

# DUAL CAMERA SYSTEM FOR SYNCHRONOUS IMAGING IN VISIBLE AND NON-VISIBLE SPECTRAL REGIONS

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER  
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY  
IN THE FACULTY OF SCIENCE AND ENGINEERING

2020

Yiran Chen  
School of Engineering  
Department of Electrical and Electronic Engineering

Word count: 42900

# Table of Contents

Table of Contents .....	3
List of Figures .....	10
List of Tables .....	16
List of Abbreviations .....	17
Abstract .....	19
Declaration .....	20
Copyright .....	21
Acknowledgements .....	22
Chapter 1 Introduction .....	23
1.1 Infrared and visible image fusion .....	24
1.1.1. Applications .....	24
1.1.2. Methods .....	25
1.1.3. Systems and products .....	26
1.2 Overview of the PhD project .....	27
1.3 Objectives of the research .....	28
1.4 Research outcomes .....	29
1.5 Contributions and novelty of the research .....	30

1.6	Dissertation overview .....	31
Chapter 2	Hardware support .....	33
2.1.	Thermographic camera Xenics' XTM-640 .....	34
2.1.1.	Camera operation .....	35
2.1.2.	Output data timing .....	36
2.1.3.	Build-in image processing technique .....	38
2.1.4.	UART communication .....	40
2.2.	Visible camera Terasic's TRDB-D5M .....	42
2.2.1.	Output data timing .....	42
2.2.2.	Pixel array structure .....	43
2.2.3.	I2C communication .....	44
2.3.	The platform Intel® DE1-SoC .....	46
2.3.1.	HPS .....	48
2.3.2.	FPGA .....	49
2.3.3.	FPGA bridges .....	50
2.4.	Hardware choices .....	50
2.5.	System hardware design .....	51
2.5.1.	System requirement .....	51
2.5.2.	Design of the box enclosure .....	52
2.5.3.	Design of the printed circuit board to condition XTM-640 infrared camera communication .....	53



2.6. Data flow design .....	54
2.7. Summary.....	56
Chapter 3 System design methodology .....	57
3.1. Design methodology.....	58
3.2. Gateware architecture.....	59
3.3. Support of OpenCL on the dual-camera system .....	61
3.3.1. Intel FPGA OpenCL Framework .....	62
3.3.2. Board support package design - FPGA.....	66
3.3.3. Board support package design – Software.....	68
3.3.4. Design partitioning for partial reconfiguration.....	68
3.4. Summary.....	71
Chapter 4 Gateware design .....	72
4.1. Avalon® interfaces.....	72
4.1.1. Avalon® Streaming Interface.....	73
4.1.2. Avalon® Memory Mapped Interface.....	74
4.2. Camera modules .....	75
4.2.1. VISIBLE_CAM and INFRARED_CAM .....	76
4.2.2. VIDEO_CSR .....	77
4.3. Summary.....	78
Chapter 5 Software design for the dual-camera system .....	79
5.1. TCP/IP Server-Client implementation.....	80

5.2. Server running on ARM processor .....	81
5.2.1. Server design .....	81
5.2.2. Integration OpenCL host program with the server .....	84
5.3. Client running on PC.....	85
5.3.1. Python client for fast data acquisition .....	86
5.3.2. C++ client for image processing algorithm prototype and testing.....	87
5.3.3. Custom GUI client .....	88
5.3.4. Result of frame rate.....	89
5.4. Summary.....	90
Chapter 6 Geometric camera calibration .....	92
6.1. Geometric camera calibration overview .....	93
6.2. Camera model and lens.....	94
6.2.1. Pinhole camera model .....	94
6.2.2. Lens distortion.....	97
6.3. Single-camera calibration.....	99
6.3.1. Calibration board .....	101
6.3.2. Image points and parameters .....	104
6.3.3. Single calibration procedure .....	105
6.4. Camera parameter calculation .....	108
6.4.1. Intrinsic and extrinsic parameters .....	108
6.4.2. Distortion parameter calculation.....	111

6.4.3. Calibration result and analysis.....	113
6.5. Evaluating the accuracy of geometric calibration.....	116
6.5.1. Reprojection error .....	118
6.5.2. Manufacturing error .....	119
6.5.3. Comparing calibration result with camera datasheet.....	120
6.6. Rectify lens distortion.....	122
6.7. Experimental result.....	125
6.8. Summary.....	127
Chapter 7 Infrared and visible image matching methods.....	129
7.1. Stereo system geometry .....	130
7.1.1. Dual-camera stereo system overview.....	130
7.1.2. Epipolar geometry .....	134
7.1.3. The matrices for stereo calibration .....	136
7.2. Stereo calibration.....	140
7.2.1. Stereo calibration procedure.....	141
7.2.2. Rotation and translation parameter calculation .....	143
7.2.3. Stereo calibration result and accuracy evaluation.....	144
7.2.4. Stereo rectification.....	147
7.2.5. Stereo rectification result .....	151
7.2.6. Stereo image matching and result .....	153
7.3. Image registration.....	158

7.3.1. Cross-correlation based image registration.....	159
7.3.2. Registration result and analysis .....	168
7.4. Comparison of two image matching methods .....	170
7.4.1. Limitations and advantages.....	171
7.4.2. Image processing time and comparison.....	173
7.5. Summary.....	175
Chapter 8 Radiometric calibration.....	176
8.1. Uncooled infrared camera introduction .....	177
8.1.1. Comparison between cooled and uncooled thermal system .....	177
8.1.2. Related features of the uncooled infrared module XTM-640 .....	179
8.1.3. Infrared camera XTM-640 UART configuration .....	180
8.1.4. System challenges in the absence of infrared detector temperature stabilization .....	181
8.2. Radiometric calibration approach.....	183
8.2.1. Calibration system configuration .....	185
8.2.2. Data acquisition .....	187
8.2.3. Temperature model .....	189
8.3. Evaluation.....	192
8.3.1. Analysis of residuals .....	193
8.3.2. Evaluation and verification experiment.....	195
8.3.3. Limitations .....	196

8.4. Temperature mapping and data visualization .....	197
8.4.1. Infrared image contrast enhancement .....	198
8.4.2. Pseudo-colour mapping for data visualisation.....	200
8.4.3. Comparison between colour palettes .....	203
8.4.4. Infrared and visible image combination .....	205
8.4.5. Temperature thresholding .....	206
8.5. Summary.....	207
Chapter 9 Conclusion and future work .....	209
9.1. Conclusion.....	209
9.1.1. The journey of designing and implementing the system .....	210
9.1.2. A conclusive view .....	212
9.2. Future work.....	213
9.2.1. Frame rate improvement .....	213
9.2.2. Image matching improvement .....	214
9.2.3. Usability improvement .....	215
Appendix A TRDB-D5M camera binning mode.....	216
Appendix B TRDB-D5M visible camera specifications .....	217
Appendix C XTM-640 thermal camera specification .....	218
References.....	219

# List of Figures

Figure 2-1 Block diagram of microbolometer sensor on XTM-640 camera module .....	35
Figure 2-2 Diagram of an individual microbolometer detector and working principle...	35
Figure 2-3 The functional diagram of the thermographic camera XTM-640 [27] .....	36
Figure 2-4 XTM-640 digital video timing diagram [27] .....	37
Figure 2-5 XTM-640 image processing chain .....	38
Figure 2-6 XTM-640 histogram control-related terms [27].....	38
Figure 2-7 Infrared camera raw image (left) and its histogram (right).....	39
Figure 2-8 Infrared camera raw image after histogram control (left) and its histogram (right) .....	40
Figure 2-9 UART interface.....	41
Figure 2-10 XTM-640 UART communication text message example .....	41
Figure 2-11 TRDB-D5M digital video timing diagram [37] .....	43
Figure 2-12 TRDB-D5M pixel array architecture [37] .....	44
Figure 2-13 TRDB-D5M I2C communication timing diagram.....	45
Figure 2-14 DE1-SoC development board top view [40] .....	46
Figure 2-15 Block diagram of the DE1-SoC system overview [41] .....	47
Figure 2-16 DE1-SoC AXI bridge block diagram [43] .....	50
Figure 2-17 The enclosure of the dual-camera embedded system .....	53
Figure 2-18 XTM-640 interface board block diagram .....	54
Figure 2-19 Data flow on the dual-camera embedded system.....	55
Figure 3-1 Design flow of the dual-camera embedded system.....	58

Figure 3-2 Main blocks of the gateway system designed on DE1-SoC.....	59
Figure 3-3 Platform Designer modules that form the readout process .....	60
Figure 3-4 Overview of Intel® FPGA SDK for OpenCL programming model [51] .....	63
Figure 3-5 DE1-SoC OpenCL execution model.....	64
Figure 3-6 NDRange example .....	65
Figure 3-7 DE1-SoC OpenCL memory model.....	65
Figure 3-8 BSP overview on SoC FPGA.....	66
Figure 3-9 Kernel interface.....	67
Figure 3-10 A complete view of dual-camera embedded system Platform Designer modules.....	68
Figure 3-11 Chip planner view (left) and routing utilisation view of LogicLock region (right) .....	69
Figure 4-1 Waveforms of Avalon-ST packet transfer with <b>readyLatency=1</b> and backpressure .....	74
Figure 4-2 Waveforms of Avalon-MM slave read (left) and Avalon-MM slave write (right) .....	75
Figure 4-3 Camera data acquisition and control submodule.....	76
Figure 4-4 Stage diagram of camera readout modules .....	77
Figure 5-1 Workflow of server-client communication .....	80
Figure 5-2 Flow diagram of the top-level server application.....	82
Figure 5-3 Server state when receiving a new command.....	83
Figure 5-4 Block diagram of the server with OpenCL host .....	84
Figure 5-5 Flow of an OpenCL host execution .....	85
Figure 5-6 Flow diagram of the Python client .....	87
Figure 5-7 Qt GUI client start window for setting IP and port.....	88

Figure 5-8 Qt GUI device setting panel .....	88
Figure 5-9 Qt GUI stream panel showing an infrared image (up) and a visible image (down) .....	89
Figure 5-10 Overall schematic of the hardware and associated software .....	91
Figure 6-1 Pinhole camera model.....	95
Figure 6-2 Pinhole camera parameters and their relationship .....	95
Figure 6-3 Radial lens distortion shown on an example of a square grid.....	97
Figure 6-4 The cause of tangential lens distortion (left) and the resulting image shown on an example of a square grid (right).....	99
Figure 6-5 Infrared image (left) and infrared image with reference line (right).....	100
Figure 6-6 Visible image (left) and visible image with reference line (right) .....	100
Figure 6-7 Infrared image of a repeating dot pattern .....	101
Figure 6-8 A visible image (left) and an infrared image (right) with a chessboard pattern .....	102
Figure 6-9 Chessboard pattern for the infrared camera: front view (left) and back view (right) .....	103
Figure 6-10 A visible image (left) and an infrared image (right) with a circle grid pattern .....	103
Figure 6-11 10 infrared images of the calibration patterns .....	106
Figure 6-12 10 visible images of the calibration patterns.....	106
Figure 6-13 Detected image points.....	107
Figure 6-14 Pinhole camera model with a frontal image plane.....	111
Figure 6-15 Reprojection error on this image is the distance between $a$ and $a'$ .....	113
Figure 6-16 Visualized distortion model.....	116
Figure 6-17 10 calibration planes and its relative position of the cameras .....	117



Figure 6-18 Single camera calibration reprojection error with an example of 10 images .....	118
Figure 6-19 Undistorted image comparing to the original image.....	126
Figure 7-1 Heterogeneous dual-camera system front view (left) and side view (right)	131
Figure 7-2 Stereo vision system operation.....	132
Figure 7-3 Disparity in stereo vision system .....	132
Figure 7-4 Ideal disparity triangulation from two perfectly aligned cameras.....	133
Figure 7-5 The relationship between disparity and depth.....	134
Figure 7-6 Practical situation of a stereo vision system (left) and epipolar geometry (right) .....	135
Figure 7-7 Practical disparity triangulation from stereo cameras.....	138
Figure 7-8 10 examples of stereo image pair of a circle grid pattern .....	142
Figure 7-9 5 examples of pose estimation of the infrared (up) and visible (down) calibration pattern.....	143
Figure 7-10 10 calibration image pairs and its relative position of the stereo cameras	146
Figure 7-11 Stereo camera calibration reprojection error with an example of 10 images .....	146
Figure 7-12 Relationship between a point on the original image plane and on the rectified image plane.....	149
Figure 7-13 Procedure of stereo image pair rectification.....	152
Figure 7-14 Stereo rectification results in comparison to their original image.....	152
Figure 7-15 Stereo image pair set 1 (left) and set 2 (right) overlay result.....	153
Figure 7-16 Stereo image pair set 1 and set 2 overlay result after translation.....	155
Figure 7-17 Images taken from a variety of distance and the result after stereo rectify .....	156

Figure 7-18 Infrared image and visible image overlay and mosaic overlay result .....	157
Figure 7-19 Flow chart of FFT-based image registration algorithm .....	163
Figure 7-20 A pair of undistorted original images and grey scaled visible images.....	164
Figure 7-21 Fourier log-magnitude spectrum in cartesian coordinates .....	164
Figure 7-22 The high-pass emphasis filter and high-pass filtered Fourier log-magnitude spectrum.....	166
Figure 7-23 High-pass emphasised image pair.....	166
Figure 7-24 Fourier log-magnitude spectrum in polar coordinates .....	167
Figure 7-25 An example of the cross-power spectrum of a calibration pattern image pair .....	168
Figure 7-26 Calibration pattern image registration result .....	169
Figure 7-27 An example of an arbitrary scene image pair cross-correlation result .....	170
Figure 8-1 Infrared camera detector temperature plotted in the function of time.....	181
Figure 8-2 Infrared image grey level as the infrared camera heats up .....	182
Figure 8-3 Setup of the temperature target and the camera for calibration.....	186
Figure 8-4 Data capture procedure for infrared camera XTM-640 radiometric calibration .....	188
Figure 8-5 Radiometric calibration sample curves .....	191
Figure 8-6 Temperature model 3-D mesh plot .....	192
Figure 8-7 Kettle surface temperature measurement from XTM-640 and FLIR TG167 .....	196
Figure 8-8 Histogram equalization applied to low contrast infrared image.....	199
Figure 8-9 Colour mappings to enhance the greyscale infrared image.....	201
Figure 8-10 CIELUV colour space and sampling points .....	202
Figure 8-11 Analysis of the novel colourmap .....	203

Figure 8-12 Colourmap comparison of different temperature ranges .....	204
Figure 8-13 Combined views of the visible image and colour mapped infrared image.	205
Figure 8-14 Steps of creating a combined view of thresholding overlaid image .....	207
Figure 8-15 Threshold view of the visible image and colour mapped infrared image ..	207
Figure 9-1 Flow chart of designing and implementing the dual-camera system .....	210
Figure A - 1 TRDB-D5M visible camera binning mode [37] .....	216

# List of Tables

Table 2-1 XTM-640 UART communication text message format .....	41
Table 3-1 Statistic of resource usage for the fixed partition .....	70
Table 4-1 Register map for VIDEO_CSR module.....	78
Table 6-1 Calibration result of the intrinsic parameters .....	114
Table 6-2 Calibration result of the distortion coefficients ( $\times 10^{-2}$ ) .....	114
Table 6-3 Overall mean reprojection error of single-camera calibration .....	119
Table 6-4 Specification of cameras .....	120
Table 6-5 Camera parameter comparison between datasheet and calibration result ...	121
Table 6-6 Frame rate of producing an undistorted image in different platforms .....	127
Table 7-1 Frame rate of producing an overlaid image in different platforms.....	173
Table 7-2 Comparison result .....	174
Table 8-1 Main statistic indices of the linear and the non-linear polynomial model ...	194
Table B - 1 TRDB-D5M module specification.....	217
Table C - 1 XTM-640 module specifications.....	218

# List of Abbreviations

<b>ALM</b>	Adaptive Logic Module
<b>ALUT</b>	Adaptive Look-Up Tables
<b>AOCL</b>	Intel FPGA OpenCL
<b>API</b>	Application Programming Interfaces
<b>ASCII</b>	American Standard Code for Information Interchange
<b>ASCII</b>	American Standard Code for Information Interchange
<b>a-Si</b>	amorphous Silicon
<b>Avalon-MM</b>	Avalon® Memory-Mapped
<b>Avalon-ST</b>	Avalon® Streaming
<b>BPR</b>	Bad Pixel Replacement
<b>BSP</b>	Board Support Package
<b>CDF</b>	Cumulative Distribution Function
<b>CIELAB</b>	International Commission on Illumination L*(lightness from black to white)a*(from green to red)b*(from blue to yellow)
<b>CMOS</b>	Complementary Metal–Oxide–Semiconductor
<b>CPU</b>	Central Processing Units
<b>CRA</b>	Control Register Access
<b>CSR</b>	Control and Status Register
<b>DSP</b>	Digital Signal Processor
<b>DVAL</b>	Data Valid
<b>FFT</b>	Fast Fourier Transform
<b>FFT</b>	Fast Fourier Transform
<b>FPGA</b>	Field-Programmable Gate Array
<b>FPN</b>	Fixed Pattern Noise
<b>FPS</b>	Frame Per Second
<b>FPS</b>	Frame Per Second
<b>FVAL</b>	Frame Valid
<b>GPU</b>	Graphics Processing Unit
<b>HDL</b>	Hardware Description Languages
<b>HITL</b>	human-in-the-loop
<b>HPS</b>	Hard Processor System
<b>I2C</b>	Inter-Integrated Circuit
<b>IRQ</b>	Interrupt Request

<b>JTAG</b>	Joint Test Action Group
<b>LAB</b>	Logic Array Block
<b>LCD</b>	Liquid-Crystal Display
<b>LED</b>	Light-Emitting Diodes
<b>LSB</b>	Least Significant Bit
<b>LVAL</b>	Line Valid
<b>LVDS</b>	Low Voltage Differential Signalling
<b>LWIR</b>	Long-Wavelength Infrared
<b>MM</b>	Memory Mapped
<b>OS</b>	Operating System
<b>PCIe</b>	Peripheral Component Interconnect express
<b>PHY</b>	Physical Layer
<b>PLL</b>	Phase-Locked Loop
<b>PPM</b>	Perspective Projection Matrix
<b>PRNU</b>	Photo-Response Non-Uniformity
<b>RMS</b>	Root Mean Square
<b>Rx</b>	Receive wire
<b>SCLK</b>	serial clock
<b>SD</b>	Secure Digital
<b>SDATA</b>	serial data
<b>SDK</b>	Software Development Kit
<b>SDK</b>	Software Development Kit
<b>SDRAM</b>	Synchronous Dynamic Random-Access Memory
<b>SSRE</b>	Sum of Squared Reprojection Error
<b>SURF</b>	Speeded-Up Robust Features
<b>SWIR</b>	Short-Wavelength Infrared
<b>TCP/IP</b>	Transmission Control Protocol/ Internet Protocol
<b>TCR</b>	Temperature Coefficient of Resistance
<b>TEC</b>	Thermo-Electric Cooler
<b>TSE</b>	Triple-Speed Ethernet
<b>TTM</b>	Time-To-Market
<b>TV</b>	Television
<b>Tx</b>	Transmit wire
<b>UART</b>	Universal Asynchronous Receiver/Transmitter
<b>U-boot</b>	Universal Boot Loader
<b>XML</b>	Extensible Mark-up Language
<b>XTM</b>	Xenics Thermal Module

# Abstract

A dual-camera system comprised of a visible camera and a long wavelength infrared camera system is presented. This solution for synchronous imaging in visible and non-visible spectral regions has important applications in localising and identifying objects with different thermal signatures.

A detailed description of the design process and implementation of the dual-camera system is presented, highlighting all aspects of the image sensors used, their control and interfacing requirement and the necessary image processing to combine the acquired images into a single multi-spectrum image. The developed system consists of an acrylic box containing both sensors, an electronic interface board for conditioning and readout of the infrared sensor, and an image processing system that also handles network connection and communication with client programs by acting as a server. A field programmable gate array (FPGA) device is used to implement the readout protocols for the two sensor modules. Apart from the readout function, the FPGA is also a hardware accelerator which can be combined with a host central processing unit (CPU) to form an Open Computing Language (OpenCL) framework for a heterogeneous platform to perform complex image processing functions.

As part of the research, geometric calibration of this multi-sensor system is conducted to achieve the intrinsic and extrinsic parameters of the cameras, for the purpose of remove lens distortion effect. Stereo camera calibration and stereo image rectification are performed to obtain the relative position and orientation of the sensors. A method is proposed to match the two views from separate cameras using triangulation such that a combinational image can be formed, which consist of information from the visible and infrared band. With the acceleration of OpenCL framework, this system can give a continuous stream of image pairs for 10.15 images/s.

In the work of radiometric calibration of the uncooled infrared camera, a method which creates a temperature-controlled environment has been used, and this method does not require expensive blackbody equipment. From the image data analysis, a temperature model has been built to represent the relationship between the target surface temperature and the pixel value on the infrared image. A novel colour map is employed on the infrared image that can achieve better performance in the imaging modality than the conventional colour maps.

# Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.



# Copyright

i. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the “Copyright”) and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made **only** in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii. The ownership of certain Copyright, patents, designs, trademarks and other intellectual property (the “Intellectual Property”) and any reproductions of copyright works in the thesis, for example graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see <http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=487>), in any relevant Thesis restriction declarations deposited in the University Library, The University Library’s regulations (see <http://www.manchester.ac.uk/library/aboutus/regulations>) and in The University’s policy on presentation of thesis.

# Acknowledgements

There are many people whom I am eternally grateful for the possibility to start and accomplish this work and their support throughout. Firstly, I would like to express my thanks to my patient and supportive supervisor, Dr Roelof van Silfhout, whose insight and knowledge into the subject matter steered me through this research. I have been fortunate to have a supervisor who has combined a highly professional attitude with incredible tolerance and patience.

Completing this work would have been all the more difficult were it not for the support and friendship provided by the other members of the group. I give my innumerable thanks to all my colleagues and friends who gave me a hand and incited me to strive for my goal: Chao, Bart, Rand, Zeinab, and my cousin Zaiyi.

Finally, I must express my special and warmest gratitude to my parents for their continued support and encouragement through my whole journey of seeking the degree of PhD, and my boyfriend (husband - after 24<sup>th</sup> May 2021) Yingzhao for his support and standing with me all the time.

# Chapter 1

## Introduction

Cameras play an essential role in various modern imaging and photography systems, including industrial monitoring, security surveillance, filming special effects and scientific researches. Visible camera sensors are a kind of imager that collect visible light and convert it to electrical signals; then, these signals are organised for image rendering and video streaming. Visible sensors utilise light with a wavelength from  $400\sim700nm$ , which is the same spectrum range that the human eye perceives. The visible sensors require light to work, whereas an infrared camera does not need particular illumination to operate. An infrared camera can detect infrared energy emitted, transmitted, or reflected by an object's surface, whose temperature should go above absolute zero. The extension from the red edge of the visible spectrum at  $700nm\sim1mm$  is defined as infrared radiation. The infrared sensor converts the received infrared energy into temperature readings or visual images [1].

By associating the visible and infrared cameras, images from different parts of the frequency spectrum are combined as well as different information. A new image can be produced from the combination of the infrared and visible image to integrate useful information since the new image has higher reliability and understandability for human or computer vision applications.

## **1.1 Infrared and visible image fusion**

The system combines an infrared camera and a visible camera usually produces fusion images that contain information from both sources and have many complementary characteristics. For example, the fused image makes the infrared part more vivid through the textured background. Also, the temperature information can be marked on specific textures. In many applications, the matching algorithm is embedded so that the original display method can be improved. Majority area of using infrared and visible image fusion techniques including recognition [2] [3] [4], surveillance [5] [6], and object detection and tracking [7] [8] [9].

### **1.1.1. Applications**

One of the major applications in recognition that utilises infrared and visible image is face recognition. [2] solves the problem of recognising human faces at a long distance using the combination images, where it is often necessary to identify a person from a distance rather than at the close range. The human face's visible image quality is enhanced using the infrared image, and this process makes the recognition precision improved. Moreover, [10] proposed an algorithm for face recognition to exploit the relatively lower precision of visible imagery to occlusions caused by eyeglasses.

Detection technologies based on the dual-camera system are widely used in real-life

applications, such as fruit detection [11] and pedestrian detection [12]. Detection often determines the position of a target at the same time when recognising it. [13] is an algorithm proposed by He *et al.* that adopts the infrared and visible fused image for the enhancement of target visibility. The combination of two image sources exhibit superiority in display the detailed scene and detection precision. Target tracking using infrared and visible image combination share some similarities with object detection. The relationship between frames and the target location in a time sequence is determined based on the fused image. The tracking algorithm is implemented by one of the dual-camera target tracking systems established by [14]. Another example of the tracking system was built and used in Unmanned Aerial Vehicles (UAV) [15]. The unique merit of using an infrared camera in this kind of system makes object segmentation more effective.

### 1.1.2. Methods

Conventional infrared and visible image registration, which can be formulated by matching two sets of feature points extracted from the infrared and visible image pair, the work in [16] proposed a method a feature guided Gaussian mixture model to act as the feature descriptor to improve the precision of registration result. However, some mismatched points are eliminated, and that leads to unsatisfactory image matching successfulness and longer calculation time. [17] designed an artful method based on constrained point features to register the infrared and visible images. An object detection algorithm is developed to extract the constrained point features, then, a left value rule is proposed to match these points strictly without deleting the mismatched points, transformation matrix is finally calculated to match the image pair.

In recent years, apart from traditional image registration methods, many methods

adopt machine learning to tackle the infrared and visible image matching challenges. [18] is a approach to detect Mildew disease in vine field using the deep learning segmentation method on infrared and visible image pairs. Pixels are classified according to different instances using a fully convolutional neural network approach, and then register using these pixels. A Dual-Y network based semi-supervised transfer learning method was proposed by Mao *et al.* match the infrared and visible image patches. They solve the problem that that the infrared-visible image usually has small number of supervised samples. Visible images are seen as source domain and infrared images are seen as the target domain used by adversarial domain adaptation to solve different imaging principles. The two convolutional autoencoders in their network can register the image pair patches through a large number of unannotated samples training.

### 1.1.3. Systems and products

For surveillance applications, the video source needs to be clear, robust, and expressive. The properties of infrared and visible image combinations make them a suitable source for this task. Since the thermal camera can capture the infrared radiation from the target surface that is unaffected by the change of weather, illumination or other disturbance, this type of dual-camera surveillance system can acquire information in various aspect given their different imagery characteristics. Furthermore, the visible camera can provide the system with high spatial resolution images, where the texture details are distinguishable. The company FLIR has developed a commercial vehicle camera system composed of an infrared camera and a colour Television (TV) camera named ThermoVision Dual Sensor [19]. Since the visible camera has fields of view matched to the infrared camera, the visible picture can be blended with the infrared images to give situational awareness. The field of view can be switched between the two cameras and

the fused version for increased situational awareness.

The Fluke Corporation is another famous manufacturer of industrial thermal camera equipment. Its popular product including Compact PTi120 Pocket Thermal Camera, Fluke TiS20+ Thermal Imaging Camera, Fluke Ti401 PRO Thermal Camera and Fluke TiX580 Infrared Camera. These devices are typically applied for trouble shooting industrial appliance problems as well as condition monitoring, especially for motors and drives. The infrared camera can show temperature of an element, and the visible image fusion technique can help identify where the critical components are.

## **1.2 Overview of the PhD project**

This PhD project aims to solve the core problem of designing a real-time dual-camera system that can process the data output from the infrared camera and the visible camera and better preserve the target region and its corresponding detail information on different spectra. This project is categorised according to four open areas of research that have been identified relating to the process of establishing a dual-camera embedded system to synchronise the visible and infrared frames:

- Image data acquisition and processing
- Geometric camera calibration
- Infrared and visible image matching
- Infrared camera radiometric calibration and infrared image temperature modelling

The research has been divided into these four main phases, with a continuously operating integration phase. A matched view of the infrared and visible images can generate models that are superior to a single-camera system for the purpose of qualitative evaluation and quantitative thermal analysis. This information is vital because it is an advantage to

achieve the reliable geometric knowledge through the dual-camera system while utilising the temperature data from the infrared image, and the coherent incorporation of representing this information in a single image.

## 1.3 Objectives of the research

This project aims to design and implement a dual-camera embedded system on a processor board Intel® DE1-SoC to synchronise image captured by a visible sensor - Terasic's TRDB-D5M, and an infrared sensor - Xenics' XTM-640 with surface infrared radiation information.

The research questions can be classified into six main sections as follow:

- Retrieve pixel data from camera sensors using FPGA
- Present real-time image data on desktop computer for users
- Remove lens effect of the visible camera and the infrared camera
- Match the visible image and the infrared image to generate an overlaid view
- Relate the infrared image pixel data to object surface temperature
- Provide the overlay images in an accelerated way

Most previous studies focus on a single-camera system, either a visible camera or an infrared camera. These systems' performance is affected by various factors, such as illumination change or high background temperatures. Several previous research pieces use two visible cameras to form a dual-camera system to composite high-resolution images or 3D reconstruction models.

A new dual-camera system that combines visible light and thermal cameras in real-time is proposed in this thesis. Image from different cameras and different spectrum are



fused to contain more information for a better human visual effect. The infrared images are sensitive to the object in the scene area with distinguishable thermal characteristics, which cannot be captured by a visible camera. However, the infrared images are often lack of rich texture information due to the images' dim background and sometimes low contrast. Visible cameras are able to capture target edges and details, providing a more perceptual image for human eyes.

## 1.4 Research outcomes

This section outlines the contribution to knowledge achieved by this research project:

- **Design and implementation of a dual-camera data acquisition system using microprocessor:** The electronics, firmware and software for the infrared camera, the visible camera and the processor board are specifically designed for this project. A detailed explanation of the design choices and the design is given.
- **Implement OpenCL accelerate features on image processing:** Having OpenCL enabled for the processor board enables the use of a low-cost embedded system as a hardware-accelerated computing device. The image processing time is reduced, and the frame rate is higher to produce a matched image.
- **Perform geometric calibration of the infrared camera and the visible camera:** The geometric characteristics of both cameras are achieved after geometric camera calibration. The lens distortion effect is removed using the parameter obtained. Some manufacturing error is found during the calibration session.
- **Experiment on infrared and visible image matching methods:** The infrared and visible cameras are mounted in a fixed position. The image pairs are collected in an automatic way. Stereo matching and image registration approach to overlay

the image pairs are proposed, implemented, and analysed. The stereo matching method is chosen to rectify the infrared and visible image pairs.

- **Perform radiometric calibration and create temperature model for the infrared camera:** The radiometric calibration of the infrared camera is carried out under a temperature-controlled environment, without using a blackbody source. The target surface temperature is modelled, and a novel pseudo colourisation scheme is proposed to achieve a visualised image with infrared and visible information in the same representation.

## 1.5 Contributions and novelty of the research

The interconnected modules to acquire data output from each camera sensors and store them in the FPGA memory is customised for the specific camera and processor board. It will aid future data streaming applications for similar embedded systems. The board support package for the processor board DE1-SoC, designed for this work, provides a pre-configured environment to accelerate the execution of image processing algorithms, especially to process two image sources. It enables the flexibility to choose the configuration for other applications.

The camera characteristic and parameters are obtained during the geometric camera calibration session. This research provides a novel way of matching the images from different spectrum in real-time. The design of the stereo camera system consists of an infrared camera and a visible camera giving a solution of synchronising two images that have a difference in their field of view, lens conditions, and detector features. Techniques and methods developed as part of this section can also have applications in areas such as medical imaging, machine vision and industrial monitoring.

Within the area of target surface temperature modelling, the radiometric calibration

approach is invented for this work to be budget-saving and controllable. The temperature model which relates the pixel value with surface temperature allows future project, needing to acquire surface temperature from the infrared image, to use this model directly for further applications. The proposed pseudo colourisation method helps the visualisation of the infrared and visible information.

## 1.6 Dissertation overview

The thesis is structured as follows:

Chapter 1 describes the objective of this PhD thesis and explains the contribution to knowledge. A brief introduction of the dual-camera system and some of its many applications is given.

Chapter 2 gives an introduction on thermographic camera Xenics' XTM-640, visible camera Terasic's TRDB-D5M. The camera operation, digital output format, electronic features and applications are included.

Chapter 3 includes an overview of the platform field-programmable gate array (FPGA) Terasic's DE1-SoC. The design in hardware and the interconnections of the FPGA to the thermal camera is explained, as well as the data flow design of the whole system is included.

Chapter 4 explains the dual-camera system design methodology. The platforms and environment to implement the system are described, and the supportive design to enable OpenCL environment is explained.

Chapter 5 illustrate the data transmission module design for the FPGA to communicate and control the two cameras. A description is given on data streaming protocols and the design of camera data acquisition and control modules.

Chapter 6 presents the software designed for this thesis work. A description of the

server program implemented on the processor board as well as the client software to control the streaming data is given.

Chapter 7 focus on geometrical camera calibration. It introduces the camera model and lens distortion. The camera parameters and the rectification of the lens distortion effect are obtained after the geometric calibration process.

Chapter 8 shows the process of figuring out a method to match the image pair captured by this multi-sensor system. The stereo camera system geometry is introduced as well as the practice of stereo calibration. Image registration is implemented in comparison with the stereo matching method.

Chapter 9 describe the process of radiometric calibration of the thermographic camera to obtain the temperature estimation of the target surface from the image data.

Chapter 10 look back to the journey of design this dual-camera system. It concludes the thesis and discusses some possible future work.

## Chapter 2

### Hardware support

In this chapter, the fundamentals of the infrared image sensor and visible image sensor are presented, as well as their data communication interfaces. The Xenics' XTM-640 infrared camera and Terasic's TRDB-D5M visible camera are introduced and their output data timing, feature and usage are presented. The serial communication method to configure the cameras are explained. These two cameras are the central points over the rest of the work of this thesis is based on. To design a fully functional system, a development board, Intel® DE1-SoC, connects two cameras and reads out images. The connections between the two cameras and the board need to be constructed for signal processing. This chapter also introduce the processor board DE1-SoC with its capability and configuration. Then, the electronics layout requirement is declared, followed by the design of an interface board with the infrared camera and the portable box designed to contain the whole system. The data flow design for the system is presented at the end.

## 2.1. Thermographic camera Xenics' XTM-640

A thermographic camera is also called an infrared camera and can be seen as a non-contact temperature measurement device. It is a sensor that creates an image from infrared energy emitted, transmitted, or reflected by extraneous material. The detection process for the XTM-640 series is based on an array type of uncooled microbolometer detector receiving infrared interaction with the material and the processing of the generated signal [20]. High-performance Long-Wavelength Infrared (LWIR) detection functionality in the spectral band ranges from  $8\mu\text{m}$  to  $14\mu\text{m}$ . This camera's resolution is  $640 \times 480$ , and the pixel pitch is  $17\mu\text{m}$  for longer detection, recognition, and identification. An LWIR sensor would be superior to the Short-Wavelength Infrared (SWIR) sensor, which detects  $1.4\mu\text{m} \sim 2.5\mu\text{m}$  wavelength infrared rays. For example, human activities cannot be detected by SWIR sensors without an external light source [21]. This thermographic sensor is used in this work as the size is compact, with its advantage in frame rate up to 60Hz, and the digital video signal can be easily acquired.

The use of cameras based on infrared detecting technology has expanded in recent decades. These sensors can be found in many applications, including security surveillance [22], firefighting [23], road safety, search, and rescue [24]. These examples all benefit from the cameras' feature of capturing people or animals beyond the reach of natural lights because thermal cameras do not require any external visible light to operate. In the field of medical care, these sensors can be applied to detect skin temperature with the benefit that they are non-invasive tools [25], and some thermal scanners can help detect the presence of deep vein thromboses and other circulatory disorders.

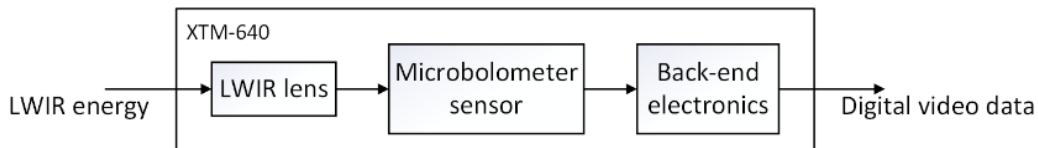


Figure 2-1 Block diagram of microbolometer sensor on XTM-640 camera module

As shown in Figure 2-1, the thermal camera XTM-640 uses resistors called microbolometer as a detector. It is a tiny amorphous Silicon (a-Si) resistor with a significant temperature coefficient deposited on a silicon element with a large surface area, suitable thermal isolation, and low heat capacity. The microbolometer camera module can be divided into several parts: optics, microbolometer sensor and back-end electronics. The lens is designed specifically for this camera module to image infrared radiation on the microbolometer sensor array; the sensor generates a signal to be processed by the back-end electronics to achieve digital video data.

### 2.1.1. Camera operation

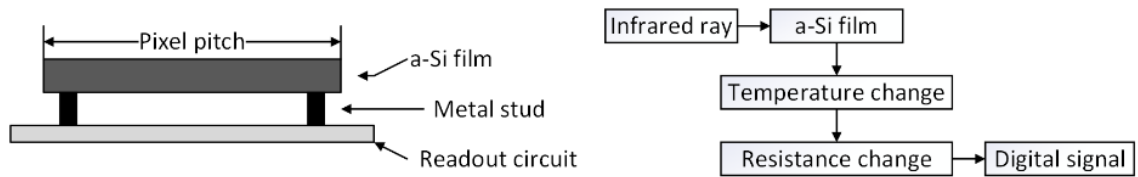


Figure 2-2 Diagram of an individual microbolometer detector and working principle

More detailed information about the microbolometer sensor is present in Figure 2-2. When infrared radiation from the object is incident on the amorphous-Silicon (a-Si), its electrical resistance changes as the element increases in temperature. The resistance change is monitored by the readout circuit and processed into digital values represented graphically as an infrared image [26]. The detector bias voltage and detector gain are unique for this module and are pre-programmed into the non-volatile memory. In response to an internal or external trigger, a frame will be generated by the module. Each time when a calibration, load, save and user setting commands are given by the camera module itself or by the user, there will be a loss of 9 frames as they are used to make calibration images. From the 10<sup>th</sup> trigger pulse, the generated image will be sent to the

output interface [27]. The sets of image processing options provided by the camera module are described in section 2.1.3.

As the camera module is register-based, the control of this module is happening by changing the appropriate register. Users can alter the register contents through multiple ways, such as Universal Serial Bus (USB) or Ethernet. A serial communication protocol named Universal Asynchronous Receiver-Transmitter (UART) is chosen over the interface because it is more reliable on full-duplex communication and lower on the cost to exchange data between the FPGA and the infrared module peripherals. Section 2.1.4 will have more information on this protocol. The memory space in the module uses 32-bit addressing, each address is on a 4-byte boundary, and each address contains 32 bits. Every pre-defined register contains the start-up defaults or the contents of the last write operation stored in the module's non-volatile memory. The following figure shows the basic functional diagram of this infrared camera module.

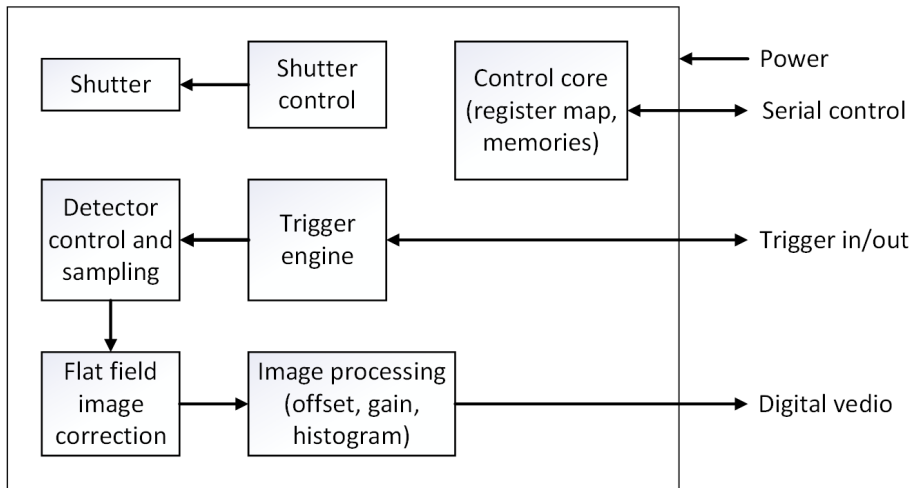


Figure 2-3 The functional diagram of the thermographic camera XTM-640 [27]

### 2.1.2. Output data timing

Data is output from the module using the Camera Link standard [28]. It is an open specification of communication interface designed for vision applications, which uses Low



Voltage Differential Signalling (LVDS) technology for transmitting digital data. The XTM-640 uses three of its pixel transfer signal: Frame Valid (FVAL), Line Valid (LVAL) and Data Valid (DVAL):

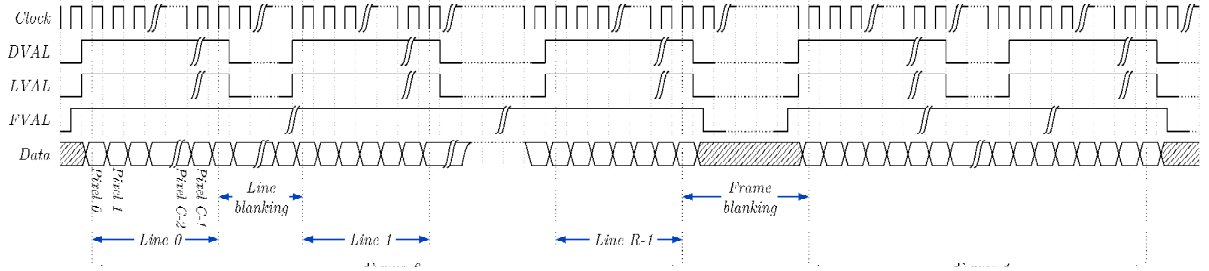


Figure 2-4 XTM-640 digital video timing diagram [27]

- **FVAL:** The FVAL signal is high whenever a frame is sent out on the Camera Link. The frame valid signal active four clock cycles before DVAL and will remain active until four clock cycles after DVAL.
- **LVAL:** This signal is high whenever a line is being transmitted. The number of LVAL pulses within a frame is 480, representing the height of the applied field of interest. LVAL is always low when the FVAL signal is low.
- **DVAL:** The DVAL signal is high whenever valid pixel data is sent out. The number of DVAL signal high cycles will always equal the number of pixels in a line, which is 640. The DVAL signal is always low when the LVAL signal is low. One thing that needs to be noticed is that the DVAL signal can be toggling as it is equivalent to the LVAL signal.

The digital video clock rate is 16 MHz, and the 16-bit digital video data signal shown in Figure 2-4 is valid only when FVAL, LVAL and DVAL are all high, and it will be sent as an output through the interface.

### 2.1.3. Build-in image processing technique

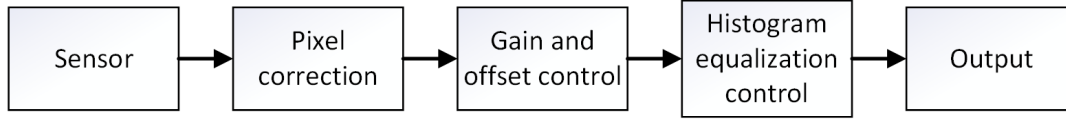


Figure 2-5 XTM-640 image processing chain

In Figure 2-5, the XTM-640 camera module includes a dedicated image processing chain that provides background removal, flat-field corrections, gain and offset control, and histogram equalisation control. Next to capturing and sending raw image data, the camera module can apply a flat field image correction. The back-end electronics will send a uniform image data signal to fix some image artefacts in the optical path, such as specs of dust on the lens surfaces. The following image processing options are in-build on the camera module [27]:

- **Non-uniformity corrections:** This is the first set of processing options related to image calibration correction. It is a combination of Fixed Pattern Noise (FPN) correction, Photo-Response Non-Uniformity (PRNU) correction and Bad Pixel Replacement (BPR) correction.

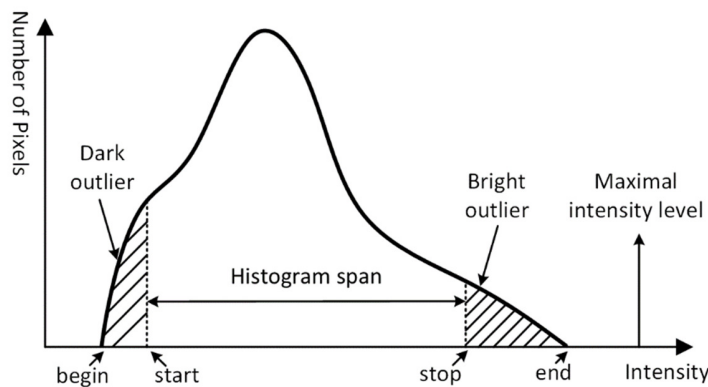


Figure 2-6 XTM-640 histogram control-related terms [27]

- **Histogram control:** Another set of processing option is global image processing

options, only one gain, offset, and equalisation strength parameters are set for the complete infrared image. It is a control algorithm that uses the accumulative histogram information in each frame to improve the contrast of the image. Figure 2-6 defines some of the important histogram related terms, assuming the curve in this figure representing the number of pixels in each density level. Those dark/bright outliers are pixels with very low/high-intensity levels and are expressed as a percentage of the total amount of pixels. These outlier pixels are removed in the actual image. The histogram's start and stop are the first and last histogram bin different from 0 after those outlier pixels are ignored. The distance between these two points is the histogram span.

The offset parameter manipulates the difference between the midpoints of two histograms, whereas the gain parameter controls the histogram span's width. The equalisation strength control is achieved by adjusting the histogram distribution more uniformly, with a higher strength value. There will be a significant improvement in image contrast, as shown in Figure 2-7. The raw image directly taken from the camera has many valuable data that are not obvious in the image.

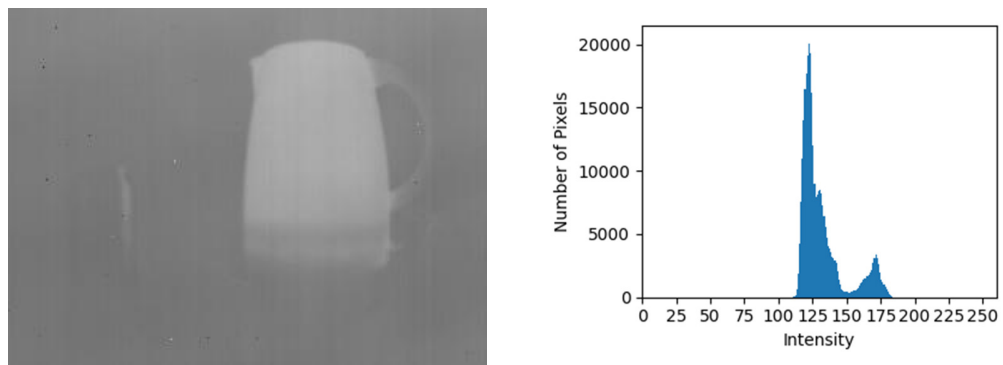


Figure 2-7 Infrared camera raw image (left) and its histogram (right)

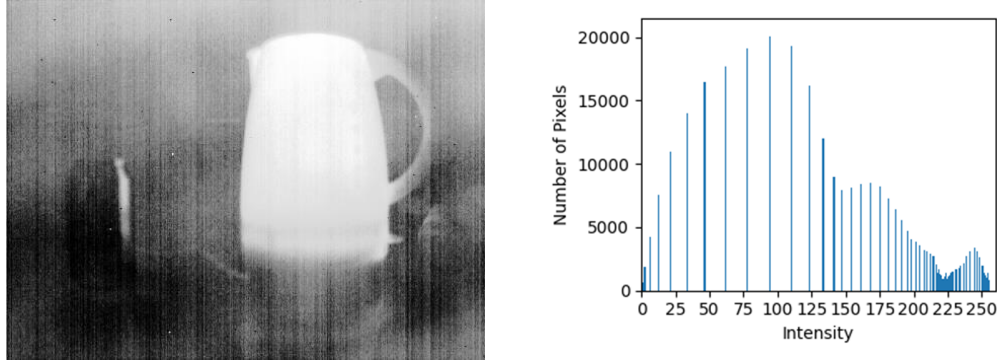


Figure 2-8 Infrared camera raw image after histogram control (left) and its histogram (right)

Figure 2-8 shows the output image after the histogram control stage but without the non-uniform control. Within the scene of interest in an image, even the smallest relative temperature differences can be visualised and recognised [29].

#### 2.1.4. UART communication

A Universal Asynchronous Receiver-Transmitter (UART) is a circuit in microcontrollers that is used for serial communication by transmitting and receiving data within the transmit wire (Tx) and the receive wire (Rx) [30]. This circuit provides robust, decent speed and full-duplex communication in an embedded system [31]. The communication scheme over this serial data link between the XTM-640 camera module and an external device is that of a Master and Slave configuration, as the interface shown in Figure 2-9. This camera module will initiate any communication but will only answer the command issued by an external device or detect errors when receiving a command, making it a slave. The speed supported by this UART is 115200 baud (bit per second), with one start bit, one stop bit, eight data bits, no parity, and no handshaking.

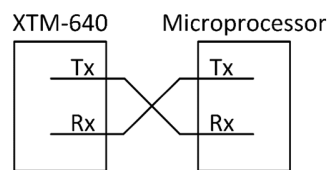


Figure 2-9 UART interface

In this work, the camera module is connected to the microprocessor, where a program can be designed to debug settings and change specific registers' value. The UART communication is also provided by Xenics software designed for this camera module; thus, this method can be easily verified. The program is based on text message communication. The sending and replying message are constructed as a sequence of American Standard Code for Information Interchange (ASCII) characters, and the definitions can be found in Table 2-1.

Table 2-1 XTM-640 UART communication text message format

Position	Contents	Characters	Remarks
0	#	1	Start of message
1	Register address	1-10	32-bit address in hexadecimal (0-9, a-f, A-F) preceded with 'h' or 'H.'
2	Operation	1	'=' assignment operation '?' inquire operation
3	Value	1-10	32-bit address in hexadecimal (0-9, a-f, A-F) preceded with 'h' or 'H.' Only when the operation is '='.
4	Carriage return(0x0d)	1	Pressing of the enter key

```

1  #h1980? ←Text message send
2  recieved msg: "#h1980?", length: 7
3  | sending 8 characters
4  recieved msg: "h00000001", length: 9 ←Text message receive
5  104 48 48 48 48 48 48 48 49 0 0 0 0 0 0 0 0 0 0 ←ASCII character receive

```

Figure 2-10 XTM-640 UART communication text message example

For example, when testing the UART communication with the XTM-640 module in Figure 2-10, if sending '#h1980?', The camera would respond and send the content in base address 1980 to the microcontroller, which is 'h00000001'. In the case of a positive

acknowledge, the reply would be ‘ACK’.

## **2.2. Visible camera Terasic’s TRDB-D5M**

The camera module TRDB-D5M is developed by Terasic equipped with a Micron 5-megapixel Complementary Metal–Oxide–Semiconductor (CMOS) sensor [32]. A CMOS sensor is an image sensor construct by an integrated circuit with an array of pixel sensors [33]. Each CMOS pixel sensor is consisting of a light sensor and an active amplifier. The light sensor receives a post-processed signal from the colour filter when the light come through the lens, as the sensor is sensitive to the intensity level of the wavelength-filtered light. The sensor has an analogue-to-digital converter (ADC) to convert the analogue signal, then the voltage of the light signal is amplified so the further process could be done [34]. It is different from a Charge Coupled Device (CCD) sensor as every pixel’s charge in the CCD sensor is transferred through an output node to be converted to voltage, and the output is an analogue signal [35]. Benefiting from the CMOS camera’s small system, low power consumption and excellent image quality, this kind of modules are usually used in high-performance industrial cameras and are dominant in smartphone cameras [36].

### **2.2.1. Output data timing**

Camera Link standard is also used in this TRDB-D5M camera module. The output data timing diagram is shown in Figure 2-11. The acquisition of an image will need LVAL, FVAL and DVAL signal. According to the datasheet, this module's interface does not contain a DVAL signal, so the LVAL signal is paralleled to the DVAL signal output to match the standard.

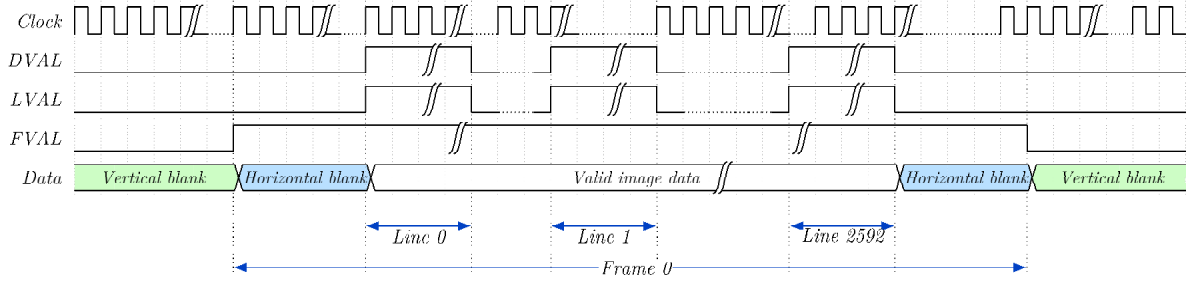


Figure 2-11 TRDB-D5M digital video timing diagram [37]

When the clock signal is at its rising edge, the sensor starts to transmit pixel data. When LVAL and FVAL signal are both high, the sensor begins to capture an image. During the time that the FVAL signal is high, there will be 1944 times a high level of LVAL signal. During each high level of the LVAL signal period, 2592 groups of 12-bit image data are being sent by the module. Furthermore, the output of TRDB-D5M supports a resolution of  $2592 \times 1944$  image with a frame rate of 15 frames per second (FPS).

### 2.2.2. Pixel array structure

The actual TRDB-D5M pixel array contains a 2,752-column by 2,004-row matrix of pixels addressed by column and row, but the active image region is only 2,592-column by 1,944-row in the centre, which represents the output image. As shown in Figure 2-12, the valid output image is surrounded by a boundary region and a border of dark pixels, which reflect in the timing diagram should be horizontal blanking and vertical blanking.

The first pixel in the default condition of valid image data starts from the top right corner (16, 54) as the camera module's mirror mode is default enabled. The output colour filter array that the module produced is called the Bayer pattern array with three colours: Red (R), Green (G1, G2) and Blue (B). Half of the filter elements are green because the human eye is more sensitive to green light than both red and blue light [38]. The first-

row output pixels alter between G1 and R, and the second-row output pixels alters between B and G2.

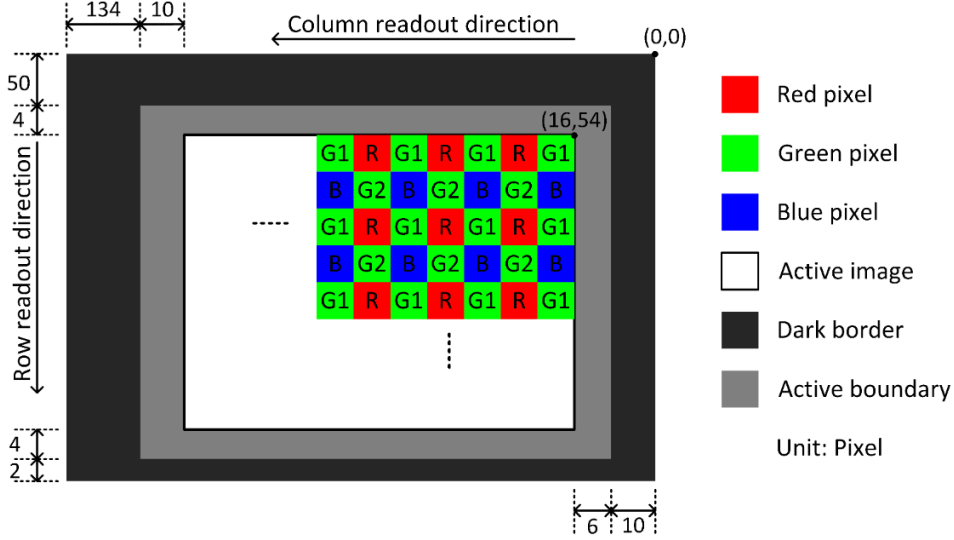


Figure 2-12 TRDB-D5M pixel array architecture [37]

The dark pixel outliers can be used to monitor the black level, while the information from the boundary region can be used to get rid of the edge effect when the module is doing colour processing to achieve the active image.

### 2.2.3. I2C communication

The TRDB-D5M camera module has registers written and read through a two-wire serial interface bus. The control protocol is named Inter-Integrated Circuit (I2C), widely used in communication between microcontrollers with external devices in an embedded system [39]. The same Master and Slave communication scheme is also applied in connecting TRDB-D5M and the microprocessor platform. The I2C control of the camera module can either be done by designing an I2C bus controller and a look-up table (LUT) in Verilog corresponding to the camera's internal register address or programming the microcontroller in C++ to open the I2C bus and write the register directly. The command



data is transmitted into and out of the camera module through the serial data (SDATA) line, and the whole serial interface is governed by the serial clock (SCLK) produced by the master interface on the microcontroller.

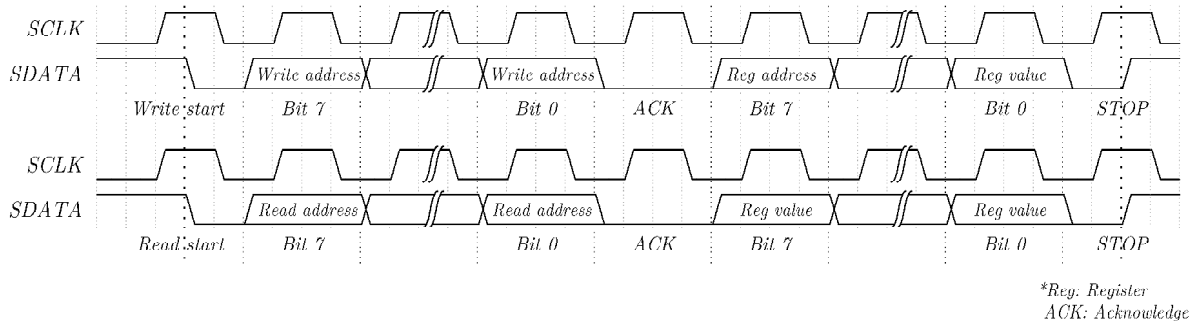


Figure 2-13 TRDB-D5M I2C communication timing diagram

The FPGA starts a read or write sequence by issuing a start bit, followed by the camera module's 8-bit address. One bit of command data is transferred in each clock pulse provided by the master. If the least significant bit (LSB) of the address is '0', then the command indicates a write, otherwise '1' indicates a read. The sensor answers to confirm its address by sending an acknowledge bit back to the master, and the acknowledge bit is indicated by the receiver pulling the data line to LOW during the acknowledge clock pulse. When a write is requested as following Figure 2-13, the master will then be sending the 8-bit register address to which a write should take place. After receiving the acknowledge bit from the slave, the master will issue 8-bit data, and the slave will send an acknowledge bit after every eight bits. The internal register address of TRDB-D5M is formed of 16-bit, so a set of two 8-bit transfer will be delivered to write one register. The register address will automatically be incremented after a total of 16 bits being transferred. Thus, the following 16-bit command will be written to the next register address. A start or stop bit is required by the slave to know the master has stopped writing. As shown in Figure 2-13, when a read is requested, the master sends the write-mode slave address and 8-bit register address in sequence. A start bit and the

read-mode slave address will be issued. The master then takes records of the register data eight bits at a time, followed by an acknowledge after each 8-bit data transfer.

## 2.3. The platform Intel® DE1-SoC

Intel® DE1-SoC Development Kit is selected for this project. It is a general-purpose heterogeneous development board with an Intel Cyclone® V FPGA and ARM Cortex-A9 dual-core processor. The embedded vision system project is designed and implemented on the DE1-SoC board, shown in Figure 2-14.

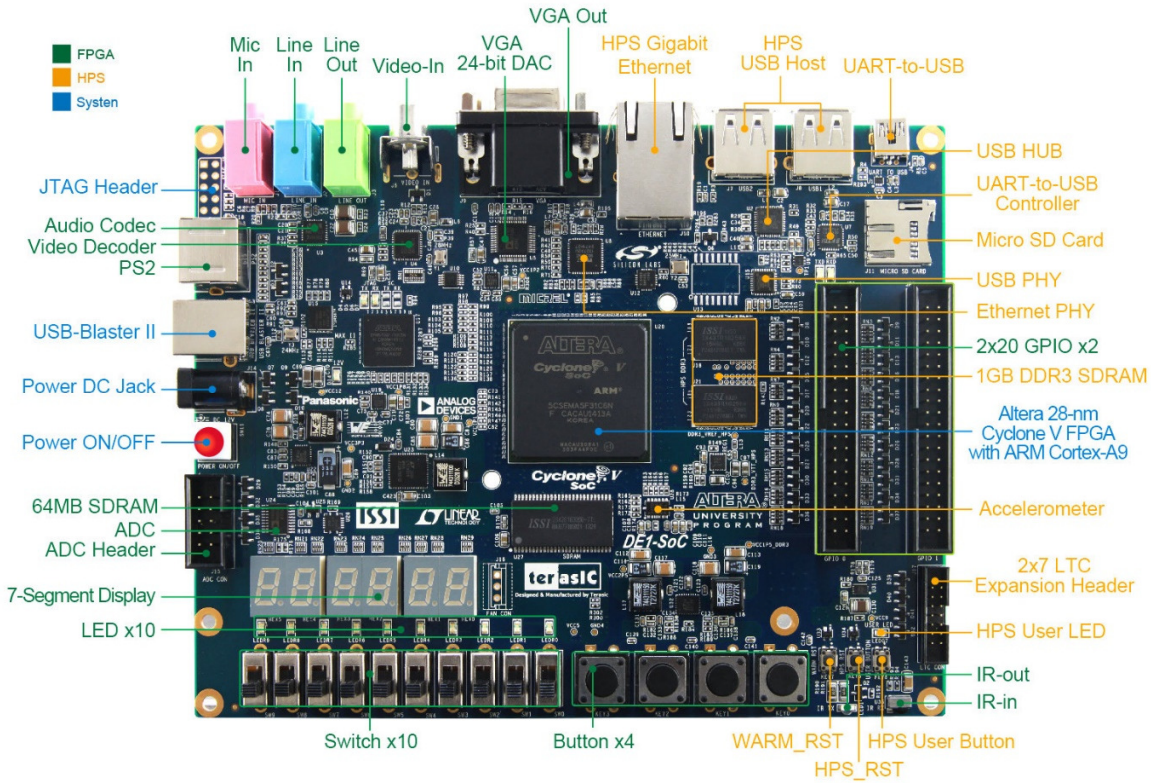


Figure 2-14 DE1-SoC development board top view [40]

Figure 2-15 shows the components in this embedded system are taking advantage of the Hard Processor System (HPS) and FPGA inside the Cyclone® V System on a Chip (SoC). The HPS part contains several peripheral devices, a Double Data Rate3 (DDR3)

synchronous dynamic random-access memory (SDRAM) port, and most importantly, the ARM A9 processor. The side of FPGA comprises two Intel Nios® II processors and a set of peripheral ports: memory, timer modules, audio and video input/output, PS/2, ADC, an infrared interface, and parallel ports connected to Light-emitting diodes (LEDs) and switches.

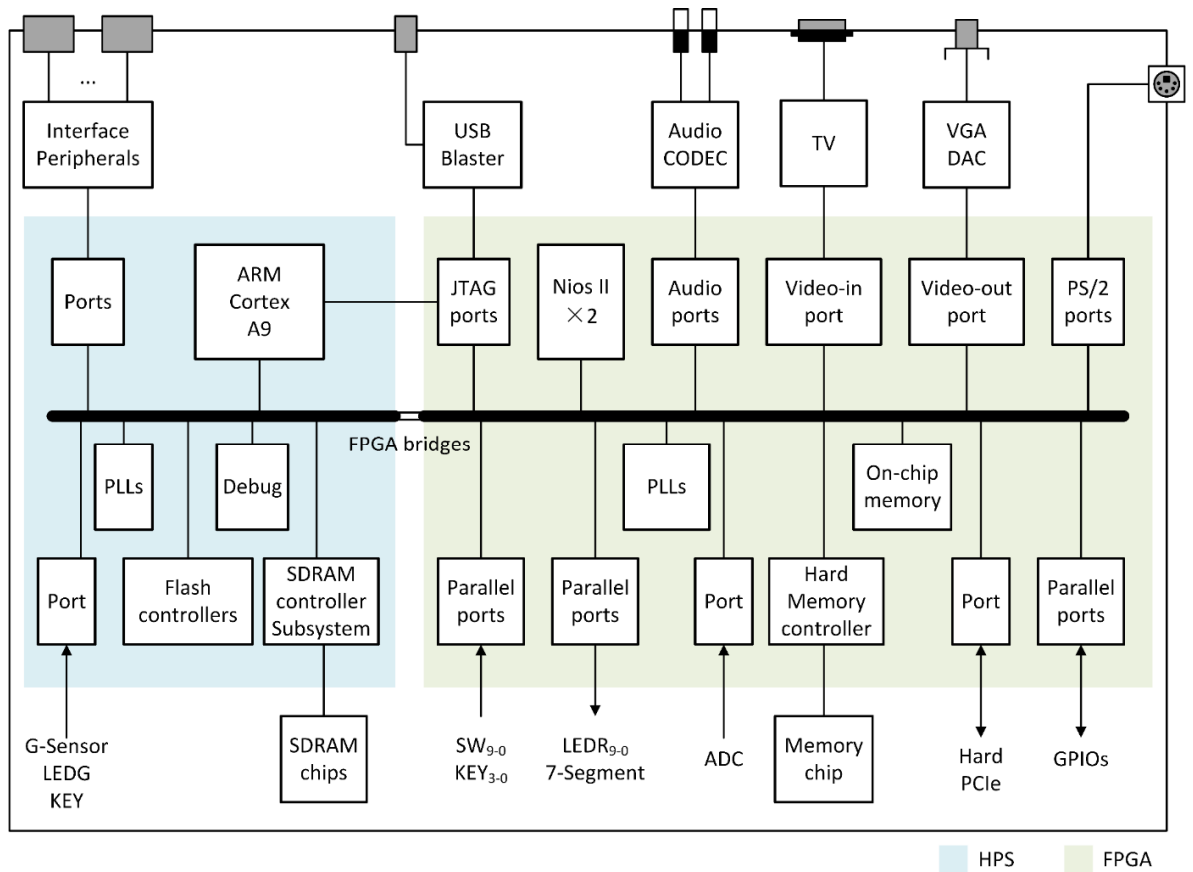


Figure 2-15 Block diagram of the DE1-SoC system overview [41]

This FPGA is selected for the camera module readout, and this allows the readout process to be handled at the maximum speed. It also gives this work the capability to deal with image and control data in parallel. The digital video data sending by the XTM-640 sensor and TRDB-D5M sensor is in a format that requires decoding and bit arranging. Furthermore, FPGA usage will give this project flexibility of designing, particularly

granting additional operation on stream data. The DE1-SoC can be configured by a Joint Test Action Group (JTAG) protocol, and it can be connected to a computer through a USB-Blaster, which gives it a faster way of debugging and configuration.

### **2.3.1. HPS**

As shown in Figure 2-15, the Hard Processor System (HPS) contains an ARM Cortex A9 dual-core processor. The ARM A9 is the microprocessor unit (MPU) present two 32-bit Central Processing Units (CPUs) and combined subsystems which are carried out as hardware circuit and peripherals in the Intel® Cyclone V SoC chip [41]. The flash memory controllers, SDRAM controller subsystem, debug components, phase-locked loops (PLLs) form the subsystem in HPS [42].

The architecture of the ARM A9 processor is Reduced Instruction Set Computer (RISC). It is a composition with arithmetic and logic operations executed on operands in the general-purpose registers. Only Load and Store instructions provided by the program will lead to data movement between the memory and those registers. The data byte addresses are in a 32-bit word format, and the word length of the processor is 32 bits. The address is designated with the little-endian style; this ordering places the LSB in lower byte addresses. Any of the input/output peripherals or devices connect to the system that the ARM processor can access are called memory-mapped devices, in which the memory locations can be read/written.

An addressing mode contains the information for the processor to specify the address of the desired memory location. The Load command handles the addressing mode and copies the contents in one memory location into a destination general-purpose register. The Store instruction copies the content in the general-purpose register and then writes it into one memory location given by the addressing mode. The addressing mode used in

this work is Offset addressing mode, which means one address is specified by adding the base register's content, and an offset is provided directly in the program.

The HPS has access to the SDRAM controller subsystem, which support the DDR3 device. The ARM MPcore is connected to a 1 GB DDR3 memory via the HPS memory ports. This part of memory is used to store stream data from the cameras and be used by the ARM processor. The memory is constructed as  $256\text{M} \times 32\text{-bits}$  and is mapped to the address space `0x00000000` to `0x3FFFFFFF`.

The HPS part will boot from a micro Secure Digital (SD) card in this project on a power-on reset. The first stage executes the Boot Read-Only Memory (ROM) code to initialise all required hardware components and preloader. Then the preloader starts the SDRAM interface and fetching the next bootloader image from the SD card. The bootloader contains a Universal Boot Loader (U-boot) file, which can set up the Linux Operating System (OS) environment and fetch the OS image to boot the embedded Linux system.

### **2.3.2. FPGA**

This work also has some parts that are implemented inside the FPGA in the Cyclone V SoC chip shown in Figure 2-15. Two Nios II processors can be implements on FPGA, and it can be used as CPU in some designs. It also has access to the SDRAM through the SDRAM controller subsystem through the FPGA-to-HPS SDRAM interface [40]. The FPGA configuration image on the SD card configures the FPGA fabric input/output pins through the HPS part.

Data transfer between the ARM A9 processor and input/output peripherals are supported by parallel ports implemented in the FPGA as they are connected to the FPGA bridges. The main functional parts in this design are two general-purpose

input/output (GPIO) ports, which will be used to connect two cameras.

### 2.3.3. FPGA bridges

The components of the HPS connect with the FPGA fabric through bridges, and one part can have a master to communicate with a slave on another. This activity is provided by HPS–FPGA memory-mapped (MM) interfaces, which includes: FPGA–to–HPS bridge, HPS-to-FPGA Bridge and Lightweight HPS-to-FPGA Bridge. As depicted in the block diagram Figure 2-16, the system interconnection consists of the main level 3 (L3) interconnect - an Arm NIC-301 module, and level 4 (L4) buses. The data width(s) is beside each master (M) or slave (S) interface.

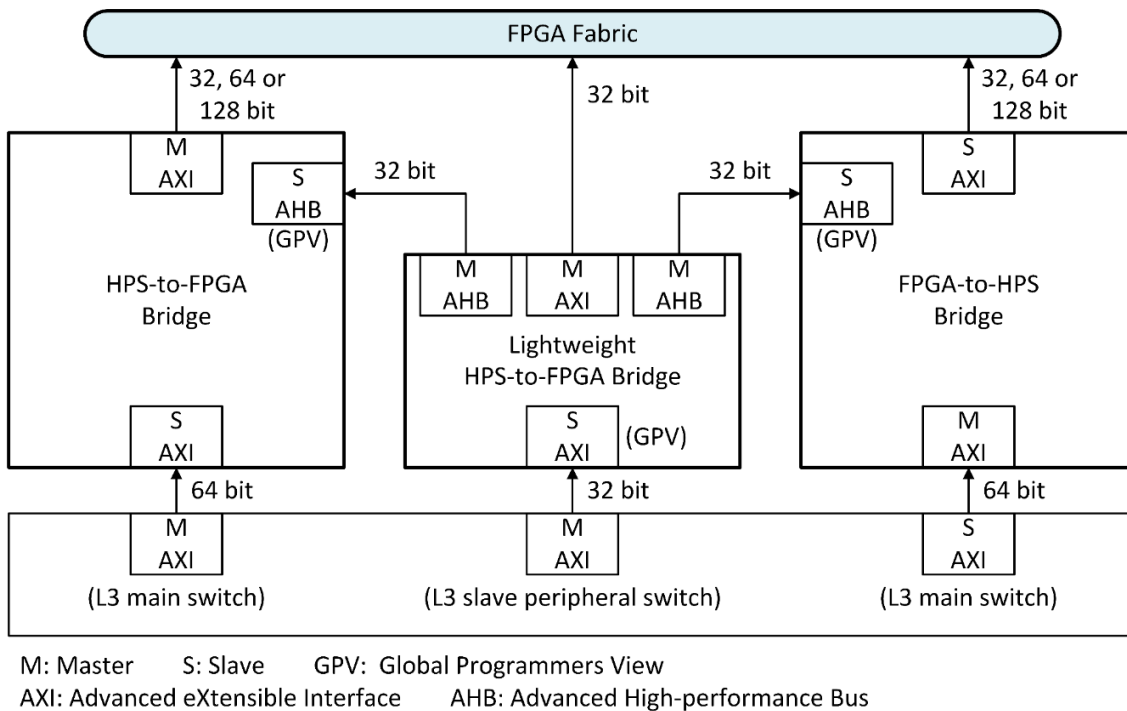


Figure 2-16 DE1-SoC AXI bridge block diagram [43]

## 2.4. Hardware choices

Instead of a cooled infrared camera, an uncooled infrared camera is chosen for this project

because the cooled camera is often proved bulky and expensive. The Xenics XTM-640 series comes with multiple interfaces, and a PCB board can be easily designed for QTE Samtec interface to make it compatible with the FPGA standard GPIO.

The Terasic TRDB-D5M digital visible camera provides all components needed to integrate with the standard GPIO, which means no add-on hardware needs to be designed, and it is compatible with the DE1-SoC board.

Several development materials are provided with the DE1-SoC development board. This development board supply ultimate design flexibility. It can take the design further by integrating an Altera SoC FPGA with an embedded ARM processor. By combining the ability of reprograming, high-speed and multi-thread processing, as well as including a wide array of features that encompassed multimedia, real-world sensing, and networking. Another processor board that could choose is DE10-nano-SoC. It has similar feature to the DE1-SoC but smaller in size, which could possibly make the dual camera system more compact in size. However, the power supply is not adequate for two cameras to operate, and the infrared camera will need an additional power supply, which will lead to the complexity in design the box enclosure and mount points.

## **2.5. System hardware design**

To control and read out from a visible sensor and an infrared sensor, conduct network communication, apply image processing algorithm, distribute power, and handle all other requirements of a dual-camera embedded vision system, an electronics structure has been designed and implemented.

### **2.5.1. System requirement**

The goal is to design the dual-camera embedded system as a standalone device to obtain

the readout stream from two camera modules and process it inside the FPGA. The general idea of designing a box enclosure for this dual-camera system in order to protect the electronics and the sensors. The main functionality is to stabilise the relative position of two camera modules for stereo calibration or image registration to generate an overlaid image. The TRDB-D5M kit is designed by Terasic, and its output ports can be directly connected to the GPIOs. The infrared sensor module XTM-640 requires an interface board to make all its signals and power supply compatible with the FPGA GPIOs.

### 2.5.2. Design of the box enclosure

The mechanical enclosure has created a whole chamber for the processor board and two camera modules. The acrylic sheets are used for as it is cheap in price and easy to manufacture. The box is assembled with nylon screws as they are good in ductility and malleability so that the acrylic board will not break under its pressure.

As shown in Figure 2-17, the TRDB-D5M camera module and the DE1-SoC processor board are fixed with the enclosure. In the top layer design of the box, two paths are extruded in order to hang the XTM-640 infrared module with screws and gives it mobility to be adjusted to front or backwards. A T-shape sheet penetrates through the chamber is giving a fail-safe of the infrared module in case it falls, and giving space to fix the jumper wires connection between the infrared module and the processor board. Two vertical sheets placed next to the infrared module with serrated extruding provide the sensor with a mounting point at the side to rotate slightly on one point. The enclosure cuts are also giving one access to pushbuttons, keys, interfaces, and FPGA Configuration Mode Switch (MSEL). The dimension of the box is  $W186 \times L138 \times H167mm$ , the weight of the system is less than 1kg.



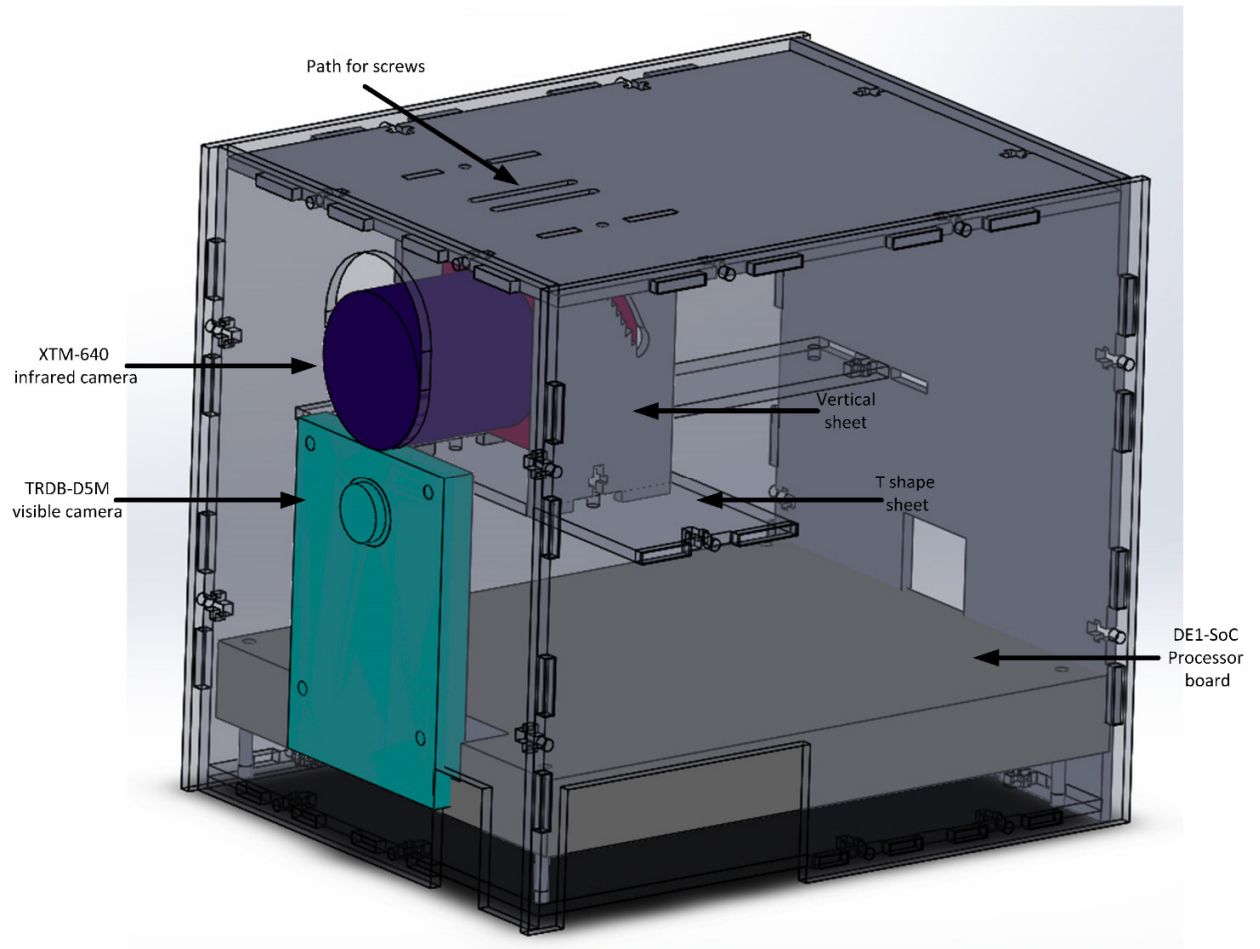


Figure 2-17 The enclosure of the dual-camera embedded system

### 2.5.3. Design of the printed circuit board to condition XTM-640 infrared camera communication

For transmitting the data from the sensor in this project, a parallel interface that uses the Samtec Connector is applied. An interface printed circuit board (PCB) has been designed to act as a bridge between the FPGA and the sensor, as well as to adapt the voltage levels for both UART and parallel signals. This connector of the camera module itself mates with connector type Samtec QSE-020-01-L-D-A is added to the interface PCB as the first output receiver.

The XTM-640 sensor has a total of 16 lines for digital video data, three lines for the serial communication protocol, two lines for UART communication and two lines for power supply. The GPIO pins on FPGA supply 3.3V voltage which is the exact amount the infrared camera needs for the power supply. Three voltage-level shifters are added in Figure 2-18 to decrease the voltage from 3.3V to 2.5V as all other I/O pins apart from voltage supply pins requires an input of 2.5V. Several capacitors are connected between the power supply and the output from the voltage-level shifter to filter out the noise may be caused by the system. A green LED is in series with a resistor to indicate if this PCB board is functioning. In order to give the camera module sufficient power, All the wires connected to the power supply or with the ground are heavier than other signal wires to reduce the voltage drop from the FPGA to the camera because a thicker wiring results in lower resistance.

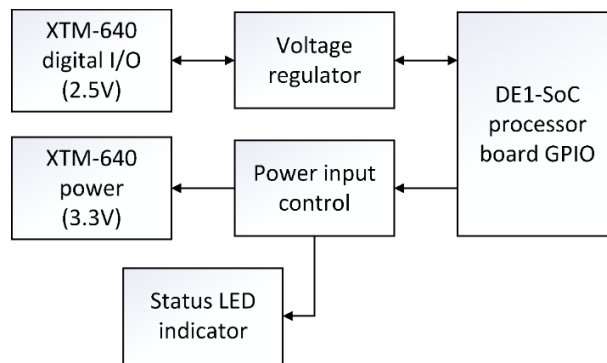


Figure 2-18 XTM-640 interface board block diagram

## 2.6. Data flow design

As shown in Figure 2-15, the FPGA has memory devices, and the Linux OS is running on it. The OS is written on an SD card, so it is easy to change the root file system's content and the Linux kernel image. Having the possibility of booting the whole OS on an SD card guarantees the system a fast and straightforward way of updating any drivers,

debugging, and restoring.

The data received from the two camera sensors are stored in the DDR3 memory on the board. This section of memory can be accessed by the HPS part via the HPS-FPGA bridge. The design of this work allows two camera sensors to store images and other data on DDR3 memory. The system has a Triple-Speed Ethernet (TSE) On-Board Physical Layer (PHY) Chip which supports the system to communicate with the computer through Ethernet, under the condition of they are both connected to the same router.

Figure 2-19 illustrates the data flow of the dual-camera embedded system. It can be divided into three sections: the gateway on FPGA, a server that operates on the ARM processor and a client running on a PC.

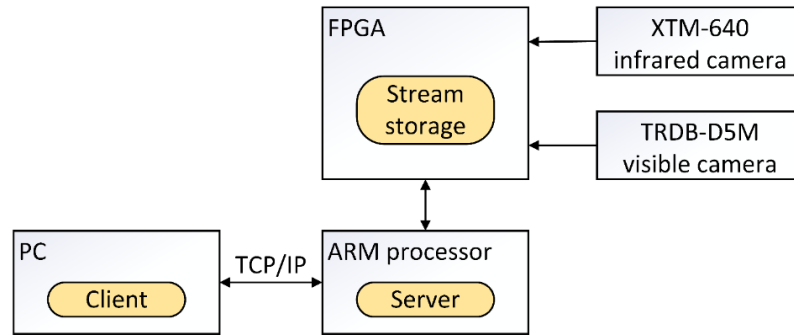


Figure 2-19 Data flow on the dual-camera embedded system

The FPGA gateway mainly implementing the readout process for two camera modules and apply most of the image processing algorithm on the stream data. The digital video data are stored in the DDR3 memory simultaneously because separate camera gateway modules are designed for each camera module, and they can operate in parallel. The server manages the client communication over the network, access the addresses where the image data is stored and prepares the image data waiting to be transferred to the client. The client on a computer connects to the server through the network to control the readout process and allows one to send commands to the server

to show the captured image on a screen or as a stream format. The overall objective of this project is to create one image with both visible and infrared information, so after transmitting the image to PC and during the early stage of software designing, some image processing algorithms are investigated and tested on the client first, then being translated to OpenCL code run on FPGA.

## 2.7. Summary

This chapter presents some detailed information on the thermographic camera module XTM-640 and the visible camera module TRDB-D5M. The infrared camera uses microbolometers as the detector. The digital video signals are generated after being processed by back-end electronics. The visible camera uses a CMOS sensor as the detector, and each sensor uses ADC to generate digital video signals. Both camera modules are using the Camera Link standard on the camera interface for transferring signals, and serial communication protocols are applied for alteration on camera settings. These two cameras are integrated into a signal system: the development kit DE1-SoC processor board, with appropriate hardware designed for connection in between. The platform DE1-SoC development kit is introduced as well. This is a processor board consist of FPGA and ARM processor, and these are selected to handle the control and readout of the two camera modules, the processing of the camera module's data and the communication over the network. The design of the required electronics hardware for the dual-camera system has been presented, the hardware is divided between the interface board to connect the DE1-SoC and the infrared camera, and the box to contain the whole system electronics. The gateware implemented on FPGA to handle part of the data flow is discussed in Chapter 3 and Chapter 4.

## Chapter 3

# System design methodology

For readout and to process the stream of data from the infrared camera and the visible camera, hardware description languages (HDLs) are used to design the logic gates. The HDLs are translated into programmable files uploaded to the FPGA, which forms a gateway system implemented on the FPGA. An open standard for cross-platform and parallel programming called Open Computing Language (OpenCL) is used to perform parallel processing and increase processing efficiency. This chapter focuses on the design tools used and explains the various task in this project. Then an introduction of OpenCL is given with the supportive design to enable this feature, following by the design of the board support package (BSP) and the OpenCL kernel interface. The processor board partition plan for resource distribution and allocation is explained at the end of this chapter.

### 3.1. Design methodology

This work uses HDLs to describe the system gateware layout and behaviour, structure, and data flow. The Intel Quartus Prime Software is used to assemble and compile the HDL code. The function of performing timing analysis and configure the target FPGA with the programmer is also applied after the HDL designs have been analysed and synthesised [44]. One advantage of using Quartus is that one can plan for initial input/output pin assignment and specify the design constraints in the early stage of constructing a large project [45].

The system integration tool called Platform Designer, formerly known as Qsys in Quartus Prime Standard Edition software, is used to facilitate the design requirements of packing individual modules, minimise the design cycle, and maximise the system performance. In the Platform Designer, one can use intellectual property (IP) cores from the Intel FPGA IP library or create their modules [46]. The interconnection between these logics is automatically generated with the interface of these modules are specified. In this project, some IP cores are taken from the official library, but the main blocks and interfaces are created particularly for this application.

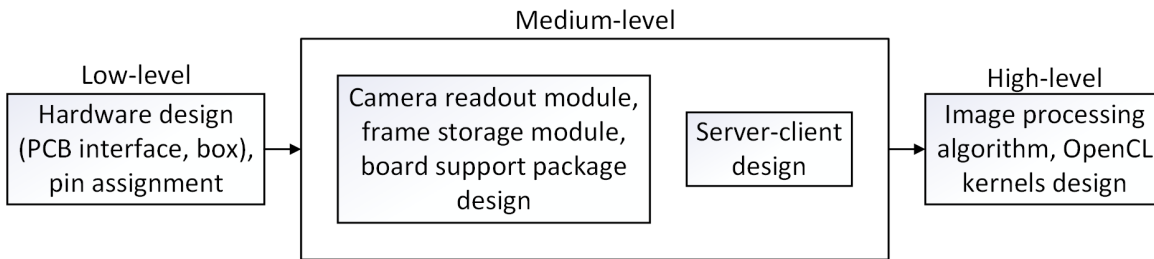


Figure 3-1 Design flow of the dual-camera embedded system

The block diagram in Figure 3-1 describes the various tasks involving in this project. This process can be divided into low, medium, and high levels. The low-level design is

mainly the hardware design; as mentioned previously in section 2.4, it is about signal generating for readout purpose and sensor control. The medium-level design includes two parts; one is to construct modules in Quartus software to produce the reading out of the image and board support package design, the other is to transfer the image to another system. After the other system receives the frames, the high-level image processing design is done on PC for verifying and then coded into OpenCL kernels to run on FPGA.

### 3.2. Gateware architecture

The gateware design in this work includes two majority modules. One module to communicate with one sensor and process the received digital video data, another block to communicate with the HPS, which has a Linux OS running on the ARM processor. With both camera modules connected to the FPGA, designing this gateware is to obtain a fast communication rate on receiving image data. The complex hardware logic is designed with different processing stages, actions of two camera frame storage can be implemented in parallel. Figure 3-2 shows how the modules are linked with external sensors.

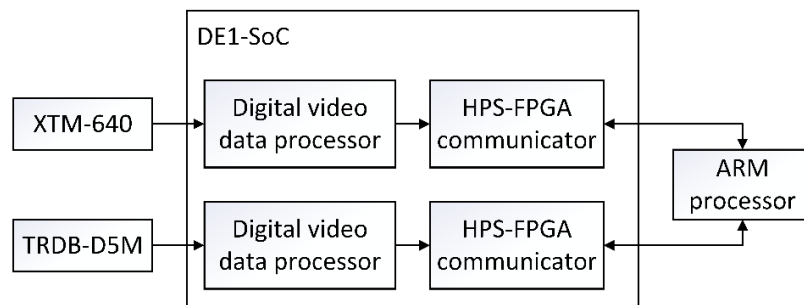


Figure 3-2 Main blocks of the gateware system designed on DE1-SoC

The ARM processor uses an HPS-FPGA bridge to communicate and control the digital video data processing block. The processing block handles the stream data and image status and manages the data storage to DDR3 memory. This essential block is

written in Verilog or System Verilog using the standardised interface and then been packed as standalone modules in Platform Designer. The top-level designs are written in Verilog, mainly for connection between pins and peripherals. Having the same characteristic in the module avoids developing custom interconnection fabrics and signalling manually, focuses on how the data will transfer instead of knowing exactly when the data is transferred. These modules are often flexible and can be applied to various applications or platforms.

Figure 3-3 represents the gateway in Platform Designer modules. To get the sensors image information and control sensors streaming in a standard method, there are control register and status registers on each camera module connected directly with sensors, where the processor can approach these registers and communicate with them. All modules in this system are linked with the simultaneous reset signal. Thus, the system needs only one reset master to control the reset operation.

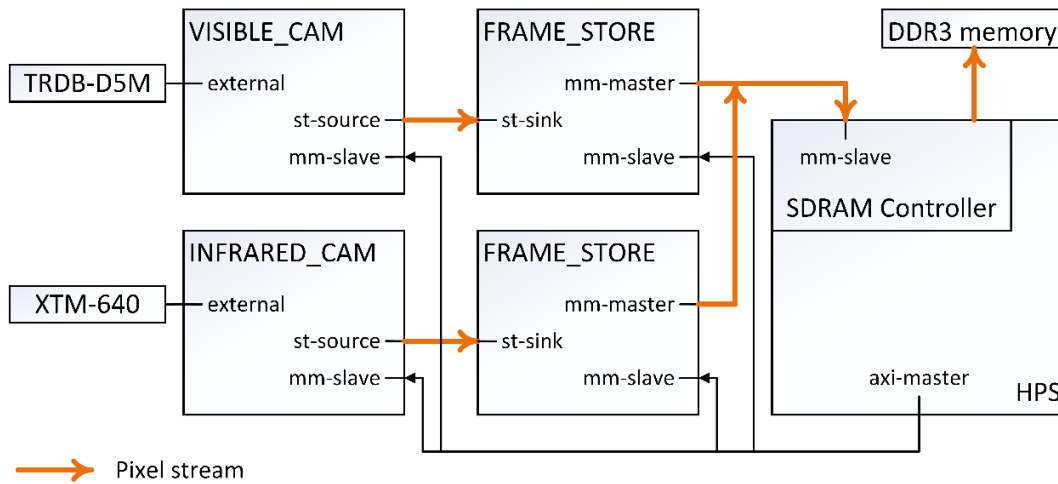


Figure 3-3 Platform Designer modules that form the readout process

The SDRAM controller subsystem on HPS forms the interface between the ARM Cortex-A9 MPU subsystem and the external SDRAM on FPGA fabric. The processor can read or write data in different memory locations, as the interface accepts Avalon®



Memory-Mapped (Avalon-MM) transactions, which makes the FPGA a memory-mapped device. The masters implemented in the FPGA fabric have access to the SDRAM controller subsystem in HPS through the FPGA-to-HPS SDRAM interface. This interface has command ports that issue read and write commands and the receiving of write acknowledge responses. It also has 64-bit read data ports and 64-bit write data ports to either receive data returned from memory read or transmit write data. These three kinds of ports are the construction of the Avalon-MM interface, and the interface is configured during the process of FPGA configuration.

### **3.3. Support of OpenCL on the dual-camera system**

OpenCL is a C-based design environment for a heterogeneous platform contain a host CPU and hardware accelerators. By offloading some computationally intensive code onto graphics processing units (GPUs), digital signal processors (DSPs) or FPGA in this work, the application speeds up as the execution of a program is in parallel [47]. The program is coded in kernel language and been passed through a device compiler, and then the accelerator conducts a parallel execution. The OpenCL standard defines application programming interfaces (APIs) included in host CPUs. For these parallel executions, compile and loading kernel programs to accelerators and discover which accelerator is compatible with the system. The whole computation can be designed with OpenCL, such as the computation on the host, the data transfer between the host CPUs and hardware accelerators, and the computation on accelerators.

By compiling and analysing the OpenCL kernel code, the equivalent FPGA circuits and interfaces circuits can be generated to fulfil the accelerator's environment. Both GPUs and FPGAs can be used as accelerators of a heterogeneous computing system. On the one hand, due to an FPGA accelerator can have more kinds of architecture adopted

[48], such as data flow, systemic array, single instruction multiple data (SIMD) and multiple instructions multiple data (MIMD) processors, the implementation on FPGA can achieve higher performance and lower energy expenditure [49]. On the other hand, task-based and data-based parallelism are allowed in OpenCL, and the code can be profiled to keep higher-level programming. The code reuse rate and compiling efficiency is raised, unlike low-level HDL languages would consume a long period of debugging and assembling [50].

Image processing on a compact FPGA requires a vast amount of processing units, the heterogeneous architecture can be seen as a solution to this issue. In addition to DSPs, FPGA further enhance the performance of data processing. Initially, FPGA are mostly made up of flip-flops and look-up tables, the image processing algorithms are then dynamically configured to form complex logic directly performed by hardware. This heterogeneous system uses Khronos' open standard – OpenCL to interact between the device and host. This structure provides the system with reconfigurability, low latency and high energy efficiency due to the OpenCL kernels are offloaded to the accelerator device.

### **3.3.1. Intel FPGA OpenCL Framework**

In this project, the DE1-SoC board is an FPGA device with both hardware accelerators and CPUs. For designs targeting this board, the Intel FPGA Software Development Kit (SDK) provides tools and compiler to build the program. Comparing to the traditional FPGA design using Verilog, VHDL or other low-level hardware description languages (HDLs), the Intel FPGA OpenCL (AOCL) gives an advantage on time-to-market (TTM) as the development and debugging process is compressed. Utilising the OpenCL standard on an FPGA-based heterogeneous system leads to more effective performance and lower

power consumption. In the FPGA environment, the OpenCL constructions are synthesised into custom logic, and OpenCL kernels are compiled to target the FPGA.

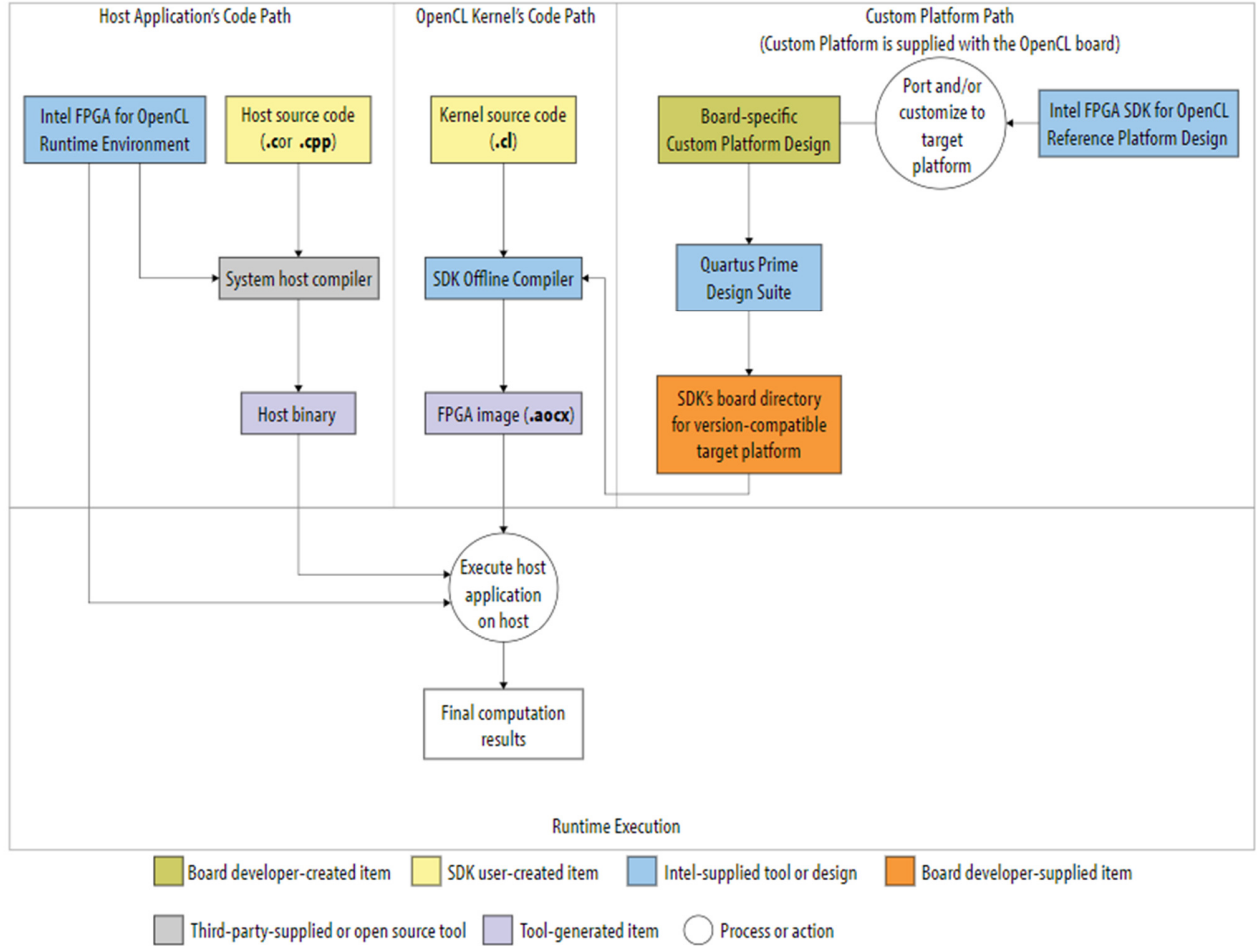


Figure 3-4 Overview of Intel® FPGA SDK for OpenCL programming model [51]

Figure 3-4 describes an overview of a complete OpenCL program implemented on Intel FPGA [43]. The design includes two code pieces that work together: the host code runs on a standard ARM CPU, and the kernel code runs on the FPGA. The host application program, which includes runtime APIs, can be compiled using any standard C/C++ compiler linking to the static and dynamic AOCL libraries. Most code lines are the host code, but the necessary performance code is the kernel code. The AOCL offline compiler builds the kernel structure, data path, and memory structures for the FPGA.

Although some data path inside the FPGA can be generated using the AOCL compiler, the interface between FPGAs and external devices and the interface between FPGAs and host programs still need to be designed using HDLs, such as the camera interface modules in this project.

The OpenCL platform in this work consists of one host program that runs on an ARM processor and DE1-SoC FPGA as a device. Figure 3-5 illustrates the execution process on the processor board DE1-SoC. The OpenCL context is created with one FPGA device; the context contains information such as host-device interaction, device control strategy and memory management. The host program uses a command queue to request action from the device. The command queue consists of instructions such as data write/read, set kernel arguments and execute kernels; these commands are issued appropriately with their execution order and dependencies [52].

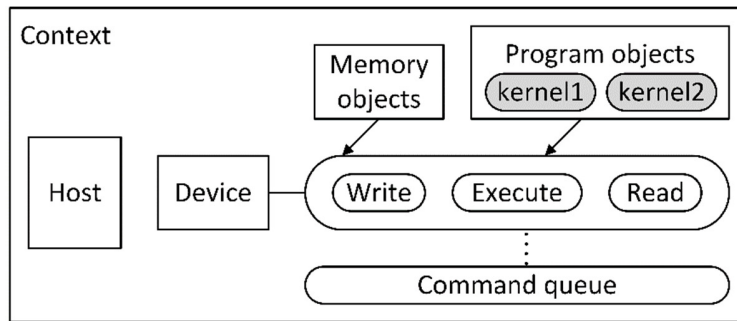


Figure 3-5 DE1-SoC OpenCL execution model

The kernels are functions or algorithms that run on an OpenCL device. In this design, some image processing algorithms are implemented as kernel program to execute on DE1-SoC. The kernel defines work-item representing the unit of execution, and work-groups organises these work-items. The integration of work-items is defined as an NDRange, the size of which is specified in the host program. An example of NDRange of one dimension is demonstrated in Figure 3-6 consists of  $n$  work-groups and each work-group contain  $m$  work-items. The kernel uses global work-item ID and local work-item ID to identify the

item. A work-item which defines as local means it is inside the work-group, and which defines as global is inside the NDRange.

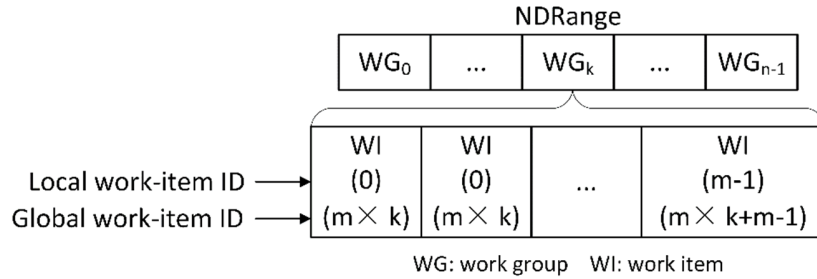


Figure 3-6 NDRange example

There are several memories on the DE1-SoC development board: host memory, device memory. The device memory can be divided into global memory, constant memory, local memory, and private memory. A detailed explanation of the location of these memories is shown in Figure 3-7.

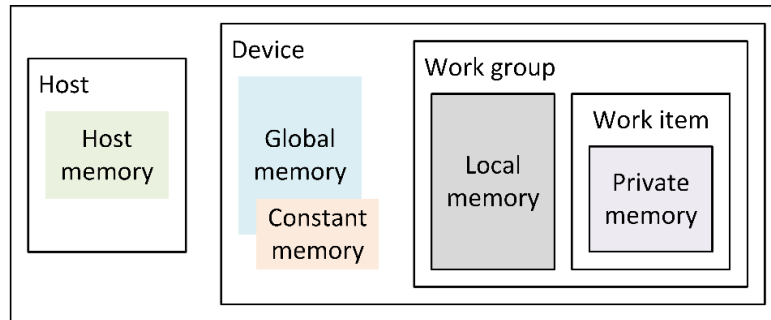


Figure 3-7 DE1-SoC OpenCL memory model

The host memory is on the ARM processor and only can be accessed by the host program. As the DE1-SoC is a SoC FPGA platform, the SDRAM on the DE1-SoC board is treated as the global memory in this design, and it is accessible to host and device, the efficient data exchange is through HPS-FPGA bridges using a memory-mapping method. The constant memory is a part of read-only memory for the device. One particular work-group owns a local memory, and the data in this local memory can be shared by the work-items in this work-group. The private memory belongs to a single work-item and

cannot be read/write by other work-items [53].

### 3.3.2. Board support package design - FPGA

In order to implement an OpenCL kernel on DE1-SoC, it must be compiled using the board support package (BSP) targeting the board. A BSP is a combination of a fixed partition of FPGA design and drivers that forms the lowest layer of the application software stack. Using a BSP also solve the problem of applying the same code on different FPGAs, as the inputs/outputs, logic and memory resources information is contained in the package. Figure 3-8 shows the relationship between BSP for the DE1-SoC board with the host CPU.

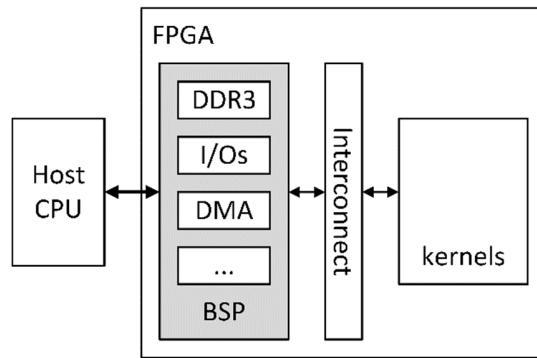


Figure 3-8 BSP overview on SoC FPGA

The Altera Offline Compiler (AOC) translates OpenCL kernel codes into an FPGA design. The kernel system is built into hardware design files and merge into BSP; then, the BSP is copied into the entire project to build a complete Intel Quartus Prime project, which generally means the fixed partition on the processor board. After compiling the OpenCL kernels, the kernels can access the input/outputs or memory blocks through the BSP interfaces. In this design, the kernels are taken camera streams from the SDRAM controller, the image processing algorithm implement on kernels then process these images for a result.

The kernel interface offers a way for the host program to control the execution of the generated kernels. In Figure 3-9, the inputs and outputs of the kernel interface, as well as its sub-components, are shown. The control port is a slave port inside the interface block connected to the host and allows access to the system's control registers. The kernel control register access (CRA) is an Avalon-MM master exported and connected to the kernel by the AOC compiler. There is an interrupt request (IRQ) signal exported on the kernel side and connected to the host. The BSP memory organisation output is connected to the memory controller on HPS; this component is configured in the Platform Designer for the number of global memory systems in the board system.

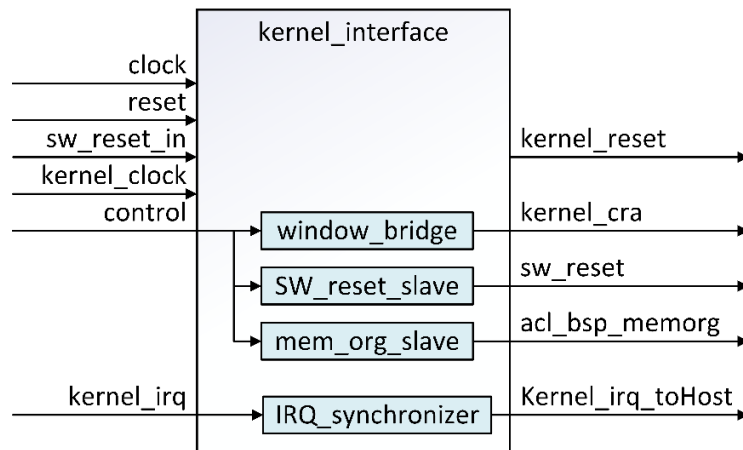


Figure 3-9 Kernel interface

Figure 3-10 illustrates the complete system design, including the OpenCL interface in Platform Designer. First, the camera modules and modules to store frame are the same as in section 3, but the kernel interface takes the stream. After that, there is a block to implement the HPS interface, and the DDR3 controller interface is in the HPS module. There are several blocks providing clocking to the system and reset control, as well as several ID registers to help the host identify the board. The platform DE1-SoC with BSP will constantly automatically adjust the PLL that drives the kernel clock to meet timing requirements.

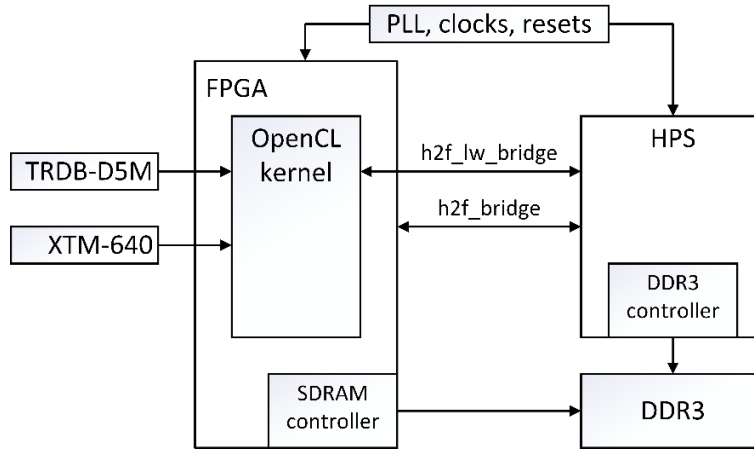


Figure 3-10 A complete view of dual-camera embedded system Platform Designer modules

### 3.3.3. Board support package design – Software

The BSP also contains the drivers' part, which is customised to the provided hardware description of FPGA. The drivers support software implementation and provide communication method between gateway on FPGA and the OpenCL APIs. A set of source files consists of a Linux kernel, a bootloader, and a root file system, which are necessary to generate a Linux image ready to run on the DE1-SoC board [54]. Altera Opensource provides the newest Linux kernel files, but some drivers such as FPGA bridges are modified as the newest version specify a different name for those bridges, and the host program cannot locate the bridges due to two contrasting names. A custom compiled Linux image is written to the SD card inserted into the board, keeping the FPGA configuration with the OpenCL execution sequence.

### 3.3.4. Design partitioning for partial reconfiguration

A complete Quartus project usually contains the designed HDL files, settings and constraint files. When compiling such a large project as the dual-camera embedded system requires a long compilation time. Introducing the partial reconfiguration to this



design reduces the compilation time, resulting in a faster debugging and implementing process [55]. This project is cumulatively built modules to a whole system. The incremental compilation and partial reconfiguration feature preserve the performance for some unchanged Platform Designer blocks and Intel IP cores, such as the camera readout modules, the frame storage modules and version ID modules.

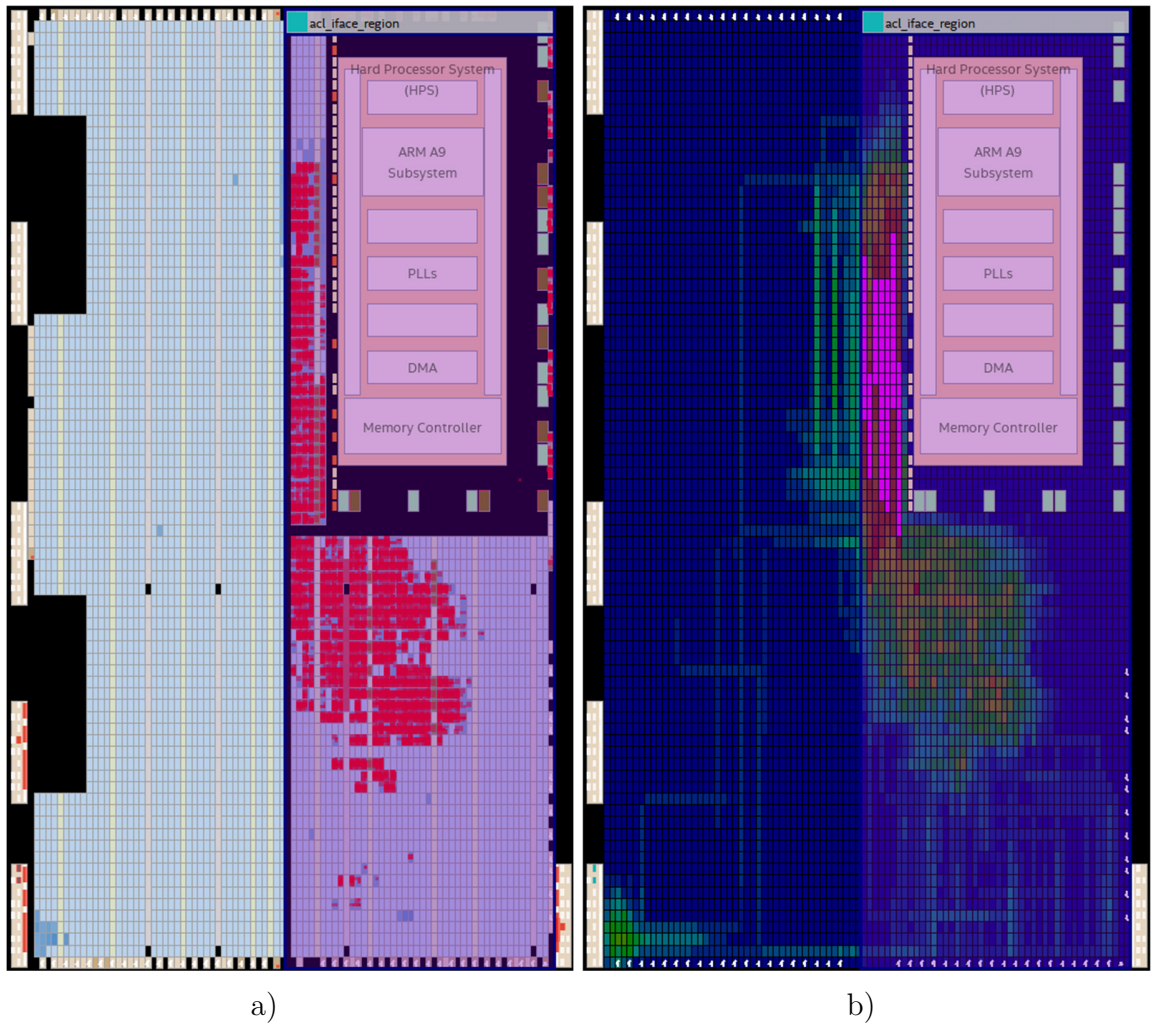


Figure 3-11 Chip planner view (left) and routing utilisation view of LogicLock region (right)

When compiling the kernel codes using AOC offline compiler, the kernel interfaces, the HPS logic, clocks, resets, PLLs and camera modules in this design are remain unchanged. These logics will not be modified during future design iterations. These unchanged parts

form a partition in Figure 3-11, and it is synthesised independently during the early stage of designing and debugging. Then the logical partition is merged into an entire netlist for further compilation, during which the AOC compiler can allocate the remaining resource blocks that able to be given to the OpenCL kernel.

The fixed partition is associated with a LogicLock region, so the logic and interconnection within the partition are constrained into a specific physical area in the FPGA fabric. The design in Figure 3-11 minimises the number of paths through partition boundaries, so signal delays are prevented. Clocks, resets, camera modules, frame storage modules, HPS interface and kernel interface are taken into the partition and HPS-FPGA pipeline bridges to shorten the critical length hence a better timing. These entity instances and nodes are placed within the region during fitting compilation. There is a disadvantage of not using the default fitting plan: Quartus Prime software set the size and location of a region base on its content and timing requirement, but after modifying and constraining the region manually, the routing and performance may not be maximised.

Table 3-1 Statistic of resource usage for the fixed partition

Statistic	Count	Percentage	Statistic	Count	Percentage
ALUTs	4962/64140	7	M10K block	36/397	9
Registers	6129/128280	4	DSP block	6/87	6
ALMs	4595/32070	14	LABs	523/3207	16

In Table 3-1, the statistic of resources used for the infrastructure partition is shown. In total, this fixed partition uses less than 20% of the total resource on the FPGA. More than 80% of the total available resources are left for the kernel, giving the AOC compiler an extensible environment to assemble the codes and enhance the whole project performance.

### 3.4. Summary

The design methodology, including software selection, has been discussed in this chapter. A more detailed illustration of the gateway is presented. The gateway designed is mainly on the camera readout modules, which will be demonstrated in Chapter 4. The program on DE1-SoC allows the HPS processor to control all the actions of the two camera modules and access the DDR3 memory onboard. The AOCL framework is supported on DE1-SoC as the board support package are designed. The board partition is discussed as the partitioning enables faster compiling and debugging. The OpenCL host program design, which is present in Chapter 5, enables the board to run OpenCL kernels and allow the server to access the memories where the image data are stored.

## Chapter 4

# Gateway design

The FPGA can control and communicate with the two sensors using interfaces supported by the Avalon element family. Two modules are designed mainly for readout signals and store frames; the Avalon interface is added to them. This chapter introduces the Avalon Stream and Avalon Memory-Mapped protocol for designing the camera readout modules. A detailed illustration of the main readout blocks and their interconnection in gateway design is explained, then each block is demonstrated separately.

### 4.1. Avalon<sup>®</sup> interfaces

When initiating the Cortex-A9 processor within Platform Designer, the Avalon<sup>®</sup> interfaces are available for connections to memories and peripherals in the system. Altera designs these interfaces, now part of Intel Corporation, to simplify and unify the interconnection design between custom components or Intel FPGA IP cores. Based on

camera modules' particular behaviour, the Avalon<sup>®</sup> Streaming Interface (Avalon-ST) is used to support the unidirectional transfer of image data packets and image profile information. The Avalon-MM interface is a communication method between master and slave pairs, which uses a command to read and write address simultaneously. Using the Avalon interface simplify the design in this work as it benefits from separating the ports of address, data and control signals, there is no need to decode each bus cycle to distinguish address from data or other signals [56]. These standard ports have been added to the designed module with arbitration logic and connect to the peripherals to build up the system-on-a-chip (SoC) system.

#### **4.1.1.1. Avalon<sup>®</sup> Streaming Interface**

The Avalon-ST interface is designed to fulfil the development requirement on high-bandwidth and low-latency stream packets data transit [57]. The source interface and sink interface are defined in this protocol when these two interfaces are connected. The data flow direction is always from source port to sink port. One packet in this standard is characterised as an aggregation of data and control signals, and this packet is transmitted simultaneously.

The source to sink transfer timing diagram is shown in Figure 4-1. One ready latency cycle is introduced between the gap of `ready` signal pulling high and `valid` signal pulling high to ensure the synchronisation between two connected modules. This work uses the most straightforward version interface that only needs `data` and `valid` signals from camera modules. The source generates a `startofpacket` and an `endofpacket` signal to represent the beginning of a stream packet and the end of a stream packet, during which a full-frame and control bits are transferred. Only if the `valid` signal is active, the `data` signal contains the bulk of digital video data sampled from camera

modules. When the sink is ready to accept the source data, the `ready` signal is asserted high only if the source asserts the `valid` signal high. This action prevents data loss from happening as no data can be shifted into if the first-in-first-out buffer (FIFO) at the sink is full.

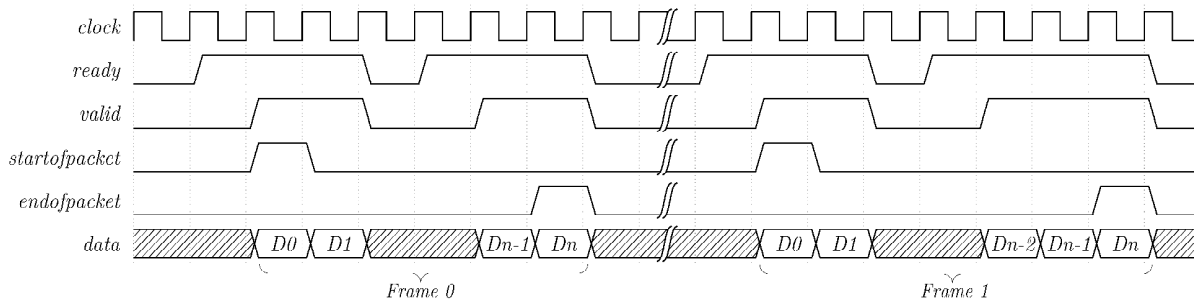


Figure 4-1 Waveforms of Avalon-ST packet transfer with `readyLatency=1` and backpressure

#### 4.1.2. Avalon® Memory Mapped Interface

Avalon-MM interface oriented to the connection of memory-mapped master and slave pairs. Avalon-MM masters in this work mainly are the peripherals on the HPS component, which set up the transfer of the readout modules `VISIBLE_CAM`, `INFRARED_CAM` and stream storage modules `FRAME_STORE` in Figure 3-3. A master interface is also on `FRAME_STORE` to control the stream writing process to the DDR3 controller on HPS. An Avalon-MM read and write slave interface designed on each camera readout modules to allow the control from master on HPS. The waveform of slave writing and reading is shown in Figure 4-2. In these figures, the address signals carry a word address in the slave's address space. `byteenable` signals represent the allowance of transferring data width higher than 8 bits.

The read controller is responsible for handling the HPS read requests. It contains the information of the stream buffer's base address and valid data. The `read` signal is asserted to indicate a read transfer, whereas a `write` signal represents a write transfer.

The slave write controller is subject to handle the HPS write transaction with a request generating every time. It requests writing access to an address that is the same as the interface address. Every bit in `byteenable` is equal to a byte in `readdata` or `writedata`; these data respond to master-to-slave read transfer or write transfer.

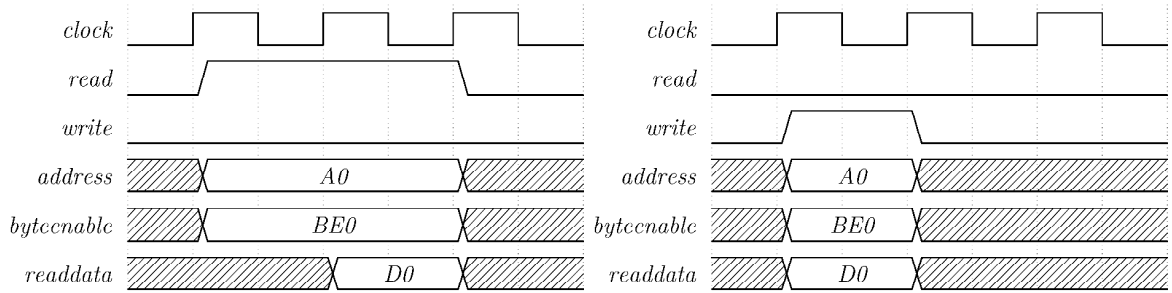


Figure 4-2 Waveforms of Avalon-MM slave read (left) and Avalon-MM slave write (right)

## 4.2. Camera modules

The camera interface modules depicted in Figure 3-3 can be divide into several submodules in Figure 4-3. The readout modules communicated directly with the sensors are `VISIBLE_CAM` and `INFRARED_CAM`. These modules are designed to configure and control the performance of the cameras. The readout process mainly converts the digital image data into stream data using a standardised protocol called Avalon® Streaming interfaces. Parallel with the camera readout modules, the control and status registers (CSR) submodules are defined as `VIDEO_CSR`. These modules contain the information of image height and width, as well as the module's enable control and test stream enable control. The I2C protocol wires and UART protocol wires to realise the serial communication with cameras are connected directly with the supported controller bus on the HPS IP core.

The `FRAME_STORE` module in Figure 4-3 has been written by the research group. There is one `FRAME_STORE` module that connects after each camera readout module.

The frame storage module takes data from the readout module and save the stream images into the DDR3 memory through the SDRAM controller subsystem on HPS.

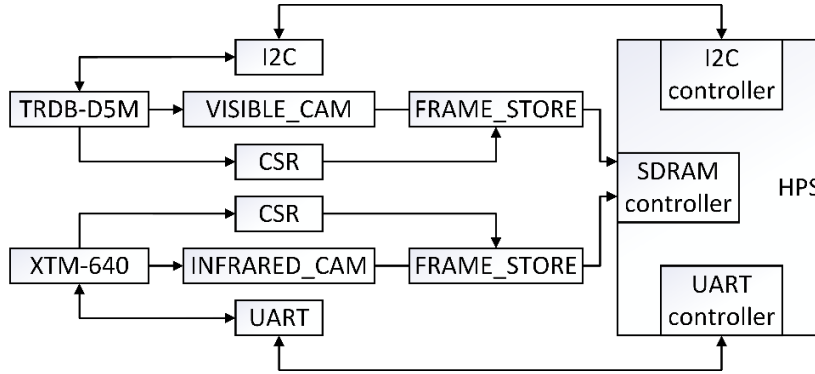


Figure 4-3 Camera data acquisition and control submodule

The different clock domain in this FPGA system is designed using Platform Designer. The DE1-SoC has four 50MHz clock signals connected to the FPGA; one is used to cluster the design logic. The control system Phase-Locked Loop (PLL) is introduced in this design with an input clock signal taking from DE1-SoC. One output from the PLL is tuned to 25MHz, and it is used as the clock input to the TRDB\_D5M module. Another output is set to 100MHz and used on camera readout modules and logics related to SDRAM controllers. These frequencies are chosen to optimise both cameras' frame rate and ensure faster storage to DDR3 memory.

#### 4.2.1. VISIBLE\_CAM and INFRARED\_CAM

The camera digital data readout modules in this work are designed following the Avalon-ST interface standard. By inserting the Avalon-ST interface into both VISIBLE\_CAM and INFRARED\_CAM modules, it allows the digital video data to be processed in a pipeline manner.

There are three stages shown in Figure 4-4 included in this packing and inserting process. The stage 1 stream packet connects the digital video input module named



video\_in and dual-clock first-in-first-out buffer (DCFIFO). The video\_in module transforms the pixel signal into a packet within the pixel clock domain from sensors that follows the Avalon-ST standard. This module is not affected by the backpressure from the DCFIFO, which means the port receiving stage 1 stream cannot signal a source to stop sending data, but the pixel data will be dropped if the DCFIFO is full.

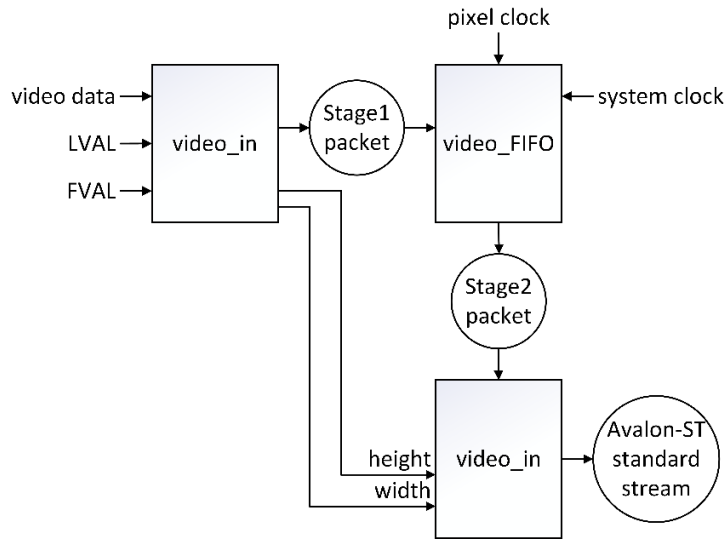


Figure 4-4 Stage diagram of camera readout modules

The stage 2 stream packet linked between the DCFIFO and the control packet inserter module called ctrl\_pkt\_inserter. The DCFIFO is used to synchronise the stage 1 packet at the pixel clock domain to stage 2 packet at the system clock domain. The stage 3 stream packet is the final output of the camera output module after stage 2 packet been processed by the ctrl\_pkt\_inserter. The packet inserter module adds one control packet to the start of a stream, and the control packet is generated by another submodule called VIDEO\_CSR.

#### 4.2.2. VIDEO\_CSR

The VIDEO\_CSR module counts the image column and row information out of the

DVAL and LVAL edge detector. The generated image profile information is stored in registers and shifted to the beginning of each full stream packet. Because there is an Avalon-MM slave interface designed on the VIDEO\_CSR module, the HPS access these registers in specific camera modules address using the memory-mapped method. Table 4-1 list the register map of these CSRs; only the offset address is given as other masters in the gateway can change the base address locations. The Camera Enable register is read whenever FRAME\_STORE fetch a data set to ensure the camera working status. If Test Enable is set to true, a test stream module replaces the stream fetching module. Instead of getting the stream from camera modules, the digital video data is generated by the test module with test pattern signals.

Table 4-1 Register map for VIDEO\_CSR module

Address offset	Name	Description
0x 0	Camera Enable	The camera status register, 0 represents disable
0x 1	Image Width	Return the image column information
0x 2	Image Height	Return the image row information
0x 3	Test Enable	Enable the test pattern module, 0 represents disable

### 4.3. Summary

The presented camera readout module design is capable of transferring the digital video data packet to the frame storage module, following the Avalon interface standard. The camera readout modules can also decode the image data from the camera module and calculating the image profile. A server program running on the ARM processor, explained in detail in Chapter 5, can access the memory where the image data stored and control the gateway system through a set of registers. A client program running on a PC can communicate with the server for data transmission over the network.

## Chapter 5

# Software design for the dual-camera system

In order to visualise the stream reading from the visible camera and the infrared camera, a Video Graphics Array (VGA) cable can be connected directly to the processor board DE1-SoC. The stream data can be displayed on a VGA compatible monitor with a compatible module designed linking the SDRAM and VGA controller. This method has the limitation of unable to save images as a file format for further processing and testing, so a combination of a server and a client program is designed for this dual-camera embedded vision system. The server runs on the embedded Linux operating system installed on the ARM processor, designed to manage network communication collaborate with client software. It also accesses the I2C and UART adapter and controls the gateway system implementing on FPGA. A client program is software running on a host PC connecting the server via a network protocol, collecting digital video data and information. This chapter explains the designing of both server and client application,

starts by describing the server program and the communication protocol applied to design it. Then an explanation of the different type of client software is given.

### 5.1. TCP/IP Server-Client implementation

With the host PC and DE1-SoC wired to the same router, an Ethernet connection between the server and client is established. The client application can send commands to the server program using a Transmission Control Protocol/ Internet Protocol (TCP/IP) protocol, and the server answer the information requested by the client. This protocol is suitable for data transmission in this project because it is highly reliable and has relatively less critical transmission time. The TCP data transfer can guarantee that the data remain intact in data packets and in the same order as it is sent [58]. The entire process can be broken down into the following processes:

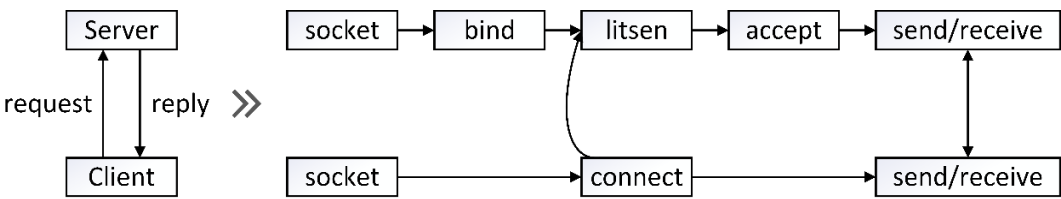


Figure 5-1 Workflow of server-client communication

The server and client program constitutes a socket-based communication over the network. The IP address and port number are used to identify which client is asking for connection or which service is being requested. In the beginning, the server creates a TCP socket and bind the socket to the server address. Then, the server is kept in a passive mode, where it expects the client to approach the server to establish a connection. If the client opens the socket and declares a request to the server by specifying the unique IP address and port number, the server will accept the connection request. At this point, the connection is established, and the server replies to the services requested by the client.

## 5.2. Server running on ARM processor

By introducing C++ library boost.asio to the server design, the network code is portable and is easy to maintain; and as it is a kind of cross-platform networking code library, it can be compiled and run on an embedded Linux system. The server is designed to control part of the dual-camera system; it executes variable tasks base on commands sent by the client program. These commands including controlled the gateway system, switching between two cameras' configuration and reading data from separate sensors. In conjunction with commands, the I2C settings of TRDB-D5M and the UART settings of XTM-640 is pre-programmed on the server, so two cameras are configured when initialise the server program.

### 5.2.1. Server design

The image acquisition and control server's function is to read the information and stream stored on HPS DDR3 memory by gateway. The server's top-level operations can be seen in Figure 5-2. The program exits if any error occurs in checking hardware configuration or failure in the Ethernet connection. There is no end state of this program after entering the control loop as the server is intended to run without stopping while the dual-camera system is powered on; thus, the image data can be taken continuously to form a stream.

The fundamental tasks carried out by the server consist of picking up data and sending it through the Ethernet. There is no image processing algorithm designed on the server as it causes lag in pulling data. The server can be initiated when the system boots, but after all network structure has been launched on the embedded Linux system during the booting sequence because the server would require a network connection when it starts to function. Firstly, the server starts by checking all hardware descriptor presented on

the system, such as the FPGA and processor information, the firmware version and on-board time. For the I2C and UART control of the sensors, they are considered as I2C and UART device, and the server interacts with them through the identifier /dev/i2c and /dev/ttyS1 interfaces. These file address names follow the Linux serial port' Unix tradition, and they are pre-compiled into the Linux kernel. After the server obtains the correct device drivers, the camera module configurations are assigned through reading and writing register values; this process follows the serial communication protocols as mentioned in 2.1.4 and section 2.2.3.

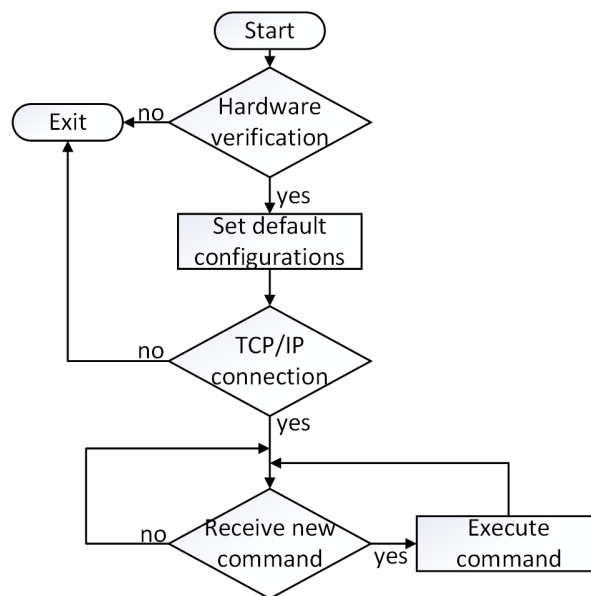


Figure 5-2 Flow diagram of the top-level server application

After the hardware configuration has been verified and camera configurations have been customised, the server starts to read the ID registers from the gateway system. This verification operation ensures that the configuration data is correct, and the HPS2FPGA communication is functioning properly. The server also checks the gateway module availability by reading their ID registers, and this specific function will be enabled or disabled based on the module's status. For example, if the `VISIBLE_CAM` module is

not found when initialising the server, assuming that the client sends a command to ask for a visible frame, the server will reply with an error message shown on the serial terminal stating that the visible camera has not been enabled. However, if the client sends a command to acquire an infrared frame in this situation, a full infrared image will be sent as the infrared module is enabled and remain intact. This design provides the server with flexible and stable communication, allowing one camera to function for debugging, and the server can adapt to the changes in the gateway system without another compilation.

The thread to acquiring images is executed as long as there is a connection between server and client. If the connection lost, the server stops the image acquisition process and will remain on a wait status for an incoming connection. Only one client is allowed to connect with the server simultaneously, and the server will deny all other attempt connections when one client is successfully connected. The server evaluates every command sent by the client to see if the request is reasonable. If the request is on the command list written in the server, the server will parse the command and form an execution request.

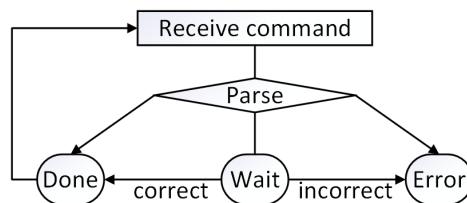


Figure 5-3 Server state when receiving a new command

As shown in Figure 5-3, the server will have three statuses upon receiving a new command. An Error states indicate a wrong command or error occurs during parsing. A Wait state means the command is not complete, and more data is needed. When enough bytes of correct commands received, a request is fulfilled, and the server will execute the

corresponding case, thus a Done state. After a command is executed, a reply object such as image data, profile data or general information will be created and send to the client.

The HPS's DDR3 memory is shared memory as it can also be accessed directly by the FPGA. While the FPGA is operating, HPS can immediately detect the alter in register contents. The server mainly uses the `mmap` function to perform a mapping from hardware addresses to software addresses. The gateway on FPGA stores visible and infrared image in two different base addresses, and each address span is enough so that the data will not overlap. Upon receiving the command of transferring a visible or infrared image, the server allocates the address of the stored image data and determines whether a complete image is stored in the DDR3 memory, then the data will be sent to the client at once.

### 5.2.2. Integration OpenCL host program with the server

As the OpenCL framework is introduced in section 3.3, the essential task of the host code in the server is to verify the obtained OpenCL platform handler and devices, create a context, create a command queue, and program the FPGA with the binary file, which is generated after compiling the kernel code using the offline compiler. The relationship between the OpenCL host program and the server is depicted in Figure 5-4. The server program handles the communication with the client over the Ethernet and configures the camera modules via serial connections. The OpenCL host manages the operation of an OpenCL kernel, where the image processing algorithms are written.

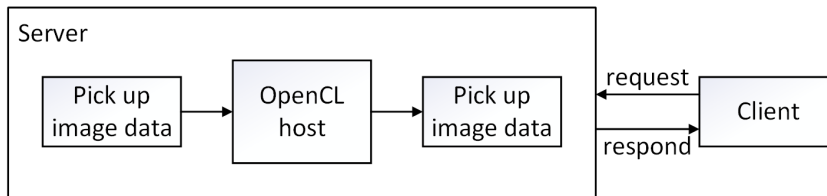


Figure 5-4 Block diagram of the server with OpenCL host

A complete task execution order of an OpenCL host is shown in Figure 5-5. A



memory-mapped method is defined by the AOCL framework for the communication between the fixed partition on FPGA and the host program. The first step is that the OpenCL APIs allocate shared memory for OpenCL Kernels targeting the board DE1-SoC, in where the visible stream and infrared stream are stored. After a program is created using the binary file, the kernel is initialised, and the arguments of the kernel function are specified. Then the input image data is transferred from the host to the device, which is the FPGA in this project. At this point, the host establishes the kernel execution and detect the completion of it. When the completion returns a True value, the output image data is stored to the device memory by the kernel. The host can access the memory location and read from the buffer object to the host memory.

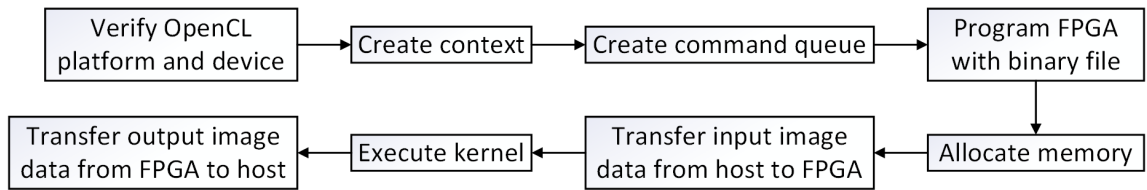


Figure 5-5 Flow of an OpenCL host execution

If the client disconnects or no more stream is needed, the host will have to make sure that there is no command left in the command queue, and to release the allocated memory in the OpenCL host and the FPGA, so that all the allocated memory spaces are recovered, and the runtime cost is reduced. All the above commands have a flag representing true or false; this flag returns a value after each line of delivery a task. This structure ensures the host program with precision and reliability, allowing the server to detect any failure during the kernel execution.

### 5.3. Client running on PC

Several client programs were designed to request data obtained by the server. There is a

third-party software designed by Xenics for the infrared camera module XTM-640 but requires a frame grabber compatible with the module and a Peripheral Component Interconnect Express (PCIe) board on a computer. This is a pure image acquisition and module register modifying software, and there are no custom image processing algorithms that can be applied. Thus, client software has been developed to communicate with the server over Ethernet and debug the server program during the server developing stage. During the dual-camera embedded system development stage, the first client is designed using Python for fast image and information gathering. A second client written in C++ is designed to calibrate camera modules and implement some algorithms when processing the images. A graphical user interface (GUI) client is created using the software Qt to better present image and information.

### **5.3.1. Python client for fast data acquisition**

The Python client is developed in parallel with the C++ server for early-stage debugging and data acquisition, as Python is higher in readability and simpler in syntax than the C++ language. This Python client can also run on various operating systems and platforms, which fulfil the demand of making a portable dual-camera system. The prebuilt library in Python makes the client development process more accessible [59], and the auto-installed shell helps with the debugging process. This is an essential client to ensure that all commands are proper and the device configurations are correct.

A flow diagram of the python client is present in Figure 5-6. The IP address and port need to be set when the client starts. Then the camera has to be selected by the user, and the corresponding image parameters are applied. If a single capture is needed, only one command will be sent to ask for one frame. Otherwise, the client will continuously send commands to acquire frames, which forms a stream.

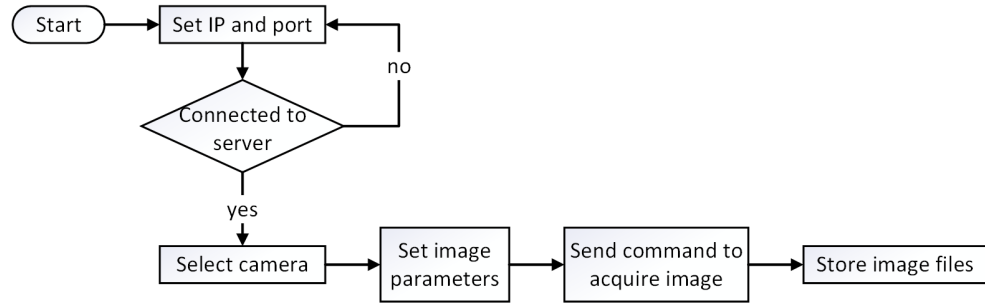


Figure 5-6 Flow diagram of the Python client

This client is designed without a GUI, a simple command-line interface is used, and the image is shown in pop-up windows. The reply from the server, which contains information and configuration details, can be saved as Extensible Mark-up Language (XML) files, the image data acquired can be saved as text format or an image file. The saved image files are used to ensure that the readout from two camera modules is processed correctly on the gateway system.

### 5.3.2. C++ client for image processing algorithm prototype and testing

The C++ based client is designed for camera calibration and image processing algorithm implementing. Most of the geometric camera calibration and stereo camera calibration is performed with the support of OpenCV libraries. OpenCV is written in C++ and has its primary interface in C++, the combination will allow better performance. Since the OpenCL kernels define a C or C++ like language for writing programs, developing this client would reduce the developing time when writing the kernel code running on FPGA, as kernel code mainly contains functions that deal with images. The detailed camera calibration process is present in Chapter 6, along with the rectified images. Moreover, the C++ client working principle is similar to the Python client on retrieving images, but C++ provides performance and memory efficiency, which means it can save time on

image data transfer and result in a higher frame rate than the Python client.

### 5.3.3. Custom GUI client

The graphical user interface client developed using Qt is designed to ease the process of presenting images. As shown in Figure 5-7, users need to set the IP address and port first and then connect the client to the server program. The device setting panel is shown in Figure 5-8. Users can set some data acquisition parameters, such as the region of interest and integration time. The capture mode button in the stream panel in Figure 5-9 is designed to present the image or stream format of the digital video data.

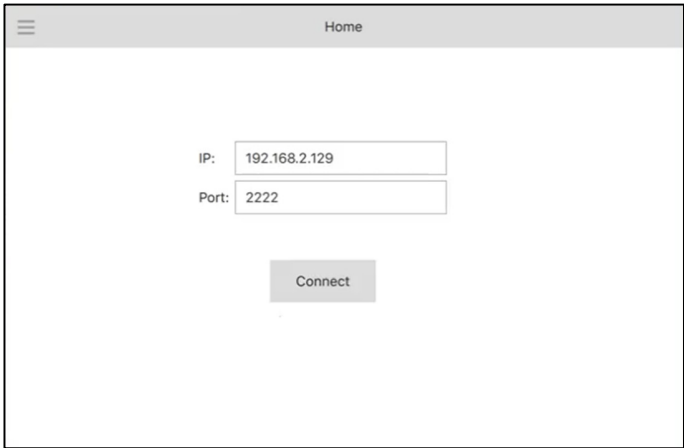


Figure 5-7 Qt GUI client start window for setting IP and port

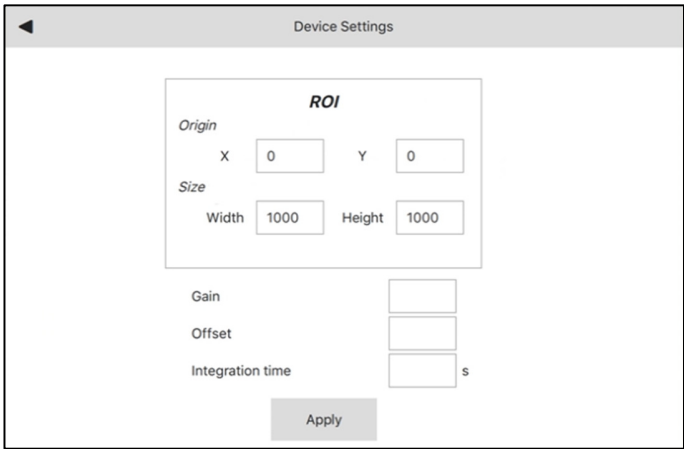


Figure 5-8 Qt GUI device setting panel

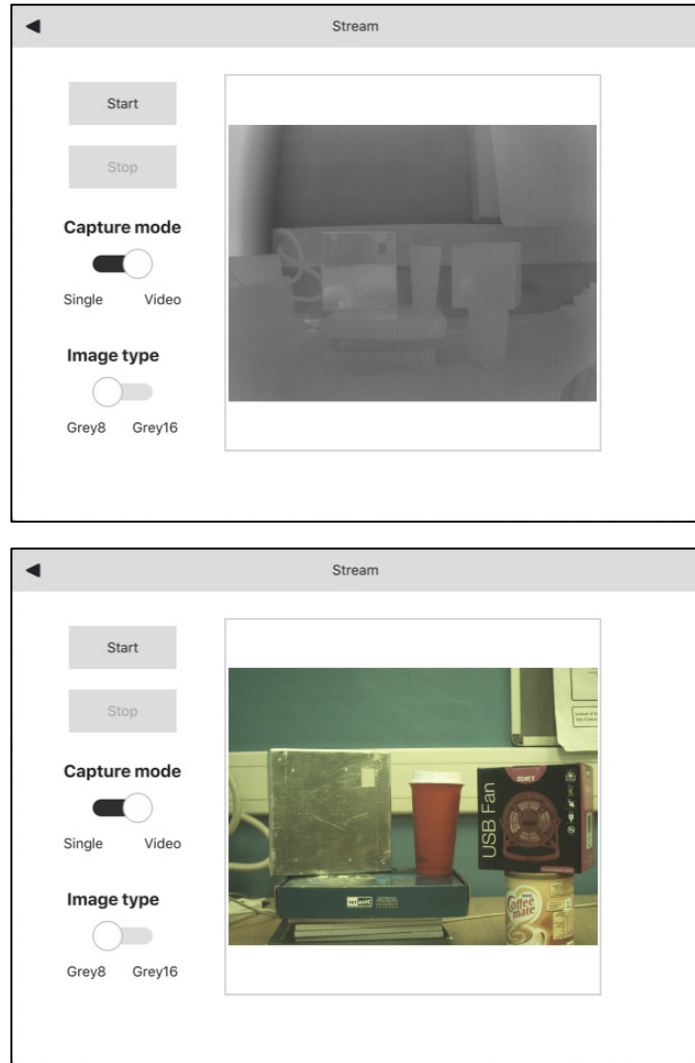


Figure 5-9 Qt GUI stream panel showing an infrared image (up) and a visible image (down)

#### 5.3.4. Result of frame rate

The maximum frame rate achieved for a resolution of  $2592 \times 1944$  image from visible camera TRDB-D5M is 0.83 frame per second (FPS). By configuring the visible camera module, a resolution of  $640 \times 480$  image could be streamed with 8.86 frames per second. For the infrared camera XTM-640, the frame per second is 17.74; this value is higher than the FPS of the visible image with the same resolution because the visible camera

module is configured a binning mode on reducing the resolution. The binning mode is a method that blend adjacent same-colour pixels to produce one output pixel, and this method present a smoother output image with reduced subsampling artefacts. The binning mode is explained in Appendix A.

All the FPS mentioned above is calculated by the C++ client. The start time of a frame is asserted when the server read data from the DDR3 memory, and the end time is asserted when the server's next reading starts. The frame rate is constant and can be kept as long as the client asks for a stream. For a 16-bit full resolution visible image, the file size is about 80.6Mbits, the transfer speed obtained is 66.9Mb/s. However, the bandwidth capability is 1Gb/s for DE1-SoC Ethernet connection. It gives evidence that retrieving the image data from DDR3 memory by the server is the limitation of the frame rate.

## 5.4. Summary

In this chapter, the server runs on the ARM processor is presented. The primary task of the server is to copy the image data from the DDR3 memory of the FPGA to the processor. The server is also capable of configuring the cameras using serial communication protocols to alter camera registers' values. This chapter also demonstrates three different client programs, and they can connect to the server via the TCP/IP port. A client implemented in Python has been used for debugging the server and acquiring image data in the early design stage. A C++ client is designed to carry out the camera calibration, illustrated in Chapter 6, and to implement image processing algorithms that will be translated to OpenCL kernel code. Finally, a client with a graphic user interface is presented for better visualisation of the stream and ease the process of setting image parameters.

Figure 5-10 provides an overall view of the system, including all the hardware and its associated software. The infrared sensor is connected to the FPGA board through a PCB interface. The gateway pack the paralleled pixel data and store them on the DDR3 memory on the FPGA. The OpenCL kernel obtain full frames from the memory and conduct the image processing algorithm to synchronize the infrared and visible image, which will be saved on a separate location on the DDR3 memory. Upon receiving commands from client that ask for an image, the server prepare the data and send it back for client display.

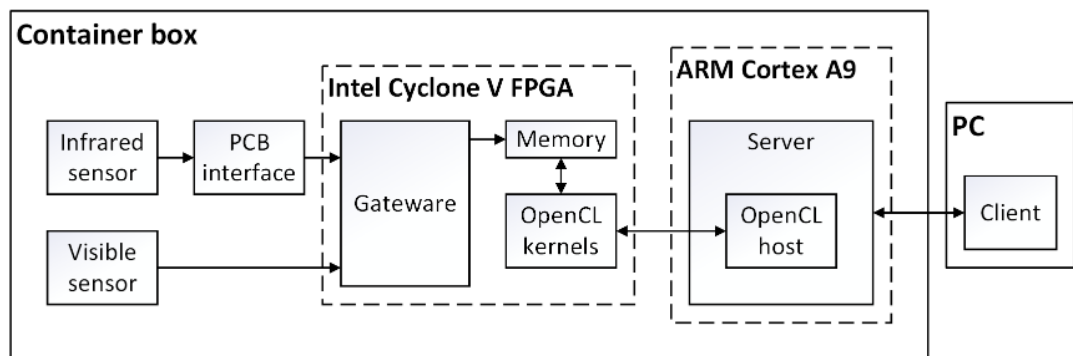


Figure 5-10 Overall schematic of the hardware and associated software

The previous chapters give a detailed introduction and illustration on the electronics this dual camera system contains, the hardware and software design. From next chapter, this thesis will focus on the methodology to achieve a synchronous imaging algorithm.

## Chapter 6

# Geometric camera calibration

The geometric camera calibration is a fundamental procedure to produce accurate geometric measurement from image data [60]. Geometric camera calibration aims to calculate the geometric characteristic of a camera, such as focal length, optical centre, or lens distortion parameters. Moreover, the geometric calibration gives the exterior orientation, which refers to the location and attitude of the camera in the object coordinate system. Two different types of cameras are used in this project, and they are equipped with different lenses. Due to the imperfection of lenses, the captured images can suffer from various kinds of distortions, which result in errors when mapping a 3-D world object onto image planes. For the accuracy of further image processing steps, geometric camera calibration is necessary for this project to estimate the camera parameters and produce an accurate image. This chapter will begin by explaining the pinhole camera model, and how an item in the real-world has been mapped to an image.



Then some examples will be given to illustrate the lens distortion effect. After that, images taken from cameras used in this experiment will emphasize the necessity of geometric camera calibration. The camera calibration method used in this work is presented, along with rectified images and error analysis.

## **6.1. Geometric camera calibration overview**

In this machine vision project, a vital step is to accurately obtain the relationship between the actual object in the 3-D world and its mapping onto the image, which gives a sufficient reason to performing a geometric camera calibration process. The tremendous growth in the machine vision application recently has become a driving force to highlight the importance of precision imaging. In recent decades, geometric camera calibration methods close-range photogrammetry is well established, and a representative approach is Zhang's method [61]. This method is adopted in this work as it is a fundamental approach to get a camera's projective feature by using a planar pattern as the calibration target, whose geometric properties are known. It is a well-known approach for its robustness, flexibility, and most importantly, the calibration pattern board can be made in the faculty workshop and low in cost. The camera observes the calibration pattern and the measurement taken from the image control points generated by the calibration pattern. However, the imperfection of the calibration target would result in inaccurate camera parameter estimation [62]. Especially for infrared cameras, the control points could be even harder to determine [63]. This work adopted two kinds of calibration patterns and will discuss the advantage and disadvantage of them. The OpenCV library and MATLAB toolbox implemented Zhang's method and are considered the conventional method used in this work.

Many cameras are considered pinhole cameras, and the lens distortion is a deviation

from the ideal pinhole camera model. A camera projects its field of view onto an image plane. Considering a camera without lens distortion, a straight line in the actual scene should remain straight in the captured image [64]. The geometric camera calibration process can remove the lens distortion after the camera parameters are calculated. There are two kinds of parameters in the pinhole camera model: the external parameter consists of a camera's orientation and position, and the internal parameters determine the projection from the camera plane to the image plane. In geometric camera calibration, the external and internal parameters are calculated, and the computation involves a lot of matrix calculation and non-linear optimization. A way to examine the precision of estimated camera parameters is to compute the difference between the ideal modelled image and a real one.

## **6.2. Camera model and lens**

A camera model represents the geometrical and physical properties of the given camera. It maps a point from the three-dimensional (3-D) world onto a point on the two-dimensional (2-D) image plane of the camera [65]. In the cameras used in this project, the most appropriate and simple model to describe them is the pinhole camera model. The pinhole camera model is an ideal camera model which does not equip with lenses. This model is presented as a first order approximation of the mapping procedure of a camera. The geometric distortion caused by the lens can be eliminated after the images are rectified using the estimated camera parameters.

### **6.2.1. Pinhole camera model**

The pinhole camera model characterises the mathematical relationship between a world point and its projection onto a point on an image. Figure 6-1 depicts the camera model,

and Figure 6-2 illustrated the relationship between the 3-D world coordinates and the image plane coordinates. The image plane is centred on the principle point  $P$ , and the pinhole camera plane is centred on the optical centre  $C$ . The distance between the point  $P$  and the point  $C$  is defined as focal length  $f$ . Both planes are perpendicular to the focal length  $f$  and the line passing through the principle point  $P$  and the optical centre  $C$  is defined as the optical axis.

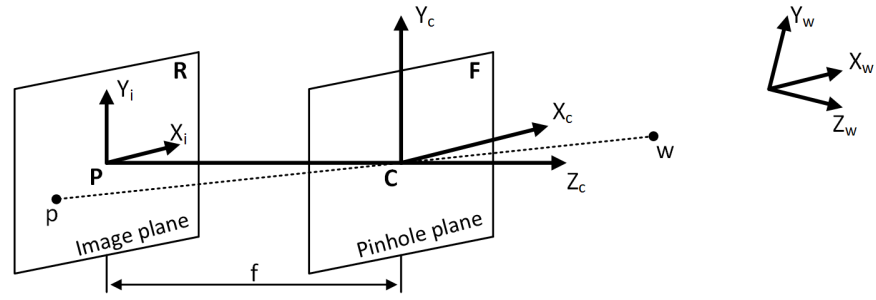


Figure 6-1 Pinhole camera model

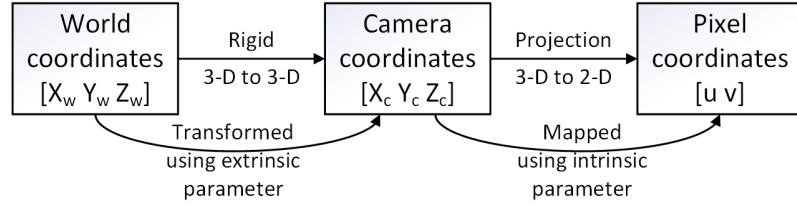


Figure 6-2 Pinhole camera parameters and their relationship

Assuming that the world coordinates is  $[X_w Y_w Z_w]$ , and a 3-D point  $\tilde{\mathbf{w}} = [x y z]^T$  is defined in the world reference frame. The extrinsic parameters represent the transformation of a world point to the camera coordinates  $[X_c Y_c Z_c]$ . The intrinsic parameters represent the mapping of camera coordinates to the pixel coordinates  $(X_i Y_i)$ , and the world point mapped here is  $\tilde{\mathbf{p}} = [u v]^T$ . The relation between the world point  $\tilde{\mathbf{w}}$  and the image point  $\tilde{\mathbf{p}}$  can be formulated as:

$$\tilde{\mathbf{p}} = s\tilde{\mathbf{M}}\tilde{\mathbf{w}} \quad (7.2.1)$$

In equation (7.2.1),  $s$  is the scale factor,  $\tilde{\mathbf{M}}$  is a homogeneous transformation matrix

and is also defined as the perspective projection matrix (PPM). It is a combination of transformations that consist of the extrinsic parameters  $\mathbf{M}_e$  and the intrinsic parameters  $\mathbf{M}_i$ . Accordingly, the homogeneous coordinates  $\tilde{\mathbf{m}}$  in the image plane can be written as:

$$\tilde{\mathbf{p}} = s\tilde{\mathbf{M}}\tilde{\mathbf{w}} = s\mathbf{M}_e\mathbf{M}_i\tilde{\mathbf{w}}, \quad \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = s\mathbf{M}_e\mathbf{M}_i \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (7.2.2)$$

As mentioned previously, the extrinsic matrix contains parameters that represent the rigid transformation from the 3-D world coordinates to the 3-D pinhole camera coordinates. It defines the position and the orientation by a rotation parameter  $\mathbf{R}$  and a translation parameter  $\mathbf{T}$ :

$$\mathbf{M}_e \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}, \quad \mathbf{M}_e = [\mathbf{R} \mid \mathbf{T}] \quad (7.2.3)$$

where  $\tilde{\mathbf{q}} = [X \ Y \ Z]$  is the point on the camera plane, and:

$$\mathbf{R} = [\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{r}_3] = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \mathbf{T} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad (7.2.4)$$

The rotation parameter  $\mathbf{R}$  is defined as the matrix of the 3-D rotation of the camera, and the translation matrix  $\mathbf{T}$  represent the translation vector of the camera relative to the world coordinates. By applying the rotation matrix  $\mathbf{R}$  and the translation vector  $\mathbf{T}$ , the world point  $\tilde{\mathbf{w}}$  can be mapped to the camera coordinates as point  $\tilde{\mathbf{q}}$ . The external parameters define the camera's location and orientation to the world coordinates, and these values may change. The rotational and translation component can recover the location of the camera in the world. Supposing that a 3-D world point is rotated around the  $x$ ,  $y$ , and  $z$  axis with respective rotation angles of  $\alpha$ ,  $\beta$  and  $\gamma$ , the rotation matrix  $\mathbf{R}$  can also be expressed as  $\mathbf{R} = \mathbf{R}_x(\alpha), \mathbf{R}_y(\beta), \mathbf{R}_z(\gamma)$ . The rotation matrix  $\mathbf{R}$  is also an orthogonal matrix. It meets the condition that  $\mathbf{R}^T\mathbf{R} = \mathbf{I}$ , where the matrix  $\mathbf{I}$  is the identity matrix, and  $\mathbf{R}^{-1} = \mathbf{R}^T$ .

The parameters in the intrinsic matrix  $\mathbf{M}_i$  represent the projective transformation from the 3-D pinhole camera coordinates to the 2-D image coordinates, which contains optical characteristics and the internal geometrics of the camera. It is defined as:

$$\mathbf{M}_i \begin{bmatrix} X \\ Y \\ X \end{bmatrix} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \quad \mathbf{M}_i = \begin{bmatrix} f_x & \omega & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (7.2.5)$$

In the intrinsic matrix  $\mathbf{M}_i$ ,  $f_x = f/p_0$  and  $f_y = f/p_1$  are representing the focal length in horizontal and vertical pixels, where  $f$  is the focal length in the world unit as mentioned in Figure 6-1, and  $(p_0, p_1)$  is the focal length in image units. The optical centre is at  $(c_x, c_y)$ , which is also representing the principle point in pixels.  $\omega$  is defined as the skew coefficient, which is non-zero if the image axes are not perpendicular to the optical axis.

### 6.2.2. Lens distortion

In this project, different lenses are used to on the two cameras focus more light rays or radiation rays on the imaging chip. The camera model derived above is based on a simple pinhole camera, which does not take the lens distortion into account. In this work, radial lens distortion and tangential lens distortion are considered to have more effects than other kinds of distortions [67]. Examples of radial lens distortions are depicted in Figure 6-3, a zoom lens causes pincushion distortion, and a wide-angle lens causes barrel distortion [68].

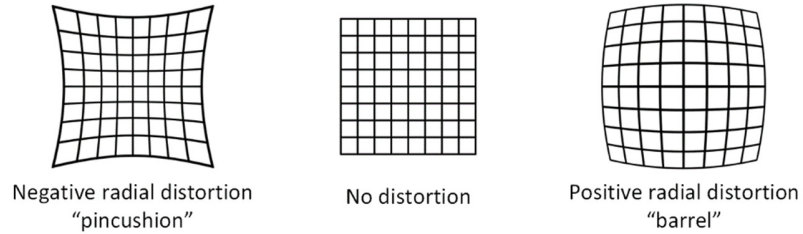


Figure 6-3 Radial lens distortion shown on an example of a square grid

The lenses are axially symmetric base on the mechanical design of the TRDB-D5M and XTM-640 camera model [32] [69], so the radial distortions are expected to be rotation-symmetric [70]. An assumption has been made that the camera lenses' principle axis perpendicular to the image plane and meet at the principle point of the image. Comparing those examples to images taken from both cameras, it can be seen that the images are similar to the type of barrel distortion.

Considering a lens distortion model with a distorted point donated as  $(x_d, y_d)$  and  $(x_{corrected}, y_{corrected})$  is the corrected/undistorted pixel locations. The coordinates  $(x, y)$  are dimensionless as it is the normalized image coordinates acquired from the pixel coordinates by translating to the optical centre and dividing by the focal length in pixel format. The radial distortion can be expressed using an infinite polynomial series as:

$$x_{corrected} = x_d(1 + k_1r^2 + k_2r^4 + k_3r^6 + \dots k_nr^{2n}) \quad (7.2.6)$$

$$y_{corrected} = y_d(1 + k_1r^2 + k_2r^4 + k_3r^6 + \dots k_nr^{2n}) \quad (7.2.7)$$

where

$$r = \sqrt{x_d^2 + y_d^2} \quad (7.2.8)$$

$k_1$ ,  $k_2$  and  $k_3$  are first, second and third-order radial distortion coefficients of the lens, respectively, and the polynomial degree  $2n$  are the parameters that can be calculated from camera calibration [71]. Typically, the high-order terms in equation (7.2.6) and (7.2.7) can be eliminated as three coefficients are sufficient for calibration [72], the fourth-order coefficient  $k_4$  can be added if the image is severely distorted. Thus:

$$x_{corrected} = x_d(1 + k_1r^2 + k_2r^4 + k_3r^6) \quad (7.2.9)$$

$$y_{corrected} = y_d(1 + k_1r^2 + k_2r^4 + k_3r^6) \quad (7.2.10)$$

Another type of lens distortion is called tangential distortion, shown in Figure 6-4, which is caused by manufacturing defects resulting from the lens that are not parallel with the image plane [73].

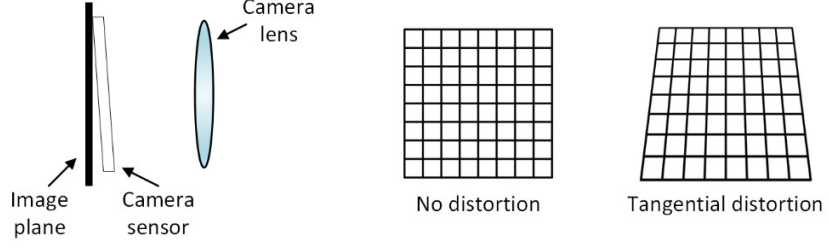


Figure 6-4 The cause of tangential lens distortion (left) and the resulting image shown on an example of a square grid (right)

When considering the tangential distortion, there are two additional parameters  $t_1$  and  $t_2$ . These parameters are named as tangential distortion coefficients of the lens, and the corrected points  $(x_{corrected}, y_{corrected})$  are donate as:

$$x_{corrected} = x_d + [2t_1x_dy_d + t_2(r^2 + 2x_d^2)] \quad (7.2.11)$$

$$y_{corrected} = y_d + [t_1(r^2 + 2y_d^2) + 2t_2x_dy_d] \quad (7.2.12)$$

Supposing there is a tilt angle  $\theta$  between the lens and the image plane due to manufacturing error, the tangential distortion parameters  $t_1$  and  $t_2$  can be defined by introducing another magnitude coefficient  $t_3$ :

$$t_1 = -t_3 \sin \theta \quad (7.2.13)$$

$$t_2 = t_3 \cos \theta \quad (7.2.14)$$

and  $t_3$  can be expressed as

$$t_3 = \frac{1}{\sqrt{t_1^2 + t_2^2}} \quad (7.2.15)$$

### 6.3. Single-camera calibration

The process of finding out a camera's geometrical and physical properties or parameters, such as its focal length and principal point, is called camera calibration [74]. The extrinsic and intrinsic parameter of a camera is computed during the camera calibration, and these parameters can be used to recover 3-D information from a 2-D image and undistort the

images taken by cameras with fisheye lens [71]. The knowledge of these parameters is essential for image overlay solutions in Chapter 7, and the accuracy of the camera calibration is strongly related to the total performance of a vision system [75]. For the further subsequent experiment on overlaying the infrared image and the visible image, the camera parameters are required as an input. The Camera Calibration Toolbox in MATLAB and some of OpenCV functions are adopted to compute these intrinsic and extrinsic parameters and analyse the calibration errors.

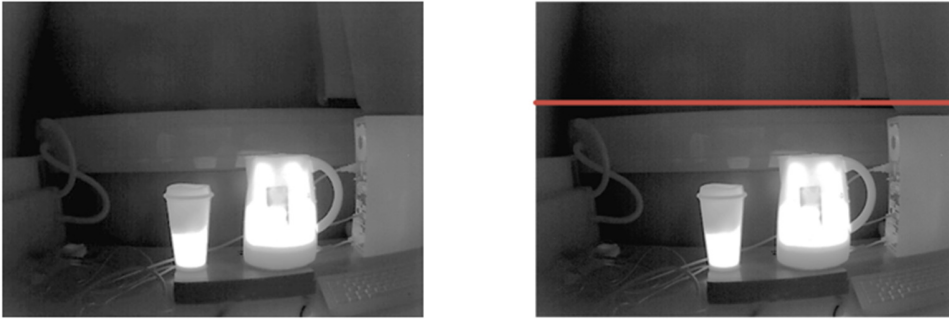


Figure 6-5 Infrared image (left) and infrared image with reference line (right)

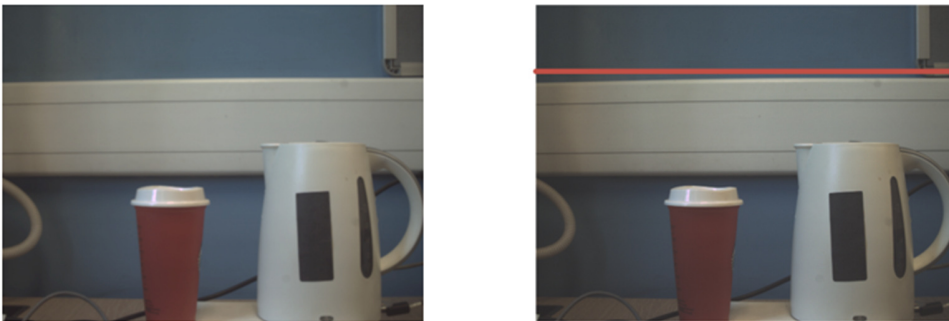


Figure 6-6 Visible image (left) and visible image with reference line (right)

Figure 6-5 and Figure 6-6 show image captured from the infrared camera and visible camera respectively, after introducing a red reference line into the two images taken from the camera, it can be seen that the edge of cable trunking is bent in the image. There is an advantage that the wide-angle effect would facilitate the camera to have a much larger field of view compared to standard conventional cameras. However, the broader field of



view results in the non-linear distortion near the boundaries of the image captured. Thus, it is necessary to calibrate both cameras and undistort the captured image for a further image processing stage. The fisheye effect leads to the inaccuracy of either stereo camera calibration or image registration, as well as range selection when calibrating the temperature using the pixel data on infrared images.

### 6.3.1. Calibration board

The calibration routine takes a set of images when the camera is targeting a known object that consists of many solitary and identifiable points [76]. By rotating and translating the calibration structure, this set of images contains the view of the object from various angles. Then, the camera's relative location and orientation can be calculated from each image, and the intrinsic matrix can be achieved. A regular pattern should be chosen as it is easier to deal with and convenient for manufacture [77]. The aim of using a calibration board is to extract the interest points in an image picturing the board; then, these 2-D image points will be used for camera parameters derivation and calculation.

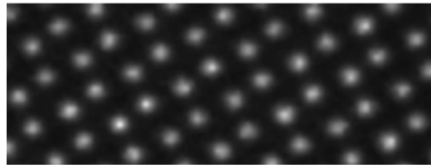


Figure 6-7 Infrared image of a repeating dot pattern

The dot grid pattern is the first choice of the calibration pattern, as the image points are apparent, and the process of shape recognition can be abandoned. For visible image calibration, it is possible; however, it may result in more substantial errors when recognising the interest points for the infrared image. Figure 6-7 shows an example of a repeating dot grid pattern generate by a dot illuminator [63]. It can be noted that the dot is blur due to the heat radiation; it will affect the surrounding temperature and lead

to imprecision when determining the objective points [78]. Thus, an image with a calibration pattern that has sharper edges can avoid imprecision.

A chessboard pattern is used in the second place of camera calibration [61] [79]. It is a kind of pattern which has alternating black and white squares, so there is no bias towards one side or the other in the measurement, and the corner coordinates on the image are easy to be identified. The camera intrinsic and extrinsic parameters will have a minor error if the corners can be detected accurately. For a chessboard to be rotation-invariant, the number of rows and columns should not be the same. When the number of rows is even, the number of columns should be odd or the other way around. The 180-degree rotation ambiguity problem only occurs when the wrong chessboard pattern is being used for stereo camera calibration, which will be discussed in Chapter 7.

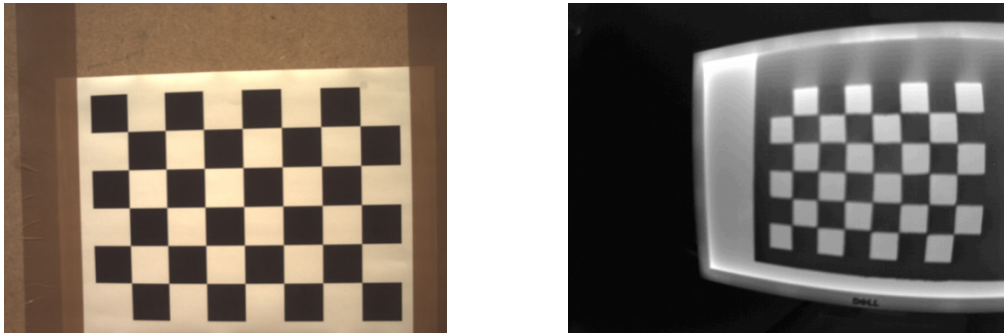


Figure 6-8 A visible image (left) and an infrared image (right) with a chessboard pattern

The chessboard pattern is printed on a paper and pasted on a hardwood board for visible camera calibration. For the infrared camera calibration, a calibration board is made from a cardboard piece with all the white squares hollowed-out. The cardboard is placed in front of a heating plane, so the heat would come through the empty squares and be detected as a bright part by the infrared camera. The histogram equalization function is kept on for the infrared camera module as high contrast image is required, as each corner where the square join can be easily located in the image of high precision.

The chessboard for infrared camera calibration is made to have larger squares with a side length of 30mm to avoid near-uniform temperature and thermal radiance. There is a limitation of using a chessboard pattern to calibrate the infrared camera. As seen from Figure 6-9, when making the calibration board, the corners of black squares should be connected, and the hollowed-out cardboard is not ideal as a chessboard printed on paper, which causes inaccuracy in the corner detection algorithm.

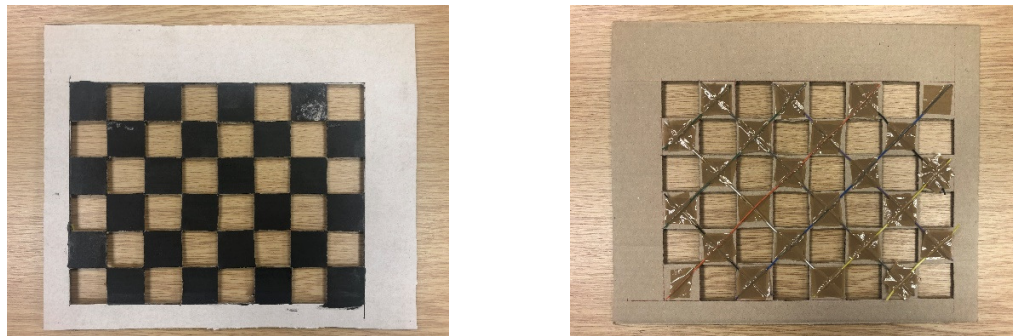


Figure 6-9 Chessboard pattern for the infrared camera: front view (left) and back view (right)

A third calibration pattern, called circle grid pattern, is chosen in the experiment period. It has white circles on a dark background, and it is symmetric. There is a disadvantage to notice that the symmetrical circle grid is not rotation-invariant, but this has been solved as the camera has a fixed position and the pixel reading sequence is always starting from the top-left corner. Figure 6-10 shows a circle grid pattern taken by both cameras.

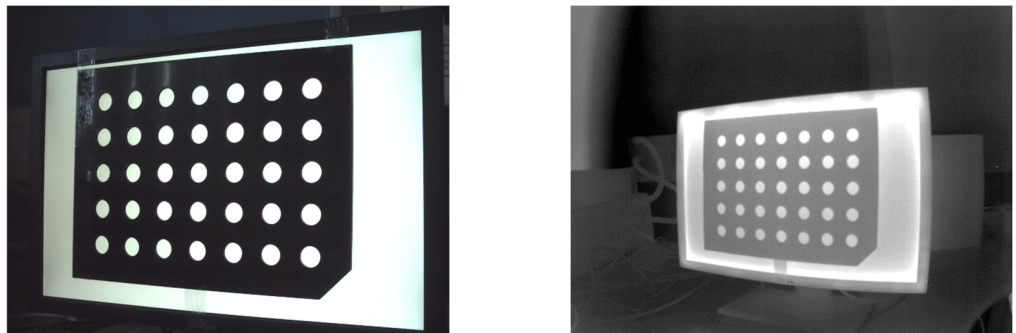


Figure 6-10 A visible image (left) and an infrared image (right) with a circle grid pattern

The 2-D image points are detected from the centres of the circles. Circles chosen as the calibration pattern enables centre detection when the image is not entirely focused, as the infrared camera and the visible camera do not have autofocus function. The circle centre detection can be done more accurately than the chessboard pattern as all pixels on the periphery of a circle can be used, which decreases the effect of image noise. Under camera perspective, circles are imaged as ellipses which can be accounted for by image rectification. Circles are not imaged as perfect ellipses due to unknown lens distortion, but the distortion model present above is considered piecewise linear to minimise the error [80].

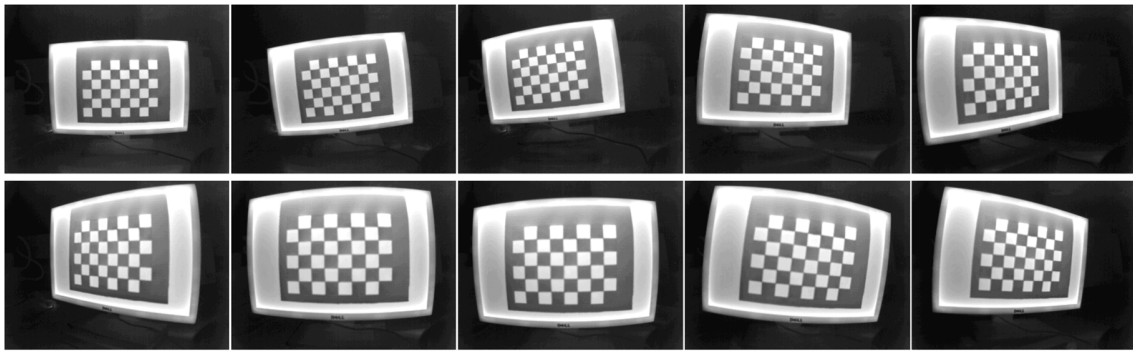
### 6.3.2. Image points and parameters

In the result of camera calibration, there are four intrinsic parameters:  $f_x, f_y, c_x, c_y$ , three radial distortion parameters:  $k_1, k_2, k_3$ , and two tangential distortion parameters:  $t_1, t_2$  needs to be solved. The intrinsic parameters, as well as the extrinsic parameters, are related to the location of the calibration board in space; the distortion parameters are related to how the points on the image being distorted. Thus, these two kinds of parameters are calculated separately. With at least three corner points' or circle centre points' coordinates, which yielding six values are known, the five distortion parameters can be solved. It means that with only one image of the calibration board, the radial distortion and tangential distortion parameters can be calculated. The extrinsic parameters can be computed from the location of the calibration pattern in space. The extrinsic matrix contains three rotation parameters:  $\alpha, \beta, \gamma$ , and three translation parameters:  $T_x, T_y, T_z$ . These parameters are different in each view of the calibration pattern as the calibration board's location is changing. Thus, the intrinsic parameters and extrinsic parameters must be solved for each calibration board's view.

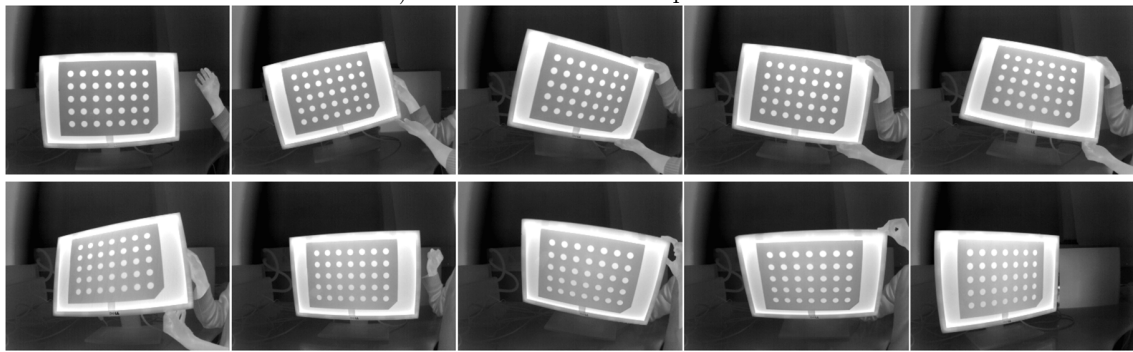
If there are in total  $P$  images, and each image has  $Q$  chessboard corners or circle centres,  $2PQ$  constraints will be provided by the calibration pattern as each point location contain  $x$ -axis value and  $y$ -axis value. The four intrinsic parameters are fixed, and there are  $6n$  extrinsic parameters for all image taken. So, solving these parameters requires that  $2PQ \geq 4 + 6P$  holds. One simple reason for using more than one calibration image is that the chessboard corners and circle centres cannot be detected with infinite precision due to noise and errors, so more points are needed for a more precise camera parameters estimation. Another reason is that for a planar perspective view, four points are enough to express it, and that means a homography can contain at most eight values from these four  $(x, y)$  points. From the image points detected on a calibration pattern, only four image points' worth information can be given, equivalent to  $(4 - 3)P > 1$ . For example, at least two different views of a  $3 \times 3$  calibration pattern is needed to compute the camera parameters. In reality, considering these two reasons and for a more precise camera parameter estimation, at least ten images should be taken for a  $5 \times 6$  calibration pattern. More than 20 images are taken for each calibration pattern to compute a high-quality result in this experiment.

### 6.3.3. Single calibration procedure

The calibration process is carried out on a PC with Python and C++. The client program has been modified for camera calibration. A function is added to the client to save the captured image every 3 seconds and, in total, 30 images. An example of 10 images taken from the visible camera of both calibration pattern is shown in Figure 6-11, and images taken from the infrared camera is shown in Figure 6-12.

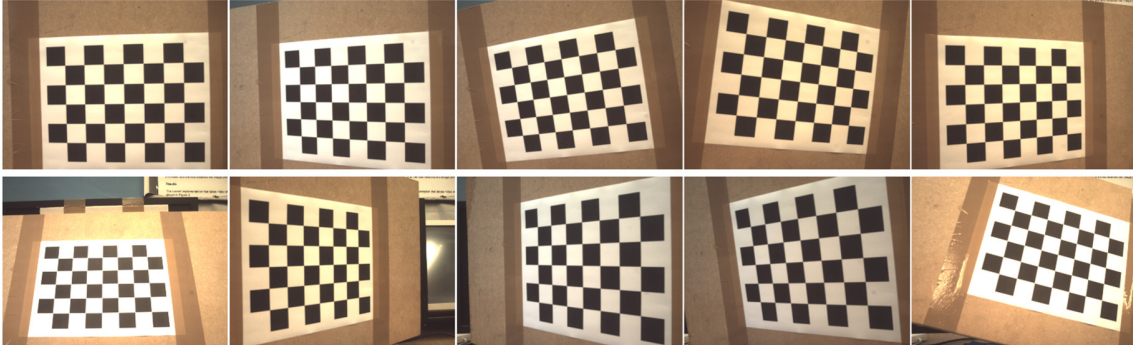


a) chessboard calibration pattern

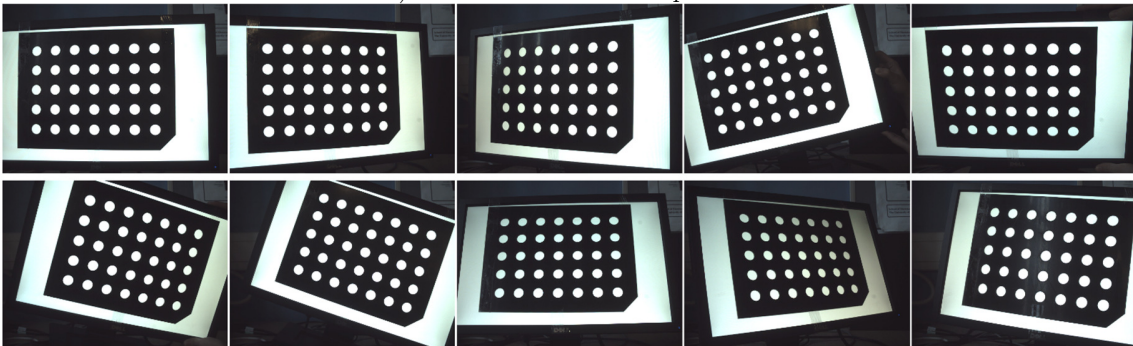


b) circle grid calibration pattern

Figure 6-11 10 infrared images of the calibration patterns



a) chessboard calibration pattern

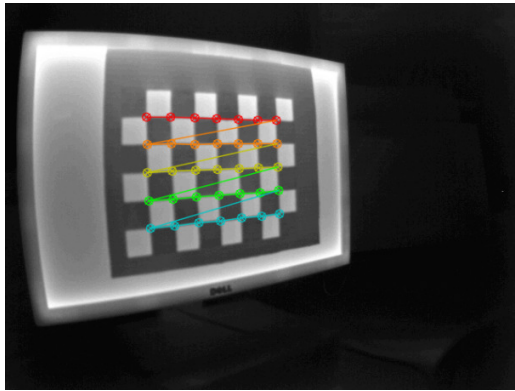


b) circle grid calibration pattern

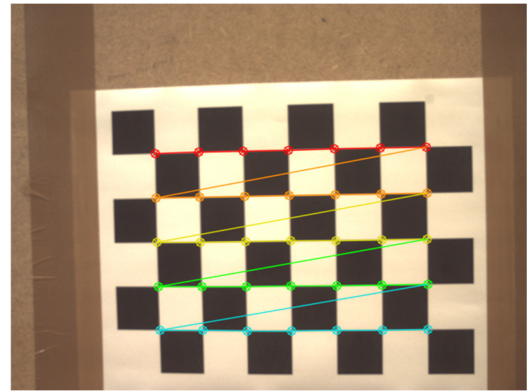
Figure 6-12 10 visible images of the calibration patterns



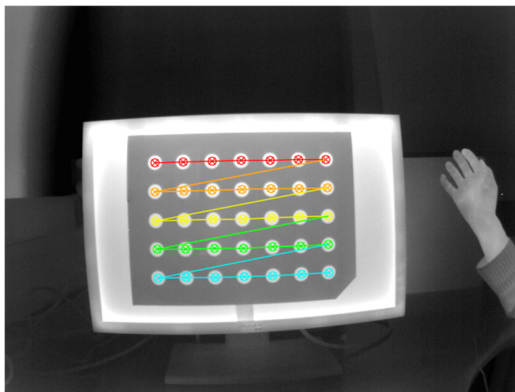
The images of the calibration boards are taken by placing the calibration board at different distance and angles. The calibration board is completely visible in every image, and the calibration area is well lit so that there is no spectral reflection from the environment. The corners of the calibration boards are distributed to all positions in the image. Then the pixel locations of the chessboard corners or the circle centres are detected and extracted to compute a projective transformation between the image points of the 30 different images. Figure 6-13 shows the detected image points among the calibration patterns. Afterwards, it is possible to compute one camera's intrinsic and extrinsic parameters.



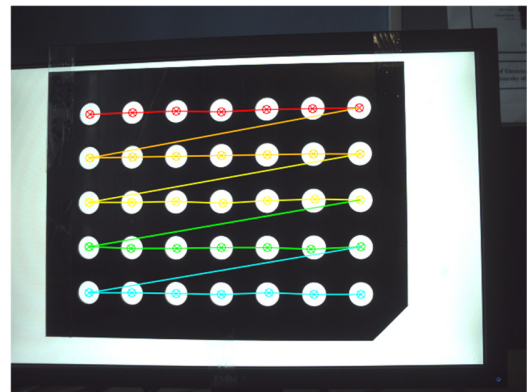
a) infrared image – chessboard pattern



b) visible image – chessboard pattern



c) infrared image – circle grid pattern



d) visible image – circle grid pattern

Figure 6-13 Detected image points

## 6.4. Camera parameter calculation

In the experiment, the camera's focal length  $(f_x, f_y)$  and offsets  $(c_x, c_y)$  are solved based on Zhang's method [61]. It is a self-calibration technique as it only needs the images taken from the camera, which requires calibration. This method relies on the correspondence between the known 3-D world point and the point on the image plane. First, multiple views of a calibration pattern are taken by the camera need calibration. Then, the perspective projection matrix  $\tilde{\mathbf{M}}$  is first estimated using the direct linear transformation method. The camera intrinsic matrix  $\tilde{\mathbf{M}}_i$  is computed using a stack of constraint equations, and the extrinsic matrix  $\tilde{\mathbf{M}}_e$  consist of rotation and translation parameters are computed for each view of the calibration pattern. The distortion parameters are solved based on Brown's technique [81]. It uses the camera's intrinsic parameters calculated earlier as an input to make an assumption of the distortion parameters. After large numbers of iteration, it comes with a more accurate distortion parameter. The camera intrinsic and extrinsic matrix will be recalculated.

### 6.4.1. Intrinsic and extrinsic parameters

From equation (7.2.2), (7.2.3) and (7.2.5), it can be derived that:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = s\mathbf{M}_i \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{T} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (7.4.1)$$

By choosing to define the world plane in Figure 6-1, rather than define a point in the world for all space, equation (7.4.1) can be simplified as  $z = 0$ , and then the homography matrix  $\mathbf{M}$  is changed from 3-by-4 to 3-by-3:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = s\mathbf{M}_i \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{T} \end{bmatrix} \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix} = s\mathbf{M}_i \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{T} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (7.4.2)$$



The PPM  $\mathbf{M}$  can be simplified to a combination of column vectors:  $\mathbf{M} = [m_1 \ m_2 \ m_3]$ , and each  $m$  is a 3-by-1 vector. It can be written out as:

$$\mathbf{M} = [m_1 \ m_2 \ m_3] = s\mathbf{M}_i[\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{T}] \quad (7.4.3)$$

By listing the vector  $m$ :

$$m_1 = s\mathbf{M}_i\mathbf{r}_1, m_2 = s\mathbf{M}_i\mathbf{r}_2 \text{ and } m_3 = s\mathbf{M}_i\mathbf{T} \quad (7.4.4)$$

Thus, the rotation vectors and the translation vector can be derived as:

$$\mathbf{r}_1 = \frac{1}{s}\mathbf{M}_i m_1, \mathbf{r}_2 = \frac{1}{s}\mathbf{M}_i m_2 \text{ and } \mathbf{T} = \frac{1}{s}\mathbf{M}_i m_3 \quad (7.4.5)$$

As the rotation matrix,  $\mathbf{R}$  is an orthogonal matrix, and the columns of an orthogonal matrix are an orthonormal basis, which means each column in the rotation matrix  $\mathbf{R}$  are mutually perpendicular and has a length of one. Then the following equation holds:

$$\mathbf{r}_1^T \mathbf{r}_2 = 0 \text{ and } \|\mathbf{r}_1\| = \|\mathbf{r}_2\| = 1 \quad (7.4.6)$$

By substituting for  $\mathbf{r}_1$  and  $\mathbf{r}_2$  Equation (7.4.6) can be derived from a constraint when calculating the camera parameters:

$$m_1^T (\mathbf{M}_i^{-1})^T \mathbf{M}_i^{-1} m_2 = 0 \text{ and } m_1^T (\mathbf{M}_i^{-1})^T \mathbf{M}_i^{-1} m_1 = m_2^T (\mathbf{M}_i^{-1})^T \mathbf{M}_i^{-1} m_2 \quad (7.4.7)$$

Supposing that  $(\mathbf{M}_i^{-1})^T \mathbf{M}_i^{-1} = \mathbf{M}_i^{-T} \mathbf{M}_i^{-1} = \mathbf{E}$  and the matrix  $\mathbf{E}$  can be written as:

$$\mathbf{E} = \begin{bmatrix} E_{11} & E_{12} & E_{13} \\ E_{21} & E_{22} & E_{23} \\ E_{31} & E_{32} & E_{33} \end{bmatrix} \quad (7.4.8)$$

When taking the intrinsic matrix  $\mathbf{M}_i$  in equation (7.2.5) in the above equation, the matrix  $\mathbf{E}$  has a general closed-form solution:

$$\mathbf{E} = \begin{bmatrix} \frac{1}{f_x^2} & 0 & \frac{-c_x}{f_x^2} \\ 0 & \frac{1}{f_y^2} & \frac{-c_y}{f_y^2} \\ \frac{-c_x}{f_x^2} & \frac{-c_y}{f_y^2} & \frac{c_x^2}{f_x^2} + \frac{c_y^2}{f_y^2} + 1 \end{bmatrix} \quad (7.4.9)$$

With the new symmetric matrix  $\mathbf{E}$ , the constraint equation (7.4.6) can be rewritten as:

$$m_1^T \mathbf{E} m_2 = 0 \text{ and } m_1^T \mathbf{E} m_1 = m_2^T \mathbf{E} m_2 \quad (7.4.10)$$

The element in equation (7.4.10) can be seen generally as  $m_i^T \mathbf{E} m_j$ . Because the matrix  $\mathbf{E}$  is symmetric, it can be expressed by the six elements above or below the matrix diagonal. In the meantime, by expanding the matrix  $\mathbf{E}$  and extract common elements as  $\mathbf{e}$ , it can be written as the dot product of a six-dimensional vector contains rotation element and common element:

$$m_i^T \mathbf{E} m_j = v_{ij}^T \mathbf{e} = \begin{bmatrix} m_{i1}m_{j1} \\ m_{i1}m_{j2} + m_{i2}m_{j1} \\ m_{i2}m_{j2} \\ m_{i3}m_{j1} + m_{i1}m_{j3} \\ m_{i3}m_{j2} + m_{i2}m_{j3} \\ m_{i3}m_{j3} \end{bmatrix}^T \begin{bmatrix} E_{11} \\ E_{12} \\ E_{22} \\ E_{13} \\ E_{23} \\ E_{33} \end{bmatrix}^T \quad (7.4.11)$$

From equation (7.4.11) above and the known homogeneous transformation matrix  $\tilde{\mathbf{M}}$ , all the elements in equation (7.4.11) is computable, and the value to be solved is  $E_{ij}$ . The constraint equation (7.4.7) can now be derived as:

$$\begin{bmatrix} v_{12}^T \\ (v_{11} - v_{22})^T \end{bmatrix} \mathbf{e} = 0 \quad (7.4.12)$$

For each view of the calibration board, two constraints above can be given. If  $P$  images are collected from a different view of the calibration board,  $2P$  equations can be achieved, that is:

$$\begin{bmatrix} (v_{12}^1)^T \\ (v_{11}^1 - v_{22}^1)^T \\ \vdots \\ (v_{12}^P)^T \\ (v_{11}^P - v_{22}^P)^T \end{bmatrix} \mathbf{e} = 0 \quad (7.4.13)$$

by stacking these equations, the following equation can be given:

$$V\mathbf{e} = 0 \quad (7.4.14)$$

Where  $\mathbf{e}$  is a 6-by-1 vector to be solved, and  $V$  is a  $2P$ -by-6 matrix. When  $P = 1$  or  $P = 2$ , due to equation (7.4.13) is a homogeneous linear equation, the solution of  $\mathbf{e}$  cannot be acquired or is not unique. Some of the parameters need to be estimated if less than three images are taken, which increase the difficulty of camera calibration and cause the

inaccuracy of camera parameters. As the value of  $P$  is greater or equal to 3 in this calibration process, the vector  $e = [E_{11}, E_{12}, E_{22}, E_{13}, E_{23}, E_{33}]$  is computable out of the closed-form solution of the matrix  $\mathbf{E}$ :

$$f_x = \sqrt{\frac{1}{sE_{11}}}, f_y = \sqrt{\frac{E_{11}}{s(E_{11}E_{22} - E_{12}^2)}} \quad (7.4.15)$$

$$c_x = -sE_{13}f_x^2, c_y = \frac{E_{12}E_{13} - E_{11}E_{23}}{E_{11}E_{22} - E_{12}^2} \quad (7.4.16)$$

$$s = \frac{1}{E_{33} - \frac{(E_{13}^2 + c_y(E_{12}E_{13} - E_{11}E_{23}))}{E_{11}}} \quad (7.4.17)$$

The extrinsic matrix consists of rotation and translation parameters that can be computed via the condition and constraint of a homography matrix:

$$\mathbf{r}_1 = \frac{1}{s}\mathbf{M}_i^{-1}\mathbf{m}_1, \mathbf{r}_2 = \frac{1}{s}\mathbf{M}_i^{-1}\mathbf{m}_2, \mathbf{r}_3 = \mathbf{r}_1 \times \mathbf{r}_2 \quad (7.4.18)$$

$$\mathbf{T} = \frac{1}{s}\mathbf{M}_i^{-1}\mathbf{m}_3 \quad (7.4.19)$$

The vector  $\mathbf{r}_3$  in equation (7.4.18) is determined from the condition of orthonormality of the rotation matrix  $\mathbf{R}$ .

### 6.4.2. Distortion parameter calculation

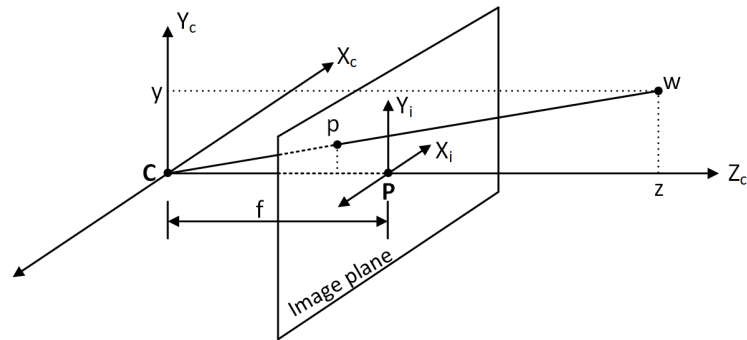


Figure 6-14 Pinhole camera model with a frontal image plane

The derivation in section 6.4.1 does not take the radial and tangential distortion into account. As shown in Figure 6-1, if the pinhole camera model is perfect, an equivalent image plane can be drawn in front of the centre of projection as depicted in Figure 6-14.

Based on the theory of similar triangles, the 3-D world point  $\tilde{\mathbf{w}} = [x \ y \ z]^T$  is mapped to the image plane as a point  $\tilde{\mathbf{p}} = [u \ v]^T$  and the relationship between them can be written as

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} f_x \left( \frac{x}{z} \right) + c_x \\ f_y \left( \frac{x}{z} \right) + c_y \end{bmatrix} \quad (7.4.20)$$

where  $(f_x, f_y)$ , as mentioned previously, is the focal length in pixels, and  $(c_x, c_y)$  is the optical centre in pixels. After substituting the radial distortion equations (7.2.6) and (7.2.7), along with the tangential distortion equation (7.2.11) and (7.2.12), the relationship between a distorted point  $(x_d, y_d)$  and the image point of the perfect pinhole camera model can be determined as:

$$\begin{bmatrix} u \\ v \end{bmatrix} = (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \begin{bmatrix} x_d \\ y_d \end{bmatrix} + \begin{bmatrix} 2t_1 x_d y_d + t_2 (r^2 + 2x_d^2) \\ t_1 (r^2 + 2y_d^2) + 2t_2 x_d y_d \end{bmatrix} \quad (7.4.21)$$

Because the camera's intrinsic parameters  $(f_x, f_y, c_x, c_y)$  is already known from the former calculation, the point  $\tilde{\mathbf{p}} = [u \ v]^T$  on image plane without distortion can then be achieved from equation (7.4.20). By substituting the point  $\tilde{\mathbf{p}}$  into equation (7.4.21) above, the distortion parameters  $(k_1, k_2, k_3, t_1, t_2)$  can be computed as a first estimation but not an accurate value because the intrinsic parameters are calculated under the hypothesis that the camera's pinhole model is perfect.

The linear calibration procedure is carried out with the assumption that there is no lens distortion. To find a better estimation of the distortion centre, an iterative search procedure is performed before carrying the non-linear refinement. First, a regular grid is placed over the image centre point  $P$ , where each grid point corresponds to it. Then the

linear calibration procedure is conducted for each grid points, and the sum of squared reprojection error (SSRE) is calculated. As illustrated in Figure 6-15, the distance in pixels between the re-projected image point and the observed image point is defined as the reprojection error [82]. The grid point with the minimum SSRE takes as the potential image centre point  $P$ . The process is repeated until the change between two potential points  $P$  is less than 0.5 pixel.

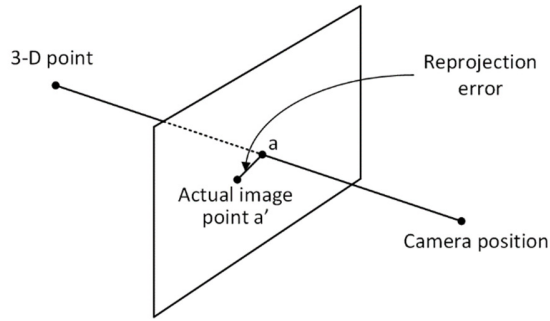


Figure 6-15 Reprojection error on this image is the distance between  $a$  and  $a'$

A non-linear least-squares minimization (Levenberg-Marquardt algorithm) process is introduced to calculate the camera parameters and distortion coefficients. This method uses the camera parameters from the previous step as the initial estimation, where the distortion coefficient is zero. The reprojection errors are calculated by projecting the chessboard corners or the circle grid centres, determined by the calibration pattern, into the image plane coordinates. A comparison is made between the re-projected points and the corresponding detected points. After a few iterations, if the value of the reprojection error is less than one pixel, the result is acceptable, and the final value of camera parameters and distortion coefficients is achieved.

### 6.4.3. Calibration result and analysis

Table 6-1 shows the camera calibration result for images taken from both cameras. Both cameras are set to have a resolution of  $640 \times 480$  pixels for comparison. Table 6-2

concludes that both cameras have a positive radial distortion in their captured images, which implies that wide-angle lenses are installed on them. It is a kind of lens that leads to a wide field of view but also introduce distortions in captured images, and it is also called a fisheye effect [40]. Once the cameras are calibrated, it is possible to project a 3-D point in the physical coordinates to the pixel coordinates. The extrinsic parameters are not constant, as each orient of the calibration pattern contributes to different rotation matrix and translation vectors. Five sets of patterns that contain different images are calibrated separately. The precise value presented in the table below is the mean value of the calibration result, and error variance represents the random variation over time.

Table 6-1 Calibration result of the intrinsic parameters

	focal length (pixel)		optical centre (pixel)	
unit: pixels	$f_x$	$f_y$	$c_x$	$c_y$
infrared camera (chessboard pattern)	569.49 $\pm 0.49$	568.33 $\pm 0.64$	305.10 $\pm 0.15$	217.61 $\pm 0.19$
infrared camera (circle grid pattern)	566.54 $\pm 0.98$	567.63 $\pm 0.78$	313.09 $\pm 0.14$	226.88 $\pm 0.15$
visible camera (chessboard pattern)	865.47 $\pm 0.76$	866.00 $\pm 0.79$	306.27 $\pm 0.21$	243.35 $\pm 0.14$
visible camera (circle grid pattern)	878.88 $\pm 0.98$	879.80 $\pm 0.97$	311.09 $\pm 0.12$	239.32 $\pm 0.15$

Table 6-2 Calibration result of the distortion coefficients ( $\times 10^{-2}$ )

	radial distortion			tangential distortion	
unit: pixels	$k_1$	$k_2$	$k_3$	$t_1$	$t_2$
infrared camera (chessboard pattern)	45.10 $\pm 15.65$	-11.73 $\pm 31.48$	0.02 $\pm 0.01$	0.03 $\pm 0.01$	0.51 $\pm 0.26$
infrared camera (circle grid pattern)	-47.64 $\pm 16.81$	11.57 $\pm 32.79$	0.04 $\pm 0.02$	0.04 $\pm 0.01$	0.97 $\pm 0.34$
visible camera (chessboard pattern)	-19.19 $\pm 5.76$	27.24 $\pm 36.48$	0.07 $\pm 0.02$	0.01 $\pm 0.01$	0.51 $\pm 0.02$
visible camera (circle grid pattern)	-18.17 $\pm 5.98$	-70.81 $\pm 35.61$	0.13 $\pm 0.05$	0.01 $\pm 0.01$	0.63 $\pm 0.02$

It can be deducted from the table above that the infrared camera and visible camera both have radial lens distortion. Although it is not obvious to observe the radial distortion from the calibration image taken from the visible camera, according to the quantified calibration result that the radial distortion exists. The visible camera model has lightly less lens distortion effect than the infrared camera because the value of the first order radial distortion coefficient  $k_1$  of the visible camera is smaller than one-half of which of the infrared camera. From the result that  $k_1$  is negative, which infers that both cameras have barrel distortion, as  $k_1$  remains positive in pincushion distortion [83].

There are several types of lens distortion, but the most common one is radial distortion, and in which barrel distortion is usually seen in modern cameras [84]. Tangential distortion is always neglected and can be compensated for by using distortion centre estimation [85]. Taking tangential distortion into account when calculating the distortion parameters gives a better result on camera calibration [86]. It has been found out from the calibration result that both cameras have a small amount of tangential distortion; this may cause by imprecision when manufacturing.

According to the value of camera parameters and distortion coefficients, the distortion model can be visualized in Figure 6-16. The rectangular mesh grid is displaced due to lens distortion, and the arrow is pointing from its actual position to its distorted position. There are differences in the value of parameters presented in Table 6-2 when calibrating using different calibration pattern, especially the value of  $k_2$ . However, while visualizing the distortion using a mesh grid plot, the displacements of points are not obvious.

The difference should be considered an experimental error here and minimized by capturing more sets of calibration images and doing more iterations when calculating the camera parameters. It can be further minimized by placing the calibration pattern board close to the corner of an image but keep all the pattern in the image, so the image points

along the border of an image can be obtained when calculating the distortion parameters. These points are more distorted than the points located near the image centre and can donate to a more precise calibration result. There is one thing to notice that both cameras' optical centre is not the same as the centre of a  $640 \times 480$  image. The reason for it is that the camera sensor had some columns and rows skipped over before the image digitizing starts, so the optical centre is slightly biasing to the top left corner of the image.

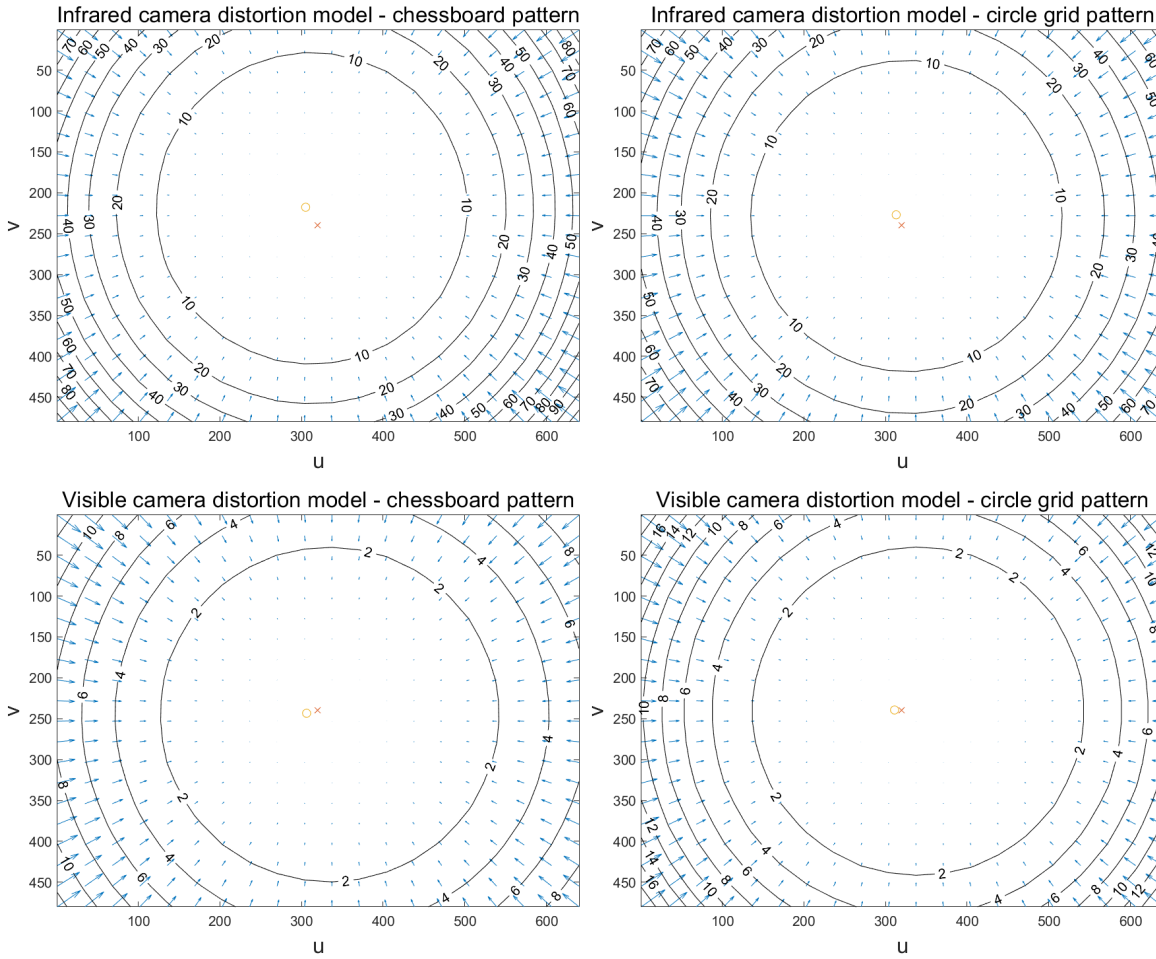


Figure 6-16 Visualized distortion model

## 6.5. Evaluating the accuracy of geometric calibration

To monitor and evaluate the performance of the calibration process, a simulator in



MATLAB is used. The simulation scenario's graphic image is given in as an example set of 10 images in Figure 7-17; each plane in this figure corresponds to the example image shown in Figure 6-12. It is a kind of camera-centric view of the calibration pattern, which means the camera's position remains stationary when capturing the calibration images. The simulator view provides a way to examine each calibration plane roughly; for example, it indicates a calibration error if a calibration plane appears behind the camera model. In this experiment, all the image captured has been put into this simulator to check whether it contains unacceptable errors, for instance, a full calibration pattern cannot be found in some of the images due to reflection of calibration board.

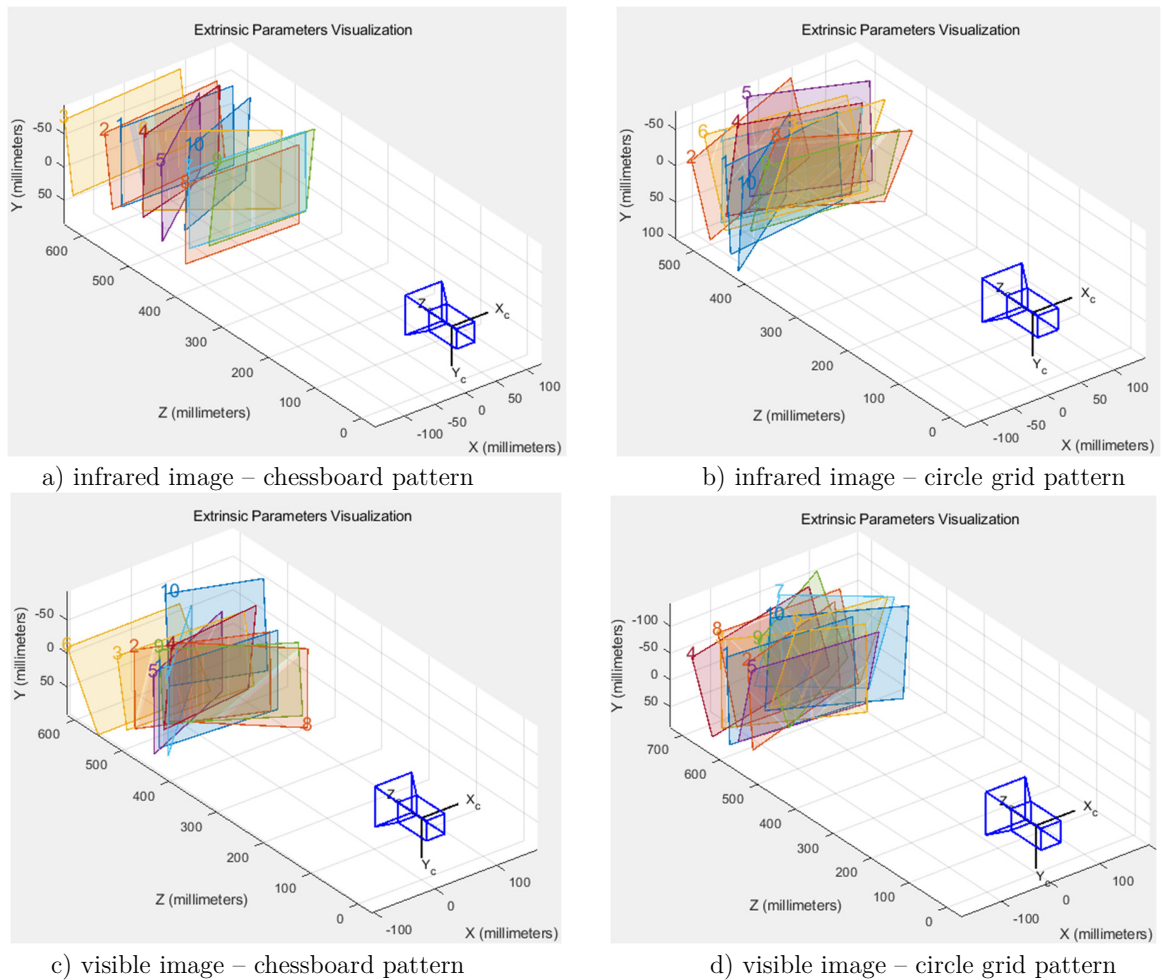


Figure 6-17 10 calibration planes and its relative position of the cameras

### 6.5.1. Reprojection error

Another easy way to judge if the image points are detected correctly is to examine all images in the format shown in Figure 6-13, as the chessboard corners and the circle centres are marked. If a marked point is not in the range, which means it does not meet the expectation, image points should be detected again to ensure correctness.

A qualitative measurement of accuracy is reprojection errors. This value is calculated from a distance between a pattern key points on a calibration image, in this case, is the chessboard corners or the circle centres, and its corresponding world point been projected onto the same image. The root mean square (RMS) reprojection error, for the number of  $P$  images, is calculated between the actual pixel location  $(u, v)$  and its projected ones  $(u_{proj}, v_{proj})$  as follow:

$$ERROR_{reprojection} = \sqrt{\frac{1}{P} \sum_{i=1}^P [(u - u_{proj})^2 + (v - v_{proj})^2]} \quad (7.5.1)$$

Figure 6-18 presents the mean reprojection error of a set of ten images captured from each camera module and in different calibration pattern.

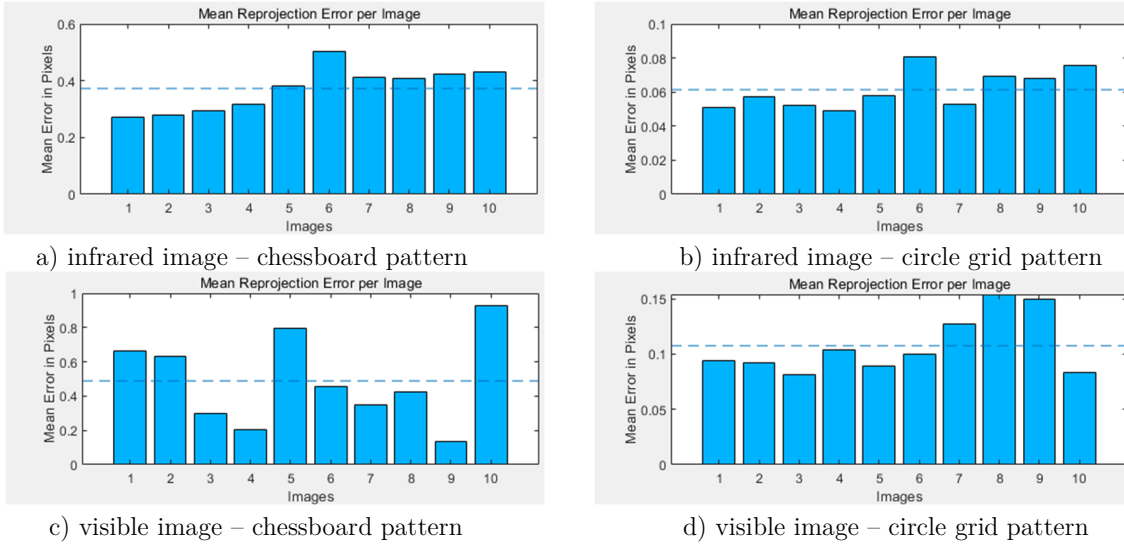


Figure 6-18 Single camera calibration reprojection error with an example of 10 images

Table 6-3 Overall mean reprojection error of single-camera calibration

	infrared image (chessboard)	infrared image (circle grid)	visible image (chessboard)	visible image (circle grid)
reprojection error	0.71	0.31	0.05	0.10

As shown from the mean reprojection error table above, the infrared camera calibration using the chessboard calibration pattern has the most significant mean reprojection error of **0.71** pixels, whereas the visible camera calibration session using the same calibration pattern has the overall mean reprojection error of 0.31 pixels. The first reason for the mean reprojection error variation is because the infrared camera is not auto-focused, which causes low resolution of the chessboard calibration pattern infrared image, especially the corners and edges of the chessboard. The second reason is that the radiation from the heated pad from the back would obscure every square's edge in the checkerboard, which affects the smoothness of the edges. The last but not least reason is that due to making the chessboard calibration board, there has to be a connection between the squares, which leads to some confusion on detecting the chessboard corners. By changing the calibration pattern to the circle grid, the reprojection error is reduced, representing more accuracy in the result of camera parameters and distortion coefficients.

On the contrary, as the circle grid calibration board is made from an acrylic plate instead of printed on paper, the thickness of the board cannot be neglected and can be seen on the visible image. This fact leads to the border of the circle is more inaccurate than the printed version, which explains that the mean reprojection error of the chessboard pattern is smaller than the circle grid calibration pattern.

### 6.5.2. Manufacturing error

During the calibration session, the tangential distortion had been found in the infrared

camera and the visible camera. Supposing that there is a tilt angle  $\theta$  between the camera lens and the image plane, as previously mentioned in section 6.2.2, the tilt angle can be calculated as

$$\theta = \arcsin\left(\frac{-t_1}{\sqrt{t_1^2 + t_2^2}}\right) \quad (7.5.2)$$

The calibration result base on the circle grid calibration pattern should be used in the equation above because it has minor reprojection errors, which allow a more precise estimation of the tilt angle. From the calculation, the tilt angle inside the infrared camera is  $\theta_{infrared} \approx 2.86^\circ$ , and the tilt angle inside the visible camera is  $\theta_{visible} \approx 1.01^\circ$ . The tilt angles are insignificant in both cameras and demonstrate the imperfection of parallelism between the lens and the imaging plane. As the value of  $t_1$  and  $t_2$  are very small in both cameras, and there should be experimental errors when calibrating the cameras. The tilt angle may be smaller or larger than the value calculated here.

### 6.5.3. Comparing calibration result with camera datasheet

An additional approach to review the calibration result's accuracy is to compare it with the manufacture datasheet. A proper estimation of the camera parameters should be highly close or even equal to the parameters presented in the camera's datasheet. Two standard camera parameters are compared with the datasheet in this work. One is focal length, and another in the field of view. Table 6-4 displays both cameras' specifications from the datasheet that will be used in further calculations.

Table 6-4 Specification of cameras

	Sensor width (mm)	Sensor height (mm)	Image width (pixel)	Image height (pixel)
Infrared camera	10.88	8.16	640	480
Visible camera	5.70	4.28	640	480

In optics, the focal length is the distance between the image plane and the projection centre [87]. The calibration result of the camera presents the focal length in  $x$  and  $y$  optical axis, which are defined as  $f_x$  and  $f_y$  respectively, and the unit of which is in pixels. There are two values for focal length in the calculation because technically, the imager is not square but rectangular, and each pixel is also rectangular. To convert the focal length  $(f_x, f_y)$  in pixel to the physical focal length in millimetre  $f_{mm}$ , they have to be multiplied by the size of a single imager element. The following equation applied:

$$f_{mm} = f_x \times \frac{\text{sensor width (mm)}}{\text{image width (mm)}} = f_y \times \frac{\text{sensor height (mm)}}{\text{image height (mm)}} \quad (7.5.3)$$

The term field of view refers to an angle over where the camera sensor spans. Similar to focal length, it contains two directional elements: horizontal field of view  $FOV_h$  and vertical field of view  $FOV_v$ . The equations for the field of view is given as:

$$FOV_h = 2 \arctan\left(\frac{\text{image width}}{2f_x}\right) \quad (7.5.4)$$

$$FOV_v = 2 \arctan\left(\frac{\text{image height}}{2f_y}\right) \quad (7.5.5)$$

Table 6-5 Camera parameter comparison between datasheet and calibration result

	$f$ (mm)	$f_x$ (mm)	$f_y$ (mm)	$FOV_h$	$FOV_v$
Infrared camera (datasheet)	9.6			63.1°	49.62°
Infrared camera (calibration)		9.63 ±0.05	9.65 ±0.05	58.91°	48.83°
Visible camera (datasheet)	7.82			45°	34.52°
Visible camera (calibration)		7.82 ±0.07	7.85 ±0.08	40.01°	30.51°

As the circle grid calibration pattern provides a smaller mean reprojection error for the infrared image, and the chessboard calibration method has a smaller mean

reprojection error for the visible camera, the camera parameters are calculated using these corresponding experimental results. Table 6-5 compares the camera parameters calculated from the equations above with the specifications from the camera's datasheet. It must be admitted that there is some variation between the two kinds of parameters. Based on the common pinhole camera model, there may be influenced coming from the principle point displacement, and the ignorance of some higher-order radial distortion parameters. Furthermore, the calibration method implemented in this work only ensures accuracy at a sub-pixel level. One thing to be noticed is that the actual field of view is all narrower than the ideal value since the actual value is calculated from the undistorted image, and the edge of an image is cropped by removing the distortion effect. From the datasheet, one can only get a single value of the focal length. However, two focal length values can be obtained from single-camera calibration, and they are focal lengths in the x-axis and y-axis.

## 6.6. Rectify lens distortion

From the previous example images taken from both cameras, a concern has been raised because a straight line in the real world cannot maintain straight in the images. The purpose of this embedded vision system includes continuous monitoring and information acquisition. However, some of the information is incorrect due to lens distortion of cameras, such as using a distorted image to estimate the length of an interesting object. It could be noticed from Figure 6-6 that neglecting distortion produces a considerable error if a measurement wanted to be taken of the cable trunking. Moreover, Figure 6-12 a) and b) shows that the infrared camera is subjected to high barrel distortion, and the information without considering distortions is vitally different from the accurate information, which in some cases, the infrared image may make no sense. In this

experiment, the lens distortion is rectified by applying the intrinsic and extrinsic camera parameters and the distortion coefficient calculated earlier to obtain a corrected image.

Generally, there are two ways to rectify lens distortion in images. One way is to use the obtained camera intrinsic matrix and the distortion coefficients to perform an inverse calculation for each image. Another way, which is used in this experiment, is an evolution of the previous method. A look-up table (LUT) is generated base on the intrinsic parameters and the distortion coefficients. The LUT contains the information of each pixel's new location, as the format of coordinates, in the undistorted image. Then pixels in the original images are remapped into an undistorted image according to its corresponding coordinates in the LUT.

The look-up table contains a new location for each pixel, and it is an undistortion and rectification transformation map. The necessary parameters to calculate the map are the camera's intrinsic parameters: focal length  $(f_x, f_y)$  and optical centre  $(c_x, c_y)$ , rotation matrix  $\mathbf{R}$ , and distortion coefficients  $(k_1, k_2, k_3, t_1, t_2)$ . The inverse mapping algorithm is used to calculate the look-up map. Supposing that the point  $(u, v)$  is a pixel on the destination corrected and rectified image, the corresponding points on the camera plane  $(X, Y)$  can be expressed as:

$$X = \frac{u - c_x}{f_x}, Y = \frac{v - c_y}{f_y} \quad (7.6.1)$$

The 3-D world point then can be written as:

$$[x \ y \ z]^T = R^{-1}[X \ Y \ 1]^T \quad (7.6.2)$$

According to Figure 6-14, if this world point is re-projected to the image plane as a new point  $(x', y')$ , then:

$$x' = \frac{x}{w}, y' = \frac{y}{w} \quad (7.6.3)$$

And with distortion parameters applied, the undistorted image point  $(x'', y'')$  can be

written as:

$$x'' = (1 + k_1 r^2 + k_2 r^4 + k_3 r^6)x' + 2t_1 x' y' + t_2 (r^2 + 2x'^2) \quad (7.6.4)$$

$$y'' = (1 + k_1 r^2 + k_2 r^4 + k_3 r^6)y' + t_1 (r^2 + 2y'^2) + 2t_2 x' y' \quad (7.6.5)$$

Finally, the look-up map at  $x$  direction  $map_x(u, v)$  and at  $y$  direction  $map_y(u, v)$  is given as:

$$map_x(u, v) \leftarrow x'' f_x + c_x \quad (7.6.6)$$

$$map_y(u, v) \leftarrow y'' f_y + c_y \quad (7.6.7)$$

A geometrical transformation is applied to the source distorted image  $src(x, y)$ , and the source image is transformed to the destinate undistorted image  $dst(u, v)$  refer to the specified lookup map. The following equation indicates where any particular pixel is to be relocated:

$$dst(u, v) = src(map_x(x, y), map_y(x, y)) \quad (7.6.8)$$

The remapping function takes a pixel from one place in the image and maps it to another place so that there will be blank pixels left in one location if to which no pixel at all is mapped. This is a typical problem in image reconstruction and is generally referred to as forwarding projection problems [88]. It is necessary to make an interpolation from pixels to avoid blank pixels. The nearest fraction pixels will be interpolated for each pixel in the destination undistorted image that remains blank and is found to need a source pixel to fill in, the nearest fraction pixels will be interpolated, and the correct value for the destination pixel will be derived. The bilinear interpolation method is used to solve this problem. That is, for a pixel at location  $(u, v)$  which remains blank, and the nearest integer location next to it is  $(u_1, v_1)$ ,  $(u_1, v_2)$ ,  $(u_2, v_1)$  and  $(u_2, v_2)$ , the value of this pixel  $f(u, v)$  is derived as follow:

$$f(u, v_1) \approx \frac{u_2 - u}{u_2 - u_1} f(u_1, v_1) + \frac{u - u_1}{u_2 - u_1} f(u_2, v_1)$$

$$f(u, v_2) \approx \frac{u_2 - u}{u_2 - u_1} f(u_1, v_2) + \frac{u - u_1}{u_2 - u_1} f(u_2, v_2)$$

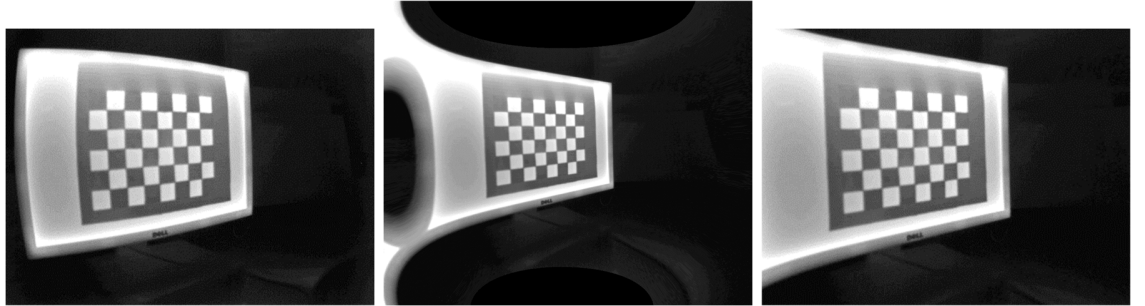


then

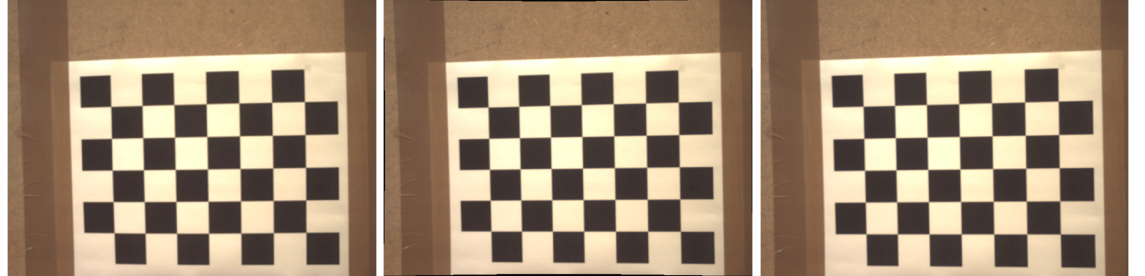
$$f(u, v) \approx \frac{v_2 - v}{v_2 - v_1} f(u, v_1) + \frac{v - v_1}{v_2 - v_1} f(u, v_2) \quad (7.6.9)$$

## 6.7. Experimental result

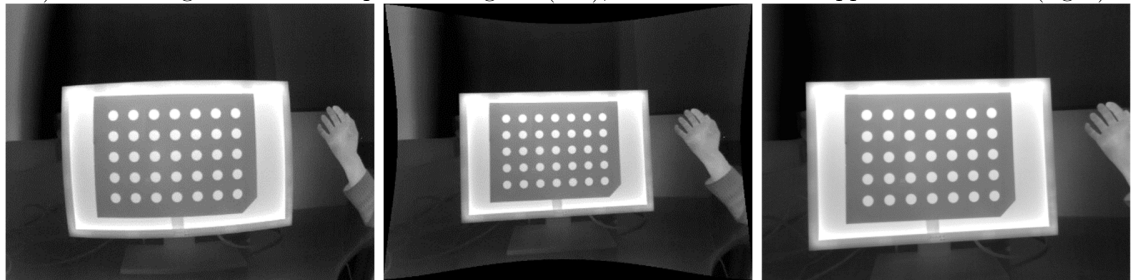
After applying the image rectification function, the original infrared image and visible image are rectified. Examples of the undistorted images compared to their original one in Figure 7-13, taken directly from cameras, are shown in Figure 6-19.



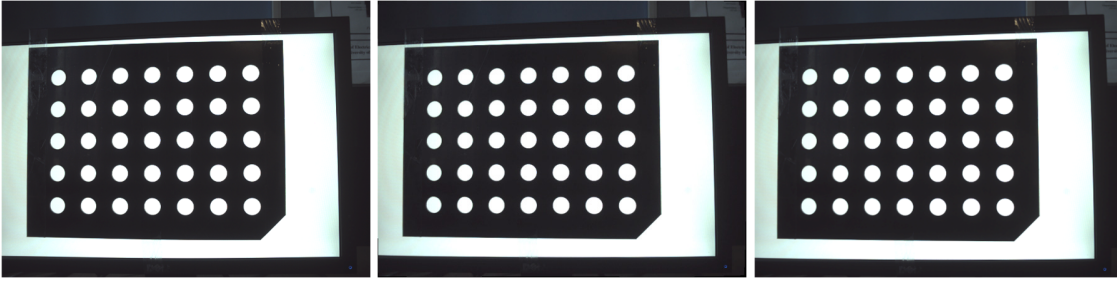
a) infrared image – chessboard pattern original (left), undistorted and cropped undistorted (right)



b) visible image – chessboard pattern original (left), undistorted and cropped undistorted (right)



c) infrared image – circle grid pattern original (left), undistorted and cropped undistorted (right)



d) visible image – circle grid pattern original (left), undistorted and cropped undistorted (right)

Figure 6-19 Undistorted image comparing to the original image

As can be seen from the undistorted images, the distorted lines in those images become straight. The images are rectified with a very low reprojection error, and the images are looking more realistic, and image pixels are now correctly related to the 3-D position in the world. It can be observed from the result above that there are black borders in the image; this is caused by the remapping algorithm as the barrel distortion rectification stretch the outlier of the image. The image can be cropped to  $640 \times 480$  to get rid of the black borders. From the effect of undistorted images and its reprojection error, a conclusion can be drawn that using circle grid calibration pattern to calibrate the infrared camera will achieve a more satisfying result. There is a small difference in reprojection error for the visible camera with less lens distortion, which makes the chessboard and circle grid calibration pattern both practicable.

The image rectification function is implemented in three different platforms. Firstly, the infrared camera and visible camera are calibrated on PC, and the server-client program is involved in supplying the real-time stream. Secondly, the LUT maps are written into a text file and stored on the embedded Linux system running on the ARM processor. The remap function is integrated with the server program. When the server program starts running, it automatically reads the map and implements the image remapping algorithm. The rectified image then be transferred to the PC for display using the server-client program. Finally, an OpenCL kernel is designed, and it contains the

remap function. The OpenCL host program parses the images and the LUT maps as the kernel arguments.

Table 6-6 Frame rate of producing an undistorted image in different platforms

	Intel Core i7 CPU	ARM A9 processor	FPGA OpenCL device
Average frame time (ms)	65.31	88.11	48.54
Frame rate (frame/s)	15.33	11.35	20.60

As mentioned above in section 5.3.4, it is time-consuming when the server program retrieving images from DDR3 memory, the optimal frame rate in Table 6-6 is calculated by just using image files as an input. After neglecting the image transfer time in a system, it is perceptible in the table above that the highest frame rate is achieved by implementing the remapping function on an OpenCL device. The frame rate is 1.48 times the same algorithm running on a PC and 1.81 times on an ARM A9 processor. It can be concluded that implementing the remap function in OpenCL platforms will give a much smoother image, and it looks more lifelike with a higher frame rate.

## 6.8. Summary

It has been found out in this chapter that the infrared camera and the visible camera has lens distortion. At the beginning of the chapter, the importance of camera calibration is declared, while performing a rectification function on the distorted image allows the image captured from the infrared and the visible camera looks more realistic and provides more practical information. The calibration method and the mathematics behind it are illustrated in detail. The two cameras are calibrated using the chessboard pattern and the circle grid pattern; the camera's parameters and distortion coefficients are obtained. The circle grid calibration pattern will be selected in Chapter 7 to create infrared and

visible image overlay, as this pattern provides both calibrations with lower reprojection error. The infrared images and visible images are undistorted using the distortion coefficients achieved. The image rectification process is implemented in different platforms, which gives evidence that the remap algorithm running on the OpenCL platform facilitated a higher frame rate.

## Chapter 7

# Infrared and visible image matching methods

The matching of the infrared and the visible image is an essential task for image analysis. The infrared image's targeted region often contains temperature information and can distinguish an object from its background based on the radiation difference [89]. The visible image contains texture detail of the field of view consistent with the human eye visual system. If overlaying an infrared image and a visible image onto one image, the fused image preserves the information from both images, with the advantage that the infrared images usually have a particular target's thermal information, which cannot be found in visible images [90]. For example, in an environment cooled for a period, the visible image can show the background in great detail, but the infrared image is less detailed because of the low thermal contrast in the field of view. If a target object is

suddenly camouflaged in the scene under this condition, the visible camera cannot detect it in the visible band. However, this object can be detected in the infrared band due to its temperature being warmer or colder than the environment temperature [91]. One method attempted to match two images is to obtain camera parameters from stereo camera calibration and rectify the stereo image pairs. Another method tried to align images is a Fast Fourier Transform (FFT) based image registration. At the end of this chapter, a discussion is given to compare the stereo rectification result and the image registration result.

## **7.1. Stereo system geometry**

This approach aims to use stereo vision to create a novel matching image consisting of infrared and visible band information. The prerequisite of overlaying two authentic scene images is that they are part of the overlapping field of view. In this work, a system containing an infrared camera and a visible camera is observing the same scene from different positions, and the system is also defined as a stereo camera system [92]. Multiple images are captured by the stereo vision cameras of the same object, and distances from the stereo system are known. The two views from separate cameras can be combined using a triangulation of the distance from the object and the distance between the two cameras to form a combinational image containing information from each band [93].

### **7.1.1. Dual-camera stereo system overview**

The system setup combines the DE1-SoC development board, an infrared camera, and a visible camera. As presented in section 2.4, the box enclosure made the relative position of these two cameras fixed, which is an essential condition to perform the stereo camera calibration [94]. Figure 2-1 reveals the hardware setup of the dual-camera system. The

infrared camera model XTM-640 is hanging above the visible camera model TRDB-D5M. The lenses centres are fixed on the same vertical line, allowing that the cameras' image planes are coplanar with each other. In this case, the infrared camera refers to the up camera, and the visible camera refers to the down camera. The visible camera is fixed with the processor board and the acrylic box, while the infrared camera can be rotated independently along its pitch axis. The optical axes of the two cameras are set to be parallel and are 400mm apart. They are mounted to aim in the same direction, so the cameras' viewing field has an overlapping partition. The two camera lenses are closed to each other in order to minimize the difference in the field of view, allowing more information to be contained in the overlaid image. The physical alignment of the two camera makes the mathematical transformation easier to control [95]. The two cameras' settings and positioning form a stereo vision system. The concept of disparity will be introduced to understand how a stereo vision system operates.

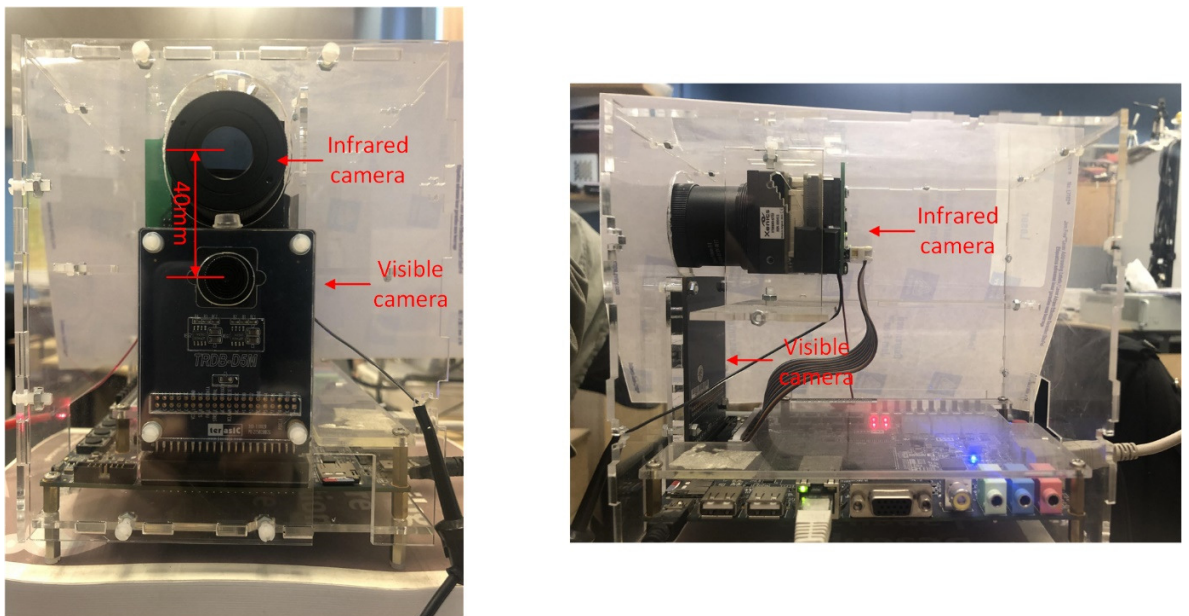


Figure 7-1 Heterogeneous dual-camera system front view (left) and side view (right)

Consider the settings of the two identical cameras without lens distortion effect are

perfectly aligned. One is placed on top of the other, as presented in Figure 7-2. The image planes are precisely coplanar with each other. The two cameras' optical axes are correctly paralleled, and the distance between the sensor centres are known. The optical centres  $c_y^{up}$  and  $c_y^{down}$  are calibrated so the image coordinates are the same in up and down images. Most importantly, the two cameras' captured images have corresponding columns that are located on the same line. Assuming that the two cameras view the same direction, the captured images demonstrate the difference in viewing the same object.

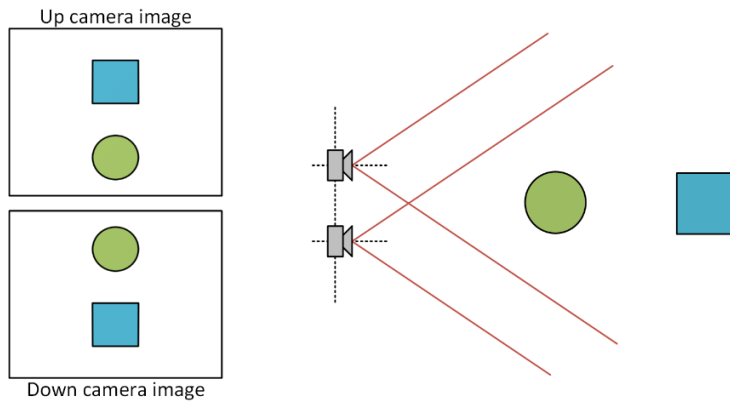


Figure 7-2 Stereo vision system operation

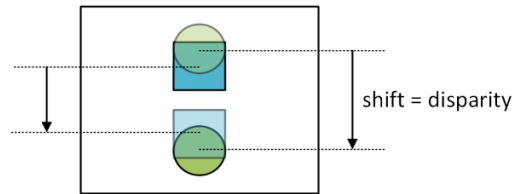


Figure 7-3 Disparity in stereo vision system

As presented in Figure 7-3, the disparity is the shift difference between a point in the captured image of the up camera and its counterpart in the captured image on the down camera. The disparity is usually observable in a stereo camera system, and the farther an object is from the cameras, the smaller the disparity will be [96]. It is the same meaning as the depth of an object in the field of view decreases as the disparity increases. In this study, the dual-camera system is built intended for long-range observation such



as surveillance or environment monitoring, so the target objects' disparity should be expected small.

The calculation of disparity can be obtained from a simple model of the pinhole cameras. Figure 7-4 represents a view from the up and down cameras with the optical centre  $O_R$  and  $O_T$ , where the subscript  $R$  and  $T$  stands for 'reference camera' and 'target camera', respectively. The distance between the two optical centres is defined as baseline  $T$ . An assumption has been made here that both cameras have the same focal length  $f$ . The bold red line indicates the frontal image plane; the vertical points  $p_r$  and  $p_t$  are respectively the projection of the real-world point  $P$  onto the reference image and the target image. The disparity can be simply represented as  $d = x_R - x_T$ .

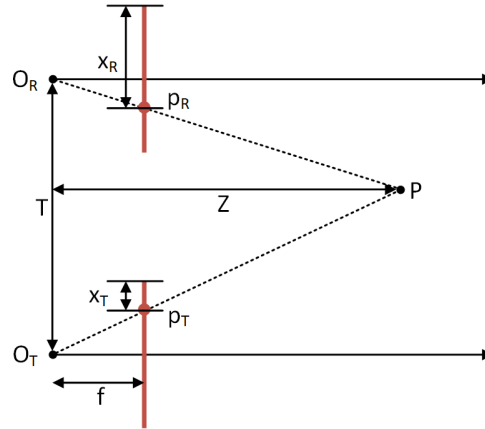


Figure 7-4 Ideal disparity triangulation from two perfectly aligned cameras

By using the similar triangles, the depth  $Z$  of the world point  $P$  can be obtained by:

$$\frac{T}{Z} = \frac{T - (x_R - x_T)}{Z - f} \quad \rightarrow \quad Z = \frac{Tf}{x_R - x_T} \quad (8.1.1)$$

Equation (8.1.1) further illustrates the nonlinear relationship between the disparity and the depth; the depth is inversely proportional to the disparity. Figure 7-5 indicates that when an object's distance is further away from the stereo vision cameras, the disparity difference does not change much.

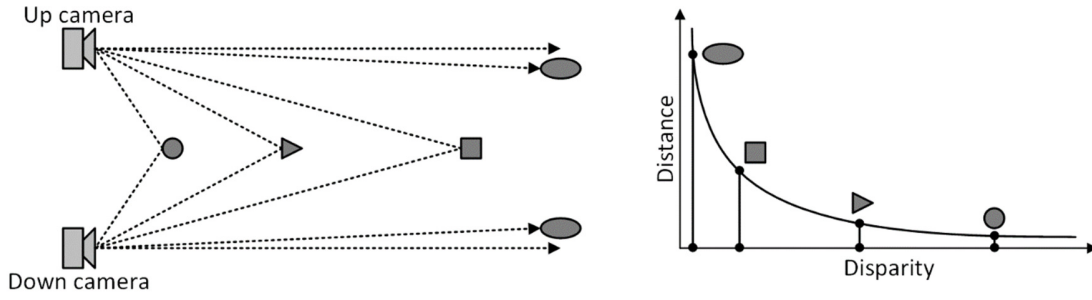


Figure 7-5 The relationship between disparity and depth

The stereo vision system consists of an infrared camera and a visible camera is aimed to distinguish warm objects metres away. The infrared image often reflects the temperature of the objects' surface, and the infrared camera cannot reveal the texture information. Assuming that there is no obstruction in front of the target objects in the field of view, in order to overlay two images whose columns are on the same line, the algorithm proposed here is to translate the reference image by the disparity, so the infrared information of the target object would cover the texture information. For an object with a known distance of  $Z$ , the shift distance of the reference image is equal to disparity and can be written from equation (8.1.1) that:

$$d = x_R - x_T = \frac{Tf}{Z} \quad (8.1.2)$$

Knowing all above, the main problem of overlapping the images from a stereo vision system is thus to be able to obtain the pixels that correspond to each other in the same column. This step is called stereo calibration or stereo matching and will be demonstrated in section 7.2.

### 7.1.2. Epipolar geometry

The ideal arrangement for two cameras to form a stereo vision system requires two cameras: identical, distortion-free, and perfectly aligned. In this case, the image captured

from two cameras would have corresponding pixels located on the same vertical line. However, it is impossible in an experiment that the two cameras can be aligned precisely, and it usually ends up in a structure similar to the one shown in Figure 7-6. This experiment tries its best to align two cameras at least approximately so that the stereo overlap area can be increased in the resulting images. Moreover, the cameras are tilted slightly in towards each other to ensure that the two principal rays intersect at a certain distance. The inward verging stereo cameras resulted in an offset in the y-axis and subtracted from the disparity.

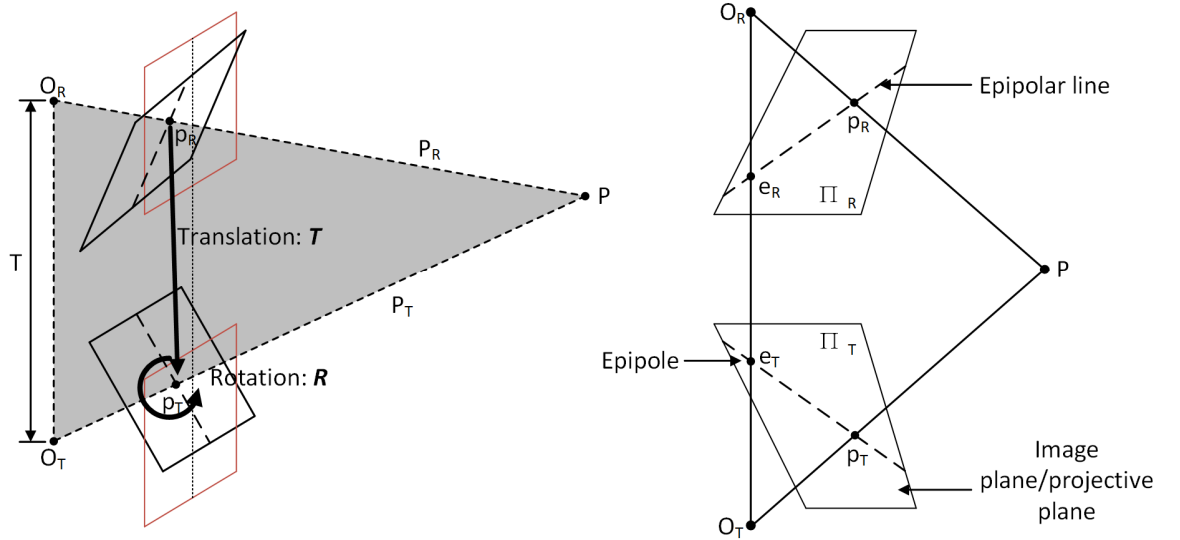


Figure 7-6 Practical situation of a stereo vision system (left) and epipolar geometry (right)

The epipolar geometry is the definition of the basic geometry of a stereo vision system [97]. It is formed of two pinhole camera models. The intersection points of the line joining the two optical centres with the image plane are defined as the epipoles  $e_R$  and  $e_T$ . The epipole of the reference camera  $e_R$  can be seen as its image of the target camera's optical centre  $e_T$ . The epipolar geometry in Figure 7-6 also shows that the cameras' centre of projection  $O_R$  and  $O_T$ , and a pair of image planes  $\Pi_R$  and  $\Pi_T$ . A physical world point  $P$  is projected onto both image planes and are labelled as  $p_R$  and  $p_T$ . An epipole plane is

then created by the physical world point  $P$  and two optical centres, and it is coloured grey in Figure 7-6. The lines  $p_R e_R$  and  $p_T e_T$ , which connect the points of projection to its epipolar point, are defined as the epipolar lines, which is also the intersection of the epipolar plane with the projective plane.

Supposing there is only one camera, the distance from the camera to the physical world  $P$  cannot be determined. Considering the target camera only, the point  $P$  could be located anywhere along the entire line defined by  $p_T$  (the projection of  $P$  onto  $\Pi_T$ ) and  $O_T$  (the optical centre of the target camera), and obviously, there are lots of other points located on the line  $p_T O_T$ . If projecting the line  $p_T O_T$  onto the reference image plane  $\Pi_R$ , the resulting line is the epipolar line  $p_R e_R$  of the reference plane.

The epipolar constraint defines that when a feature is projected onto one image, the matching view of it in the other image lies along the corresponding epipolar line, which indicates that if the epipolar geometry of the stereo camera system is known, the 2-D searching for an identical feature in two stereo image turns into 1-D searching along the epipolar lines [98]. This method could save time on computing and avoiding all other points that may result in spurious correspondences.

### 7.1.3. The matrices for stereo calibration

Two matrices need to be introduced for stereo calibration, one is the essential matrix  $E$ , and the other is the fundamental matrix  $F$ . The essential matrix  $E$  consist of rotation and translation information of the two cameras in the physical world, the translation  $T$  and rotation  $R$  describe the location of the reference camera relative to the first camera in the 3-D world coordinates. From previous section 6.2.1, the homography matrix  $M$  is introduced and it contains similar information as the essential matrix  $E$ . However, they are two totally different matrices as the matrix  $M$  defines the relationship between one

point in the plane viewed by a single camera and the point on the camera plane. The matrix  $E$  defines the relationship between the point in one image plane and the line in another image plane, which means it relates  $p_R$  to  $p_T$ , and it has nothing to do with the imagers. The fundamental matrix  $F$  consist of the cameras' intrinsic information, which presents the relationship of the points in the reference image coordinates to the points on the target image coordinates.

### **The essential matrix $E$**

The definition of the essential matrix  $E$  can be derived from the relation which connects the projected position  $p_R$  and  $p_T$  of the point  $P$  on the two image planes. The physical location of  $p_R$  and  $p_T$  can be related by using the epipolar geometry, which was introduced previously. In order to perform the calculation in one coordinate, the reference camera's coordinates centred on  $O_R$  is chosen; the observed point is located on  $P_R$  and the optical centre of the target camera is  $T$  in these coordinates. In the target camera's coordinates, the observed point of  $P$  is located at  $P_T$ , with the rotation matrix  $R$ ,  $P_T$  can be expressed as:

$$P_T = R(P_R - T) \quad (8.1.3)$$

which can be rewritten as

$$(P_R - T) = R^{-1}P_T \quad (8.1.4)$$

Then the epipolar plane should be derived as it contains and relates all of these points and lines. A conventional method to constrain all the point  $\mathbf{x}$  on a plane  $\Pi$  with its normal vector  $\mathbf{n}$ , and the plane passing through a point  $\mathbf{a}$  is

$$(\mathbf{x} - \mathbf{a}) \cdot \mathbf{n} = 0 \quad (8.1.5)$$

There are two vectors in the epipolar plane that are already known, which are  $P_R$  and  $T$ . A vector who perpendicular to both points can be expressed as  $P_R \times T$ , as the cross product of two vectors, resulting in a third vector orthogonal to the two [99], and the normal vector  $\mathbf{n}$  can be recovered from it. The dot product with matrix multiplication

in equation (8.1.4) can then be replaced by the transpose of the normal vector  $\mathbf{n}$ . Hence, to present all potential point  $P_R$  through the target camera centre  $T$ , the following equation can be written:

$$(P_R - T)^T (T \times P_R) = 0 \quad (8.1.6)$$

To associate the points  $p_R$  with  $p_T$ ,  $P_R$  and  $P_T$  should be related first. By substituting equation (8.1.3) into equation (8.1.6) and using  $\mathbf{R}^T = \mathbf{R}^{-1}$  yields:

$$(\mathbf{R}^T P_T)(T \times P_R) = 0 \quad (8.1.7)$$

The cross product of  $T$  and  $P_R$  can be edited as a matrix multiplication if a new matrix  $G$  is defined:

$$T \times P_R = GP_R \quad (8.1.8)$$

where

$$G = \begin{pmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{pmatrix} \quad (8.1.9)$$

Substituting this for the cross product of  $T$  and  $P_R$ , which gives:

$$(P_T)^T \mathbf{R} G P_R = 0 \quad (8.1.10)$$

In the meantime, the essential matrix  $E$  is defined by the product of  $\mathbf{R}$  and  $G$ . So, the equation above can be rewritten as

$$(P_T)^T E P_R = 0 \quad (8.1.11)$$

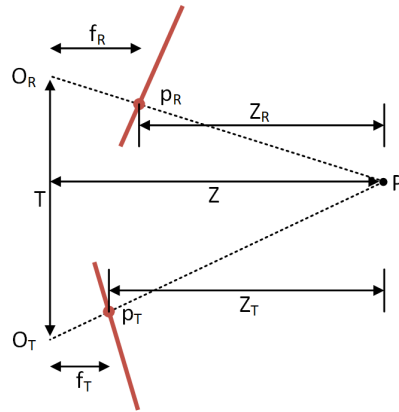


Figure 7-7 Practical disparity triangulation from stereo cameras

Figure 7-7 depicts the stereo camera system and its disparity view in reality, where  $f_R$  and  $f_T$  is the focal length of the reference camera and target camera, respectively,  $Z_R$  and  $Z_T$  is the depth of the physical point  $P$  from two imagers. From Figure 7-7, the projection equation can be derived as

$$p_R = f_R \frac{P_R}{Z_R} \text{ and } p_T = f_T \frac{P_T}{Z_T} \quad (8.1.12)$$

By replacing  $P_T$  and  $P_R$  from equation (8.1.12) to equation (8.1.11) and divide both sides of the equation by  $Z_R Z_T / f_R f_T$ , yields:

$$p_R^T E p_T = 0 \quad (8.1.13)$$

However, the essential matrix  $E$  is proven to be a rank-deficient matrix [100]. It is a 3-by-3 matrix but has less than three nonzero eigenvalues. This leads to the uncertainty of a structure specified by the essential matrix, and the structure does not have a unique solution. The rank of the essential matrix  $E$  is 2, so the equation (8.1.13) turns out to be an equation for a line formed by all the points that comply with the equation.

There are seven constraints in the essential matrix  $E$ , and it declares the relationship between points to each other cameras in world coordinates or camera coordinates. In which there are three rotation parameters and two translation parameters. The two translation parameters indicate the direction of translation. There are two additional constraints in the essential matrix. One is specified as the determinant, which has a value of 0 because  $E$  is a rank-deficient matrix. Another constraint is two equivalent nonzero singular parameters, as the matrix  $G$  is skew-symmetric and  $R$  is a 3-by-3 rotation matrix.

### **The fundamental matrix $F$**

The definition of the fundamental matrix  $F$  is determined by a point in one image and its corresponding epipolar line in another image. For a point  $p$  in the pixel coordinates,  $q$  is defined to be its corresponding point on the camera plane, and the relation between them is mentioned previously in section 6.2.1 that  $q = M_i p$  or  $p =$

$\mathbf{M}_i^{-1}\mathbf{q}$ , where  $\mathbf{M}_i$  is the camera's intrinsic matrix. Thus, the equation (8.1.13) which contains the essential matrix  $E$  is rewritten as:

$$\mathbf{q}_T^T (\mathbf{M}_{i_T}^{-1})^T E \mathbf{M}_{i_R}^{-1} \mathbf{q}_R = 0 \quad (8.1.14)$$

Furthermore, the fundamental matrix is derived from the above as:

$$\mathbf{F} = (\mathbf{M}_{i_T}^{-1})^T E \mathbf{M}_{i_R}^{-1} \quad (8.1.15)$$

So, equation (8.1.14) is able to be simplified as:

$$\mathbf{q}_T^T \mathbf{F} \mathbf{q}_R = 0 \quad (8.1.16)$$

Similar to the essential matrix  $E$ , the fundamental matrix  $F$  is also of rank 2. Apart from that, the fundamental matrix represents the relationship in pixel coordinates, whereas the essential matrix indicates the relationship of two cameras in a real 3-D world. The equation (8.1.14) and (8.1.16) correlates the essential matrix to the fundamental matrix. If the lens distortion effect is removed and all the points on images are normalized by timing  $1/f$ , the camera's intrinsic matrix  $\mathbf{M}_i$  turns into an identity matrix, and the equation  $F = E$  stands.

The same as the calculation for the camera's intrinsic matrix in section 6.4.1, the fundamental matrix can be computed with the adoption of the known correspondences. The calculation of  $F$  is performed in MATLAB and Python; it can be solved directly without calibrating separate cameras in the stereo vision system.

## 7.2. Stereo calibration

The method to calibrate the stereo camera system is to calibrate each camera separately at first, and this process is conducted and explained in Chapter 6. The single-camera calibration procedure does not aim to infer stereo information, and the result of this procedure is to remove the radial and tangential image distortion mathematically and to reproject distorted image points to their actual position. Then the stereo calibration is



responsible for establishing the geometric relationship between two different cameras in physical space. The rotation and translation parameters are considered between the same feature view in up and down cameras, such as the calibration pattern in the same position taken by separate cameras.

### **7.2.1. Stereo calibration procedure**

In common with the single-camera calibration, the stereo calibration can be performed using Python and some of the functions in OpenCV libraries. The client program is changed base on the requirement of stereo calibration, allowing the client to save one visible image immediately after saving one infrared image. In such way, the two images would have a stable view of the same direction and can avoid sudden changes in the scene.

There are several existing stereo calibration techniques; this project implemented the most used method developed by Zhang [61]. The stereo cameras observe the 2-D calibration pattern, and the cameras' intrinsic matrix and distortion coefficient is retrieved. In the meantime, the physical world position of each frame is obtained from the calibration pattern points. The cameras form a stereo rig and observe the calibration pattern at the same time; from the relative 3-D points of the image points in its respective frames, it is then possible to achieve the rotation and translation between the two cameras.

One conclusion drawn from Chapter 6 is that the circle grid calibration pattern is better than the chessboard pattern as the circle grid pattern shows minor reprojection error, so the circle grid calibration pattern is chosen for stereo calibration in this section. The first stage of stereo calibration is to capture a series of stereo image pairs of the circle grid calibration pattern. Thirty image pairs with various orientations of the

calibration pattern are taken for more precise results in this experiment. Figure 7-8 presents the stereo image pairs captured by the two cameras; they have been rectified to remove the lens distortion effect. The same as the single-camera calibration procedure, each of these images then contributes to the algorithms that detect the circle grid's centre and calculation for camera parameters.

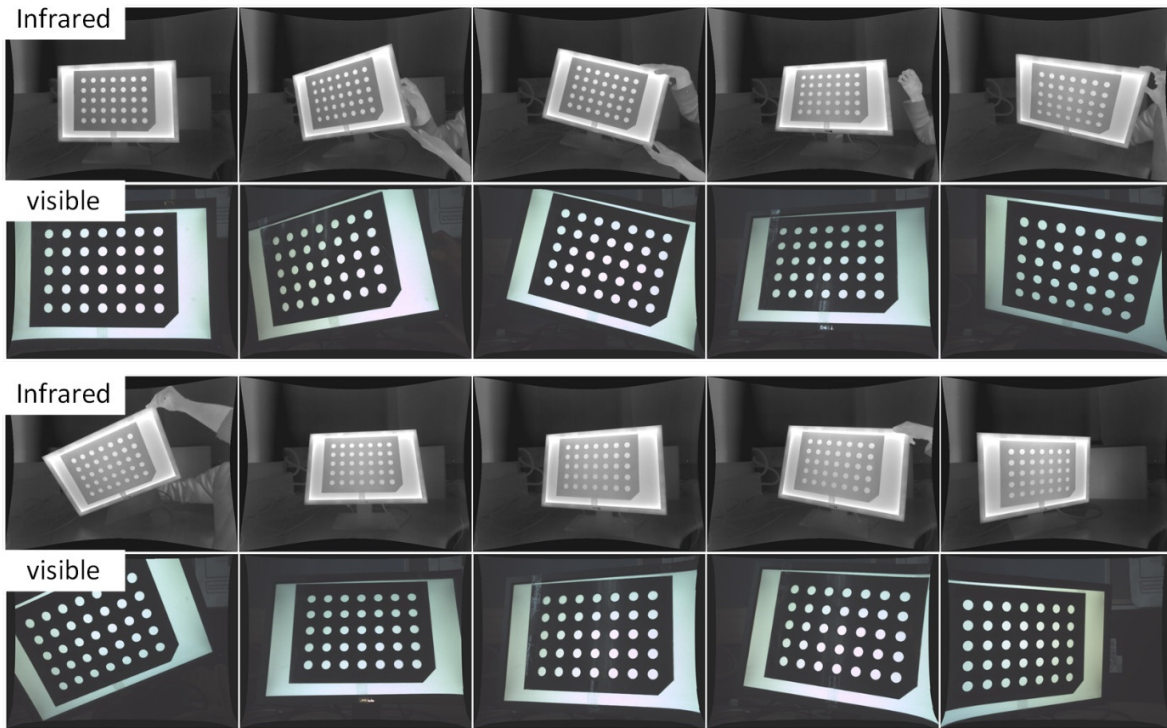


Figure 7-8 10 examples of stereo image pair of a circle grid pattern

Once the circle centre positions are known on the image plane, the coordinates of these points in the physical world can also be obtained as each circle's size and the spacing between the circles are known. The pose of each calibration pattern can be achieved to demonstrate how it is rotated and displaced. A 2-D diagram can be drawn to simulate the 3-D effect of each calibration pattern. As shown in Figure 7-9, a 3-D coordinate axis is drawn on the circle grid's first centre with  $x$ -axis in blue,  $y$ -axis in green and  $z$ -axis in red. This pose estimation can be a rough approach to check if the

rotation matrix and the translation vector is correct.

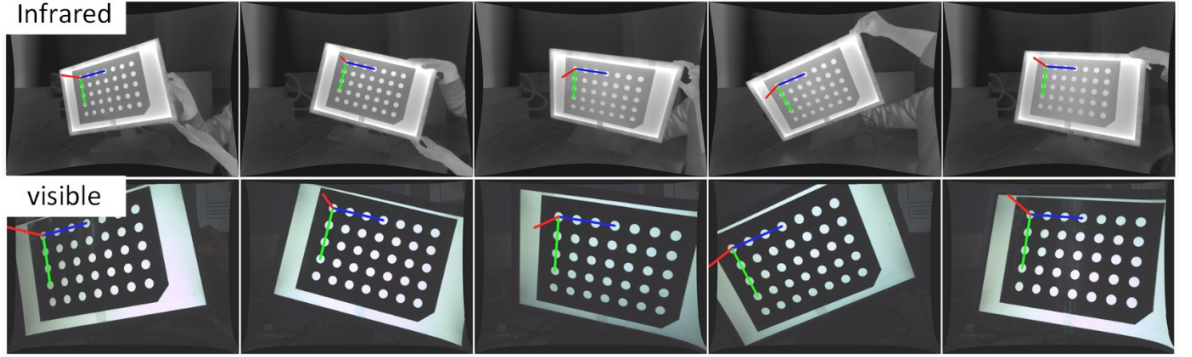


Figure 7-9 5 examples of pose estimation of the infrared (up) and visible (down) calibration pattern

### 7.2.2. Rotation and translation parameter calculation

As mentioned earlier, the purpose of stereo calibration is to seek the rotation matrix  $\mathbf{R}$  and the translation vector  $\mathbf{T}$  between the reference camera and the target camera. All these parameters must rectify stereo image pairs before attempting to work out the stereo matching problem. In the meantime, these parameters can be used to re-project a 3-D world point based on disparity, as illustrated in previous sections. Assume that there is a point  $P$  in the physical coordinates, the transformation of it to the reference and target camera coordinate can be correspondingly written as

$$P_R = \mathbf{R}_R P + \mathbf{T}_R \text{ and } P_T = \mathbf{R}_T P + \mathbf{T}_T \quad (8.2.1)$$

where  $\mathbf{R}_R$  and  $\mathbf{T}_R$  represent the rotation matrix and translation vector from the camera to the point  $P$  for the reference camera, whereas  $\mathbf{R}_T$  and  $\mathbf{T}_T$  donate to the target camera correspondingly. As noted from Figure 7-6 that the physical point  $P$  viewed by the stereo camera rigs are donated as point  $P_R$  and point  $P_T$  on the camera's coordinates, they have the relationship that:

$$P_R = \mathbf{R}^T (P_T - \mathbf{T}) \quad (8.2.2)$$

By considering the equation (8.2.1) and (8.2.2) together, the rotation and translation

between the reference camera and the target camera can be solved as:

$$\mathbf{R} = \mathbf{R}_T(\mathbf{R}_R)^T \text{ and } \mathbf{T} = \mathbf{T}_T - \mathbf{R}\mathbf{T}_R \quad (8.2.3)$$

One thing to be noticed is that the reference and target camera's subscription can be reversed in the equations above. From the single calibration procedure explained in Chapter 6, the rotation matrix and translation vector are solved for each calibration pattern view and each camera separately. These parameters solutions are then adopted by equation (8.2.3) to figure out the rotation matrix  $\mathbf{R}$  and translation vector  $\mathbf{T}$  between the reference camera and the target camera. The major experimental error created by the stereo calibration procedure is due to image noise and rounding errors. As the rotation matrix and the translation matrix can be calculated from a single calibration pattern, each solution is slightly different. The median value of the solution is taken for  $\mathbf{R}$  and  $\mathbf{T}$  parameters, and the reprojection error of the image points for both cameras is calculated based on a robust Levenberg-Marquardt iterative algorithm with this initial approximation.

### 7.2.3. Stereo calibration result and accuracy evaluation

The geometric parameters are determined from stereo camera calibration. As the location of the image points from the calibration pattern are obtained and adopted into the stereo calibration algorithm, the rotation and translation parameters along with the essential and fundamental matrix can be computed. The following parameters represent the stereo calibration result after the distortion effect been removed:

**Essential matrix:**

$$E = \begin{bmatrix} -1.41 \pm 0.51 & 19.39 \pm 5.98 & -105.78 \pm 16.48 \\ -19.35 \pm 0.71 & 0.23 \pm 0.17 & 7.71 \pm 0.84 \\ 105.67 \pm 1.01 & -8.63 \pm 0.68 & -1.63 \pm 0.79 \end{bmatrix} \times 10^{-2}$$

**Fundamental matrix:**

$$F = \begin{bmatrix} (-1.80 \pm 0.56) \times 10^{-7} & (5.17 \pm 1.65) \times 10^{-5} & (-5.93 \pm 0.12) \times 10^{-2} \\ (-5.1 \pm 2.08) \times 10^{-5} & (5.52 \pm 2.00) \times 10^{-7} & (1.56 \pm 0.16) \times 10^{-2} \\ (8.42 \pm 0.87) \times 10^{-2} & (-1.54 \pm 0.15) \times 10^{-2} & -8.94 \pm 1.71 \end{bmatrix}$$

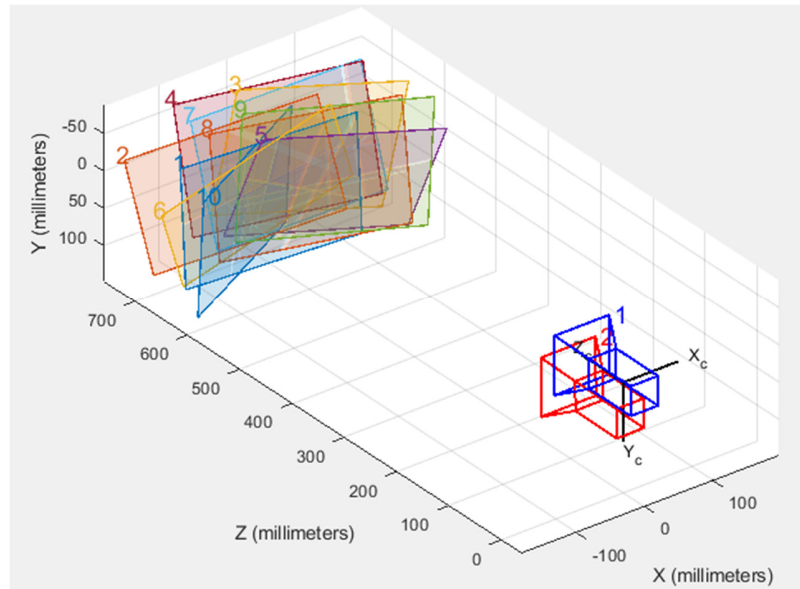
**Rotation matrix:**

$$R = \begin{bmatrix} 99.98 \pm 0.03 & 1.15 \pm 0.20 & 1.54 \pm 0.14 \\ 1.49 \pm 0.71 & 99.99 \pm 0.02 & 0.09 \pm 0.90 \\ 1.54 \pm 0.14 & 0.07 \pm 0.05 & 99.98 \pm 0.03 \end{bmatrix} \times 10^{-2}$$

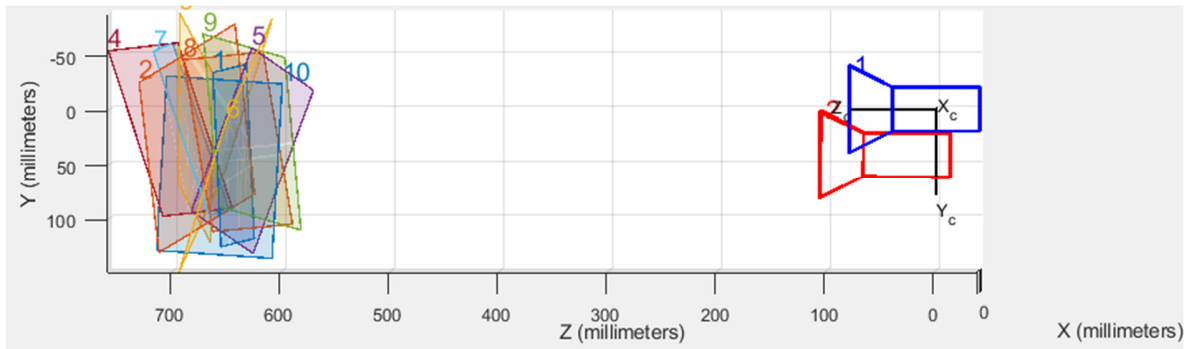
**Translation vector:**

$$T = [-7.41 \pm 0.78 \quad -105.78 \pm 1.73 \quad -19.47 \pm 0.96] \times 10^{-2}$$

One thing to be noticed is that the rotation matrix and translation vector above demonstrate the position and orientation of the reference camera's image plane being homogeneous transformed into the target camera's image plane.



a) 3-D view



b) front view

Figure 7-10 10 calibration image pairs and its relative position of the stereo cameras

Figure 7-10 depicts the stereo camera settings and ten stereo image pairs' simulation scenario. Each coloured image plane in this figure represents one stereo image pair. As can be seen from the cameras' relevant position, they are set to be as aligned as possible. Moreover, the calibration board is set to be 650mm away from the stereo system so that the disparity in the stereo image pairs could be calculated from equation (8.1.2).

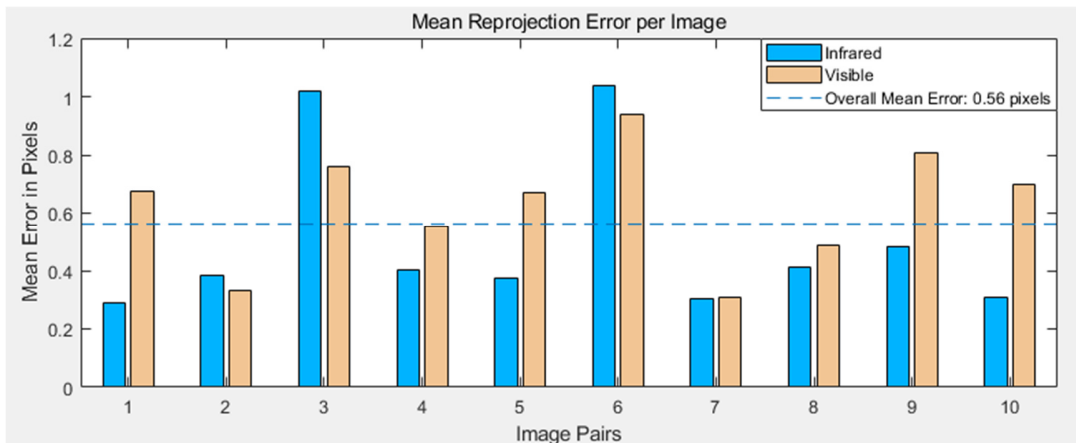


Figure 7-11 Stereo camera calibration reprojection error with an example of 10 images

The same as single-camera calibration, the reprojection error represents the distance between the detected and reprojected points in pixels. The reprojection will be accepted if the reprojection error is less than one pixel. Figure 7-11 shows the reprojection error in stereo image pairs, and the overall mean projection, in this case, is 0.56 pixels. This

experiment has taken 70 stereo image pairs to compute the rotation and translation parameters, and the mean projection error of these 140 images is 0.45 pixels.

#### 7.2.4. Stereo rectification

The stereo rectification of a stereo camera system, which has an up and down camera setup, is to rectify the image captured by each camera, allowing every corresponding point on the image to be located on the same column coordinates and make two image planes column aligned. Once the rotation matrix and translation vector ( $\mathbf{R}, \mathbf{T}$ ) or the fundamental matrix  $\mathbf{F}$  is acquired, the result can be used to rectify stereo image pairs. The returned solution of rotation matrix  $\mathbf{R}$  allows the target camera to be tuned in the same plane with the reference camera, which composes the two image planes coplanar but not column-aligned, and the solution of translation vector  $\mathbf{T}$  can create column-alignment between two stereo images. After the rectification process, one can assume that the two cameras' optical rays are parallel, and they will never intersect. The red planes marked in Figure 7-6 demonstrate the aim of stereo rectification: the two cameras' image planes should be located on the same plane.

The same points in two different images of the same object are defined as stereo correspondence [101]. To achieve a more reliable stereo correspondence, the two images' columns should be aligned after stereo rectification. To reduce computational work load, the searching for corresponding columns in two images can be altered by searching only one row in one image and its matching point in the other image. The epipoles on the two image planes will change their position to infinity if the two images are perfectly column-aligned. Moreover, the epipolar lines are positioned along with the image columns, and the camera scan lines are identical across the two images. Consider that countless possible image planes could be the actual image plane on one optical ray; more constraints are

necessary. A practical constraint, and also this experiment aims to minimize the disparities between the two stereo images. As mentioned earlier, the rotation and translation parameter  $(\mathbf{R}, \mathbf{T})$  or the fundamental matrix  $\mathbf{F}$  can both do the job to rectify stereo image pairs. In this experiment, Bouguet's algorithm, which adopts the rotation matrix and the translation vector obtained from two calibrated cameras, will be used to acquire maximum overlay area and minimum distortion.

Bouguet's method is maximizing the overlapping area of the two images, as well as trying to figure the homography matrix to map the epipoles to infinity. It can be achieved by matching the points detected in the calibration pattern between the stereo image pairs. The intrinsic parameters of the two cameras used in this work are already obtained from single-camera calibration. Then, between any two matching views of the calibration board, the rotation matrix  $\mathbf{R}$  and translation  $\mathbf{T}$  can be computed. To accomplish the requirement to make the most of the image area, the rotation matrix  $\mathbf{R}$  is split into half between the reference frame and the target frame, and the two resulting rotation matrices are defined as  $\mathbf{r}_R$  and  $\mathbf{r}_T$ , respectively. To put the two cameras into coplanar, each camera rotates half of the whole rotation  $\mathbf{R}$ , and their optical axis after rotation is parallel to the vector sum of their original optical axis. However, the rotation can only make two image planes coplanar, and the image pair is not column aligned after the rotation.

Supposing that there is a rotation matrix  $\mathbf{R}_{rect}$  makes the reference camera's epipole to infinity and set the epipolar line horizontally. To calculate this rotation matrix, a vector  $\mathbf{e}_1$  pointing from the epipole  $\mathbf{e}_R$  to the optical centre  $(c_{x_R}, c_{y_R})$  of the reference frame should be set. The vector  $\mathbf{e}_1$  can be written as:

$$\mathbf{e}_R = \frac{\mathbf{T}}{\|\mathbf{T}\|} \quad (8.2.4)$$

It is a unit normalized direction vector, and it is aligned along the translation vector between the centres of two image planes. Then a vector  $\mathbf{e}_2$  orthogonal to  $\mathbf{e}_1$  should be



created, and it is along with the frame plane and orthogonal to the optical axis. The second unit normalized vector  $e_2$  can be built by the cross product of  $e_1$  with the direction of the optical axis:

$$e_2 = \frac{[-T_y \quad T_x \quad 0]^T}{\sqrt{T_x^2 + T_y^2}} \quad (8.2.5)$$

The third unit vector  $e_3$  should be created and is orthogonal to both  $e_1$  and  $e_2$ , this is accomplished by using the cross product that:

$$e_3 = e_1 \times e_2 \quad (8.2.6)$$

The rotation matrix  $R_{rect}$  that transform the epipole to infinity can be then derived as:

$$R_{rect} = \begin{bmatrix} (e_1)^T \\ (e_2)^T \\ (e_3)^T \end{bmatrix} \quad (8.2.7)$$

The reference frame's epipolar line should be horizontal, and the epipoles should be at infinity after applying this rotation matrix to rotate the reference image around the centre of the projection. To reach the column alignment of the two cameras, the rectification rotation matrix  $R_R$  and  $R_T$  can be calculated from:

$$R_R = R_{rect} r_R \text{ and } R_T = R_{rect} r_T \quad (8.2.8)$$

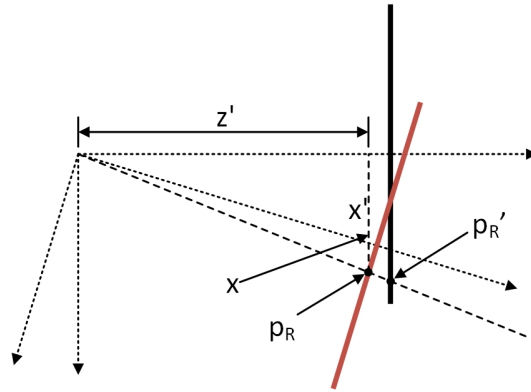


Figure 7-12 Relationship between a point on the original image plane and on the rectified image plane

The same as the rectification method mentioned in section 6.6, maps are created to remap all the pixels from the infrared image and the visible image. Before that, the relationship between the original pixel and its rectified one should be illustrated. The projection of a point onto the original image plane and its corresponding rectified point are presented in Figure 7-12.

In this figure, the point  $\mathbf{p}_R = [x, y, f]^T$  represent a point in the initial camera plane, the point  $\mathbf{p}_R'$  is the expression of  $\mathbf{p}_R$  in the reference camera of the rectified camera plane, which aligned with the baseline. Using the rotation matrix  $\mathbf{R}_{rect}$  to compute:

$$\mathbf{R}_{rect}\mathbf{p}_R = [x', y', z']^T \quad (8.2.9)$$

and the coordinates of the corresponding rectified point  $\mathbf{p}_R'$  can be calculated based on the similar triangle rule as:

$$\mathbf{p}_R' = \frac{f}{z'}[x', y', z']^T = [x'', y'', z'']^T \quad (8.2.10)$$

The pixel coordinates  $(u, v)$  can then be retrieved by adopting the reference camera's intrinsic parameters:

$$\begin{aligned} u &= f_{x_R} \frac{x''}{z''} + c_{x_R} = f_{x_R} \frac{f x' 1}{z' f} + c_{x_R} = \frac{f_{x_R} x'}{z'} + c_{x_R} \\ v &= f_{y_R} \frac{y''}{z''} + c_{y_R} = f_{y_R} \frac{f y' 1}{z' f} + c_{y_R} = \frac{f_{y_R} y'}{z'} + c_{y_R} \end{aligned} \quad (8.2.11)$$

Similarly, for each point  $\mathbf{p}_T$  on the target camera plane, compute:

$$\mathbf{R}\mathbf{R}_{rect}\mathbf{p}_T = [x', y', z']^T \quad (8.2.12)$$

where the rotation matrix  $\mathbf{R}$  obtained from stereo camera calibration between the two unaligned cameras. Similarly, the remapping points of the target image plane can be obtained. Then the target image plane can be aligned with the new reference image plane.

The remapping process just described maps the pixel from its initial image plane to the location on the rectified image plane. As the lens distortion effect is already removed, the new image is built by filling all the pixel values from the original image. Moreover,

the detailed mathematics of remapping is previously discussed and can be found in section 6.6.

Using this method to rectify stereo image pairs can be demonstrated in the left image of Figure 7-6. The frames with a black border show the original stereo images, and the frames with a red border show the image after rectification. In this case, the ideal stereo rectification will tune the two stereo images into coplanar and column aligned. A new image pair is produced will have the maximal height and width of the common area.

The advantage of Bouguet's algorithm is that it minimizes the image reprojection distortion by minimizing the difference reprojection when producing the rectified image pairs. Compared to the traditional method that only rotates and translates the reference frame to match the target frame, the disadvantage is that Bouguet's method increases computing workload as there are two frames that need to be rectified simultaneously.

### **7.2.5. Stereo rectification result**

To virtually align the cameras of an up-down stereo vision system, the lens distortion effect should be removed at first, and the stereo images pairs have to be rectified so that the corresponding pixels will be positioned on the same column in the two rectified image pairs. Moreover, the depth of an object can be correctly recovered from the stereo image pairs.

With the two cameras intrinsic matrix, extrinsic matrix, and distortion coefficient already known and the transformation between the reference camera and the target camera been computed, the stereo image pairs can be rectified. The flow chart in Figure 7-13 demonstrate the steps of it.

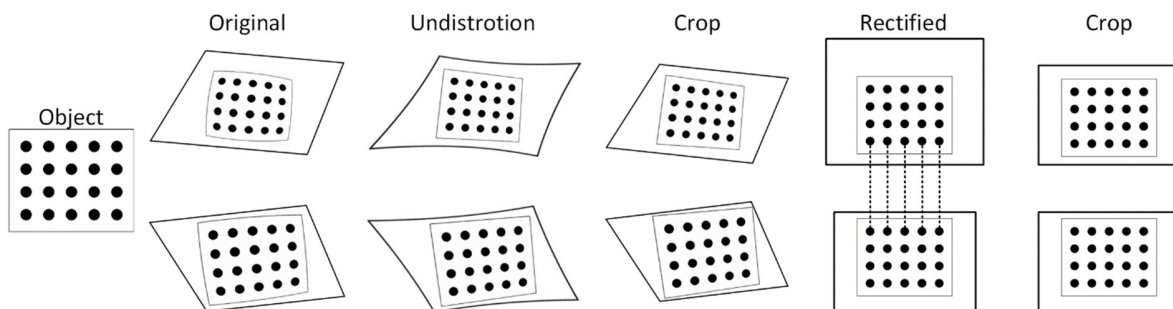


Figure 7-13 Procedure of stereo image pair rectification

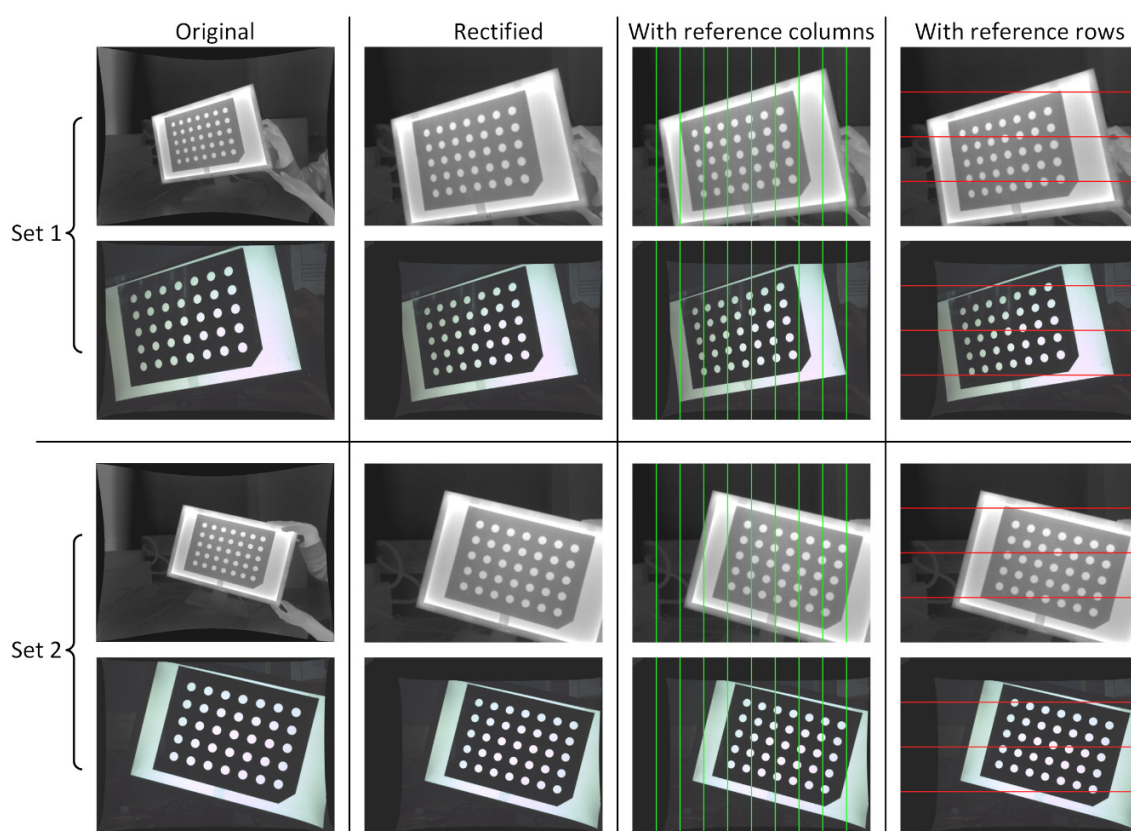


Figure 7-14 Stereo rectification results in comparison to their original image

Furthermore, two of the example stereo image pairs are selected from Figure 7-8, and Figure 7-14 shows the result of stereo rectification of the infrared and visible image pairs. It can be seen from the stereo image pair that each image is rotated so the coplanar of the image plane can be achieved. With reference to green lines drawn on the rectified

image pairs, one can see that the white dot become vertically aligned, and that means the stereo correspondence is achieved. Because the stereo camera rig is arranged to be as nearly parallel as possible, and as vertical to ground as possible, a better stereo correspondence result is obtained [73].

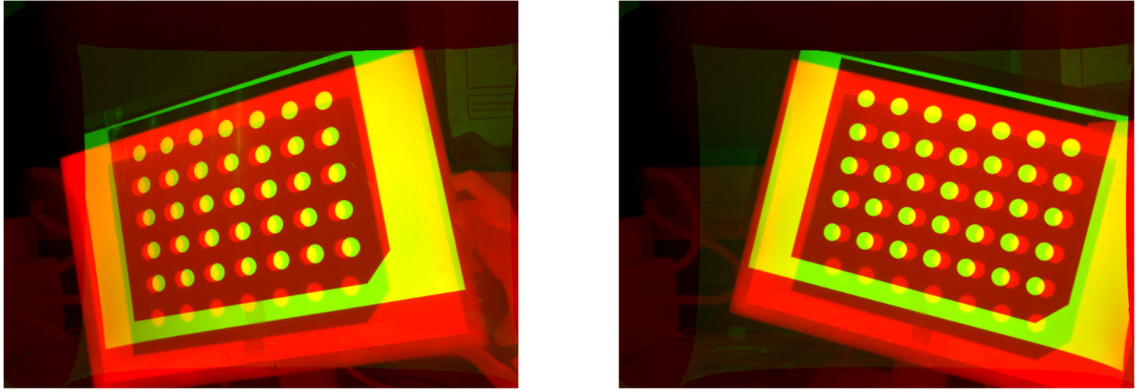


Figure 7-15 Stereo image pair set 1 (left) and set 2 (right) overlay result

The disparity can be noticed from the image with blue reference lines as the white dots are not row aligned. Figure 7-15 shows that if an overlaid image is created now from Figure 7-14, the infrared image is coloured red, and the visible image is coloured green for a better-visualized result. From the fused images, one can see that the image points for the same object are not in the same position.

#### 7.2.6. Stereo image matching and result

The stereo image overlay problem is the problem of finding corresponding pixels in a rectified stereo image pair. The goal of overlaying in this project is to obtain an image that contains both visible and infrared information. From the previous section of stereo rectification that the overlapping area is already maximized; the stereo image pair is column-aligned but not row-aligned. This section will focus on the approach that is used to solve the stereo matching problem.

As mentioned in section 7.1.1, equation (8.1.2) indicates a way to calculate disparity

in an image from the distance  $Z$  from the observing object to the cameras and the length of baseline  $T$  that:

$$d = x_R - x_T = \frac{Tf}{Z} \quad (8.1.2)$$

where  $f$  stands for the focal length of the stereo system if the two images are coplanar to each other. The focal length  $f = 724.9841 \pm 0.4623$  is in pixel, and it is obtained from a single camera calibration process using either a stereo rectified infrared image set or visible image set. The length of the baseline  $T = 40mm$  was previously measured. The unit of the resulting disparity should be in pixel, and the value of disparity defines the reference frame's translation.

After the reference image is rectilinear shifted, the image pixel for an observing object should be in the same position on an overlaid image. Thus, equation (8.1.2) can be expressed as the inverse proportional relationship between the object depth and the disparity:

$$d \approx \frac{28999.4}{Z} \text{ pixels} \quad (8.2.13)$$

All the calibration images presented in Figure 7-8 are taken by the fixed position stereo cameras of the calibration board, and the calibration board placed **650mm** apart from the baseline. According to equation (8.2.13), the disparity in a stereo image pair is about 45 pixels. By subtracting the amount of disparity to  $y$ -axis of the infrared image or adding it to the visible image, a stereo overlay will be created. Figure 7-16 shows the overlay result based on the rectified stereo image pairs in Figure 7-15, and they are achieved by shifting up the infrared image.

As can be observed from Figure 7-16, the circles on the visible image overlap with its corresponding ones on the infrared image. The overlaid image border could be cropped for better demonstration, but a part of the infrared information will be lost. It is a

method to create alignment between the infrared image and the visible image with predetermined camera parameters and remapping look-up tables. This information is stored in the FPGA as files; the server program can access it directly, and the client program can obtain them via communication with the server.

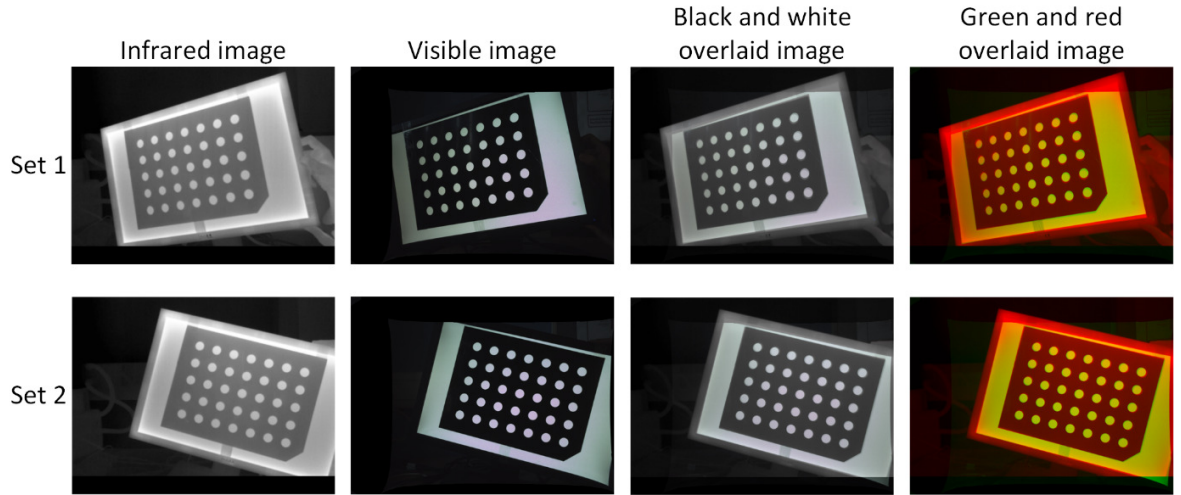


Figure 7-16 Stereo image pair set 1 and set 2 overlay result after translation

To evaluate the algorithm of stereo matching the infrared and the visible image, then to create a single image without disparity, several images are taken with objects in the scene are placed at different distance. Figure 7-17 demonstrates the stereo rectified result of the field of view, and Figure 7-18 presents the overlay views and the mosaic views created to have a detailed look at the matching result. It has to be admitted that some misalignment can be seen in the edge of the object. It is mainly because the two cameras' optical rays are not on the same line, so they captured the scene from different angles. It is the same truth as human eyes that the left eye and right eye's views are different [102].

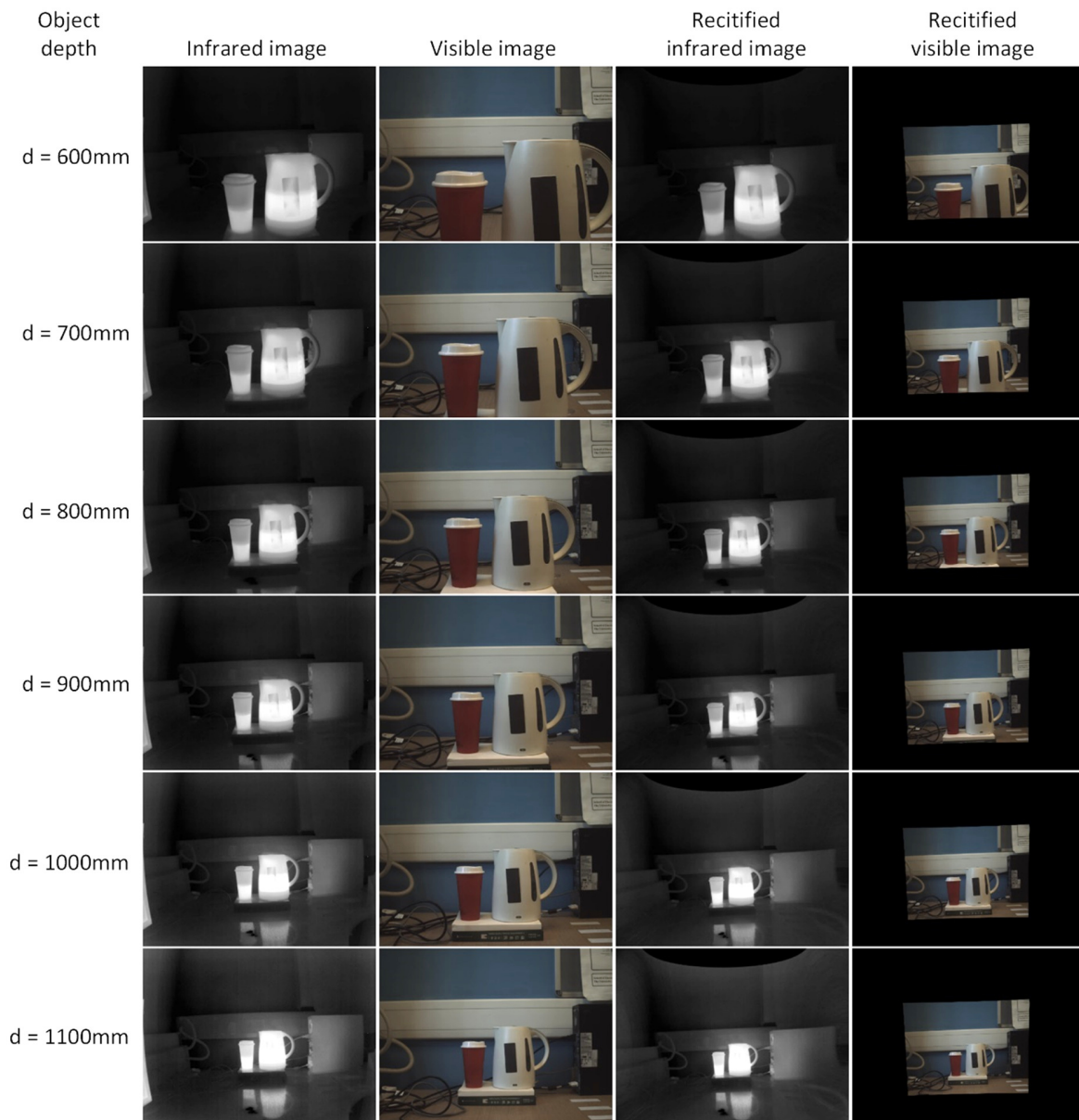


Figure 7-17 Images taken from a variety of distance and the result after stereo rectify



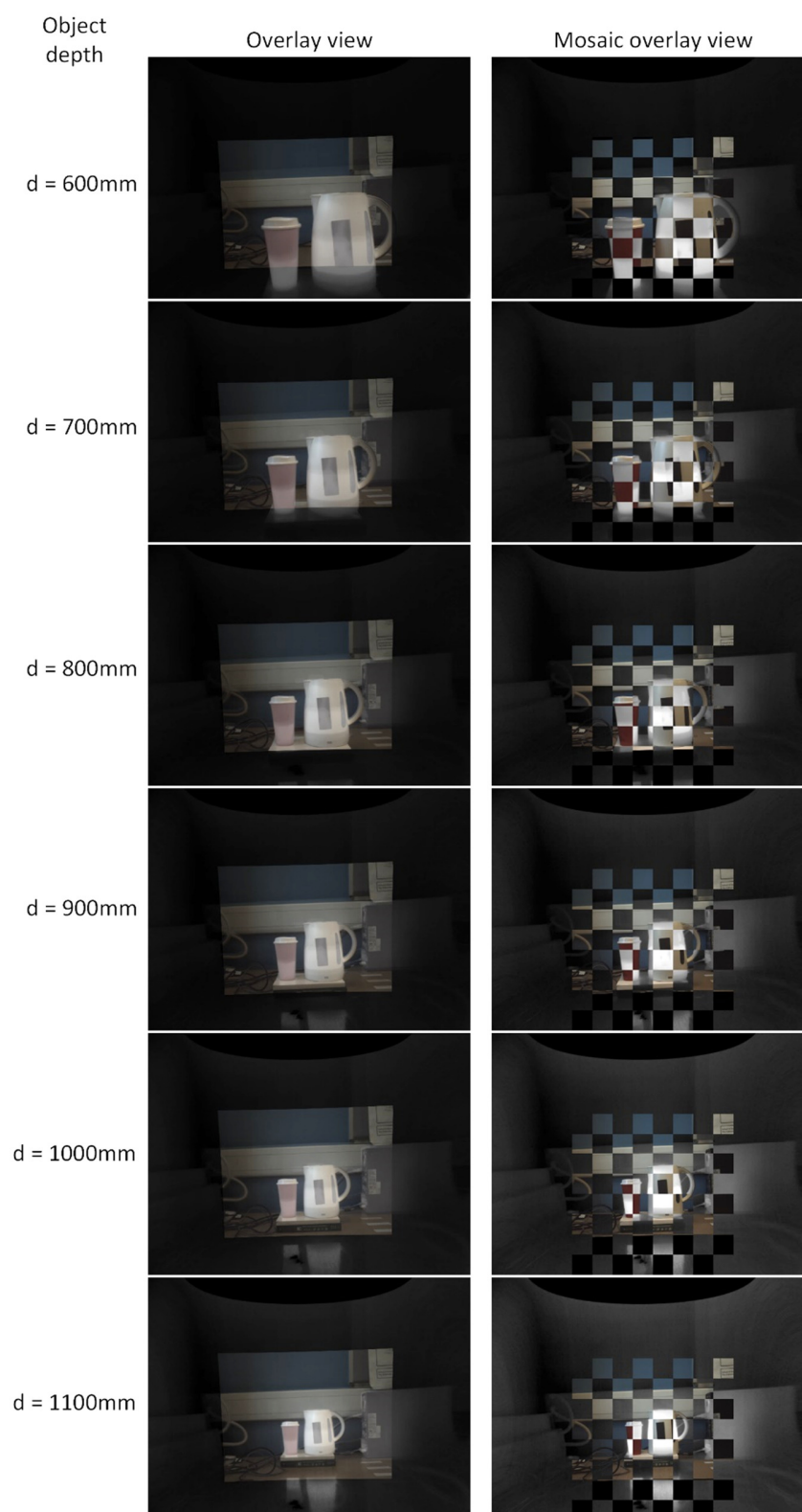


Figure 7-18 Infrared image and visible image overlay and mosaic overlay result

### 7.3. Image registration

Apart from the stereo matching method to overlay the infrared and visible image, another method called image registration is tested to see if the frame rate of overlaying two images can be higher. Image registration is a process to obtain an overlaid two or more images of the same scene that may be taken at different time, from a different point of view, and/or from different sensors [103]. The infrared image and the visible image are taken from different sensors, which means that the two sensors have different coordinates. Disregarding the distortion effect, the misalignment between the two images is mainly caused by the position of the two sensors and their point of view. Image registration is an approach to transform multiple images into one coordinate system. This approach determines the correct correspondence between the two images and provides the parameters for spatial transformation accordingly.

There are usually two categories of infrared and visible image registration methods: feature-based method and area-based method [104]. The feature-based image registration extracts two sets of salient structure from a pair of the infrared and visible image, and then these structures are comprised of feature edges [105], feature lines and/or feature points [106]. The correct correspondence between the images is determined, and the spatial transformation is estimated accordingly, thereby matching the infrared and visible image pair. The area-based method deals with the intensity of pixel values in infrared and visible image pairs rather than the salient features. The cross-correlation method is area-based and is preferable in this project, so the infrared image can be fully utilized without detecting the corresponding structures [107].

As it was previously seen that the infrared camera's field of view is larger than the visible camera's, the infrared image should be considered as the reference frame, and the

two images can be aligned by scaling, rotating and translating the visible image using the parameters obtained. The similarity metric could achieve its maximum between the two images taken by two different sensors. In this experiment, an image registration scheme based on phase correlation is implemented to achieve the translation, rotation, and scale parameters. This method is worth trying as it conducts an entirely different procedure to process the infrared and the visible image to create a new matching image, as parameters to overlay two images are not predetermined.

### **7.3.1. Cross-correlation based image registration**

Phase correlation is commonly used in the image registration technique; it analyses the input images in the Fourier domain and determines the transformation between the input images [108]. The Fourier domain contains the counterpart of scale, rotation, and translation. An advantage of the Fourier method compares to other registration algorithms, such as the feature-based method, is that the Fourier methods seek the optimal match automatically according to the information acquired in the frequency domain [109].

The fast Fourier transform is an essential tool in image processing. By converting the visible image to greyscale, the Fourier Transform can work with pixel value intensity and convert it to frequencies. When applying fast Fourier transform (FFT) on an image, it decomposes the image into its sine and cosine components, and the output of the transforming represent the input image in the frequency domain [110]. The input image is the spatial domain equivalent to the output of the transformation. Each point in the Fourier domain image represents a particular frequency contained in the spatial domain image. An extension of the cross-correlation method is applied in this experiment to conduct an automatic image registration process and attempted to minimize the

computational cost comparing to the remapping method in stereo calibration.

The infrared image and the visible image are seen as the reference image and the target image. Supposing that there is no scale and rotation change between the reference frame  $f_r$  and the target frame  $f_t$ , the only difference between the two images is the displacement of pixel  $f(u, v)$ , and the relation of the pixels on these two frames can be given as:

$$f_r(u, v) = f_t(u - \Delta u, v - \Delta v) \quad (8.3.1)$$

where  $(\Delta u, \Delta v)$  represent the displacement of the pixels in the target frame. Then the relationship between the corresponding Fourier transforms  $\mathcal{F}_r$  and  $\mathcal{F}_t$  can be achieved using the Fourier shift theorem as:

$$\mathcal{F}_r(f_u, f_v) = e^{-j2\pi(f_u\Delta u + f_v\Delta v)} \mathcal{F}_t(f_u, f_v) \quad (8.3.2)$$

where  $e$  is the natural logarithm base, and  $(f_u, f_v)$  represent  $(u, v)$  in the frequency domain. The Fourier shift theory ensures that the phase difference between the two images is equal to the cross-power spectrum. The term cross-power spectrum represents the similarity of two time series, and in particular, the point that the similarity matches the most [111]. The cross-power spectrum of the reference image and the target image with Fourier transform  $\mathcal{F}_r$  and  $\mathcal{F}_t$  respectively can be related by

$$\frac{\mathcal{F}_r(f_u, f_v) \mathcal{F}_t^*(f_u, f_v)}{|\mathcal{F}_r(f_u, f_v) \mathcal{F}_t^*(f_u, f_v)|} = \frac{\mathcal{F}_r(f_u, f_v) \mathcal{F}_r^*(f_u, f_v) e^{-j2\pi(f_u\Delta u + f_v\Delta v)}}{|\mathcal{F}_r(f_u, f_v) \mathcal{F}_r^*(f_u, f_v) e^{-j2\pi(f_u\Delta u + f_v\Delta v)}|} = e^{-j2\pi(f_u\Delta u + f_v\Delta v)} \quad (8.3.3)$$

where  $\mathcal{F}_t^*$  is the complex conjugate of  $\mathcal{F}_t$  and  $\mathcal{F}_r^*$  is the complex conjugate of  $\mathcal{F}_r$ . The inverse Fourier transform of  $e^{-j2\pi(f_u u_0 + f_v v_0)}$  is a Dirac delta function ( $\delta$  function) which located at the point of registration [112] and can be written as:

$$\mathcal{F}^{-1} e^{-j2\pi(f_u\Delta u + f_v\Delta v)} = \delta(u - u_0, v - v_0) \quad (8.3.4)$$

The phase correlation method is to calculate the inverse Fourier transform of the equation (8.3.3), and then to determine the shift parameters  $(u_0, v_0)$  by finding the location of the peak value in the impulse function. Moreover, the impulse function guarantees that

it is high at the translation, which is the most appropriate to register the two images, and approximately low elsewhere. The location of  $(u_0, v_0)$  is also the displacement that needed to register the images optimally.

After that, the 2-D rotation should be considered, and the rotation angle is defined as  $\theta_0$  here, and  $\theta_0$  is also called the phase correlation angle. To relate the points  $f_r(u, v)$  on reference image to the points  $f_t(u, v)$  on the target image, the following equation is provided:

$$f_r(u, v) = f_t(u \cos \theta_0 + v \sin \theta_0 - \Delta u, -u \sin \theta_0 + v \cos \theta_0 - \Delta v) \quad (8.3.5)$$

The magnitudes of the Fourier transform  $\mathcal{F}_t$  and  $\mathcal{F}_r$  are defined as  $M_t$  and  $M_r$  respectively, they can be derived from equation (8.2.5) that:

$$M_r(f_u, f_v) = M_t(f_u \cos \theta_0 + f_v \sin \theta_0, -f_u \sin \theta_0 + f_v \cos \theta_0) \quad (8.3.6)$$

If squares both side of the equation above, it can be found out that the magnitudes of  $\mathcal{F}_t$  and  $\mathcal{F}_r$  are equal, and the difference comes from the rotation part, which means that  $\mathcal{F}_t$  is the rotation replica of  $\mathcal{F}_r$  or vice versa. The equation above can be simplified in polar representation as;

$$M_t(\rho, \theta) = M_r(\rho, \theta - \theta_0) \quad (8.2.7)$$

Equation (8.2.7) represent the rotation change but not include the translation movement. The phase correlation angle  $\theta_0$  can then be calculated by using equation (8.3.3).

All the derivation above has not taken the scale parameter into account. Consider that there are scale vectors in  $x$ -axis and  $y$ -axis given as  $(a, b)$  that replicate the points on the target image  $f_t$  to the point on the reference image  $f_r$ , the relationship between  $\mathcal{F}_t$  and  $\mathcal{F}_r$  can be derived using the Fourier scale theorem that:

$$\mathcal{F}_r(f_u, f_v) = \frac{1}{|ab|} \mathcal{F}_t\left(\frac{f_u}{a}, \frac{f_v}{b}\right) \quad (8.3.8)$$

By taking the logarithmic scale on both sides of the equation above, the scaling in images can be reduced to a form of translation and the multiplication factor  $1/|ab|$  can be

ignored as following:

$$\mathcal{F}_r(\log f_u, \log f_v) = \mathcal{F}_t(\log f_u - \log a, \log f_v - \log b) \quad (8.3.9)$$

it can be simplified as:

$$\mathcal{F}_r(x, y) = \mathcal{F}_t(x - c, y - d) \quad (8.3.10)$$

where  $x = \log f_u$ ,  $y = \log f_v$ ,  $c = \log a$  and  $d = \log b$ . Moreover, the translation of the target image is expressed in equation (8.3.10) as  $(c, d)$ , and it can be obtained from the phase correlation mentioned previously. Therefore, the scaling  $(a, b)$  can be achieved from the translation  $(c, d)$  that:

$$a = e^c \text{ and } b = e^d \quad (8.3.11)$$

Supposing that an image point  $f_t(u, v)$  is scaled by the parameter  $a$  and the new location is  $f_r(u/a, v/a)$ , the polar representation  $(\rho_1, \theta_1)$  of the original point and  $(\rho_2, \theta_2)$  of the new point are given as:

$$\rho_t = \sqrt{(u^2 + v^2)} \text{ and } \theta_t = \arctan \frac{v}{u} \quad (8.3.12)$$

$$\rho_r = \frac{1}{a} \sqrt{(u^2 + v^2)} \text{ and } \theta_r = \arctan \frac{v/a}{u/a} = \arctan \frac{v}{u} = \theta_1 \quad (8.3.13)$$

The relationship between the magnitude spectra of  $f_t$  and  $f_r$  can be written as:

$$M_t(\rho, \theta) = M_r\left(\frac{\rho}{a}, \theta - \theta_0\right) \quad (8.3.14)$$

The equation above can be reduced to translational movement by converting the  $\rho$ -axis to a logarithmic scale. By using the logarithm expression, it can be rewritten as:

$$\begin{aligned} M_t(\log \rho, \theta) &= M_r(\log \rho - \log a, \theta - \theta_0) \\ &= M_t(f_u, \theta) = M_r(f_u - d, \theta - \theta_0) \end{aligned} \quad (8.3.15)$$

where  $f_u = \log \rho$  and  $d = \log a$ .

According to the cross-correlation theory and adopting equation (8.3.15), the scale parameter  $a$  and the phase correlation angle  $\theta_0$  can be calculated. In this stage, if scaling and rotating the reference image, the only difference between the reference image and the target image should be the translation difference. The translation parameter can be

obtained using the transformed image pair by equation (8.3.3).

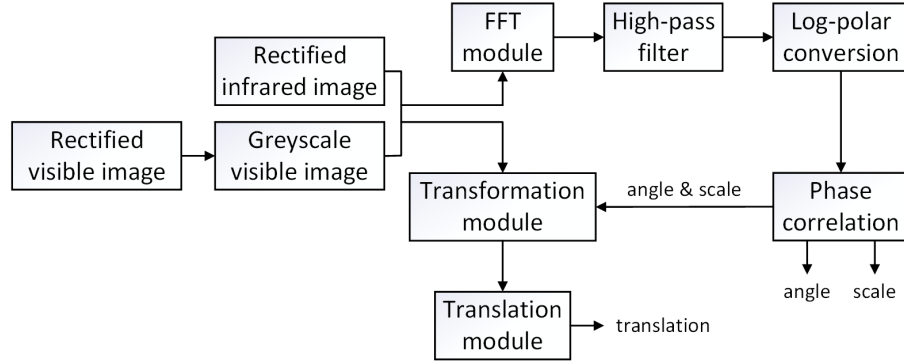
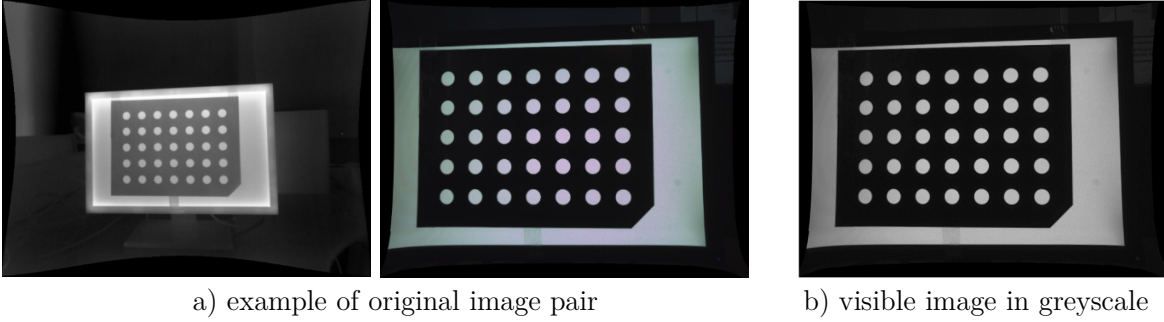


Figure 7-19 Flow chart of FFT-based image registration algorithm

The flow chart in Figure 7-19 outlines the process of the image registration algorithm using FFT. The algorithm can be summarized as follow:

1. Apply FFT to reference images  $f_r$  and target image  $f_t$  to obtain  $\mathcal{F}_r$  and  $\mathcal{F}_t$ ;
2. Compute the absolute values of  $\mathcal{F}_r$  and  $\mathcal{F}_t$ ;
3. Apply high-pass emphasis filter to  $|\mathcal{F}_r|$  and  $|\mathcal{F}_t|$  to remove noise and emphases edges;
4. Transform the resulting value from cartesian coordinates to log-polar coordinates;
5. Compute the cross-power spectrum of the two resulting phase spectra;
6. Compute the inverse FFT of the cross-power spectrum to achieve the value of scale and rotation angle;
7. Construct a new image  $f_{t_{new}}$  by applying reverse scaling and rotation to the target image;
8. Compute the cross-power spectrum of  $f_r$  and  $f_{t_{new}}$  and calculate the inverse FFT of it to obtain  $(u_0, v_0)$  as the translation parameters.



a) example of original image pair  
b) visible image in greyscale  
Figure 7-20 A pair of undistorted original images and grey scaled visible images

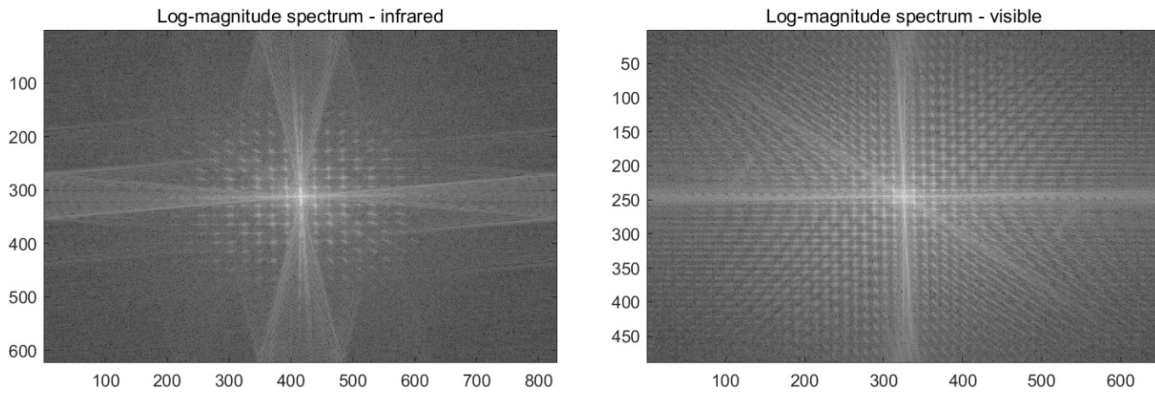


Figure 7-21 Fourier log-magnitude spectrum in cartesian coordinates

Figure 7-20 shows the original image pair and the visible image are converted to greyscale for Fourier transform; the infrared image does not need to be converted to greyscale as the original image taken by the infrared camera is in greyscale. Figure 7-21 presents the Fourier log-magnitude spectrum of the greyscale image pair in Figure 7-20 in cartesian coordinates. The Fourier transform of the infrared and visible image shows that they contain components of all frequencies. The higher frequencies in an image imply a bigger change in values and correspond to detailed texture, while the lower frequency represents smaller changes in pixel values and always related to plain areas [113]. For the polar transform mentioned previously, which remapping points  $(u, v)$  from the 2-D Cartesian coordinate to the 2-D log-polar coordinate system as  $(\rho, \theta)$ , the Fourier log-magnitude spectrum is applied instead of the Fourier magnitude spectrum so that the scaling parameter of the image is changed to a translational movement.



As illustrated in Figure 7-19, a 2-D Gaussian high-pass frequency filter is applied after the FFT module. The high-frequency domain image filtering emphasises the high frequencies in the image, and the low-frequency noises are eliminated [114]. The Fourier log-magnitude spectrum of an image multiplies a high-pass transfer function and then been transformed to log-polar coordinate. The transfer function is:

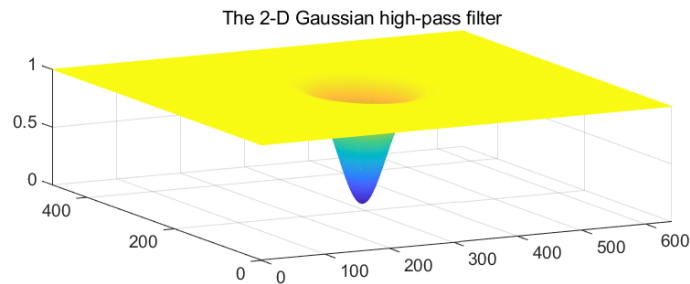
$$H(f_u, f_v) = 1 - X(f_u, f_v) \quad (8.3.16)$$

where

$$X(f_u, f_v) = e^{\frac{-\left(\left(f_u - \frac{h}{2}\right)^2 + \left(f_v - \frac{w}{2}\right)^2\right)}{2d_0^2}} \quad (8.3.17)$$

and  $(h, w)$  represents the image height and width.  $d_0 = 30$  is the standard deviation of the distribution, and it can be altered to control the high-pass filter scale in spatial [115].

Figure 7-22 depicts the high-pass emphasis filter and the high-pass filtered Fourier log-magnitude spectrum of the image pair, and Figure 7-23 present the image pair after passing a high-pass filter. It can be seen in the high-pass image pair that the boarder pixels between the contrasting areas are emphasised, the highlighted pixel contrasts associated with the linear features and edge detail of the image, so a more accurate result can be obtained when conducting the phase correlation process.



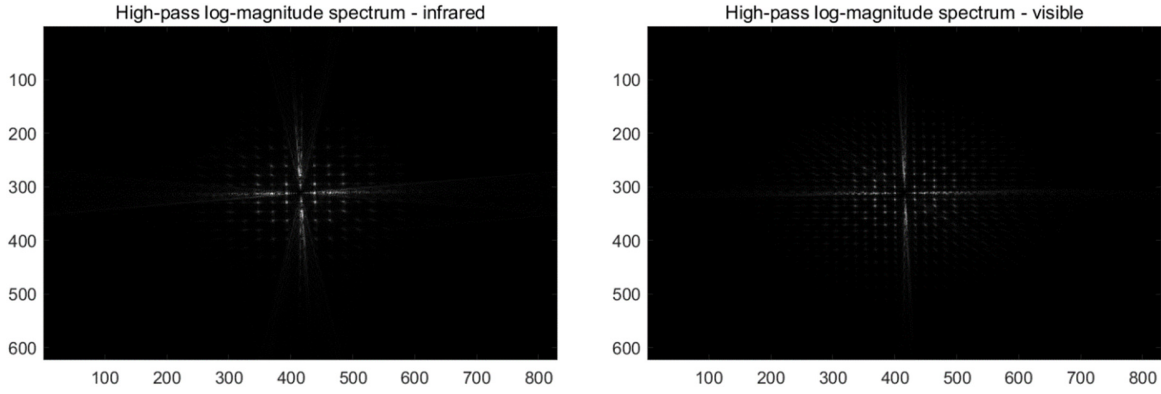


Figure 7-22 The high-pass emphasis filter and high-pass filtered Fourier log-magnitude spectrum

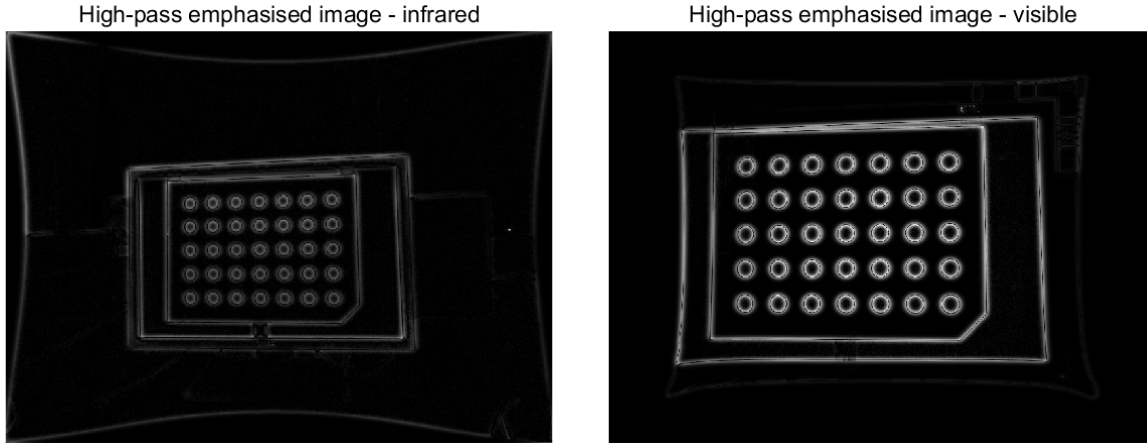


Figure 7-23 High-pass emphasised image pair

Since the Fourier spectrum of a signal/image  $F(f_u, f_v)$  is conjugate symmetric as  $F^*(-f_u, -f_v)$ , only the lower two quadrants of the Fourier log-magnitude spectrum are used for the mapping to reduce the workload of computing. The logarithmic conversion of  $\rho$ -axis in the process when converting  $(f_u, f_v)$  to  $(\rho, \theta)$  is done with a base of 1.026 because  $\log_{1.026} 480 \approx 240$ , which means that 480 rows of image are mapped to 240 rows in polar coordinates. Other values can also be chosen for the base value depending on the demanding level of accuracy; a larger base value will lead to less accuracy but less time consuming when computing. Furthermore, Figure 7-24 present the Fourier log-magnitude spectrum of the image pair after converting it to polar coordinates.

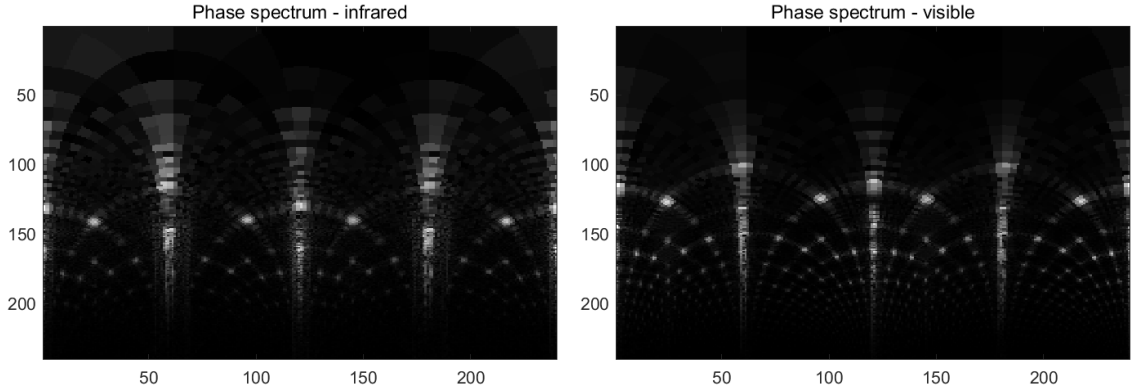


Figure 7-24 Fourier log-magnitude spectrum in polar coordinates

The images in this experiment are of size  $640 \times 480$ , and it is of size  $240 \times 240$  in log-polar coordinates. If a calculation result  $(x, y)$  represent the location of the peak point when conducting the inverse Fourier transform the cross-power spectrum, the scale and rotation angle obtained from it should be converted to achieve the actual scale and angle, the conversion is as follow:

$$actual\ scale = \sqrt{1.026} \times scale\ obtained \quad and \quad actual\ angle = 180y/480$$

These are donating as the affine transformation parameters and are used to transform the reference image. The same linear interpolation method as in stereo matching is used for image transformation to fill the gaps when scaling and rotating. Then the transformed image is used for the calculation to find out the translation between the transformed image and the target image.

There is a  $180^\circ$  ambiguity of the rotation angle obtained from the above calculation as their absolute values replace some of the calculated values. To solve this problem, the spectrum of the target image should be rotated by  $\theta_0$  and  $\theta_0 + 180^\circ$  separately, and passing these results together to the translation module for calculation. Then inverse Fourier transform of the cross-power spectrum of both rotation angle is computed for comparison, and the one with greater peak value should be the correct angle of rotation.

### 7.3.2. Registration result and analysis

To evaluate the image registration algorithm's performance at the first stage, the input image pairs are chosen from the chessboard pattern images and the circle grid pattern images, as these calibration images have clear object lines and edges. The cross-correlation is calculated for the two input images to measure their similarity in the frequency domain, and Figure 8-25 shows the cross-power spectrum of the image pair after the reference image is scaled and rotated. Many peaks occur in the cross-power spectrum; a particular strong peak among all peaks suggests that the two frequencies' similarity reach the most at that location [116]. The strongest peak occurs at [197,163] in the cartesian coordinates cross-power spectrum, which implies that the reference image should be shift 197 pixels in  $x$ -direction and 163 pixels in  $y$ -direction.

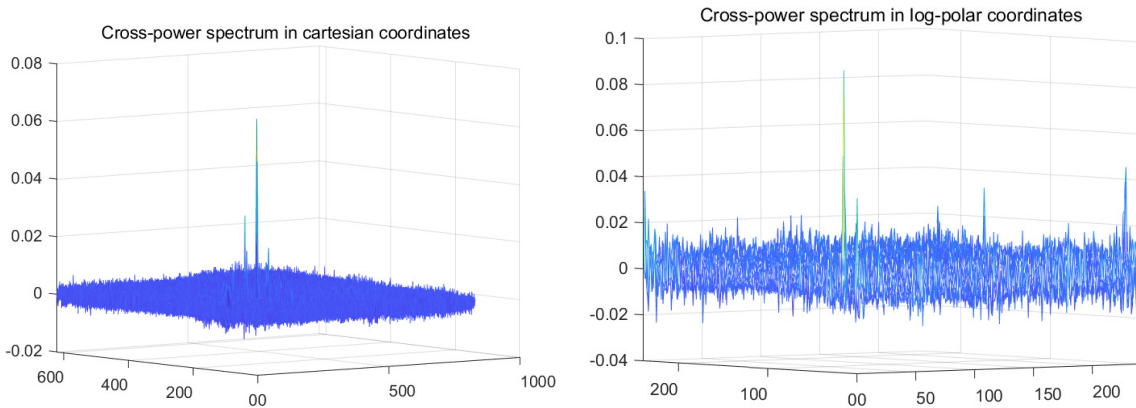


Figure 7-25 An example of the cross-power spectrum of a calibration pattern image pair

Figure 7-26 demonstrates that the image registration method is working correctly on the undistorted calibration pattern images. The image pairs with fisheye effect are also tested as the inputs, but the chance of finding peak points dropped by approximately 14%. The majority cause of the rate drop is that the effectiveness of lens distortion is different from the two cameras, leading to varying levels of curvature of the edges and lines in the image.

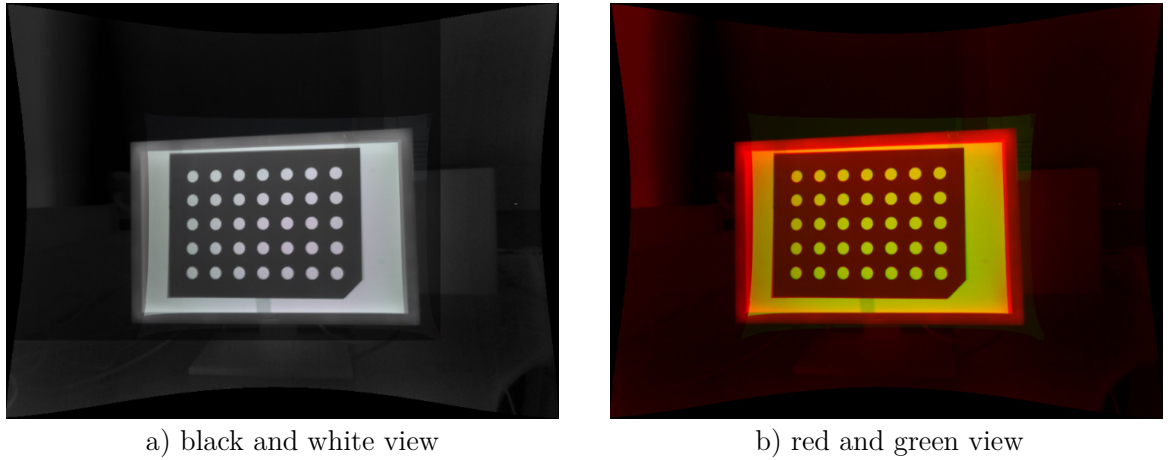
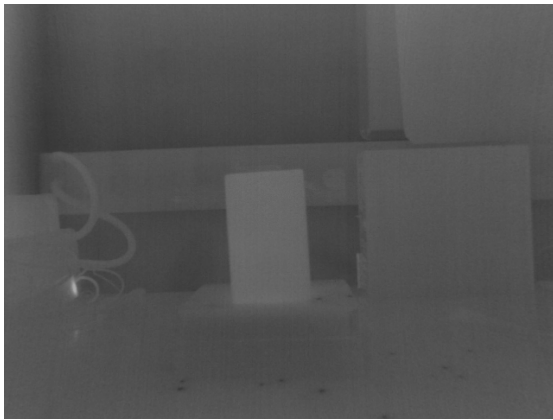


Figure 7-26 Calibration pattern image registration result

As introduced in stereo calibration section 7.1.1, the disparity between the infrared and visible image is an unavoidable problem when conducting an image registration algorithm. Because that the two cameras are taking images from different positions, which result in a diverse of pixel locations in spatial relations. To eliminate the effect caused by camera positioning, several image pairs were taken under the condition that the camera's mounting point is adjusted manually. The image registration algorithm can be accomplished, and the outcome is the same for the calibration pattern image pairs.

Other than the calibration pattern image pairs, a number of image pairs are captured with arbitrary objects in the field of view, with the camera's position being adjusted manually. The success of registration decreases by 83% when using arbitrary scenes. Figure 7-27 presents an image pair and the cross-power spectrum. The visible image shows two black stickers on the box; however, because the black sticker is the same temperature as the box, they merge into one piece on the infrared image. Thus, the features cannot be extracted and appropriately matched. Furthermore, the cross-correlation result is clear that there are too many peaks but no existence of a particular strong peak when applying cross-correlation on the image pair. In reality, it is hard to manipulate what is captured by the cameras, and there is little chance that there will be

a calibration pattern or anything with multiple vivid edges in the scene.



a) infrared image



b) visible image

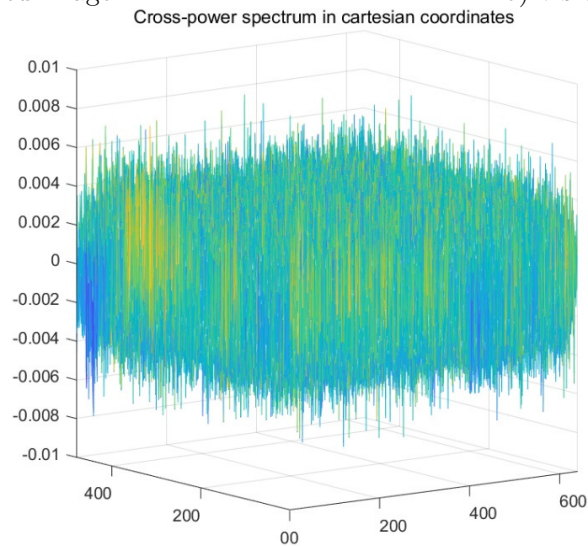


Figure 7-27 An example of an arbitrary scene image pair cross-correlation result

## 7.4. Comparison of two image matching methods

In summary, the stereo matching method is to achieve two aligned images using the camera parameters and the distance of the observing object. The FFT based image registration method analyse the image pairs in the frequency domain and using the Fourier shift theorem to determine the parameters to match two images. A more suitable method should be chosen for this experiment; thus, the stereo matching method and the image registration method are compared in their prerequisites, efficiency, and

computational cost.

#### 7.4.1. Limitations and advantages

**a) Stereo matching limitations:** The prerequisite of the stereo matching method in this project is first to conduct single-camera calibration to obtain the camera's parameters: intrinsic matrix and distortion coefficients. The stereo camera calibration should be conducted to achieve the related parameters between the two cameras: rotation matrix and translation vector. With all these parameters being resolved, a look-up map can then be generated to remap the original images to create a column aligned stereo frame.

This stereo system setup can be considered a drawback because the two cameras are in fixed positions and must be mounted as parallel as possible to achieve the best image overlay result. The physical alignment of the two cameras can make the mathematical transformations more tractable [73]. Suppose the two cameras are not aligned at least approximately. In that case, the calculated mathematical alignment can result in abnormal or extreme image distortions, leading to the reduction or elimination of the stereo overlap area of the resulting stereo matching image, particular for the interesting object in at close range.

The major problem in stereo matching is that any object's distance in the scene cannot be obtained automatically. Consider a small object in the front scene of the camera, and a large object farther away from the camera, the size may be the same in both cameras. However, the disparity should be different for these two objects; thus, the shift parameter should vary.

**b) Stereo matching advantages:** Benefiting from the look-up map, the stereo matching algorithm can remove lens distortion and rectified the image pair at the same

time. It simplifies the matching process by not processing the texture or colour information from each frame individually. The remapping process significantly reduces the processing time to overlay the infrared and visible image pairs, especially in this real-time program aiming to achieve a higher frame rate. The stereo matching can deal with the scene not containing sufficient features, such as lines, edges, or corners. The issue caused by pixel intensity discontinuities is absent after the two cameras are calibrated.

**c) Image registration limitations:**

The first and significant disadvantage of the image registration method is that the infrared image mainly shows the heat radiation intensity of an object, which is different from a visible image. As evidenced in Figure 7-17, the visible image shows that the sticker on the kettle is black; however, the infrared image reflects that the sticker is warm because there is hot water in the kettle, and the colour of the sticker is not black. It is nearly an inversion of the pixel greyscale between the infrared image background and the visible image. Thus, extracting common features from an infrared and visible image pair should be difficult due to the difference in the camera sensors; also, the result of area-based registration can be easily influenced by the intensity of pixel value.

The second disadvantage is that the visible image must be converted to greyscale for Fourier transform because the Fourier transform only process the intensity of pixel value and convert it to the frequency domain. It is possible to split the visible image into three different images, one of each of the red, green, and blue components, so each of those can be processed with a Fourier transform independently. Although this process would make the most information usage of the visible image, it also increases the program's complexity and time consumption.

**d) Image registration advantages:**

The area-based image registration method has the advantage that if the salient structure



in the infrared and visible image is not sufficient, the prominent information can be provided by the intensity of pixel value but not the prominent structures in the scene because the pixel value of a pictured structure also affected by environment illumination changes.

The registration of infrared image and visible image can match image pairs under certain conditions; it does not require that the two cameras' optical ray be set as parallel as possible, and the position of the two cameras does not have to be fixed. Moreover, single-camera calibration will not be required if the two cameras' lens distortion effect is not significant.

#### 7.4.2. Image processing time and comparison

Table 7-1 presents the average frame rate for creating an overlaid image when using different methods. The frame rate is calculated based on  $640 \times 480$  resolution images. The same as section 6.7, the calculation was done using image files as inputs because the execution time for the server program to retrieve the image from DDR3 memory is long. The start of the image processing point is inserted after the OpenCL kernel parses the LUT for remapping, and the finish point of stereo matching is when the server program generates a full stereo matching frame and is ready to send it to the client. Compared to the frame rate in Table 7-6 when producing an undistorted image, the frame rate of producing an overlaid image drops more than half because two images need to be remapped, and one of which needs to be translated to create an overlaid view.

Table 7-1 Frame rate of producing an overlaid image in different platforms

	Intel Core i7 CPU	ARM A9 processor	FPGA OpenCL device
Stereo matching (frame/s)	7.67	5.66	10.15
Image registration (frame/s)	1.12	0.30	

The operation time of the two image matching methods is compared on PC when the two methods are determined. There is a significant reduction in the frame rate when using the image registration method, and it does not perform well on the ARM A9 processor compared to the stereo matching method. Thus, image registration was not transplanted to OpenCL kernels when finalising the image overlay algorithm to the FPGA OpenCL device.

Most lens undistortion and stereo matching for dual camera vision system developed recent decades are based on software running on a desktop environment, most of their object is to create a depth map of the scene from the stereo vision. A smaller number of them implemented the algorithm on embedded system using FPGAs, in which it is hard to find a system like this work adopt multispectral cameras and use OpenCL framework to accelerate the image remap speed. A comparison has been made and analysed between the real-time performance of this system with other reported similar implementation on various FPGAs.

Table 7-2 Comparison result

Evaluated design	Resolution	Frame/s	FPGA
Zheng <i>et al.</i> [117]	$320 \times 256$	25	Xilinx Virtex-5
Bellas <i>et al.</i> [118]	$640 \times 480$	11	Xilinx Virtex-4
Honegger <i>et al.</i> [119]	$752 \times 480$	30	Xilinx Artix-7
Castells-Rufas <i>et al.</i> [120]	$1280 \times 960$	2.73	Xilinx Zynq-7020 SoC
Wang <i>et al.</i> [121]	$1600 \times 1200$	21	Altera Stratix V
Domański <i>et al.</i> [122]	$1920 \times 1080$	5.5	Xilinx Artix-7
This work	$640 \times 480$	10.15	Altera Cyclone V SoC

In fact, the frame rate not only depends on the algorithm designed, it also depends on the processing speed of the FPGA, the number of free DSP units, the resolution of image and the number of bits that a pixel has. The OpenCL framework here proves it

useful for the acceleration of the stereo matching algorithm. Compare to the compile and test time of work [117], [118] and [119], OpenCL method shows its advantage as it take minutes to compile rather than hours. Thus, a flexible platform can be provided by the OpenCL frame work with early performance estimation that can ease the programming decisions before investing a long time in HDL synthesis.

## 7.5. Summary

In this chapter, the stereo camera system setup was presented; the infrared camera and the visible camera are mounted inside the box and pointing in the same direction. The two cameras form a stereo system, and the geometry of the system is introduced. This chapter's primary issue is to minimize the disparity in images caused by the positioning difference of the two cameras and to match the image pairs. Two methods are proposed to match the infrared and visible image pairs. The stereo calibration method requires the two cameras are calibrated, and the cameras' intrinsic parameters are obtained. The two images are rectified to column aligned first, and then using the object's distance to determine the translate parameter. The image registration method is area-based and required the two images to have more common lines or edges. The FFT based image registration was not a successful attempt for matching the infrared and visible image pairs. At the end of this chapter, the two image matching methods are compared regarding their advantages and limitations. A conclusion can be drawn that the stereo calibration method is more suitable for this project because it neglects the problem caused by inverse pixel intensity, and its processing time is shorter than the image registration method.

## Chapter 8

# Radiometric calibration

It is a common misconception that thermal cameras are capable of directly measuring the temperature of the surfaces in the scene. Thermal cameras can measure the object surface's infrared radiation but have a complicated relationship with its surface temperature. Radiometric calibration is a process to establish the relationship between the response of the infrared sensor and the temperature of the object surface. Then the infrared camera is enabled to be used for actual surface temperature estimation by modelling its response [123]. The infrared camera used is Xenics' XTM-640 thermal camera module; it is based on an uncooled microbolometer, and the working principle had been presented in section 2.1.

Regarding an uncooled thermal camera, the infrared-detecting element is in a chamber operating at surrounding environmental temperature, which means that the microbolometers are not stabilized to a constant temperature. This makes the XTM-640

infrared module has low precision for many applications involving temperature measurement. In this chapter, further information regarding the uncooled camera module XTM-640 is provided, as well as its comparison to the cooled infrared cameras. Several configurations are made to the XTM-640 module to perform the radiometric calibration. The purpose and nature of radiometric calibration are illustrated, and the method to achieve a reliable calibration result is presented. At the end of this chapter, a merging view of the infrared image and visible image is created to visualise the temperature information on top of the visible image.

## **8.1. Uncooled infrared camera introduction**

Infrared cameras are adopted in a wide range of different applications, such as night surveillance [124], building thermal-leakage detection [125], forest fire early warning system [126] and imaging CO<sub>2</sub> gas-leakage detection from vegetation [127]. According to their image detector type, these thermal devices can be classified as cooled thermal imagers and uncooled thermal imagers. Modern cooled infrared sensors are usually packaged in a unit integrated with a cryocooler, which keeps the sensor at a very low temperature [128]. Uncooled infrared sensors operate at ambient temperature, and they make use of uncooled pyroelectric or microbolometer to measure incident infrared radiation [129]. Uncooled infrared modules are overwhelmingly the most common type of infrared detectors used in commercial thermal cameras [130], and are the kind applied in this project.

### **8.1.1. Comparison between cooled and uncooled thermal system**

The temperature of the infrared sensor in a cooled infrared system is cooled by cryocooler, so the thermally-induced noise is reduced. As a result, the signals from the scene being

images are above the level of noise [131]. Each time the system initiates, it must be cooled to prevent itself from being affected by its infrared emissions. The cooled infrared sensors are more sensitive to tiny differences in temperature between objects in the field of view, and it has the ability to perform spectral filtering when taking a measurement or revealing details [128]. However, the requirement of an external cooling system is generally at least an order of magnitude more expensive. The cooled infrared cameras are often larger in size and consume more energy than those uncooled ones. Referring to this project, an FPGA is used to collect data and power the infrared camera, which makes it impracticable to employ an uncooled infrared module and to mount it into the box.

In contrast, uncooled thermal cameras coupled to an FPGA are viable because they are usually lighter and smaller in size [132]. The infrared detector employed by the uncooled infrared camera in this project is based on the uncooled microbolometer. It is a tiny vanadium oxide resistor with a large temperature coefficient on a silicon element with a large surface area exhibiting low heat capability and good thermal isolation [133]. The variance in ambient temperature causes changes in microbolometer temperature, then the microbolometer converts these changes into electrical signals, which will be processed into the final images, and this relationship is verified in experiments. Furthermore, microbolometer-type infrared cameras are less expensive than their cooled equivalents, making them compatible with multi-task, compact and affordable devices [134]. In surveillance applications, as an example, equipping uncooled microbolometer-type infrared cameras requires less maintenance service than cooled infrared cameras, and the initiating speed is faster as it does not require cooling [135].

### 8.1.2. Related features of the uncooled infrared module XTM-640

In the XTM-640 uncooled thermal camera, there are other two sources of inaccuracies that demand calibration, but they are already corrected by the firmware included in the system and was briefly introduced in section 2.1.3. These corrections are:

- **Non-uniformity corrections:** It refers to a smoothing process carried out to equalize the performance of the individual pixels of a microbolometer. These corrections are automatically triggered when the current uncooled infrared camera is powered on. If pixels that either stop working or whose pixel value vary significantly from the mean value, a correction will be performed based on these pixels' location. The values are obtained from neighbourhood pixels, and the interpolation is calculated to replace the defected pixel. This non-uniformity correction is aimed to deliver a high-quality image visually rather than a high-quality radiometric value.
- **Shutter correction:** The XTM-640 infrared module is equipped with a mechanical shutter. It performs an automatic shutter correction base on the time or change in sensor temperature. The sensor temperature change causes the offset uniformity changes of the detector, which is compensated for by the shutter correction.

These are the corrections that need to be kept when collecting infrared images for radiometric calibration because they cannot lead to temperature measurement errors. However, the histogram control that was mentioned in section 2.1.3 can cause a significant error. The histogram control consists of offset control, gain control and histogram equalization. The offset control alters the midpoint of the output image histogram, and the gain control adapts automatically to use the full dynamic range at the output histogram. The automatic offset mode in combination with the automatic

gain mode maximizes the contrast of the output infrared image and the histogram equalization algorithm making the histogram distribution more uniform. However, these algorithms cause a significant error as the output image has been manipulated by the in-built image processing techniques and cannot reflect the scene's actual radiation. For this reason, these image processing procedures should be disabled, and their corresponding camera register value is needed to be changed.

### **8.1.3. Infrared camera XTM-640 UART configuration**

The current infrared camera XTM-640 is register-based; all the module configurations happen by changing the appropriate registers. The physical interface to the camera module is serial, and a register-based protocol is used for communication over the UART interface, which is already introduced in section 2.1.4. From the previous study, it has been found out that the automatic offset control, gain control, and histogram equalization functions should be switched off.

The corresponding registers addresses and their proper values are found from the datasheet, and a list is created to be accessed by the server program. A sub-module is added to the server program to communicate with the camera according to the serial protocol and change the infrared camera register values. As the infrared camera is powered by the FPGA board, in order to achieve the correct infrared images from the moment that the infrared camera is powered on, the server program should be set to execute when the embedded Linux system boots and the client program should be executed immediately after the server is operating for infrared image acquisition.



#### 8.1.4. System challenges in the absence of infrared detector temperature stabilization

It has been found out that surface temperature measurement requires that the sensor temperature be stabilized to display a reliable result [136]. During the preparation stage of radiometric calibration, a module is added to the server program to acquire the infrared detector temperature along with its corresponding image sets. It has been found out that the infrared camera module detector's temperature is changing after it is powered on. An investigation is carried out to test the detector temperature. A scatter plot is created in Figure 8-1 to demonstrate the infrared detector temperature change over time after the infrared camera being initialized indoor. The infrared detector temperature was tested under various ambient temperatures so that the optimal time for stabilization of the camera can be estimated.

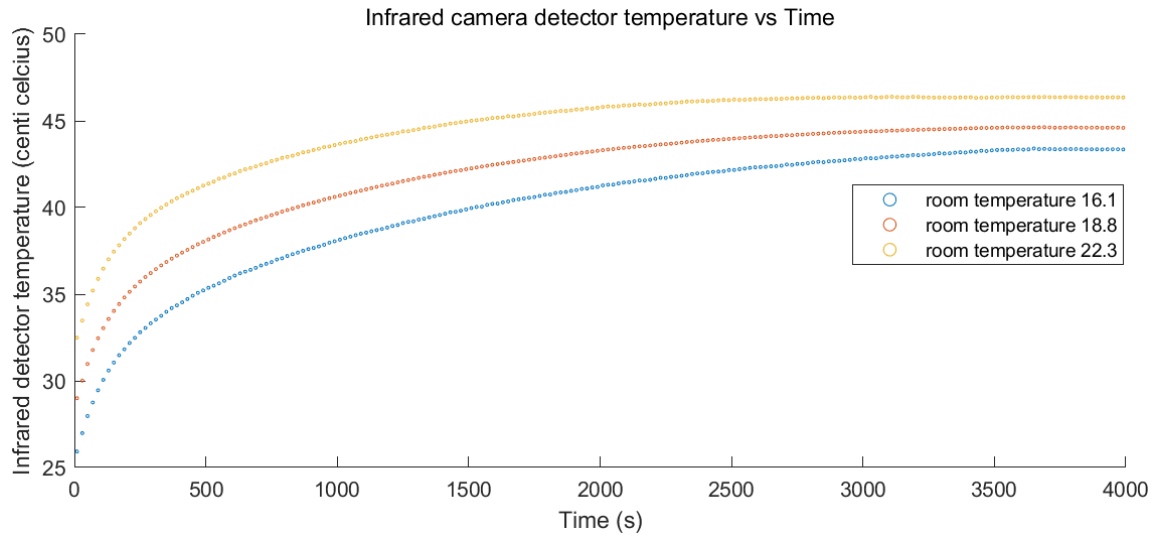


Figure 8-1 Infrared camera detector temperature plotted in the function of time

Because the indoor environment and temperature are controlled, and the air mass stability is high, the detector temperature measured appeared to stabilize after about 55 to 65 minutes. It will remain stable after a period if the environment temperature remains

unchanged. It has been found out that the higher the ambient temperature is, the quicker that the detector temperature reaches a steady condition.

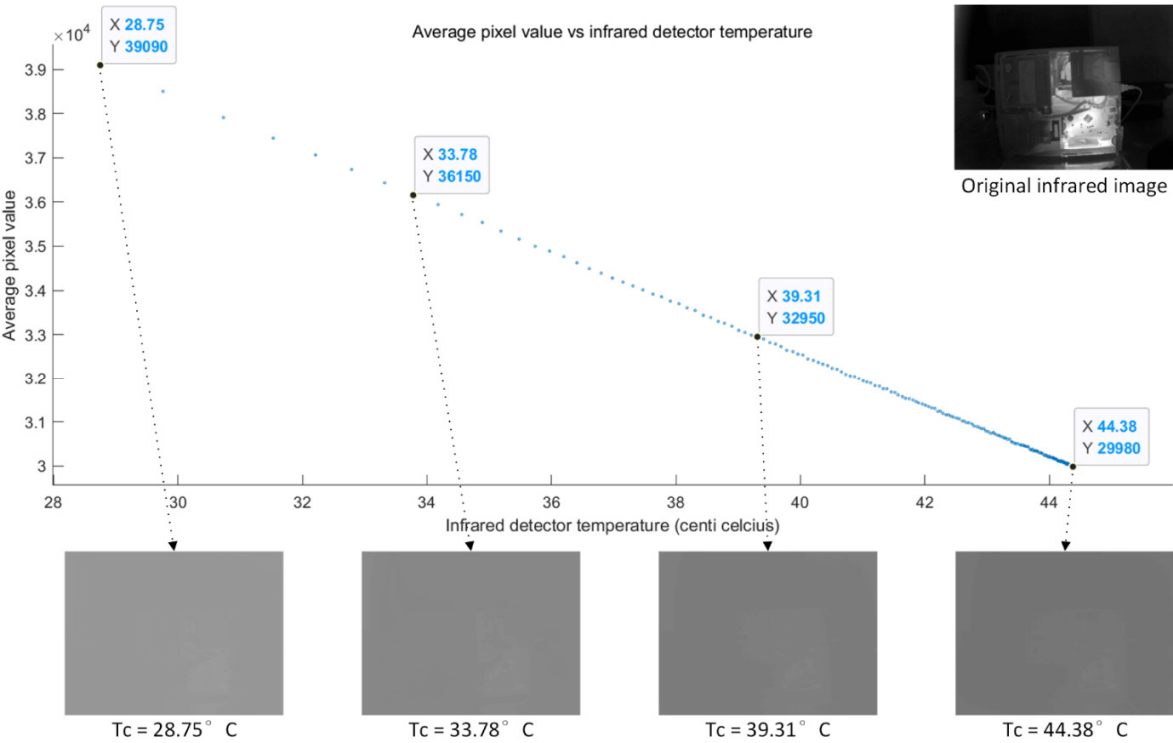


Figure 8-2 Infrared image grey level as the infrared camera heats up

Another experiment has been done with the infrared camera mounted inside the box and facing a still scene to investigate the effect on infrared image pixel grey level when the detector temperature is changing. The gain control, offset control, and histogram equalization is switched off, and several sets of infrared images are taken from the moment the infrared camera is powered on, to the moment that the detector temperature is stable. The average pixel value is calculated to represent the image grey level. Figure 8-2 depict the curve that shows the grey level intensity of the infrared image when the camera is heated up when the room temperature is 18.8°C. The infrared camera module XTM-640 provides a 16-bit output, these permit  $2^{16} = 65536$  different pixel values within range (0,65535). The average pixel value dramatically drops from 39090 to

29980 when the infrared detector temperature increases from 28.75°C to 44.38°C.

The analysis of these infrared image sets shows that the changing of detector temperature causes changes in the pixel value of the same scene; thus, it will affect the temperature measurement getting from these pixel values. A conclusion can be drawn from this figure that the pixel value for the same surface temperature should decrease if the detector temperature increases, and the relationship between pixel value and temperature could be linear. It is because the microbolometer has a negative temperature coefficient of resistance (TCR) [137]. It means that the increment of microbolometer temperature will decrease its resistance, which leads to a reduction in bias voltage; thus, the pixel value decreases as well.

Therefore, images captured when the temperature of the detector is in transient stated should be discarded. Considerably, wind and irradiance would also appear to affect the temperature measurement. If this infrared camera is used outdoor, the cloud cover condition should be observed, and the wind speed should be measured at a nearby weather station.

## **8.2. Radiometric calibration approach**

In order to make any of the applications mentioned in section 8.1 viable, radiometric calibration has to be performed to establish a defined quantitative relation between the image pixel values and the surface temperature of an object. As a result, the infrared camera would become a temperature measurement device, and the temperature can be displayed for each pixel in the thermal image. The incident radiative power is converted to voltage signals by each microbolometer pixel in the infrared camera sensor, and it is then linearly quantised to a grey level intensity. More specifically, radiometric calibration can be defined as an approach to build a quantitatively accurate mapping between grey-

level pixel intensity and the incident radiative power [138]. Converting the pixel value to an estimate of temperature is performed on the ARM processor, and it is achieved by using a model achieved by radiometric calibration.

The output of the current infrared camera is in the form of images rather than radiation estimations. The output infrared images are a kind of quantised temperature estimations regardless of several external factors such as transmission medium, background radiation and object material properties. These factors could affect the relationship between temperature and infrared radiation power. Thus, either sufficient environmental conditions are known, or the radiometric calibration should be conducted under modification of the other environment.

The disadvantage of using an uncooled infrared camera is evident that these microbolometers are not as sensitive and accurate as those in a cooled infrared camera. If an uncooled sensor is not calibrated, the pixel values can only be used to calculate the relative temperature. For most remote sensing applications, the accurate surface temperature of an object needs to be acquired, which emphasizes the necessity of conduct radiometric calibration. Moreover, a practical method should be applied to conduct a proper radiometric calibration of the uncooled infrared camera. The objective of radiometric calibration is to establish a model of the camera's output (the infrared image) to input, so several essential parameters should be simultaneously measured and varied. Two parameters are considered as the model input, which is: digital camera response – pixel value  $p$ , and camera detector temperature  $T_c$ ; the model output is the object surface temperature  $T_o$ . It is beneficial, but not critical, to control the variation of other parameters such as room temperature, air humidity or wind flow.

### 8.2.1. Calibration system configuration

The thermal camera used to obtain the thermal images is the Xenics XTM-640 infrared module, and it was geometrically calibrated. The main features of this camera are a focal length of  $9.6\text{mm}$ , the field of view  $63.1^\circ \times 49.62^\circ$ , an uncooled microbolometer of height 640 and width 480, a pixel size of  $17\mu\text{m}$ , a spectral band ranging  $8\sim 14\mu\text{m}$ , and a weight of approximately  $89\text{g}$  with a lens installed. According to factory specification, this camera has not been calibrated by the factory. It is beneficial that this thermal camera has one in-built thermistor to measure the infrared detector's temperature. The reading of detector temperature can be accessed via serial communication and can be reliably tracked in real-time.

Usually, the thermal camera's radiometric calibration is carried out using a blackbody as an input source. A blackbody source is an ideal physical object that absorbs all incident radiations to ensure that all the radiation from its output window is its emission. Moreover, the temperature of blackbody equipment can be digitally adjusted to acquire a more accurate reference temperature. However, due to the limitation of test rigs, the radiometric calibration is performed without any costly equipment. The equipment required for this radiometric calibration is only an object whose temperature can be easily measured and controlled so that the target temperature can be varied in a range and handled by the radiometric model. This experiment uses water as the target item as the water temperature is easy to adjust, and the temperature range can be adjusted from  $0^\circ\text{C}$  (ice water mixture) to  $100^\circ\text{C}$  (boiling water). This temperature range is able to cover most of the applications; any out-of-range temperature would be able to be predicted using the resulting radiometric model.

The setup of the calibration system is depicted in Figure 8-3. The infrared camera was mounted in a fixed position against the saucepan at a distance of  $650\text{mm}$ . The sauce

pan is used to contain water, and an induction hob is used to heat the water. An alcohol thermometer is hanged inside the water to avoid direct contact with the saucepan, and the reading from the alcohol thermometer is adopted as the object surface reference temperature  $T_o$ .

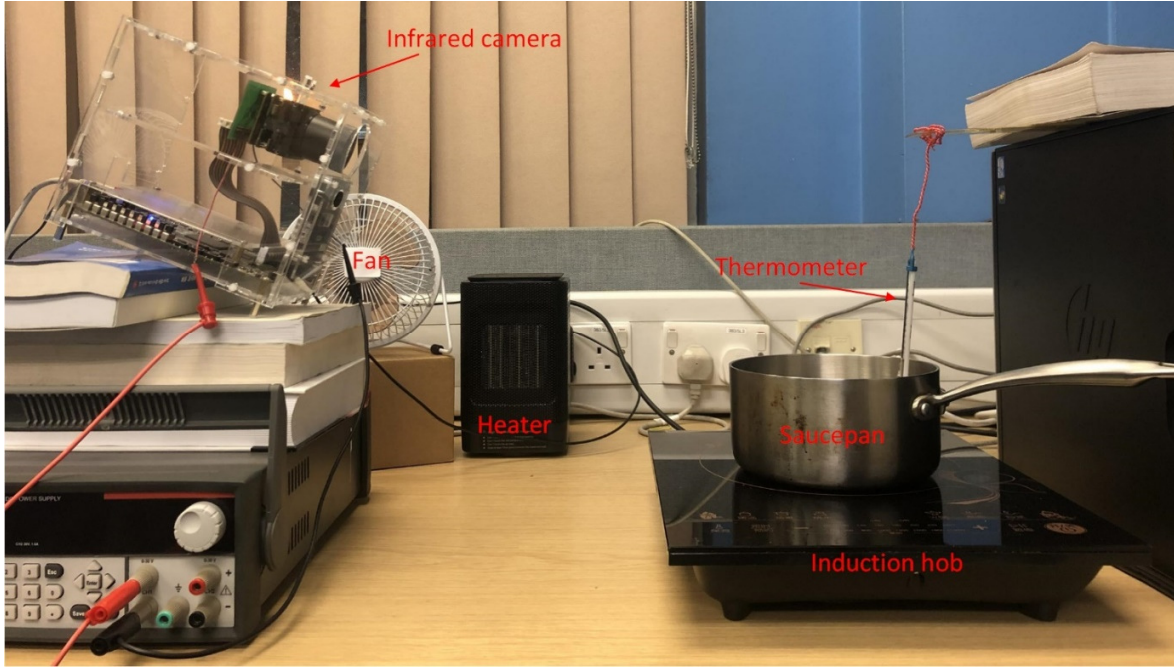


Figure 8-3 Setup of the temperature target and the camera for calibration

The ambient temperature is considered the main additional parameter that can be varied to develop a more comprehensive temperature model. A temperature control environment can be easily achieved using air conditioners, heaters, and fans. The experiment is carried out in a room that temperature is modified to induce low and high ambient temperatures with the cooling/heater system, and the fan is used to control the airflow around the infrared camera so that the detector temperature can be modified more specifically. Therefore, a wide temperature range of the infrared detector can be achieved to generalize the calibration model.

The temperature change of the infrared detector can be model in its transient state;

however, it is not reliable to form a generalized model with the internal temperature is changing rapidly along with the pixel value changing at the same time. Another easy way is to wait for a short period, usually between 60 to 70 minutes, for the infrared detector temperature to stabilise before image capture. Therefore, the radiometric calibration in this work is generally performed in and for steady-state condition only.

The proposed method above takes inspirations from the work of [139] and [140], but performs without the need for blackbody equipment. The camera's digital response is in the form of pixel values, and a relationship will be established between these values and the infrared radiance, which is represented as the surface temperature. The relationship model should enable the accuracy of temperature estimation for various environmental demands and applications, under the condition that the environmental conditions and object surface properties are known or reasonably assumed.

### **8.2.2. Data acquisition**

Many associations will be acquired with a sufficient amount of data collected between different grey-level pixel values, different operating conditions, and different target temperatures. A temperature model can be deducted from those data, which allows future pixel values to be mapped to an estimation of surface temperature based on the information about the operating conditions, particularly from the measurement of infrared detector temperature.

The data collecting process has been highly automated, with control, image streaming and processing implemented in C++, Python and MATLAB, benefiting from the server and client program. The exceptions are the requirement of human-in-the-loop (HITL) for turn on the induction hob to heat the water and record water temperature every few minutes. The infrared camera has to be manually powered off periodically to let it cool

down to enable the detector temperature to be adjusted for the next set of images capturing.

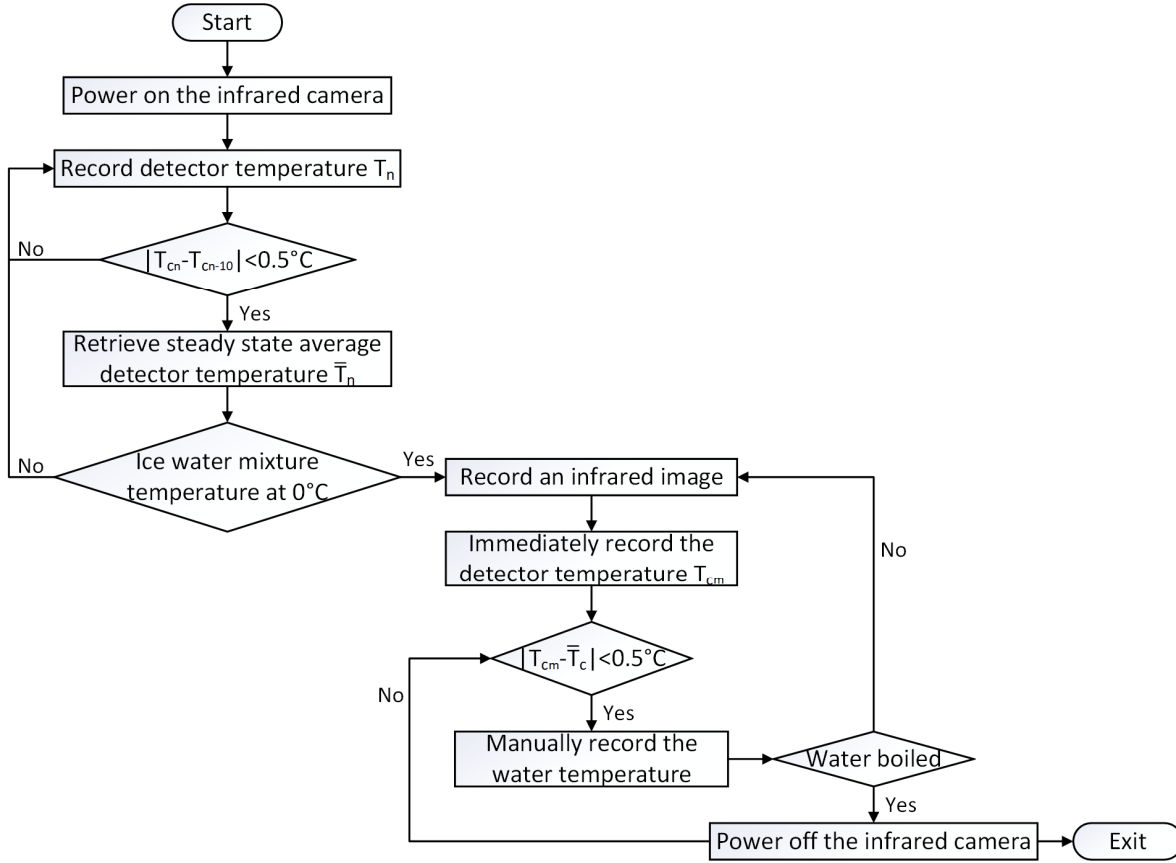


Figure 8-4 Data capture procedure for infrared camera XTM-640 radiometric calibration

In case that the detector temperature changes unexpectedly, the temperature of the infrared detector is measured, recorded, and checked for each image acquired. The detector temperature is adjusted in the range between  $22^\circ\text{C}$  and  $55^\circ\text{C}$  in a step of  $3^\circ\text{C}$ . The proposed data acquisition process is illustrated in Figure 8-4. The infrared camera should be in a proper position so that the field of view covers the saucepan, and the microbolometer's temperature should be determined first. The server program keeps retrieving the temperature of the infrared detector every 30 seconds for comparison until the temperature difference between  $T_{cn}$  (the  $n^{th}$  temperature being recorded) and  $T_{cn-10}$



(the temperature being recorded 5 minutes ago) is less than  $0.05^{\circ}\text{C}$ . Then, another ten temperatures of the detector are recorded, and the average temperature value is calculated as  $\bar{T}_c$ , which is also be defined as the steady-state detector temperature.

After that, a portion of ice is poured into the saucepan along with a certain amount of water to cover the ice cubes. The capturing process starts when the thermometer inside the ice water mixture has a reading of 0. The temperature of the infrared detector is measured and recorded for each image acquisition event, and these temperatures are used to compare with the steady-state temperature  $\bar{T}_c$  in case that the detector temperature changes incidentally. This data acquisition algorithm results in a large number of infrared images with the corresponding target (water in a saucepan) and the infrared detector temperature, which can be adopted to generate the temperature model.

A lower number of infrared images are captured for detector temperature lower than  $30^{\circ}\text{C}$ , which corresponds with those obtained in the cooling room with a fan assist system; and for detector temperature higher than  $50^{\circ}\text{C}$ , which corresponds with the using of a heater as higher ambient temperature could damage the camera module itself.

### 8.2.3. Temperature model

In the XTM-640 uncooled thermal camera, which consists of thermo-electric cooler (TEC)-less microbolometers, the microbolometers cannot be stabilized to a constant temperature if there is no external temperature control. It makes the camera temperature changes along with the microbolometer sensors' temperature and causes the pixel value change for the same surface temperature [141]. This fact should be taken into account in the pixel-temperature calibration model.

In most research, a linear model is considered to regulate the object's surface temperature and the pixel value output from the camera [139]. Also, it has been found

out in the previous experiment that the pixel value response of the camera is affected by the detector temperature more linearly. In that case, the linear polynomial model to relate the pixel value with temperature should be considered first because polynomial models are a great tool to estimate and predict the response values over a range of input parameter values, especially there are two input values in this experiment. The quadratic polynomial models are also considered in this study if the assumption is incorrect, and there may be some curvature in the camera's response. Also, the cross-product terms  $T_c p$  can be used to model the interactions between the detector temperature and the pixel values. These two kinds of models are compared, so one can be chosen to represent the correct relationship. The linear polynomial model can be written as:

$$T_o = a_{10}T_c + a_{01}p + a \quad (9-1)$$

Moreover, the non-linear polynomial model is:

$$T_o = a_{10}T_c + a_{20}T_c^2 + a_{01}p + a_{02}p^2 + a_{11}T_cp + a \quad (9-2)$$

where  $T_o$  is the object surface temperature,  $T_c$  is the infrared detector temperature,  $p$  is the pixel value, and  $a_{mn}$  are the polynomial regression coefficients.

One thing that has to be noticed is the actual infrared images provide views of non-uniform-temperature water surface, which could be caused by radiation interference from other objects. Thus, all pixels in a single image representing the same water surface are segmented, and the average grey level of these pixels is calculated. This process also prevents accidentally choosing a bad pixel to represent the water surface. An assumption has been made that the thermometer reading indicates the water surface temperature; the average pixel value is associated with the estimated surface temperature, be used as  $p$  in the equation (9-1) and (9-2). Therefore, for a range of thermometer readings, the grey level corresponding to a water surface temperature is achieved. Figure 8-5 shows the plot of water temperature and its corresponding average pixel values. Each curve

shows the average grey level of the infrared image segmented area as the water heats up, for different infrared detector temperature.

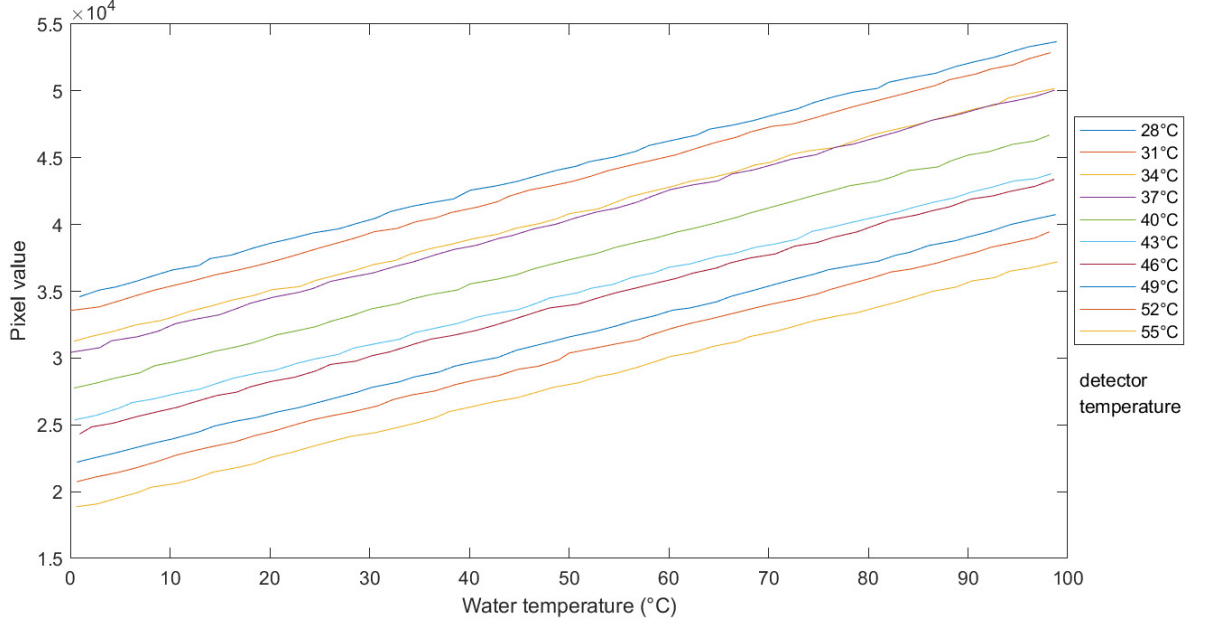


Figure 8-5 Radiometric calibration sample curves

To determine the most suitable model, 65% of the data are used for solving the polynomial models, and 35% are used to validate the resulting model. Furthermore, the calibration and validation sets are randomly chosen to ensure that both data set can cover the full range of the measurement. The relationship between the pixel grey level, detector temperature, and object surface temperature can be learned by solving the multiple linear regression model. The temperature models is shown below:

$$T_o = 3.0621T_c + 0.0051p - 265.1156 \quad (9-3)$$

and

$$\begin{aligned} T_o = & 4.5185T_c + 0.0175T_c^2 \\ & + 0.0054p - 3.8961 \times 10^{-9}p^2 \\ & + 1.0307 \times 10^{-7}T_cp - 299.444 \end{aligned} \quad (9-4)$$

Reasonable limits have to be determined for both temperature models to constrain the infrared detector temperature and the pixel value. These limits can define the input

range for each model, so estimating object surface temperature is valid. For this experiment, the infrared camera module operation temperature is from  $-40^{\circ}\text{C}$  to  $60^{\circ}\text{C}$ , the pixel grey level is in the range of (0,65535). Thus, the measured surface temperature limits should be defined in the full range below and above the minimum and maximum recorded pixel values and detector temperature. The highest temperature that this infrared camera can measure is  $252.61^{\circ}\text{C}$  base on the linear model (9 – 3), and  $245.35^{\circ}\text{C}$  base on the non-linear model (9 – 4).

These two polynomial models can be visualized in 3-D planes, as illustrated in Figure 8-6. These mesh plots present a mapping between pairs of pixel grey-level and detector temperature, and object surface temperature.

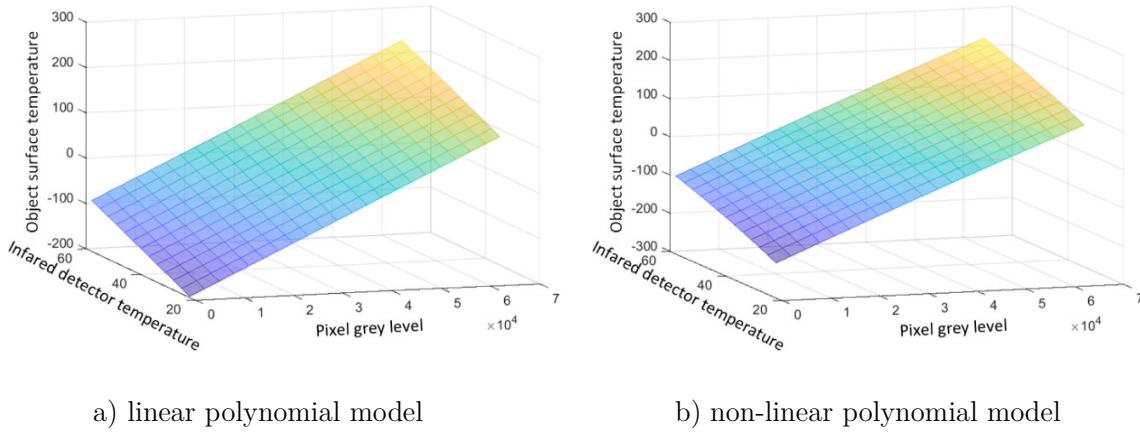


Figure 8-6 Temperature model 3-D mesh plot

### 8.3. Evaluation

After conducting the radiometric calibration approach, two models are generated to represent the relationship between pixel value, infrared detector temperature and object surface temperature. Some statistic methods can be utilized to analyse the model's adjustments so that a better model can be chosen to define the relationship correctly. Further verification experiment should be performed to evaluate the selected model.

### 8.3.1. Analysis of residuals

Because two different temperature models are being proposed to relate the pixel value to the object surface temperature, the one that has higher measurement accuracy should be selected so that a more precise surface temperature can be estimated. There are in total 500 images captured by the XTM-640 infrared camera module after changing the configurations of the manufacturer;  $500 \times 65\% = 325$  images are used to generate the two different temperature model, and  $500 \times 35\% = 175$  images are used in the comparison of the adjustment of the linear and non-linear polynomial models.

One statistic way to analyse the temperature polynomial regression coefficients  $a_{mn}$  is to compute and compare the coefficient of determination  $R^2$  and its adjusted version named as adjusted  $R^2$ . The value coefficient of determination is used to judge the goodness of fit in a polynomial regression model. The  $R^2$  is based on the proportion of the variability of the response variables, which can be calculated if a set of explanatory variables is given. The  $R^2$  represent the square of the multiple correlation coefficient between the response variable and all the explanatory variables included in the polynomial regression model [142]. In this experiment, the research variable is the object surface temperature; the explanatory variable is the pixel grey level and infrared detector temperature, which the researcher can manipulate. If there are a total number of  $n$  observations, the equation of coefficient of determination is written as:

$$R^2 = \left[ \sum_{i=1}^n \frac{(O_n - \overline{O_n})(S_n - \overline{S_n})}{\sqrt{\sum_{i=1}^n (O_n - \overline{O_n})^2 \sum_{i=1}^n (S_n - \overline{S_n})^2}} \right]^2 \quad (9 - 5)$$

where  $O_n$  are the observed object surface temperature values, and  $\overline{O_n}$  is the average value of the  $n$  observed temperatures;  $S_n$  is the simulated object surface temperature values that calculated from the generated polynomial model above, and  $\overline{S_n}$  is the average of

simulated values. The adjusted coefficient of determination  $R_{adj}^2$  is calculated from:

$$R_{adj}^2 = 1 - \frac{(1 - R^2)(n - 1)}{n - k - 1} \quad (9 - 6)$$

where  $n$  is the number of observations, and  $k$  is the number of explanatory variables in the polynomial models.

Both  $R^2$  and  $R_{adj}^2$  are utilized for temperature model selection. For the water surface infrared images, there are two fitted models available. The model with the least lack of fit should be preferred, which can be determined depending on the value of the coefficient of determination or its adjusted version. In the best case,  $R^2 = 1$  means that the simulation model matches precisely with the observations. Moreover, the closer the value of  $R^2$  to 1, the better the model is.

Another statistic approach to evaluate the two different temperature models' performance is to compare their root mean square error  $RMSE$ . The  $RMSE$  represents the sample standard deviation of the difference between simulated and observed values. It also describes the degree of coincidence among the model's value and the measured data [143]. The  $RMSE$  is calculated from the square root of the mean of the square of all of the errors, and the formula to calculate it is:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (S_n - O_n)^2}{n}} \quad (9 - 7)$$

where  $S_n$  is the simulated object surface temperature values, and  $O_n$  is the observed object surface temperatures. The absolute fit of the model is indicated by the value of  $RMSE$ , and lower values of  $RMSE$  indicate better fitting of the temperature model.

Table 8-1 Main statistic indices of the linear and the non-linear polynomial model

	Linear model	Non-linear model
Coefficient of determination $R^2$	0.99485	0.99293
Adjusted coefficient of determination $R^2$	0.99479	0.99290
Root mean square error $RMSE$ (°C)	2.07114	2.42625

Table 8-1 illustrate the statistics that represent the measuring accuracy of the two different temperature models. It can be observed from the table that the coefficient of determination of the linear model is closer to 1 than the linear model, and the *RMSE* decreases  $0.35^{\circ}\text{C}$  for the linear model. In this case, the most suitable model should be the linear polynomial model present in the equation (9 – 4), with an *RMSE* of  $2.07^{\circ}\text{C}$ . The usage of the linear temperature model to convert the pixel value to surface temperature should have an expected final precision of  $\pm 4^{\circ}\text{C}$ .

### 8.3.2. Evaluation and verification experiment

Since the linear polynomial model is chosen to represent the relationship between infrared image pixel value and object surface temperature, some evaluation and verification experiment should be performed to reveal the chosen temperature model's performance. This experiment involves testing the temperature model when applying it to a kettle's surface temperature. There is water in the kettle, so the kettle's surface temperature can be set to various values just by boiling the water inside.

In order to obtain reference temperatures, an infrared thermometer is used to measure the surface temperature of the kettle. The infrared thermometer module is FLIR TG167; it is a commercial thermal camera that can carry out spot temperature measurement. Several sets of images are taken from both cameras when the kettle is boiling; some are depicted in Figure 8-7.

As the FLIR TG167 only measures the surface temperature of a spot, the temperature conversion from the linear polynomial model attempts to use the same spot of the infrared images taken by XTM-640. The surface temperature is calculated and shown in the figure next page. Supposing that the FLIR thermal camera measurement is correct, it can be seen that the proposed temperature model achieves an accuracy within  $4^{\circ}\text{C}$ .

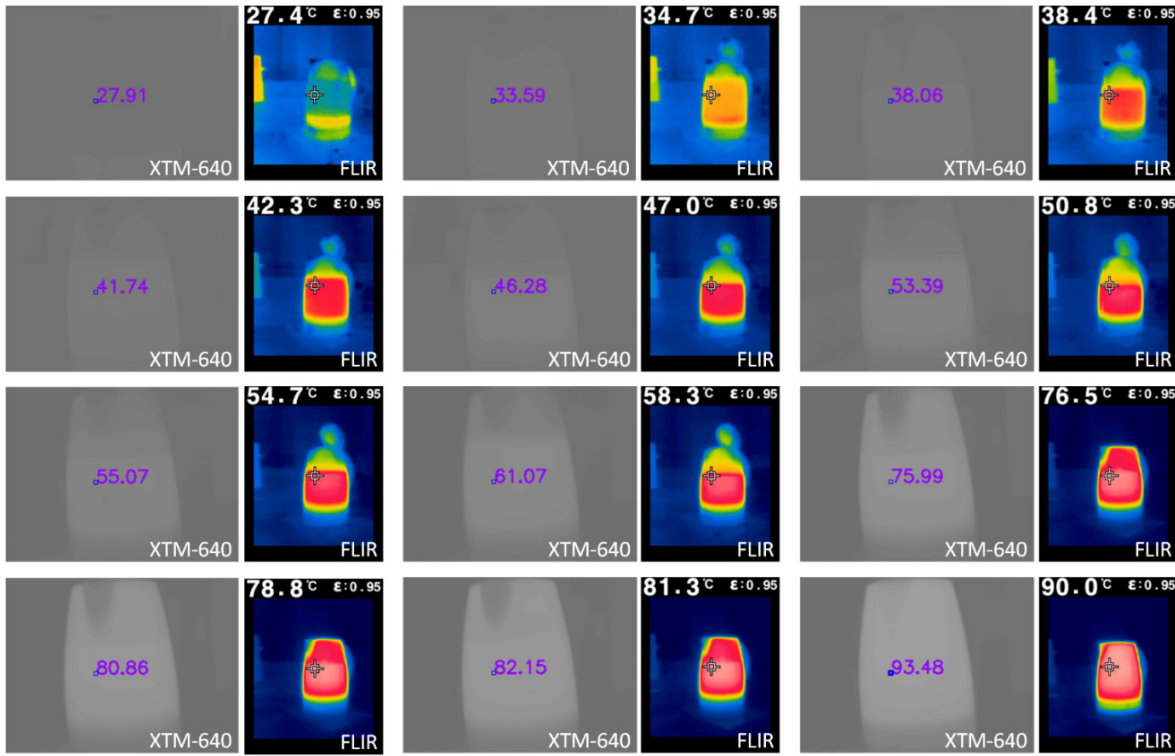


Figure 8-7 Kettle surface temperature measurement from XTM-640 and FLIR TG167

### 8.3.3. Limitations

The proposed approach to measure object surface temperature is based on converting the infrared image pixel value to the corresponding temperature using a temperature model. The infrared image is produced by a thermal camera that can detect infrared radiation from an object. Infrared radiation comes from all objects at temperatures greater than 0 degrees Kelvin. A piece of ideal equipment to conduct this experiment above should be an ideal blackbody because it absorbs all incident infrared radiation, and as a result, it reflects none. For that reason, all the radiation emitting from the surface of an ideal blackbody is due to its temperature. In the meantime, if the medium between the infrared sensor and the blackbody transmit all the radiation, the thermal cameras can be effectively measuring the surface temperature from any distance [144].



However, in reality, the earth's atmosphere cannot transmit all the radiation and there is a specific amount of energy is absorbed by molecules in the air [145]. This experiment uses water to replace the blackbody source, and water is a non-ideal material. The thermal radiation emitted from all solid and liquid can be written according to the Stefan-Boltzmann Equation:

$$E = \epsilon \sigma T^4 \quad (9 - 8)$$

where  $E$  is the radiant emittance which has a unit of  $W/m^2$ ,  $T$  is the absolute temperature of the object's surface,  $\sigma = 5.67 \times 10^{-8}$  is the Stefan-Boltzmann constant. Most importantly,  $\epsilon$  is the object's emissivity, and it usually depends on the object material. In this experiment, the emissivity of the water surface is assumed to be close to 1. However, in practice, the value of  $\epsilon$  depends on water transparency and surface smoothness [146]. The transmission of infrared radiation is affected by the transmission medium. Due to both absorption and scattering effect, atmospheric extinction or power attenuation occurs if the transmission distance increases [147].

The choice of using water to take the place of a blackbody source is made base on the excellent emissivity of water. During the experiment, the thermometer measures the water temperature inside the water. Another assumption has been made that the temperature inside the water equals the temperature of the water surface. Because the water surface has direct contact with air, when the air temperature not the same as the water temperature, heat transfer will happen, therefore the water surface temperature could be interfered by the surroundings.

## 8.4. Temperature mapping and data visualization

One goal of this radiometric calibration experiment is to perform accurate surface temperature mapping on the infrared image and create a combination view using the

mapped infrared image with the visible spectrum information to spread the heat over the target area be visualised. The resulting temperature representation of the infrared image can then be used for various applications where the temperature difference is needed to be known, and the estimation of surface temperature must be acquired. The potentially dangerous hotspots or cold spots can be clearly visible under the temperature mapped infrared image. The histogram of the raw infrared image should be adjusted so that the whole image's contrast is improved and the target can be observed more clearly.

#### 8.4.1. Infrared image contrast enhancement

As mentioned in section 2.1.3, the infrared module XTM-640 has a built-in histogram equalization process for image contrast adjustment. However, due to radiometric calibration requirement, the histogram equalization function is switched off because the histogram equalised pixel value cannot be directly linked to the surface temperature. For that reason, the raw image output format should be kept for surface temperature estimation, and the histogram equalization algorithm can be implemented on the client program after retrieving the raw data.

Conventional histogram equalization is used to easily distinguish the detailed pixels in a large quantity of low-contrast background pixels. The remapping function to spread out the pixel distribution values is the cumulative distribution function (CDF). Let  $f$  denote the captured infrared image represented as a  $640 \times 480$  matrix of integer pixel intensities ranging from  $min = 0$  to  $max = 65535$ . Let  $h$  be the normalized histogram of  $f$  with a bin for each possible intensity, the probability mass function of all the pixel in this image is:

$$h_n = \frac{\text{number of pixels with intensity of } n}{\text{total number of pixels}}, \quad n = 0, 1, \dots, max \quad (9 - 9)$$

The histogram equalized infrared image  $h$  then can be defined by:

$$g_{i,j} = \max \times \sum_{n=0}^{f_{i,j}} h_n \quad (9 - 10)$$

where  $(i,j)$  indicates the pixel locations of the pixels. The equation (9 – 10) is equivalent to transform the infrared image pixel intensity  $p$  by the function:

$$T(p) = \max \times \sum_{n=0}^p h_n \quad (9 - 11)$$

The transformation above comes from taking the pixel intensities of image  $f$  and  $g$  as continuous random variables  $X, Y$  on  $[0,65535]$  with  $Y$  defined by:

$$Y = T(X) = \max \times \int_0^X h_x(x)dx \quad (9 - 12)$$

where  $h_x$  is the probability density function of  $f$ ,  $T$  represents the CDF of  $X$  multiplied by 65535.

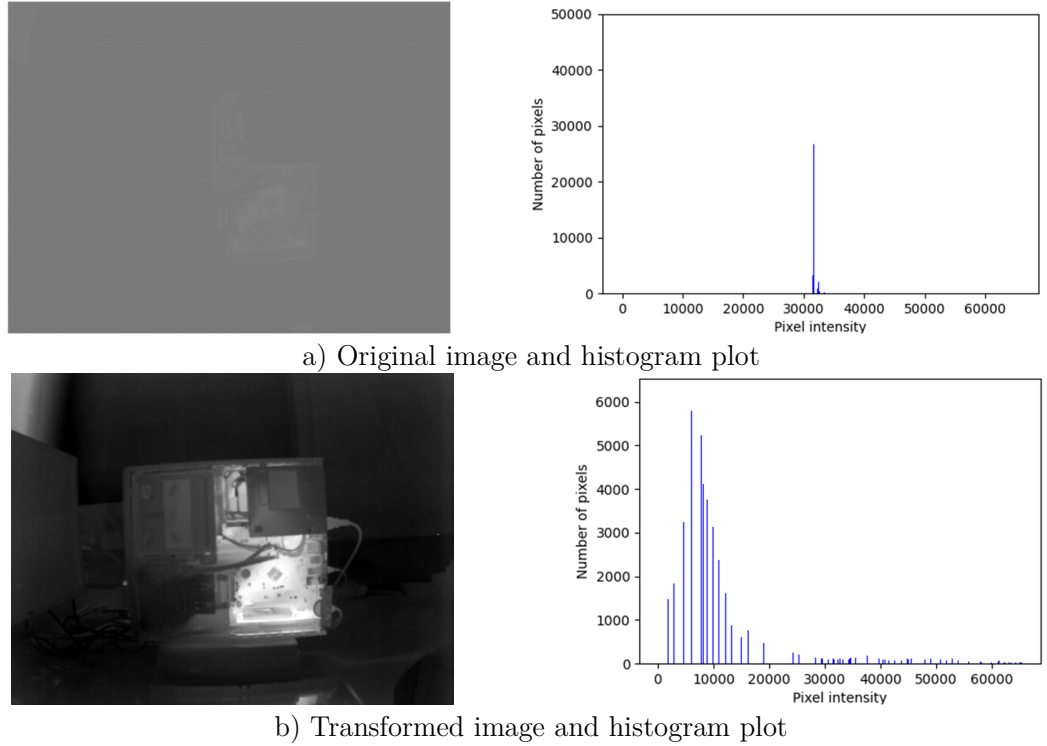


Figure 8-8 Histogram equalization applied to low contrast infrared image

For simplicity, assuming that  $T$  is differentiable and invertible, it can then be seen that  $Y$  defined by  $T(X)$  is uniformly distributed on  $[0,65535]$ . The result of applying equation (9 – 10) is shown in Figure 8-8. It can be clearly seen that the contrast of the

original infrared image has been improved.

The cumulative distribution function can guarantee that the relationship between each pixel remains the same. The brighter part still stands for a warmer area, and the darker part stands for the colder area. The infrared image is in the 16-bit format before and after the histogram equalization process, and the pixel intensity is kept in the range of (0,65535).

#### **8.4.2. Pseudo-colour mapping for data visualisation**

As human vision is limited to the visible spectrum, the information captured by thermal cameras is not directly perceptible to human eyes. Therefore, a straightforward method is an attempt here to represent information from these infrared sensors to visualise the infrared images better. The infrared image output from the module XTM-640 is monochromatic data, as the microbolometer only response to the wavelength in the infrared range. Thus, the infrared images are displayed in greyscale, and the pixel intensity is scaled between black (cold) and white (hot). From the data output of the infrared sensor, there are 65536 different shade of grey, whereas human eyes can typically distinguish only around 100 different shade of grey [148], which is significantly less than the pixel grey levels.

Because human eyes and brains are capable of perceiving many times more different colours than the shades of grey, applying a pseudo-colour mapping scheme to the monocoloured infrared image can help observers extract more information from the representation. Pseudo-colour mapping is a process that can convert the greyscale infrared image to a coloured one. Moreover, as long as the surface temperature is known for each pixel value, a colour bar is usually added so that pixel RGB values can be associated with the approximate temperature estimations; the relatively warmer or cooler

values can be known easily from the colour mapped infrared images.

Like the FLIR TG167 thermal camera model, many thermal cameras manufacturers have added several pseudo colour palettes that can be applied to their greyscale infrared image output in real-time. The most widely used pseudo-colour palettes are rainbow colours and blackbody colours, namely, the colour changing when the wavelength of light from blackbody radiation changes [149]. Figure 8-9 shows the examples of the two typical colour mapping schemes on an arbitrary infrared image taken in the lab.

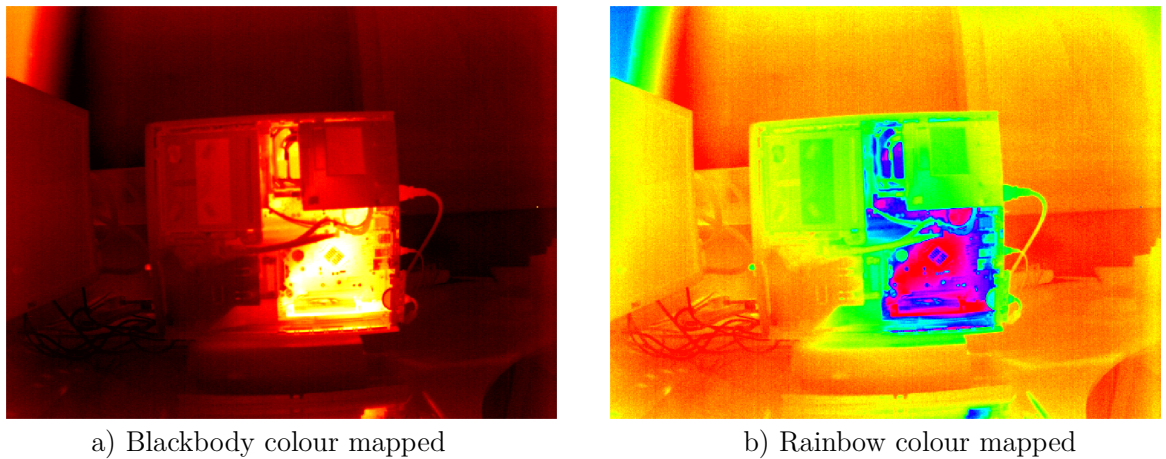


Figure 8-9 Colour mappings to enhance the greyscale infrared image

The blackbody colour mapped infrared image are designed to be perceptually uniform. Specific colour brightness and hue are chosen, and then the filling colours are adjusted according to the International Commission on Illumination  $L^*a^*b^*$ (CIELAB) colour space so that the luminance is linear. Although it is an effective colourmap, many users would find it more useful and appealing to have more hues, such as the rainbow colourmap. However, the main criticisms of the rainbow colourmap are that it may cause visual confusion as the lack of perceptual ordering can slow down the observing because users have to refer to the colour bar more often to interpret the data [150].

A novel colour mapping scheme has been designed to improve the capability of infrared image visualisation. This colour map is created on the basis of the CIE 1976

$L^*u^*v^*$  (LUV) colour space shown in Figure 8-10. The brightness and chromaticity difference between every two colours is proportional to their perceived difference in the CIELUV colour space. The primary advantage of using CIELUV colour space is that it can be represented in a chromaticity diagram [151], and colours can be located more quickly in a figure [152]. The design of the new colourmap should be easily discerned between different temperatures. The association between hue and temperature difference should be intuitive, and proper balance should be given between those colours.

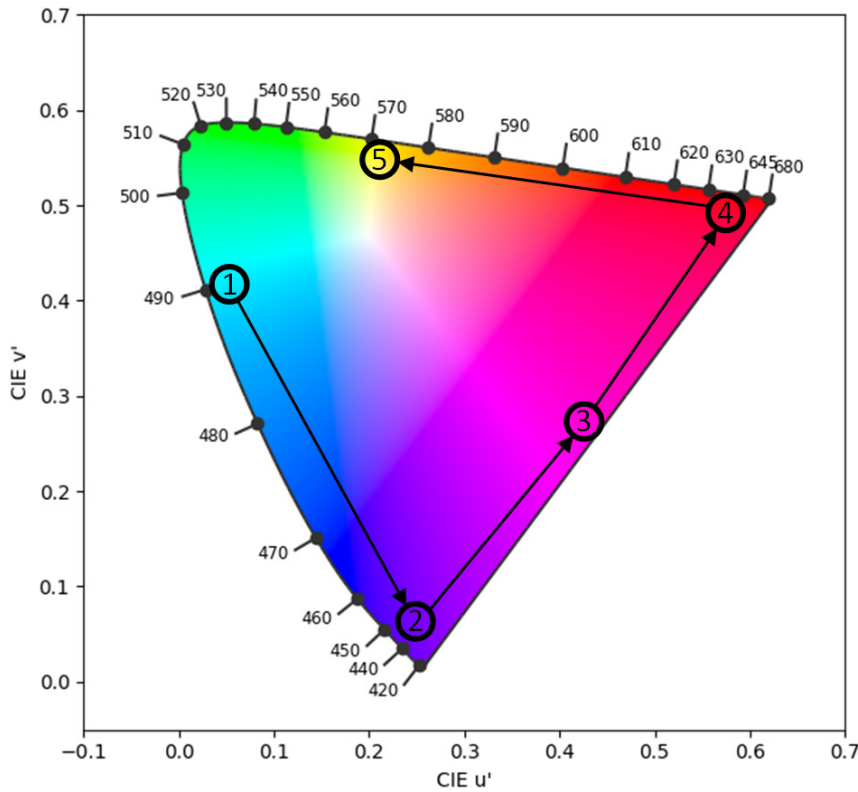


Figure 8-10 CIELUV colour space and sampling points

The five sampled points in Figure 8-10 are used to create the colour palette. #5 is bright yellow representing warmer temperatures, whereas #1 is icy blue, representing colder temperatures. The colourmap is created in the order of the labelled points. The choice of colours is carefully made to avoid the CIELUV colour space edges to create a printer-friendly colour palette. The distribution of these sampling points can increase the

number of unique temperatures in the image effectively, and the difference in hue allows the observer can distinguish the temperature difference. Because the temperature model is linear, linear interpolation is implemented to create the colours in between the five sampled points so that the transition is smooth. An observer is usually looking for targets that are particularly hot or cold in an infrared image but not on moderate temperature items, so the brightness of #2, #3 and #4 from the CIELUV colour space has been reduced. Furthermore, as the infrared image is in 16-bit format, the colour palette is designed to have enough resolution for the grey scaled representation of the infrared image. The analysis of the novel colourmap is illustrated in Figure 8-11.

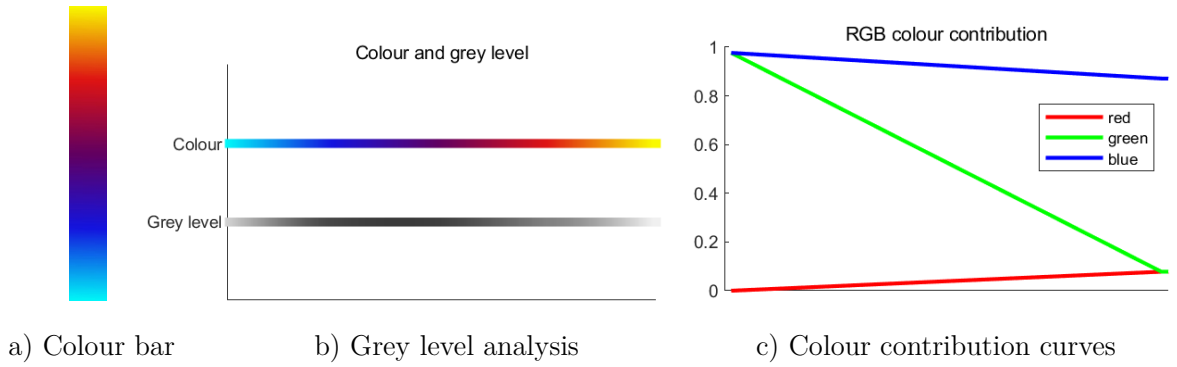


Figure 8-11 Analysis of the novel colourmap

### 8.4.3. Comparison between colour palettes

In this work, many colour palettes are used to implement the colour mapping of the grey level infrared images. Two typical colour maps are used on different colour range infrared images compared to the proposed colour map; the comparison is illustrated in Figure 8-12. The colour bar is created, and the colours are converted to temperatures base on the raw infrared image directly retrieved by the client program and the linear temperature model.

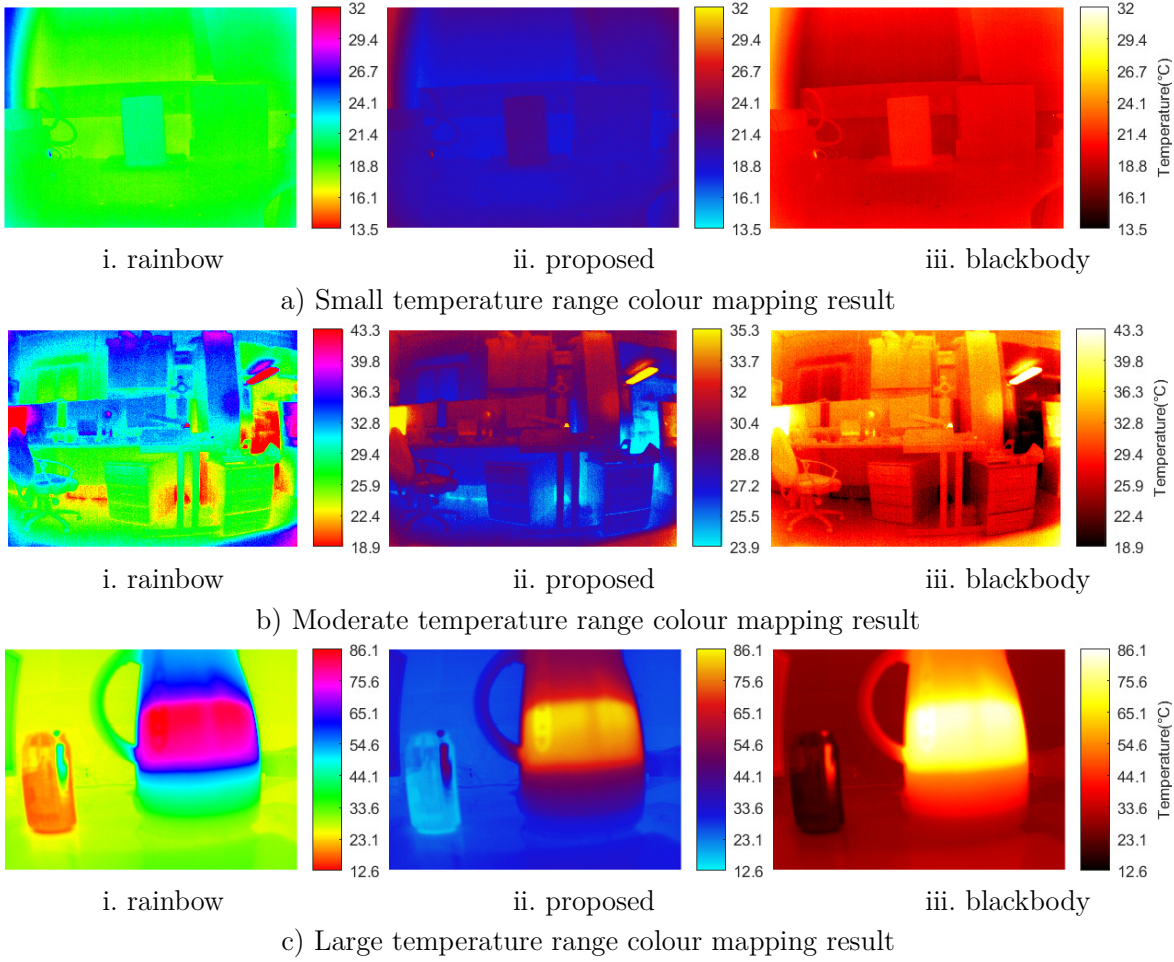


Figure 8-12 Colourmap comparison of different temperature ranges

Although the rainbow colour map usage allows the infrared image to be converted in a more detailed way, the artificial discontinuities in colour mapped image will exaggerate the temperature difference in the scene. Without referring to the colour bar, it can be difficult to perceive which surface is warmer or colder as the nature of colours in the rainbow colour palette is not particularly intuitive. On the contrary, the blackbody colour mapped image has intuitive colour shades so that the temperature information can be recognized without noting from the colour bar. However, it is hard to distinguish small temperature changes. As can be seen from the infrared images above, which use the novel colour map, the temperature details are preserved, and the colours vary smoothly in between. The proposed colour map can achieve better perceptual balance than the



rainbow and blackbody colourmaps.

#### 8.4.4. Infrared and visible image combination

To produce an overlaid view of the infrared image and the visible image, alpha blending [153] [154] is implemented to create a new pixel from the two source pixels. The advantage of the alpha compositing algorithm is that it can achieve partial transparency of both source images, and the transparency level can be easily alternated [155]. Supposing that the pixel location is  $(x,y)$  on the two source images, the destinate pixel value can be written as:

$$dst_{x,y} = \alpha \times src1_{x,y} + (1 - \alpha) \times src2_{x,y} \quad (9 - 13)$$

where  $dst_{x,y}$  represent the pixel value at location  $(x,y)$  on the overlaid image,  $src1_{x,y}$  and  $src2_{x,y}$  represent the two source images. The value of  $\alpha$  is chosen to be 0.5 so that a weight balanced overlaid image can be created. All the images are stereo rectified, and the combinational view of the visible image and the colour mapped infrared image is shown in Figure 8-13. To find the corresponding values for the colour bar, the greyscale visible spectrum intensity is used to determine the luminance. The luminance of the visible image is remapped from (0,1) to (0.1,0.9), so those dark or bright outliers does not obscure the colour to represent the temperature.

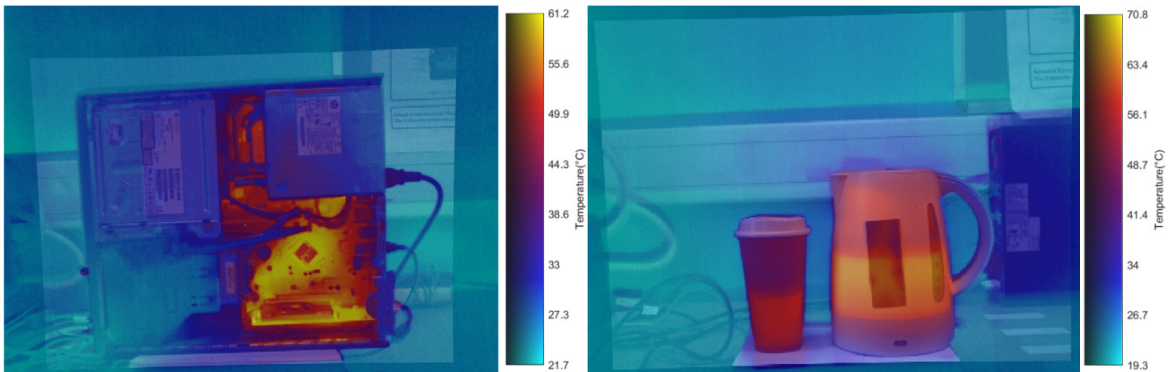


Figure 8-13 Combined views of the visible image and colour mapped infrared image

Comparing to the greyscale infrared image in Figure 8-13a, the colour mapped infrared image in Figure 8-13b clearly displayed the temperature variation over the scene, including hot spots such as kettle, cup, and motherboard. The visible vision in Figure 8-13c reveals much more texture details hidden in the infrared image, such as the text on the computer case and the scale on the kettle. These potential texture details can help with some practical inspection application as they are more identifying than the infrared information.

#### **8.4.5. Temperature thresholding**

A segmentation algorithm can be meaningful to divide the infrared image into regions to highlight some critical part to serve particular purposes [156]. This study also aims to create a temperature thresholding-based thermal image to combine with the visible image. The thresholding algorithm should be developed in accordance with the infrared camera characteristics, particularly the temperature model derived in section 8.2.

The purpose of developing a threshold view of the infrared image is to filter out anything in the scene that are particular warmer so that an alarm can be created to indicate the signal of dangerous. As the grey-level pixel value can be related to the surface temperature by using the linear temperature model, the first step of infrared image thresholding is to determine a threshold temperature and convert it to a certain pixel value. One thing to be noticed is that the thresholding pixel value should be calculated in real-time. Because if the infrared detector temperature is varying, the corresponding pixel grey level should be changing. The temperature dependence correction should be performed to minimize the effect of the sensor temperature on the response of the sensor. Furthermore, the infrared image has been histogram equalized, so the thresholding pixel value should then be converted to its corresponding value after the histogram equalization.

The process of creating a finalised thresholding overlaid view of the visible image, and the infrared image can be illustrated in the flowchart in Figure 8-14.

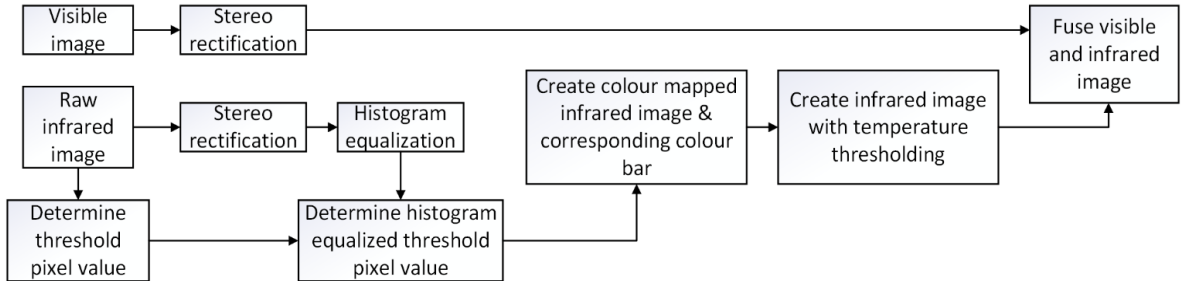
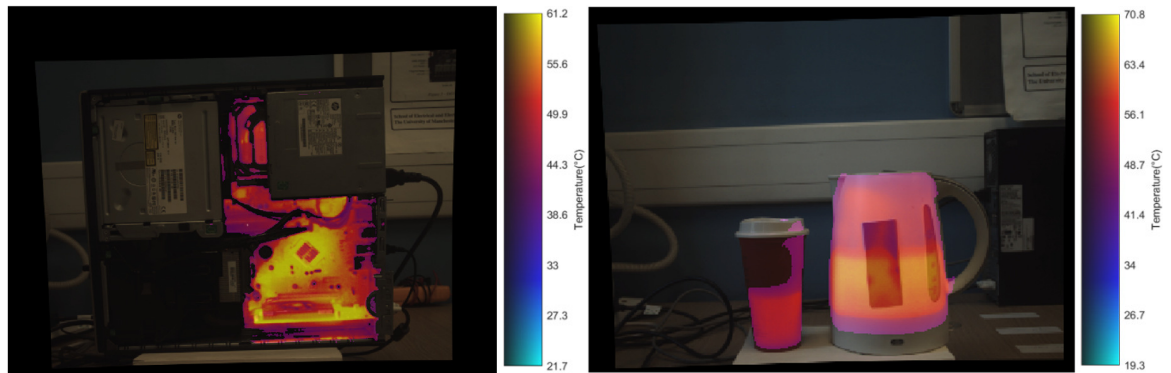


Figure 8-14 Steps of creating a combined view of thresholding overlaid image

After the thresholding intensity level is determined, the value of each pixel in the infrared image is compared with the threshold value and based on the comparison result, the pixels which have a value smaller than the threshold value are changed to 0. Figure 8-15 shows the thermal highlighted view of the combined image. The histogram equalization, threshold pixel calculation and colour mapping process are performed on the client software. The frame rate of producing a threshold view of the visible and colour mapped infrared image is around 8.23 frames/s.



a) Threshold = 43°C

b) Threshold = 50°C

Figure 8-15 Threshold view of the visible image and colour mapped infrared image

## 8.5. Summary

In this chapter, further information regarding the uncooled infrared camera module

XTM-640 is provided in section 8. This infrared camera utilized uncooled microbolometers to capture the infrared radiation from the target object. The main challenge in this experiment is that the infrared detector temperature is changing with time since it is powered on, which result in pixel grey level change for the same temperature. Therefore, in section 8.2, an adapted method is used, so the detector temperature is recorded at the same time as the infrared image is captured. Two different radiometric calibration models are generated based on several sets of infrared images taken under different stable detector temperatures. The linear polynomial temperature model is selected based on the evaluation results to represent the correct relationship between pixel value and surface temperature. In order to have a better visualized infrared image, a colour mapping scheme is used to make the discernibility between temperature levels easy, and an overlay view of the colour mapped infrared image, and the visible image is presented in section 8.4.

## Chapter 9

## Conclusion

### and future work

This thesis has proposed several novel methods for solving problems in the area of building a dual-camera system to synchronise image in the visible and non-visible spectral region. These methods have been integrated into a proposed embedded system, which has been demonstrated to be effective for generating a synchronous representation model for the purpose of creating situational awareness. This chapter looks back to the journey of design the dual-camera system and the outcomes of this research project. Conclusions of this works are summarised and discussed here, along with some possible future work and improvement suggestions to help future research in this area.

### **9.1. Conclusion**

The dual-camera system designed in this project could be a convenient and powerful portable system that overcomes some limitations of current real-time visible and infrared

image matching approaches. Apart from processing time, an example of the limitations can be sought from the attempt of the area-based image registration method mentioned in section 7.4.1b. For the feature-based image registration method, the visible image and the infrared image are two different phenomena with quite different appearance features. It could be challenging to extract highly repeatable control points in different spectrum domain images, especially for two sensors with different lens specifications. This chapter will take a review of this PhD project journey, and the importance of each chapter in the context of developing the dual-camera image synchronising system.

### 9.1.1. The journey of designing and implementing the system

The development of this dual-camera system presents new techniques for solving problems, particularly in image data streaming, sensor calibration, vision matching and surface temperature modelling. In these areas, many effort and contributions have been made and some wrong paths that have gone through.

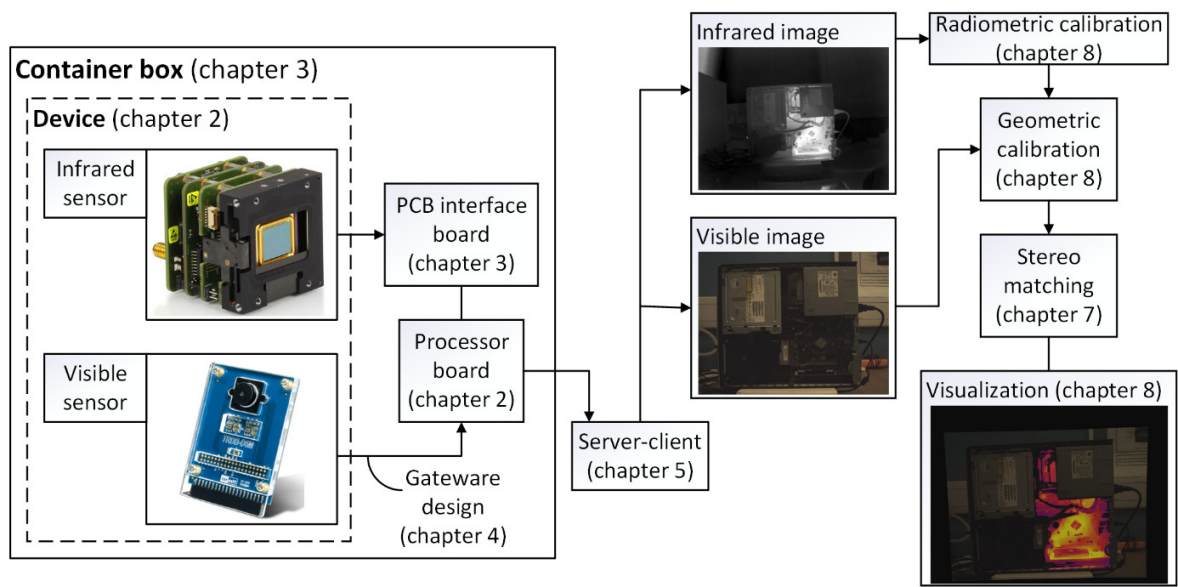


Figure 9-1 Flow chart of designing and implementing the dual-camera system

Figure 9-1 illustrate the route of developing the dual-camera system; their associate

chapters are integrated into the figure. This figure represents the order of steps designed and implemented to build a portable system for getting the matching view of the infrared and visible image. First, the interfacing problem between the infrared sensor and the processor board had to be resolved, and the relative position of the two cameras is needed to be determined. Therefore, a container should have been constructed to fix the location of the two cameras and the processor board. After the sensor's assembly is prepared, and the sensors and processor board configuration are completed, the image data is captured and processed through specific algorithms.

It can be seen from the figure that the possess of image data should be considered as the first step of the whole project, and it is the most critical task for enabling the usefulness of data to be captured by the two cameras. The geometric calibration is critical for enabling accurate texture representation in image data. The result of stereo matching then creates an aligned view of the visible and infrared image. The infrared camera's radiometric calibration can offer extra information on object surface temperature to enhance this dual-camera system's performance.

In a large project like this, it is hard not to get into any wrong path due to lacking expertise and experience. The first reckless thought happened when designing the interface PCB board. The wires were designed very thin, which caused the problem of power shortage to the infrared camera. The second half-baked idea is to use bare-metal way and NIOS II processor on the FPGA to obtain the image data, and it is a failure due to the time of acquiring one whole frame of the visible image takes over 5 minutes. Furthermore, it could be a bit of practical advice to check the embedded Linux kernel version before compiling it and put it into use, as specific FPGA drivers had been removed from newer kernels.

### **9.1.2. A conclusive view**

This thesis's conclusions are presented in the same order as the proposed research outcomes in section 1.4.

#### **Design and implementation of a dual-camera data acquisition system.**

A fully functioned dual-camera data streaming system has been developed and tested. The gateway modules are capable of storing the image data into DDR3 memory. The cameras settings are configured by changing the values in camera registers via serial communication. Control and data transfer of the two cameras is designed through the network using server-client software developed for this project. Continuous frame streaming has been achieved with the system.

#### **Implement OpenCL accelerate features on image processing.**

Board support packages are built for this dual-camera embedded system to enable the AOCL framework support on DE1-SoC. The OpenCL host and kernels are developed to implement the remap function in an accelerated approach. By testing the remapping algorithm in different platforms, the OpenCL platform facilitates a higher frame rate to produce a rectified image.

#### **Perform geometric calibration of the infrared camera and the visible camera.**

Since the infrared camera and the visible camera has lens distortion, geometric camera calibration has been performed, and the distortion coefficients are obtained. A more realistic image is achieved after the image rectification process. Some of the manufacture errors has been found using the calibrated data and some calculations.

#### **Experiment on infrared and visible image matching methods.**

The two camera's geometric relationship and the visible camera have been found out using stereo camera calibration. One of the image pair matching methods is designed base on the stereo matching algorithm. This matching application has proven to be able



to overlay the infrared image and the visible image of a target object if the distance of the object is known. Another matching approach is based on an image registration algorithm, but it has been determined that it is not reliable as the matching rate is low.

### **Perform radiometric calibration and create a temperature model for the infrared camera.**

A novel approach has been proposed for establishing a temperature model to assign temperature values to pixels. This approach is budget-saving and easy to perform as it does not require a complicated temperature-controlled environment. The established temperature model has been tested and verified. A visualisation scheme has been proposed that enable better interpretation of the infrared image data, and the combination of infrared and visible spectrum data. Visualisation using such a novel colour mapping scheme allows observers to access the data-rich result from the real-time dual-camera embedded system.

## **9.2. Future work**

Throughout this research project, many future work opportunities can be identified, of both an application-driven and theory-driven nature. This section indicates some of the future works that could directly lead to improved performance of the proposed dual-camera embedded system.

### **9.2.1. Frame rate improvement**

To improve the frame rate of this real-time image capturing system, a few options are available. The current implementation of the gateway design on FPGA is copying data from the camera output and store them in the DDR3 memory onboard. The OpenCL kernels take those data to perform the remapping algorithm. A better implementation of

this process could remove the unnecessary data copying by sending the cameras output data directly for kernel parsing. Data compression techniques can be implemented to reduce the amount of image data transfer between the server and the client software design to increase the frame rate.

### **9.2.2. Image matching improvement**

As shown in the result of chapter 8, the suitable image pair matching method is based on the parameters obtained from stereo camera calibration and the distance of the target object from the camera plane. For short distance detection, an ultrasonic distance sensor could be added as an input on FPGA so that a value can be given according to the distance of an obstacle. If the distance's value is adopted when producing image pairs, this system's usability is increased as the knowledge of distance removes constraints on how the image is captured.

In the aspect of improving the infrared and visible image pair matching technique, feature detection and description algorithms could be evaluated. Speeded-Up Robust Features (SURF) algorithms could be able to detect local features and describe each image pair independently. These matched features are then able to be matched between the two views from two cameras. If a multi-scale feature detector is used, the detected features can be different in size, which suits this project as the infrared camera and the visible camera vary in lens specifications. The feature detection algorithm can help accommodate scaling and FOV changes, as the same physical texture may be different from the camera for each image. It could be worth to investigate if there exist a more appropriate mathematical expression to modelling the lens of the XTM-640 infrared sensor, rather than the pinhole model.

In future researches, deep neural network maybe the solution to infrared and visible

image registration. The conventional feature-based registration method extracts the similar feature and then match them separately, whereas the deep learning framework needs researcher to pair patches from the target and reference frame, then learn the mapping method between these patched pairs to obtain a registered result. This method can optimise the mapping function through its information feedback gained from network training. Furthermore, self-learning can be introduced by learning the mapping algorithm using images and their registered copies.

### **9.2.3. Usability improvement**

To improve the systems' usability, instead of using a combination of server-client software to visualise the data on a PC, a smaller Liquid-Crystal Display (LCD) screen can be connected to the development board so the whole system can be handheld. Another option is to attach a tablet computer to the processor board. The tablet can take the responsibility of data streaming, and perform some simple instructions. Moreover, a tablet can communicate with a more powerful computer via a dedicated network connection.

One Light Detection and Ranging (LiDAR) device could possibly integrate with the system to act as a 3-D laser scanner to generate a 3-D point cloud mesh. Then the visible and infrared data can be register to the model to provide a more intuitive view of the scene. To achieve high level of accuracy and automation to map the pixel data to the 3-D plane can be another research topic.

## Appendix A TRDB-D5M camera binning mode

By default, the resolution of the visible output image is the full width and height of  $2592 \times 1944$ . To reduce the output resolution for camera geometric calibration and stereo camera calibration, the binning method is used. The binning method is an in-built technique to reduce the output image resolution. This mode can be enabled by changing the corresponding register value via I2C controls designed on the server program.

The binning algorithm reduces the visible image resolution by combining adjacent same-colour imager pixels to produce one output pixel. All the pixel outputs in the field of view contribute to the output image in the binning mode, which results in a more pleasing output visible image with reduced subsampling artefacts, and have an improvement in low-light performance. For image columns, the combination step can be either the operation of averaging or summing, which can be chosen manually depends on lighting conditions. The summing algorithm produces a gain roughly equivalent to the column bin factor in low-light conditions. The binning mode is illustrated in Figure A - 1

**Error! Reference source not found..**

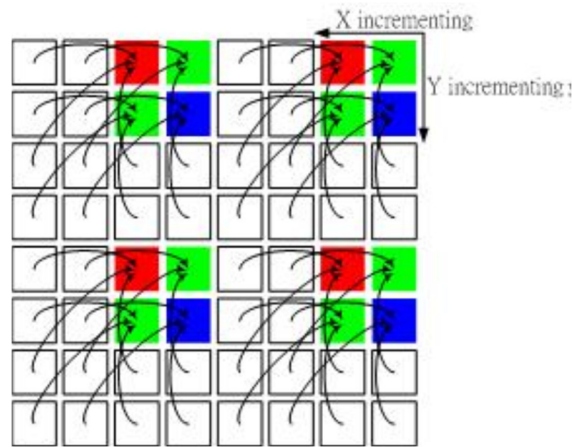


Figure A - 1 TRDB-D5M visible camera binning mode [37]

## Appendix B TRDB-D5M visible camera specifications

Table B - 1 TRDB-D5M module specification

Parameter		Specification
Active pixels		2592(width) $\times$ 1944(height)
Pixel size		2.2 $\mu$ m $\times$ 2.2 $\mu$ m
Colour filter array		RGB Bayer pattern
Shutter type		Global reset release
Maximum data rate/master		96 Mp/s at 96 MHz
Frame rate	Full resolution	Programmable up to 15 fps
	VGA (640 $\times$ 480)	Programmable up to 70 fps
ADC resolution		12 - bit
Responsivity		1.4 V/lux-sec (550nm)
Pixel dynamic range		70.1dB
SNRMAX		38.1dB
Supply voltage	Power	3.3V
	I/O	1.7V $\sim$ 3.1V

## Appendix C XTM-640 thermal camera specification

Table C - 1 XTM-640 module specifications

Parameter	Specification
Array Type	a-Si micro-bolometer
Active pixels	2592(width) $\times$ 1944(height)
Pixel pitch	17 $\mu\text{m}$
Spectral Band	8 $\sim$ 14 $\mu\text{m}$
Noise Equivalent Temperature Difference (NETD)	(1) =50 mK @ 30°C with F/1 lens (typical value) (2) <75 mK @ 30°C with F/1 lens
Detector Operating Mode	Rolling Shutter Mode
Window of interest	Yes (not accessible for Shutterless cameras) – Higher frame rates possible for smaller windows
Exposure time range	1 – 80 $\mu\text{s}$
Pixel Operability	>99%
Frame rate (full-frame)	50Hz
Cooling	No
A/D conversion resolution	16 - bit
Input Voltage	3.3V $\pm$ 10%
Power consumption	1.65W
Ambient Operating Temperature	-40°C to 60°C
Dimensions	45 $\times$ 45 $\times$ 35.8 mm <sup>3</sup>

# References

- [1] A. C. Caputo, “3 - Digital Video Hardware,” in *Digital Video Surveillance and Security (Second Edition)*, Elsevier Inc, 2014, pp. 49-105.
- [2] F. Omri, S. Foufou and M. Abidi, “NIR and Visible Image Fusion for Improving Face Recognition at Long Distance,” in *International Conference on Image and Signal Processing*, 2014.
- [3] H. Hariharan, A. Koschan, B. Abidi, A. Gribok and M. Abidi, “Fusion of Visible and Infrared Images using Empirical Mode Decomposition to Improve Face Recognition,” in *2006 IEEE International Conference on Image Processing*, 2006.
- [4] S. G. Kong, J. Heo, B. R. Abidi, J. Paik and M. A. Abidi, “Recent advances in visual and infrared face recognition—a review,” *Computer Vision and Image Understanding*, vol. 97, no. 1, pp. 103-135, 2005.
- [5] C. O. Conaire, N. E. O'Connor, E. Cooke and A. F. Smeaton, “Comparison of Fusion Methods for Thermo-Visual Surveillance Tracking,” in *2006 9th International Conference on Information Fusion*, Florence, Italy, 2006.
- [6] P. Kumar, A. Mittal and P. Kumar, “Fusion of Thermal Infrared and Visible Spectrum Video for Robust Surveillance,” in *Computer Vision, Graphics and Image Processing. Lecture Notes in Computer Science, vol 4338*, Berlin, Heidelberg, 2006.
- [7] J. Han and B. Bhanu, “Fusion of color and infrared video for moving human detection,” *Pattern Recognition*, vol. 40, no. 6, pp. 1771-1784, 2007.
- [8] I. Ulusoy and H. Yuruk, “New method for the fusion of complementary information from

- infrared and visual images for object detection,” *IET Image Processing*, vol. 5, no. 1, pp. 36-48, 2011.
- [9] S. Gao, Y. Cheng and Y. Zhao, “Method of visual and infrared fusion for moving object detection,” *Optics Letters*, vol. 38, no. 11, pp. 1981-1983, 2013.
- [10] S. Singh, A. Gyaourova, G. Bebis and I. Pavlidis, “Infrared and visible image fusion for face recognition,” in *Proceedings Volume 5404, Biometric Technology for Human Identification*, Orlando, Florida, United States, 2004.
- [11] D. M. Bulanon, T. F. Burks and V. Alchanatis, “Image fusion of visible and thermal images for fruit detection,” *Biosystems Engineering*, vol. 103, no. 1, pp. 12-22, 2009.
- [12] A. González, Z. Fang, Y. Socarras, J. Serrat, D. Vázquez, J. Xu and A. M. López, “Pedestrian Detection at Day/Night Time with Visible and FIR Cameras: A Comparison,” *Sensors*, vol. 16, no. 6, p. 820, 2016.
- [13] W. He, W. Feng, Y. Peng, Q. Chen, G. Gu and Z. Miao, “Multi-level image fusion and enhancement for target detection,” *Optik*, vol. 126, no. 11-12, pp. 1203-1208, 2015.
- [14] S. R. Schnelle and A. L. Chan, “Enhanced target tracking through infrared-visible image fusion,” in *14th International Conference on Information Fusion*, Chicago, 2011.
- [15] K. S. Kumar, G. Kavitha, S. Ramasamy and G. Ramesh, “Visual and Thermal Image Fusion for UAV Based Target Tracking,” in *MATLAB - A Ubiquitous Tool for the Practical Engineer*, 2011.
- [16] J. Ma, J. Jiang, C. Liu and Y. Li, “Feature guided Gaussian mixture model with semi-supervised EM and local geometric constraint for retinal image registration,” *Information Sciences*, vol. 417, p. 128–142, 2017.
- [17] Q. Li, G. Han, P. Liu, H. Yang, H. Luo and J. Wu, “An Infrared-Visible Image Registration Method Based on the Constrained Point Feature,” *Sensors*, vol. 21, no. 1188, 2021.
- [18] M. Kerkech, A. Hafiane and R. Canals, “Vine disease detection in UAV multispectral



- images using optimized image registration and deep learning segmentation approach,” *Computers and Electronics Agriculture*, vol. 174, no. 105446, 2020.
- [19] FLIR, “ThermoVision Dual Sensor (DS),” 2020 © FLIR® Systems, Inc, 2020. [Online]. Available: <https://www.flir.co.uk/products/thermovision-dual-sensor-ds/>. [Accessed 12 12 2020].
- [20] Xenics Infrared Solutions, “XTM 640 Series,” Xenics, [Online]. Available: <https://www.xenics.com/long-wave-infrared-imagers/xtm-640-series/>. [Accessed 5 May 2020].
- [21] T. Bourlai, N. Kalka, A. Ross, B. Cukic and L. Hornak, “Cross-Spectral Face Verification in the Short Wave Infrared (SWIR) Band,” in *2010 International Conference on Pattern Recognition*, Istanbul, Turkey, 2010.
- [22] A. El Maadi and X. Maldague, “Outdoor infrared video surveillance: A novel dynamic technique for the subtraction of a changing background of IR images,” *Infrared Physics & Technology*, no. 49, pp. 261-265, 2007.
- [23] M. Ulrich, T. Hess, . S. Abdulatif and B. Yang, “Person Recognition based on Micro-Doppler and Thermal Infrared Camera Fusion for Firefighting,” in *21st International Conference on Information Fusion (FUSION)*, Stuttgart, Germany, 2018.
- [24] E. Y. Zhang, “Infrared imaging system for advanced rescue vision system”. United States of America Patent US6476391B1, 22 May 1999.
- [25] B. B. Lahiri, S. Bagavathiappan, T. Jayakumar and J. Philip, “Medical applications of infrared thermography: A review,” *Infrared Physics & Technology*, no. 55, pp. 221-235, 2012.
- [26] FLIR Commercial Vision Systems B.V., “Uncooled detectors for thermal imaging cameras,” [Online]. Available: [http://www.flirmedia.com/MMC/CVS/Apl\\_Stories/AS\\_0015\\_EN.pdf](http://www.flirmedia.com/MMC/CVS/Apl_Stories/AS_0015_EN.pdf). [Accessed 4 May 2020].
- [27] Xenics nv, “Interface Control Document: Control and Operation of Bolometer 640-17um

& 384-25um Cameras and Cores,” Leuven, Belgium, 2015.

- [28] Basler Vision Technologies, Camera Link Technology Brief, 2001.
- [29] T. Kim and J. Paik, “Adaptive contrast enhancement using gain-controllable clipped histogram equalization,” *IEEE Transactions on Consumer Electronics*, vol. 54, no. 4, pp. 1803-1810, 2008.
- [30] P. Barry and P. Crowley, Modern Embedded Computing, Morgan Kaufmann, 2012.
- [31] D. Bhadra, V. S. Vij and K. S. Stevens, “A Low Power UART Design Based on Asynchronous Techniques,” in *2013 IEEE 56th International Midwest Symposium on Circuits and Systems (MWSCAS)*, Columbus, 2013.
- [32] Terasic Inc, “5 Mega Pixel Digital Camera Package,” Terasic Inc, 2013. [Online]. Available: <https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=68&No=281>. [Accessed 3 June 2020].
- [33] E. R. Fossum, “CMOS Image Sensors: Electronic Camera-On-A-Chip,” *IEEE Transactions on Electron Devices*, vol. 44, no. 10, pp. 1689 - 1698, 1997.
- [34] A. Holms and A. Quach, “Complementary Metal-Oxide Semiconductor Sensors,” University of California Santa Barbara, California.
- [35] J. Janesick, J. T. Andrews and T. Elliott, “Fundamental performance differences between CMOS and CCD imagers: Part 1,” in *SPIE Astronomical Telescopes + Instrumentation*, Orlando, Florida , United States, 2006.
- [36] C. Danakis, M. Afgani, G. Povey, I. Underwood and H. Haas, “Using a CMOS camera sensor for visible light communication,” in *2012 IEEE Globecom Workshops*, Anaheim, CA, USA, 2012.
- [37] Terasic, “Terasic TRDB-D5M Hardware specification”.
- [38] S. Tasch, C. Brandstatter, F. Meghdadi, G. Leising, G. Froyer and L. Athouel, “Red–

- green–blue light emission from a thin film electroluminescence device based on parahexaphenyl,” *Advanced Material*, vol. 9, no. 1, pp. 33-36, 1996.
- [39] i2c.info, “2020 I2C Info – I2C Bus, Interface and Protocol,” Ezoic Limited, [Online]. Available: <https://i2c.info/>. [Accessed 8 May 2020].
  - [40] Terasic Technologies, “DE1-SoC User Manual,” Terasic, 2015.
  - [41] Intel Corporation - FPGA University Program, “DE1-SOC COMPUTER SYSTEM WITH ARM CORTEX A9,” Intel FPGA, 2017.
  - [42] Altera Corporation, “Cyclone V Hard Processor System Technical Reference Manual,” 2018.
  - [43] Intel Corporation, “Intel® FPGA SDK for OpenCL™ Standard Edition Getting Started Guide,” 24 9 2018. [Online]. Available: [https://www.intel.com/content/dam/www/progr\\_ammable/us/en/pdfs/literature/hb/opencl-sdk/ug-aoclstd-getting-started.pdf](https://www.intel.com/content/dam/www/progr_ammable/us/en/pdfs/literature/hb/opencl-sdk/ug-aoclstd-getting-started.pdf). [Accessed 10 5 2020].
  - [44] Intel Corporation, “Intel® Quartus® Prime Software Suite,” Intel, 2020. [Online]. Available: <https://www.intel.co.uk/content/www/uk/en/software/programmable/quartus-prime/overview.html>. [Accessed 4 June 2020].
  - [45] Intel Inc, “Intel Quartus Prime Standard Edition User Guide: Getting Started,” Intel, 2020.
  - [46] Intel Corporation, “Platform Designer (formerly Qsys),” Intel Corporation, 2020. [Online]. Available: <https://www.intel.com/content/www/us/en/programmable/products/design-software/fpga-design/quartus-prime/features/qts-platform-designer.html>. [Accessed 4 June 2020].
  - [47] The Khronos® Group Inc, “Open standard for parallel programming of heterogeneous systems,” The Khronos® Group Inc, 2020. [Online]. [Accessed 19 May 2020].
  - [48] B. d. Silva, A. Braeken, E. H. D'Hollander and A. Touhafi, “Comparing and combining GPU and FPGA accelerators in an image processing context,” in *2013 23rd International*

*Conference on Field Programmable Logic and Applications (FPL)*, 2013.

- [49] H. M. Waidyasooriya, Y. Takei, S. Tatsumi and M. Hariyama, “OpenCL-Based FPGA-Platform for Stencil Computation and Its Optimization Methodology,” *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS*, vol. 28, no. 5, pp. 1390-1402, 2017.
- [50] K. Ostaszewski, P. Heinisch and H. Ranocha, “Advantages and pitfalls of OpenCL in computational physics,” in *IWOCL '18: Proceedings of the International Workshop on OpenCL*, New York, 2018.
- [51] Intel Cooperation, “Intel FPGA SDK for OpenCL Standard Edition: Programming Guide,” 2020.
- [52] Altera Corporation, “Implementing FPGA Design with the OpenCL Standard,” 11 2013. [Online]. Available: <https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/wp/wp-01173-opencl.pdf>. [Accessed 19 5 2020].
- [53] M. H. K. U. Hasitha Muthumala Waidyasooriya, Design of FPGA-Based Computing Systems with OpenCL, Sendai, Japan: Springer International Publishing, 2018.
- [54] Intel Corporation, “Intel® Edison Board Support Package,” 2015.
- [55] Altera, “Best Practices for Incremental Compilation Partitions and Floorplan Assignments,” in *Quartus II Handbook Version 13.1*, 2013.
- [56] E. d. I. T. A. M. D. V. Juan Jose Rodriguez Andina, FPGAs: Fundamentals, Advanced Features, and Applications in Industrial Electronics, CRC Press, 2017.
- [57] Intel Corporation, “Avalon® Interface Specifications,” 2020.
- [58] “TCP-like congestion control for layered multicast data transfer,” in *Proceedings. IEEE INFOCOM '98, the Conference on Computer Communications. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Gateway to the 21st Century (Cat. No.98)*, San Francisco, CA, USA, 1998.

- [59] Python Software Foundation, “Python,” [Online]. Available: <https://www.python.org/>.
- [60] J. Kannala, J. Heikkilä and S. S. Brandt, “Geometric Camera Calibration,” in *Wiley Encyclopedia of Computer Science and Engineering*, Finland, 2008.
- [61] Z. Zhang, “A flexible new technique for camera calibration,” in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.
- [62] M. Vo, Z. Wang, L. Luu and J. Ma, “Advanced geometric camera calibration for machine vision,” *Optical Engineering*, vol. 50, no. 11, pp. 110503-1-110503-3, 2011.
- [63] H. Levy, “Determining Local Depth From Structured Light Using A Regular Dot Grid,” in *Technical Disclosure Commons*, 2019.
- [64] W. Birkfellner, *Applied Medical Image Processing: A Basic Course*, Austria: CRC Press, 2016.
- [65] The MathWorks, Inc., *Computer Vision System Toolbox™ User's Guide*, Natick, 2018.
- [66] J. P. David A. Forsyth, *Computer Vision A Modern Approach*, New Jersey: Pearson Education, Inc., 2003.
- [67] B. Hönlinger and H. H. Nasse, “Distortion,” Carl Zeiss , 2009.
- [68] C. G. Leonardo Romero, “Correcting Radial Distortion of Cameras with Wide Angle Lens Using Point Correspondences,” in *Scene Reconstruction Pose Estimation and Tracking*, Morelia, Michoacán, México, InTech, 2001, pp. 51-64.
- [69] Xenics nv, “Technical drawing of LWIR XTM-640 lens,” Leuven,Belgium.
- [70] H. Wolfgang, “Correcting Lens Distortions in Digital Photographs,” Chicago, 2010.
- [71] A. M. R. S. Davide Scaramuzza, “A Toolbox for Easily Calibrating Omnidirectional Cameras,” in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, 2006.
- [72] M.-H. Chiang, H.-T. Lin and C.-L. Hou, “Development of a Stereo Vision Measurement System for a 3D Three-Axial Pneumatic Parallel Mechanism Robot Arm,” *Sensors*, vol.

11, no. 2, pp. 2257-2281, 2011.

- [73] A. K. Gary Bradski, *Learning OpenCV*, Sebastopol: O'Reilly Media, Inc, 2008.
- [74] T. Hanning, *High Precision Camera Calibration*, Vieweg+Teubner, 2011.
- [75] O. S. Janne Heikkila, "A four-step camera calibration procedure with implicit image correction," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, USA, 1997.
- [76] A. D. I. Escalera and J. M. Armingol, "Automatic Chessboard Detection for Intrinsic and Extrinsic Camera Parameter Calibration," *Sensors*, vol. 3, no. 10, pp. 2027-2044, 2010.
- [77] V. Douskos, I. Kalisperakis and G. Karras, "Automatic calibration of digital cameras using planar chess-board patterns," in *Proceedings of the 8th Conference on Optical*, 2007.
- [78] M. Yamaguchi, P. Truong, S. Mori and V. Nozick, "Superimposing Thermal-Infrared Data on 3D Structure Reconstructed by RGB Visual Odometry," *IEICE Transactions on Information and Systems*, vol. 101, no. 5, pp. 1296-1307, 2018.
- [79] S. M. Peter Sturm, "On Plane-Based Camera Calibration: A General Algorithm, Singularities, Applications," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR '99)*, Fort Collins, United States, 1999.
- [80] H. H. Sangyoon Lee, "A Robust Camera-Based Method for Optical Distortion Calibration of Head-Mounted Displays," *DISPLAY TECHNOLOGY*, vol. 11, no. 10, pp. 845-853, 2015.
- [81] D. C. Brown, "Close-range camera calibration," *Photogram Engineering*, vol. 37, pp. 855-866, 1971.
- [82] H. M. Nguyen, B. Wünsche, P. J. Delmas and C. Lutteroth, "3D Models from the Black Box: Investigating the Current State of Image-Based Modeling," in *Proceedings of the 20th International Conference on Computer Graphics, Visualisation and Computer Vision (WSCG), 2012*, 2012.
- [83] L. Ma, Y. Chen and K. L. Moore, "Rational Radial Distortion Models of Camera Lenses

- with Analytical Solution for Distortion Correction,” *International Journal of Information Acquisition*, vol. 1, no. 02, pp. 135-147, 2004.
- [84] H. Blasinski, W. Hai and F. Lohier, “FPGA architecture for real-time barrel distortion correction of colour images,” in *IEEE International Conference on Multimedia and Expo (ICME)*, Barcelona, Spain, 2011.
  - [85] G. P. Stein, “Lens distortion calibration using point correspondences,” in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, USA, 1997.
  - [86] C. Ricolfe-Viala and A.-J. Sanchez-Salmeron, “Lens distortion models evaluation,” *Applied Optics*, vol. 49, no. 30, pp. 5914-5928, 2010.
  - [87] P. Sturm, Focal Length, Boston: Springer, 2014.
  - [88] J. N. Aarsvold and M. N. Wernick, Emission Tomography, The Fundamentals of PET and SPECT, Academic Press, 2004.
  - [89] J. Ma, Y. Ma and C. Li, “Infrared and visible image fusion methods and applications: A survey,” *Information Fusion*, vol. 45, pp. 153-178, 2019.
  - [90] J. Zhao, Q. Zhou, Y. Chen, H. Feng, Z. Xu and Q. Li, “Fusion of visible and infrared images using saliency analysis and detailpreserving based image decomposition,” *Infrared Physics & Technology*, vol. 56, pp. 93-99, 2013.
  - [91] A. Toet, J. K. IJspeert, A. M. Waxman and M. Aguilar, “Fusion of visible and thermal imagery improves situational awareness,” *Display*, vol. 18, no. 2, pp. 85-95, 1997.
  - [92] T. Hanning, High Precision Camera Calibration, Vieweg+Teubner, 2011.
  - [93] A. O’Riordan, T. Newe, D. J. F. Toal and G. Dooly, “Stereo Vision Sensing: Review of existing systems,” in *2018 Twelfth International Conference on Sensing Technology (ICST)*, 2018.
  - [94] S. Beriault, “Multi-camera system design, calibration and three-dimensional reconstruction

for markerless motion capture,” PhD Thesis, 2008.

- [95] P. Rathnayaka, S.-H. Baek and S.-Y. Park, “An Efficient Calibration Method for a Stereo Camera System with Heterogeneous Lenses Using an Embedded Checkerboard Pattern,” *Journal of Sensors*, vol. 2017, pp. 1-12, 2017.
- [96] O. Stankiewicz, G. Lafruit and M. Domański, “Chapter 1 - Multiview video: Acquisition, processing, compression, and virtual view rendering,” in *Academic Press Library in Signal Processing, Volume 6*, Academic Press, 2018, pp. 3-74.
- [97] E. R. Davies, *Computer Vision (Fifth Edition)-Principles, Algorithms, Applications, Learning*, Academic Press, 2018.
- [98] Y. Liu and J. K. Aggarwal, *Handbook of Image and Video Processing (Second Edition)*, Austin: ACADEMIC PRESS, 2005.
- [99] G. B. Arfken, D. F. Griffing , D. C. Kelly and J. Priest, *International Edition University Physics*, Oxford, Ohio: Academic Press, 1984.
- [100] S. Agarwal, H.-L. Lee, B. Sturmfels and R. R. Thomas , “On the Existence of Epipolar Matrices,” *International Journal of Computer Vision*, vol. 121, no. 2017, pp. 403-514, 2016.
- [101] R. Szeliski, “Stereo correspondence,” in *Computer Vision*, London, Springer, 2011, pp. 467-503.
- [102] L. B. Stelmach and W. J. Tam, “Stereoscopic image coding: Effect of disparate image-quality in left- and right-eye views,” *Signal Processing: Image Communication*, vol. 14, no. 1-2, pp. 111-117, 1998.
- [103] B. Zitová and J. Flusser, “Image registration methods: a survey,” *Image and Vision Computing*, vol. 21, no. 11, pp. 977-1000, 2003.
- [104] K. Yu, J. Ma, F. Hu, T. Ma, S. Quan and B. Fang, “A grayscale weight with window algorithm for infrared and visible image registration,” *Infrared Physics & Technology*, vol. 99, pp. 178-186, 2019.



- [105] Y. Kim, J. Lee and J. Ra, "Multi-sensor image registration based on intensity and edge orientation information," *Pattern Recognition*, vol. 41, no. 11, pp. 3356-3365, 2008.
- [106] S. Paul and U. C. Pati, "Remote Sensing Optical Image Registration Using Modified Uniform Robust SIFT," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 9, pp. 1300 - 1304, 2016.
- [107] F. Zhao, Q. Huang and W. Gao, "Image Matching by Normalized Cross-Correlation," in *2006 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2006.
- [108] D. Foley, D. J. O'Brien, L. León-Vintró and B. McClean, "Phase correlation applied to the 3D registration of CT and CBCT image volumes," *Physica Medica*, vol. 32, no. 4, pp. 618-624, 2016.
- [109] L. G. Brown, "A survey of image registration techniques," *ACM Computing Surveys*, vol. 24, no. 4, pp. 325-376, 1992.
- [110] H. Buijs, A. Pomerleau, M. Fournier and W. Tam, "Implementation of a fast Fourier transform (FFT) for image processing applications," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 22, no. 6, pp. 420-424, 1974.
- [111] W. Menke and J. Menke, *Environmental Data Analysis with MatLab*, Elsevier, 2012.
- [112] C. Kuglin and D. A. HINES, "The phase correlation image alignment method," in *Proceedings of the International Conference on Cybernetics and Society*, 1975.
- [113] D. Cheng and K. I. Kou, "FFT multichannel interpolation and application to image super-resolution," *Signal Processing*, vol. 162, pp. 21-34, 2019.
- [114] M. I. Rajab, T. A. El-Benawy and M. W. Al-Hazmi, "Application of frequency domain processing to X-ray radiographic images of welding defects," *X-Ray Science and Technology*, vol. 15, no. 3, pp. 147-156, 2007.
- [115] E. R. Davies, *Machine Vision: Theory, Algorithms, Practicalities*, A volume in Signal Processing and its Applications, London: Morgan Kaufmann, 2005.

- [116] J. Semmlow, "Chapter 2 - Basic Concepts in Signal Processing," in *Signals and Systems for Bioengineers (Second Edition)*, Academic Press, 2012, pp. 35-80.
- [117] H. Zheng and J. Li, "Real-time correction of distortion image based on FPGA," in *2010 International Conference on Intelligent Computing and Integrated System*, 2010.
- [118] N. Bellas, S. M. Chai, M. Dwyer and D. Linzmeier, "Real-time Fisheye Lens Distortion Correction Using Automatically Generated Streaming Accelerators," in *IEEE Symposium on Field Programmable Custom Computing Machines*, 2017.
- [119] D. Honegger, H. Oleynikova and M. Pollefey, "Real-time and Low Latency Embedded Computer Vision Hardware Based on Combination of FPGA and Mobile CPU," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago, 2014.
- [120] D. Castells-Rufas and J. Carrabina, OpenCL-based FPGA accelerator for disparity map generation with stereoscopic event cameras, Barcelona, 2019.
- [121] W. Wang, J. Yan, N. Xu, Y. Wang and F. H. Hsu, "Real-time high-quality stereo vision system in FPGA," *IEEE Trans. Circuits Syst. Video Technol*, 2015.
- [122] M. Domański, J. Konieczny, M. Kurc, A. Łuczak, J. Siast, O. Stankiewicz and K. Wegner, "Fast depth estimation on mobile platforms and FPGA devices," in *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video*, 2015.
- [123] C. L. Wyatt, Radiometric Calibration: Theory and Methods, Logan, Utah: Academic Press, 1978.
- [124] Z. Liu, E. P. Blasch, Z. Xue, J. Zhao, R. Laganier and W. Wu, "Objective Assessment of Multiresolution Image Fusion Algorithms for Context Enhancement in Night Vision: A Comparative Study," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 1, pp. 94-109, 2012.
- [125] L. Hoegner, S. Tuttas and U. Stilla, "3D building reconstruction and construction site monitoring from RGB and TIR image sets," in *2016 12th IEEE International Symposium on Electronics and Telecommunications (ISETC)*, Timisoara, 2016.

- [126] V. Ambrosia, S. S. Wegener, D. V. Sullivan and S. W. Buechel, "Demonstrating UAV-Acquired Real-Time Thermal Data over Fires," *Photogrammetric Engineering and Remote Sensing*, vol. 69, no. 4, pp. 391-402, 2003.
- [127] J. E. Johnson, J. A. Shaw, R. L. Lawrence and P. Nugent, "Long-wave infrared imaging of vegetation for detecting leaking CO<sub>2</sub> gas," *Journal of Applied Remote Sensing*, vol. 6, no. 1, pp. 063612-1–063612-9, 2012.
- [128] FLIR, "Cooled or Uncooled?," 2020 © FLIR® Systems, Inc., 2020. [Online]. Available: <https://www.flir.co.uk/discover/rd-science/cooled-or-uncooled/>.
- [129] T. Luhmann, J. Piechel and T. Roelfs, "Thermal Infrared Remote Sensing, Geometric Calibration of Thermographic Cameras," in *Remote Sensing and Digital Image Processing*, Newcastle upon Tyne, UK, Springer, 2013, pp. 27-42.
- [130] A. Schaufelbühl, N. Schneeberger, U. Münch, M. Waelti, O. Paul, O. Brand, H. Baltes, C. Menolf, Q. Huang, E. Doering and M. Loepfe, "Uncooled Low-Cost Thermal Imager Based on Micromachined CMOS Integrated Sensor Array," *Journal of Microelectromechanical Systems*, vol. 10, no. 4, pp. 503-510, 2001.
- [131] R. K. Bhan and V. Dhar, "Recent infrared detector technologies, applications, trends and development of HgCdTe based cooled infrared focal plane arrays and their characterization," *Opto-Electronics Review*, vol. 27, no. 2, pp. 174-193, 2019.
- [132] J. A. Berni, P. J. Zarco-Tejada, L. Suarez and E. Fereres, "Thermal and Narrowband Multispectral Remote Sensing for Vegetation Monitoring From an Unmanned Aerial Vehicle," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 3, pp. 722-738, 2009.
- [133] T. W. Parker, C. A. Marshall, M. Kohin and R. Murphy, "Uncooled infrared sensors for surveillance and law enforcement applications," in *Proceedings Volume 2935, Surveillance and Assessment Technologies for Law Enforcement*, Boston, MA, United States, 1997.
- [134] R. S. Balcerak, "Uncooled IR imaging: technology for the next generation," in *Proceedings*

Volume 3698, *Infrared Technology and Applications XXV*, Orlando, FL, United States, 1999.

- [135] N. Teranishi, “Chapter 6 Thermoelectric Uncooled Infrared Focal Plane Arrays,” in *Semiconductors and Semimetals*, Kanagawa, Japan, Academic Press, 1997, pp. 203-218.
- [136] S. Vidas, R. Lakemond, S. Denman, C. Fookes, S. Sridharan and T. Wark, “A Mask-Based Approach for the Geometric Calibration of Thermal-Infrared Cameras,” *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 6, pp. 1625-1635, 2012.
- [137] Y. Deng, S. Tang, H. Zeng, Z. Wu and D. Tung, “Experiments on Temperature Changes of Microbolometer under Blackbody Radiation and Predictions Using Thermal Modeling by COMSOL Multiphysics Simulator,” *Sensors (Basel)*, vol. 18, no. 8, 2018.
- [138] D. Lin, H. Maas, P. Westfeld, H. Budzier and G. Gerlach, “An advanced radiometric calibration approach for uncooled thermal cameras,” *The Photogrammetric Record*, vol. 33, no. 161, p. 30–48, 2018.
- [139] P. W. Nugent, J. A. Shaw and N. J. Pust, “Correcting for focal-plane-array temperature dependence in microbolometer infrared cameras lacking thermal stabilization,” *Optical Engineering*, vol. 52, no. 6, 2013.
- [140] S. Vidas and P. Moghadam, “Ad Hoc Radiometric Calibration of a Thermal-Infrared Camera,” in *2013 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, Hobart, TAS, 2013.
- [141] H. Budzier and G. Gerlach, “Calibration of uncooled thermal infrared cameras,” *Journal of Sensors and Sensor Systems*, vol. 4, p. 187–197, 2015.
- [142] C. L. Cheng, Shalabh and G. Garg, “Coefficient of determination for multiple measurement error models,” *Journal of Multivariate Analysis*, vol. 126, pp. 137-152, 2014.
- [143] R. J. Hyndman and A. B. Koehler, “Another look at measures of forecast accuracy,” *International Journal of Forecasting*, vol. 22, pp. 679-688, 2006.

- [144] Y. Zhang, Q. Li and H. Zhou, “Chapter 1 - Theoretical Foundation and Basic Properties of Thermal Radiation,” in *Theory and Calculation of Heat Transfer in Furnaces*, Academic Press, 2016, pp. 1-43.
- [145] B. Hoffschmidt, S. Alexopoulos, J. Götsche, M. Sauerborn and O. Kaufhold, “High Concentration Solar Collectors,” in *Reference Module in Earth Systems and Environmental Sciences: Comprehensive Renewable Energy*, 2012.
- [146] M. Konda, N. Imasato, K. Nishi and T. Toda , “Measurement of the sea surface emissivity,” *Journal of Oceanography*, vol. 50, p. 17–30, 1994.
- [147] I. J. Spiro, R. C. Jones and D. Q. Wark, “Atmospheric transmission: Concepts, symbols, units and nomenclature,” *Infrared Physics*, vol. 5, no. 1, pp. 11-36, 1965.
- [148] A. Toet, “Natural colour mapping for multiband nightvision imagery,” *Information Fusion*, vol. 4, no. 3, pp. 155-166, 2003.
- [149] K. Moreland, “Why We Use Bad Color Maps and What You Can Do About It,” *Electronic Imaging*, vol. 16, pp. 1-6, 2016.
- [150] S. Eddins, “A New Colormap for MATLAB – Part 2 – Troubles with Rainbows,” MATLAB® R2014b, 20 10 2014. [Online]. Available: <https://blogs.mathworks.com/steve/2014/10/20/a-new-colormap-for-matlab-part-2-troubles-with-rainbows/>. [Accessed 15 11 2020].
- [151] R. T. Marcus, “chapter 2 - The Measurement of Color,” in *Color for Science, Art and Technology*, North-Holland, Elsevier B.V. , 1998, pp. 31-96.
- [152] D.-S. Parkyz, J.-S. Parky and J. Hanny, “Image Indexing Using Color Histogram in the Cieluv Color Space,” 2007.
- [153] A. R. Smith, “Painting tutorial notes,” *SIGGRAPH*, vol. 79, 1979.
- [154] T. Porter and T. Duff, “Compositing Digital Images,” *Computer Graphics*, vol. 18, no. 3, pp. 253-259, 1984.

- [155] S. Maji and A. Nath, "Scope and Issues in Alpha Compositing Technology," *International Journal of Innovative Research in Advanced Engineering (IJIRAE)*, vol. 2, no. 12, pp. 38-43, 2015.
- [156] S. Shaikh, G. Hanumant, R. Manza and K. Kale, "Segmentation of Thermal Images Using Thresholding-Based Methods for Detection of Malignant Tumours," in *The International Symposium on Intelligent Systems Technologies and Applications*, 2016.
- [157] A. W. F. David Claus, "A Rational Function Lens Distortion Model for General Cameras," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, USA, 2005.
- [158] W. J. Carper, T. M. Lillesand and R. W. Kiefer, "The use of Intensity-Hue-Saturation Transformations for Merging SPOT Panchromatic and Multispectral Image Data," in *Photogrammetric Engineering and Remote Sensing*, 1990.