distortions are radially symmetric due to the symmetry of the camera lens. This gives rise to two main categories of lens distortions: 'barrel distortion' and 'pincushion distortion'. In 'barrel distortion' the image magnification decreases with the distance from the optical center of the lens. For 'pincushion distortion', the image magnification increases with the distance from the optical center of the lens. Most real lenses that are not designed for wide field of view applications, so they have little tangential distortion with slight radial distortion.

For presented augmented reality system lens distortion must be accounted for since there are three transformations that take place between the IR camera, RGB camera and projector. If lens distortions are not accounted for each sensor, the re-projection error from sensor will be accumulated resulting in poor correspondence.

To summarize the effect of lens distortion, consider two 2D image points $(u_D, v_D)^T$ and $(u_U, v_U)^T$ that represent the same pixel element. The first point, $(u_D, v_D)^T$, is the original distorted image point and the second $(u_U, v_U)^T$, is the corrected undistorted point. Then the mathematical relation between these two points can be represented by Equation 31 as proposed by Brown.

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = R \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + t$$

$$u_D = x/z$$

$$v_D = y/z$$

$$x'' = u_D \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + 2p_1 u_D v_D + p_2 (r^2 + 2u_D^2)$$

$$y'' = v_D \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6} + p_1 (r^2 + 2v_D^2) + 2p_2 u_D v_D$$

$$r^2 = (u_D^2 + v_D^2)$$

$$u_U = f_x x'' + c_x$$

$$v_U = f_y y'' + c_y$$

Equation 31. Radial and tangential distortion parameters

The radial distortion coefficients are: $k_1, k_2, k_3, k_4, k_5, k_6$. The tangential distortion coefficients are: p_1, p_2 . For most calibration applications the last three radial distortions (k_4, k_5, k_6) are set to zero because they have little effect on the distortion model. This is true for Bouget's calibration toolbox and OpenCV's calibration function. The tangential distortion coefficients are usually estimated to be zero. A tangential distortion takes place when image acquisition lens is not perfectly parallel to the imaging plane.

A.2.2 EXTRINSIC PARAMETERS

The previous development focused on identifying the intrinsic parameters that define the internal camera parameters. In addition, there exist external camera parameters that define the position and orientation of the camera coordinate frame with respect to a world coordinate frame; these are known as the extrinsic parameters. To be specific, the position of the camera center is defined using, \mathbf{t} , a three dimensional translation vector. The orientation of the camera may be defined using a direction cosine matrix, R , which performs a rotation from camera frame to world frame. It is important to note that there are a total of six degrees of freedom for the extrinsic parameters, three for position and three for orientation.

To obtain a 2D image point, \mathbf{w}' , of 3D-world point, \mathbf{W} , the camera origin must first be translated to the world coordinate origin, then the camera coordinate frame must be rotated such that its axes are aligned with the world coordinate frame. These steps may be expressed using matrix notation as shown in Equation 32.

$$\lambda \mathbf{w}' = [K_{Int.} | \mathbf{0}_{3 \times 1}] \begin{bmatrix} R & -R\mathbf{t} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \mathbf{W} = [K_{Int.} | \mathbf{0}_{3 \times 1}] K_{Ext.} \mathbf{W}$$
$$K_{Ext.} = \begin{bmatrix} R & -R\mathbf{t} \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}$$

Equation 32. Projection mapping using both intrinsic and extrinsic camera parameters