

A Miniaturized Insect Eye Inspired Multi-camera Real-time Panoramic Imaging System

THÈSE N° 6844 (2015)

PRÉSENTÉE LE 20 NOVEMBRE 2015

À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR
LABORATOIRE DE SYSTÈMES MICROÉLECTRONIQUES
PROGRAMME DOCTORAL EN MICROSYSTÈMES ET MICROÉLECTRONIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Ömer ÇOĞAL

acceptée sur proposition du jury:

Prof. C. Dehollain, présidente du jury
Prof. Y. Leblebici, directeur de thèse
Prof. J.-I. Guo, rapporteur
Prof. R. Dekker, rapporteur
Prof. J.-Ph. Thiran, rapporteur



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2015

"..don't forget that on top of everything, you are engineers and electronic engineers, think about the chip design, analog or digital design just as your different hobbies.."

Prof. Duran Leblebici
(from a talk during the technical trip to YITAL, 2005)

best things
come in
small packages

To my parents, sister and wife ...
Anneme babama, ablama ve eşime ...

Acknowledgements

I would never have been able to finish my dissertation without the guidance of my advisor, help from friends, and support from my family and wife.

First of all, I would like to thank my thesis director Prof. Yusuf Leblebici not only for the opportunity he gave to start this PhD work but also being supportive and giving advises not only by words but also by life examples. Simply, a great leader.

I would like to thank my thesis committee members Prof. Catherine Dehollain, Prof. Ronald Dekker, Prof. Jiun In Guo, Prof. Jean-Philippe Thiran for evaluating my work and their constructive comments and discussions.

I would like to thank to the Master and Bachelor degree students to whom I have collaborated in different parts of my PhD work; Mattia Cacciotti for his collaboration on ASIC design of the miniature panoramic imaging system and Jonathan Narinx for his help during the ASIC design, Nicola Gerber for his contributions on the research for analysis of different Bayer demosaicing and white balancing algorithms. I would like to thank to Abdulkadir Uzun, for his work on the virtual stereo construction study. I would like to thank to our Professors at LSM, Alain Vachoux and Alexandre Schmid. Being their lab assistant in various classes was a great experience and pleasure for me. I also would like to thank to Sylvain Hauser for help in the mechanical and PCB assemblies and Peter Brühlmeier for PCB designs. Thanks to our two lab secretaries at LSM, Patricia Volanthen and Melinda Mischler, for supporting in all the paper works and solving many issues in daily life during my PhD work.

I would like to thank to many collaborators from outside of the EPFL, Alban Kakulya for his great artistic photo shootings of our prototypes that we created at LSM. Thanks to the armasuisse team, Beat Ott and Peter Wellig, for their collaboration on the development and field test of the Giga-Eye system. Thanks to the University of Geneva Hospital Service of Gastroenterology and Hepatology division and especially to the head of the division Prof. MD Jean-Louis Frossard for discussions on their needs related to colonoscopy applications. I would like to thank to Kurt Müller from Fiberoptic P&P AG company from Zürich for fabrication of my fiber-optic illumination designs.

Acknowledgements

I would like to thank to my friends from Turkey, TUBITAK, Salih Ergün, Tefvik Nur, Halil Özçiçek who have encouraged and helped me to start my journey in Switzerland.

I would like to thank to my friends from EPFL and LSM: To the Serbian Gang; (Brate) Radisav Cojbasic, (Professor Lazar) Nikola Katic, to Vladan Popovic for his friendship and collaboration on technical issues on panoramic imaging systems. To Hossein Afshari for sharing past knowledge on the Panoptic camera systems. To Abdulkadir Akın, for not only extensive collaboration during our joint work on the Giga-Eye system but also his great friendship during my last 4 years of PhD life. To Tuğba Demirci for her great support in revisions of the thesis script and also being a very good friend during the past years. Special thanks to Kerem Seyid for all of his jokes and funny videos from various resources. Thanks to my friends Oğuz&Bilge Atasoy for helping me during my application to the EPFL as an intern. Thanks to my friends for sharing memorable moments in Switzerland; Gülperi Özsema, Gözen Köklü&Ali Galip Bayrak, Tuna&Yasemin Çiftlik, Selman&Şerife Ergünay, Mustafa&Züleyha Kılıç, Şeniz&Deniz Küçük, İpek Baz, Can Baltacı, İrem Boyabat, Zuhul Taşdemir, Ziya Köstereli, Enver Gürhan Kilinc, Okan Yılmaz, Ece Boran, Baran Gözcü, Kiarash Gharibdoust, Elmira Shahrabi&Mahmoud Hadad, Behnoush Attarimashalkoubeh, Reza Ranjandish, Halima Najibi, Cosimo Aprile, Jury Sandrini, Juan Sebastian Rodriguez, Gain Kim, Clemens Nyffeler, Davide Sacchetto, Yüksel Temiz, Alessandro Cevrero, Giulia Beanato, Vahid Majidzadeh, Mahsa Shoran.

Of course, thanks a lot to my parents and sister; my mother Cemile Çoğal, my father Hüseyin Çoğal and my sister İlknur Çoğal. I would like to thank to them not only being a very supportive family during all my life and during this PhD work but also when I was a kid, giving me opportunity to crash things and also repair them at home. I think this was one of the key points lead to my engineering life.

Finally, to my life time team mate, my wife Betül Emgin Çoğal: For standing with me during these challenging times, for being indulgent and supportive during the times that I was too busy with my work and studies, extensive help in the revision process of the thesis and correcting my other English texts, giving her love and devoting herself to our life-time friendship, making this world a meaningful place for me.

06 November 2015 Lausanne

Ömer Çoğal

Abstract

During the last 50 years miniaturization became a key element in human history, since it opens doors for manufacturing new devices that enhances the quality of human life. Camera and imaging systems are following this miniaturization trend as well. Meanwhile, in the imaging domain, usage of multiple aperture camera systems are gaining significance in every aspects of daily life such as entertainment, surveillance, and medical imaging field. Many works focus on multiple camera panoramic and wide field of view imaging systems in industry and academia. Insect vision is the magnificent example of natural multi-aperture wide angle of view imaging systems. There are many attempts to mimic the insect vision capabilities. Current multi-camera systems that are utilizing off-the-shelf components are big in scale and the miniaturization limits are not explored. On the other hand, the multi-aperture systems fabricated using micro-machining techniques cannot meet high resolution requirements due to the micro-machining precision and optical limitations.

This thesis discloses a set of methods to enable development of miniaturized, multiple camera, large angle of view imaging systems. A second target is to explore the smart vision capabilities of the proposed imaging system such as detection of object boundaries by using multiple camera overlapping field of views.

The main methodology is combining the real-time image processing techniques with off-the-shelf miniature cameras. The presented work includes the methods for combining many miniature cameras mechanically to have a compact vision system similar to the insect eyes. Moreover, image processing techniques for creating high quality panoramic images and extracting useful information from the multiple camera images are applied.

Furthermore, digital hardware system design methodologies are implemented for real-time panoramic video generation from the multiple camera video streams. FPGA implementation of the methods are performed and tested. Migration of the system from FPGA to ASIC design is achieved in 40 nm technology node.

In the scope of the thesis work, the proposed methods are implemented and tested by constructing a 5 mm radius hemispherical compound eye, which is capable of imaging a $180^\circ \times 180^\circ$ field of view at 18 mm radial distance.

Abstract

An FPGA implementation of the image processing system is performed, which is able to generate 25 fps panoramic video with 1080×1080 pixel resolution at a 120 MHz processing clock frequency. When compared to the insect eye mimicking systems in literature, the system proposed features more than $1000\times$ resolution increase within the same or even smaller physical dimensions.

What is more, by utilizing fiber-optic technology, a built-in illumination capability is added to the compound eye. This is the first time that a compound eye with built-in illumination idea is reported.

With this work, the current limits of off-the-shelf component based methods in terms of physical dimension and resolution are explored for multiple aperture, miniature insect eye mimicking vision systems. The system is tested inside a human colon model for endoscopic applications like colonoscopy where there is a need for large field of view high definition imagery. The possible applications are not limited to medical domain and due to its miniature size and high quality video capabilities, the proposed methods and the system built can be utilized in search and rescue systems and robotic applications.

Keywords: real time, image processing, FPGA, ASIC, multiple camera, insect eye, compound eye, miniaturization, colonoscopy.

Résumé

Au cours des 50 dernières années , la miniaturisation est devenu un élément clé dans l'histoire de l'homme puisqu'elle permet la fabrication de nouveaux dispositifs augmentant sa qualité de vie. Les caméras et systèmes d'imageries actuels suivent aussi cette tendance de miniaturisation. Entretemps, l'utilisation de systèmes à caméras multiples gagne en importance dans tous les aspects de la vie quotidienne de l'homme tels que le divertissement, la surveillance et l'imagerie médicale. En effet, beaucoup de travaux sont concentrés sur le développement de systèmes d'imagerie panoramiques à caméras multiples et à large champs de vision. La vision des insectes est l'exemple parfait d'un système d'imagerie à ouvertures multiples et à grand angle de champ. Il y a d'ailleurs beaucoup de tentatives à imiter les capacités de la vision des insectes. Les systèmes multi-camera actuels utilisant des composants du commerce sont volumineux et les limites de la miniaturisation ne sont pas explorées. Or, d'un autre côté, les systèmes à ouvertures multiples fabriqués par des techniques de microfabrication ne peuvent atteindre une haute résolution dues à leurs imprécisions et aux limitations optiques. Ce travail de thèse divulgue un ensemble de méthodes permettant le développement de systèmes d'imageries miniaturisés à caméras multiples et à grand angle de champs. Un second objectif est d'explorer les capacités de visions intelligentes du système proposé tel que la détection automatique d'objets d'intérêt.

La principale méthodologie est de combiner les techniques de traitements d'images temps réel avec des caméras miniatures disponibles dans le commerce. Le travail présenté inclut les méthodes permettant de combiner mécaniquement une multitude de caméras miniatures et d'avoir un système de vision compact similaire aux yeux composés d'insectes. De plus, des techniques de traitement d'images sont appliqués afin de créer des images panoramiques de hautes qualités puis d'en extraire des informations utiles.

Par ailleurs, des méthodes de conception matérielle numérique on été implémentées pour la génération de vidéos panoramiques à partir de multiple flux video des caméras. Ces méthodes ont été implémentés et testés sur FPGA, puis une conception ASIC du système a été réalisé. Dans le cadre de ce travail de thèse, les méthodes proposées ont été implémentées et testées en construisant un œil composé hémisphérique de 5mm de rayon, capable d'acquérir des images selon un angle de champ de $180^\circ \times 180^\circ$ à une distance radiale de 18 mm.

Résumé

Une implémentation du système de traitement d'image a été réalisé sur FPGA, capable de générer de la vidéo panoramique à 25 images par secondes avec une résolution de 1080×1080 pixel, cadencée à une fréquence d'horloge de 120 MHz. Comparé aux autres systèmes imitant l'œil composé publiés jusqu'à maintenant, le système proposé présente une amélioration de la résolution d'un facteur 1000 pour une même taille physique, voir plus petite.

De plus, une illumination intégrée a été rajoutée à l'œil composé en utilisant la technologie fibre optique. C'est la première fois que l'idée d'un œil composé avec sa propre illumination intégrée est rapportée.

Avec ce travail, les limites actuels en termes de dimensions physiques et de résolution des méthodes basés sur des composants du commerce sont explorés pour des systèmes de vision imitant l'œil d'insecte. Le système a été testé à l'intérieur d'un modèle de colon humain pour des applications endoscopiques tel que la colonoscopie qui requiert une imagerie avec un grand angle de champ et une résolution élevé. Les possibles applications ne se limitent pas seulement au domaine médicale. La taille miniature et la capacité à acquérir des vidéos de haute qualité de ce système peuvent aussi être utilisés pour des systèmes de secours et sauvetage ou des applications robotiques.

Mots clefs : temps réel, traitement d'image, FPGA, ASIC, caméras multiples, yeux d'insectes, œil composé, miniaturisation, colonoscopie.

Abbreviations

2D	Two-Dimensional
3D	Three-Dimensional
AGC	Automatic Color Gain Controller
AOV	Angle of View
APCO	Artificial Apposition Compound Eyes
ARM	Acorn RISC Machine
ASIC	Application-Specific Integrated Circuit
ASW	Automatic Shutter Width Controller
AXI	Advanced eXtensible Interface
BRAM	Block Random-Access Memory
CCD	Charge Coupled Device
CIF	Common Intermediate Format
CLF	Constant Light Flux
CMOS	Complementary Metal Oxide Semiconductor
CNC	Computer Numerical Control
CPU	Central Processing Unit
CVT	Central Voronoi Tessellation
DDR	Double Data Rate
DDR3	Double Data Rate 3
DE	Data Enable
DFF	D Flip-Flop
DMA	Direct Memory Access
DRAM	Dynamic Random Access Memory
DRC	Design Rule Check
DSP	Digital Signal Processor
DVI	Digital Visual Interface
EDA	Electronic Design Automation

Abbreviations

EDK	Embedded Development Kit
EV	Expected Value
FIFO	First In First Out
FMC	FPGA Mezzanine Card
FPGA	Field Programmable Gate Array
FPS	Frames Per Second
FSM	Finite State Machine
GB	Gigabyte
GI	Gastro Intestinal
HD	High Definition
HDD	Hard Disk Drive
HDMI	High Definition Multimedia Interface
I²C	Inter-Integrated Circuit
IO	Input Output
IP	Intellectual Property
ISE	Integrated Synthesis Environment
ISP	Image Signal Processor
JTAG	Joint Test Action Group
LED	Light Emmiting Diode
LEF	Layout Exchange Format
LUT	Look-Up-Table
LVDS	Low Voltage Differential Signal
MAP	Maximum a Posteriori
MATLAB	Matrix Laboratory
M-FPGA	Master FPGA
MB	Megabyte
MLE	Maximum Likelihood Estimation
MHz	Megahertz
MP	Megapixel
MPMC	Multi-Port Memory Controller
MRF	Markov Random Field
MSB	Most Significant Bit
MSE	Mean Square Error
NPI	Native Port Interface
P&R	Place And Route

PC	Personal Computer
PCB	Printed Circuit Board
POF	Plastic Optical Fiber
PLB	Processor Local Bus
PSNR	Peak Signal to Noise Ratio
PTZ	Pan Tilt Zoom
RAM	Random-Access Memory
RGB	Red Green Blue
RISC	Reduced Instruction Set Computing
ROM	Read-Only Memory
RTL	Register Transfer Level
S-FPGA	Slave FPGA
SAD	Sum of Absolute Difference
SATA	Serial Advanced Technology Attachment
SCL	Serial Clock
SCLK	Serial Clock
SCI	Serial Communication Interface
SDA	Serial Data
SDF	Standard Delay Format
SDK	Software Development Kit
SDRAM	Synchronous Dynamic Random Access Memory
SIFT	Scale Invariant Feature Transform
SMA	Sub-Miniature Version A
SRAM	Static Random-Access Memory
SSD	Solid State Drive
SURF	Speeded-up Robust Features
TB	Terabyte
TCL	Tool Command Language
TSMC	Taiwan Semiconductor Manufacturing Company
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
XGA	Extended Graphics Array
VGA	Video Graphics Array
VDMA	Video Direct Memory Access
VHDCI	Very High Density Cable Interconnect
VHDL	VHSIC Hardware Description Language

Abbreviations

VHSIC	Very High Speed Integrated Circuit
VO	Virtual Ommatidia
YCbCr	Luminance Chrominance-Blue Chrominance-Red

Contents

Acknowledgements	i
Abstract (English/Français)	iii
Abbreviations	vii
List of figures	xv
List of tables	xxi
1 Introduction	1
1.1 Multi-camera panoramic imaging	2
1.2 Miniaturized panoramic imaging	2
1.3 Insect Eyes	5
1.4 Bio-mimicking problem of insect eyes	6
1.5 Contribution of the Thesis	7
1.6 Thesis organization	8
2 State of the Art and Preliminaries	9
2.1 Wide Field of View Imaging	9
2.2 Multi-aperture imaging	10
2.2.1 Miniaturized insect eye mimicking systems based on micro-machining techniques	10
2.2.2 Component integration based methods and image processing for panorama generation	12
2.3 Large FOV imaging for medical endoscopy	13
2.4 Preliminaries of Multi-camera image formation and processing	15
2.4.1 Pinhole camera model and camera calibration	15
2.4.2 <i>Panoptic camera</i>	16
2.5 Thesis Goals	20
3 Opto-Mechanical Aspects for Insect Eye Model	23
3.1 Mapping of insect eye to camera type eye	23
3.2 Effect of single camera dimensions	24

Contents

3.3	Analysis for camera placement	26
3.4	Proposed camera placement for miniaturized camera model	27
3.4.1	Method for minimizing the number of cameras for a certain overlap distance	29
3.4.2	Method for maximizing the number of cameras in a limited volume	33
3.4.3	Generalized solution by using uniform distribution	35
3.5	Proposal of illuminating compound eye	37
3.6	Calibration and Software Based Stitching Analysis	39
3.6.1	Discussion	41
3.7	Conclusion	42
4	Image Processing Techniques developed for Multi-camera systems	43
4.1	Light Field Imaging for a Neural Superposition Virtual Ommatidia	43
4.2	Better quality panorama generation with probabilistic methods	45
4.2.1	Panorama generation as an Inference problem	45
4.2.2	Proposed approach	47
4.2.3	Experimental results	51
4.2.4	Discussion	52
4.3	Inter-Camera pixel intensity differences and its applications	52
4.3.1	Object boundary detection	53
4.3.2	Inter-camera pixel intensity differences as inference evidences	55
4.3.3	Inter-Camera Pixel Intensity Differences as a Quality Measure	56
4.3.4	Discussion	57
4.4	Conclusion	57
5	FPGA Embedded System Design	63
5.1	Single Camera Interface and Image Processing Blocks	63
5.1.1	Interface Printed Circuit Board (PCB)	64
5.1.2	Single Camera Image processing pipeline	66
5.1.3	Camera Model for Test Environment	71
5.2	Memory and Resource Analysis for Full System Implementation	73
5.2.1	Xilinx FPGA Board	73
5.3	System Level Design Considerations	73
5.4	Panorama Image Processing Hardware	74
5.4.1	Pipeline Building blocks	75
5.4.2	Implementation for the probabilistic method for evidences	78
5.4.3	System Implementation	80
5.4.4	Design for pipeline throughput increase	83
5.4.5	Hardware implementation for object boundary detection	86
5.5	Experiments and Results	88
5.5.1	Visual Results	89
5.5.2	Efficiency of the system size	90
5.5.3	Comparison with Different Insect Eye Based Systems	91

5.6	Conclusion	91
6	ASIC Design for Miniature Insect Eye Inspired Image Processing System	93
6.1	Introduction	93
6.2	ASIC Specifications	93
6.2.1	Design Constraints	94
6.3	I/Os Analysis	94
6.3.1	Memory Analysis	97
6.4	FPGA to ASIC Conversion	99
6.4.1	FIFO Design	99
6.4.2	Dual-Clock FIFO	100
6.4.3	Single-Clock FIFO	101
6.4.4	Serial Communication Interfaces	102
6.4.5	I ² C Design	103
6.4.6	Memory Wrappers	105
6.5	ASIC Design	106
6.5.1	RTL Synthesis	106
6.5.2	Place and Route	107
6.6	Logic Simulations	110
6.6.1	RTL Simulation	110
6.7	Conclusion	110
7	Multi-camera large FOV or panoramic imaging system design in macro scale	119
7.1	System Parameters and Requirements	120
7.2	System Architecture	122
7.3	Implementation Results	126
7.4	Conclusion	129
8	Conclusion	131
8.1	A model for miniaturization of insect eye inspired imaging system by using off-the-shelf components	131
8.2	Image processing for seamless compound image generation	132
8.3	Digital system design for image processing	132
8.4	ASIC Implementation	132
8.5	Future Directions	132
A	Analysis and implementations of different Bayer Demosaicing and Automatic White Balancing Methods	135
A.1	Analyzed and implemented Bayer to RGB methods	135
A.1.1	Gradient Based Methods	135
A.1.2	Adam and Hamilton's Method	136
A.1.3	Improved Gradient Estimation	138
A.2	Hybrid Method	139

Contents

A.3	Automatic White Balancing	141
A.4	Gray World Assumption	143
A.5	White Patch	144
A.6	Implementation Details	145
A.7	Results	146
A.8	Bayer to RGB	146
A.8.1	Optical Analysis	147
A.8.2	Measurements	148
A.8.3	Resource Usage	150
A.9	Automatic White Balancing	151
A.9.1	Optical Analysis	151
A.9.2	Resource Usage	152
A.10	Visual Results from the implemented single camera system	153
B	Details of sub blocks FPGA to ASIC conversion	155
B.1	Details of the dual clock FIFO design	155
B.2	Details of serial communication interface choice	157
B.3	The memory generator used for custom memory blocks for ASIC design	159
B.4	I ² C Communication Protocol	159
C	Example video output links for the miniaturized compound eye imaging system	161
	Bibliography	171
	Curriculum Vitae	173

List of Figures

1.1	Different areas of interest for panoramic miniaturized imaging systems (a) 360° drone from photosiphone company, image from [6] (b) different panoramic imaging solutions with wide angle optical systems, from GoPano [7], Kogeto [8], and the Bubblepix [9] companies from left to right.	3
1.2	The problem of narrow field of view in colonoscopy applications.	4
1.3	Anatomy of an insect eye, image from [20]	5
1.4	Insect eye types according to their image formation, image redrawn from [19] (a) apposition (b) optical superposition (c) neural superposition type	6
2.1	Wide FOV imaging classification	10
2.2	Pinhole Camera Model	14
2.3	An example arrangment of circular faces as individual cameras on a hemisphere. 16	
2.4	(a) Example pixelization of the spherical surface, (b) 3 vectors \vec{t} , \vec{u} , \vec{v} representing camera orientation and positions	17
2.5	Projections of camera centers contributing in direction $\vec{\omega}$ onto planar surface normal to $\vec{\omega}$	18
3.1	The relation between insect and human eye resolving capability	24
3.2	The camera chosen for the implementation, (a) illustration for the physical dimensions of the single camera (b) a close photo view of the camera taken under a microscopic lens	25
3.3	Horizontal axis shows the angle of view of each individual camera, vertical axis shows the required number of cameras for having virtual ommatidias at distance $R_{virtual} = 8, 10, 12, 25, 30, 40$ mm with a $R_{physical}=5$ mm.	28
3.4	New camera placement model (a) single camera circular surface model on dome y-z plane (b) Geometrical relations of the cameras in one quarter of the dome in y-z plane	29
3.5	Illustration for the layers for extending the camera placement around the hemisphere	31
3.6	Illustration for the overlap analysis for 3 cameras positioned at layer 0 with $\beta_0 = 15^\circ$. The smallest latitudinal angle from the north-pole that is not seen by any camera is $\theta_{uc0} = 19.6^\circ$	32

List of Figures

3.7	Analysis of the growth of the camera positions for single camera AOV=64°, starting in a overlap range $R_{vir}/R_{phy} = 50/5 : 8/5$. The proposed method is plotted in red color and the method in [65] in blue color.	33
3.8	The camera positions on the final design for prototyping, (a) 24 camera positions, (b),(c) and (d) the mechanical model drawing from different viewing angles . .	35
3.9	The comparison of the growth of the number of cameras for previous method [64] and the CVT based method proposed. Horizontal axis shows the angle of view of each individual camera, vertical axis shows the required number of cameras for having virtual ommatidias at distance $R_{virtual} = 8, 10, 12, 25, 30, 40$ mm with a $R_{physical}=5$ mm.	36
3.10	An example for the camera positions obtained from CVT based method for $R_{vir} = 18$ mm and $R_{vir} = 5$ mm, and matching the approximate positions with our final design.	37
3.11	The final mechanical design for the prototype, (a) and (b) Solidworks Drawings ,(c) and (d) the fabricated and assembled prototype	38
3.12	The calibration environment and single camera frames of the prototype (a) calibration tube with 35 mm diameter (b) 24-camera single frames used in calibration (c) 180° × 180° image generated from the 24 single frames at calibration procedure.	41
4.1	Virtual ommatidia sampling concept with the proposed prototype	44
4.2	Example graph representation for the panoramic image	49
4.3	The proposed method for accurate prior estimation from spherical model, (a) the previous planar model, (b) proposed spherical arrangement for estimating the priors	51
4.4	The comparison of the probabilistic inference method (d) with nearest neighbor method [64](a), linear blending method [70](b),linear blending with Gaussian smoothed weights [66] (c). The images have 1920x512 resolution, generated in MATLAB using the output images from our miniaturized 24-camera system. . .	58
4.5	The comparison for 2x zoomed on the polyp to show the effect on object boundaries. nearest neighbor(a), linear blending(b), gaussian smoothed linear blending (c) and proposed probabilistic inference method (d). The images have 512x512 resolution, generated in MATLAB using the output images from our miniaturized 24-camera system.	59
4.6	The results obtained from the proposed method for object boundary detection. (a) The object boundary map in gray scale, (b) Constructed panorama image (c) the regions seen only one camera in shown in black (d) final object boundary map with a threshold $th_{ob} = 0.2$	60
4.7	The results obtained from the proposed method for object boundary detection. (a) The object boundary map in gray scale, (b) Constructed panorama image (c) the regions seen only one camera in shown in black (d) final object boundary map with a threshold $th_{ob} = 0.6$	61

4.8	The results obtained from the proposed method for SAD based interpolation. (a) The resulting image in $180^\circ \times 180^\circ$ angle of view 1080x1080p resolution, (b) The same reconstruction with method in [74] (c) the zoomed image on the polyp object for SAD based method (d) the zoomed image with previous method . . .	62
5.1	Single image sensor block diagram	64
5.2	Circuit block diagram for the PCB, simplified view for one camera interfacing .	65
5.3	The custom PCB designs for camera interfacing. (a) camera interface PCB for 2 cameras, (b) and (c) main board PCB supporting 4 and 30 cameras respectively.	65
5.4	The custom single camera pipeline as an FPGA AXI slave IP, and the embedded system designed for testing the interface	66
5.5	Two possible configuration for color image sensing [85] (a) three sensors arrangement for capturing R,G,B channels separately, (b) using Bayer pattern with a single sensor.	68
5.6	Individual and combined representation of Bayer patterns (a) Possible regular Bayer patterns [86] (b) A 5x5 bayer matrix [87]	69
5.7	Schematic of the bilinear Bayer to RGB converter architecture	71
5.8	The embedded system block diagram	74
5.9	The panorama pixel generation pipeline	75
5.10	The block diagram used for angle and Cartesian vector calculation	76
5.11	The block diagram used for camera search and distance calculation	77
5.12	The block diagram for implementation of the probabilistic evidence calculation	79
5.13	The pipeline analysis of the system	80
5.14	FPGA system block diagram	81
5.15	Interface of the panorama generation pipeline with camera frame memories and video DMA output for DDR3 RAM	82
5.16	The tasks and the flow of the Microblaze embedded processor firmware	83
5.17	The design and test flow methodology followed during development and tests.	84
5.18	The proposed hardware for object/poyp boundary detection	86
5.19	An example of the hardware generated output image for colon boundary detection. (a) the $180^\circ \times 180^\circ$ panoramic image in 1024×1024 resolution (b) the possible boundary regions are marked with green color by the hardware described.	88
5.20	The complete prototyping chain used for the implementation of system	88
5.21	The complete system built and used for experiments	89
5.22	An example of the output image from the system	89
5.23	An example image for the USAF 1951 measurements, the image is captured in 1024×1024 resolution, at 25 mm distance from the hemispherical camera center.	91
6.1	Capsule Endoscopy concept with miniaturized insect eye imaging system . . .	94
6.2	ASIC design conceptual illustration with EndoPano imaging tip.	95
6.3	ASIC design top level (a) and block diagram (b)	96
6.4	Panorama generation block diagram.	97
6.5	Raster scanning principle for different DE_out pulse width.	97

List of Figures

6.6	Dual-clock FIFO block diagram.	101
6.7	Single-clock FIFO block diagram.	102
6.8	I ² C block diagram.	103
6.9	Register update process.	104
6.10	Register array structure.	105
6.11	Memory wrapper example.	106
6.12	Physical design steps.	108
6.13	First P&R design.	114
6.14	Second P&R design.	116
6.15	An example of panoramic frame obtained through RTL logic simulation of the ASIC system.	117
7.1	Top-level block diagram of the system hardware architecture	124
7.2	The complete omnidirectional imaging and recording system (Giga-Eye), overall system dimensions are 56x48x78 cm	127
7.3	Omnidirectional image obtained with the Giga-Eye system at 21.6 MP resolution showing the central campus square of EPFL, and two selected details (sub-regions) in this image. This omnidirectional image corresponds to one single frame of the 30 fps video obtained by the system.	128
7.4	Omnidirectional image obtained with the Giga-Eye system at 82.3 MP resolution. This omnidirectional image corresponds to one single frame of the 9.5 fps video obtained by the system. Flying plane and the moving car are shown in sub-windows	128
7.5	Measured coverage map of the omnidirectional imaging system showing a high pixel redundancy especially close to the equator. The color labels indicate the number of the overlapping individual camera AOVs	129
A.1	Schematic of the blocks for a gradient based Bayer to RGB conversion system	140
A.2	Block diagram of the hybrid Bayer to RGB converter architecture	141
A.3	Spectral power distribution of various common types of illuminations: (a) sunlight, (b) tungsten light, (c) fluorescent light, and (d) LED [100]	142
A.4	Spectral sensitivities of: (a) the three types of cones in a human eye, and (b) a typical digital camera [100]	143
A.5	Block diagram of a part of the gray and the white patch automatic white balancing algorithm	146
A.6	Block diagram of the common part of the gray and white patch automatic white balancing algorithm	146
A.7	Interpolation results using a 3 × 3 window	147
A.8	Bayer CFA interpolation results using a 5 × 5 window	149
A.9	White balance comparison for images taken under a tungsten light source	152
A.10	Image taken in human guts with balanced light	152
A.11	Image taken with a live system with different AWB methods	154
B.1	Multi-flop synchronizer [103].	156

B.2	Example of data incoherency [103].	157
B.3	ARM Artisan Physical IP GUI.	159
B.4	I ² C data transfer [106].	160

List of Tables

1.1	Current solutions for panoramic colonoscopy	4
2.1	Summary of curved optics/electronics based solutions for multi-aperture insect eye imaging.	12
3.1	Single image sensor opto-mehcanical specifications	25
5.1	Single image sensor electrical specifications	63
5.2	Programmable features for single camera IP	67
5.3	Specifications of VC707 FPGA board chosen for embedded system and image processing hardware implementation	73
5.4	Resource overview of Virtex-7 XC7VX485T FPGA chosen for embedded system and image processing hardware implementation	74
5.5	List of critical resource consuming blocks in the camera search and distance calculation module.	79
5.6	The resource usage results of the sub blocks and the full system on the Virtex7 FPGA for the first implementation	82
5.7	The resource usage results of the sub blocks and the full system on the Virtex7 FPGA for the increased throughput pipeline	86
5.8	Virtex-7 FPGA resource usage overhead for object boundary detection method	87
5.9	Comparison with insect eye systems in terms of resolution and size	92
6.1	System inputs for ASIC design	98
6.2	System outputs for the ASIC design	98
6.3	Possible pins configurations for ASIC design	99
6.4	List of ASIC memory elements	100
6.5	FIFOs present in the ASIC design	100
6.6	Channels bits partition for configuring the internal register arrays and memories, which contains the calibration information	105
6.7	Configuration registers specifications for the ASIC design	112
6.8	Memory wrappers implemented for the ASIC design	113
6.9	Critical paths for different synthesized gate-level net-lists of the ASIC design . .	113
6.10	Core area percentage occupied by macro cells of the first ASIC design configuration	113

List of Tables

6.11	First ASIC design P&R summary reports	115
6.12	First ASIC design power dissipation report for a slow/slow corner VDD=0.81V, $T_j=125^\circ\text{C}$	115
6.13	Second ASIC design P&R summary reports.	115
7.1	Properties of the omnidirectional imaging system.	120
7.2	System Constraints to generate 30 fps 21.6 MP Omnidirectional Video.	122
7.3	System Constraints to generate 9.5 fps 82.3 MP Omnidirectional Video.	122
7.4	Comparison of the Giga-Eye with existing high-resolution omnidirectional camera systems.	129
A.1	PSNR comparison of the Kodak image set 19th image for different Bayer CFA interpolation methods	150
A.2	Resource usage of the different Bayer-to-RGB algorithms	150
A.3	Resource usage of the different AWB algorithms	153
B.1	Comparison for different serial communication interfaces	158

1 Introduction

In World War II, during the bloody Normandy landings, famous with the name D-day, number of lives lost was estimated as 10000. Today, each year, in 1 week that many people are dying because of bowel cancer. This means while you are reading these lines for a minute time interval, 1 person lost his/her life because of colon cancer. In last 50 years there are many efforts focused on offering new vision systems to have a better vision inside human body to solve such problems. For improving the detection rate of the cancer suspicious tissues, which are named as *Polyps*, large angle field of view (FOV) vision systems with smart vision capabilities are one of the key directions. Multiple aperture camera systems are one of the members of the vision systems family which offer large angle and smart vision features. Since they are not only providing large angle of view but also offering unique information by providing images from different view points, they are becoming alternative to the single camera large optic imaging systems.

During the last 50 years, miniaturization became a key element in human history since it opens doors for manufacturing new devices that enhance the quality of human life. Camera and imaging systems are following this miniaturization trend as well. Meanwhile, in the imaging domain, usage of multiple aperture camera systems are gaining significance in every aspects of daily life such as entertainment, surveillance, and medical imaging fields. Many works are focused on multiple camera panoramic and wide field of view imaging systems in industry and academia. As in every scientific development, nature is the first place to investigate for miniaturized multiple aperture vision as well. Insect vision is a magnificent example of multi-aperture wide angle of view imaging systems. There are many attempts to mimic the insect vision capabilities. Current multi-camera systems that are utilizing off-the-shelf components are big in scale and the miniaturization limits are not explored. On the other hand, the multi-aperture systems fabricated using micro-machining techniques, cannot meet high resolution requirements due to the micro-machining precision and optical limitations.

This thesis discloses a set of methods to enable development of miniaturized, multiple camera,

large angle of view imaging systems. A second target is to explore the smart vision capabilities of the proposed imaging system such as detection of objects boundaries by using multiple camera overlapping field of views. In the remaining of this chapter, recent developments in the field and the sources of inspiration for this work with some specific life problems are explained in brief.

1.1 Multi-camera panoramic imaging

Panoramic imaging with multiple cameras has become a trend in the recent years. There is a wide variety of multi-camera panoramic imaging systems used in different applications such as large area surveillance, 360° video capturing or telepresence. Examples for such kind of systems are : Ladybug5 camera from Pointgrey [1], which is a 6 camera system with a 197 mm diameter cylindrical case with 160 mm height. The system can provide 2048 × 2448 360° × 162° panoramic video stream at 10fps with 6 cameras of each 5 MP. Another recent system is from Nokia named as ozo [2], 360° × 180° full spherical camera with 8 2Kx2K resolution cameras. Another semi-panoramic camera is Panacast [3] from Altia systems, with around 10 cm diameter, 3 cameras giving an output of 4K resolution at 180° × 54° field of view (FOV). There are also different applications with spherical cameras for entertainment purposes like [4, 5]. In all these solutions dimensions are ranging from 60 mm-200 mm in diameter. They are bulky systems and designed for far field imaging, and they cannot be considered for the applications which need miniaturized panoramic imaging systems.

1.2 Miniaturized panoramic imaging

Miniaturization in general is the key trend for many years in different areas of interest in human life. Different aspects of miniaturization in different fields are driven by these trends and needs. In imaging and vision systems, with the interest in mobile applications, robotics and medical systems, the miniaturization concepts have been developed and improved in the last century.

In the field of robotics the recent developments of the drones bring new requirements for machine vision systems utilized in drone applications [10]. Since such flying vehicles are light-weight, power-limited systems and requires large FOV imaging, miniaturized panoramic imaging solutions are becoming crucial for such systems. Likewise, in the mobile phone and handheld-devices industry, there are many new systems attempted to add the panoramic imaging capabilities to the new generation mobile devices [11]. Again, the device sizes are becoming the key aspect in this area. Hence, there is a need for panoramic miniaturized imaging solutions in this field. Some of the example solutions from different providers to the applications in the drone and smart phone fields are illustrated in Fig.1.1.

In addition to the mentioned applications above there are also certain applications in the medical imaging field, which requires panoramic imaging. Colonoscopy and Laparoscopic

1.2. Miniaturized panoramic imaging



(a) Example for panoramic imaging in drones



(b) The current solutions for panoramic imaging in smart phones

Figure 1.1 – Different areas of interest for panoramic miniaturized imaging systems (a) 360° drone from photosiphone company, image from [6] (b) different panoramic imaging solutions with wide angle optical systems, from GoPano [7], Kogeto [8], and the Bubblepix [9] companies from left to right.

surgery are very well defined applications as an example of these applications. In Laparoscopic or minimal invasive surgery (MIS) where an imaging device is inserted into human body through a minimal hole, it is crucial to have miniaturized imaging and capability of large FOV to be able to see the whole surgery area. Therefore, the need for miniaturized panoramic imagery is a desired requirement. Current systems in MIS domain utilize relatively small field of view cameras and need movement of the imaging equipment during operation to see certain areas on the area of interest.

Each year, over six hundred thousand people lose their lives due to the colon cancer in the world [12], which means more than 10000 deaths per week in average. Even though the colonoscopy examination procedures are applied, there is a certain amount of miss rate in colonoscopy procedures due to narrow field of view (FOV) imagery employed in current systems. To illustrate the problem in colonoscopy, in Fig.1.2 a simplified sketch is represented for the problem of narrow field of view in the colonoscopy applications. Since the human colon has a folded structure, while the colonoscope moves toward the colon, it misses the behind-fold regions at the peripheral areas with respect to the forward and backward movement of the colonoscope.

In Table-1.1, a feature comparison of the current colonoscopy devices from different

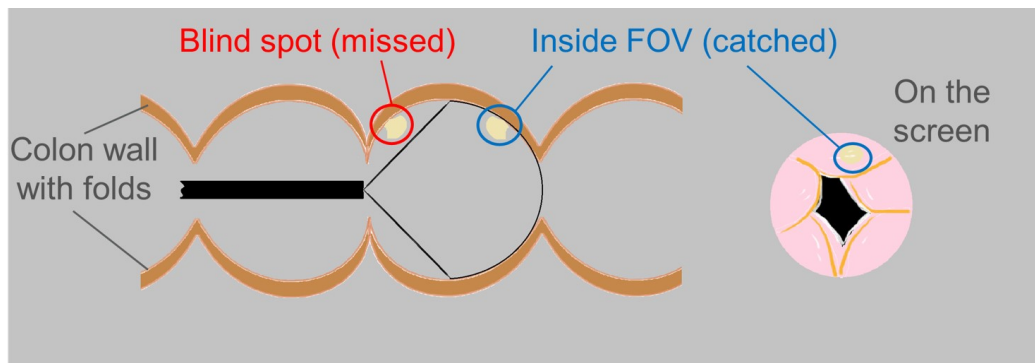


Figure 1.2 – The problem of narrow field of view in colonoscopy applications.

companies, Olympus, Avantis, Endochoice, Naviaid and Giview is given. For example, the solutions from Endochoice [13] and Avantis [14] utilize 3 cameras and do not have a compound panoramic image which ends up with 3 separated windows. Hence it results in difficulty for the operator to follow the whole imaging area. Other solutions from Naviaid [15] and Giview [16], utilize parabolic-like mirror solutions. Their monolithic optical solution occupies the whole diameter of the colonoscopy device due to their size and does not allow any working channels, which is required to remove the polyps or make small operations during examination. A recent solution from Olympus [17] accommodates a forward large angle of view lens and a surrounding parabolic mirror. The solution has a large FOV reflected onto a single image sensor, which makes it limited to the single image sensor's resolution. Therefore, the works and attempts from different companies show that there is a certain need for wide FOV imaging in colonoscopy domain.

Table 1.1 – Current solutions for panoramic colonoscopy

Company	Solution	# cameras	FOV	multi-camera image processing	Main Drawbacks
Olympus [17]	large FOV optics	1	N/A	no	single camera limited resolution
Endochoice [13]	multi-camera	3	330° × 330°	no	seperated windows hard to follow
Avantis [14]	multi-camera	3	330° × 330°	no	seperated windows hard to follow
Naviaid [15]	large FOV mirror	1	180° × 180°	no	size doesn't allow working channel
Giview [16]	large FOV mirror	1	180° × 180°	no	size doesn't allow working channel

From the applications and needs from the real-world problems, the first question seeking for an immediate answer is how to have a wide FOV or (panoramic) imaging system, which has relatively high resolution in a limited volume that can fit into commercial applications like flying drones, smart phones and medical applications like colonoscopy and minimal invasive

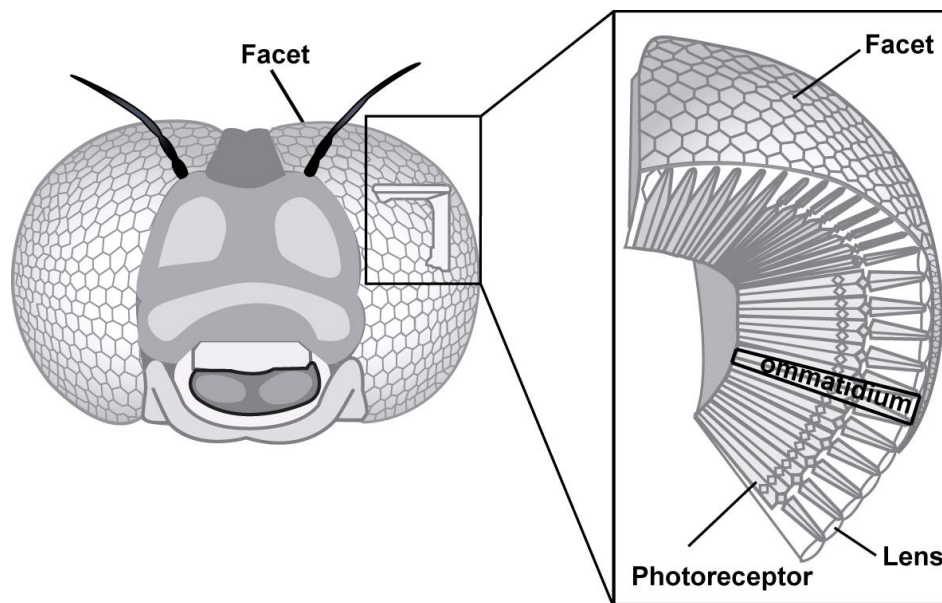


Figure 1.3 – Anatomy of an insect eye, image from [20]

surgery.

1.3 Insect Eyes

The nature has its solutions for panoramic imaging. The compound eye structure of insects is a good example of this kind of solution, which has been a topic for biologists. In Fig.1.3, the structure of an insect eye is shown. The insect eyes, at their outer shell, have small lens facets adjacent to each other, looking into different directions. In this way, they have the 360° complete vision of their surrounding world. According to their type of vision, they have one or more sensors named as rhabdoms under each lens. Each unit with a lens and sensor is called as ommatidia. These units collect light from different angles with the help of their lenses and turn the absorbed light into a signal with the corresponding sensor and interpret the information in their brain [18].

The types of insect eyes are divided into different categories according to the way they form the image. These types are apposition, optical superposition and neural superposition [18, 19]. In apposition type, the lens and sensor pairs are separated from neighboring pairs forming a small portion ($1^\circ - 2^\circ$) of the full field of view. In optical superposition type, the light collected by adjacent lenses are focused and superposed optically on each of the neighboring sensors, creating a lumped information of the surrounding light field on each sensor. In neural superposition type, the sensor signals from the adjacent cell rhabdom are superposed in the signal domain on the way to the brain or inside the brain. The illustration is given in Fig. 1.4 for the different kinds of insect eyes.

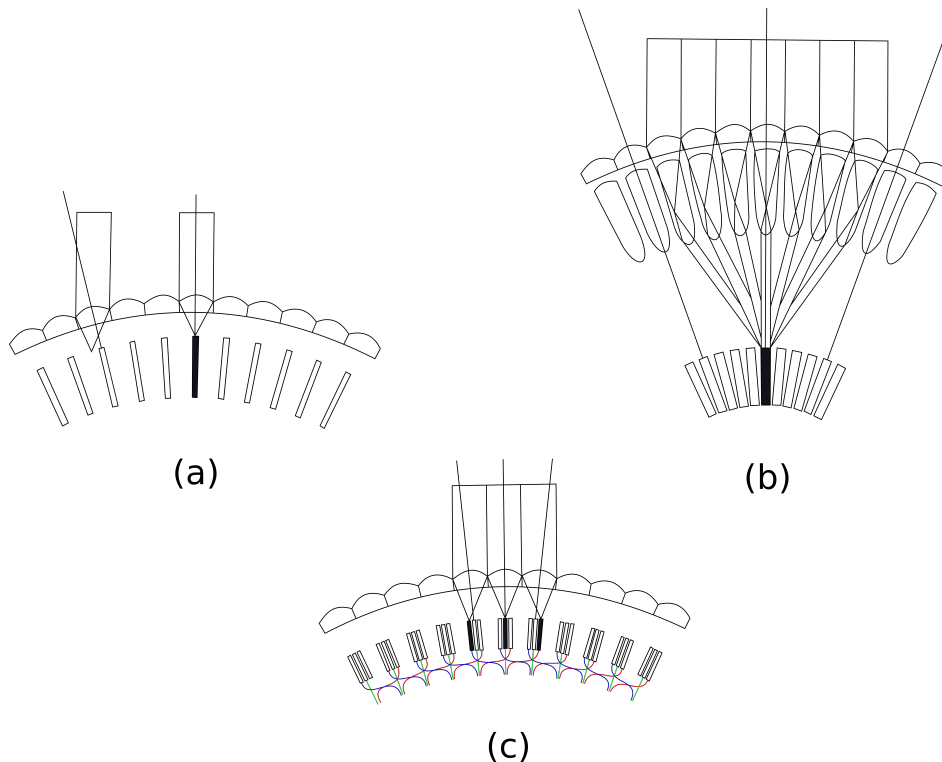


Figure 1.4 – Insect eye types according to their image formation, image redrawn from [19] (a) apposition (b) optical superposition (c) neural superposition type

1.4 Bio-mimicking problem of insect eyes

There are many attempts made to copy the insect eyes from nature to design vision systems for large FOV or panoramic imaging. The main reasons for mimicking these natural vision systems can be counted as follows. First of all, insect eyes provide large FOV. When compared to the single wide FOV lens camera-type eyes, they provide more scalability and uniform resolution. Due to the distributed behavior, they provide fast decision information. However, they have main drawbacks as the limited resolving capability due to the limited size and optical limitations like diffraction. There is a review [21], which states that blind one-to-one mimicking of the insect eyes is not the best approach. Thus, a system which is both small in dimension and has high resolving capability and large FOV is not an easily achievable design.

The mentioned trade-off items are forming a search space for insect eye mimicking problem. This is one of the main driving motivations for this thesis to search and find an optimal solution for having a miniaturized high resolution and large field of view imaging system. This imaging system is ornamented with its optical and electronic components to provide a real-time video capability (minimum 25 frames per second) as well. Therefore, the main goal is to propose a set of methods for designing and manufacturing insect-eye inspired large FOV multi-aperture systems. As a following objective, it is aimed to explore the size and the quality limits of the methods proposed.

1.5 Contribution of the Thesis

In this thesis, a set of methods for generating multi-camera miniaturized $360^\circ \times 90^\circ$ or $180^\circ \times 180^\circ$ field of view imaging system are proposed. Different uses of the overlapping field of views of the cameras are proposed and tested as image processing methods. For a complete solution with the proposed image processing techniques, hardware system designs are performed, implemented and tested for generating real-time panoramic video output from the multi-camera system. Both FPGA (field programmable gate array) and ASIC (application specific integrated circuit) implementation of the hardware systems are achieved. This is the first time such a system with the resulting capabilities is proposed, manufactured and tested for endoscopic applications. Moreover, with the achieved size and resolution results, the methods and designs can be extended to the robotic and smart-phone applications. The details of the contributions are described below.

Analysis of design requirements for miniaturization An analysis of the previous methods for component integration level multi-camera systems is done.

A camera placement method optimized for component sizes and miniaturization is proposed for generating millimeter scale opto-mechanical hemispherical imaging apparatus in 10 mm diameter with fiber-optic illumination capability. A prototype for the mechanical housing of the multi-camera system according to this model is designed, manufactured, assembled and verified. The multi-camera imaging apparatus is integrated to the designed digital image processing hardware system; functionality and capability of the dome is verified by this way.

Illuminating compound eye for endoscopy A distributed illumination for the designed compound eye is introduced within the system created. There are no built-in illumination capability reported before for the compound eye systems. In this way, an active illumination type compound eye is created, which gives opportunity to work in dark environments like in-vivo applications.

Image processing algorithms for multi-camera imaging Inspired from neural superposition of the insect eyes, proposal of an inference method for generation of compound panoramic image from multiple camera information is presented. Different methods are proposed taking advantage of information from overlapping FOV of multi-cameras, which is not possible to be achieved by its single camera counterparts. These methods can be listed briefly as: utilizing parallax information as object edge features, using the inter-camera differences as quality measure, and using the overlapping camera intensity values as evidence for generating a final compound panoramic image.

Chapter 1. Introduction

Miniaturization at electronic circuit and systems level Design of a digital system for real-time panoramic video generation on FPGA is achieved. The components of the digital system are minimized and designed to be compatible for migrating to an application specific integrated circuit (ASIC). The FPGA system is run and verified on Xilinx VC707 FPGA board. Following the FPGA design and verification, the digital system is migrated to ASIC design using a 40 nm technology node.

Work done on big scale devices Apart from the contributions on miniaturized systems, design of a very high resolution panoramic image acquisition system is performed for large area surveillance. The system is designed, assembled, manufactured and tested on the field.

1.6 Thesis organization

This thesis is structured as follows. Chapter 2 explains the state-of-the-art miniaturized and panoramic imaging systems and preliminary information that are used as a base for some parts of the work presented here. Chapter 3 explains the analysis and details for modeling the insect eye for miniaturization with pinhole cameras. The camera placement methods are revised and an optimization of the camera placement method for the target system is disclosed. In Chapter 4, the image processing contributions for generating seamless compound panoramic image from multi-camera images are described. The methods take the advantage of multi-camera overlapping field of views are presented. Chapter 5 discloses the real-time digital circuit and system design and implementations for the fabricated miniature compound eye. In Chapter 6, details of ASIC design for miniaturization at electronics level are given. In Chapter 7, the work done on the macro-scale multi-camera devices is presented. Chapter 8 concludes the thesis and includes some future perspectives for further research directions with a discussion.

2 State of the Art and Preliminaries

In this chapter, state of the art for miniaturization of insect eye mimicking systems as well as multi-camera systems are discussed. Moreover, the current solutions for the applications are presented where panoramic or large field of view (FOV) miniature imaging systems are required. Then a brief discussion on the current implementations of the real-time image processing techniques for multi-camera compound panoramic image reconstruction is given. Finally, the new directions taken in the sphere of the thesis and the variations between the current work and the previous works have been explained in detail.

2.1 Wide Field of View Imaging

In general the wide FOV imaging systems can be classified into 3 main groups as shown in Fig.2.1.

In the first group the well-known pan-tilt-zoom (PTZ) type cameras are used [22]. Generally PTZ cameras are utilized for large area surveillance. They have a single camera which can be rotated in 3 axis. In this way the system can scan and have the panoramic image of the visualized environment. The main drawback of such systems is that they cannot capture all the directions concurrently and cannot create a 360° FOV video in real-time. Also miniaturization for such systems needs complex and high precision micro-machining due to the movable parts.

Another group of methods for wide FOV imaging includes single wide lens or mirror based solutions like fish eye lens [23]. There are also works in which miniaturization of such systems are considered. For example in [24], a miniaturized hyperbolic mirror based solution is proposed for flying robot applications. They achieved a 128x64 resolution wide angle of view imaging system in a 14.4 mmx11.4 mmx11.4 mm cubic shape. In [25] and [11], a miniaturized lens technology is proposed for large field of view endoscopic imaging and

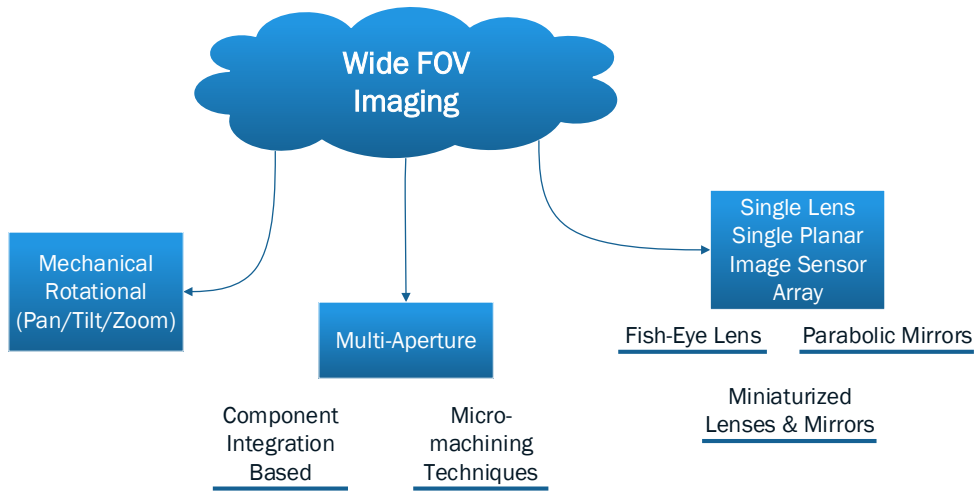


Figure 2.1 – Wide FOV imaging classification

consumer electronics. The authors reported a 6.5 mm diameter and 3.25 mm high cylindrical lens solution without the electronic components. The general drawback of such solutions is that they are limited to the resolution of a single image sensor under the monolithic lens optics and they have high level of distortion due to the optical design. Also they don't have the multi-view point capabilities such as uniform resolution distribution in real-time or depth sensing.

2.2 Multi-aperture imaging

The multi-aperture imaging is another alternative for wide FOV imaging and is also in the scope of the work represented in this thesis. In multi-aperture imaging systems, more than one optical axis and more than one sensing elements are utilized, resulting in a compound imaging system. The surrounding light field that is in the FOV of the compound imaging system is captured and sensed by the multiple light gathering elements. This configuration is also named as *Plenoptic* camera [26]. This type of imaging systems are generally designed in a planar fashion [27] and methodologies for image reconstruction from plenoptic camera arrays are proposed in literature [28]. Wide field of view plenoptic camera variants are gaining significance. In the following sections methods which are based on multi-camera imaging are analyzed in detail.

2.2.1 Miniaturized insect eye mimicking systems based on micro-machining techniques

This group of methods based on one-to-one morphological replica of the natural insect eyes. They usually try to mimic the natural counterparts by focusing on designing and replicating

the smallest units, ommatidias with the help of micro-machining methods and planar and/or curved electronic sensors.

The microlens array based solutions can be classified into two main groups according to their shape as planar (2 dimensional-2D) and curved (3 Dimensional-3D). The planar shape solutions [29–33] are limited in terms of field of view but they are easy to fabricate when compared to the curved ones. Due to their planar nature, 2D systems are far away from providing 360° FOV.

The other group in the literature, 3D or curved type microlens based implementations began to be developed more recently [21, 34–37].

In [35], an optical solution with 3D microprism array and a 2D microlens array on top of a 2D sensor array is proposed. They utilized complex and high precision micromachining processes such as regular diamond turning, diamond broaching, slow tool servo and micromilling processes. The diameter of the optical system was implemented as 20 mm. The authors did not report an assembly with a sensor array but they made measurements with a charge coupled device (CCD) array based camera with regular zoom optics.

In [38], a multi-aperture insect eye inspired system is described. Their design is inspired by housefly with a main objective of improving motion detection capability of insect eye inspired imaging systems. The design presented in [38], utilizes a lens-fiberoptic pair to mimic each facet of the compound eye of the insects. As sensing elements they utilized photodarlington devices, which are simple light capture elements generating output current proportional to the gathered light. Although the authors reported improvement in motion detection capability, they reported a 7-pixel resolution in a diameter around 13 mm.

In [36], a methodology is introduced for multi-lens curved imagers and authors proposed integration techniques of elastomeric compound optical elements with thin silicon photo detectors and a method for deforming these integrated sheets to get hemispherical surfaces. They reported a number of 180 bio-mimicked ommatidias, capable of imaging in a 360° × 80° degree field of view and physical dimensions around 12 mm in diameter. Authors compare their image definition capability with the eyes of fire ants (*Solenopsis fugax*) and bar beetles. They have also reported ray tracing simulation results but not real captured images with their proposed imaging system. Although this approach is promising for further miniaturization, it lacks of the trade of between the number of ommatidias and physical restrictions for building the photo detector and the lens of each ommatidia.

In [37] another micro machining approach is proposed. In their system they have achieved 180° × 60° with 630 ommatidias on a partial cylindrical shape and they reported interommatidial angle of $\Delta\phi = 4.7^\circ$ and acceptance angle of $\Delta\rho = 4.7^\circ$. Their methods are based on complex integration and fabrication techniques. While the final spatial resolution is not notably large, the main advantage of this solution is reported as the high frame rate which allows optical flow estimation. The details of the system are also described in [39].

Both [36] and [37] are combination of micro-lens and micro-integration based methods, which are trying to copy natural insect eyes one to one basis. Therefore, as the insect eyes in nature have some limitations, the aforementioned methods also display similar diffraction limitation problem [19] which leads to low resolution with the given limited diameters [18, 40]. Because of this problem, although they are promising for further miniaturization from the mechanical and manufacturing point of view, these blind mimicking methods have to tackle with the need of relatively high resolution despite the diffraction limit of the individual lenses.

Table 2.1 – Summary of curved optics/electronics based solutions for multi-aperture insect eye imaging.

Solution	# of pixels	Field of view	size (diameter)	frame rate
[35]	N/A	$180^\circ \times 180^\circ$	20 mm	N/A
[36]	180	$330^\circ \times 330^\circ$	12 mm	N/A
[37]	630	$180^\circ \times 60^\circ$	12 mm	300 fps

2.2.2 Component integration based methods and image processing for panorama generation

There are off-line software implementations based image features on image bundle adjustment and local alignment [41], usually for cylindrical or 1 dimensional camera arrangements, or extended versions for 3D camera arrangements [42]. There are methods using different feature extraction approaches for the alignment like scale invariant feature transform (SIFT) [43], speeded-up robust features (SURF) [44]. Since these methods are invariant to scaling, illumination and rotation of the multiple views, they result in high quality and seamless panoramic mosaics. There are many software based panorama generation schemes proposed using these methods [45–47]. Although there are attempts [48, 49] to implement these algorithms for real-time video generation, they are still memory hungry and far from utilizing for multi-camera panorama generation and hardware implementation for miniaturized single application specific integrated circuit (ASIC) system implementation.

In [50], a design which includes a pre-processing calibration for intrinsic and extrinsic parameters of 5 cameras has been proposed. Their system accommodates a cylindrical 5-camera array and one camera for the top view. They describe an alignment method for the images captured from 5 cameras in lateral circular arrangement and a blending scheme at the camera overlap regions similar to methods in [41, 42]. They make a mapping of the image video frames on to a cylindrical projection and keep them on external dynamic random access memory (DRAM). Moreover, the maximum frame rate is 15 fps due to the limited bandwidth of the memories. Thus, their design is far from 25 fps real-time video and miniaturization due to the bulky external memories.

In [51], a 6-camera system in a cylindrical arrangement is represented. They used a

Lucas–Kanade algorithm [52] based approach for extracting extrinsic camera parameters and used graphic processing unit (GPU) based system on personal computer (PC). Their system is in the order of 30 centimeters diameter due to the bulky single cameras they use. Since their system also needs big memory resources at gigabyte order of magnitudes, the compatibility for miniaturization is not feasible.

In [53], a combination of mammalian vision and insect vision is proposed. It is a 3-camera array and the authors focused on the lens design for individual image sensors and the image processing is done offline by using free stitching software. They reported a maximum FOV of 130° degrees and final pixel resolution of 643 × 366 but not a real-time panoramic video construction capability.

As a conclusion, the component integration based methods are capable of providing high spatial resolution. Many methods proposed for panoramic compound image reconstruction by using systems designed with off-shelf component based multi-camera systems. However, the methods proposed for designing multi-camera systems are not optimized and utilized for miniaturized large field of view imaging. This is mainly because of two reasons. For some of the solutions the real-time image processing demands could not be met by the proposed panorama reconstruction algorithms. And the second reason is that, for the ones which meet the real-time constraints, with the utilized bulky component sizes, miniaturization is not considered.

2.3 Large FOV imaging for medical endoscopy

There are many studies that show the need of large field of view, or panoramic imaging capability for endoscopy [25, 54–59]. Most of these systems are optical solutions with either mirror or wide angle lens based solutions. The main drawbacks of these solutions are limited resolution, optical distortions and lack of multi-perspective view. In [60], new technologies to overcome the narrow field of view problem in colonoscopy procedures are reviewed. One group of these emerging technologies in the market does not have the capability of giving a compact view [13, 14]. Hence, it leads to an increase in the diagnostics time [61]. The other group which has limited resolution capabilities, nonuniform resolution and optical distortions uses mirror or lens based solutions [15–17]. The latter group does not offer multi-point view capability to have 3D imaging capabilities either.

In the capsule endoscopy domain, there is a recent device from CapsoVision company, their device is named as CapsoCam [62]. This device has 4 cameras arranged in a cylindrical placement around the capsule body and capable of collecting 360° × 50° panoramic video at 5 frame per second from each camera. Their system is not offering a compound image from their single camera images and is not capable of capturing full spherical field of view at one shot.

In order to have a uniform and high resolution imaging capability in such applications, the

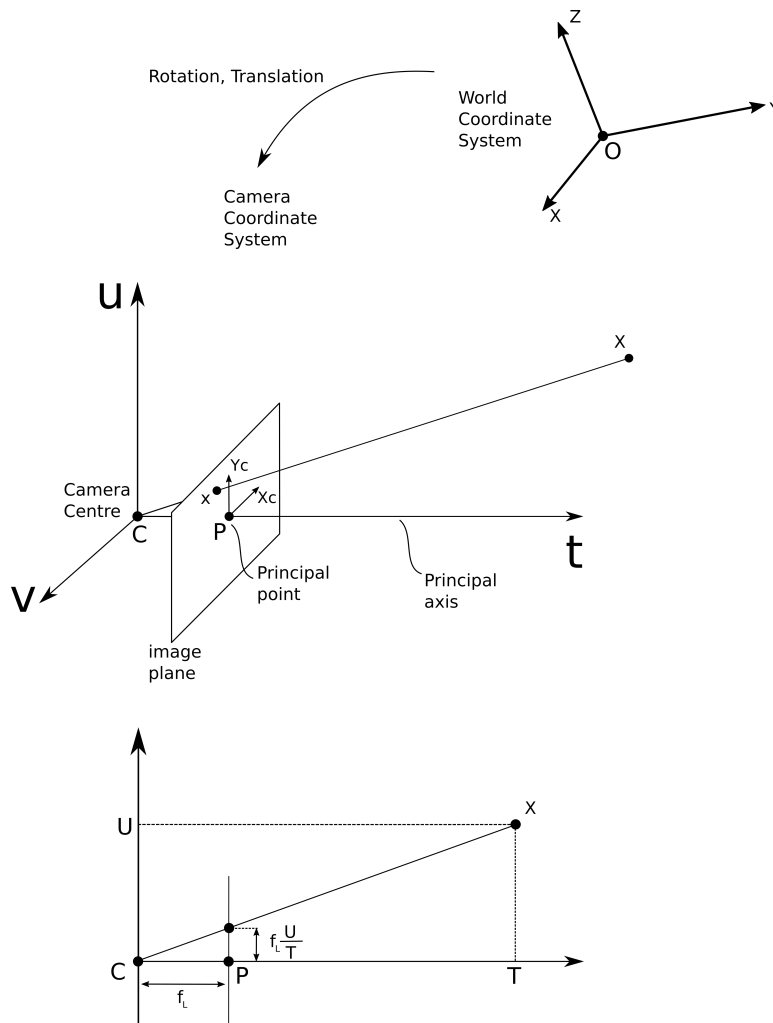


Figure 2.2 – Pinhole Camera Model

proposal stated in this thesis work is to exploit the insect eye imaging to the endoscopy domain. For this purpose, in this work, I focus on developing set of methods for creating miniature multi-camera system, which also have 3D image reconstruction capability. For endoscopic applications the illumination capability is an obvious requirement. Beyond the limitations of previous insect eye inspired systems, there is no illumination method proposed before for insect eye inspired imaging systems to work in dark environment. Therefore, this requirement is also taken into account during the research and development phases of this thesis work.

2.4 Preliminaries of Multi-camera image formation and processing

2.4.1 Pinhole camera model and camera calibration

The pinhole camera model [63] is a fundamental model for characterizing the single view point camera and it is used to map 3D world points onto an image plane. The illustration modified from [63] is shown in Fig.2.2. In pinhole camera model, basically the central projection of points in 3D world onto a plane -called image plane- is performed. The center of projection is the origin of a Euclidean coordinate system represented with $\{u, v, t\}$, the plane $t = f$ is called the image plane and the point $X = (V, U, T)^T$ is mapped to the point on the image plane. The light ray along the point X through the center of projection is crossing the image plane at point $x = (x_c, y_c)$. By the similar triangle relation, the point $X = (V, U, T)^T$ in the space is mapped to the point $(fV/T, fU/T)^T$. Here the point C is the camera center, P is the principal point and axis t is named as principal axis of the camera. Also, the camera center is defined in the world coordinate system and can be obtained by a 3×3 rotation matrix R and a 3×1 translation vector τ . Then the mapping of the point X into the camera image plane as x can be defined as given in (2.1) or in open form (2.2) with homogeneous representation.

$$x = K [R|\tau] X \quad (2.1)$$

$$\begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} V \\ U \\ T \\ 1 \end{bmatrix} \quad (2.2)$$

In equation (2.1), the 3×3 matrix K is called the camera matrix and as shown in equation (2.2), contains the information of the focal length of the camera (f_x, f_y) , and the principle point offset (p_x, p_y) . These parameters are known as intrinsic parameters of the camera. The 3×3 matrix R is the rotation matrix and with the 3×1 translation vector τ , represents the relative position of the camera center in the world coordinate system.

After obtaining the (x_c, y_c) points on the camera image frame, there is also a need for a distortion correction, which is added by the imperfect nature of the optical lenses used in the camera systems. Especially for cheap optic lenses and short focal distances, which are supposed to be used in the low cost miniature systems, the effect known as radial distortion becomes an important source of distortion [63].

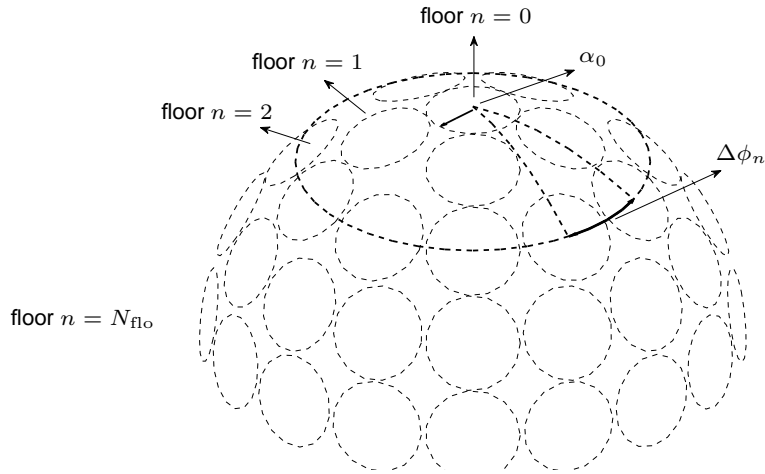


Figure 2.3 – An example arrangement of circular faces as individual cameras on a hemisphere.

2.4.2 Panoptic camera

The *Panoptic camera* described in [64], is a generalized solution for generating $360^\circ \times 90^\circ$ FOV panoramic images as well as providing infrastructure to solve different imaging problems such as omnidirectional 3D visualization and high quality surveillance. The system is a multi-camera spherical array of individual cameras as illustrated in Fig. 2.4a. Each individual camera is assumed to be a combination of relatively low resolution image sensor and a single lens system in front of the image sensor, which can be easily found in the market. Methods for generating a hemispherically arranged multi-camera system by using off-the-shelf camera units and components and omnidirectional vision construction by using real-time image processing techniques are described previously in [65], [66] and [67]. In this section, a brief description of the common steps and approaches used in previous methods have been explained.

Camera placement In [64, 65, 68], the insect eye mimicking approach, named as *Panoptic* is taken as constructing a hemispherical multi-camera system with many lenses and sensors placed on a hemispherical frame similar to the insects' natural many-lens-sensor designs. The placement of the cameras are done layer by layer granularity as illustrated in Fig.2.3. After having a hemispherical arrangement of cameras around a hemispherical frame, the coverage analysis of the *Panoptic camera* is done. In this way, the minimum distance from the camera center where any point in the space is observable by at least one of the cameras is determined. This is achieved by making a Voronoi tessellation [69] by taking the camera center positions as cell centers and assigning the maximum distance inside a Voronoi cell as the minimum required angle of view of each camera. This implies that, if the cameras have less angle of view than required, then there is a need for more camera positions to cover the required distance. This can be done by adding new layers of positions to the camera.

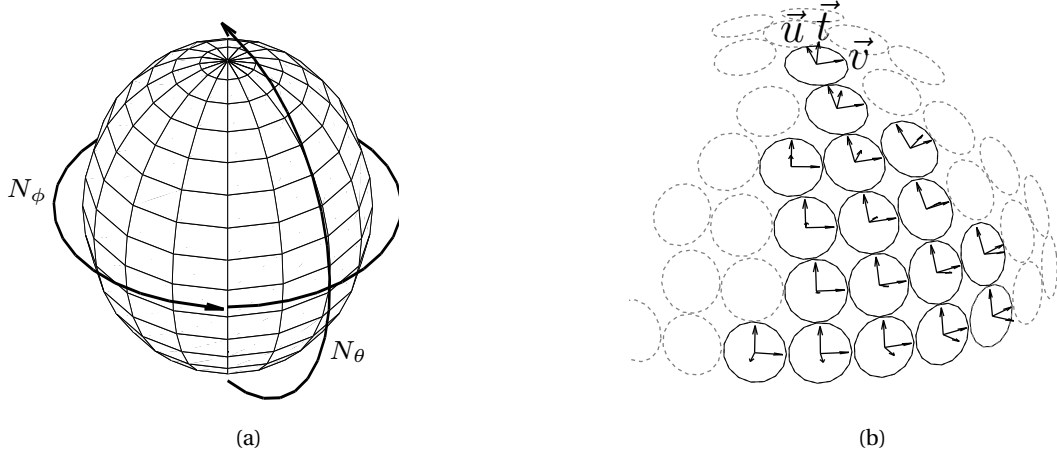


Figure 2.4 – (a) Example pixelization of the spherical surface, (b) 3 vectors \vec{t} , \vec{u} , \vec{v} representing camera orientation and positions

Panoramic image reconstruction The panorama generation techniques proposed in [64, 65] and in [66] are based on the geometrical relations between the cameras and pinhole camera model [63]. The methods are composed of two algorithmic phases. First the $360^\circ \times 90^\circ$ FOV of the camera system is divided into pixels as illustrated in Fig.2.4a. Then, for each pixel on the hemispherical surface, which is assumed as far away from the camera center, a unit vector $\vec{\omega}$ is assigned to describe its position in space. Then, the cameras observing this location in space are determined by using (2.3), by comparing the angle between the direction of interest and the camera vector \vec{t} and the angle of view (α) of the corresponding camera. Here, \vec{t} is the unit vector describing the direction of the observation of the camera, orthogonal to the camera image frame and it resides on the optical axis of the camera. The other two vectors, describing the camera orientation are also illustrated in Fig.2.4b, up and side directions of the each camera image frame with respect to the hemisphere center. After the candidate cameras are determined, the exact camera pixels contributing to this panorama direction are calculated by using the camera orientation vectors and pinhole camera model according to (2.4). In (2.4), the unit vector $\vec{\omega}$ is projected onto the camera coordinate system which is composed of unit vectors \vec{t} , \vec{u} and \vec{v} . And then by using the pinhole equations given in previous section the exact pixel positions are determined. Here the assumption is that the objects are sufficiently far away from the camera center when compared to the radius of the hemisphere, where the cameras are positioned. Therefore, the $\vec{\omega}$, \vec{t} , \vec{u} and \vec{v} vectors are all supposed to be unit vectors.

$$\vec{\omega} \cdot \vec{t} \geq \cos\left(\frac{\alpha}{2}\right) \quad (2.3)$$

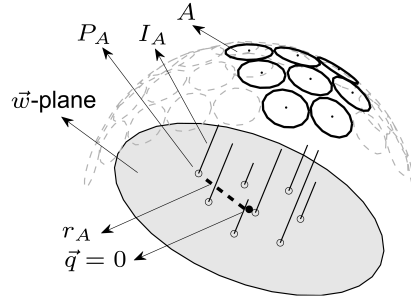


Figure 2.5 – Projections of camera centers contributing in direction $\vec{\omega}$ onto planar surface normal to $\vec{\omega}$

$$(X_u, X_v) = -(\vec{\omega} \cdot \vec{u} \times \frac{f_u}{\omega_t}, \vec{\omega} \cdot \vec{v} \times \frac{f_v}{\omega_t}) \quad (2.4)$$

Then on the second algorithmic step, candidate pixels are either selected or combined with a blending approach in order to have a seamless high quality panoramic image. The last algorithmic step is implemented by determining the best seeing camera for a particular $\vec{\omega}$ direction at the previous algorithmic step. Following that step, the pixel intensity of the corresponding pixel on the chosen camera's image plane is assigned as the panorama pixel. To improve the quality, a linear blending scheme is proposed [70] where the intensity value for each pixel in the panorama is calculated by a linear combination of the candidate pixels. The coefficients in this linear interpolation are determined as the distance values of the contributing camera focal point projections onto an orthographic $\vec{\omega}$ -plane. This representation is illustrated in Fig.2.5 where the outer circles on the hemispherical structure represent each individual image sensor and $\vec{q} = 0$ is the virtual observing point that is assumed to be anywhere inside the hemispherical structure of the multi-camera array. The gray shaded region is the orthogonal plane of a particular $\vec{\omega}$ vector. The $\vec{\omega}$ -plane model is based on the constant light flux (CLF) assumption where it is assumed that the light intensity remains constant on the trajectory of any light ray [64]. For the case in Fig.2.5, the example $\vec{\omega}$ direction is in the FOV of 8 individual image sensors. The small hollow dots labeled by P_i are the projected focal points of the individual cameras seeing the light ray at the direction. The $|r_i|$ values are the relative distances of each projection point to the observer point \vec{q} on the orthographic plane, which is expressed in (2.5). The nearest neighbor, linear blending and Gaussian weighted blending for the second algorithmic phase are then given by (2.5), (2.8) and (2.9) respectively.

$$\vec{r}_i = (\vec{q} - \vec{t}_i) - ((\vec{q} - \vec{t}_i) \cdot \vec{\omega}) \times \vec{\omega} \quad (2.5)$$

$$j = \underset{i \in I}{\operatorname{argmin}}(|r_i|) \quad (2.6)$$

$$L(\vec{q}, \vec{\omega}) = L(\vec{c}_j, \vec{\omega}) \quad (2.7)$$

$$L(\vec{q}, \vec{\omega}) = \frac{\sum_{i \in I} \frac{1}{|r_i|} L(\vec{c}_i, \vec{\omega})}{\sum_{i \in I} \frac{1}{|r_i|}} \quad (2.8)$$

$$L(\vec{q}, \vec{\omega}) = \frac{\sum_{i \in I} \frac{1}{|r_i|} e^{-\frac{d_j^2}{2\sigma_d^2}} L(\vec{c}_i, \vec{\omega})}{\sum_{i \in I} \frac{1}{|r_i|} e^{-\frac{d_j^2}{2\sigma_d^2}}} \quad (2.9)$$

In (2.5), \vec{t} is the vector directed to the direction of observation of each individual camera. In (2.5), (2.8) and (2.9), the $L(\vec{q}, \vec{\omega})$ represents the light intensity of the reconstructed panorama pixel for the direction $\vec{\omega}$, observed from the point represented by \vec{q} . $L(\vec{c}_i, \vec{\omega})$ is the pixel intensity value that is extracted at the first algorithmic step from the i^{th} sensor for the direction $\vec{\omega}$ where c_i is the vector pointing to the projected center of i^{th} sensor. I is the set of cameras which see the direction $\vec{\omega}$. In (2.9), d_j is the distance of the candidate pixel j from the center of i^{th} camera on the respective camera's image plane and σ_d is the variance of the Gaussian distribution function. Thus, providing a confidence level according to the candidate camera pixels' distance from the image center on the corresponding single camera image frame. The hardware implementations for these methods are represented in [64] for the nearest neighbor, [70] for linear interpolation and [66] for Gaussian interpolation. In [67], the implementation of nearest neighbor and linear interpolation is represented on a distributed network of cameras.

For all the multi-camera systems proposed in [64, 66–68, 70], the camera placement methods are not optimized for the miniaturized camera dimensions. The physical shapes of the cameras are not considered. The hardware implementations for real-time panoramic video generation are not optimized for single chip ASIC design. However, the algorithms and FPGA hardware implementations are suitable for optimizing for a miniaturized system. On the other hand, there are no attempts made to realize such system and optimizations for miniaturization.

2.5 Thesis Goals

As seen in the state of the art methods and devices, the insect eye inspired wide FOV imaging systems either suffer from the resolution or occupy large space in order to provide more resolution. The limitations for the resolution are due to the diffraction limits and process technology capabilities of the micro-machining techniques. On the other hand, for off-the-shelf component integration methods mixed with complex image processing techniques, the components sizes and arrangement are not optimized for miniaturization.

In terms of the application point of view, the first target of the current thesis is to apply the final solution to the endoscopy domain. Therefore, the acceptable sizes at this domain have been examined in detail. The first example domain is the capsule type endoscopy, which is a recent development to investigate the whole gastrointestinal tract (GI) from mouth to colons and rectum. The acceptable diameter sizes are 10-12 mm range. The colonoscopy which is a procedure to examine the human rectum and colons has been analyzed and it was observed that the colonoscopy device sizes range from 10 mm to 15 mm in diameter. For gastroscopy, which is used to investigate the upper GI tract, the sizes are ranging from 8 mm to 13 mm. The other possible applications areas might be smart phone industry and robotics/drone applications. There is a wide variety of size constraints but it can be said that 8 mm to 10 mm diameter cameras can be acceptable. As a result, the final target for the device dimension is bounded with these constraints as 8 mm to 10 mm. Regarding to the information specified here, a panoramic imaging system at 10 mm diameter size has been determined as the first goal of the study.

One important target of this thesis is to explore the implementation limits of miniaturized insect eye inspired, multi-aperture imaging systems in terms of size and resolution capability. The main aspect of the techniques in this thesis work is to use off-the-shelf components like pinhole cameras, up to date hardware platforms like FPGAs without going into complex micro-machining techniques.

The other major target is, by using real-time image processing and digital circuit implementation techniques, to get high quality compound panoramic video system that can be miniaturized to an ASIC design with minimum external components. The starting point of this target is a bunch of methods from the literature as described in the previous section, for generating panoramic images from hemispherically arranged multi-camera system which is called as *Panoptic*. To improve quality and performance, the panorama generation problem is revised as a neural superposition type compound eye of insects and it is treated as an interface problem. A method is proposed to get better quality panoramic images and videos.

With these two points, possible target applications of such a system are smartphone industry, flying drones or other robotic applications, and endoscopy applications. For the last application domain, i.e. endoscopy, there is also a specific target of having a miniaturized large angle of view multi-camera system that also has illumination capability. I proposed a distributed illumination method for the designed system.

Another target of this thesis is to create novel approaches for image processing that take advantage of multi-view point camera system such as improving the visual quality of the final panoramic image or detecting object boundaries.

3 Opto-Mechanical Aspects for Insect Eye Model

In this chapter, mapping of the insect eye capabilities to the targeted multi-camera system is described. For mechanical construction, the opto-mechanical constraints and limiting factors of the targeted system are addressed. Selection criteria for the individual camera optics and electronics are defined. Methods for placing the individual cameras on the hemispherical arrangement are proposed. The final opto-mechanical specifications of the constructed prototype are given. A calibration method for obtaining optical characteristics of the final compound-eye camera is disclosed, which will be used in the image processing steps as described in the next chapter.

3.1 Mapping of insect eye to camera type eye

In nature, the insects have the tiny sensor units called ommatidia [19]. The ommatidias are composed of an optical part that is the lens of the unit and a sensing element called rhabdom. The capabilities and the structure of such a system are analyzed and the mapping of such a distributed vision system to a human-eye type system is done in [19]. This mapping is illustrated in Fig.3.1 where R_i is the radius of curvature of the insect eye, f_h is the human eye type camera focal length, s is the human sensor's pitch, and the common parameter $\Delta\phi$ is the angle between the receptor units. Similar to the natural insect eyes, in the ongoing model, there are multiple cameras looking out from a hemispherical surface; however, in the end the final design will be a single eye or imaging system. In other words, the final *compound eye* will be characterized with its some basic properties like resolution and physical size [18].

The resolution or acuity is defined with $\frac{1}{2\Delta\phi}$ as the common property of both eye types [19]. The total number of pixels that an insect eye can provide is related to the number of ommatidia units for an insect eye, which is determined by $\Delta\phi$ and also total size of the eye. By combining the human vision system and insect eye model, our model is also based on the relation between number of ommatidias and the size of the eye. The resolution is also defined in terms of ommatidias we can reconstruct by image processing techniques and the sampled images

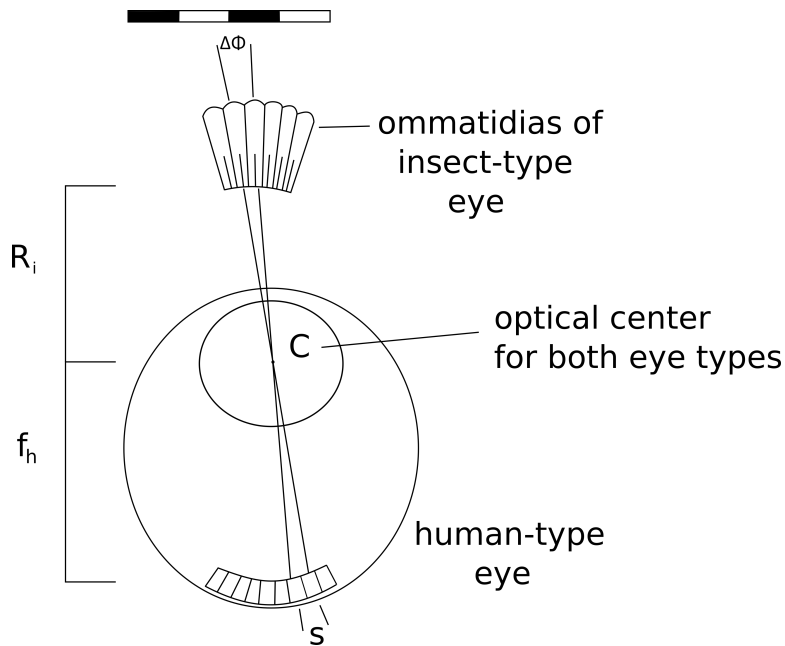


Figure 3.1 – The relation between insect and human eye resolving capability

from the multi-camera system I designed.

My approach is to utilize insect eye model and realize the same functionality by using human-eye type off-the-shelf camera systems modeled as pinhole camera [63]. Hence, the single cameras that will be used in the implementation have a great impact on the size and resolution capability of the final system. In the next section the criteria and the analysis for the decisions given in choosing a single camera are presented.

3.2 Effect of single camera dimensions

The dimensions and the volume of each individual camera unit play an important role for the miniaturized level multi-camera system. Therefore, the first criterion for choosing the individual cameras that will be used in the final construction is its volume and dimensions. In [64, 65], for implementation of the large scale devices, the single camera units are modeled as circular shapes that will be placed on a hemispherical surface. Thus, the third dimension of the cameras are omitted since the volume of the individual cameras V_{cam} is negligible when compared to the hemispherical structure volume, $V_{hemisphere}$.

For the miniaturized model, the volume and the shape of the cameras are determining factors. Hence, the selected individual cameras are getting more important. The camera chosen has a 1 mm x 1 mm image sensor combined with boroglass optics and a pinhole aperture. The minimality and the simplicity of the individual cameras are also essential in terms of having a minimal processing circuit. The electrical specifications of the cameras for embedded system

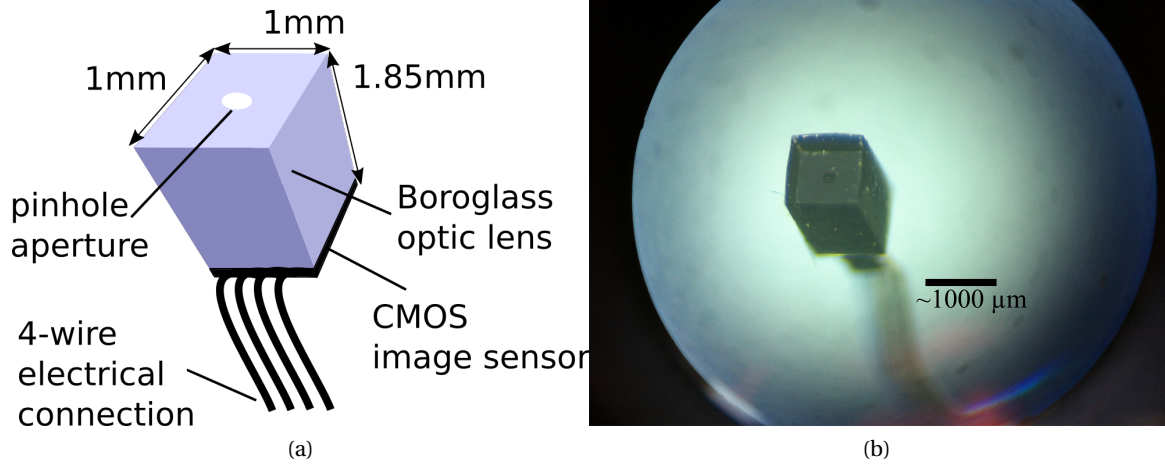


Figure 3.2 – The camera chosen for the implementation, (a) illustration for the physical dimensions of the single camera (b) a close photo view of the camera taken under a microscopic lens

and image processing circuit design are considered in Chapter-5.

The limits for the dimensions of the image sensor array are mainly dependent on the pitch size of an individual pixel. On the optics side, a single minimal lens design and an appropriate pinhole (aperture) for desired resolving capability at a desired distance are required. The optical and mechanical properties of the chosen individual camera is given in Table-3.1. The mechanical illustration of the individual camera unit is given in Fig.3.2a and a closeup photo of the camera can be seen in Fig.3.2b.

Table 3.1 – Single image sensor opto-mehcanical specifications

Specification	Value
Size	1mm × 1mm × 1.85mm
F# number	6.0
FOV (<i>vertical × horizontal</i>)	64° × 64°
Focal length	0.66mm
Aperture	0.11mm
Depth of Focus	3-50 mm
Pixel Array	250x250

The vision capability of the camera system is bounded by the vision capability of each individual camera unit. The chosen cameras are utilizing a pinhole aperture with boroglass optic lens, where the aperture diameter is $d_{pinhole} = 110 \mu m$ and a focal length of $f_{pinhole} = 660 \mu m$. For relatively far objects from the camera, according to Rayleigh criterion, the theoretical capability of separating two point source airy discs at image plane is given by: $x_i = 1,22 \times \lambda \times \frac{f_{pinhole}}{d_{pinhole}}$. For the visible spectrum, a minimum $\lambda_{min} = 420nm$ can be assumed. Hence an $x_i \approx 3 \mu m$ is obtained. The CMOS image sensor of the chosen camera is 250x250

photo-sensor array with a sensor pitch size $3\ \mu\text{m}$. Therefore, the optical capability is supported by the electronic part of each individual camera unit. Assuming that the two point sources are separated by an angle of θ_0 , this angle can be interpreted as $\theta_0 = \frac{x_i}{f_{pinhole}}$. As a result, the angular separation capability of the individual cameras is calculated as $\Delta\phi = \theta_0 = \frac{3\ \mu\text{m}}{660\ \mu\text{m}} \approx 0.26^\circ$ at a distance of 3-50 mm, which is given as the depth of field of the individual cameras I utilized for the system.

Since the resolving capability of the single camera will determine the limit of the resolving capability of the whole system, an initial comparison with the existing miniaturized camera systems and natural counter parts leads to the choice of the individual cameras in terms of resolution. For example the state of the art system implemented using micro-machining techniques and special materials in [36] have a resolving capability of $\Delta\phi = 11^\circ$. For the system in [37], the resolving capability is reported as $\Delta\phi = 4.7^\circ$ where for a natural system, which is known as a superior resolution insect eye, the dragonfly eye has a resolving capability of $\Delta\phi = 0.24^\circ$ [71]. Therefore, the resolving capability of the chosen system is reasonable when compared to the state of the art systems and natural systems. Of course, the size and the quality of the utilized optical design have an effect on the optimal focus distance of the camera. For the camera that is chosen for implementation, the depth of field is 3-50 mm, which is an acceptable distance for applications like endoscopy where the objects of interest are around 1-100 mm far.

As a conclusion, the minimal dimensions and reasonable resolving capability of the individual camera unit chosen for the system implementation are determined as appropriate and it is used in the rest of the design.

3.3 Analysis for camera placement

The first target for the model is to have many cameras placed on a curved surface and looking outside of the center of the curved surface. A hemispherical surface which is appropriate to the natural counterparts was chosen. The question that arises has been how the cameras will be placed optimally on the surface. In [64] and [65], there is a layer by layer placement of the cameras proposed, which is not optimized for miniaturization. With the decreased angle of view and desired minimum distance to be able to generate the virtual ommatidias, the number of cameras is increasing rapidly with the method in [64], and it generates unnecessary camera positions, which end up with bigger physical sizes for a given individual camera size and the angle of view. This becomes a bottleneck for the miniaturization step and not appropriate for densely packing of the cameras in to a limited volume.

To reach a solution for the miniaturized model, I first analyzed how the number of cameras are changing for a given desired imaging distance from the hemispherical compound camera center ($R_{virtual}$) and angle of view of the individual cameras (AOV). The desired distance ($R_{virtual}$) is said to be achieved or covered if all the cameras' angle of view are intersecting at a

3.4. Proposed camera placement for miniaturized camera model

spherical surface closer or equal distance to the desired final application distance.

A method is introduced in [64] for determining if the angle of views of the cameras are intersecting at a desired distance or not for hemispherically arranged multi-camera systems. In [64], the AOV of each camera is taken as a free variable and the camera positions are determined. Following the camera positions generation, the overlap distance is analyzed and the desired AOV for each camera is chosen to cover the desired distance by at least one camera [64].

In [64, 65], the cameras are placed without taking care about minimizing the number of cameras or overlap distance. Instead, the placement starts with an initial camera placed at the north-pole of the hemisphere and then the 90 degrees in the longitudinal direction is divided into an arbitrary number of floors, then these floors are populated with cameras in the latitudinal direction. The shape of each camera is considered as a circular surface shape on the hemisphere defined by its radius. Then before physical construction, the full hemisphere is scaled by equalizing the actual camera diameter and the diameter of the circular surface on the unit hemisphere in order to fit the number of camera positions on the hemispherical structure. So the number of cameras, physical dimensions of the cameras and the hemisphere diameter are not taken as constraints, which is not suitable for miniaturization.

The total number of cameras desired for different $R_{virtual}/R_{physical}$ for the method in [65] can be seen in Fig.3.3. The graphs in Fig.3.3 are generated by defining certain overlap distances $R_{virtual}/R_{physical}$ for each graph and sweeping the angle of views of the single cameras. In each sweep the overlap is checked by an analysis of the coverage as described in [65]. Here $R_{physical}$ is the actual radius of curvature of the eye and $R_{virtual}$ is the radial distance of overlap, at which all the points reside in the angle of view of at least one camera. As shown in Fig.3.3, the need for more cameras with decreasing angle of views of individual cameras is represented by the increase on the curve. It can be seen that there are certain jumps on the curves for the method in [65] when there is a need for more number of cameras to be able to generate virtual ommatidias without a gap at a certain distance from the camera. The reason for this is whenever there is a need for increase in the number of the cameras, a whole new layer of camera positions are added, which can vary a few to hundreds of positions. The camera positions are physically appear on the final mechanical structure, this results in unnecessary occupancy of camera positions and bigger physical housing for the compound eye.

3.4 Proposed camera placement for miniaturized camera model

There can be different approaches to place the cameras on a dome according to the application needs. One approach is to minimize the number of cameras while having an overlap at a certain distance determined by the application from the camera surfaces or hemispherical compound camera center. With this approach, total cost, power consumption and the total dimensions of the final hemispherical camera can be minimized. The other approach can

Chapter 3. Opto-Mechanical Aspects for Insect Eye Model

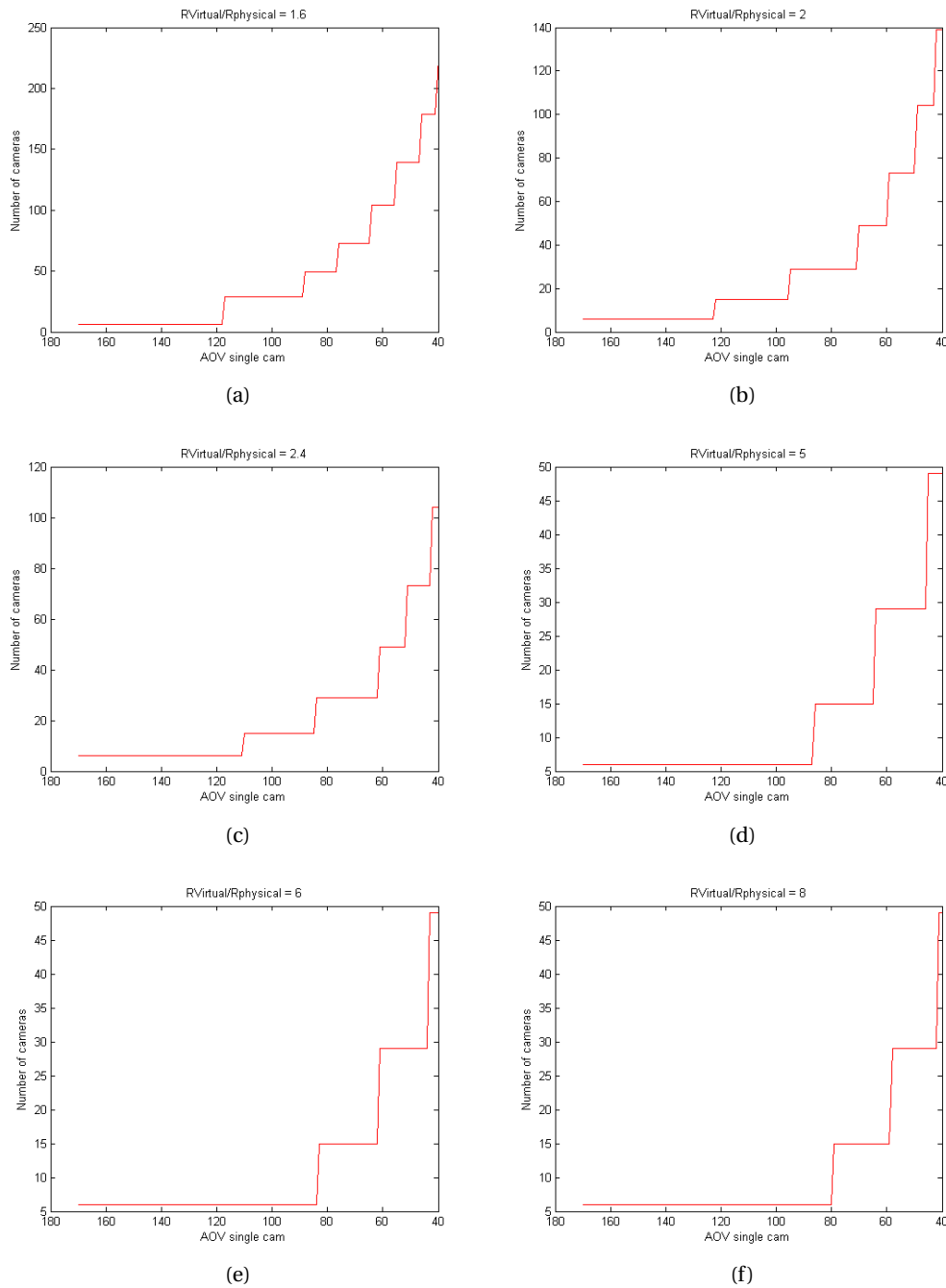


Figure 3.3 – Horizontal axis shows the angle of view of each individual camera, vertical axis shows the required number of cameras for having virtual ommatidias at distance $R_{virtual} = 8, 10, 12, 25, 30, 40$ mm with a $R_{physical}=5$ mm.

be maximizing the number of cameras that can fit into the given total dimensions of the hemispherical volume. In the later approach, there will be more overlap of the camera angle of views, hence closer observable distance and better 3D infrastructure.

3.4. Proposed camera placement for miniaturized camera model

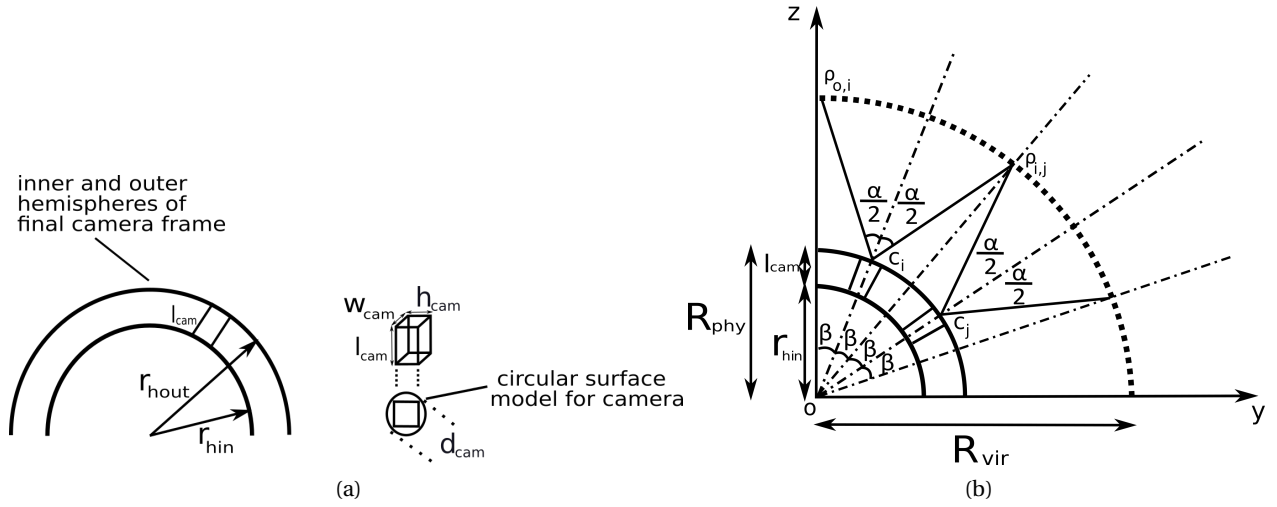


Figure 3.4 – New camera placement model (a) single camera circular surface model on dome y-z plane (b) Geometrical relations of the cameras in one quarter of the dome in y-z plane

To start the placement, the camera positions can be defined as circles on hemisphere, however hemispherical surface chosen will not be the final outer surface of the hemispherical structure used in this study. Instead, it will be an inner surface with a radius $r_{hin} = r_{hout} - l_{cam}$. Here the l_{cam} is the length of the camera from its optical pinhole aperture to the end of the CMOS image sensor which is the omitted dimension in the previous work [64, 65]. This new model is illustrated in Fig.3.4a.

Then, the cameras are placed on a quarter of the inner hemisphere without having interference with the circular surfaces of the cameras. . The first placement will be in one dimension to cover 90 degrees on one quarter of the semi-circle on the y-z plane as shown in Fig.3.4b. Here, there might be two choices as mentioned before: minimize the number of cameras or minimize the distance of overlap of the camera AOVs by maximizing the number of cameras.

3.4.1 Method for minimizing the number of cameras for a certain overlap distance

The first method is to minimize the number of cameras for a certain overlap distance in order to have a minimum number of components and to reduce cost. To populate the hemisphere, the cameras are placed in the latitudinal directions starting from the initial camera positions on one quarter of the y-z plane. Then on the x-y plane, placement of camera positions done on virtual latitudinal circles.

In order to minimize the number of cameras, the angle between each camera has to be maximized which is given as 2β in Fig.3.4b. Concurrently, the whole 90 degrees FOV between the y-z axes should be covered at least one of the camera's angle of view, α , at least at a very far

Chapter 3. Opto-Mechanical Aspects for Insect Eye Model

point, let's say infinity. For farther R_{vir} distances, less number of cameras will be needed. This overlap distance R_{vir} can be chosen as another input to determine the specific application needs.

The relation between β and α can be obtained by the sine law on the $\widehat{oc_i\rho_{i,j}}$ triangle. This relation is given by (3.1) and (3.2). Here, the R_{vir} is the desired radial distance from the hemisphere center o to the point where adjacent camera angle of views intersect. R_{phy} is the physical constraint for the actual radius of the hemisphere, which is also given as the $r_{hout} = r_{hin} + l_{cam}$ as shown in Fig. 3.4a.

$$\frac{\sin(\pi - \alpha/2)}{R_{vir}} = \frac{\sin(\alpha/2 - \beta)}{R_{phy}} \quad (3.1)$$

$$\beta = \alpha/2 - \sin^{-1} \left[\frac{R_{phy}}{R_{vir}} \sin \frac{\alpha}{2} \right] \quad (3.2)$$

Then the desired minimum number of cameras to cover the 90 degree FOV can be given by (3.3). Since this equation is not necessarily results in an integer value, the closest bigger integer value is taken as the number of cameras needed. Then, to find the actual value for β_{fine} , (3.5) is utilized by using the integer value of N_{camvup} .

$$N_{camv} = \frac{\pi/2}{2\beta} \quad (3.3)$$

$$N_{camvup} = \left\lceil \frac{\pi}{2\alpha - 4 \sin^{-1} \left[\frac{R_{phy}}{R_{vir}} \sin \frac{\alpha}{2} \right]} \right\rceil \quad (3.4)$$

$$\beta_{fine} = \frac{\pi/2}{2N_{camvup}} \quad (3.5)$$

Later, this approach can be extended by defining virtual longitudinal circular layers as illustrated in Fig.3.5 passing through the camera locations on the quarter hemisphere. Here, the method for determining the number of cameras is defined at each longitudinal virtual circle (layer). To achieve this, it starts with one camera at the position determined by the quarter circle as described above. Then one camera has been added to the layer under consideration at a time and a coverage analysis of the full field of view is performed by utilizing the method first described in [65]. In this method the full field of view is first pixelized. Each pixel is treated as a direction starting from the hemisphere center and ending at unique point in space and it is represented by a vector which is defined with its spherical coordinates (θ, ϕ) . Then the same triangulation given in (3.1) is used. This time β angle will become the angle between the camera center and the vector direction. In this way, we have the result of either a direction in space which is *seen* by a camera or not. Here the angle of view of each camera

3.4. Proposed camera placement for miniaturized camera model

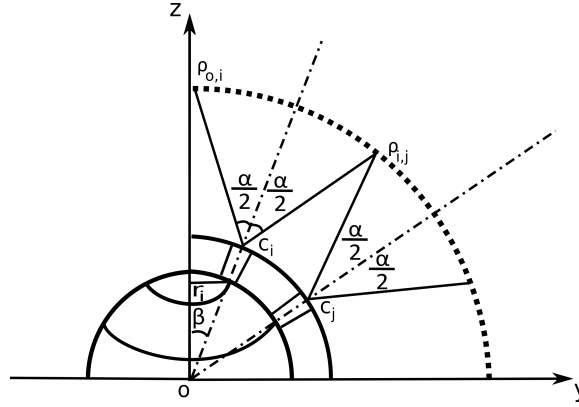


Figure 3.5 – Illustration for the layers for extending the camera placement around the hemisphere

is assumed to be isometric, meaning that the vertical and horizontal angle of view of each camera are equal. This assumption holds for the sensors chosen since they have a square image sensor plane. For the sensors with aspect ratios different from 1:1, the smallest angle of view should be chosen as α .

In our method, 1 sensor is added to a layer, an overlap analysis is made; and the smallest lateral angle θ_{uc_i} that is not seen by any of the current cameras placed at layer i is determined. The north-pole of the hemisphere is chosen as starting point, i.e. $\theta_{northpole} = 0$. The illustration for the coverage analysis and the θ_{uc_i} is shown in Fig.3.6 for an example placement where there are 3 cameras at the first layer. In order that the θ_{uc_i} of the current placement to be seen by at least one of the cameras at the next layer ($i + 1$), it should satisfy the condition given in (3.6).

$$(2(i + 1) + 1)\beta_{fine} - \theta_{uc_i} \leq \frac{\alpha}{2} - \sin^{-1} \left[\frac{R_{phy}}{R_{vir}} \sin \frac{\alpha}{2} \right], i \in [1, N] \quad (3.6)$$

Equation (3.6) implies that, in order the direction that has the latitudinal angle θ_{uc_i} to be covered by any camera at the layer ($i + 1$), the angle between this direction and the angle determined by the center of the next layer ($i + 1$), should be less than or equal to the angle of view of the camera on the sphere with radius R_{vir} . With this condition, the number of cameras that should be placed in each layer is determined. To do that, the method I proposed starts with 1 camera at each layer and continue adding the cameras until the condition in (3.6) has achieved. Then it continues with the next layer until the last layer is reached. For last layer a search for a condition is performed where all the directions on the $360^\circ \times 90^\circ$ FOV are seen by at least one of the cameras at the desired virtual radius R_{vir} .

In order to minimize the number of cameras and the overlapping FOVs, two refinement steps are also added to the method. First refinement is that, for each layer when one camera is added

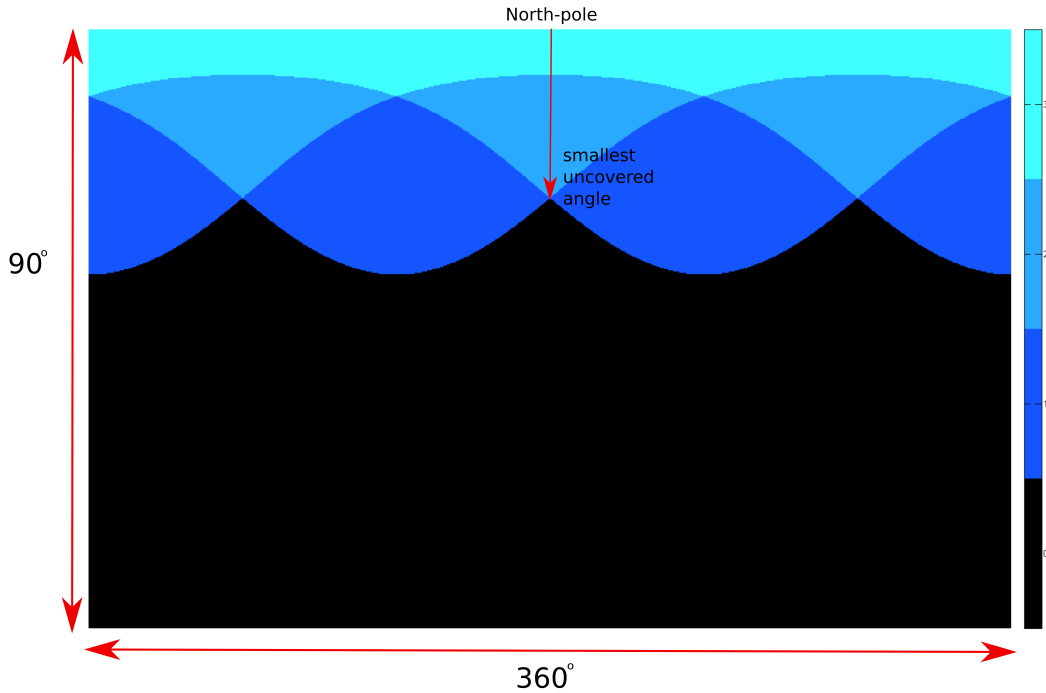


Figure 3.6 – Illustration for the overlap analysis for 3 cameras positioned at layer 0 with $\beta_0 = 15^\circ$. The smallest latitudinal angle from the north-pole that is not seen by any camera is $\theta_{uc_0} = 19.6^\circ$

in each step, a rotational search is made in order to discover if any combination of placement satisfies the condition in (3.6). If such a position is found, then increment operation for the number of cameras stops for the current layer. Then the placement of next layer starts after the second refinement step. In the second refinement step, the latitudinal position of each layer after placement is trimmed with fine angular steps to get as closer as possible to the equator of the hemisphere. To do this, the latitudinal angle of the layer is increased with 1 step and the condition in (3.6) checked. If the condition holds, it continues to increment until it comes to a position that violates (3.6). Then the final positions of the cameras are determined. The algorithm continues to add cameras until a full coverage of the FOV is achieved.

In Fig.3.7, an example analysis is shown for the proposed method in comparison with the method from [65]. This analysis is for a fixed angle of view (64°) and the number of cameras needed to cover different R_{vir}/R_{phy} is analyzed. Here it is observed that the method I proposed has a smooth increase in the number of cameras when there is a need for design a camera dome for imaging proximity objects. However the method in [65], ends up with a sharp increases at certain points. The reason for this is that in my approach a camera by camera increase is performed where the method in [65] accommodates a layer by layer increase. In Fig.3.7, when the interval of $7.2 > R_{vir}/R_{phy} > 2.4$ is observed, it is seen that my method always gives a smaller number of cameras and has smooth increase. This means for a 5 mm

3.4. Proposed camera placement for miniaturized camera model

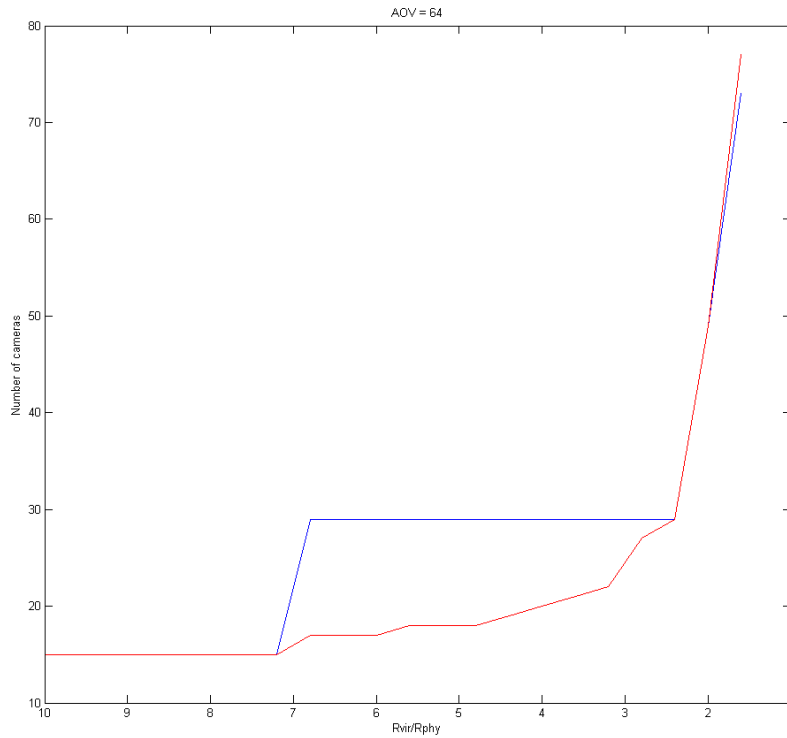


Figure 3.7 – Analysis of the growth of the camera positions for single camera $AOV=64^\circ$, starting in a overlap range $R_{vir}/R_{phy} = 50/5 : 8/5$. The proposed method is plotted in red color and the method in [65] in blue color.

dome, if it is required to design a camera system for imaging at ranges from 12 mm to 50 mm, the method ends up with fewer or same number of cameras in worst case. For closer distances, the two approaches converges, since very large number of cameras and layers are needed. Hence, the number of cameras and number of layers are converging.

When the constraints of 5 mm radius and image distances 15-50 mm are examined, the current design gives fewer number of cameras, which will end up with achievable final physical design. For the method in [65], to have a $R_{vir}/R_{phy} = 15/5$, it is ended up with 29 cameras where in this approach, 22 cameras are needed. This has an effect on the diameter of the final design. Which I analyzed in the next section and propose a method for maximizing the number of cameras in a limited volume.

3.4.2 Method for maximizing the number of cameras in a limited volume

The distance desired for observing $360^\circ \times 90^\circ$ field of view without any gap is determined by choosing the application specifications. As an example application, the endoscopic imaging, the colonoscopy case is chosen. The human colon is about 40-60mm in diameter. Then the

Chapter 3. Opto-Mechanical Aspects for Insect Eye Model

acceptable object distances will be at 20-50mm range from the hemispherical camera center. And as defined earlier, the acceptable image device diameter will be at 10mm range.

In order to maximize the number of cameras the hemisphere inner radius r_{hin} and the diameter of the cameras' base area of the circumscribed circle shown in Fig.3.4a, given by (3.7) should be taken as the constraints. Then the number of cameras in one quarter and in one dimension on the y-z plane is obtained by dividing the arc length to the d_{cam} as given by (3.8). Since this division gives the maximum number of cameras that can be fit on this arc and not necessarily an integer number, the result is floored to the next smaller integer.

$$d_{cam} = \sqrt{w_{cam}^2 + h_{cam}^2} \quad (3.7)$$

$$N_{camv} = \left\lfloor \frac{\frac{\pi}{2} r_{hin}}{d_{cam}} \right\rfloor \quad (3.8)$$

$$\beta = \frac{\pi/2}{2N_{camv}} \quad (3.9)$$

Then, this approach can be extended by defining longitudinal circular layers as illustrated in Fig.3.5 passing through the camera locations on the quarter hemisphere. Here, the radius of each circle is given by (3.10). Thereupon, the number of cameras are obtained at each layer i by dividing the perimeter of the circle at i by r_i as given by (3.11).

$$r_i = r_{hin} \sin[(2i - 1)\beta], \quad i \in [1, N_{camv}] \quad (3.10)$$

$$N_{camhi} = \frac{2\pi r_{hin} \sin[(2i - 1)\beta]}{d_{cam}} \quad (3.11)$$

In this way, the maximum number of cameras in horizontal (N_{camh}) and vertical (N_{camv}) directions are determined. For the cameras to be used, when the calculations are made, the result becomes $N_{camv} = 3$ and $N_{camh1} = 3$, $N_{camh2} = 9$ and $N_{camh3} = 12$ and 24 cameras in total. The illustration of the positions of the cameras is shown in Fig.3.8a. And the actual model generated in Solidworks software is shown in Fig.3.8b, 3.8c, and 3.8d.

By making the overlap analysis with the camera positions determined in this step, $R_{vir} = 18$ mm is obtained. This means the final design can observe all the angles at 18 mm distance far from the hemisphere center and virtual ommatidias can be generated without any empty space between them as panorama pixels starting from this distance.

3.4. Proposed camera placement for miniaturized camera model

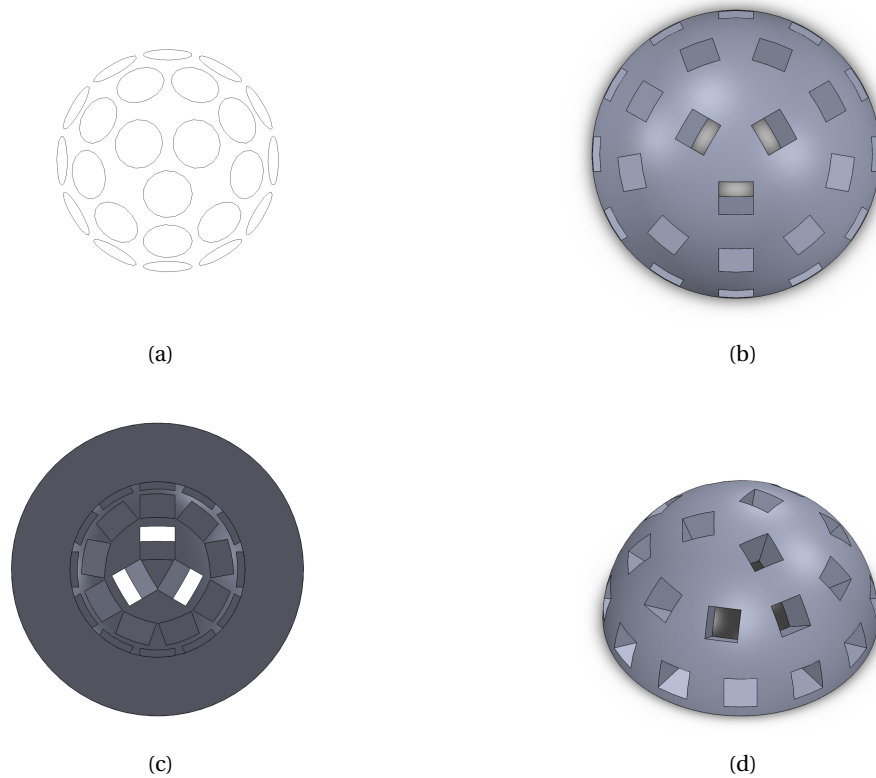


Figure 3.8 – The camera positions on the final design for prototyping, (a) 24 camera positions, (b),(c) and (d) the mechanical model drawing from different viewing angles

3.4.3 Generalized solution by using uniform distribution

A general solution for placing the cameras on a spherical surface can be obtained by approximation to uniform central Voronoi tessellation (CVT) of the sphere [72, 73] as shown in Fig.3.10. For this approach, again, the initial inputs should be the angle of view of each camera and the desired distance that should be covered completely by the intersecting angle of views of adjacent cameras. Then the placement starts with 1 camera and a central Voronoi tessellation is performed on the sphere with a limited number of iterations. The coverage analysis of the desired distance is made with the given angle of the views of the cameras. If it is not covered, 1 more camera is added and the tessellation restarts. In this way the number of cameras increase gradually and reaches to an optimum number of camera positions. This smooth increase can be seen on the graphs in Fig.3.9 as well. Along these lines, more uniform distribution of the cameras can be obtained, which can be an alternative method for miniaturized multi-camera systems for limited physical space. The simplified procedure is given in Algorithm 1.

However, since the tessellation gives a uniform distribution for sphere, and the target has been a hemispherical arrangement, there is a problem of choosing a proper point on the sphere as our $xyz_{hemisphere} = [0, 0, 1]$, the north pole point for the hemisphere from the full sphere. If a

Chapter 3. Opto-Mechanical Aspects for Insect Eye Model

Algorithm 1 Camera placement algorithm with CVT

```

numcam = Nc
numCVTiteration = Ni
while the full sphere is not covered at Rvirtual do
    make CVT(numcam, numCVTiteration)
    if coverage achieved at Rvirtual then
        break;
    else
        numcam = numcam + 1
    end if
end while

```

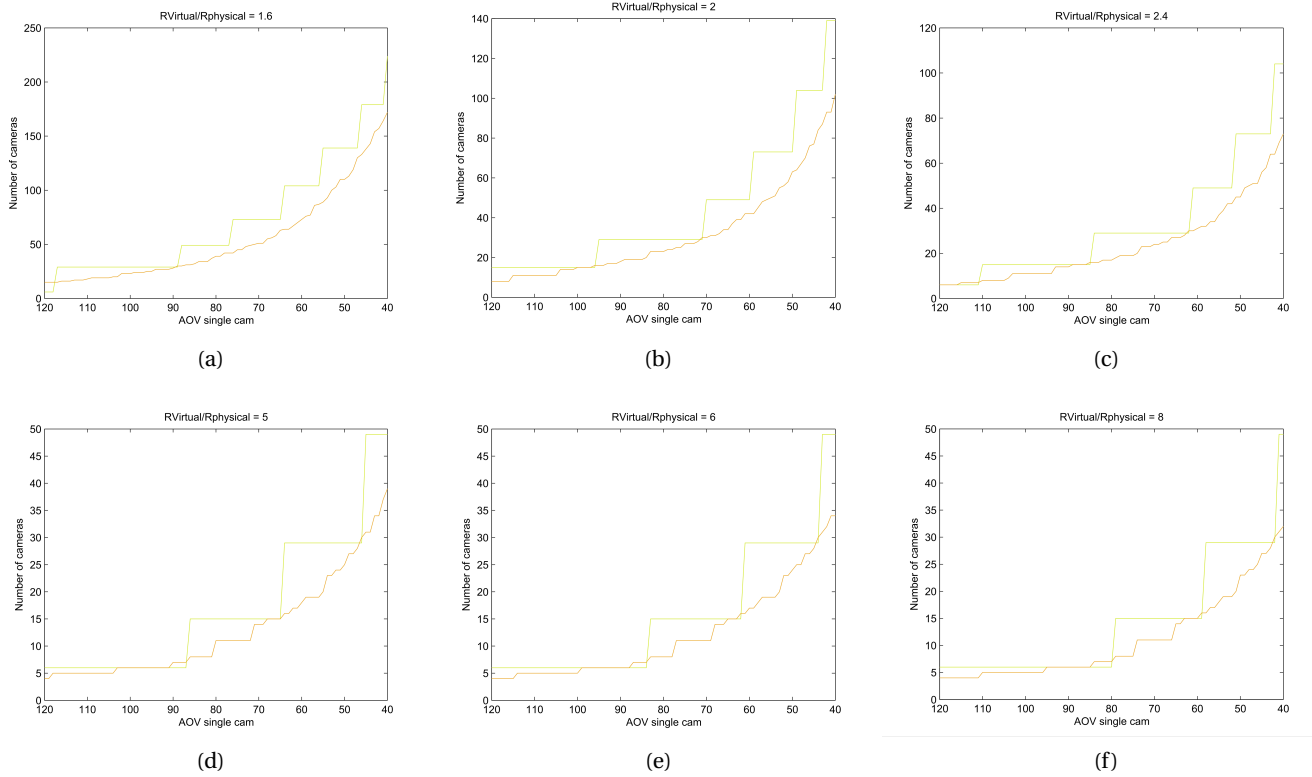


Figure 3.9 – The comparison of the growth of the number of cameras for previous method [64] and the CVT based method proposed. Horizontal axis shows the angle of view of each individual camera, vertical axis shows the required number of cameras for having virtual ommatidias at distance $R_{virtual} = 8, 10, 12, 25, 30, 40$ mm with a $R_{physical} = 5$ mm.

point as the north pole of the hemisphere is chosen such that it is the intersection point of Voronoi regions of 3 cameras to cover the north pole direction, the arrangement of the cameras on one half of the sphere approximates to the method I proposed for having maximum number of cameras, with original CVT, with slight differences in the camera positions. This matching is illustrated in the Fig.3.10. For the $R_{vir} = 18$ mm and $R_{vir} = 5$ mm, CVT based method

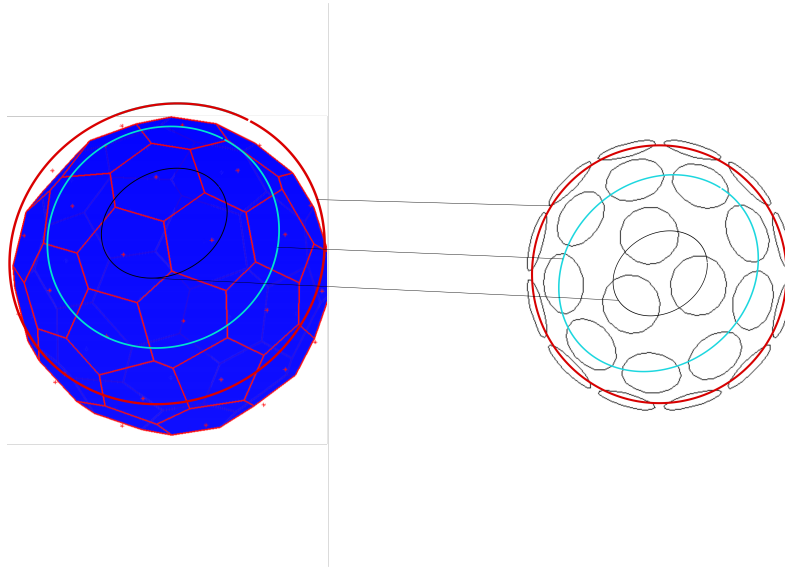


Figure 3.10 – An example for the camera positions obtained from CVT based method for $R_{vir} = 18 \text{ mm}$ and $R_{vir} = 5 \text{ mm}$, and matching the approximate positions with our final design.

results in 48 camera positions for the full sphere, half of which is matching with the number of cameras in our final design as well.

3.5 Proposal of illuminating compound eye

No matter which of the methods in [36], [37], [65] are taken into account, there is no illumination method which is proposed before for compound eye systems to be able to operate at dark conditions especially for close distance imaging such as endoscopic imaging. This feature, which is unique to our compound eye model, does not exist even in the natural compound eyes [18].

Fiber optic illumination channels reaching from a light source 2m away to the hemispherical camera tip can be attached to the fiber channels on the hemisphere, which give an opportunity to illuminate the targets at a proximity to the hemisphere. This feature can be utilized for the applications like endoscopy or any other dark environment applications for robotics. In the current prototype there are 112 illumination channels distributed around the cameras to the empty spaces on the hemispherical frame. The theoretical model for illumination capability can be given by (3.12) [18]

$$F = \frac{L \times A_e \times A_r}{d^2} \text{ lumen} \quad (3.12)$$

The area for us will be the surrounding surface of the bowels. d will be approximately 20mm



Figure 3.11 – The final mechanical design for the prototype, (a) and (b) Solidworks Drawings ,(c) and (d) the fabricated and assembled prototype

from the surface of our imaging system and 25mm from its center. If we assume a 25mm radius hemispherical area around our imaging system, the total area will be $A_r = 2 \times \pi \times 625mm^2$, which is $1250mm^2$.

L is the luminous flux of the light source per unit area, A_e is emitter area, the total area of the emitter. We use a commercial light source with 6000K white light emitting diode (LED) illumination which is capable of delivering 700 lumen at a diameter of 13.5mm, which is equivalent to $A_e = 143.1 mm^2$. The actual emitter area is the total area of the fiber opening

3.6. Calibration and Software Based Stitching Analysis

of the illumination system. There are 108 channels with 250 μm diameter and 4 channels at the top with 500 μm diameter. So the total area of our emitter channels is $A_c = 6.1 \text{ mm}^2$. Then the actual L for us will be proportional to the area ratio of the light source and our total channel area multiplied by the Luminous flux of the light source: $\frac{A_c}{A_e} \times 700 = 29.8 \text{ lumen}$. By substituting $L \times A_e$ with this number in (3.12), we end up with the light delivering capability of our system at 20 mm from the surface of the imaging system on a hemispherical area $F = 93.2 \text{ lumen}$.

After the illumination channels are included, the final model is generated in Solidworks shown in Fig.3.11a and Fig.3.11a. Then the mechanical model is realized by using 5 axis computer numerical control (CNC) machining technique using Polyoxymethylene (POM) plastic as the substrate material of the hemisphere housing. For the initial prototype, which is used for the visual experiments, the assembly of the cameras and fiber-optic cables on the hemispherical housing are done by hand work. The final assembly of the cameras and 30 fiber-optic channels on the top part of the dome are shown in Fig.3.11c and 3.11d.

3.6 Calibration and Software Based Stitching Analysis

For the calibration process, a SIFT [43] and bundle adjustment based commercial software is used. To do this, I first take shots inside a 35 mm diameter cylindrical closed tube with texture. The tube is shown in Fig.3.12a and an example shot is shown in Fig.3.12b, then I make a construction in the software environment to determine the extrinsic parameters which are defined with yaw (α), pitch(β), roll(γ) and intrinsic parameters, focal length (f_L), principle point(cc), k parameters (k_1, k_2, k_3) for lens distortion correction. From yaw, pitch, roll rotations, by using (3.13) I define a camera coordinate system with 3 unit vectors, ($\vec{t}, \vec{u}, \vec{v}$). I calculate these camera rotation vectors at calibration step because the cameras in the system are fixed at their location with respect to the center of the hemispherical surface.

$$R_x(\gamma) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{pmatrix} \quad (3.13a)$$

$$R_y(\beta) = \begin{pmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{pmatrix} \quad (3.13b)$$

$$R_z(\alpha) = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.13c)$$

$$R_t = R_x(\gamma)R_y(\beta)R_z(\alpha) \quad (3.13d)$$

$$= \begin{pmatrix} \cos \beta \cos \alpha & -\sin \alpha \cos \beta & \sin \beta \\ \cos \alpha \sin \beta \sin \gamma + \cos \gamma \sin \alpha & \cos \gamma \cos \alpha - \sin \gamma \sin \alpha \sin \beta & -\cos \beta \sin \alpha \\ \sin \alpha \sin \gamma - \cos \alpha \cos \gamma \sin \beta & \cos \gamma \sin \alpha \sin \beta + \sin \gamma \cos \alpha & \cos \beta \cos \gamma \end{pmatrix} \quad (3.13e)$$

$$\begin{pmatrix} \vec{i} \\ \vec{u} \\ \vec{v} \end{pmatrix} = R_t \begin{pmatrix} \vec{x} \\ \vec{y} \\ \vec{z} \end{pmatrix} \quad (3.13f)$$

$$(3.13g)$$

For image acquisition from the single cameras, I used the single camera interface and FPGA system described later in the Section-5.1. Since the calibration scene is static, the images can be captured camera by camera or with acquisition of 24-camera image frames simultaneously in one shot. For the 24-camera simultaneous capture case, the image frames are embedded in a 1920x1080 frame side by side as shown in Fig.3.12b. Then in MATLAB environment, I parsed the single camera images to use in the calibration step.

The SIFT based software is utilizing the feature and key point extraction, feature matching and pinhole camera model to calculate intrinsic and extrinsic parameters of each camera. Then I use a MATLAB script to calculate the camera coordinate vectors and pack the parameters as look up tables to feed the hardware system. The calibration is done once offline and the parameters are used from these look up tables during online processing.

The diameter of the calibration tube is important and should be compatible with the final application since the calibration process can give slightly different results due to the geometrical relation of the individual cameras and the optimal focus depth of each camera. For example I choose a 35 mm diameter tube which is compatible with the human bowel inner diameter.

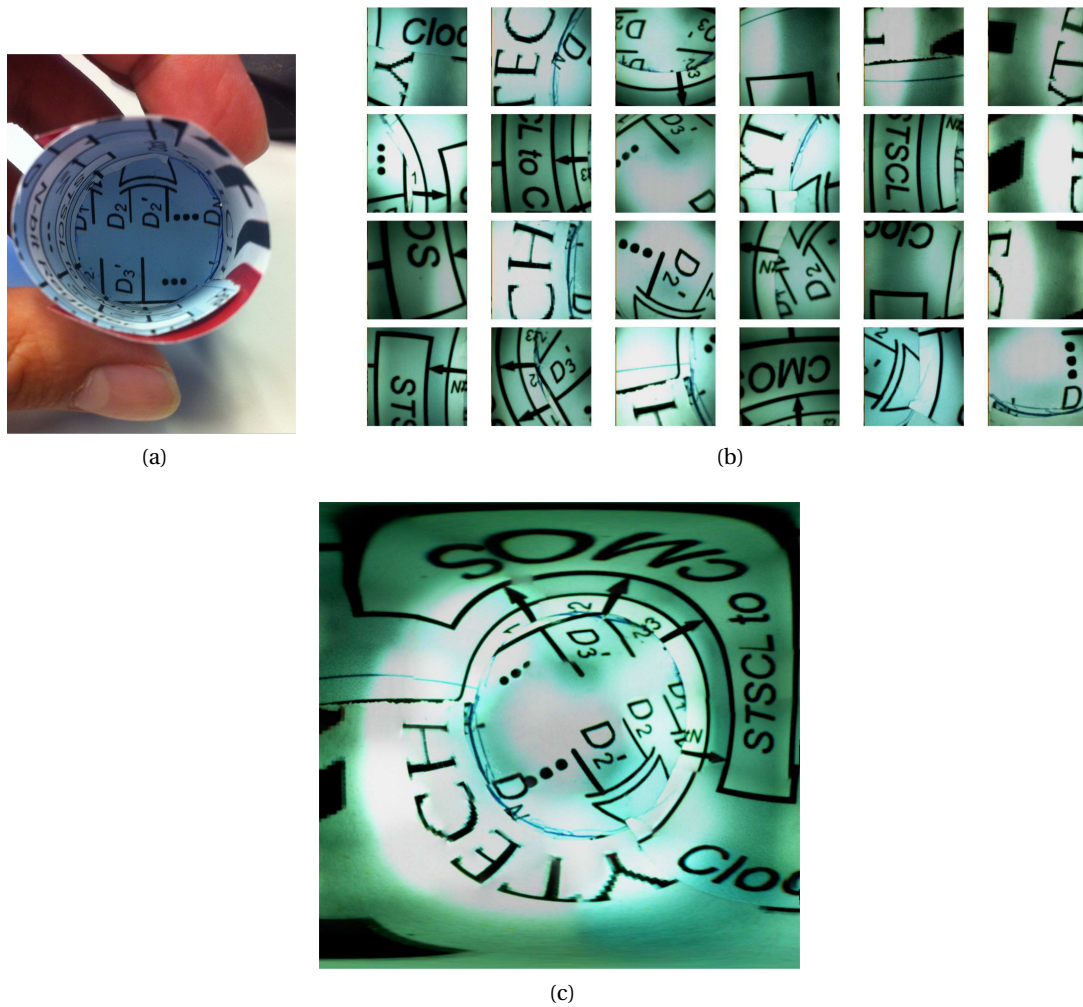


Figure 3.12 – The calibration environment and single camera frames of the prototype (a) calibration tube with 35 mm diameter (b) 24-camera single frames used in calibration (c) $180^{\circ} \times 180^{\circ}$ image generated from the 24 single frames at calibration procedure.

3.6.1 Discussion

The proposed calibration method is dependent on the texture of the scene that is captured by each of the cameras. This means the pixel resolution of each individual camera has an effect on the quality of the calibration process, i.e. determining the relative positions of the cameras to each other and the intrinsic parameters of each camera. We could also rely on the physical positioning of the cameras, however as mentioned, then this would be misleading due to the fabrication errors. As the resolution of the individual cameras decrease, the quality of the calibration decreases proportionally. As a limit, 1 pixel cameras could be used for each individual camera, which is equivalent to the approaches in [36], [37]. Then, there is no possibility to make such a calibration based on texture captured by each individual camera unit. Therefore, the methods like [36] and [37], which utilize very low (usually 1)

pixel resolution for individual cameras, need very precise mechanical placement or special fabrication of each individual unit. On the other hand, for systems like proposed in this work, there is no need for high precision in manufacturing since there is the option to be calibrated subsequently, thanks to the high pixel resolution of each individual camera unit.

Therefore the precision of the placement is affected by this hand placement and fixing method. However, since we use a post calibration method for defining the relative positions of the cameras and the intrinsic optical characteristics of each individual camera such as focal length, we overcome errors due to imprecise manufacturing at image processing level.

3.7 Conclusion

In this Chapter, the opto-mechanical aspects of the methods for designing miniaturized multi-camera hemispherical imaging system are explained. The approach for mimicking the insect eye capabilities is described. Since the previous methodologies are not optimized for miniaturization, new methods are introduced for camera arrangement. Alternative methods for camera arrangement are also discussed. The results show that the camera arrangement should be driven by the application needs such as size and imaging distance. Moreover, the camera size should be taken into consideration as a 3 dimensional constraint for designing miniature multi-camera systems. The proposed camera arrangement method leads to compact hemispherical multi-camera imaging system design. The built-in distributed illumination idea for the designed compound eye is explained in the scope of the chapter as well.

4 Image Processing Techniques developed for Multi-camera systems

In this chapter, image processing solutions for creating seamless and high quality compound panoramic image out of the multiple camera system is described. The work explained in this chapter is described partially in [74]. The neural superposition type eyes of insects combine the intensity information from different optical channels and form an erect single image [18]. This is the base for the second part of the modeling of the insect eye concept for the system proposed in this thesis.

4.1 Light Field Imaging for a Neural Superposition Virtual Ommatidia

After having the camera placement suitable for miniaturization, and calibration of the camera system, for generating the final composite image, a virtual ommatidia (VO) is defined, which is supposed to sample a light ray from space. Then the whole panoramic composite image will be the combination of the intensity values of these VOs. In this way the problem of generation of the intensity values of each virtual ommatidia is reduced to a light ray tracing problem where the proper optical channels (cameras) that are contributing to the corresponding VO should be selected first and a superposition of the intensity values from different cameras should be made. Therefore, on the image processing side, to generate the VO intensity values I utilize the vector based ray tracing method described in [65]. Here, I use the ray tracing method on one half of a unit sphere, which is then assumed to be positioned at a relatively far distance from the compound eye, $R_{virtual}$, as shown in Fig.4.1. then the question arises here is that how to estimate the intensity value for a VO. The second question is that, what kind of useful information can be obtained by utilizing the overlapping field of views of the adjacent cameras contributing to the each VO direction. In this Chapter mainly these two questions are addressed.

As described in Section-2.4.2, to generate a final intensity value per virtual ommatidia, it is

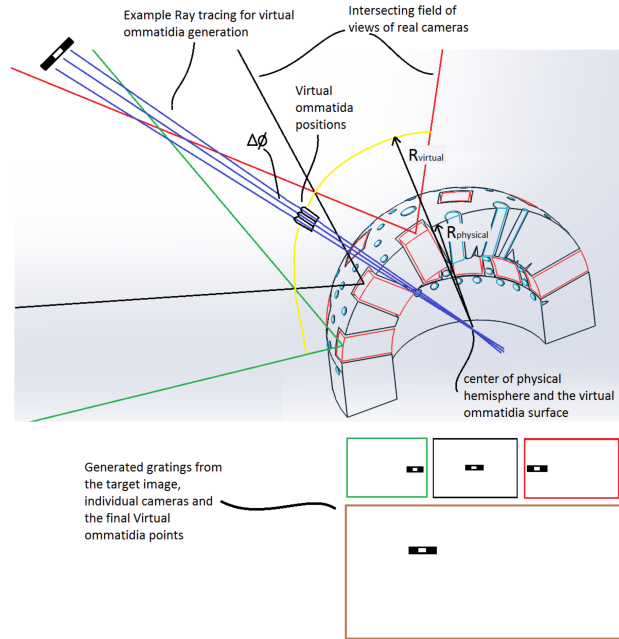


Figure 4.1 – Virtual ommatidia sampling concept with the proposed prototype

needed to determine the two angles θ and φ on a spherical coordinate system, which has its origin at the center of the insect eye camera. For that the angles are generated starting from 0 and the whole hemispherical field of view is scanned with certain number of steps, which corresponds to the $\Delta\varphi$ in the insect eye model. After this step, the cameras which are *seeing* this direction should be determined.

For this purpose, as described in Section-2.4.2 the direction vector is projected onto the camera coordinate system. Since the camera coordinate vectors and the virtual ommatidia direction vector are chosen as unit vectors, the operation is reduced to dot products. Here, the assumption is that the camera centers and the projection center are the same point, from which the VO or panorama direction vectors are originated. The second assumption is that, the observed objects by the camera system are relatively far from the camera surfaces when compared to the outer radius of mechanical hemisphere construction. In this way, the pinhole camera [63] definitions become applicable provided in the Section-2.4.1, in the expense of sacrificing the third dimension information related to the objects observed at the VO directions.

To sum up, the method described in the Section-2.4.2 can be summarized as follows. First, it is vital to make a dot product of the direction vector $\vec{\omega}$ and each camera unit vector \vec{i} , which represents the viewing direction of the camera and originates from the insect eye camera center as well. If the angle between the camera and the ommatidia vector $\vec{\omega}_j$ is less than half of the angle of view of the i th camera, then the camera i is said to be *contributing* to the intensity value of ommatidia represented by $\vec{\omega}_j$. After this step, it is needed to estimate the ray

position $\vec{\omega}$ by projecting it on the each *contributing camera* image frame to determine and pick the intensity value from the camera image frames. Then an estimation of the intensity value should be made. Different methods are proposed literature for the estimation of the pixel value, which are nearest neighbor [64], linear interpolation [70], Gaussian smoothed weighted average [66]. These methods are suffering from either the visible camera seams or the blurring at the object edges due to the missed third dimension information and parallax. In the following section, a better quality panorama generation method to overcome these problems is described which I proposed first in [74].

4.2 Better quality panorama generation with probabilistic methods

In this section, a novel method is presented to improve the quality of panoramic images on a spherically arranged multi-sensor imaging system. The new method is composed of two approaches. The first approach proposed is based on mapping the panorama generation problem onto a Markov Random Field (MRF) and then estimating posterior probabilities from initial likelihoods. The novelty of approach is based on extracting the prior evidence from the registration information of multiple cameras and estimating expected value on an undirected graph. The second part of the method is a geometrical approach targeting a better estimation for the initial priors, which is also not applied before. The aim of both approaches is to decrease the parallax errors and ghosting effects, which occur due to the nature of multi-camera systems.

It is shown that, instead of directly using independent intensity coefficients extracted from registration information, applying a neighborhood based local probability distribution for each pixel of panorama gives better results. The registration information is considered as a prior knowledge to find more accurate evidences. Visual comparisons are provided to show the achieved quality enhancement in terms of seamless and more natural panoramic image with less ghosting effects. Since the registration priors are used effectively with a single iteration step in a 4 connected neighborhood, the need for an intensity based loopy and iterative inference method is prohibited. Hence, the proposed methods are suitable for real-time hardware implementation. The proposed hardware implementation of the method for real-time operation is described in Section-5.4.2.

4.2.1 Panorama generation as an Inference problem

I define the panorama generation problem as a probabilistic inference problem and then propose a method for reducing the seam and parallax errors in the final panoramic image. High quality image mosaicing and panorama generation are well studied in many works [41, 45, 75, 76]. Generating panorama sequences is a challenging task, and in the past, it has been conventionally achieved by different approaches such as very wide field of view

(FOV) lens or convex mirror based single image sensor systems [77]. Recently, multi-sensor systems where the surrounding scene of the cameras is partially captured by different sensors are proposed for panorama generation [64, 78]. To achieve high quality and high resolution panoramic images, multi-sensor systems cater to better results due to less optical distortion and better sampling capability. However, accomplishing the processing of multiple images captured by multiple sensors seeing partially the same scene for the panorama generation is not trivial and requires special algorithmic approaches [64, 66, 78].

As mentioned in the Section-2.4.2, in [64], authors describe a method for implementing a spherical camera system, named as Panoptic, where the individual image sensors are arranged on a hemispherical frame. The construction aspects such as camera placement and manufacturing tolerances for the spherical multi-camera system are discussed in detail where two methods are reported for generating hemispherical panorama image [64]. First approach is choosing a best seeing camera for each panorama pixel based on calibration information. Second method is blending partially overlapping images captured by different cameras with weighted linear interpolation. The weights are chosen according to calibration information. The output image has artifacts such as ghosting and blurring effects at object edges due to the linear aggregation. The output quality is enhanced in [66] by applying a vignetting correction before interpolation and Gaussian coefficients during the aggregation step. However, since none of the algorithms take into account the information of neighboring light rays, the inferred panorama at the output is less realistic and visually poor. Moreover, there are steps in the algorithm which do not meet the geometrical constraints of the proposed system. Hence, the methodology is still open for development to improve the output image quality.

In [78], the construction of a $360^\circ \times 90^\circ$ FOV panorama from six cameras placed on surfaces of a cubic shape is achieved by mapping the problem on a Markov random field (MRF) and applying belief propagation (BP) based energy minimization method. Although the system proposed in [78] is composed of a small number of cameras when compared to the one in [64], real-time operation is not reported. Nevertheless, the authors of [78] show that minimizing the energy of the graph provides a final panoramic image that has less ghosting effects; and the method reduces the errors due to mis-registration of individual image sensors.

Using MRF and Bayesian methods for image processing problems is not new and these methods are well studied in many works [79–81]. A comparative study is presented in [80] on the utilization of MRFs and applying different energy minimization methods for different vision problems such as object detection, resolution enhancement and panorama generation. In [79], the belief propagation and graph cut methods are analyzed for energy minimization on MRFs.

In the proposed approach here, the real-time panorama generation problem on a multi-camera system is defined as an inference problem; and a methodology is developed to solve it by using MRF for calculating a posterior probability distribution. The method explained in the scope of this work first extracts the priors for generating evidence from the

4.2. Better quality panorama generation with probabilistic methods

sensor structure and calibration information of the individual cameras: and then, it generates coefficients from the resulting marginal probabilities. In other words, a posterior probability distribution for each hidden node of MRF is calculated. Then, instead of choosing the best label for the corresponding node, expected value estimation is done for the intensity value of each pixel of the panorama by using the joint probability distributions. Visual results are represented in Section-4.2.3. It is shown that using the proposed method provides better results than the panorama outputs presented in [64] and [66].

4.2.2 Proposed approach

The methods for calculation of the intensity values for panorama pixels proposed in [64] and [66] are explained in section-2.4.2. The 3 techniques named as the nearest neighbor, linear blending and Gaussian weighted blending are given by the equations (4.1), (4.4) and (4.5) respectively as described in Chapter-2, Section-2.4.2.

$$\vec{r}_i = (\vec{q} - \vec{t}_i) - ((\vec{q} - \vec{t}_i) \cdot \vec{\omega}) \times \vec{\omega} \quad (4.1)$$

$$j = \underset{i \in I}{\operatorname{argmin}}(|r_i|) \quad (4.2)$$

$$L(\vec{q}, \vec{\omega}) = L(\vec{c}_j, \vec{\omega}) \quad (4.3)$$

$$L(\vec{q}, \vec{\omega}) = \frac{\sum_{i \in I} \frac{1}{|r_i|} L(\vec{c}_i, \vec{\omega})}{\sum_{i \in I} \frac{1}{|r_i|}} \quad (4.4)$$

$$L(\vec{q}, \vec{\omega}) = \frac{\sum_{i \in I} \frac{1}{|r_i|} e^{-\frac{d_j^2}{2\sigma d^2}} L(\vec{c}_i, \vec{\omega})}{\sum_{i \in I} \frac{1}{|r_i|} e^{-\frac{d_j^2}{2\sigma d^2}}} \quad (4.5)$$

In my approach, the panorama pixel construction problem is considered in the probabilistic domain unlike the previous methods proposed in [64] and [66]. Then the previously proposed blending methods for the spherical multi-camera system [64] and [66] can be classified in the context of this work as follows: The nearest neighbor method [64] given by (4.1) can be seen as choosing the maximum likelihood estimate (MLE) over the probability distribution on the planar $\vec{\omega}$ -plane. The linear blending [64] expressed by (4.4) can be interpreted as Expected Value (EV) calculation by using the independent probabilities extracted from the $\vec{\omega}$ -plane projection. Finally, the Gaussian blending given in (4.5) also calculates expected value after

redistributing the probabilities by multiplying with Gaussian coefficients. Reconsidering the equation (4.4), which is a weighted average, the problem of inferring the panorama pixel can be viewed as an expected value calculation by using (4.6).

$$E[X] = \frac{\sum_i p_i x_i}{\sum_i p_i} \quad (4.6)$$

In (4.6), X is a random variable that can take the value from a set of $\{x_i\}$ with a corresponding probability from a set of $\{p_i\}$; and E is the expected value of X . In the same manner, any direction intensity value in the panorama L can be expressed as an expected value in (4.7) which is equivalent to (4.4).

$$E[L(\vec{q}, \vec{\omega})] = \frac{\sum_i p_i l_i}{\sum_i p_i}, \quad p_i = \frac{1}{|r_i|}, \quad l_i = L(\vec{c}_i, \vec{\omega}) \quad (4.7)$$

The MLE or nearest neighbor chooses one exact pixel captured by one of the cameras. Therefore, at camera transition points on the panorama, visible seams are inevitable and the image is not natural for a human observation as seen in Fig.4.4a and Fig.4.5a. EV method or linear interpolation makes a combination of the camera pixels by using the distance values on the $\vec{\omega}$ -plane as weights, which is referred as probabilities in the scope of this work. It provides smoother transitions but since the probability values are independent of the distribution of the adjacent panorama pixels, it causes ghosting effects at object boundaries.

Finally, EV with Gaussian coefficients (4.5) targets a better distribution for the independent probabilities of each $\vec{\omega}$ direction regarding to the position of the candidate pixel on the camera image planes. Although the observational choice of Gaussian distribution smoothens the transitions at seams and reduces the intensity differences among the panorama, it does not consider the distribution at neighboring directions. Thus the ghosting effects are not removed completely.

However, by the nature of image reconstruction problem, there is a strong spatial dependency between neighboring pixels. Hence, just applying an independent probability distribution by using projected distances on the planar plane cannot sufficiently deal with the parallax errors and blurring near object boundaries, which is the main driving motivation of this work.

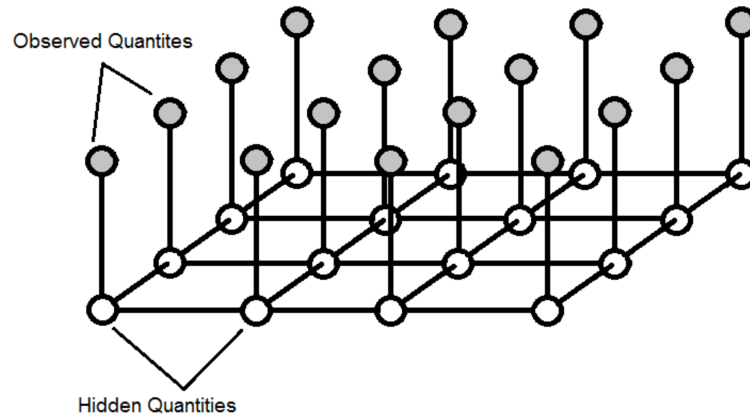


Figure 4.2 – Example graph representation for the panoramic image

Graph representation

It is proposed to apply a marginalizing process among the probabilities of neighboring $\vec{\omega}$ directions, and to get more accurate probability distribution for each $\vec{\omega}$ direction before calculating the expected intensity value of each panorama pixel. To achieve this, mapping the problem onto an undirected graph where dependency is defined as the connectivity of the nodes is required. Markov random field (MRF) representations are appropriate models for this kind of imaging problems.

The general structure of a MRF is an undirected graph $G = (V,E)$ with observed nodes and hidden nodes as shown in Fig.4.2. The panoramic image is mapped onto a MRF where the set of nodes V are composed of the hidden nodes X , and the observed random variables Y . The intensity value of a particular direction $\vec{\omega}_u$ is a hidden quantity that will be inferred. The observed values are the intensity values captured by the individual image sensors in the system. Each observed variable has a bias that is calculated as the prior probability value at the $\vec{\omega}$ -plane projection step. The target is to strengthen this bias by considering the neighboring probability distributions such that the resulting panorama pixel intensity will be more natural.

Unlike the many cases where MRFs are used to label each hidden node and choose a best matching random variable, the target in this work is to calculate an expected value for the hidden node by using the probability distribution of the observed variables. Hence, a *maximum a posteriori* (MAP) solution is not taken. Since the final intensity value for each $\vec{\omega}_u$ direction will be calculated by weighted aggregation at the last algorithmic step, we do not take the intensity differences of the neighboring $\vec{\omega}_u$ directions; therefore, a compatibility potential function is not used. Thus, the only potentials used between neighboring hidden nodes X are the ϕ_{iu} as given in (4.8) where u represents a hidden node, i represents the camera indices in the system. So ϕ_{iu} are the probability values of each camera neighboring nodes extracted from the distances on the $\vec{\omega}_u$ -plane. Then the joint probability distribution for each

4-connected neighbor set of a direction $\vec{\omega}_u$ is defined as (4.8). In (4.8), N_u represents the 4-connected neighbor set (i.e. up, down, left, and right neighbors) of direction $\vec{\omega}_u$ where Z is a normalization factor.

$$p_{iu} = \frac{1}{Z} \phi_{iu} \prod_{v \in N_u} \phi_{iv}, \quad \phi_{iu} = \frac{1}{r_{iu}} \quad (4.8)$$

The expression (4.8) is based on the local Markov property that is a variable which is conditionally independent of all other variables given its neighbors. With (4.8), the posterior probability distribution for each node is achieved, which is equivalent to attaining marginal probabilities in the 4-connected neighborhood of a certain direction. Then, these posterior probabilities are replaced as new weights in (4.7) to calculate the expected value as the final intensity for each of the panorama pixel u as in equation (4.9)

$$E[L(\vec{q}, \vec{\omega}_u)] = \frac{\sum_i p_{iu} l_{iu}}{\sum_i p_{iu}}, \quad l_{iu} = L(\vec{c}_i, \vec{\omega}_u) \quad (4.9)$$

By taking the probability distributions of neighboring nodes into account, stronger evidence is calculated from the initial priors. Hence, the reconstructed panorama is more natural with the reduced defects like ghosting effects and the visible seams. Visual comparisons are provided in experimental results section.

Accurate prior estimation from spherical model

The planar $\vec{\omega}$ -plane approach as described in the Section-2.4.2 and illustrated in Fig.2.5, which is used for extracting distance values, is based on the constant light flux (CLF) assumption. It is assumed that the light intensity remains constant on the trajectory of any light ray. This assumption is true but not sufficient for the case of the spherical arrangement of the cameras. For a better extraction, the geometry of the camera array should be considered as well. Therefore, the CLF assumption is extended by changing the projection surface to a spherical geometry, which is inspired from the actual physical placements of the individual cameras on the hemispherical frame. This modification is illustrated in Fig.4.3; the previous projection method, which can be also extracted from Fig.2.5, is provided in Fig.4.3a. In Fig.4.3b, where the new proposal is shown, the planar projection surface is replaced by the spherical surface. The light flux is still assumed to be constant but a direction factor is added due to the spherical camera arrangement geometry. According to this new proposal, the light flux is assumed constant on a trajectory of any light ray through the point of observation, which is the center of the sphere in default case.

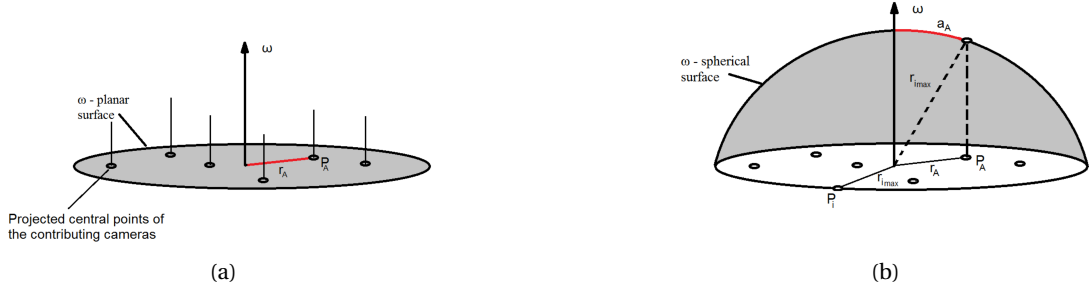


Figure 4.3 – The proposed method for accurate prior estimation from spherical model, (a) the previous planar model, (b) proposed spherical arrangement for estimating the priors

The virtual spherical surface is constructed by bounding the planar surface with a circle that has a radius of the maximum distance $|r_i|$ for the current $\vec{\omega}$ vector projection. Extending this circle to 3D symmetrically, which results in a hemisphere surface with a radius of r_{imax} . The arc values a_i are defined as the arc distance from the pole of the $\vec{\omega}$ -sphere and the re-projected position of the camera center point on the sphere. The relation between the a_i and the corresponding r_i can be solved easily by using simple geometry, which can be expressed in (4.10); and when the distances are normalized to 1, the projection surface becomes a unit sphere. Then, arc distance can be calculated as (4.11).

$$a_i = r_{imax} \left[\frac{\pi}{2} - \arccos \left(\frac{|r_i|}{r_{imax}} \right) \right], \quad r_{imax} = \max\{|r_i|\} \quad (4.10)$$

$$a_{inorm} = \frac{\pi}{2} - \arccos(|r_i|) \quad (4.11)$$

By applying this transform to the projection surface and the distance values, the absolute values of the relative distances between the projected camera centers on the surface and the vector have become more realistic. That is, further points, which are less likely to be candidate label for the current $\vec{\omega}$ direction, become more distant with respect to the closer points since they move on to the curved surface of the new spherical surface.

4.2.3 Experimental results

In Fig.4.4, an example spherical panorama image of a complex scene generated by using the proposed method is provided. The resolution of the final panorama image is 1920x512. The single images are captured by the 24 camera system I designed and the calibration is done as described in Chapter-3. The proposed method is implemented as a MATLAB script and compared with the previous systems visually. The hardware implementation of the method is

described in Section-5.4.2.

The image in Fig.4.4a is the panorama image of the same scene generated by using the nearest neighbor technique proposed in [64]. The comparison of two images in terms of seams can be done on these images. In Fig.4.4b, the linear blending method [70] is shown where the image looks more seamless, the camera boundaries are not as sharp as nearest neighbor. However there are ghosting effects at the camera boundaries especially at object edges. The results of the blending method [66], which is smoothing the weights of the linear blending with Gaussian filtering is shown in Fig.4.4c. The object edge ghosting effects and seams are removed when compared to the linear blending but they are still visible. The result of the method I propose is shown in Fig.4.4d. The improvement at the camera boundaries and around objects like the polyp in the middle can be seen. The 512x512 zoomed version of the object of interest, and the comparisons on the constructed images with the previous methods can be seen in Fig. 4.5a and 4.5d.

4.2.4 Discussion

In this section, panorama generation problem on a spherically arranged multi-camera system is considered in probabilistic domain and two novel approaches are developed. The final panorama is mapped on to an undirected graph, which is chosen as Markov random field; and the joint probability distributions are calculated on 4-connected neighborhood. Registration information extracted from the camera calibration parameters is used as likelihood prior and the neighboring pixel probability distributions are used to get more accurate probability distribution in order to get a seamless and natural spherical panorama. It is visually justified that instead of using linear combination [64] or aggregation by using pre-determined coefficients like Gaussian distribution [66], using the joint probability distribution gives better results. The reconstruction step is considered as Expected Value (EV) estimation for the intensity values of panorama pixels, which give more natural and seamless panoramic image when compared to a maximum likelihood estimate, named as nearest neighbor method in [64]. In order to start with more accurate priors before calculating joint probabilities, a geometrical correction is also proposed.

However, the used registration priors are actually fixed with the geometry of the structure and they are limited by the mapping method using the vector projection and the pinhole camera assumption. To use the dynamic scene information, the sampled intensity values in the light field can be utilized. In the next section this proposal has been analyzed.

4.3 Inter-Camera pixel intensity differences and its applications

For each virtual ommatidia, there are more than one contributing cameras. In ideal case, if all the cameras are identical and if we do not omit the distances of the objects from the camera surface, and if our calibration process is perfect, then the contributions of the each camera for

4.3. Inter-Camera pixel intensity differences and its applications

a specific virtual ommatidia direction should be same. The reason is simple; all the cameras should have sampled the same light ray from the surrounding light space for that specific direction.

However, in reality there are non-ideal behavior of the various components. As a first non-ideality, it is assumed that the objects are relatively far from the cameras and a weighted average of the intensity values from different contributing cameras are done. This causes blurry edges and ghosting in the final panoramic image.

Moreover, our calibration method is dependent on the environment, the distances of the objects at the moment that the calibration takes place. If the calibration is done with the objects farther than the expected distance ranges during the operation of the camera system, this will also cause using wrong calibration parameters for the camera positions and intrinsic parameters of the cameras. The last point is that, since the cameras are physically not identical, their color spectrum is slightly different from each other. As a result, when we add up all these points, obviously, there will be intensity differences of the cameras which are supposed to contribute to the same virtual ommatidia direction.

We analyzed this result and utilize these inter-camera differences for different purposes and create hypothesis for each case and tested them by applying to the visual data.

4.3.1 Object boundary detection

In our camera system as described in Chapter-3, the distance values on the orthogonal plane of VO directions are used as the measure of how likely a camera can contribute to a virtual ommatidia direction. Smaller distance r_i on the orthogonal plane of $\vec{\omega}_j$ means closer focal vector \vec{l}_i of the camera to the vector $\vec{\omega}_j$. This also means that for the cameras with the larger orthogonal distances, there will be more disparity on the object boundaries. So my claim here is that, for a given virtual ommatidia direction $\vec{\omega}_j$ if the intensity values of the largest and smallest distances are compared, this intensity difference should result in a quantity that contains information on either there is an object or not. The reason for that is, at the object boundaries there will be parallax which means high level of intensity difference. And the more different the compared contributing camera centers are, the more chance to have a parallax is.

To test and set up this hypothesis, for each VO direction, it is needed to find the cameras which have the maximum and minimum distances from the observing point center on the orthogonal plane of the VO under consideration. The orthogonal plane is illustrated in Fig.4.3a and the distances are calculated given by the (4.1), rewritten as (4.12). The camera indices with minimum and maximum distances are given by (4.13) and (4.14) respectively. The \vec{q} is the coordinate vector for observation point inside the hemisphere, which is $\{0, 0, 0\}$ by default but can be any point theoretically. Then the object boundary value L_{ob} of each direction is given by (4.15). If we have all the L_{ob} values for all the directions in the compound panoramic image, then we can have the object boundary map of whole field of view.

For the comparison of the camera pixel intensities, the gray scale intensities of the cameras can be utilized. For obtaining the gray scale intensity values YCbCr [82] scheme is used. Then the Y channel is used as gray level.

$$\vec{r}_i = (\vec{q} - \vec{t}_i) - ((\vec{q} - \vec{t}_i) \cdot \vec{\omega}) \times \vec{\omega} \quad (4.12)$$

$$j = \underset{i \in I}{\operatorname{argmin}}(|r_i|) \quad (4.13)$$

$$k = \underset{i \in I}{\operatorname{argmax}}(|r_i|) \quad (4.14)$$

$$L_{ob}(\vec{q}, \vec{\omega}) = |L(\vec{c}_k, \vec{\omega}) - L(\vec{c}_j, \vec{\omega})| \quad (4.15)$$

I have tested this hypothesis with the 24-camera miniaturized system and also with different multi-camera hemispherical systems. For example in Fig.4.6a, the images are captured with the system explained in [64]. This system have 15 cameras and have a 30 mm radius without illumination capability. Each of the cameras have 353x288 pixel resolution. The panoramic image reconstructed is shown in Fig.4.6b. The method I proposed [74], which described in section-4.2.2, is utilized for a 1920x512 pixel resolution. The images are generated in MATLAB environment by implementing the equations (4.12)-(4.15).

The object boundary map generated by the proposed method is in gray scale as seen in Fig.4.6a. There are certain regions seen in complete black with $L_{ob} = 0$. Actually these regions are the regions that are seen by only one camera, hence there is no possible intensity difference value. The regions seen only by one camera are shown in Fig.4.6c.

In order to have a decision image, a binary image can be obtained by applying a threshold value to the object boundary map as given in (4.16). The choice of ideal threshold value of th_{ob} can be made by considering the default intensity value differences which are not dependent on the scene. As described early in this section, there will be intensity differences due to the calibration imperfections and inter-camera color inconsistency. A resulting image with a threshold value of $th_{ob} = 0.2$ is shown in Fig.4.6d obtained from the image in Fig.4.6a where the gray intensity values are in the $[0, 1]$ interval. Here the threshold value is chosen empirically for illustration.

$$L_{ob}(\vec{q}, \vec{\omega}) = \begin{cases} 1, & \text{if } L_{ob} > th_{ob}. \\ 0, & \text{otherwise.} \end{cases} \quad (4.16)$$

I have also tested the method by using the colonoscopy image taken from a human colon model as shown in Fig.4.7. The images in Fig. 4.7 are $180^\circ \times 180^\circ$ panoramic images with

4.3. Inter-Camera pixel intensity differences and its applications

1080 × 1080 pixel resolution.

With the visual results, the hypothesis stated in the beginning is tested and validated. That is, for the proposed hemispherical camera system using the pixel mapping method utilized for panorama generation, the inter-camera pixel intensity differences of the most and less likely cameras contains an information about the boundaries of the objects resides in the field of view of the camera system. Although this result of the proposed approach is not adequate by itself for a automatized object detection method, it can be extended to machine vision applications such as automatic polyp detection [83].

4.3.2 Inter-camera pixel intensity differences as inference evidences

In the previous panorama construction methods [64, 66, 70, 74] the evidences used as interpolation weights are purely extracted from the geometrical relations of the cameras under the pinhole camera assumption. The geometrical relations between cameras are fixed and do not change according to the scene. Therefore, these methods are not taking into consideration the dynamically changing inter-camera pixel intensity differences. So here the proposal is to utilize the camera intensity values as evidences or weights for their contribution to the final scene. In this way the ghosting effects at the object boundaries can be even reduced.

To use the intensity differences a difference measure has to be defined between the intensity values of the contributing cameras to a particular direction. I use sum of absolute differences (SAD) as the difference measure due to its simplicity. The gray-scale intensity values are accommodated by converting the RGB values into the YCbCr [82] domain.

For each virtual ommatidia direction constructed, the pixel intensity difference of each camera with respect to the other contributing cameras is calculated. Then the total SAD value calculated for each contributing camera for each virtual ommatidia direction seen by more than one camera, given by (4.17). Then we use the reciprocal of the total SAD value as an evidence and combine with the geometrical evidences value by multiplying for each camera in each VO direction as given in (4.18) extending the method described in Section-4.2.2. In this way if the total SAD value is small, it will increase the probability of contribution for the camera to the direction in consideration, if SAD value is high, it will decrease the probability.

$$SAD_i = \sum_{j=1}^{N_{cont}} |I_i - I_j| \quad (4.17)$$

$$E [L(\vec{q}, \vec{\omega}_u)] = \frac{\sum_i \frac{1}{SAD_i} p_{iu} l_{iu}}{\sum_i \frac{1}{SAD_i} p_{iu}}, \quad l_{iu} = L(\vec{c}_i, \vec{\omega}_u) \quad (4.18)$$

I applied the method to the colonoscopy scene and the resulting image is shown in Fig.4.8a. The visual result is compared with the resulting image with the method described in Section-4.2.2 as shown in Fig.4.8b. The method performs accurately at the object boundaries for directions seen by more than two cameras since the SAD values are equal and bring equal weights for the directions seen by only two cameras. A zoomed image comparison for the polyp is given in Fig.4.8, where a small portion at the polyp boundary is seen by 3 cameras. The resulting image is shown in Fig.4.8c and the image for the method described in Section-4.2.2 in Fig.4.8d.

4.3.3 Inter-Camera Pixel Intensity Differences as a Quality Measure

The inter-camera differences can also be used for estimating a noise figure of the camera system. The noise figure for images is generally measured with peak signal to noise ratio (PSNR). For the previous work in [65–67], there are no quantitative methods proposed for measuring the quality of the resulting systems. In [65–67], a visual judgment of the resulting images is done since there are no ground truth images available.

The PSNR measure is used commonly for measuring the noise added with an image processing method applied to an original image by making a comparison with the resulting noisy image. Since there is not a ground truth image to compare for panorama generation, the classical approach for taking an original image and a noisy image is not taken. Instead, the proposed method is a measure of how much error we can produce for a given virtual ommatidia direction. As described early in this section, in the ideal case the contributions of the cameras should be the same for a given VO direction.

An estimate of how noisy images are created by the multi-camera system can be achieved by using inter-camera pixel intensity differences as follows: For any given direction that is tried to be reconstructed from the contributing cameras, a mean square error is calculated by using (4.19). In (4.19), the N_{cont} is the number of cameras contributing to the direction. Then we calculate the mean square error (MSE) for the whole image by averaging the VO MSE values over the whole image. Then with the MSE of the whole image we calculate the PSNR value as given in 4.21 where MAX_I denotes the maximum possible intensity value.

$$MSE_{\omega} = \frac{1}{N_{cont}} \sum_{j=1}^{N_{cont}} \sum_{i=j}^{N_{cont}} [I_i - I_j]^2 \quad (4.19)$$

$$MSE_{pan} = \frac{1}{mn} \sum_{k=1}^m \sum_{l=1}^n MSE_{\omega} [k][l] \quad (4.20)$$

$$PSNR_{pan} = 20 \log_{10}(MAX_I) - 10 \log_{10}(MSE_{pan}) \quad (4.21)$$

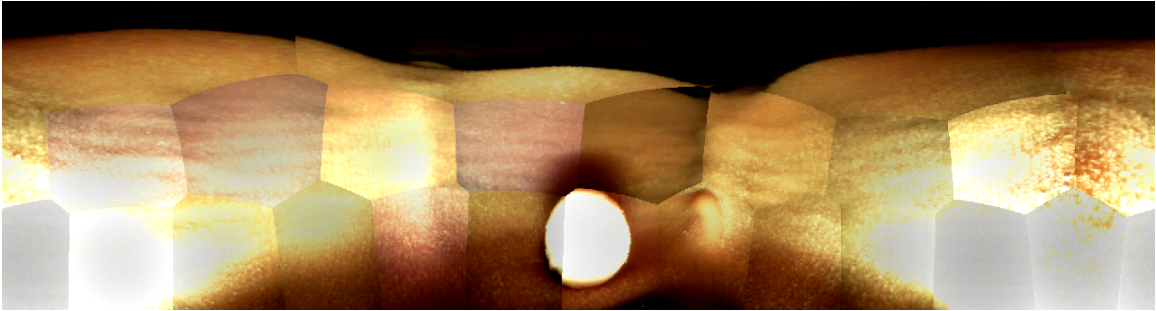
In this way, a total error estimation for the system can be made which combines the errors due to the calibration imperfections, inter-camera color inconsistency and the error caused by the assumption of far objects. Since the method is dependent on the textures and colors in the scene, for each of the cases, the scene should be chosen with different criterion. For example for estimating the PSNR for inter-camera color inconsistency, a single color spherical board can be utilized with sufficiently large radius, minimizing the calibration mismatches and parallax. For estimating PSNR caused by the calibration mismatches a textured spherical checker board can be used with different diameters and the change of the PSNR can be calculated. For parallax errors, a textured scene with objects at different distances should be chosen. As an example, for the image shown in Fig.4.8a, a PSNR=23.55 dB is calculated.

4.3.4 Discussion

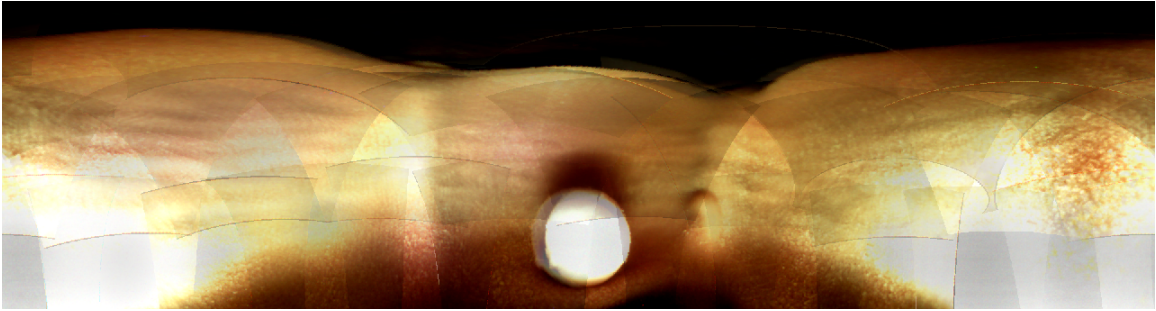
The methods described in this section for utilizing inter-camera differences are given as example uses of the capabilities of the designed multi-camera panoramic imaging system. They are intended for better quality panorama generation and providing a basic input for further machine vision tasks such as automatic polyp detection. To accommodate the methods in full performance, systems that have more camera overlap would be ideal.

4.4 Conclusion

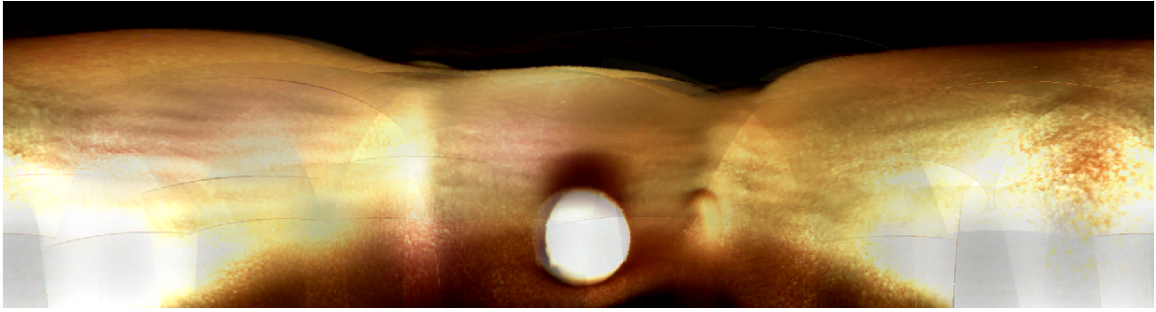
In this Chapter, the image processing methods to generate high quality panoramic images from the hemispherical multi-camera system and utilizing the overlapping field of views for different applications are explained. The approach that is considering the panorama generation problem as an inference problem is described. With the visual results, it is shown that the proposed method performs better when compared to the methods in literature. By considering the overlapping field of views of the adjacent cameras, methods are disclosed to generate pre-processing information for different applications like object boundary detection and parallax removal. A qualitative method is described for assessing the performance of the designed multi-camera system by using inter-camera pixel intensity differences.



(a)



(b)



(c)



(d)

Figure 4.4 – The comparison of the probabilistic inference method (d) with nearest neighbor method [64](a), linear blending method [70](b), linear blending with Gaussian smoothed weights [66] (c). The images have 1920x512 resolution, generated in MATLAB using the output images from our miniaturized 24-camera system.

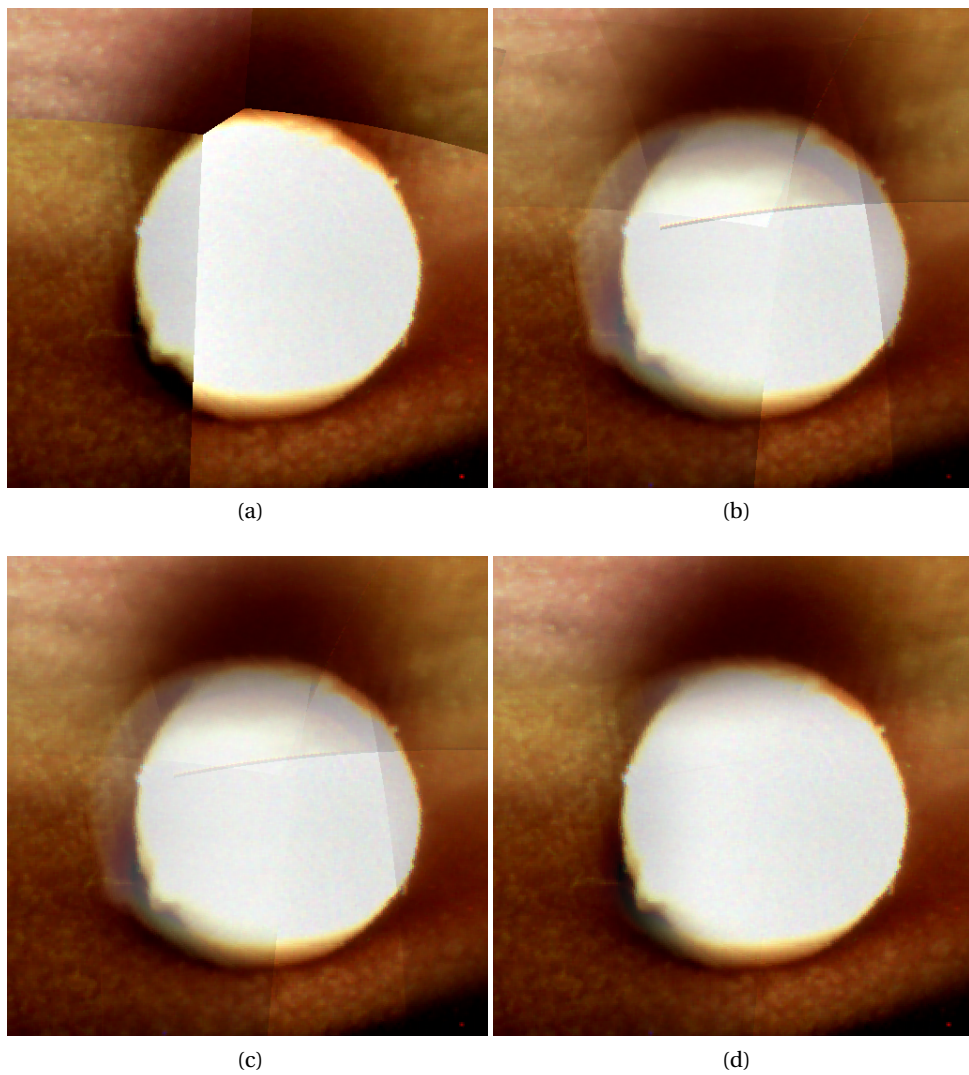


Figure 4.5 – The comparison for 2x zoomed on the polyp to show the effect on object boundaries. nearest neighbor(a), linear blending(b), gaussian smoothed linear blending (c) and proposed probabilistic inference method (d). The images have 512x512 resolution, generated in MATLAB using the output images from our miniaturized 24-camera system.

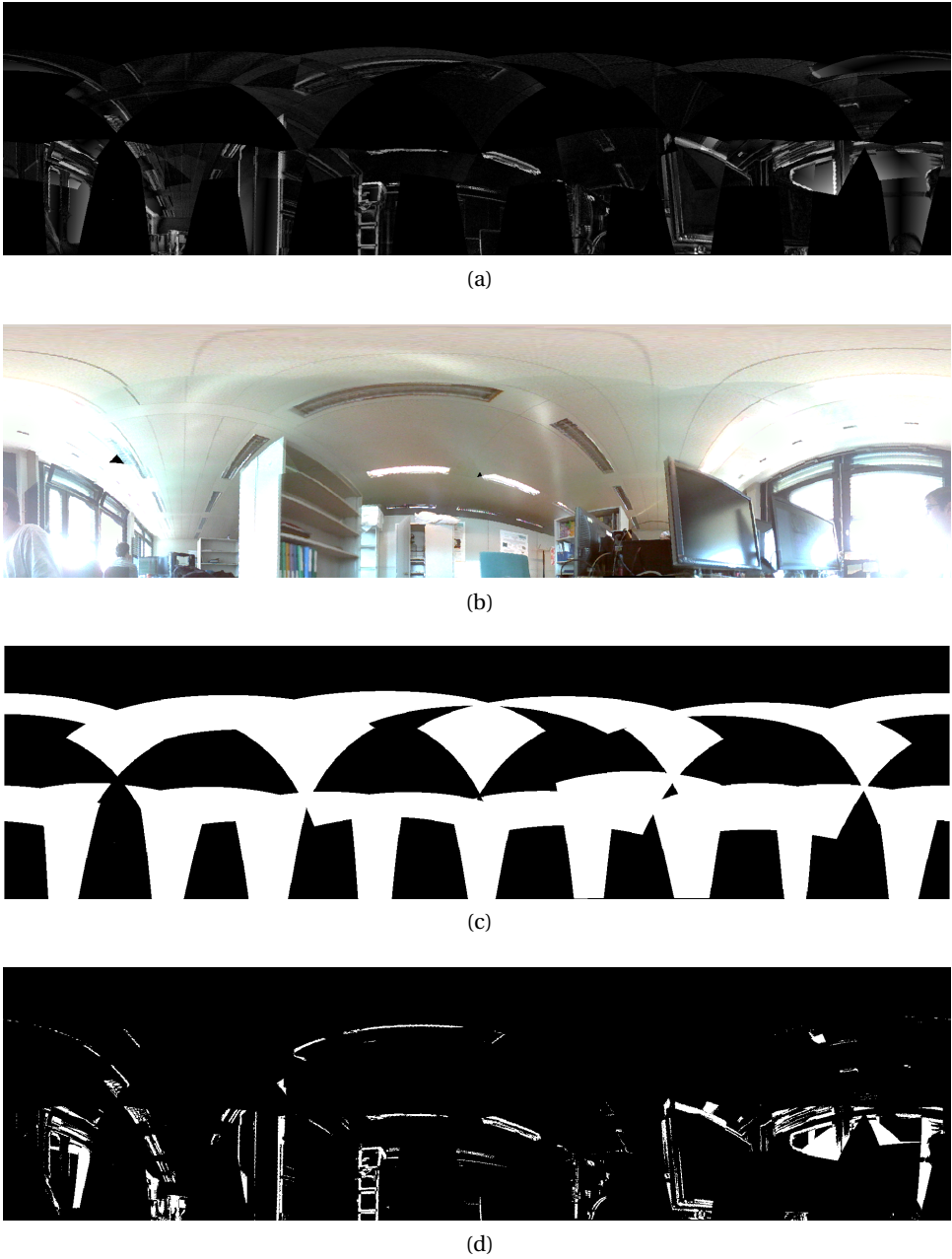


Figure 4.6 – The results obtained from the proposed method for object boundary detection. (a) The object boundary map in gray scale, (b) Constructed panorama image (c) the regions seen only one camera in shown in black (d) final object boundary map with a threshold $th_{ob} = 0.2$

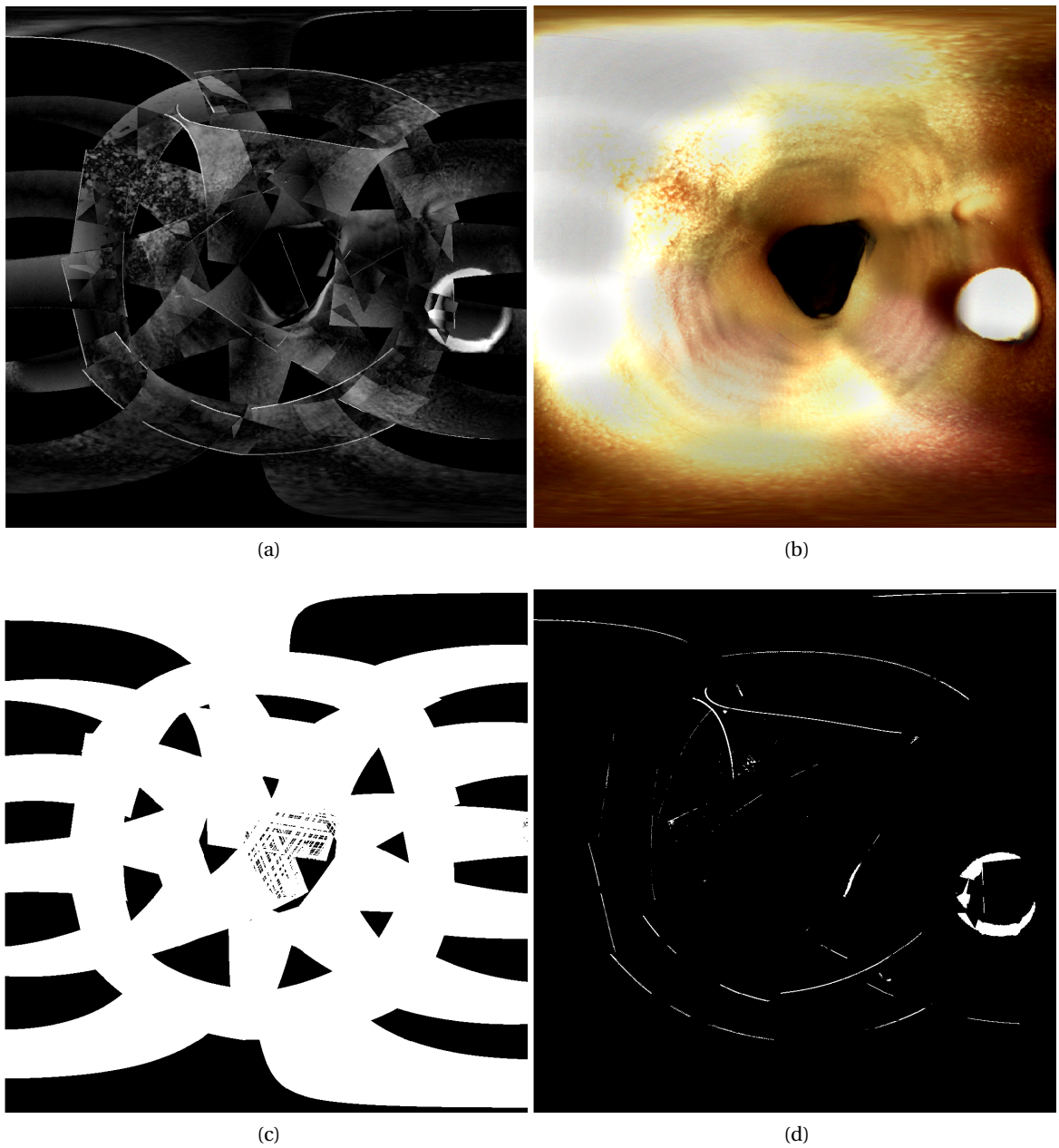


Figure 4.7 – The results obtained from the proposed method for object boundary detection. (a) The object boundary map in gray scale, (b) Constructed panorama image (c) the regions seen only one camera in shown in black (d) final object boundary map with a threshold $th_{ob} = 0.6$

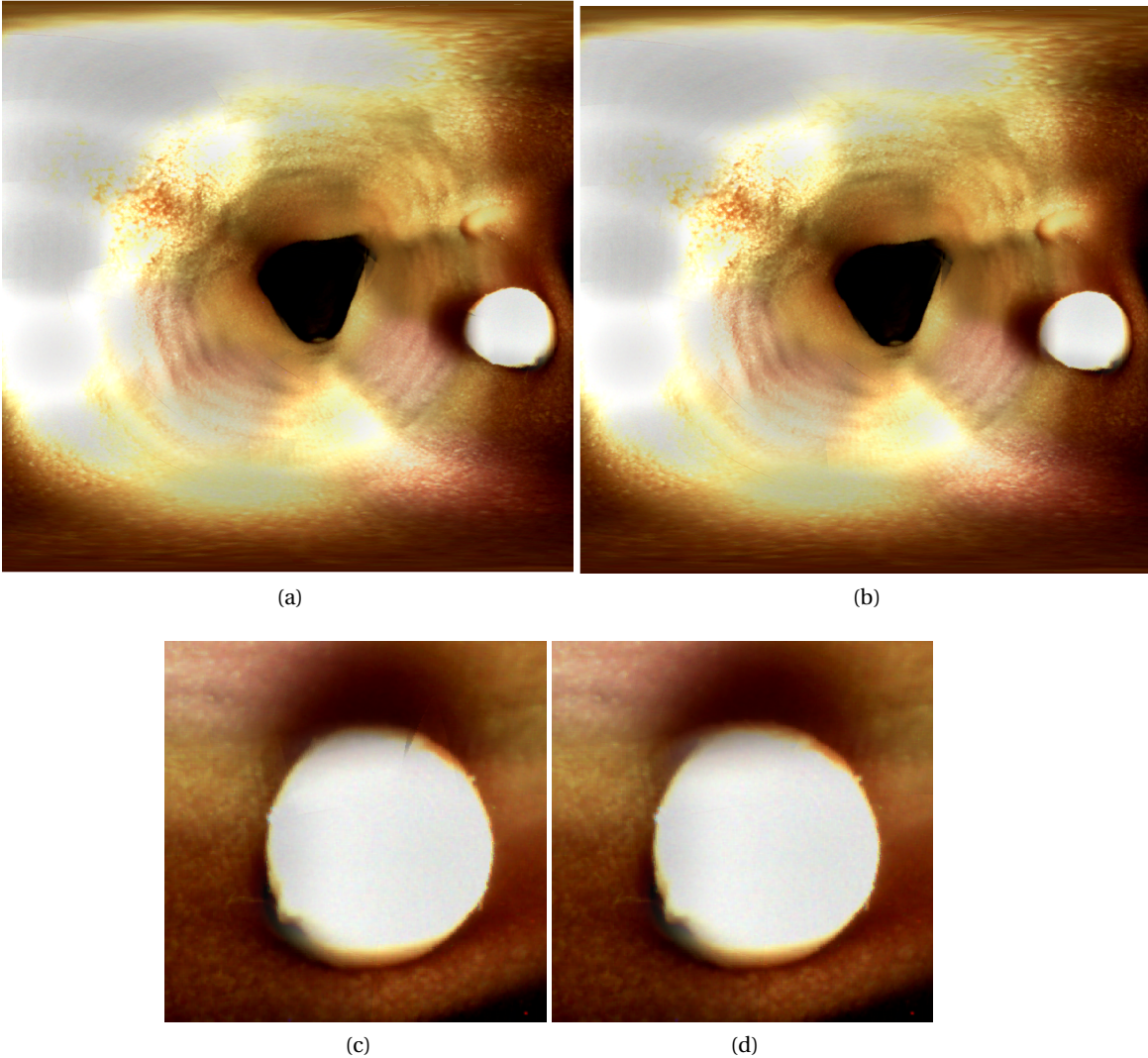


Figure 4.8 – The results obtained from the proposed method for SAD based interpolation. (a) The resulting image in $180^{\circ} \times 180^{\circ}$ angle of view 1080x1080p resolution, (b) The same reconstruction with method in [74] (c) the zoomed image on the polyp object for SAD based method (d) the zoomed image with previous method

5 FPGA Embedded System Design

In this Chapter, the digital hardware design details for the designed and fabricated miniaturized multiple camera system are explained. The system is composed of custom PCB design for electrical interfacing, an FPGA system where the digital circuit design for the real-time panoramic video generation is embedded.

5.1 Single Camera Interface and Image Processing Blocks

Since the miniaturization is one of the key targets for the work presented here, the first selection criterion for the single camera is its size. Also having minimal I/O and programming requirements in terms of simple interfacing is important. Since the target applications are mostly mobile, the power figure of the sensor should also be as minimal as possible. The physical and optical specifications of the chosen camera for the realization is described in Section-3. The summary of the electrical characteristics of the camera is given in Table-5.1 and a simple block diagram of the image sensor is illustrated in Fig.5.1.

Table 5.1 – Single image sensor electrical specifications

Specification	Value
Shutter mode	rolling
ADC Resolution	10-bit
Data interface	Serial transfer, Low voltage differential signal (LVDS), 10 bit signal + 1 start 1 stop bit
Operation Mode	free running
Frame Rate	44-56 fps
Pixel Array	250x250

Since the image sensors chosen are simple and minimal, there are no additional image processing inside the image sensor. It generates raw Bayer data and does not accommodate

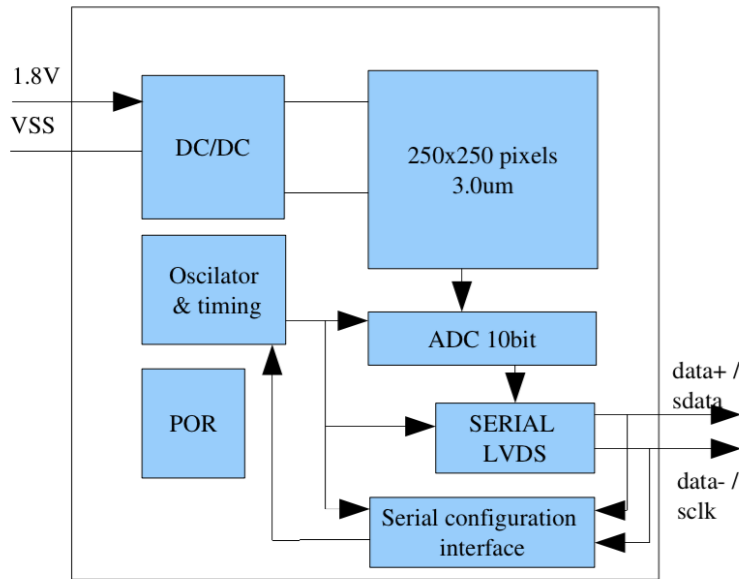


Figure 5.1 – Single image sensor block diagram

any additional processing such as demosaicing, white balancing etc. The design of such blocks is needed as a pipeline at the output of the cameras.

5.1.1 Interface Printed Circuit Board (PCB)

The image sensor is providing a low voltage differential signal (LVDS) serial output but does not have a standard differential impedance of 100 ohm. There was a need to make an interface PCB design to convert the signal to standard 100 ohm LVDS signal, compatible with the inputs of the chosen FPGA. Moreover, the 2-line interface of the single cameras is used as configuration serial clock and serial data for writing to the internal register of the camera. In the PCB design, a high output impedance buffer is also included to access to the line during the configuration time of the camera frame cycle. Another important aspect of the single cameras used is that the frame rate can only be changed by tuning the supply voltage of the camera between 1.8V-2.4V, which corresponds to a frame rate interval of 44fps-56fps.

A circuit block diagram of the designed PCB simplified for one camera interfacing is illustrated in Fig.5.2. The camera interface PCB contains a fast comparator to sample and convert the LVDS line to 1.8V 100 ohm standard LVDS, output of which can be connected directly to the LVDS inputs of FPGA. On the PCB, an adjustable voltage controller is also placed with a digitally controlled resistor at its adjust node through I²C interface. In this way, the supply voltage of each camera can be controlled digitally by the FPGA. Each of the camera interface PCBs supports 2 cameras. An example image of the designed and fabricated PCB is shown in Fig.5.3a. Since there will be more than one cameras, there is a need for addressing the I²C lines, for this purpose an I²C switch is included on the main board.

5.1. Single Camera Interface and Image Processing Blocks

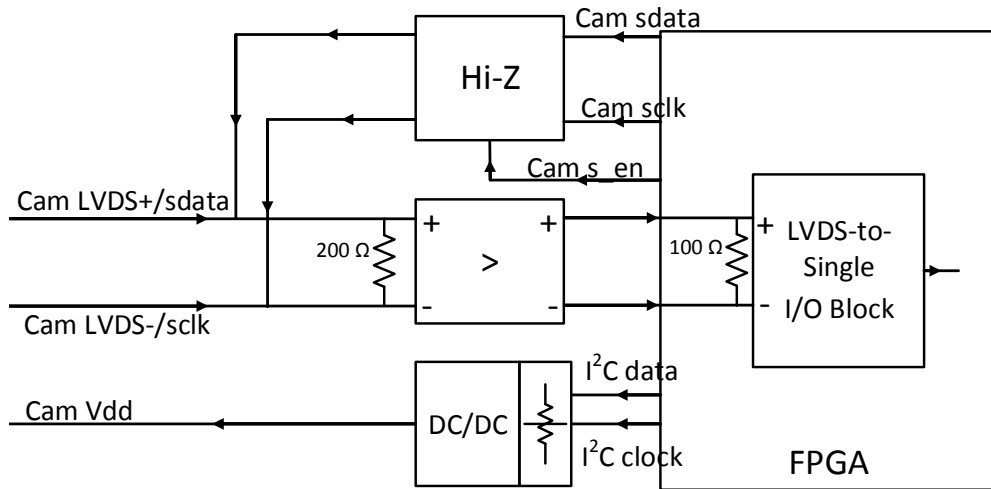


Figure 5.2 – Circuit block diagram for the PCB, simplified view for one camera interfacing

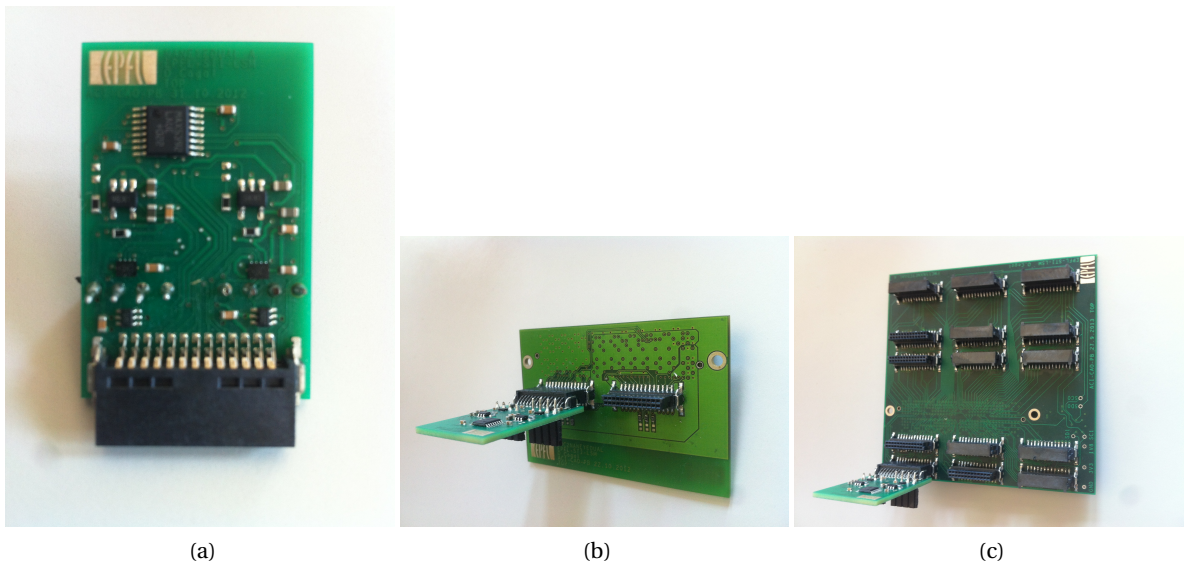


Figure 5.3 – The custom PCB designs for camera interfacing. (a) camera interface PCB for 2 cameras, (b) and (c) main board PCB supporting 4 and 30 cameras respectively.

The camera interface PCB units are connected to the FPGA through another custom made PCB as a main board which has a FPGA Mezzanine Card (FMC) connector compatible with the FPGA board used. For the development and analysis of the single camera hardware, a main board as shown in Fig.5.3b which supports 4 cameras and 2 single camera PCBs are designed and fabricated. After the verification of the PCB and hardware for the full system design, a main board that is capable of supporting 30 cameras is fabricated. The designed and fabricated final main PCB is shown in Fig.5.3c.

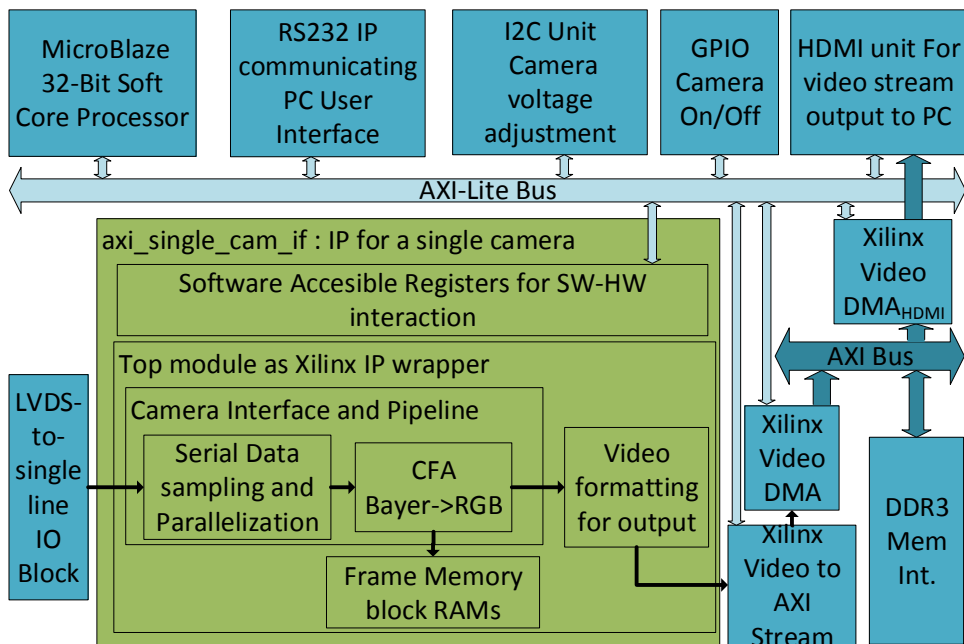


Figure 5.4 – The custom single camera pipeline as an FPGA AXI slave IP, and the embedded system designed for testing the interface

5.1.2 Single Camera Image processing pipeline

The designed pipeline for the single cameras is shown in Fig.5.4 with the whole microprocessor system for testing and verifying the unit. Since the target FPGA vendor is Xilinx and since the design should be configurable, the camera pipeline circuit is designed compatible with the Xilinx custom peripheral interface. The single camera module is an advanced extensible interface lite (AXI-lite) slave unit and it is accessible by the Microblaze embedded processor. In this way, it can be configured during run-time via software accessible registers. For the single camera embedded system, the configurable and controllable features are listed in Table-5.2. The design of sub-blocks of the camera pipeline is described in the following sections.

Sampling and Deserialization

Inside the FPGA, the single camera interface samples the incoming serial data with a 200 MHz clock and decodes the serial stream by detecting the edges and pulse widths. The encoding used in the serial stream is similar to Manchester encoding [84]. The average frequency of the camera signal is around 30 MHz. After sampling, the pixel stream is deserialized and 10-bit precision Bayer filtered pixel intensity values are obtained. Then Bayer demosaicing is applied to the image as described later in this section. The Bayer demosaicing methods and automatic white balancing techniques are explored and different method implementations for single camera pipeline are performed and compared as described in Appendix-A.

5.1. Single Camera Interface and Image Processing Blocks

Table 5.2 – Programmable features for single camera IP

Feature	Interface	Explanation
Camera Power ON/OFF	GPIO	to shut down and power up the camera with precise timing
Camera Supply Voltage	I ² C	controllable between 1.8V-2.4V with 256 steps
Camera gain register	GPIO	to change the gain register of the camera between 4 different levels
Frame buffering mode	Software accessible registers	select between Bayer/RGB frame buffering modes
Test pattern selection	Software accessible registers	select camera output or pre-defined test pattern registers

After the demosaicing, the pixel values fed into two different paths as shown in Fig.5.4. One of them is frame memory which is accessed by the panorama generation unit. The second path is converting the video stream to Xilinx AXI-stream and it feeds the data to double data rate 3 (DDR3) synchronous dynamic random access memory (SDRAM) interface through Xilinx video direct memory access (VDMA) units which are utilizing the AXI Bus interface for burst write operations. This latter path is to get the single camera streams visually through high definition multimedia interface (HDMI) for making the calibration step and showing the single camera videos individually as well during the operation. Then the HDMI IP, which is provided by the Analog Devices, is utilized for accessing the DDR3 RAM and it streams out the video frame through HDMI port.

The interface of Video-to-AXI stream is a simple interface and needs two synchronizations signals *hblank*, *vblank* for indicating line and frame inactive intervals respectively. In addition to that, as a standard interface, it also needs a data enable signal *DE* in case of modulating the line length. A pixel clock is also expected from the Xilinx Video-to-AXI stream interface block.

The cameras used for the system have a fixed sequencing for multiplexing the LVDS output and serial data input. Simply, the camera listens the serial line for a certain amount of time between two output image frames. The designed sampling and deserialization block is also responsible for following this sequencing and make the multiplexing from the FPGA side by using the *Cam_s_en* output when necessary. In this way access to the camera gain register is achievable.

Bayer Demosaicing

Modern digital imaging sensors are only able to measure gray scales whereas the human eye is able to distinguish colors rangin from 380 nm to 780 nm, which corresponds to a color range from violet to red. Fig.5.5 shows two different configurations for taking colored images.

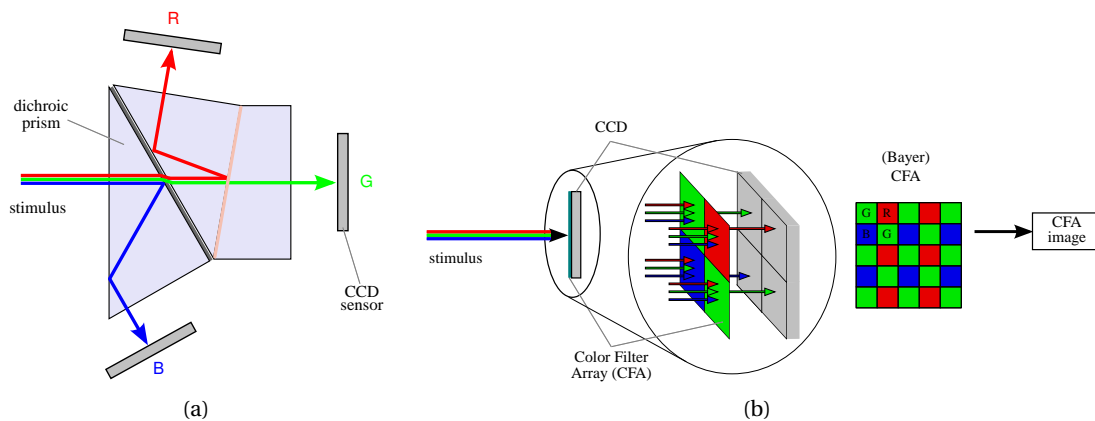


Figure 5.5 – Two possible configuration for color image sensing [85] (a) three sensors arrangement for capturing R,G,B channels separately, (b) using Bayer pattern with a single sensor.

Fig.5.5a shows that with a prism, the incident light can be separated by wavelength/color and then be passed to three different image sensor arrays which measure the intensities for each pixel. The upside of this configuration is that for each and every pixel all channels are directly measured, but it has several downsides: the first and obvious one is that it uses three sensor arrays, whereas other configurations only use one. However, this is not the most important drawback. Even more essential disadvantage is that the prism needs to have a certain size in order to guarantee a sufficiently linear light separation over each sensor. In addition to aforementioned drawbacks, this kind of systems demands a very high precision in manufacturing and therefore is rather expensive. Therefore, although these kind of cameras have advantages, the size itself is a sufficient reason not to use them for endoscopic imaging. Fig.5.5b shows another approach for producing colored images. This time all pixels are filtered individually. This makes big optical components obsolete as the filter can be implemented in a very small scale. However, this kind of system has other implications as described in [85].

In order to distinguish between the different colors, a pattern (Bayer pattern) of optical filters is laid over the sensor array. Usually, Bayer patterns have twice as many green passing filters than red or blue passing filters. The reason for the preference of green is that in the human eye 72% of the luminosity and the contrast perception are based on the green sensitive photoreceptor while red and blue sensitive photoreceptor contribute 21% and 7% respectively. Another and more practical reason, is that green lies in the middle of the visible spectrum and usually, lenses used in the optical part of an imaging device are normalized to a green wavelength and therefore, the green channel undergoes less optical distortion than the red and the blue channel.

For the sake of the reconstruction's simplicity, the Bayer pattern is most of the time a regular repetition of 2×2 cells as shown in Fig.5.6a and the combination of them is depicted as in Fig.5.6b.

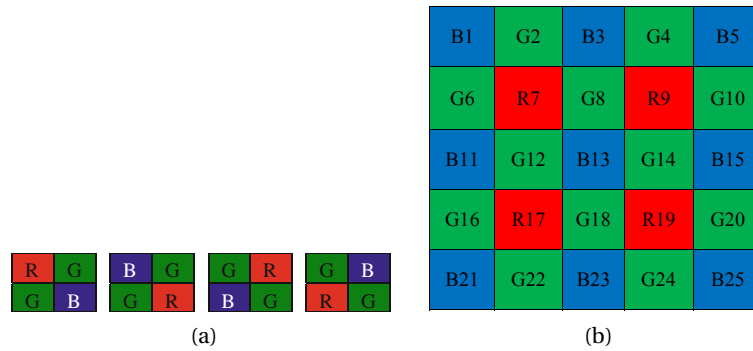


Figure 5.6 – Individual and combined representation of Bayer patterns (a) Possible regular Bayer patterns [86] (b) A 5x5 bayer matrix [87]

Naive Bayer to red-green-blue (RGB) Conversion The simplest way to extract the RGB information from the raw Bayer intensities (the actual measured values) would be to simply take a 2×2 section of the image, take the mean of the green pixels and then attribute the measured red and blue values directly. Even though this is done for producing high quality reference images, for most applications this is not desirable because the resolution of the image is divided by four.

Bilinear Interpolation Bilinear interpolation is the extension of linear interpolation from a 1 dimensional space to a 2 dimensional space. The simplest way to apply this method is to consider a 3×3 frame around the searched pixel. One could use bigger frames, but as this method is so simple, a bigger frame does not bring any improvement.

In order to do the interpolation, it is needed to distinguish between four cases: green pixel intensity value on a red line, green pixel value in a blue line, red and blue pixel. Due to the symmetry of each atomic 2×2 cell, the mathematical formulation of the interpolation for the red pixel is the same as for the blue pixel.

For the descriptions of the interpolation in (5.1) through (5.10), Fig.5.6b is used as a reference of the pixel location. Example: $R(B13)$ means the red value at the location of pixel $B13$.

Red Pixel For the red pixel four cases need to be distinguished [85]:

$$R(B13) = \frac{R7 + R9 + R17 + R19}{4} \quad (5.1)$$

$$R(R7) = R7 \quad (5.2)$$

$$R(G8) = \frac{R7 + R9}{2} \quad (5.3)$$

$$R(G12) = \frac{R7 + R17}{2} \quad (5.4)$$

Green pixel For the green pixels two cases need to be distinguished [85]:

$$G(B13) = \frac{G8 + G12 + G14 + G18}{4} \quad (5.5)$$

$$G(G12) = G12 \quad (5.6)$$

The estimation at red locations is the same as at blue locations.

Blue Pixel For the blue pixel four cases need to be distinguished [85]:

$$B(B13) = B13 \quad (5.7)$$

$$B(R7) = \frac{B1 + B3 + B11 + B13}{4} \quad (5.8)$$

$$B(G8) = \frac{B3 + B13}{2} \quad (5.9)$$

$$B(G12) = \frac{B11 + B13}{2} \quad (5.10)$$

Implementation Details The architecture of a bilinear demosaicing algorithm is rather basic as shown in Fig.5.7. After having passed the Bayer input through a frame buffer to construct a 3×3 sliding window, the Bayer values are combined in a sum generator where the sums are described in (5.1) to (5.5) are calculated for the same position. The sums can easily be generated using unsigned data types for the addition and shifts for the division. Before outputting the values, 3 of the sums need to be put to the correct output channel, corresponding to which kind of pixel the frame buffer's center is currently at.

The main problem of the bilinear and other linear demosaicing algorithm is that they have a

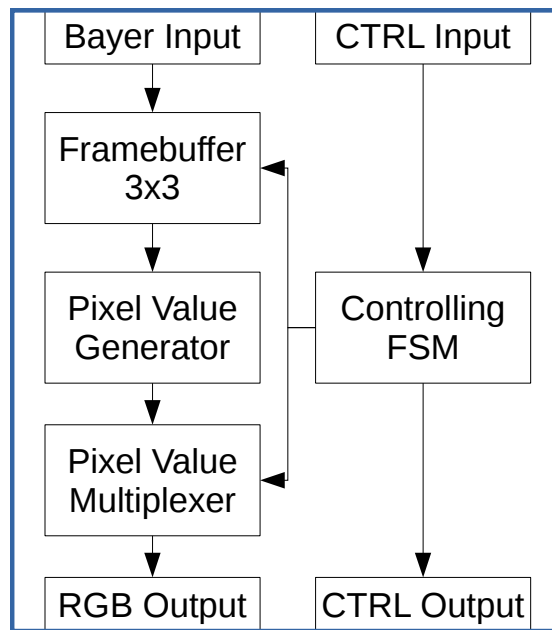


Figure 5.7 – Schematic of the bilinear Bayer to RGB converter architecture

low pass filtering effect which distorts edges. A way to prevent this low pass filtering effect is to estimate the edge direction and then do the still low-pass-like interpolation orthogonally to the estimated edge direction, in order to preserve the sharpness of the edge [87]. State of the art methods generating better quality RGB images are analyzed and implemented for comparison. From the conclusion of the analysis an implementation of a hybrid method [88] of gradient based demosaicing with linear filtering is also performed. The details of these analysis and implementations are presented in Appendix-A. From the resources point of view, since there are 24 camera interfaces, the FPGA resources needed for Bayer demosaicing are multiplied by 24 for the whole design. As a result, after analysis of the other implemented methods and bilinear, the trade-off has lead to the choice of bilinear interpolation in the final design.

5.1.3 Camera Model for Test Environment

In order to provide a test bench for the camera pipeline, a VHDL behavioral model of the image sensor has been developed. The developed model is a reduced version with the following features:

- The clock is given as an input (clk) instead of being generated by the on chip oscillator.
- The reset is given as an input (rst) instead of being generated by the power-on reset (POR) block.
- Only data interface downstream is included. No LVDS output is present but the serialized

data is available on a single bit output (dout).

- The serial configuration signals (serial_clk, serial_din) are given as inputs on two additional pins.
- Instead of the pixel array and ADC, there is a text-to-signal conversion.

The text file is generated by MATLAB and represents one single frame Bayer encoded data, with at maximum 250×250 entries, i.e. the total number of pixels, each of them corresponding to an intensity value ranging from 0 to 1023.

First each intensity value is transformed into a 10 bit data word and a start bit (equal to 1 and next to the MSB) and stop bit (equal to 0 and next to the LSB) are added. Then, in order to increase the robustness of the de-serialization on the receiver side under the presence of significant jitter, the data is XOR gated with the clock signal, thus producing an upward or downward transition for each sent bit.

As described in the sensor specifications, after power up the chip starts to loop autonomously the following sequence (1 PP-pixel period- is 12 clock cycles long):

1. Row 1 readout
 - 1.1 Transmission of continuous 0 (no start bit, XOR with clk, duration 3 PP).
 - 1.2 Transmission of 250 pixel values, first and last pixel are black (start bit, XOR with clk, duration 250 PP).

[...]
250. Row 250 readout
 - 250.1 Transmission of continuous 0 (no start bit, XOR with clk, duration 3 PP).
 - 250.2 Transmission of 250 pixel values, first and last pixel are black (start bit, XOR with clk, duration 250 PP).
251. Indication for end of frame: Transmission of continuous 0 (no start bit, no XOR with clk, duration 4 PP).
252. Time for serial configuration (LVDS receives inputs, duration 505 PP (actually less)).
253. Resynchronization pattern before start of frame: Transmission of continuous 0 (no start bit, XOR with clk, duration 250 PP (actually less)).

This sequence has been emulated using an FSM architecture. Pixel intensity values are provided at the output of a component behaving like a ROM: effectively, the text file is read, each entry is converted into 10 bit vector and then stored into an array signal.

5.2 Memory and Resource Analysis for Full System Implementation

The mechanical design of the system has 24 cameras in total. Since there is a final target of having a miniaturized single IC, reducing the external components as possible is a key design choice. For this purpose, having on-chip frame memories to store the entire frames of the 24 single camera is needed. The cameras chosen have 250×250 pixels. Each pixel is represented by 10 bit intensity value. During the Bayer to RGB conversion 8 most significant bits are used and the RGB result is represented in 24 bits. There are two choices for keeping the pixels as color data: RGB565 or RGB888. In the RGB565 representation, the red, blue and green channels are represented with 5, 6 and 5 bits respectively. So this representation is chosen since it requires less memory space. Then, each pixel is represented with 16-bit words instead of 24. Then the total requirement for one camera image frame is: $250 \times 250 \times 16$, which is slightly less than 1 Mbit of memory. Since there are 24 cameras in total, the memory need for keeping the frames is 24 Mbit. This result effects the choice of the FPGA system that is used for implementation of the image processing hardware.

5.2.1 Xilinx FPGA Board

The main specifications for the board is given in the Table-5.3.

Table 5.3 – Specifications of VC707 FPGA board chosen for embedded system and image processing hardware implementation

Specification	Value
FPGA Device	Virtex-7 XC7VX485T-2FFG1761C
Video Interface	HDMI
Debug Interface	JTAG
General Purpose Connectivity	FPGA Mezzanine Card (FMC) interface
Serial Interface	RS232
On-Board Memory Resource	1Gbit DDR3 SDRAM

The analysis for the memory need for image frame buffering resulted with at least 24 Mbit on-chip memory resources. The chosen board for implementation has the Virtex-7 XC7VX485T, which has the main resources given in the Table-5.4. The detailed internal architecture and specifications can be found by the manufacturer company [89]. As seen in Table-5.4, the total memory resources are 37 Mbit, providing enough space for implementation.

5.3 System Level Design Considerations

The system is targeted for Xilinx VC707 board. This board has an FMC connector that can be used for camera connections, RS232 serial port for user interface, HDMI output for video stream output and a dedicated I²C port. As the next step, the target is an ASIC design

Table 5.4 – Resource overview of Virtex-7 XC7VX485T FPGA chosen for embedded system and image processing hardware implementation

Specification	Value
Block RAM Resources	37 Mbit
Slices	75900
User I/O	700
DSPs	2800

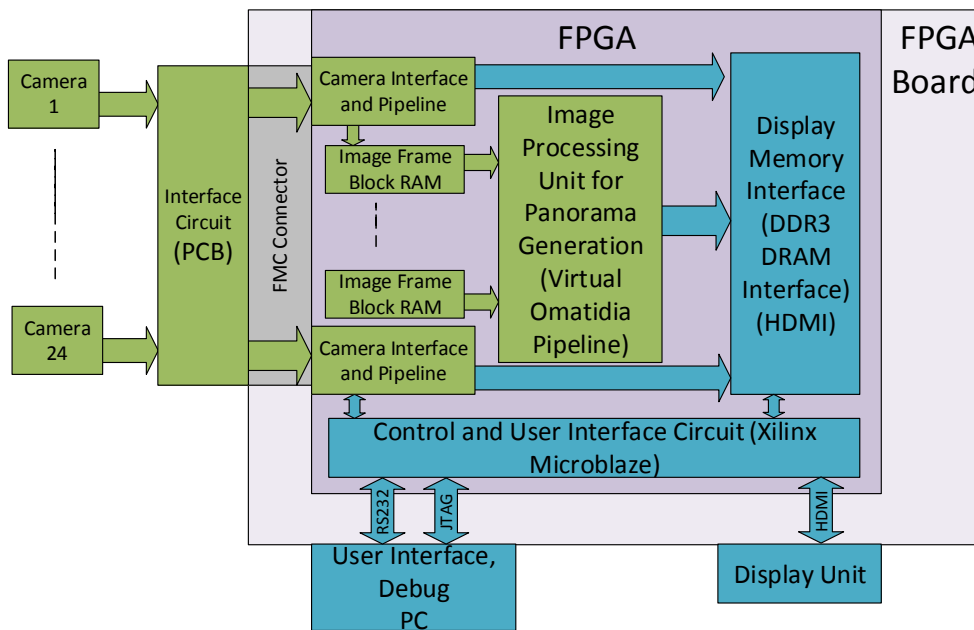


Figure 5.8 – The embedded system block diagram

that can be packaged with the miniaturized hemispherical head, so I tried to minimize the external components needed for the system. To that end, I use on-chip block RAMs as the camera frame memories which are accessed by the panorama pixel generation block unlike the implementations in [65, 66, 74]. For streaming out the single camera images and the panoramic image through HDMI, the DDR3 SDRAM available on the board is utilized. However, for the ASIC design, the aim is to use a simple parallel video interface similar to any conventional image sensor available on the market. The proposed system block diagram is shown in Fig.5.8.

5.4 Panorama Image Processing Hardware

There are different hardware approaches in literature for ray tracing based panorama generation methods [64, 67]. In [65] and [66], a central processing unit generates the final panorama by utilizing a pipeline structure. In [67], a distributed approach is reported, where

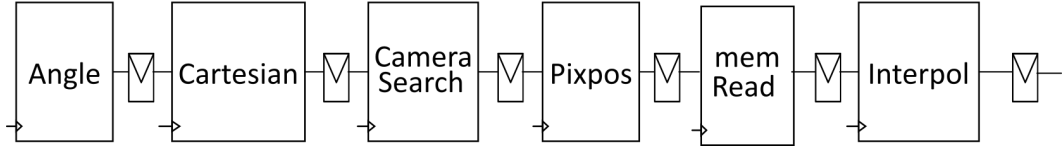


Figure 5.9 – The panorama pixel generation pipeline

each node has a camera connection and sends the pixel values to a central unit via a network on chip. Since we are targeting a miniaturized system which should be also affordable in ASIC design, I also utilize a pipeline structure as described in [64].

For each panoramic image, the pipeline is fed with the corresponding latitude and longitude angle, starting from the top left corner of the image. Then it calculates the Cartesian vector for that direction. Later it finds the candidate cameras looking in that direction by searching in all the cameras in the system where each camera check brings 1 clock cycle. This pipeline block diagram is shown in Fig.5.9. The sub-elements of the pipeline is described briefly below.

5.4.1 Pipeline Building blocks

Angle and Cartesian vector generation The generation of the angles are done by a simple accumulation units according to (5.11) through (5.14) [64]. Here each angle is a M-bit binary number. The maximum number of angles (N_θ, N_ϕ) that can be generated is then bounded by 2^M . The choice of M is dependent on the maximum resolution limit required. For resolutions $1080 \times 1080p$ or 1920×512 , $M = 11$ is sufficient. The increment of the angles is done by an increment factor of $\Delta\theta$ or $\Delta\phi$. By choosing $\Delta\theta, \phi$ and M , the number of total virtual ommatidia directions are determined.

$$\phi_i = \Delta\phi \times i, \quad \Delta\phi = \frac{2\pi}{N_\phi} \quad (5.11)$$

$$\theta_i = \Delta\theta \times i, \quad \Delta\theta = \frac{2\pi}{N_\theta} \quad (5.12)$$

$$\{\phi, \theta\} \equiv \{B_\phi, B_\theta\} \in [0, 2^{M-1}] \quad (5.13)$$

$$B = b_0 b_1 \cdots b_{M-1} = 2\pi \sum_{j=0}^{M-1} b_j 2^{-(j+1)} \quad (5.14)$$

Then another unit is responsible for generating Cartesian vector coordinates given by (5.15) which is trigonometrically equivalent to (5.16), (5.17) and (5.18). A block diagram for the angle and Cartesian vector generation unit is shown in Fig.5.10.

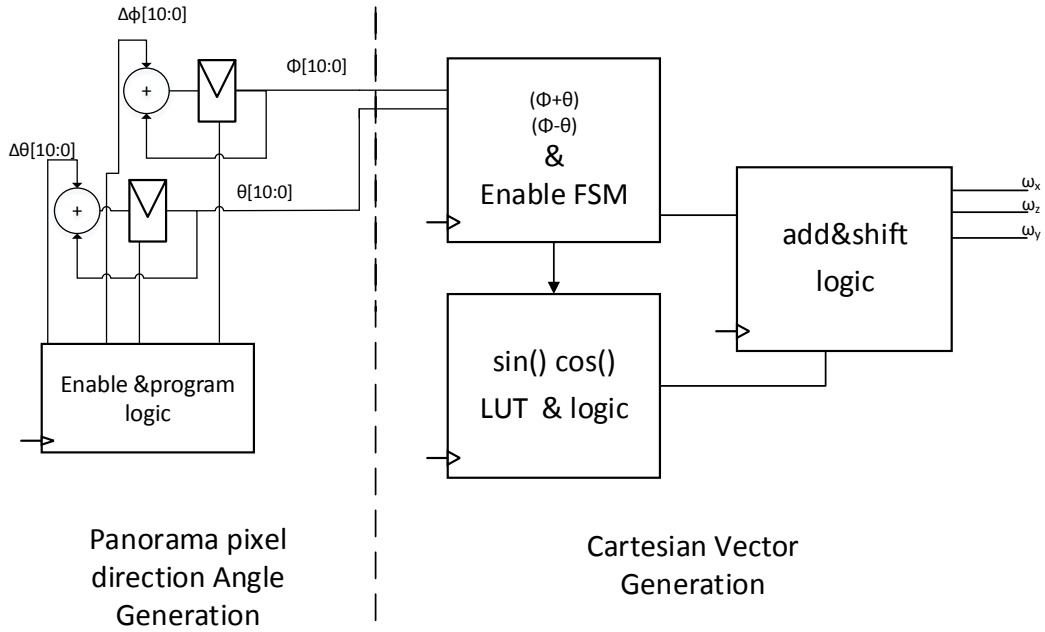


Figure 5.10 – The block diagram used for angle and Cartesian vector calculation

$$\vec{\omega} = \sin(\theta_\omega) \cos(\phi_\omega) \vec{x} + \sin(\theta_\omega) \sin(\phi_\omega) \vec{y} + \cos(\theta_\omega) \vec{z} \quad (5.15)$$

$$\omega_x = \frac{\sin(\theta_\omega + \phi_\omega) + \sin(\theta_\omega - \phi_\omega)}{2} \quad (5.16)$$

$$\omega_y = \frac{\cos(\theta_\omega - \phi_\omega) - \sin(\theta_\omega + \phi_\omega)}{2} \quad (5.17)$$

$$\omega_z = \cos(\theta_\omega) \quad (5.18)$$

Calculation of distance values and contributing cameras The calculation of the distance values at the orthographic plane of each direction is done according to (5.19) [64]. A simplified block diagram for distance calculation is shown in Fig.5.11. During the distance calculation the dot product of the direction vector $\vec{\omega}$ and the camera focal vector \vec{f} is calculated, so this value is also passed to the next unit for projecting the direction vector on to the camera coordinate system and estimating the position of the pixel on the camera image frame by using the pinhole camera relation [63]. A simplified block diagram is shown in Fig.5.11.

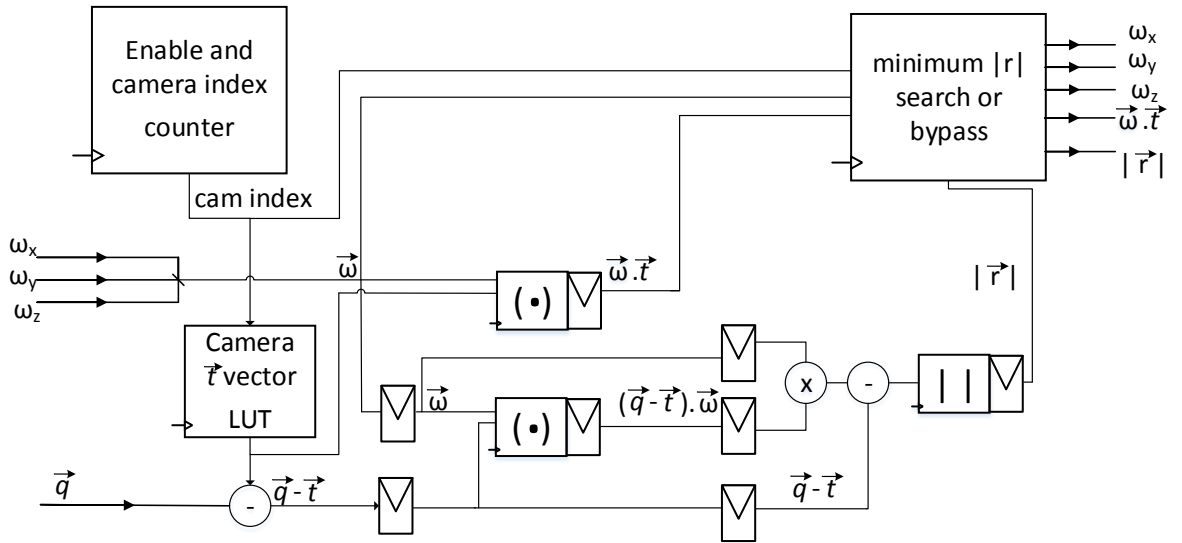


Figure 5.11 – The block diagram used for camera search and distance calculation

$$\vec{r}_i = (\vec{q} - \vec{t}_i) - ((\vec{q} - \vec{t}_i) \cdot \vec{\omega}) \times \vec{\omega} \quad (5.19)$$

$$d = |\vec{r}| \quad (5.20)$$

Pinhole camera approximation and camera image frame pixel position calculation For determining the position of the panorama pixel direction $\vec{\omega}$ on the camera image frame, the vector is first projected onto the camera coordinate system. This was done by making dot products of the $\vec{\omega}$ and the camera coordinate vectors $\{\vec{t}, \vec{u}, \vec{v}\}$. Then by using the pinhole approximation for projecting a point in space onto the camera image frame, the initial pixel positions are determined given by (5.21). However, since there is distortion of the optics, a correction is done according to (5.21). Here the k_i values are calculated at the calibration step. Also the camera image plane reference point is shifted from the top left corner $\{0, 0\}$ to the principle point $\{c_u, c_v\}$ which is calculated at the calibration step as well. After generating the pixel values an address generation is done to read the related pixel from the camera image frame buffer. The frames are kept in the memory linearly, meaning that they are written to the frame memory starting from address 0 of the memory starting from the line 1 and for each line starting from the pixel 1. So to read the pixel $\{X, Y\}$ from the corresponding memory, (5.27) is sufficient where l_w is the line width. Here X is the column index (pixel index in any row) and Y is the line index. In the hardware implementation a 16-bit precision is used for the variables in (5.21) to (5.27).

$$X_u = f_u \frac{\vec{\omega} \cdot \vec{u}}{\vec{\omega} \cdot \vec{t}} \quad (5.21)$$

$$X_v = f_v \frac{\vec{\omega} \cdot \vec{v}}{\vec{\omega} \cdot \vec{t}} \quad (5.22)$$

$$R^2 = X_u^2 + X_v^2 \quad (5.23)$$

$$p = k_5 R^6 + k_2 R^4 + k_1 R^2 + 1 \quad (5.24)$$

$$X_u' = p X_u + 2k_4 X_v X_u + k_3 (R^2 + 2X_u^2) \quad (5.25)$$

$$X_v' = p X_v + 2k_3 X_v X_u + k_4 (R^2 + 2X_v^2) \quad (5.26)$$

$$Addr = l_w(Y - 1) + X - 1 \quad (5.27)$$

Pixel Intensity value generation For pixel intensity value generation the pixel intensity values picked from the camera image frames are either selected with the nearest neighbor method [64] or interpolated according to the different methods proposed in [64, 66, 74]. For the interpolation the pixel values coming from the frame memory and the distance values from the distance generation module should come at the same time. Since the distance generation is performed before the intensity value reading, this can be achieved by using registers for the distance value. Then a reciprocal operation and accumulation of the distance values are performed to calculate the weighted average of the pixel intensity values of the contributing cameras.

5.4.2 Implementation for the probabilistic method for evidences

For the hardware implementation of the proposed method [74] described in Section-4.2.2, there is need to generate concurrently the $d = |\vec{r}|$ distance values of the 4 neighboring panorama pixels. These are up, down, left and right neighbors of the panorama direction. The distance values for these directions should be calculated for the cameras which are contributing to the central panorama pixel that is being generated in the pipeline.

To achieve this, in the angle generation unit the $(\theta_\omega, \phi_\omega)$, $(\theta_{\omega up}, \theta_{\omega down})$, $(\phi_{\omega left}, \phi_{\omega right})$ angles are generated in parallel. For the 4-neighbors, 4 more Cartesian vector generation units are added. In the camera search and distance calculation unit, the pipeline paths for calculating the distance values are replicated for additional distance values. Finally, the distance values are multiplied and the final evidence value for the direction being generated is obtained. A simplified block diagram for the hardware design is shown in Fig.5.12.

The hardware overhead added by the probabilistic method is dominated by the size of Cartesian vector generation unit and the distance generation operation. The resource usage

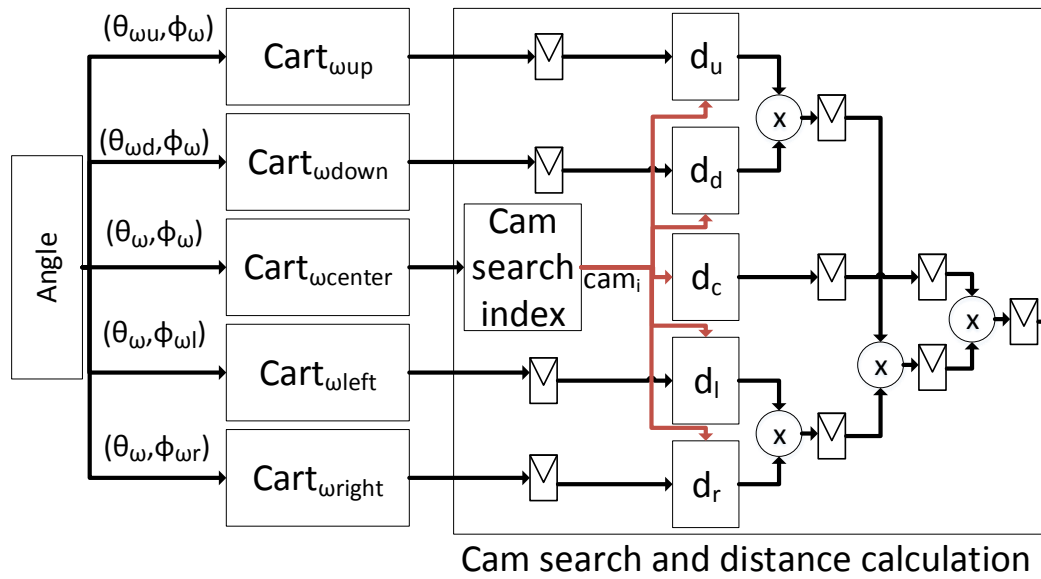


Figure 5.12 – The block diagram for implementation of the probabilistic evidence calculation

results of modules are given in the next section in Table-5.6. It can be observed from the result that the added hardware overhead with the proposed method is roughly around 4×100 slices by adding 4 more Cartesian vector calculation units.

From the synthesis results, the Table-5.5 is also extracted, which provides the resource usage information of the critical blocks. The hardware cost that will be added by the implementation of the probabilistic evidence calculation method will be $4 \times$ the hardware resources shown in Table-5.5.

In total the hardware overhead added with the method is 24 digital signal processing (DSP) blocks, roughly 4000 LUTs or 1600 Slices and 3600 Slice registers. By comparing this overhead with the total system resource usage, it is less than 1%. From this result, it can be said that when the quality increase achieved by the probabilistic evidence calculation is considered, the hardware overhead added to the design is negligible. The system implementation including the probabilistic evidence calculation method is described in the next section.

Table 5.5 – List of critical resource consuming blocks in the camera search and distance calculation module.

Module	Slices	Slice Reg	LUTs	DSP48s
Dot product	5	18	16	3
vector magnitude calculation	296	528	960	3
Pipeline registers	78	352	0	0

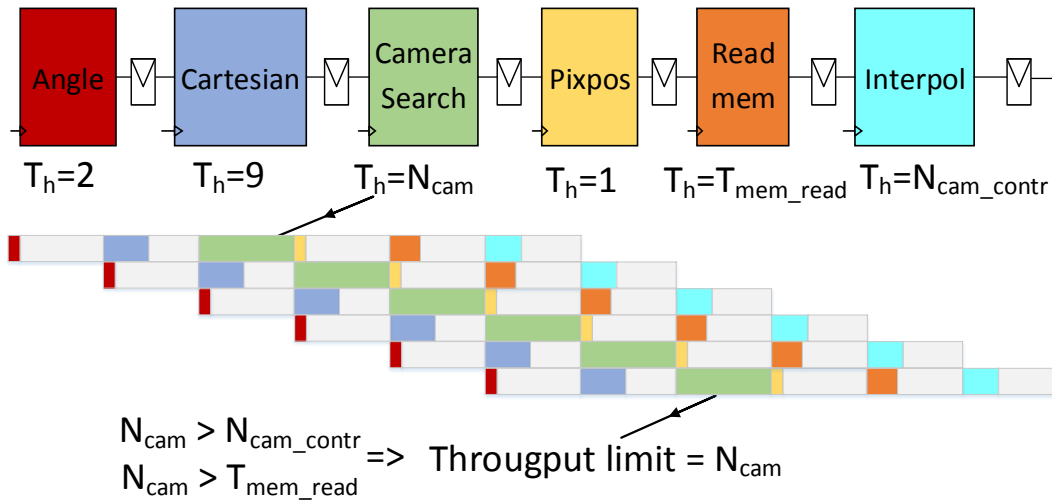


Figure 5.13 – The pipeline analysis of the system

5.4.3 System Implementation

For the pipeline design, a previous implementation methodology described in [64] is utilized. The building blocks for the pipeline implementation are described in Section-5.4.1. However, the maximum number of cameras were bounded to 20 in the previous approach. So the design of the pipeline for the 24 camera system is performed by retiming. Then the system is mapped onto the Virtex-7 FPGA device.

The pipeline structure is described in Fig.5.13. The throughput limit of each main block is shown as minimum required clock cycles which are needed to complete the operation and get ready for the next one. In Fig.5.13, each color is representing a different main block in the pipeline. For example the red colored unit is the angle generation unit and the minimum clock cycles required for that unit is $T_h = 2$ cycles. The throughput or the minimum clock cycles required to generate the intensity value of one pixel in the final panorama T_{hsys} is dictated by the largest value in the pipeline $\max\{T_h\}$.

Then assuming a 25 fps for real-time video requirement, for a $N \times M$ final panorama image the real-time operation clock frequency is defined as $F_{min} = 25 \times N \times M \times T_{hsys} \times 10^{-6}$ MHz. For example for a system with 20 cameras, to generate a 1024×1024 pixel panoramic video at 25 fps, a minimum of 525 MHz operation clock is required.

The first implementation in the scope of thesis work is done with the constraints above with a 100 MHz clock frequency. Then the obtained maximum frame size at 25 fps is observed as 166 Kpix, 408×408 video for $180^\circ \times 180^\circ$ representation and 730×182 video for $360^\circ \times 90^\circ$ representation. The hardware resources used on the Virtex7 FPGA is given in Table-5.6. In the first column of the Table-5.6 the sub-blocks of the panorama pipeline and the full system are

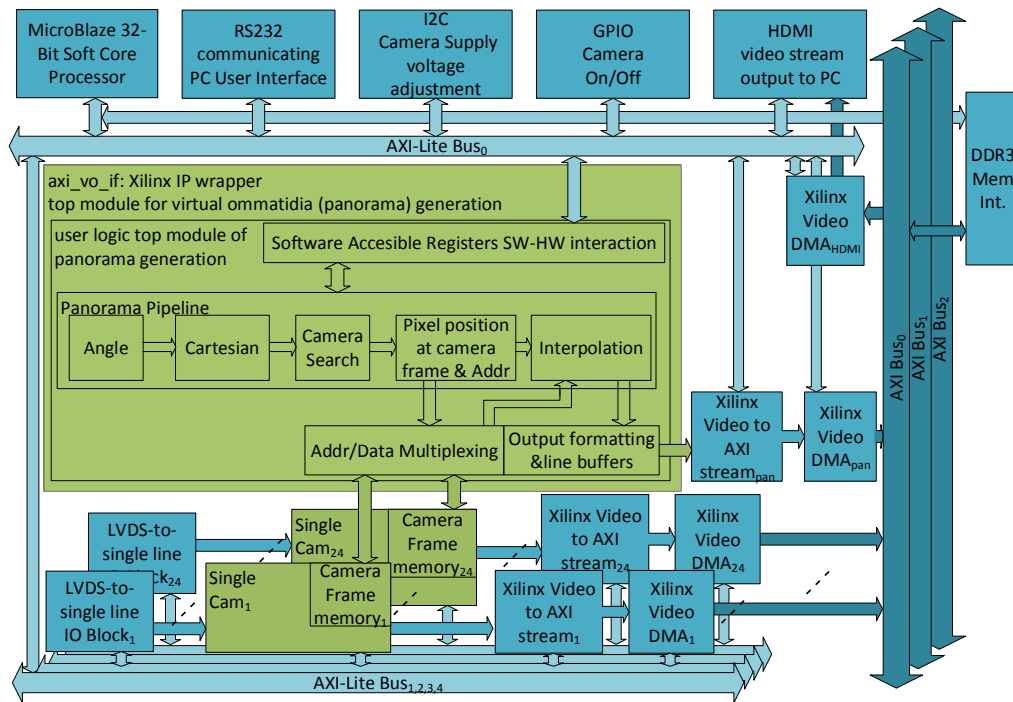


Figure 5.14 – FPGA system block diagram

listed; the second column shows the number of FPGA Slices used; the third column indicates the number of LUTs used; the fourth column indicates the number of block RAMs used; the fifth column indicates the number of DSP-slices used.

The designed system block diagram is given in Fig.5.14. In total, there are 5 AXI-lite buses in the system. The 24 camera interfaces are utilizing 4 AXI-lite buses and each of these AXI-lite buses are connected to 6 camera interfaces. Each AXI-lite bus have 32 bit data width. The interface between the panorama generation pipeline and output interface to stream out the panorama frame line by line is shown in Fig.5.15.

In the designed system, since the cameras used do not have the option to be fed with an external clock, there is no possibility for cycle accurate camera synchronization. Instead, we rely on the concurrent ON/OFF capability that is provided with a voltage regulator as described in the single camera interface design section. To do that the general purpose I/Os on the FPGA are used and controlled by the firmware on the Microblaze processor. Moreover, the cameras used have a frame rate between 44-56 fps, which is almost double frame rate of the panorama generated. This means the cameras are updating the frame memories nearly twice during a panorama frame generation, which also reduces the need for a cycle accurate synchronization.

The Microblaze firmware is developed in Xilinx software development kit (SDK) environment.

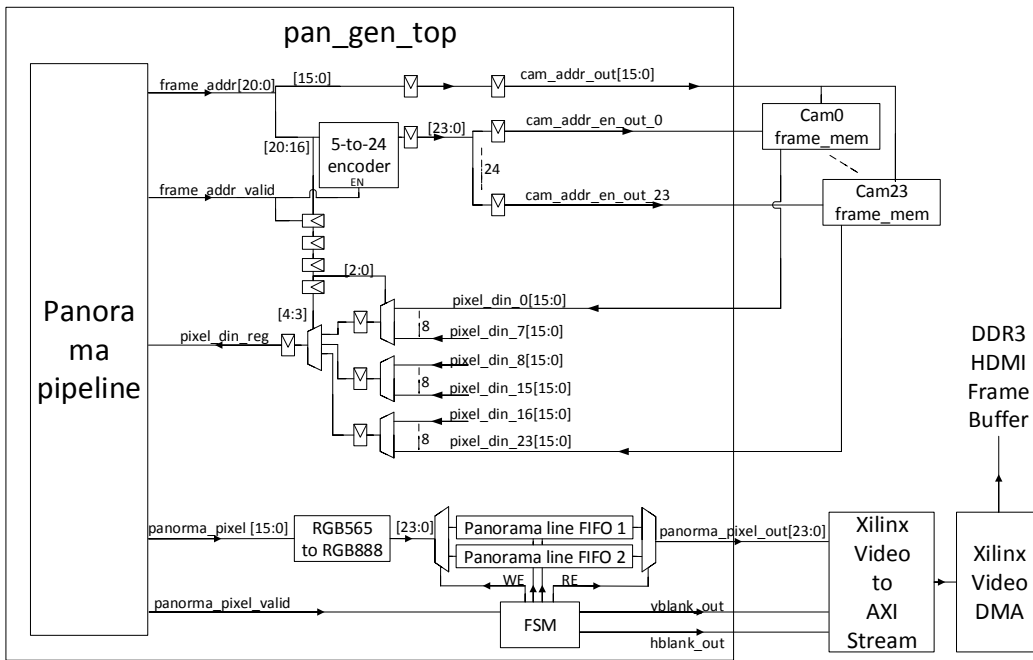


Figure 5.15 – Interface of the panorama generation pipeline with camera frame memories and video DMA output for DDR3 RAM

Table 5.6 – The resource usage results of the sub blocks and the full system on the Virtex7 FPGA for the first implementation

Module	Slices	LUTs	BRAMs (32Kbit)	DSP48s
Angle generation	62	167	4	0
Cartesian coordinate	85	177	2	0
Camera search	1496	4510	0	43
Pixel position	371	1038	0	27
Interpolation	232	667	4	16
Panorama Pipeline Total	2448	6928	10	86
Single Camera Int.	478	831	37	0
System total (with microprocessor, AXI bus and other peripherals)	58568	145619	946	92
FPGA device utilization	77%	47%	86%	3%

It is a single thread program with interrupt handling capability, written in C programming language and is utilized for the accessing the programmable registers and setting up the peripherals accordingly. The firmware also outputs messages to the user and accepts information from the user via RS232 serial I/O. The flow of the firmware is shown in Fig.5.16.

The methodology used during the design and tests of the sub-blocks and the whole system is summarized in the Fig.5.17. For each designed unit after generating the HDL code, a behavioral

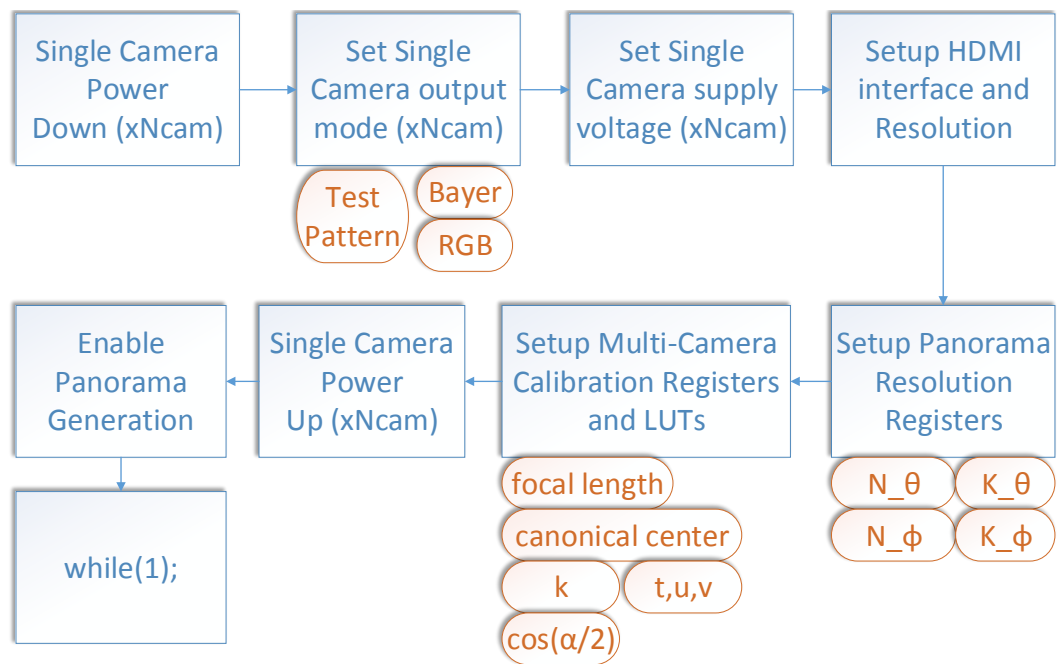


Figure 5.16 – The tasks and the flow of the Microblaze embedded processor firmware

simulation is done first in Modelsim environment, if the designed unit is generating an image or image derivative, the pixel intensity outputs are dumped to text files and converted to image files in MATLAB environment without any image processing. In this way the verification of the hardware design is done by comparing the output images to the images generated at the software level in MATLAB. Following this verification, the design is implemented in Xilinx integrated synthesis (ISE) and embedded development kit (EDK) environments for placing and routing and FPGA bit stream generation. The bit stream is downloaded to VC707 board and the video outputs are visually checked: and also downloaded PC via HDMI interface, HDMI-USB3 frame grabber without compression, and image processing.

5.4.4 Design for pipeline throughput increase

The theoretical limits for optimizing the pipeline can be described as follows: If any interpolation method is chosen other than the nearest neighbor, then even in all the blocks a parallelism methodology is applied and the T_h values are minimized, the pixel value calculation module will still need a throughput limit of $T_h = N_{camcontr} \text{ cycles}$, which is the maximum number of cameras contributing to any given panorama direction. The reason is that it accumulates the distance values and the weighted camera pixel values and finally makes the division operation.

If we look at the other modules, the most time consuming block is the camera search and distance calculation module. Once a new Cartesian coordinate set $(x_\omega, y_\omega, z_\omega)$ of a panorama direction is reached to the camera search module, it searches all the cameras in the system

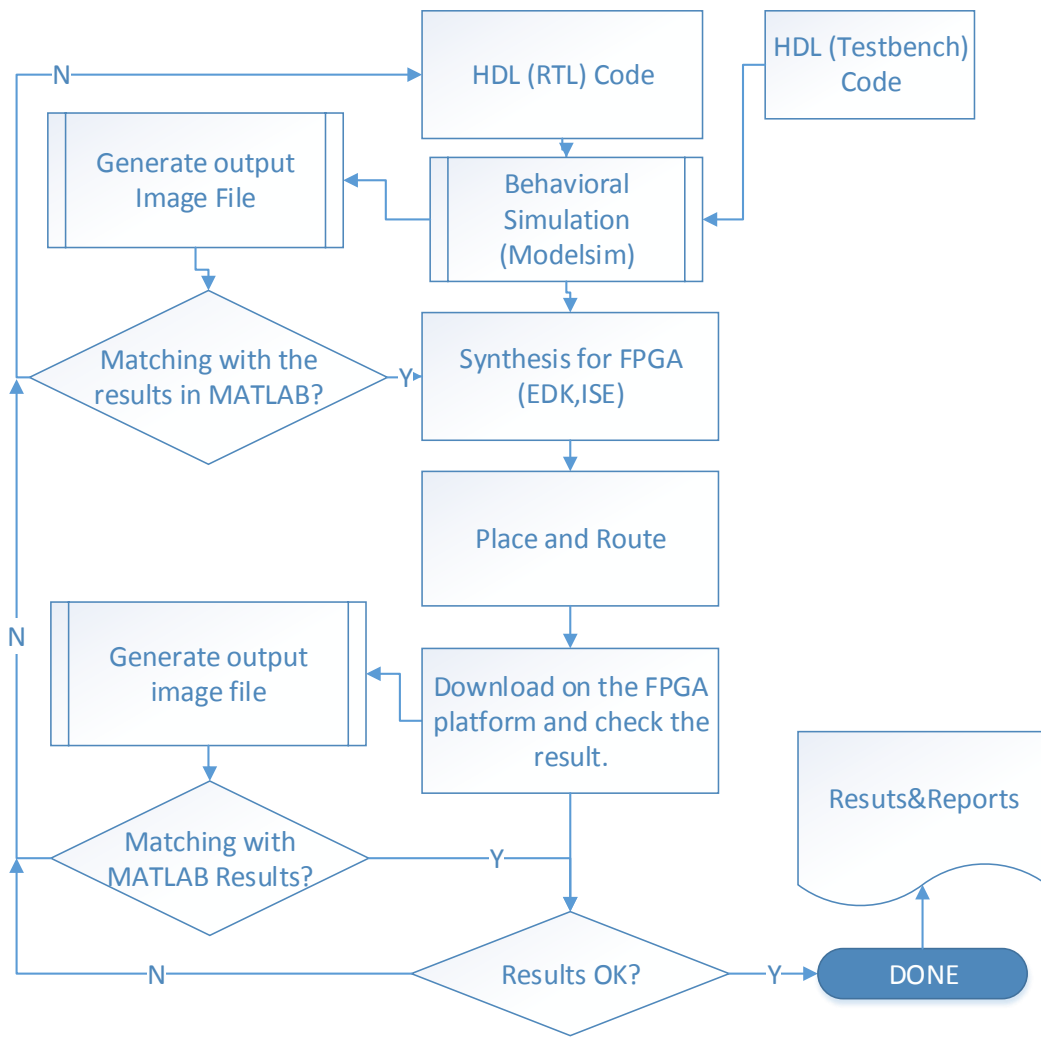


Figure 5.17 – The design and test flow methodology followed during development and tests.

and finds which cameras are contributing to the given direction. To do this, it accommodates a loop which is implemented by checking a camera index counter value as shown in the block diagram in Fig.5.11. For each camera index, it reads the \vec{t} value, performs a dot production $\vec{\omega} \cdot \vec{t}$ and compares the value with $\cos \frac{\alpha}{2}$ of the corresponding camera which is also read from a look-up table (LUT). If the camera is contributing to the given direction, then it propagates the distance calculation and camera index to the output for the following blocks in the pipeline.

In order to reduce the number of cycles spent on the camera search block, a parallelism is applied by utilizing a loop unrolling technique while checking the number of cameras seeing the particular direction. Instead of making the dot product operation for each camera in a loop of 24 cameras, I designed the camera index generation by utilizing 24 parallel dot product operations and 24-to-5 priority encoder to determine all the contributing cameras. In this way the number of cameras for the rest of the calculations are decreased to N_{cam_contr} cycles, which is equal to number of maximum contributing cameras to any particular direction. In this

way, the limiting throughput cycles for the camera search block is decreased to $T_{hcamsearch} = N_{cam_contr} \text{ cycles}$.

As an example, for the 24 camera system I designed, at any direction, there are at most 4 cameras are contributing to generation of the intensity value for that direction. Therefore, by this low level parallelism, a 6x frame rate increase can be achieved with a given clock frequency for the system. However, the Cartesian vector calculation block needs $T_{hcart} = 9$ clock cycles at least. The reason for that is, it reads the sin and cos values from a two LUTs, one for sin and one for cos, for the 3 angles given by (5.16)-(5.18) which are $(\theta_\omega - \phi_\omega)$, $(\theta_\omega + \phi_\omega)$ and (θ_ω) . For each read&calculate operation it spends 3 clock cycles. To reduce this time, 3 LUT sets are accommodated and all the sin and cos values are read in parallel. In this way the $T_{hcart} = 3 \text{ cycles}$ is achieved. The single camera frame memory access time can be another bottleneck, especially for the systems utilizes external memory components. However, the internal BRAMs are utilized in the design proposed here, which reduces the memory access time to 2 clock cycles with a pipelined read-write logic included.

The resource usage difference from the previous implementation with the increased throughput pipeline is given in Table-5.7. In the pipeline, the most resource consuming part is the camera search unit. With the added 23 dot products there is more than 3x increase in the number of DSPs utilized. The reason for this is there are multiplication operations in the dot product unit. Also with the 24-to-5 encoding and camera search indexing logic, the occupied slices and LUTs for the camera search unit are almost doubled. For the current design, this overhead is affordable when compared to the rest of the system resource usage and when the number of remaining unallocated resources are considered. In this design, the panorama generation unit is synthesized for a 120 MHz clock. Then it is capable of generating $180^\circ \times 180^\circ$ panoramic video frames with 1080×1080 pixels resolution at more than 25 fps (25.7 fps precisely). The current limitation is coming from the DDR3 memory bandwidth. Since the HDMI core and the panorama generation units are accessing the DDR3 memory at the same time, the AXI bus and the DDR3 memory controller throughput is decreasing. The measured video output frame rate from the HDMI interface is 24 fps.

However, this approach is utilized for decreasing the number of clock cycles needed for generation of a panorama pixel from 24 to 4 and have limitations. As described above, here the 24 is the total number of cameras and 4 is number of contributing cameras. Therefore, for the systems with more total number of cameras and more contributing cameras, a trade-off should be considered between the resource usage and throughput increase. In the end, the available resources of the utilized FPGA platform and the real-time specifications of the target applications will be the two factors, leading to the decision. For the frame rate objective of 25 fps and high resolution as 1080×1080 pixels, with the chosen FPGA platform and the designed multi-camera apparatus, this design decision is valid and effective.

Table 5.7 – The resource usage results of the sub blocks and the full system on the Virtex7 FPGA for the increased throughput pipeline

Module	Slices	LUTs	BRAMs (32Kbit)	DSP48s
Angle generation	62	167	4	0
Cartesian coordinate	85	177	2	0
Camera search	2662	8168	0	136
Pixel position	371	1038	0	27
Interpolation	232	667	4	16
Panorama Pipeline Total	3614	10586	10	179
Single Camera Int.	478	831	37	0
System total (with microprocessor, AXI bus and other peripherals)	59849	149593	964	185
FPGA device utilization	77%	47%	86%	3%

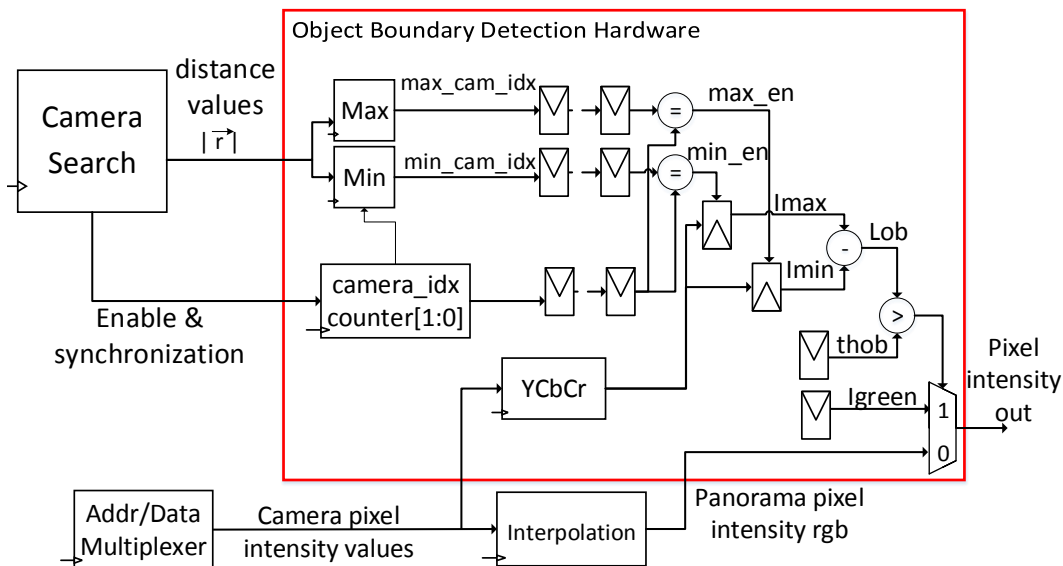


Figure 5.18 – The proposed hardware for object/poly boundary detection

5.4.5 Hardware implementation for object boundary detection

In this Section, the hardware implementation of the image processing method presented. The image processing method is described in section-4.3.1. For the hardware implementation, the FPGA system described in previous sections is utilized.

In the panorama generation pipeline, the camera indices and the corresponding distance values are generated cycle by cycle in the camera search module. Hence, the maximum and minimum distance calculation blocks are designed in a single-pass fashion. That is, the *Max(Min)* calculation block updates the *max_cam_idx(min_cam_idx)* if it detects a new greater (smaller) distance value. This search operation is done for each panoramic direction

and reset at the beginning of each new direction supplied to the pipeline. The *Max(Min)* calculation block outputs the result synchronous to the last generated camera index and distance value. Since there is the pixel position generation module in between the camera search and interpolation blocks, the resulting *max_cam_idx(min_cam_idx)* values are pass through a register chain to be aligned with the latency of the pixel position and address generation and individual camera pixel value fetching operation. Then in parallel with the intensity interpolation, the corresponding intensity values from the contributing cameras coming from the address-data multiplexer, are issued in the same order they have generated from the camera search module. So the camera indexes for maximum and minimum distances are indexed between [0-3], since there are 4 cameras can be contributing to any given direction. In parallel to the RGB intensity value generation, the incoming intensity values of the cameras are converted to gray scale first for making the difference operation. Since the maximum and minimum camera indexes are already received, as the corresponding intensity values received, the difference of the max-min distance pixel intensity values are calculated by a subtraction operation and fed to the output. For visualization, the RGB value and the corresponding difference value are multiplexed at the output. At this output stage, the thresholding is applied and a marker added to the object boundaries as a constant pixel intensity value. A constant green value is chosen since the dominant color is in red tone for the colon images.

Implementation results

Hardware resource overhead added with the implementation of the method is shown in Table-5.8. The hardware is synthesized for Virtex-7 FPGAs and can support up to 120 MHz operation. In this way, it can support the real-time operation for 25 fps 1080 × 1080 pixel output resolution for a panoramic video represented as 180° × 180° as described for system in previous sections.

Table 5.8 – Virtex-7 FPGA resource usage overhead for object boundary detection method

Resource Type	Total Count	Utilization ratio
Virtex-7 Slice LUTs	65	< 1%
Virtex-7 Slice registers	120	< 1%
Total Occupied Slices	48	< 1%
BRAMs	0	
DSP48s	0	

An example output of the hardware is shown in Fig.5.19. It can be seen that around the polyp structure close to the right down corner, the boundary detection method provides indicator pixels interpreted with green color. The green pixels on the other parts are due to calibration errors and lack of color calibration among the 24 cameras.

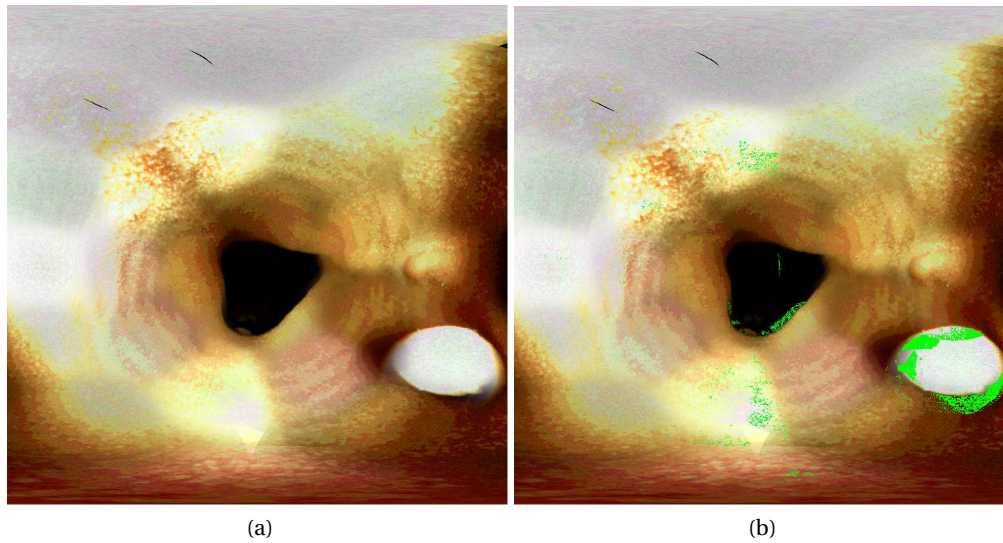


Figure 5.19 – An example of the hardware generated output image for colon boundary detection. (a) the $180^\circ \times 180^\circ$ panoramic image in 1024×1024 resolution (b) the possible boundary regions are marked with green color by the hardware described.

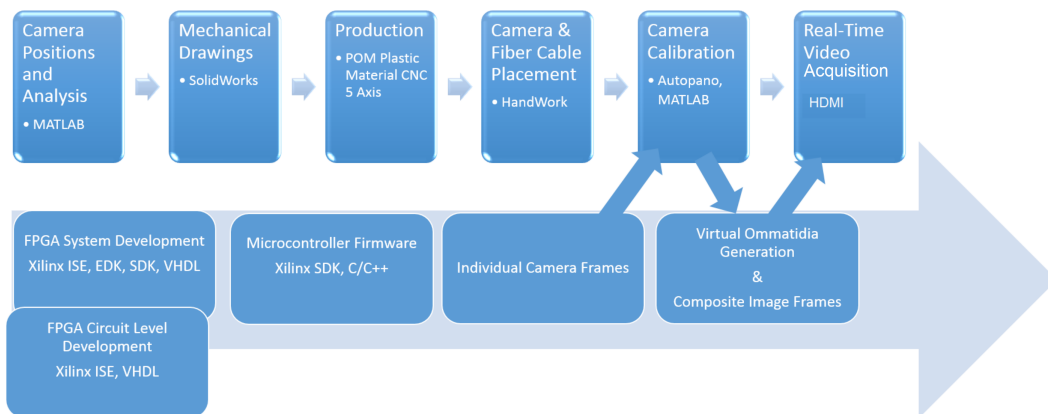


Figure 5.20 – The complete prototyping chain used for the implementation of system

5.5 Experiments and Results

The whole prototyping chain with the used tools is summarized in Fig.5.20. The system built is shown in Fig.5.21 with the 24 cameras hemispherical tip, cabling, Xilinx evaluation FPGA board and the in-house designed custom interface PCB. The human colon model used during the experiments is also shown next to the system in Fig.5.20. In the following subsections, the experimental setup with all of its components and the results obtained from the experiments are described.

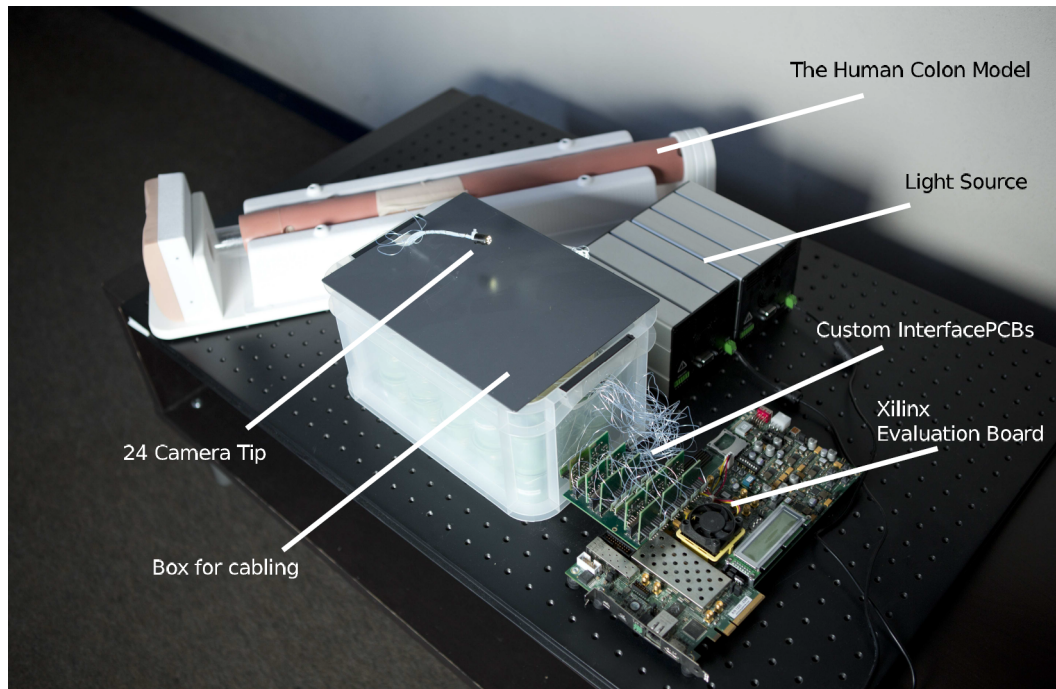


Figure 5.21 – The complete system built and used for experiments

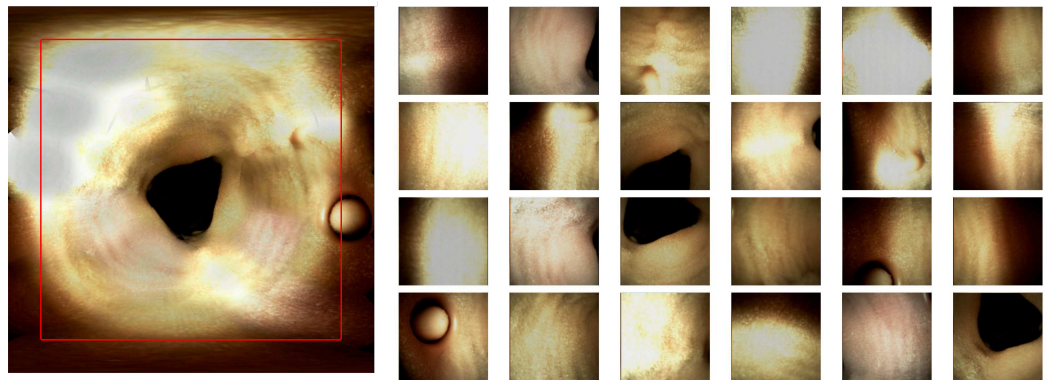


Figure 5.22 – An example of the output image from the system

5.5.1 Visual Results

In Fig.5.22, an example output from the realistic colon model is shown. The single 24 camera streams and the compound panoramic image can be seen together in this figure. As seen on the $180^\circ \times 180^\circ$ compound panoramic image, the polyps at the sides, even behind the folds of the colons can be easily captured by the system designed. Video links demonstrating this case is also provided on the Appendix-C.

5.5.2 Efficiency of the system size

In [19], it is shown that for diffraction limited compound eyes, the relation of the eye radius, the acuity parameter $\Delta\phi$ and the light wavelength λ is given by (5.28).

$$R = \frac{\lambda}{2\Delta\phi^2} \quad (5.28)$$

By using (5.28), it is possible to estimate the expected ideal radius of curvature of a compound eye by using the sampled wavelength and the given inter-ommatidial angle of the compound eye. For example given inter-ommatidial angle $\Delta\phi$ of the systems in [36] and [37], the expected radius of the defined compound eyes should be 0.06 mm and 0.046 mm respectively. Both systems are realized as 6 mm radius nearly. From this point, we can define a size efficiency parameter for the design and implementation of compound eyes given by (5.29).

$$\zeta_c = \frac{R_{imp}}{R_{ideal}} \quad (5.29)$$

In (5.29), the R_{imp} is the empirically measured or fabricated radius and the R_{ideal} is the estimated value by using (5.28). The system is implemented at 5 mm radius for our 24 camera prototype. I have measured the acuity by using USAF 1951 line pair pattern as 0.0087 radians (0.5°). Hence as (5.28) indicates, we have an expected radius of 3.3 mm. In other words, if such a system exists in nature, it should be ideally sized around this radius of curvature for the compound eye. An example image captured at 1024x1024 resolution, at a 25 mm distance from the hemisphere center with USAF 1951 line pair pattern is shown in Fig.5.23. The 1951 USAF resolution test chart is a resolution test pattern set by US Air Force in 1951. The pattern consists of groups of three bars with dimensions from big to small. The largest bar pair that cannot be distinguished by the imaging is given as the limitation of the systems resolving power. For example for my system, the smallest resolvable line pair was group 1, element 2 at 25 mm distance. This measurement gives a resolution power of 0.5° .

Our system has a physical radius of 5 mm. However, there is a need for certain distance from the real cameras so that their angle of views can intersect and cover the whole hemispherical field of view. We defined this virtual radius by our camera placement as 18 mm, where the virtual ommatidia views can appear without any space between them. So we have an efficiency of 18% for our implementation.

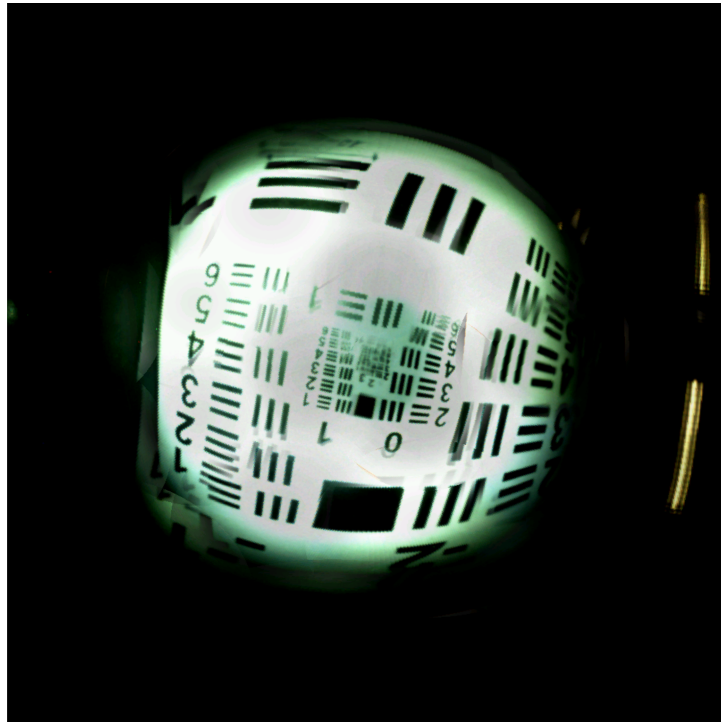


Figure 5.23 – An example image for the USAF 1951 measurements, the image is captured in 1024x1024 resolution, at 25 mm distance from the hemispherical camera center.

5.5.3 Comparison with Different Insect Eye Based Systems

In the Table-5.9, we compare our system with the curved lens array based compound eye systems and with the natural compound eyes. In this comparison, the main criteria are size, field of view and resolution. Table-5.9 shows that, our system brings a significant resolution increase of 1000x when compared to the systems which have even larger radius of curvature. As seen on Table-5.9, our system is closer to natural systems in terms of resolution/size efficiency, which is very important for miniaturized wide FOV imaging especially for applications like colonoscopy [56].

5.6 Conclusion

In this Chapter, the real-time hardware design methodologies proposed for generating high quality, high throughput panoramic video streams from the miniaturized multi-camera system designed in the scope of this thesis. A hardware implementation for the image processing method is described, which is intended for reducing the seams and object boundary ghosting effects in the reconstructed panoramic image frames. When the image quality added with the probabilistic evidence calculation method is considered, the hardware overhead added is below 1%, which is negligible. A method for increasing the frame rate by increasing the throughput of the pipeline design is explained. The limitations of the method is also discussed.

Chapter 5. FPGA Embedded System Design

Table 5.9 – Comparison with insect eye systems in terms of resolution and size

Approach	Size (Radius of Curvature) (mm)	Number of Ommatidia (pixels)	FOV	$\Delta\phi$	$\Delta\rho$	R_{ideal} 500nm wavelength (mm)	Additional Features
Dragonfly [71]	5-6	30000	$180^\circ \times 160^\circ$	0.24°		5-6 $\zeta_c=100\%$	
[36]	6-7	180 (12×15)	$360^\circ \times 80^\circ$	11°	9.7°	0.06 $\zeta_c=1\%$	
[37]	6.4	630 (42×15)	$180^\circ \times 60^\circ$	4.2°	4.2°	0.046 $\zeta_c=0.7\%$	Adaptability to illuminance Crosstalk prevention
This work	5	1600×400 817×817 (≈ 640000)	$360^\circ \times 90^\circ$ $180^\circ \times 180^\circ$	0.5°	0.5°	3.3 $\zeta_c=18\%$	Distributed omnidirectional Illumination

The FPGA implementation and the results of the hardware design are disclosed. The general principle for the methods is keeping the miniaturization goal as achievable by an ASIC design which is the subject of the next Chapter.

6 ASIC Design for Miniature Insect Eye Inspired Image Processing System

6.1 Introduction

The development of semiconductor technologies in the past 50 years raised the need for electronics miniaturization. The ASIC implementations show an advantage in terms of cost for large number of production units, especially when it comes to the camera design and image processing domain. For instance, in the smartphone industry, using huge and high cost FPGAs is not attainable to execute special image processing functions. Instead there are many different types of image processors are developed for this purpose, usually named as Image Signal Processor (ISP).

The reduction in the cost is a requirement, especially in the endoscopic imaging domain, therefore, it is essential to utilize a single chip processor. In addition, this kind of certain applications need miniaturized image processors and imaging solutions such as capsule endoscopy. In Fig.6.1, an illustration is given for the capsule type endoscopes and their general assembly layers.

In the current chapter, the ASIC migration is explained in detail from the FPGA-based system described in the previous chapter. Primarily, some constraints based on the mechanical size of the designed multi-camera imaging system and aforementioned applications are defined. Later, the method used for FPGA to ASIC migration is described; and the fundamental changes to make for ASIC implementation is explained. Finally, the results of the implementation are given.

6.2 ASIC Specifications

This section encompasses the specification of the designed ASIC in terms of dimensions, I/Os and memory analysis.

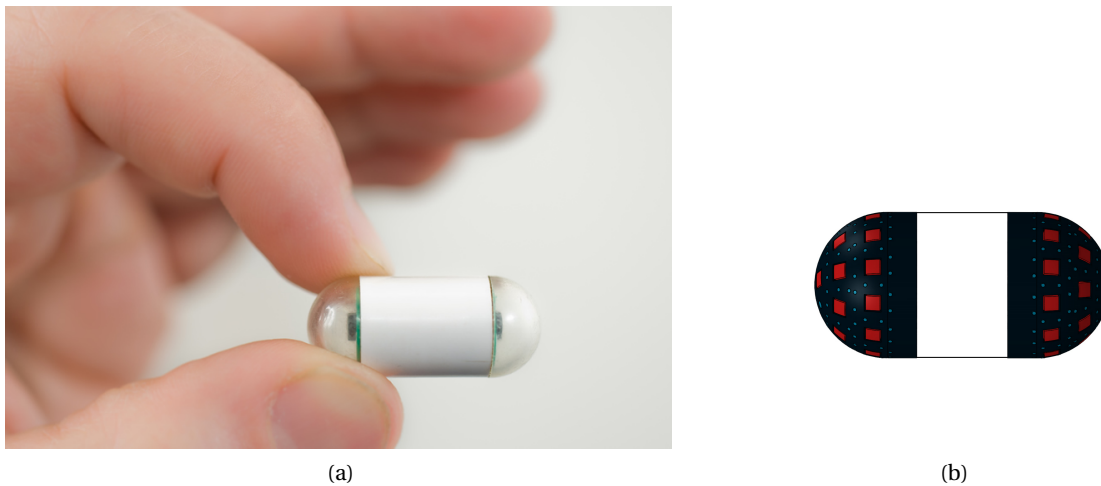


Figure 6.1 – Capsule Endoscopy concept with miniaturized insect eye imaging system

6.2.1 Design Constraints

The ASIC design, as a final goal, also aims to realize a compact design and to provide panoramic images to the outside world as though it were a single camera. Since the applications like endoscopy enforce a constraint on the design dimensions, a hemispherical tip of 10 *mm* in diameter has been defined. The final chip that is expected to fit in a 78.5 *mm*² circular area is illustrated in Fig.6.2. As shown in figure, a 7 *mm* × 7 *mm* = 49 *mm*² ASIC area is set as a target, therefore leaving around 30 *mm*² empty to be used for the fiber optic channels. The total area occupied by these fibers is less than 7 *mm*², so there is enough space left to connect them to a LED or a different kind of light source. Finally, by bonding wires or by additional PCB where the ASIC is placed on, the single cameras and the chip can be integrated.

6.3 I/Os Analysis

Fig.6.3 shows the top level and block diagram of the ASIC. It takes input as the 24 serial streams of data coming from the different cameras (*serial_data_in*), de-serializes and converts them in RGB values through the camera interface and send them to the panorama generation unit (Fig.6.4), which outputs the 24-bit RGB pixel values of the resulting panoramic image (*data_out*[29:0]). The data length is kept as 30 bits to keep the interface general and compliant with different standard interfaces for future use.

An I²C bus (see Section 6.4.5) is present to control from an external microcontroller the values of some configuration registers inside the chip. Due to a need for a full custom design, the I/O for this module has not been developed in this ASIC version. Instead, two additional pins are supplied: *I2C_data_out* to grant the communication from the ASIC to the microprocessor and *I2C_direction* to specify the communication direction

Four other signals are available at the output to generate the panoramic video on a display via

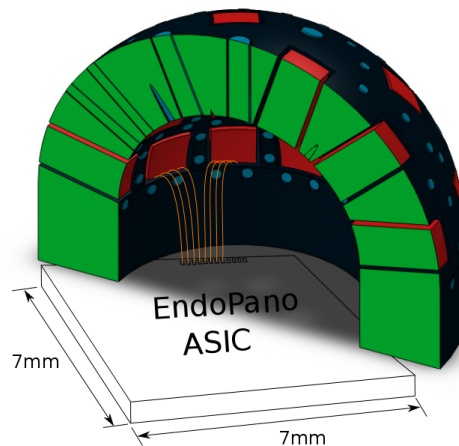


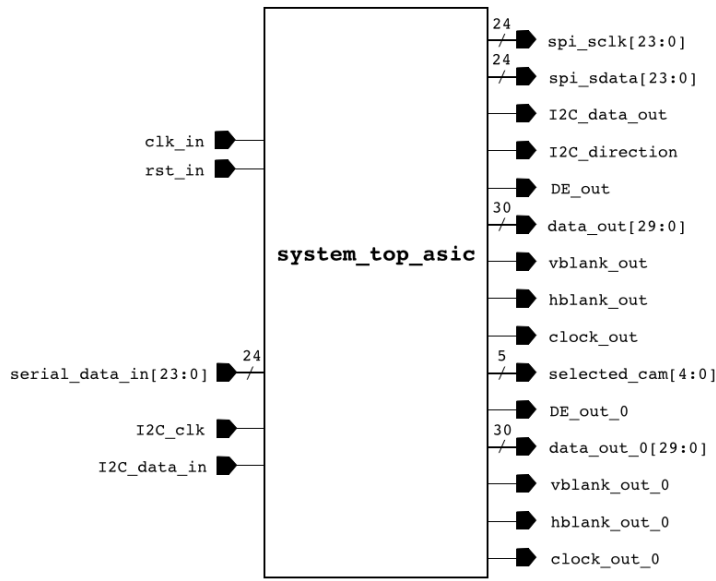
Figure 6.2 – ASIC design conceptual illustration with EndoPano imaging tip.

raster scanning (Figure 6.5):

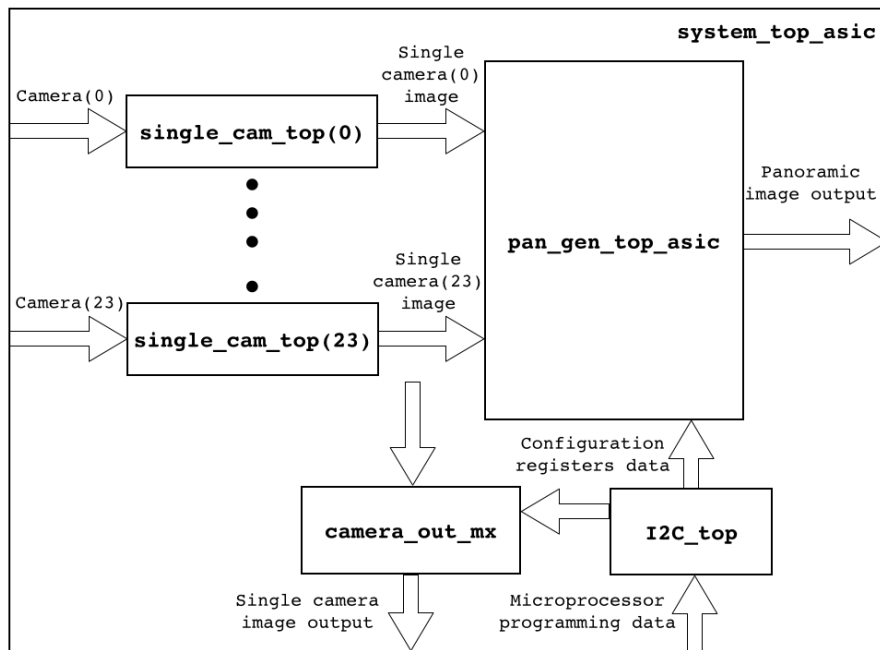
- clock_out is asserted to logic 1 each time a new pixel value is available at the output.
- vblank_out is deasserted to logic 0 between the beginning of the first line and the end of the final line of a frame.
- hblank_out is asserted to logic 1 between the end of a line and the beginning of another one. Hence in one period of vblank_out N periods of hblank_out are present, where N is the number of lines.
- DE_out is the active line length, used to specify the actual dimensions of the frame to display. If the width of its pulse is the same as hblank_out, the whole frame is displayed while if it is narrower a lower number of active pixels are selected for each line, thus producing a smaller image.

Owing to an output multiplexer the same outputs are accessible also from a single camera, with two different purposes:

1. At the beginning, to obtain a picture from the same static frame for each camera in order to calibrate them.
2. During the operation, to display the video from a desired camera.



(a)



(b)

Figure 6.3 – ASIC design top level (a) and block diagram (b)

The two signals `spi_sclk` and `spi_sdata` are utilized to update a configuration register present in each camera during data interface upstream. In this version, these outputs are available on separate pins since, as for the I²C, the LVDS I/O has not yet been implemented.

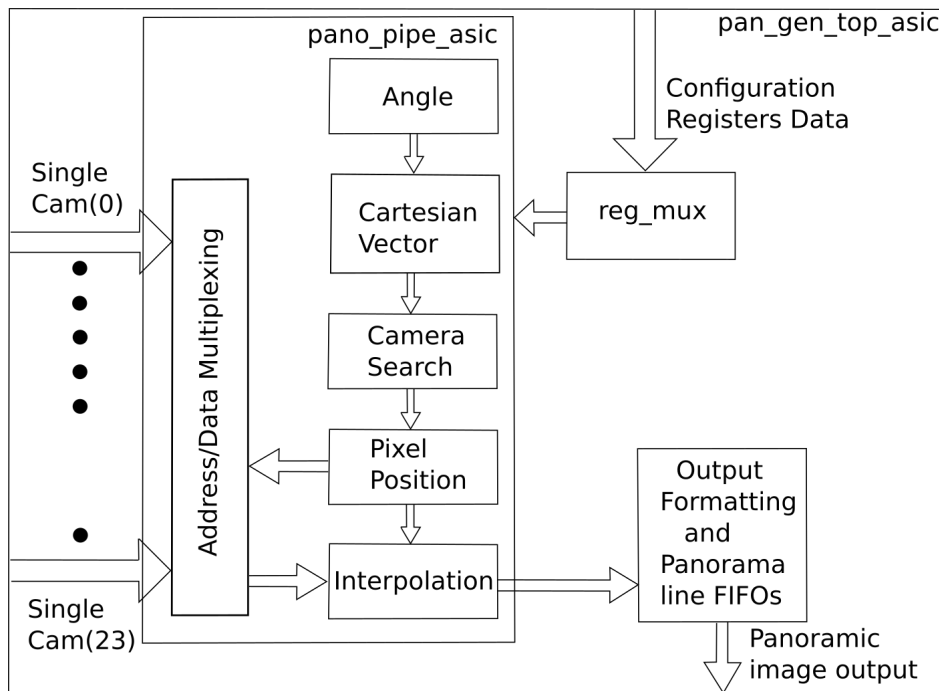


Figure 6.4 – Panorama generation block diagram.

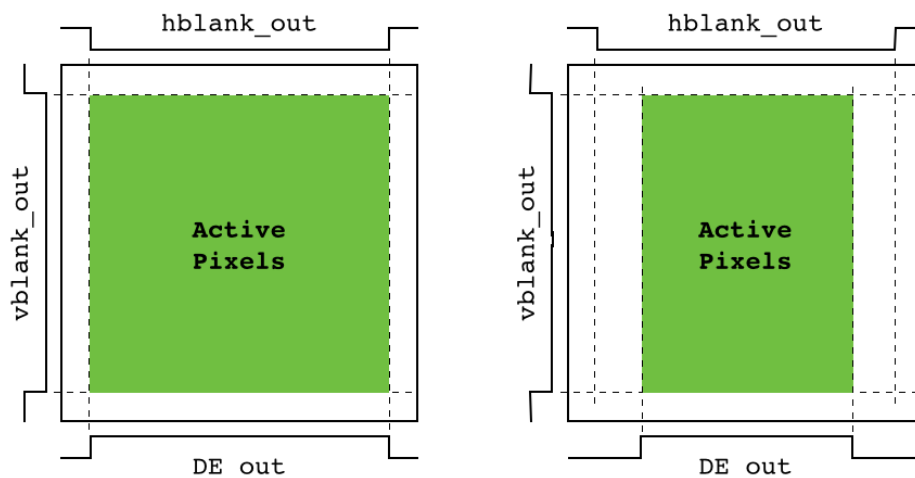


Figure 6.5 – Raster scanning principle for different DE_out pulse width.

Tables 6.1 and 6.2 report all system inputs and outputs respectively.

A list of all the pins essential for this design is displayed in Table 6.3. It is seen that the large amount of pins can be slightly reduced designing the LVDS and I²C I/Os.

6.3.1 Memory Analysis

Different types of memory elements are present in the system operating on the Virtex-7 FPGA:

Table 6.1 – System inputs for ASIC design

Signal	Description
clk_in	System clock
rst_in	System reset
serial_data_in[23:0]	Serial data from 24 cameras
I2C_data_in	I ² C serial input bus
I2C_clock	I ² C clock bus

Table 6.2 – System outputs for the ASIC design

Signal	Description
spi_sclk[23:0]	Camera configuration clock
spi_sdata[23:0]	Camera configuration data
I2C_data_out	I ² C serial output bus
I2C_direction	I ² C data direction
DE_out	Data enable
data_out[29:0]	RGB pixel value
vblank_out	Vertical blanking interval
hblank_out	Horizontal blanking interval
clock_out	Pixel clock
selected_cam[4:0]	Selected camera
DE_out_0	Selected camera data enable
data_out_0[29:0]	Selected camera RGB pixel value
vblank_out_0	Selected camera vertical blanking interval
hblank_out_0	Selected camera horizontal blanking interval
clock_out_0	Selected camera pixel clock

- Dual-port Block RAMs for each FIFO.
- Dual-port Block RAMs for each camera to store the incoming frame.
- Single-port Block RAM to update values for vignetting correction.
- Look-Up Tables implemented as distributed RAM to store initialization values for vignetting correction and sine and cosine functions values.

The memory elements in the FPGA hardware have been replaced with existing ASIC memory components: ROMs, single-port SRAMs and dual-port SRAMs from **ARM** which have been chosen, and designed by using the **TSMC 40nm CMOS process**. To compile all of these embedded memories, ARM Artisan Physical IP software has been used. An example view from the ARM Artisan Physical IP software is given in Appendix-B. The tool generates all the files

Table 6.3 – Possible pins configurations for ASIC design

ASIC pins	initial	with I/Os
clk_in	1	1
rst_in	1	1
serial_data_in[23:0]	24	48
I2C_data_in	1	1
I2C_clock	1	1
spi_sclk[23:0]	24	0
spi_sdata[23:0]	24	0
I2C_data_out	1	0
I2C_direction	1	0
DE_out	1	1
data_out[29:0]	30	30
vblank_out	1	1
hblank_out	1	1
clock_out	1	1
selected_cam[4:0]	5	5
DE_out_0	1	1
data_out_0[29:0]	30	30
vblank_out_0	1	1
hblank_out_0	1	1
clock_out_0	1	1
Total number of pins	151	125

needed for synthesis and place and route: Verilog models, Synopsys models for each corner and LEF footprints.

A list of the compiled memories together with their specifications available on-chip is illustrated in Table 6.4.

6.4 FPGA to ASIC Conversion

In this section, an overview of all of the modifications done so as to transfer the code already working on FPGA to ASIC are described. Since the aim is to keep the design minimal, most of the external components related to the FPGA design is removed. Instead, some custom blocks are designed such as FIFOs, I²C interface and memory wrappers for embedded memories.

6.4.1 FIFO Design

Beyond the buffering capabilities, First In First Out (FIFO) buffers are commonly used in digital systems for two particular applications:

Table 6.4 – List of ASIC memory elements

Type	Count	Words	Bits	Size(KBytes)	Area(mm ²)
Dual Port SRAM	2	2048	16	8.192	0.080
Dual Port SRAM	48	256	24	36.864	1.286
Dual Port SRAM	48	1024	10	61.440	0.921
Dual Port SRAM	24	256	10	7.680	0.313
Dual Port SRAM	96	8192	32	3'145.728	22.681
Single Port SRAM	2	4096	16	16.384	0.066
ROM	30	2048	16	122.880	0.251
ROM	1	8192	16	16.384	0.022
Total				3'415.552	25.620

1. Matching between modules producing and consuming data at different data rates.
2. Data transfer between unsynchronized clock domains. Such a clock difference might be intentional, like in multiple clocks designs, or caused by jitter and skew in large circuits.

The necessity of designing customized FIFOs comes from the need to be technology and platform independent, avoiding using the proprietary software provided by the FPGA vendor to generate these IPs. In Table 6.5, a list of all the FIFOs that has been replaced is shown. Both single-clock and dual-clock are present to be used as simple buffers or as clock matching respectively.

Table 6.5 – FIFOs present in the ASIC design

FIFO	Depth	Word size	Type
panorama_line_fifo	2048	16 bit	Single-clock
rgb_line_fifo249	256	24 bit	Single-clock
row_fifo249	1024	10 bit	Single-clock
cam_int_fifo	256	10 bit	Dual-clock

6.4.2 Dual-Clock FIFO

In order to implement the cam_int_fifo included in the camera interface module, a dual-clock, asynchronous FIFO has been developed. This necessity of using a dual-clock FIFO comes from the fact that each camera feeds its interface with a serial stream of data at 200 MHz. However, a new valid pixel intensity value is available only after 10 clock cycles meaning with a frequency of 20 MHz. Thus, working at such a high frequency is not a requirement neither at the consumer side. The design details for the Dual clock FIFO is presented in Appendix-B.

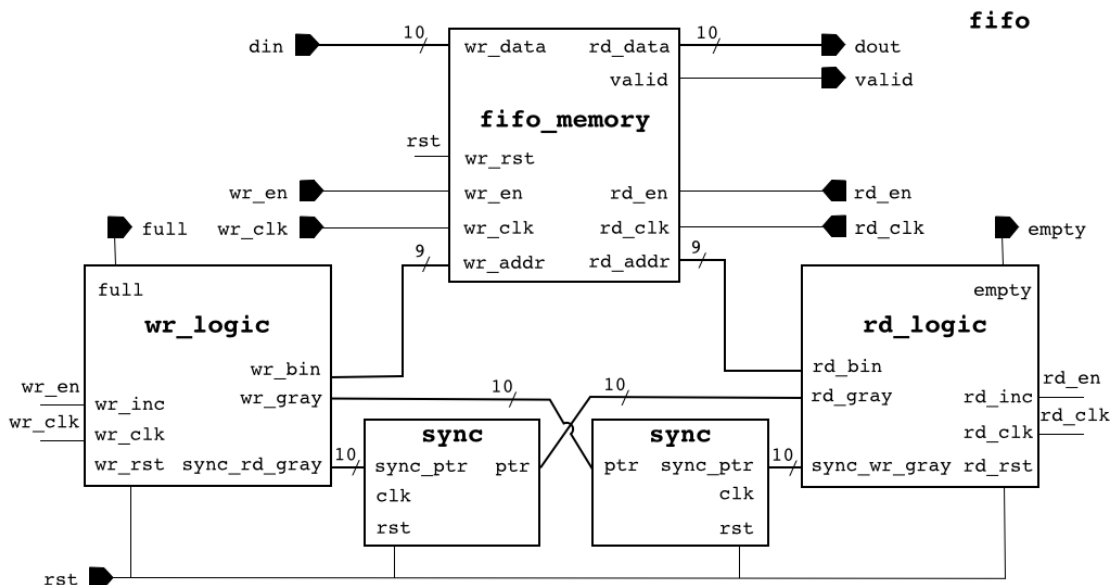


Figure 6.6 – Dual-clock FIFO block diagram.

The designed dual-clock FIFO is composed by the modules in Figure 6.6:

The designed dual-clock FIFO is composed by the modules in Fig.6.6:

- A top-level wrapper including all clock domains (fifo).
- A memory buffer (dual-port SRAM) accessed by both write and read clock domains and with depth and word width adapted to the different cases (fifo_memory).
- One multi-flop synchronizer for each clock domain, to move read and write pointers into the other domain (sync).
- A write logic synchronous to the write clock and containing the full-flag logic and the Gray counter for the write pointer generation (wr_logic).
- A read logic synchronous to the read clock and containing the empty-flag logic and the Gray counter for the read pointer generation (rd_logic).

6.4.3 Single-Clock FIFO

Since single-clock FIFOs do not suffer for the clock domain crossing issues as Dual clock FIFOs, which are reviewed in the Appendix-B the design proposed for asynchronous FIFOs has been simplified. Effectively, synchronizers and Gray counters have been removed because exchanging the write and read pointers do not cause a problem anymore.

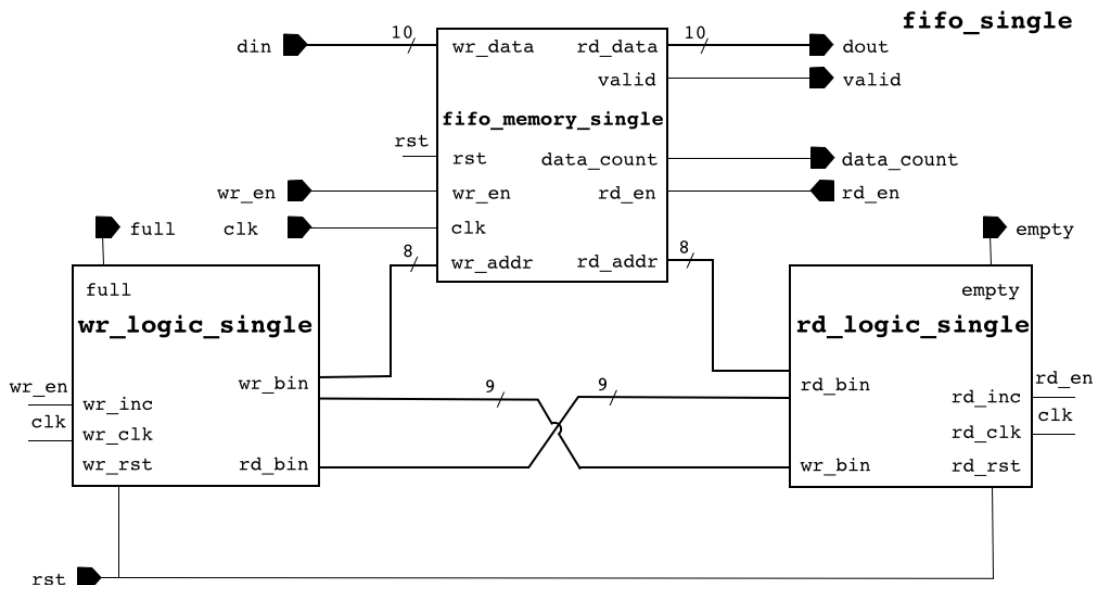


Figure 6.7 – Single-clock FIFO block diagram.

Single-clock FIFOs in Table 6.5 has been implemented with this design unit. Moreover, in order to simplify the design and work with a single clock domain in the first ASIC version, this module has been used instead of dual-clock FIFO.

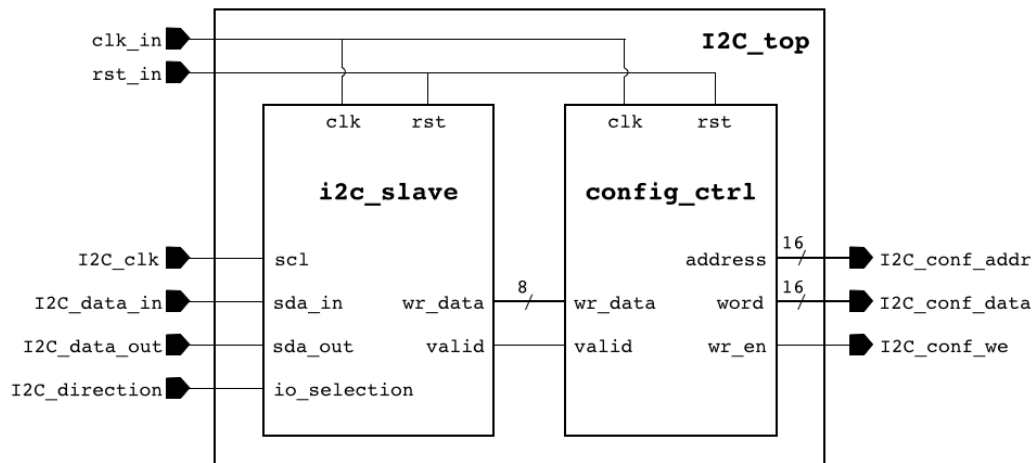
Block diagram

The resulting block diagram is displayed in Fig.6.7. In this case, write and read logic contain a simple binary counter while the memory buffer is still a dual-port SRAM since it might be necessary to read and write different memory cells simultaneously.

6.4.4 Serial Communication Interfaces

In the sphere of miniaturization, in order to minimize the system complexity, designed ASIC does not contain an on board microprocessor. Therefore, a programming interface is needed to communicate from the outside with some configuration registers. The attention has been focused on low data rate serial communication protocols and in the end the most suitable application has been selected.

The designed system emulates the behavior of a single image sensor from the output and interfacing point of view. It generates image frame streams exactly like a single image sensor. I²C protocol is commonly used as an interface in many image sensor designs. In order to be compliant with the market image sensors, the I²C protocol has been chosen, which is usually the preferred one for these applications. Moreover, the ease of implementation and the low data rate is compatible with the system needs.

Figure 6.8 – I²C block diagram.

6.4.5 I²C Design

I²C is a synchronous half-duplex bus requiring only two signal wires, i.e. serial data line SDA and serial clock line SCL. Thanks to pull-up resistors, these lines are pulled up weakly to the high logic level and are controlled by all the connected devices via open-drain drivers, which allow them to pull the bus down. Since the logic 1 level depends on the supply voltage, no standard bus voltage exists. Moreover, so as to allow the master/slave communication, each device connected to the bus is characterized by a unique address. Communication protocol details of I²C is supplemented in Appendix-B. The block diagram of the designed serial communication interface is illustrated in Figure 6.8.

The wrapper I2C_top involves two separate design units:

- i2c_slave, which contains an FSM that emulates the behavior of the communication protocol described in the Appendix-B.
- config_ctrl, which comprises an FSM that organizes the 8-bit data arriving from i2c_slave in 16-bit words, necessary to update the configuration registers.

Registers Configuration

In order to be able to program the configuration registers from an outside master device, the three outputs of the I²C communication interface have been mapped at the input of the panorama generation unit. Fig.6.9 shows the modules included in the registers update process. In the FPGA design, since the Microblaze microprocessor can access the registers via the AXI bus, there was no need for such a block.

The reg_mux unit makes it possible to address different registers. Most of the registers are

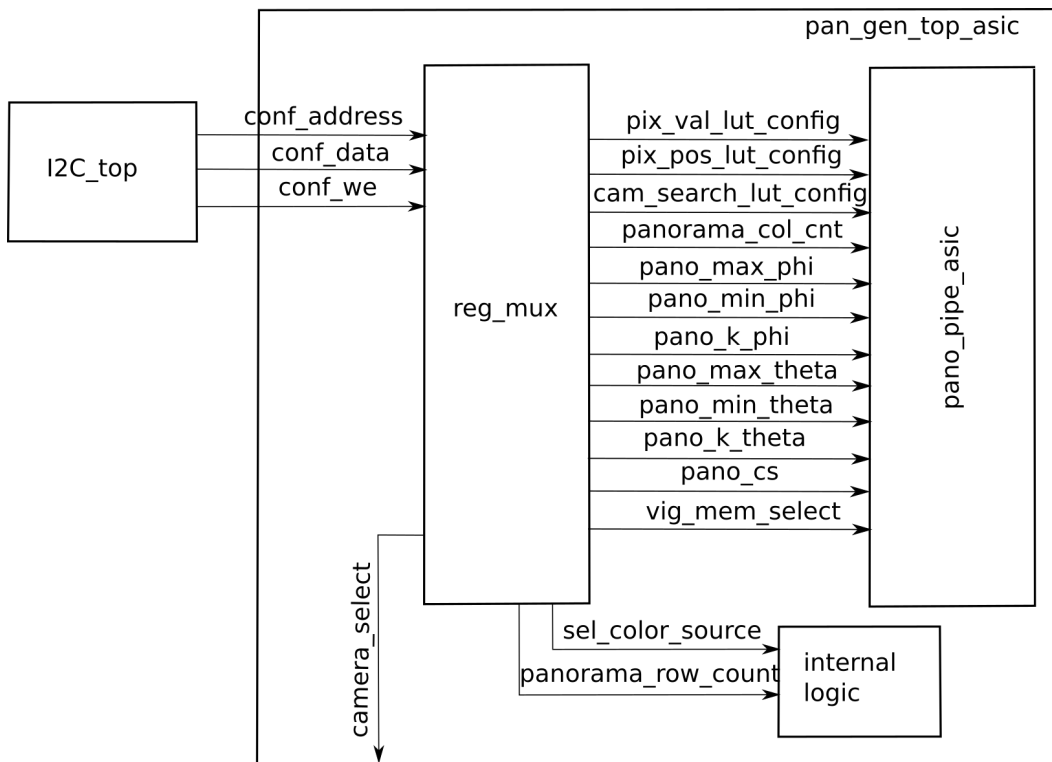


Figure 6.9 – Register update process.

connected to the panorama pipeline, while others are used in some internal logic or set as outputs. Three signals are not truly used as registers but more as channels to access memories or register arrays present in the panorama_pipeline_asic entity:

- pix_val_lutconfig[31:0] is used to access the SRAM-based LUT present inside the pixel value generation module.
- pix_pos_lutconfig[31:0] is used to write to the registers arrays in the pixel position generation module.
- cam_search_lutconfig[31:0] is used to write to the registers arrays in the camera search module.

Table 6.6 reports how the 32 bits of these signals are exploited.

These arrays are composed by 24 different registers, i.e. one for each camera, each of them 48-bit, 32-bit or 16-bit long. In Fig.6.10 an example for the arrays is illustrated.

The complete list of configuration registers present in the design is shown in Table-6.7.

Table 6.6 – Channels bits partition for configuring the internal register arrays and memories, which contains the calibration information

pix_val_lutconfig	pix_pos_lutconfig	cam_search_lutconfig
(31) write enable	[24 : 21] register array select	[24 : 21] register array select
[30 : 16] address	[20 : 16] camera select	[20 : 16] camera select
[15 : 0] data	[15 : 0] data	[15 : 0] data

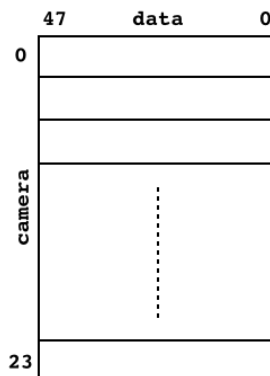


Figure 6.10 – Register array structure.

6.4.6 Memory Wrappers

As indicated in Table 6.4, there is a certain number of embedded memories on chip. In order to make them compliant with the design, memory wrappers have been developed (see Table 6.8). These units allow to:

1. Have a **better port mapping** between memories and signals present in the design, like for FIFO memories and sine and cosine LUTs.
2. Create **large memory blocks** which are not limited by the Block RAM sizes imposed by the IPs vendor, like for frame memory.
3. **Mix** between **SRAM** and **ROM** behavior in the same module, like for vignetting memory.

In frame_mem_wrapper four identical dual-port SRAM are present. In order to choose in which one write to or read from two multiplexers have been used (one at the input and one at the output) and two extra bits of the write and read addresses provide the selection signals.

On the other hand, the vig_memory_wrapper module allows to generate the SRAM-based LUT that has been mentioned in Section-6.4.5. Thanks to the selection signal vig_mem_select, programmable via I²C (see Figure 6.9), it is possible to multiplex between LUT if it is equal

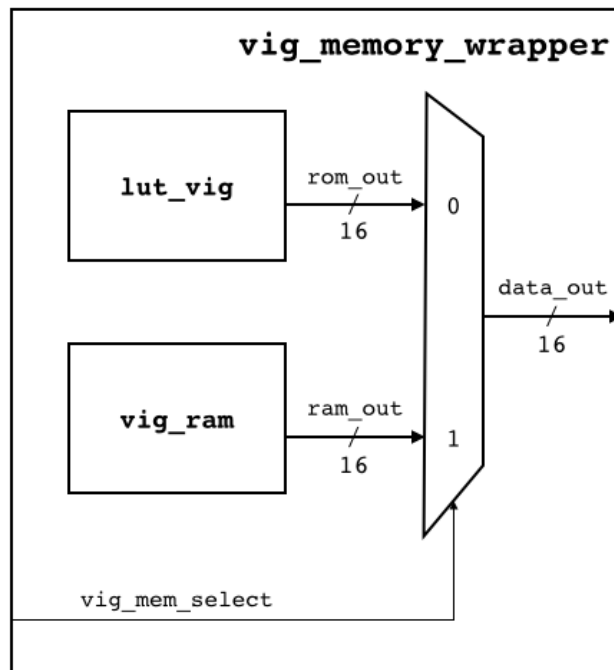


Figure 6.11 – Memory wrapper example.

to logic 0 and SRAM if it is equal to logic 1, as shown in Figure 6.11. This dual behavior is necessary for the following reason. Since the design has been parametrized, at the power up, even without configuring the parameter values a default configuration are necessary to perform panorama generation. While some of the default configuration values are provided in the configuration registers by the reset signal, some others have been stored in this LUT. Effectively, unlike FPGAs, ASICs do not allow to initialize memories at power up. Furthermore, those initialization values permit also to test the system disregarding any failure on the I²C bus.

6.5 ASIC Design

In the current section, the final steps of a top-down digital design flow is dealt with: the RTL synthesis and the place and route. The EDA tools used to perform these two tasks are, respectively, Synopsys Design Compiler and Cadence SoC Encounter. The technology chosen to fabricate the ASIC is the TSMC 40 nm CMOS process.

6.5.1 RTL Synthesis

Logic synthesis has been performed on the designed VHDL RTL models in order to get a gate-level description of them. This task has been performed using a TCL script in which area and timings constraints have been imposed and the appropriate libraries have been used to

link the design. The library for the standard cells has been taken from the TSMC 40 nm design kit while the libraries to link the different embedded memories (see section 6.2) have been provided by the IPs vendor ARM. The main outputs of a synthesis step are:

- A **Verilog netlist** model, which can be used for the post-synthesis logic simulation and as input for the P&R.
- A **SDF (Standard Delay Format)** description, which includes delay information for simulation that are correct for the gates but only estimated for the interconnections.

The system has to work at 175 MHz clock frequency in order to provide real time results at the ASIC output for a 1080 × 1080 pixel 25 fps video stream. To have a margin, 200 MHz final clock frequency is aimed. Thus, two different gate-level netlists have been synthesized by focusing on timing constraints. The first gate-level netlist uses a clock period of 5 ns while the second with a clock of 5 ns . The final critical path reports are shown in Table-6.9.

The slack is the difference between data required time and data arrival time and can be considered as a timing margin. Even if a negative value indicates a violation, a small negative slack or a slack of 0.0 are not a problem since it might be recovered later during the place and route timing optimization steps. For this reason, a shorter clock period has been imposed in order to let the synthesizer infer faster architectures. Nevertheless, the designed system will still work at 200 MHz.

6.5.2 Place and Route

The P&R step infers a geometric realization of the gate-level netlist. As for the synthesis, this task has been performed using a TCL script, taking as inputs the following information used to generate a configuration file:

- The **Verilog netlist** generated during synthesis.
- The **LEF (Layout Exchange Format)** files of standard cells and embedded memories. This format provides the minimum abstract layout information on standard cells and IPs to perform P&R, such as geometry and I/O pin placements.
- The **timing libraries** of standard cells and embedded memories, which include information on the typical, best and worst timings like delays, setup and hold times.
- The **timing constraints** file generated during synthesis.
- The **power** nets that has to be used in the layout, defined in the LEF files. In the design under analysis these are: VDD, VDDPE, VDDCE, VDDE for the power nets and VSS, VSSE for the ground nets.

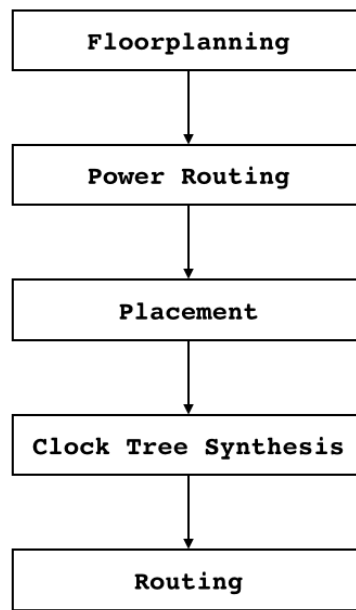


Figure 6.12 – Physical design steps.

For all the files mentioned above, the ones related to standard cells have been supplied by the provider TSMC while the ones related to embedded memories have been generated using the ARM proprietary software. Nevertheless, for the whole layout, 10 metal layers have been used as defined by the TSMC 40nm technology process. Once the design has been imported, the physical design follows the steps shown in Fig.6.12.

The main outputs generated during the place and route are:

- A **Verilog netlist** model, which differs from the one taken as input since further timing optimizations are performed during placement, clock tree generation and routing. It can be used for the post placed logic simulation.
- A **SDF (Standard Delay Format)** description, which now includes both cells and interconnect delays.
- A layout in **GDS2** format.

Some details of the P&R Considerations are provided in Appendix-B

First Design Configuration

Since the macro blocks (SRAMs and ROMs) composing the ASIC occupy most of the chip area, their placement has been done carefully in order to avoid geometry violations and timing issues. For this reason two different versions have been placed and routed choosing different floor planning configurations.

The first design is shown in Figure 6.13. The core size has been specified with an aspect ratio equal to 1 and a core utilization of 80%, leading to a chip dimension of $10\text{ mm} \times 7\text{ mm}$. Table 6.10 lists the core area percentage occupied by the macro cells present in design. The biggest contribution is given by the 96 DP SRAMs `frame_mem_2` instantiated in the 24 `frame_mem_wrapper` modules, since they occupy almost the 70% of the ASIC area. In this design, together with process antenna violations, geometry violations are present, but they are simple errors that can be easily solved manually modifying the interested wires.

The P&R summary reports for this first design are shown in Table 6.11.

An estimation for power dissipation is given in Table-6.12. Here, for the internal switching activity, a worst case situation (switching activity=1) and a moderate case (switching activity=0.4) are illustrated. Although the switching activity can go even lower depending on the internal states of the panorama generation circuitry proposed, it is observed that the strongest contributor of the power dissipation the leakage power. This is due to the fact that the most of the area is occupied by the SRAM blocks. Therefore, for future designs, selecting a low leakage process might be an option. Another improvement can be utilizing power reduction techniques like clock gating where applicable.

Since the designed system is aimed to search for feasibility of migration to the ASIC implementation, a power budget was not defined at this stage of the work. For different applications there would be different power requirements. For instance, for the capsule type endoscopy, since the whole system is stand alone and usually powered by a commercial battery, care should be taken to reduce power with different methods. For a full system implementation, the other components of the system, such as single camera and lightning source would also be important components contributing to the power dissipation. During the system implementation and experiments, I was not utilizing LED sources that can be fit into a capsule size so it is not feasible to give power dissipation estimation for that components for the state of the current work. However, for each camera a power dissipation of 4 mW is measured at 44 fps frame rate at 2.1 V supply voltage. Since there are 24 cameras in the design, this ends up with a 96 mW of power dissipation for a continuous 44 fps operating condition. However, in real applications, for capsule endoscopy such as [62], 5 fps or even less to 1-2 frames per second [90]. From this point, it can be said that a pulsed operation of the circuit and the cameras could reduce the power significantly in the order of 8-40 times by aligning the system operation to the application requirements on the frame rate.

The negative slack time is not a problem. Effectively this calculation has been done considering a clock period of 4 ns while the system will be always run at 5 ns . In this case a positive value would be achieved. Moreover, from the reports, it can be noticed the low value of the chip density subtracting physical cells (i.e. without considering the filler cells used to ensure the continuity of power/ground rails and N+/P+ wells). For this reason another version has been proposed, with smaller core dimensions.

Second Design

Figure 6.14 illustrates the second design. In this case, the core sizes have not been specified through aspect ratio and core utilization but imposing its dimensions, thus leading to a chip $9\text{ mm} \times 9\text{ mm}$ circa.

The `frame_mem_2` have been placed horizontally instead of vertically to allow a better bus sharing between the hard macros of the same memory wrapper. Moreover, thanks to the high aspect ratio, it is possible to place each group of 4 DP SRAM relative to one camera interface next to the core.

The P&R summary reports for the second design are shown in Table 6.13. The areas of the single modules are not reported here since almost equal to the one in Table 6.11.

In this case, process antenna violations, geometry violations and also connectivity violations are present, probably due to the high congestion that did not allow a proper routing. Since these are not only errors related to the physical design but may lead to malfunctioning, the first design has been preferred to perform the post pared logic simulation. Since this design is not feasible, the power dissipation analysis is not considered.

6.6 Logic Simulations

This section describes the logic simulation of the designed ASIC performed using the ModelSim tool from Mentor Graphics.

6.6.1 RTL Simulation

The functionality of the RTL VHDL model of the system under design has been validated through logic simulation. The test bench written for this purpose gives input to the 24 of the camera models by reading the pixel values from the respective text files. Each text file is representing the image frame pixel information. The text files are generated in the MATLAB environment from the single camera images without any image processing steps. As explained in the previous chapter, Section-5.1.3, the camera model transforms this file in a serial stream of bits which is then used to feed the camera interface present in `system_top_asic`. After panorama generation has been performed, the output frames are stored back in a text file which can then be converted to image through a simple MATLAB script. An example 430x430 resolution image is presented in Fig.6.15.

6.7 Conclusion

In this Chapter, an ASIC implementation of the system is explained. The ASIC design is necessary to complete the miniaturization in terms of the hardware part of the miniaturized system. Design decisions are taken to obtain a minimal configuration subset of the FPGA

based system explained in the previous chapter. Custom blocks such as FIFOs, I²C and memory wrappers designed. With this minimal design methodology, it is possible to achieve higher operation frequency, which boosts the throughput of the system. The designed system is verified at the logic simulation level. The design is placed and routed for a 40 nm TSMC technology for different placement configurations.

Table 6.7 – Configuration registers specifications for the ASIC design

Register	Word size	Address	Default value
panorama_row_cnt	11 bit	0x1800	0x1D5
panorama_col_cnt	11 bit	0x2000	0x1AC
pano_max_phi	13 bit	0x2800	0x17F8
pano_min_phi	13 bit	0x5000	0x740
pano_k_phi	13 bit	0x3000	0x00A
pano_max_theta	13 bit	0x3800	0xDF8
pano_min_theta	13 bit	0x4000	0x300
pano_k_theta	13 bit	0x4800	0x006
pano_cs	1 bit	0x6800	0x0
sel_color_source	1 bit	0x7000	0x0
camera_select	5 bit	0x7800	0x1
vig_mem_select	1 bit	0x7C00	0x0
pixel_e_u	24x48 bit	0x0800 - 0x0857	N/A
pixel_e_v	24x48 bit	0x0860 - 0x08B7	N/A
pixel_k_coefs	24x48 bit	0x08C0 - 0x0917	N/A
pixel_fc	24x32 bit	0x0920 - 0x0957	N/A
pixel_cc	24x32 bit	0x0960 - 0x0997	N/A
near_t_vec_tmp_srch	24x48 bit	0x1000 - 0x1057	N/A
near_cos_alpha_srch	24x16 bit	0x1060 - 0x1077	N/A

Register	Description
panorama_row_cnt	Number of rows in the final panoramic image
panorama_col_cnt	Number of columns in the final panoramic image
pano_max_phi	Maximum horizontal angle of view
pano_min_phi	Minimum horizontal angle of view
pano_k_phi	Horizontal step size
pano_max_theta	Maximum vertical angle of view
pano_min_theta	Minimum vertical angle of view
pano_k_theta	Vertical step size
pano_cs	Panorama generation enable
sel_color_source	Test pattern output selection
camera_select	Single camera output selection
vig_mem_select	Memory selection (ROM or SP SRAM)
pixel_e_u	Cameras unit vectors \vec{u}
pixel_e_v	Cameras unit vectors \vec{v}
pixel_k_coefs	Cameras distorsion correction coefficients
pixel_fc	Cameras focal lengths
pixel_cc	Cameras optical centers
near_t_vec_tmp_srch	Cameras unit vectors \vec{t}
near_cos_alpha_srch	Cameras cosine of half angle of view

Table 6.8 – Memory wrappers implemented for the ASIC design

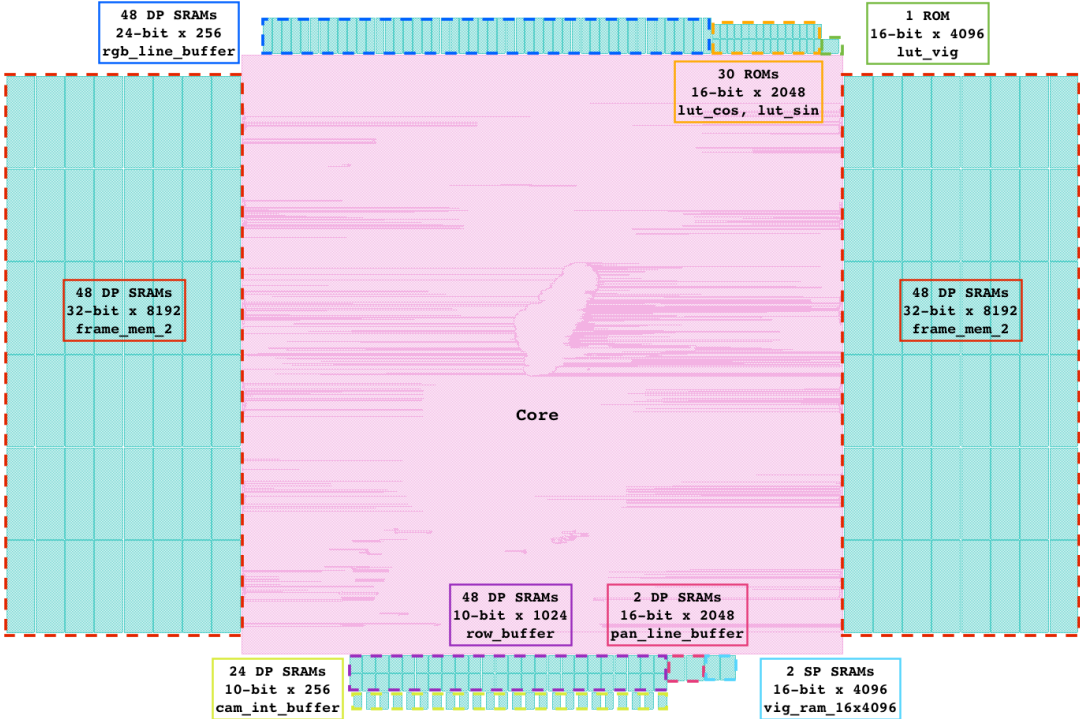
Memory wrapper	Instantiated memories	Total size(KBytes)
fifo_memory_single_w10a8	#1 DP SRAM	0.320
fifo_memory_single_w10a10	#1 DP SRAM	1.280
fifo_memory_single_w16a11	#1 DP SRAM	4.096
fifo_memory_single_w24a8	#1 DP SRAM	0.768
frame_mem_wrapper	#4 DP SRAM	131.072
lut_cos_wrapper	#1 ROM	4.096
lut_sin_wrapper	#1 ROM	4.096
vig_memory_wrapper	#2 SP SRAM - #1 ROM	16.384 - 16.384

Table 6.9 – Critical paths for different synthesized gate-level net-lists of the ASIC design

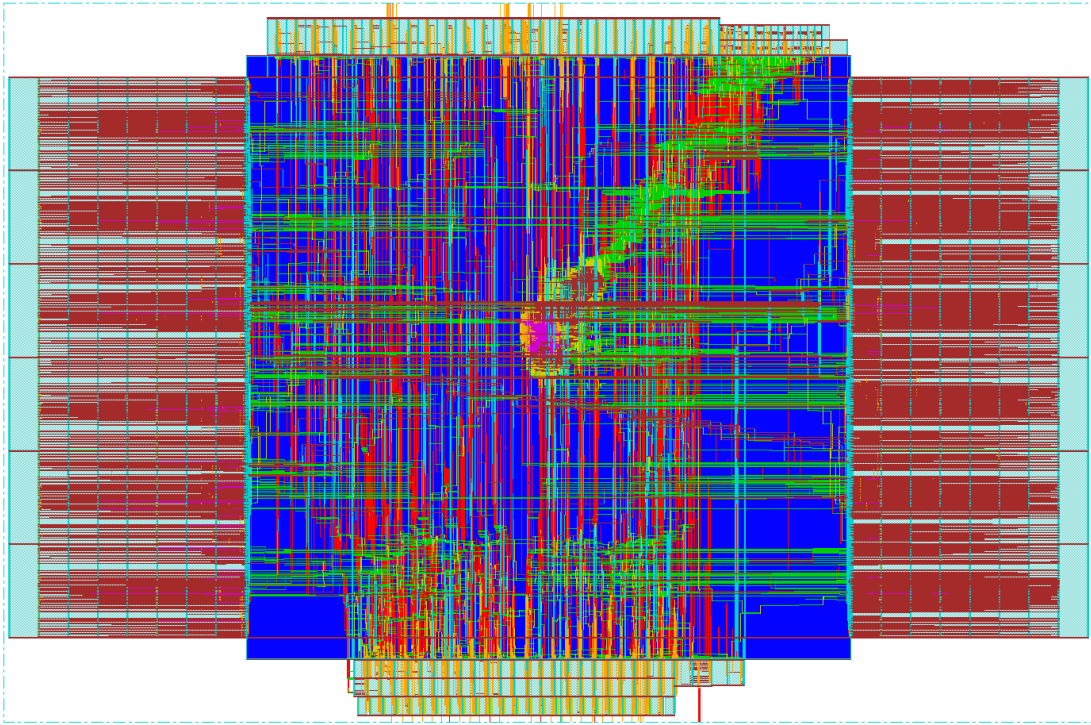
Clock period	5ns	4ns
Data required time	4.880	3.880
Data arrival time	-4.399	-3.880
Slack time	0.481	0.000

Table 6.10 – Core area percentage occupied by macro cells of the first ASIC design configuration

Macro name	Instance count	Percentage in core
cam_int_buffer	24	0.959%
frame_mem_2	96	69.490%
lut_cos	15	0.385%
lut_sin	15	0.385%
lut_vig	1	0.067%
pan_line_buffer	2	0.244%
rgb_line_buffer	48	3.939%
row_buffer	48	2.821%
vig_ram_16x4096	2	0.204%
Total percentage in core		78.494%



(a) Amoeba view.



(b) Physical view.

Figure 6.13 – First P&R design.

Table 6.11 – First ASIC design P&R summary reports

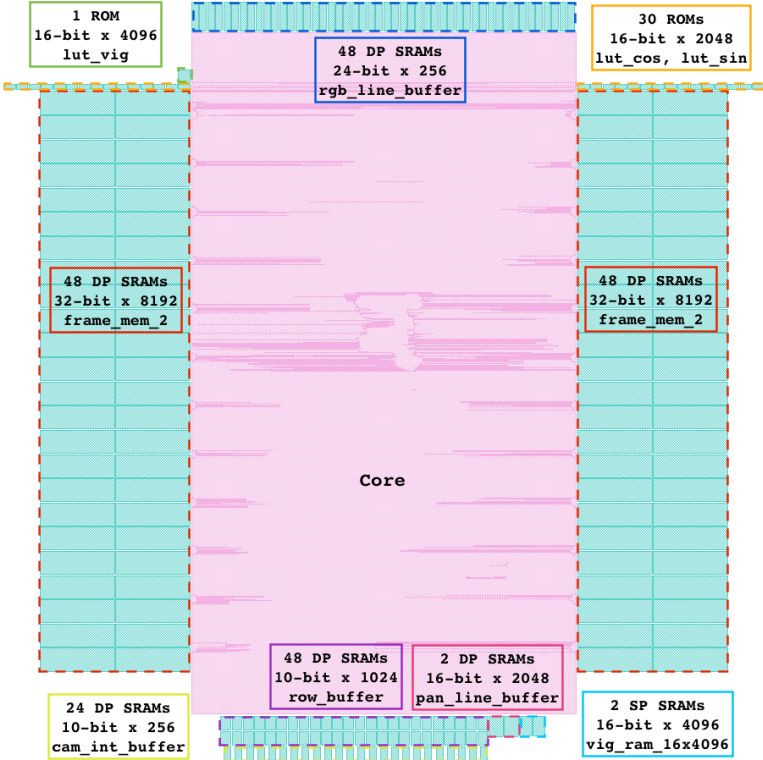
Area system_top_asic	26.189 mm ²
Area I2C_top	441.4 μm ²
Area pan_gen_top_asic	0.740 mm ²
Area single_cam_top	1.056 mm ²
Area camera_out_mux	1110.6 μm ²
Total area of core	32.639 mm ²
Total area of std cells (logic)	0.531 mm ²
Total area of macros (memory)	25.620 mm ²
Total area of chip	70.240 mm ²
2-input NAND equivalent gate count of logic	1M Gate
2-input NAND equivalent gate count of memory	48M Gate
Chip density (counting std cells, macros and routing)	80.123%
Chip density (without routing)	37.232%
Required time	6.636 ns
Arrival time	6.943 ns
Slack time	-0.307 ns

Table 6.12 – First ASIC design power dissipation report for a slow/slow corner VDD=0.81V, T_j=125 °C

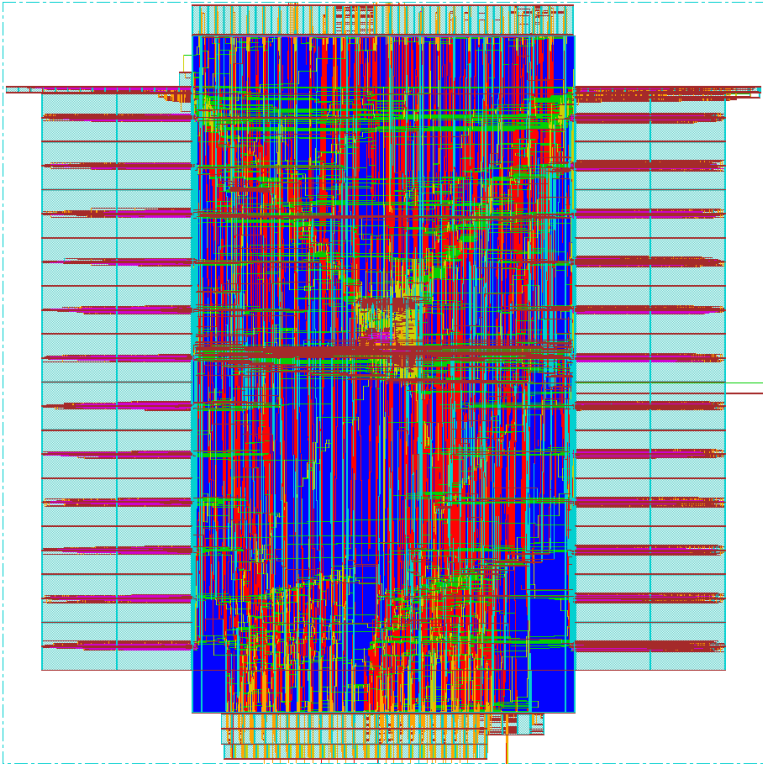
primary input activity	Switching power	Internal power	Leakage power	Total power
0.4	81.33 mW	265.5 mW	634.6 mW	981.4 mW
1	132.4 mW	381.2 mW	634.6 mW	1148 mW

Table 6.13 – Second ASIC design P&R summary reports.

Total area of core	36.299 mm ²
Total area of std cells (logic)	0.585 mm ²
Total area of macros (memory)	25.620 mm ²
Total area of chip	81.431 mm ²
2-input NAND equivalent gate count of logic	1M Gate
2-input NAND equivalent gate count of memory	48M Gate
Chip density (counting std cells, macros and routing)	76.038%
Chip density (without routing)	32.181%
Required time	6.604 ns
Arrival time	7.012 ns
Slack time	-0.408 ns



(a) Amoeba view.



(b) Physical view.

Figure 6.14 – Second P&R design.

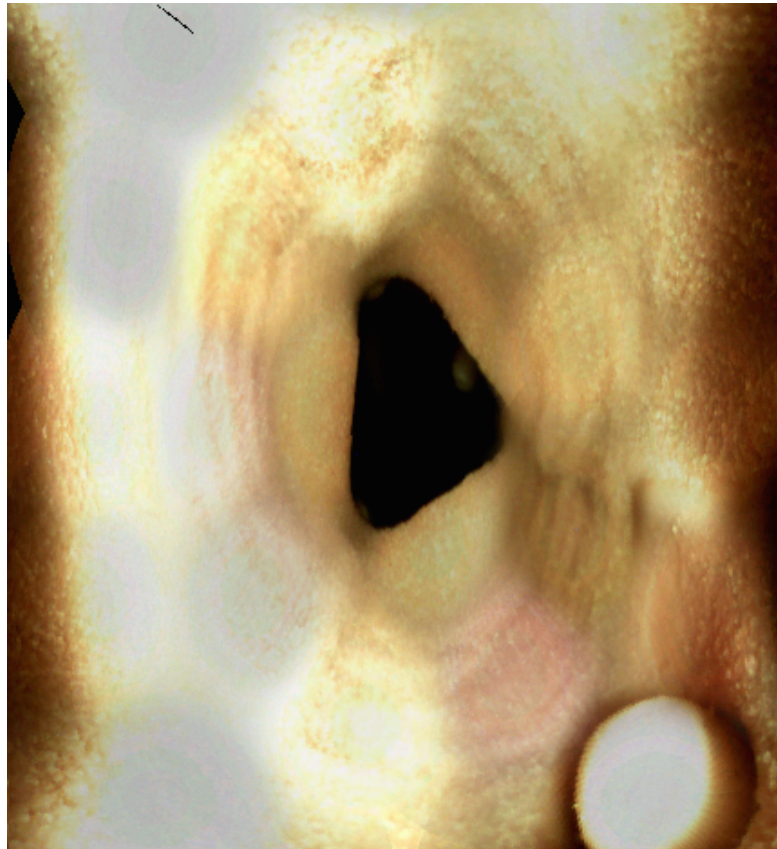


Figure 6.15 – An example of panoramic frame obtained through RTL logic simulation of the ASIC system.

7 Multi-camera large FOV or panoramic imaging system design in macro scale

In this Chapter, design details of a very high resolution, macro scale camera system is explained. This system is an initial work in the scope of the thesis. The explained design and experiments are first described in [91] and [92].

In this system, cameras are positioned in a way to guarantee that every target location in the horizontal plane is covered by at least two cameras. Covering target locations by at least two cameras enables a smooth transition between the images while constructing omnidirectional images, and enables using the device in applications that requires depth map estimation [93]. In this system, the coverage analysis methodology presented in [65] is used to measure the required angles between the camera layers and horizontal plane θ and the angles between the horizontally neighboring cameras φ .

Very-high AOV lenses cause image distortion while low AOV lenses necessitate a large number of cameras to guarantee a large effective AOV. A 5MP image sensor with a 6 mm lens is selected for the construction of the system since it offers an efficient trade-off between large AOV and distortion. The selected camera provides 53° and 43° for the horizontal and vertical axis AOVs respectively, and the full resolution of each camera is 2592×1926 pixels. Following the methodology presented in [65], 44 cameras are positioned on four levels where level-4 represents top camera and level-1 represents cameras in bottom layer. From the top to the bottom layers, 1, 6, 15 and 22 cameras are distributed, respectively. The top camera is perpendicular to the ground plane. The θ angles of the four layers from top to bottom are 0° , 39.6° , 59.3° and 80.8° . The φ angles between the cameras for the layers 1, 2, 3 are 16.36° , 24° and 60° , respectively. The resulting coverage analysis is given in Section-7.3.

Table 7.1 – Properties of the omnidirectional imaging system.

Option	Individual Camera Parameters				Obtained Video				Requirements	
	Observed Resolution	Skipping-Binning	Bit/Pixel	Clock (MHz)	AOV h×v	Output Resolution	Frame Rate	Record duration (min)	Memory Size (GB)	Bandwidth (MB/s)
1	2592 × 1944	0-0	12	96	53° × 43°	2592 × 1944	14	30	177	101
2	2592 × 1944	0-0	12	66	53° × 43°	2592 × 1944	9.5	14	56	68
3	2592 × 1944	1-1	12	86	53° × 43°	1296 × 972	30	30	95	54
4	2592 × 1944	1-0	12	66	53° × 43°	1296 × 972	30	30	95	54
5	2592 × 1944	1-0	12	66	53° × 43°	1296 × 972	30	17	54	54
6	2592 × 1944	1-0	8	66	53° × 43°	1296 × 972	30	30	63	36
7	2592 × 1944	2-0	8	36	53° × 43°	864 × 648	30	30	28	16
8	1024 × 768	0-0	12	46	21° × 16°	1024 × 768	30	30	40	23

7.1 System Parameters and Requirements

After the type, number and positions of individual cameras are fixed, the main two aspects that define the performance of the image recording system are the image resolution and frame-rate of individual sensors. The image sensor chosen in the construction of this system has programmable parameters which affect these two main aspects. Table-7.1 presents some example configurations of camera parameters and their respective memory size, bandwidth and operating frequency requirements with respect to the image capture duration.

The selected image sensor delivers 12-bit raw Bayer data output. The bit precision is not reduced, in order to improve the histogram when the camera is used in extreme dark or bright conditions. Moreover, most of the data compression techniques applied in consumer electronic video recording cameras reduce the image quality and light field capturing capability of the imaging system, thereby creating an incompatibility with high quality post processing. Therefore, a 12 bit raw Bayer recording format is maintained, which considerably impacts the memory requirement. Adjusting the frame-rate of the image sensor is possible by tuning the operating frequency of the sensor and frame size. At the fastest frame rate achieved at 96 MHz, the memory requirement for a 30 minute video recording is equal to 177 GB and the memory bandwidth requirement is 101 MB/s for a single camera. Considering 44 cameras, the memory requirement increases to 5.2 TB for the first implementation option presented in Table-7.1. Thus, the memory bandwidth and size requirements for the selected frame-rate and frame size determine the constraints for the selection of the storage device.

The chosen image sensor offers two options for sub-sampling the frame prior to delivering it, which are known as skipping mode and sub-window selection. The skipping mode allows keeping the highest AOV while reducing the resolution of the sensor, whereas selecting sub-window as exemplified in eighth implementation option of Table-7.1 reduces the AOV. Overlapping the wide AOVs of individual image sensors is crucial for better sampling of the

7.1. System Parameters and Requirements

light field and light field based image processing applications. As shown in the third and fourth analysis cases of Table-7.1, operating in skipping mode reduces the image size to one quarter of the original 5MP, 1296×972 , while maintaining a constant angle of view (AOV). The binning mode aggregates the Bayer data of the skipped pixels when it is combined with skipping mode. Using the binning mode together with skipping requires a 86MHz clock to provide 30 fps video. Providing the same resolution without using the binning mode decreases the required pixel clock frequency to 66 MHz. Recording 30 minutes using one of these options requires 95 GB of memory and a 54 MB/s bandwidth for each camera.

The chosen image sensor does not provide differential output signals, and thus, additional limitations pertaining to signal integrity arise for building the complete system. Constructing a complete system on a dome that has a radius of 20 cm requires data cables with a minimum length of 40 cm. Transmission of a 96 MHz clock over 40 cm cables is prone to noise due to signal reflection, even using very-high-density cable interconnect (VHDCI). Nevertheless, any noise in the image vanishes when the clock frequency is decreased below 80 MHz. Consequently, selecting the appropriate camera operation frequency is a determinant system-level constraint.

The system requires large-capacity and large-bandwidth storage devices. The three main candidate storage devices are i) the Compact Flash, ii) the Hard Disk Drive (HDD) and iii) the Solid State Drive (SSD). Compact Flash devices are very expensive compared to HDD and SSDs, when their considered capacity is larger than 64 GB. HDDs are the most cost-efficient storage devices. However, their mechanical structure has a crucial impact on their effective recording bandwidth under environmental vibration conditions. Vibration robustness tests carried out with several different HDDs have demonstrated that the effective data bandwidth can be reduced down to 20 MB/s. Storage devices are expected to support constant bandwidth under vibration since the complete imaging system may not remain stationary, and move during image recording. Yet, offering a constant recording bandwidth is an important constraint of the system, which only SSD technology can provide in presence of external vibrations. Therefore SSDs are selected as the primary, i.e., real-time, storage devices.

For SSD communication interface, Serial ATA (SATA) 2.0 standard is chosen, in order to fulfill the BW requirements in burst write mode and due to their availability on the market. SSD devices with SATA 2.0 usually sustain a bandwidth of approximately 150 MB/s, which is higher than the 101 MB/s bandwidth requirement of a full resolution system operating at 14 fps. Constructing a system that supports a resolution of 1296×972 pixel require a 54 MB/s bandwidth. Hence, one SSD can act as the storage device for two cameras simultaneously. 128 GB SSDs can continuously record for 17 minutes from two cameras that operate at 30 fps and with a 1296×972 pixel resolution. There is also need for intermediate buffer memory in the architecture of the targeted embedded system in order to make the burst writes to

Chapter 7. Multi-camera large FOV or panoramic imaging system design in macro scale

the SSD possible. Buffering a single frame in full resolution, and 1296×972 pixel resolution approximately requires a memory size of 8 MB and 2 MB, respectively. DDR2 SDRAM memories typically have a data size of 256 MB or more, and an approximate bandwidth of 5 GB/s. DDR2 are thus selected as the efficient devices to use in conjunction with the SATA protocol for buffering images. Moreover, benefiting from the maximum resolution of the cameras is possible, but in this case the frame rate of the cameras should be reduced due to bandwidth limitations. 128 GB SSDs can continuously record for 14 minutes from two 2592×1944 pixel resolution cameras that operate at 9.5 fps.

The developed system is planned to sustain constraints and target features summarized in Table-7.2 and Table-7.3. A maximum flexibility in terms of resolution and frame rate is aimed. When the system is configured to utilize 1296×972 resolution settings of the cameras with the system constraints presented in Table-7.2, the systems is able to provide 21.6 MP omnidirectional video at 30 fps. When the system is configured to utilize 2592×1944 resolution settings of the cameras with the system constraints presented in Table-7.3, the systems is able to provide 82.3 MP omnidirectional video at 9.5 fps.

Table 7.2 – System Constraints to generate 30 fps 21.6 MP Omnidirectional Video.

#Cameras	44	Pixel Clock	66 MHz
Resolution per Camera	1296×972	Record Duration	17 min
Skipping Mode	1	Buffer Memory	DDR2
Binning Mode	0	Storage Device	SSD
#Cameras/#SSDs	2	Back Up Device	HDD
Pixel Resolution	12-bit Bayer	Storage Protocol	SATA II

Table 7.3 – System Constraints to generate 9.5 fps 82.3 MP Omnidirectional Video.

#Cameras	44	Pixel Clock	66 MHz
Resolution per Camera	2592×1944	Record Duration	14 min
Skipping Mode	0	Buffer Memory	DDR2
Binning Mode	0	Storage Device	SSD
#Cameras/#SSDs	2	Back Up Device	HDD
Pixel Resolution	12-bit Bayer	Storage Protocol	SATA II

7.2 System Architecture

In order to support the constraints explained in Section-7.1, the hardware platform should be flexible and configurable. Moreover, it should be expandable for future developments in terms of handling critical parts of the image processing on the captured light field. Thus, an FPGA based embedded system is the most suitable hardware platform for the mentioned purposes. The choice for the hardware platform is the Xilinx XUPV5-LX110T FPGA Board. The chosen

FPGA board has 2 SATA connections, 64 available external pins for connecting two cameras, 128 MB DDR2 memory, a UART, digital video interface (DVI) and SubMiniature version A (SMA) interfaces. 128 GB Kingston SATA 2.0 SSDs are selected as storage devices, each supporting two cameras. In summary, 22 FPGA boards and 22 SSDs are used for recording 17 minutes video captured from 44 cameras at 1296×972 resolution or 14 minutes video captured from 44 cameras at 2592×1944 resolution.

One of the 22 FPGAs serves as Master FPGA (M-FPGA), which is responsible of receiving commands from a host PC system. The commands are for covering the tasks such as start/stop, snapshot, and changing the image sensor parameters by writing to their responsible registers. The M-FPGA receives commands via RS232 interface and broadcasts to the slave FPGAs (S-FPGA). A Graphical User Interface (GUI) designed to handle the communication between user and imaging system. The system architecture of the M-FPGA is shown in Fig.7.1. The S-FPGA architecture is identical to the one in Fig.7.1 except the clock scheme, which is further detailed later in this section. As presented in Fig.7.1, the hardware architecture of a single FPGA includes a Microblaze Soft Processor, Processor Local Bus (PLB), Image Capture modules, DVI Displayer modules, SATA Interface modules, and System Interconnection modules.

The Image Capture modules are used to capture images from two cameras and to buffer them into a DDR2 Memory. Additionally, an automatic color gain controller (AGC) and automatic shutter width controller (ASW) are implemented.

The Microblaze initializes the cameras through an I²C bus, and one single I²C bus controls two cameras, consistently. The Camera Controller module samples the pixels using camera control signals hsync and vsync. The Automatic Gain Control is an important feature pertaining to the white balance. The Automatic Shutter Width enables to adjust the exposure time in extreme dark and bright conditions. The AGC+ASW sub module in Fig.7.1 computes the average red, green, and blue values in the image, and transfers these values to the Microblaze processor using software-accessible registers. Depending on the application or light conditions, the software hosted by the Microblaze provides flexibility to the user for manual and automatic adjustment of the color gains and shutter width. Two native port interface (NPI) of the multi port memory controller (MPMC) module are used to simultaneously write two images into the DDR2 memory. Row Buffers are placed as a FIFO between the Camera Controller, Burst Converter and NPI Controller sub modules to correctly exchange data while they are operating with different clock rates. These buffers provide flexibility to the user to capture and save images with different resolutions and frame-rates.

The DVI Displayer module reads the images captured by one camera, converts the Bayer

Chapter 7. Multi-camera large FOV or panoramic imaging system design in macro scale

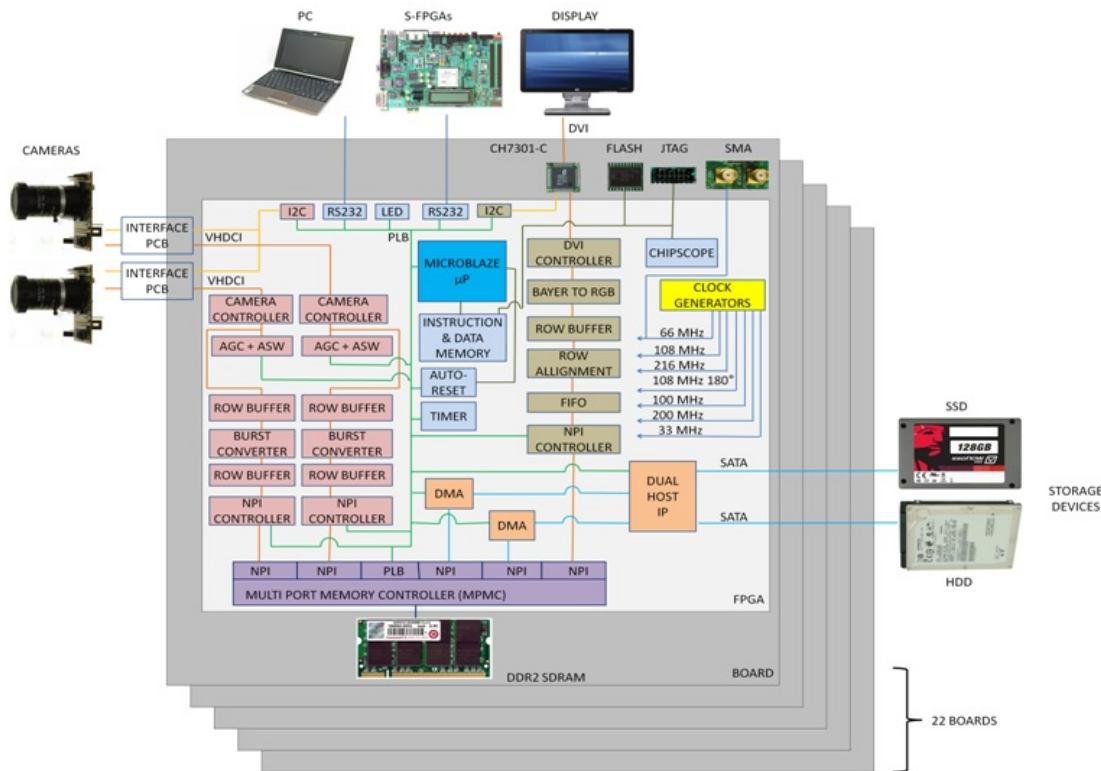


Figure 7.1 – Top-level block diagram of the system hardware architecture

images to RGB format, and displays the images on a DVI monitor. The Bayer to RGB hardware is only implemented for display purposes. The NPI controller of the DVI module is able to switch between the DDR2 memory pointers to display images of the selected camera. The DVI connection is initialized by the I²C module for 1024×768 resolution at 60 fps to display sub-window of the obtained images during the recording process. The DVI interface is operated with a 216 MHz differential clock.

The Host SATA IP can be connected to two storage devices. Although the SATA IP is able to switch between the SATA ports, it is not able to dump data into two storage devices, consistently. Therefore, a 128 GB SDD is connected to one of its two SATA ports. The other port of the IP is used for a backup purpose. Connecting one 2TB HDD for backup support increases the record duration to more than 3.5 hours. After every continuous record of 14 minutes at a 2592×1944 video resolution or 17 minutes at a 1296×972 video resolution, the system is able to make a data backup from the SSD to the HDD. Upon backup completion, the system makes itself ready to record the next continuous video on the SSD. The SATA IP is connected to the DDR2 through MPMC using two Direct Memory Access (DMA) controllers.

The System GUI includes start and stop options to enable the capture of short videos, and

enables capturing single and time-delayed repetitive shots for photographic applications. Moreover, the last used memory address is also saved into the SSD. Consequently, the system GUI provides options to resume from the last memory address or overwriting the previous records, after a system power off-on cycle.

Camera Synchronization is one of the most important technical issues in multiple-camera systems. For perfect synchronization, the number of images taken from different cameras should be identical during the capture of a video, and every image originating from the 44 cameras should be shot at the same moment.

In order to force all cameras to capture an identical number of frames, the frame-rates of the cameras must exactly match. All cameras are programmed to the same resolution and exposure time. In addition, the main factor guaranteeing setting the frame rate is the pixel clock frequency. The clock crystal mounted on the XUPV5-LX110T board provides 25MHz with a $\pm 0.0004\%$ tolerance, which means ± 100 clock cycles. If each of the 22 separate crystal oscillators is used as main clock source for each camera pair via the internal PLLs of FPGAs, the $\pm 0.0004\%$ deviation causes different frame rates, and prevents synchronization. In order to guarantee clock synchronization, a clock chain is built up that provides an identical clock rate to all of the cameras in the system. This shared clock is generated from the crystal oscillator of the M-FPGA. Each FPGA receives its clock from its neighbor, and transmits it to the FPGA located next within the chain, using coaxial SMA cables and connectors. All the clock inputs and outputs of the SMA connection are buffered to increase the clock drive strength.

In addition to generating an exact frame rate, an additional constraint for the camera synchronization resides in the timing for the acquisition of the images. The image snapshots should be acquired almost at the same moment for a successful synchronization. To this aim and as shown in Fig.7.1, one I²C module of the PLB bus is used for two cameras. Since the camera chips have the same I²C address, two cameras connected to the same I²C module can be configured at the same moment, which enables a perfect synchronization. The shutting time difference between the cameras connected to the same FPGA is 0 or 1 clock cycle at a 66 MHz pixel clock.

Two serial communication chains are implemented in the system. The first serial communication chain is implemented between the PC host system, M-FPGA and S-FPGAs. This chain shares the user commands and system orders of the M-FPGA over all connected FPGAs for starting and stopping the record on different FPGA boards synchronously. All FPGAs initialize their own connected two cameras at the same time. The posing time difference between the cameras connected to different FPGAs is measured to be less than 400 clock cycles, i.e., less than 5 μ s. While a single camera operates at 30 fps, the posing time delay

between two consecutive frames of the same camera is approximately 33 ms. Hence, the 5 μ s delay is negligible. Thus, the synchronization of the cameras that are connected to the different FPGAs can be considered almost perfect.

The second RS232 chain is implemented to provide information about the recording status of the S-FPGAs to the M-FPGA. This feature is mainly important for the SSD to HDD backup process since the backup process is not synchronized due to variable performance of HDD. Each S-FPGA informs its neighboring board via second serial chain as soon as it finishes the current backup process. When all the S-FPGAs finish their backup process, the M-FPGA configures all the S-FPGAs for the next record to the SSD using the first serial chain.

7.3 Implementation Results

The hardware architecture of the proposed video recording system is implemented using ISE 12.4 and XPS. The presented system is constructed using 44 Aptina cameras, 22 XUPV5-LX110T Virtex-5 FPGAs and 22 Kingston 128 GB SSDs. The backup process is verified using 2TB Hitachi HDDs. The XUPV5-LX110T FPGA includes 69k Look-Up Tables (LUT), 69k DFFs and 148 Block RAMs (BRAM). The proposed hardware consumes 31% of the LUTs, 25% of the DFF resources and 48% of the BRAM resources. The Microblaze microprocessor and the SATA IP are operated at 100 MHz and 200 MHz, respectively. The system is able to record 17 minutes of continuous video at a resolution of 1296 \times 972 and a raw Bayer data format at 30 fps, or 14 minutes of continuous video at a resolution of 2592 \times 1944 at 9.5 fps. The proposed backup system enables increasing the record duration to more than 3.5 hours with discrete video records of 14 min or 17 min using a 2TB HDD for each FPGA board. The system is perfectly synchronized for the two cameras connected to the same FPGA, and the time delay between the cameras connected to different FPGAs is equal to 5 μ s which is considered negligible.

The final constructed Giga-Eye system is shown in Fig.7.2, which has a size of 56 \times 48 \times 73 cm and a weight of 55kg without the carrier. The system is powered by a single power supply with a 5V output voltage. Total system consumes 92A, and thus 460W of power.

The video frames captured by Giga-Eye are converted into omnidirectional video sequences by offline processing using the Autopano-sift [94], which is a commercial stitching software. Calibration parameters that are estimated by Autopano are used for the reconstruction. Continuous omnidirectional video rendering is obtained by merging these stitched images in time-domain using Matlab. One of the 21.6MP omnidirectional picture obtained using the presented Giga-Eye system is shown in Fig.7.3. In addition, a 81.3 MP omnidirectional image result of Giga-Eye is presented in Fig. 7.4. As presented in Fig.7.3 and Fig.7.4, although the static and dynamic objects that are shown with sub-windows are quite far, Giga-Eye is able to

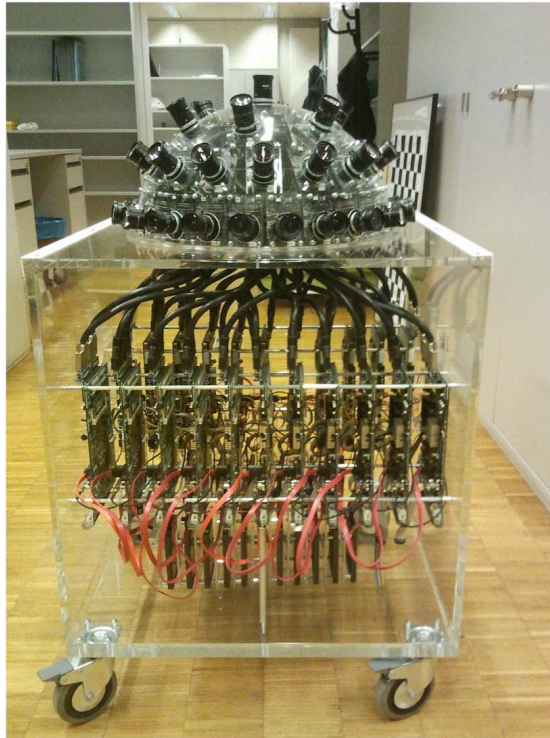


Figure 7.2 – The complete omnidirectional imaging and recording system (Giga-Eye), overall system dimensions are 56x48x78 cm

visualize these objects while providing 360° omnidirectional image.

The obtained coverage map after building the prototype and calibration of the cameras is presented in Fig.7.5. At most 7 cameras, and at least 2 cameras are capable of capturing every direction, provided that θ angle of the observed direction is below 60°. Therefore, the proposed system does not only provide panorama, but also its efficient coverage enables using the device in light field based image processing applications such as refocusing and 3D rendering.

In Table-7.4, the proposed hemispherical video record system is compared with existing large angle of view video capture systems. The frame rates below 25 fps are not typically considered as video, but they are also compared with the presented work in the table. AOVs of the compared systems and their final resolutions are provided in the comparison. Currently, the proposed Giga-Eye system is the highest resolution 360° omnidirectional camera that provides standard frame-rate video output by its 21.6 MP video output capability at 30 fps. Moreover, Giga-Eye is the highest resolution 360° omnidirectional camera with its 82.3 MP output capability at 9.5 fps. The resolution of the Giga-Eye system can be further increased by omnidirectional image based super-resolution techniques [95] or using higher resolution

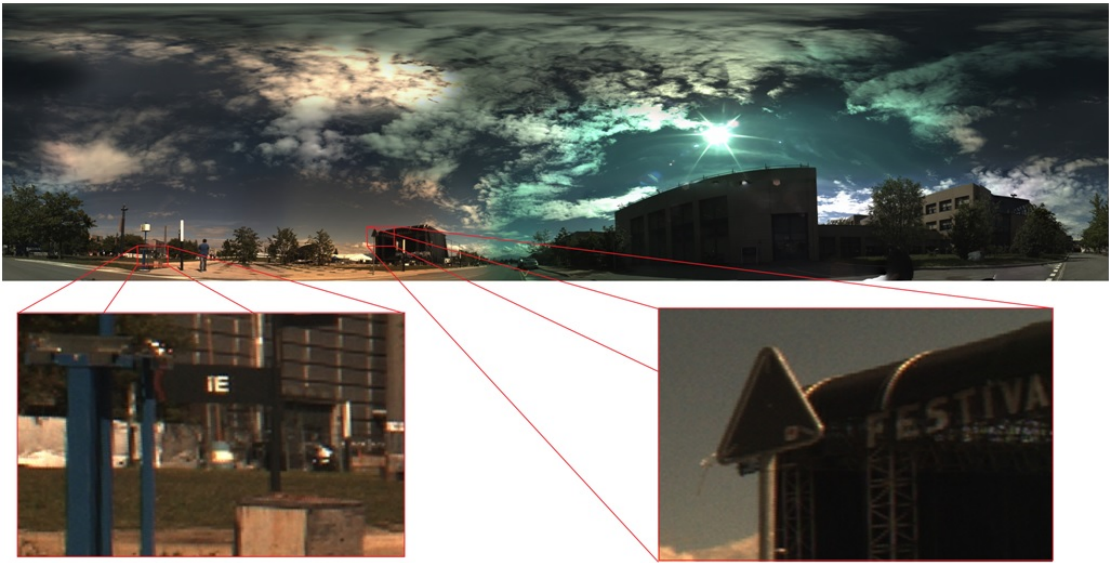


Figure 7.3 – Omnidirectional image obtained with the Giga-Eye system at 21.6 MP resolution showing the central campus square of EPFL, and two selected details (sub-regions) in this image. This omnidirectional image corresponds to one single frame of the 30 fps video obtained by the system.



Figure 7.4 – Omnidirectional image obtained with the Giga-Eye system at 82.3 MP resolution. This omnidirectional image corresponds to one single frame of the 9.5 fps video obtained by the system. Flying plane and the moving car are shown in sub-windows

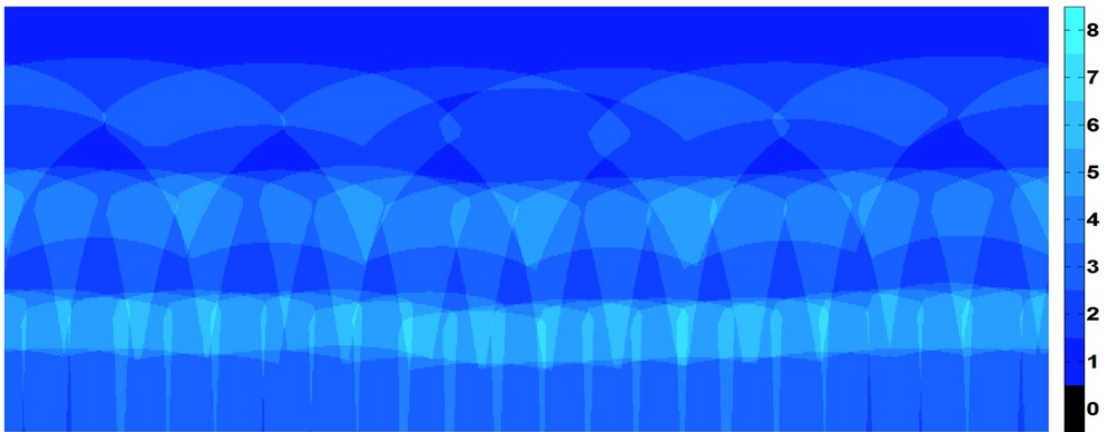


Figure 7.5 – Measured coverage map of the omnidirectional imaging system showing a high pixel redundancy especially close to the equator. The color labels indicate the number of the overlapping individual camera AOVs

sensors.

Table 7.4 – Comparison of the Giga-Eye with existing high-resolution omnidirectional camera systems.

System	Total Sampling Pixel Amount (#Cameras × Camera Resolution)	Record Frame Rate (fps)	AOV	Resolution
Ladybug3 [96]	11 MP	15	360° × 150°	2048 × 4096
Google [97]	75 MP	0.4	360° × 160°	4096 × 8192
Panoptic [65]	4 MP	25	360° × 100°	256 × 1024
Giga-Eye	55 MP	30	360° × 100°	2400 × 9000
Giga-Eye	220 MP	9.5	360° × 100°	4650 × 17700

7.4 Conclusion

In this Chapter, the implementation details of the very high resolution system are mentioned. The system is intended for large area surveillance and tested on the field with different targets as explained. The size and the resolution constraints obtained during this implementation was an initial reference for the rest of the thesis work.

8 Conclusion

In this thesis, a new, insect eye inspired imaging system is described with all the aspects from mechanical model to electronics level miniaturization. The proposed methods and the final prototype constitutes the up to date example of how small such systems can go with the current off-the-shelf cameras while preserving reasonable quality measures such as high resolution and real-time operating capability.

8.1 A model for miniaturization of insect eye inspired imaging system by using off-the-shelf components

The previous methods for mimicking the insect-eyes with multiple cameras are analyzed. Two design approaches are developed. One of them is to minimize the number of cameras utilized, under the constraint of keeping the panoramic view after a certain distance from the camera system. The second approach developed is to maximize the number of cameras in a limited volume dictated by the application, such as colonoscopy. The state-of-the-art miniature and simplified image sensor-optics combination are utilized for implementation of the second approach.

Furthermore, a distributed built-in illumination capability is added to the system by featuring fiber optic technology. First time a multi-aperture compound eye is reported with this capability.

A 24-camera prototype is designed and fabricated with the features mentioned above. The system is geometrically calibrated by using a commercial software, which is utilizing bundle adjustment and feature extraction methods.

8.2 Image processing for seamless compound image generation

Image processing methods are analyzed for previous multi-camera panoramic field of view systems. The panorama generation is handled as an inference problem, and a method is proposed for seam removal and ghosting artifact correction. The method is tested with the images from the 24-camera prototype system in a realistic human bowel model. Additional tests are done with images from different multi-camera systems.

An image pre-processing method for object boundary detection proposed. The proposed pre-processing method is based on the information obtained by utilizing the inter-camera intensity differences at neighboring camera overlap regions. The method is tested by using the images from the 24-camera prototype and from different multi-camera panoramic imaging systems.

An image processing approach is proposed to use the multi-camera overlap region inter-camera intensity difference information to remove the ghosting effects. The method is tested on the images from the 24-camera prototype.

8.3 Digital system design for image processing

An image processing pipeline is embedded and optimized for the miniature 24 camera system. A hardware implementation for the probabilistic method for panorama generation is performed. A hardware implementation of the panorama generation pipeline is done for throughput increase. The full system is implemented and tested on Xilinx Virtex7 FPGA. The system is capable of generation 1 Mpixel video at 25 fps with a 120 MHz panorama image processing clock frequency.

8.4 ASIC Implementation

An ASIC implementation of the hardware system is performed for TSMC 40 nm technology. A minimal subset of the full FPGA system is chosen to have a small area. It is observed that for the proposed design, the on-chip memories utilized in the ASIC design are bounding in terms of area and power dissipation.

8.5 Future Directions

ASIC tape out and integration: Since there is a need for having a small ASIC form factor, the next target is to map the design on a 28 nm technology, and to perform the implementation with post place and route simulations. Also there is a need for adding the proper sampling I/O blocks such as custom designed LVDS inputs, with a high impedance output driver stage to include the programmable features of the camera. I²C I/O buffers should be designed as well.

Since this is the first time with a distributed built-in illumination type compound eye, an in depth analysis of the light capabilities should be performed. The fiber optic illumination channels can be utilized for active imaging applications. The first point can be after a characterization and calibration of the illumination capability, the depth of the objects can be extracted by using even visible light spectrum. Especially for the endoscopy applications, the distributed illumination system combined with the multi-aperture imaging can bring new opportunities in terms of active imaging methods.

An integration method for the colonoscopy devices can be performed and some field tests can be done with the scientists and medical doctors from the endoscopy domain. In this way, real-world application of the system can be achieved.

A Analysis and implementations of different Bayer Demosaicing and Automatic White Balancing Methods

In this part the analysis and implementations for Bayer to RGB demosaicing and white balancing methods from the literature are included. Finally, due to the area and timing constraints, the methods are not included in the FPGA system design described in Chapter-5

A.1 Analyzed and implemented Bayer to RGB methods

A.1.1 Gradient Based Methods

For gradient based methods a 5×5 environment around the interpolated pixel has to be used because otherwise the gradients cannot be estimated in a sufficiently precise manner. This work is restricted to methods using a 5×5 window, however there are other methods which use bigger windows, but they generally do not perform much better than gradient based methods with a 5×5 neighborhood.

The main problem of the bilinear and other linear demosaicing algorithm is that they have a low pass filtering effect which distorts edges. A way to prevent this low pass filtering effect is to estimate the edge direction and then do the still low-pass-like interpolation orthogonally to the estimated edge direction, in order to preserve the sharpness of the edge [87].

Even though gradient based methods usually perform significantly better than other types of interpolation, they have the drawback that their filtering effect cannot be represented using a linear filter, which is especially problematic when doing demosaicing in software and not in hardware. The rest of this section is organized as follows: first, the original gradient based interpolation algorithm developed by Adam and Hamilton [98] is presented, second, as a recent development, an improved version of Adams and Hamilton's method is presented and finally, a linearized version of Adams and Hamilton's method is presented.

Appendix A. Analysis and implementations of different Bayer Demosaicing and Automatic White Balancing Methods

A.1.2 Adam and Hamilton's Method

Adam and Hamilton's method achieves the demosaicing in two steps: in a first step the green channel is reconstructed and in a second step, using the raw Bayer data as well as the reconstructed green channel, the red and the blue channel are reconstructed.

Pane Reconstruction To do the green pane reconstruction first a vertical gradient and a horizontal gradient are defined as follows

$$L_H = |G_{14} - G_{12}| + |2 \cdot B_{13} - B_{11} - B_{15}| \quad (\text{A.1})$$

$$L_V = |G_{18} - G_8| + |2 \cdot B_{13} - B_3 - B_{23}| \quad (\text{A.2})$$

where G_{xx} corresponds to the green value at position xx as shown in Fig.5.6b. These two gradients not only include luminance information, but also Laplacian second order term of chromaticity channel. Therefore these gradients can reflect edge directions as to where the central pixel is.

After having calculated these estimators of the edge direction, the interpolation of the green channel can be done as follows: if $L_H < L_V$ then the principal edge direction is assumed to be vertical and there we need to interpolate horizontally as shown in (A.3). The first term in the equation is like the normal bilinear interpolation just applied in the given direction and the second term is used for edge enhancement.

$$G_{13_H} = \frac{G_{12} + G_{14}}{2} + \frac{2 \cdot B_{13} - B_{11} - B_{15}}{4} \quad (\text{A.3})$$

On the contrary, when $L_H > L_V$ the interpolation direction needs to be vertical in order not to low pass filter the edge and is given in (A.4). Again, we have the normal low pass interpolation with an additional edge enhancement term.

$$G_{13_V} = \frac{G_8 + G_{18}}{2} + \frac{2 \cdot B_{13} - B_3 - B_{23}}{4} \quad (\text{A.4})$$

A.1. Analyzed and implemented Bayer to RGB methods

For the case that both gradients are equal, the green value is given as the mean of (A.3) and (A.4).

$$G13_{HV} = \frac{G13_H + G13_V}{2} \quad (\text{A.5})$$

Red/Blue Pane Reconstruction In a second step, the blue and the red channel can be interpolated. Due to the symmetry of the Bayer filter, the blue channel can be interpolated in the same way as the red channel; therefore only the interpolation of the red channel is presented here.

There are four cases which have to be distinguished for estimating the red value at a certain pixel: when the value is measured, no interpolation needs to be done and the measure value can directly be used. For the three other cases, the estimation is done as follows.

Red Value at Green Pixel, Neighboring Red Pixel in Same Column

Taking $R12$ as an example, we have

$$R12 = \frac{R7 + R17}{2} + \frac{2 \cdot G12 - G7 - G17}{2} \quad (\text{A.6})$$

Red Value at Green Pixel, Neighboring Red Pixel in Same Row

Taking $R8$ as an example, we have

$$R8 = \frac{R7 + R9}{2} + \frac{2 \cdot G8 - G7 - G9}{2} \quad (\text{A.7})$$

Red Value at Blue Pixel

Taking $R13$ as an example: first, two diagonal gradients needs to be defined as

$$\Delta N = |R7 - R19| + |2 \cdot G13 - G7 - G19| \quad (\text{A.8})$$

$$\Delta P = |R9 - R17| + |2 \cdot G13 - G9 - G17| \quad (\text{A.9})$$

Using those two gradients, the principle of interpolation orthogonally to the main edge direction can be applied again.

For the case that $\Delta N < \Delta P$, the estimation is given as

$$R13_{\Delta N} = \frac{R7 + R19}{2} + \frac{2 \cdot G13 - G7 - G19}{2} \quad (\text{A.10})$$

Appendix A. Analysis and implementations of different Bayer Demosaicing and Automatic White Balancing Methods

For the case that $\Delta N > \Delta P$, the estimation is given as

$$R13_{\Delta P} = \frac{R9 + R17}{2} + \frac{2 \cdot G13 - G9 - G17}{2} \quad (\text{A.11})$$

For the case that both gradients are equal, the red value is given as the mean of Equation A.10 and Equation A.11.

$$R13_{\Delta N \Delta P} = \frac{R13_{\Delta N} + R13_{\Delta P}}{2} \quad (\text{A.12})$$

In Equation A.10 and Equation A.11 the estimation consists again of a first low pass term orthogonal to the edge direction and a second edge enhancement term as composed for the green channel in Equation A.3 and Equation A.4 [87]

A.1.3 Improved Gradient Estimation

Even though the method presented in Section A.1.2 can determine the edge direction correctly in most cases, there are still some cases where the method is not able to find the correct edge direction and therefore produces artifacts. In [87] a new method is presented, which increases the performance of the edge direction estimation and therefore reduces the number of artifacts. The key idea of the proposed method is to improve the conventional gradients by adding color correlation information to those gradients. The assumption of this approach is that in real world image the contrast of color difference (difference R to G and B to G) should be small over a small area. Therefore the correct interpolation direction should guarantee that the color difference is small over a small area.

Using the same notation as in Fig.5.6b, the color differences of $R7$, $R9$, $R17$ and $R19$ are defined as (taking $R7$ as example)

$$K_H(R7) = R7 - G7' = R7 - \frac{G6 + G8}{2} \quad (\text{A.13})$$

$$K_V(R7) = R7 - G7' = R7 - \frac{G2 + G12}{2} \quad (\text{A.14})$$

If the red channel or the blue channel is taken for the calculation the colour difference is not important, there is only one difference that can be computed at a certain pixel. As stated above, the difference $R - G$ and $B - G$ should behave in similar ways in small area.

Let N be the set of $N = \{R7, R9, R17, R19\}$ and $(Q, Q') \in N \times N$ the Cartesian product of the set N with itself, which means that if N has cardinality 4, then (Q, Q') has cardinality 16. Using these definition, the difference of the $K(R)$ is calculated as

$$M_H = \sum_{(Q,Q')=N \times N} |K_H(Q) - K_H(Q')| \quad (\text{A.15})$$

$$M_V = \sum_{(Q,Q')=N \times N} |K_V(Q) - K_V(Q')| \quad (\text{A.16})$$

To estimate the correct edge direction, the color difference should be small and therefore the M value should be small as well. As the M parameter behaves in the same way as the conventional gradient, the two values are added to get a new estimate given as

$$\Delta H = M_H + L_H \quad (\text{A.17})$$

$$\Delta V = M_V + L_V \quad (\text{A.18})$$

Green Pane Estimation For the green pane estimation, the estimators given in (A.17) and (A.18) replace the decision estimators used to determine whether (A.3), (A.4) or (A.5) need to be applied for the green pane reconstruction. Everything else in the green pane reconstruction stays the same as described in Section-A.1.2. This new method only provides a better estimation of the edge direction, beside that, nothing else changes [87].

Red/Blue Pane Estimation The estimation of the ΔN and the ΔP was done in the same way as described in Section-A.1.2, but also the new estimators given in (A.17) and (A.18) were adapted for the diagonal case and tested and compared to the original estimator's results.

Implementation Details

Figure A.1 shows the schematic of the two blocks designed for gradient based demosaicing: the green pane interpolator (Figure A.1a) and the red/blue pane interpolator (Figure A.1b). Even though not explicitly specified on Figure A.1, the output of the green pane interpolator is fed to the input of the red/blue pane interpolator.

A.2 Hybrid Method

Even though gradient based demosaicing methods perform well, they are computationally expensive and badly suited for implementation on a processor. The method proposed by [88] aims at decreasing the computational effort by only using a linear filter while working on a

Appendix A. Analysis and implementations of different Bayer Demosaicing and Automatic White Balancing Methods

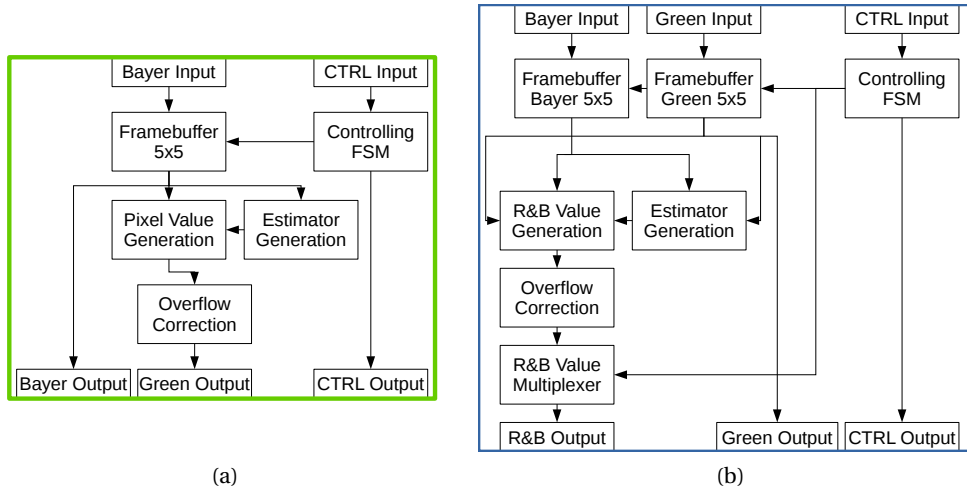


Figure A.1 – Schematic of the blocks for a gradient based Bayer to RGB conversion system

5×5 window.

The main idea of this method is to use a bilinear interpolation with a gradient correction term. Then, when estimating the green value at a red pixel's location, this leads to

$$\hat{g}(i, j) = \hat{g}_B + \alpha \Delta_R(i, j) \quad (\text{A.19})$$

where \hat{g}_B is the linear interpolation term, α the weighting factor and $\Delta_R(i, j)$ the gradient of the red channel at that location, which is given, if the central red pixel is assumed to have coordinates $(0, 0)$, as

$$\Delta_R(i, j) \triangleq r(i, j) - \frac{1}{4} \sum_{(m,n) \in \{(0,-2), (0,2), (-2,0), (2,0)\}} r(i+m, j+n) \quad (\text{A.20})$$

For interpolating red at green pixel, the interpolation is given as

$$\hat{r}(i, j) = \hat{r}_B(i, j) + \beta \Delta_G(i, j) \quad (\text{A.21})$$

For interpolating red at blue pixels, the formula is

$$\hat{r}(i, j) = \hat{r}_B(i, j) + \gamma \Delta_B(i, j) \quad (\text{A.22})$$

For the determination of the gain factors $\{\alpha, \beta, \gamma\}$ a Wiener approach was used. $\{\alpha, \beta, \gamma\}$ were determined as to minimize the mean-square error of a data set (the Kodak test images [88, 99]). Then the gain factors were approximated to be integer multiples of powers of 1/2, which lead to the final result of $\alpha = 1/2$, $\beta = 5/8$ and $\gamma = 3/4$. With this approximation the constructed FIR filter was within a 5% margin in terms of the mean-square error of the optimal Wiener filter with a 5×5 region of support [88]. The hybrid method explained in [88] is implemented as illustrated in Fig.-A.2.

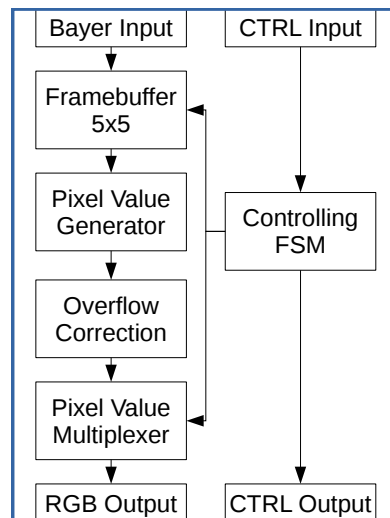


Figure A.2 – Block diagram of the hybrid Bayer to RGB converter architecture

A.3 Automatic White Balancing

One of the most amazing features of the human visual system is color consistency. While looking at the same object under different illumination conditions, the human perceives the colors of the objects in a relatively constant manner and is not very dependent on the properties of the illumination source.

A similar behavior is highly desirable in digital image capturing devices. This is achieved by post processing the captured RGB values in order to take the illumination source into account. Most people have experienced that for example, a picture taken under the illumination of a domestic tungsten lamp, whose color temperature is around 3000 K, appears reddish, while

Appendix A. Analysis and implementations of different Bayer Demosaicing and Automatic White Balancing Methods

taking the same picture with daylight, which has a color temperature of above 6000 K, the picture appears blueish. The effect of the illumination source becomes even more important using LEDs as illumination source due to their singular spectra. Figure A.3 shows the spectral power distributions of different light sources in the visible range. It can be seen that their spectra differ a lot in the visible range and therefore white balancing is an important part of any image processing pipeline.

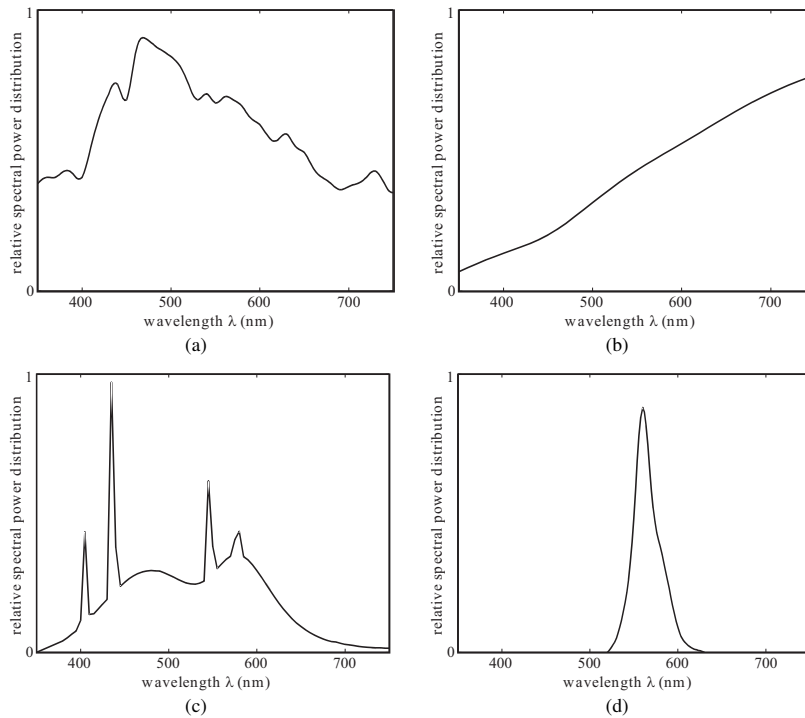


Figure A.3 – Spectral power distribution of various common types of illuminations: (a) sunlight, (b) tungsten light, (c) fluorescent light, and (d) LED [100]

Another effect of white that needs to be taken into account is that the color selectivity of the sensor cells of the human eye differ from the color selectivity of most Bayer filters. Fig.A.4 illustrates this fact: while the wavelength of the peaks of the spectral sensitivities of the Bayer filter coincide more or less with their natural counterpart, their selectivity differs. Especially a blue passing filter is a lot less selective than the S-cone it should represent.

To sum up: white balancing aims at compensating for the light temperature of the light source as well as to fight the imperfections of the Bayer filter [100].

White balancing can either be done manually or automatically. While in most high-end cameras there is at least an option to do the white balancing manually, most consumer devices do it automatically. In the rest of this Section, two simple methods for automatic white balancing is described and their FPGA implementations are presented.

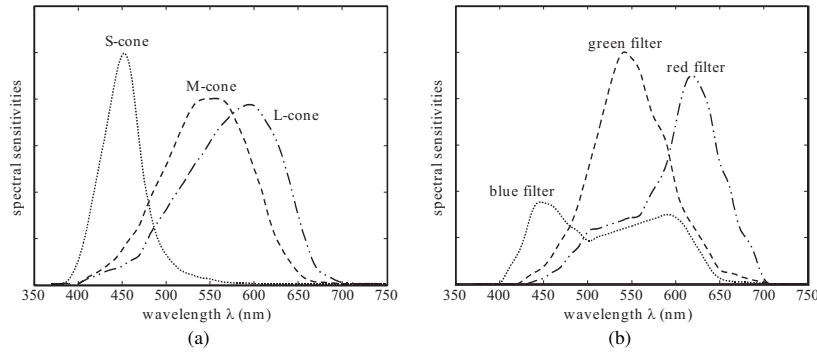


Figure A.4 – Spectral sensitivities of: (a) the three types of cones in a human eye, and (b) a typical digital camera [100]

A.4 Gray World Assumption

The basic assumption used for this method is that in average, the red, green and blue channels of a scene should roughly be equal. In other words, placing the mean value of each channel in each pixel would result in an all gray picture, which is where the assumption’s name comes from.

The method then works as follows. Assume a $n \times m$ full color image which is given by $RGB_{sensor}(x, y)$ for any pixel. As a first step in this method the average of each channel is computed as

$$R_{avg} = \frac{1}{n \cdot m} \sum_{x=1}^n \sum_{y=1}^m R_{sensor}(x, y) \tag{A.23}$$

$$G_{avg} = \frac{1}{n \cdot m} \sum_{x=1}^n \sum_{y=1}^m G_{sensor}(x, y) \tag{A.24}$$

$$B_{avg} = \frac{1}{n \cdot m} \sum_{x=1}^n \sum_{y=1}^m B_{sensor}(x, y) \tag{A.25}$$

If the three averages are identical, the gray world assumption already is satisfied. However, this normally is not the case. Taking the green channel as the fixed one, two gains $\hat{\alpha}$ and $\hat{\beta}$ need to be calculated for the red and the blue channel respectively.

Appendix A. Analysis and implementations of different Bayer Demosaicing and Automatic White Balancing Methods

$$\hat{\alpha} = \frac{G_{avg}}{R_{avg}} \quad (A.26)$$

$$\hat{\beta} = \frac{G_{avg}}{B_{avg}} \quad (A.27)$$

Then, the image which satisfies the gray world assumption is given as

$$\hat{R}_{sensor}(x, y) = \hat{\alpha} R_{sensor}(x, y) \quad (A.28)$$

$$\hat{G}_{sensor}(x, y) = G_{sensor}(x, y) \quad (A.29)$$

$$\hat{B}_{sensor}(x, y) = \hat{\beta} B_{sensor}(x, y) \quad (A.30)$$

The gray world method is quite effective in practice except in situations where one color dominates for example when a great portion of the image is covered by the blue sky. In that case, the gray world method fails and distorts the image [100].

A.5 White Patch

The white patch assumption, assumes that within a picture, the brightest point often is due to reflectance of a glossy surface, which tends to reflect the actual color of the light source. In order to avoid outliers, before determining the maximum value of each channel, the pixel needs to be low-pass filtered. After that, the maximum intensity is determined as

$$R_{max} = \max(R_{sensor}(x, y)) \quad (A.31)$$

$$G_{max} = \max(G_{sensor}(x, y)) \quad (A.32)$$

$$B_{max} = \max(B_{sensor}(x, y)) \quad (A.33)$$

Then, similar to the gray world assumption, the gains are calculated as the ration of the green channel to the other channel.

$$\hat{\alpha} = \frac{G_{max}}{R_{max}} \quad (A.34)$$

$$\hat{\beta} = \frac{G_{max}}{B_{max}} \quad (A.35)$$

The corrected image which satisfies the white patch assumption is then given as

$$\hat{R}_{sensor}(x, y) = \hat{\alpha}R_{sensor}(x, y) \quad (A.36)$$

$$\hat{G}_{sensor}(x, y) = G_{sensor}(x, y) \quad (A.37)$$

$$\hat{B}_{sensor}(x, y) = \hat{\beta}B_{sensor}(x, y) \quad (A.38)$$

However, for most images the gray world and the white patch method produce different results. In other words, the corrected image can rarely satisfy both assumptions.

A.6 Implementation Details

The structure of the block which performs automatic white balancing can be divided into two sub-blocks (for the case of the white and gray world method): one which is specific to the actual method used and a second which is general and can be used in both cases.

Fig.A.5 shows the assumption specific block of the automatic white balancing functional block: Fig.A.5a shows the block diagram for the gray world assumption and Fig.A.5b shows the block diagram for the white patch method. For the gray world method, an accumulator and a divider is used to calculate the mean of each channel for one image. When the divider block has finished calculating, the FSM triggers the start of the unspecific part of the architecture. For the white patch method, a low pass filter is used in front of a block which detects the maximal value of each channel in one image. Again, when the frame has ended and the maximum is known, the FSM starts the computations of the unspecific block.

Fig.A.6 shows the functional part which both, the gray world and the white patch method, share. The main point to note is that the actual input-output processing happens in the upper part of the design block diagram, where the RGB values are entered and then if necessary, multiplied by the gain and then the values are sent to the output. It is important to note that the output of the gain multiplier needs to be big enough to detect overflows in order to be able to correct them to produce correct 8-bit values. The lower part of the block diagram is only

Appendix A. Analysis and implementations of different Bayer Demosaicing and Automatic White Balancing Methods

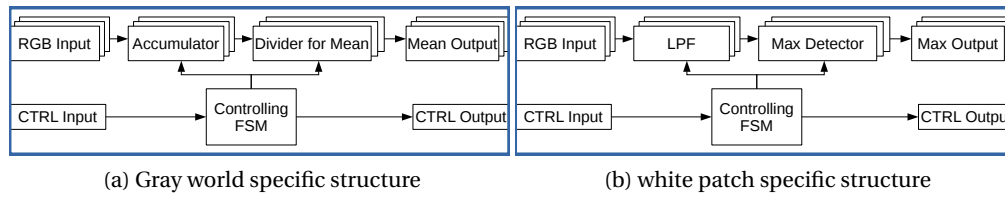


Figure A.5 – Block diagram of a part of the gray and the white patch automatic white balancing algorithm

updated at the end of each image and stays constant during the course of an image.

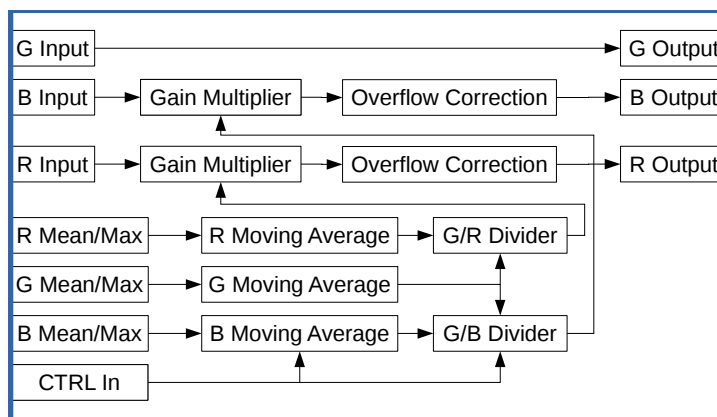


Figure A.6 – Block diagram of the common part of the gray and white patch automatic white balancing algorithm

A.7 Results

The methods are implemented and tested for the Virtex7 based VC707 board by adapting the single camera interface system described in Section-5.1.

A.8 Bayer to RGB

First, it needs to be stated that the images generated by using the color correlation criterion for the red and blue channel do not differ from the images generated by only using color correlation for an improved edge estimation on the green channel. Therefore, only the version with the color correlation on the green channel is discussed here as the other version has exactly the same performance but a higher hardware complexity, e.g. the additional hardware does not achieve anything.

All the images have been cropped to the point where the border effects due to wrapping around of the moving window are no longer visible.

A.8.1 Optical Analysis

For the optical analysis, the 19th Kodak test image was chosen as it contains two of the major difficulties for reconstruction: a plain with fine textures on the tower, and small regular structures as in the fence and the front of the house on the left side.

Fig.A.7 shows the results of the Bayer-to-RGB conversion using a 3×3 sized window on the 19th Kodak test image: on the left (Fig.A.7a) is the original image and on the right (Fig.A.7b) is the reconstructed image using the bilinear interpolation. The low-pass effect of the bilinear interpolator is clearly visible at the front of the tower where a lot of the textures are lost and the lawn has lost a lot of its details as well. Furthermore, the bilinear filter produces artifacts which are most prominent at the fence on the left side of the telescope and at the front of the house on the left side of the image. Both areas have tiny lines where artifacts are produced. In one case they are approximately horizontal and in the other they are more or less vertical.



Figure A.7 – Interpolation results using a 3×3 window

Fig.A.8 shows the result of the Bayer-to-RGB conversion using a 5×5 window on the 19th Kodak test image. Fig.A.8a shows the original image and Fig.A.8d shows the result of the linear, but gradient aided reconstruction filter. The textures at the tower as well as the lawn are reconstructed much better than by using the bilinear interpolator. However, this method still produces a lot of reconstruction artifacts which are still at the same locations as they were in the bilinear case: at the fence and the front of the house on the left side.

In order to fight those artifacts, a non-linear filter needs to be used. Two of this kind were tested: Adam and Hamilton's method and the color correlation aided improvement of their method. Their results can be seen in Fig.A.8b and Fig.A.8c respectively. Both methods were able to reconstruct the image with much less artifacts than all the previous methods while preserving the details of the textures. The only place where both methods still have artifacts is on the

Appendix A. Analysis and implementations of different Bayer Demosaicing and Automatic White Balancing Methods

left of the telescope: at this place the vertical lines are too narrow for a good reconstruction. However, it is visible that the color aided version produces much less artifacts than Adam and Hamilton's method, even though, this method already performs quite well. In most images Adam and Hamilton's method would work perfectly, it really takes an extreme case of texture for this method to fail. But if this case is too extreme, even the improved method cannot avoid to produce artifacts, although there are very few.

In comparison, the color correlation aided gradient seems to be the most promising candidate for most cases. The reason for this choice is that it produces less artifacts than Adam and Hamilton's method while it does not have a much bigger complexity than Adam and Hamilton's method; only the additional term for the estimator needs to be computed. If the system is not capable to perform this amount of computations, the hybrid method should be used as it does not have much more computations than the bilinear method but keeps the fine details of textures much better while producing approximately the same amount of artifacts. The only downside of the hybrid method compared to the 3×3 bilinear interpolator is that it needs double the number of memory for the frame buffer. But in most modern image capturing systems, memory is not a critical resource any more or at least not in this stage as only a few lines of the image need to be stored in the frame buffer while for outputting the whole image needs to be transferred and possibly stored somewhere.

A.8.2 Measurements

A common method to evaluate numerically the performance is the Peak Signal to Noise Ratio (PSNR). For 8 bit images, the PSNR is defined as follows

$$MSE = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (f_t(i, j) - f_r(i, j))^2 \quad (\text{A.39})$$

$$PSNR = 10 \cdot \log_{10} \left(\frac{(2^8 - 1)^2}{MSE} \right) \quad (\text{A.40})$$

where $f_t(i, j)$ is a pixel in the tested image and $f_r(i, j)$ a pixel in the reference image. The larger the PSNR value, the smaller is the error and therefore the better is the algorithm.

Table-A.1 shows the PSNR values of the 19th Kodak test image. As expected and predicted by literature, the color correlation aided gradient based method performed the best, tightly follow by its predecessor, the gradient based method proposed by Adam and Hamilton. With a small gap in terms of PSNR follows the hybrid method, which still performs fairly well but is a lot less complex than the gradient based methods. The difference then to the result of the bilinear method is twice as big compared to the difference from the hybrid method to



(a) Original image

(b) Adam and Hamilton's method



(c) Improved gradient estimation

(d) Hybrid method

Figure A.8 – Bayer CFA interpolation results using a 5×5 window

the gradient based methods. Furthermore, it is to note that the reconstruction of the green channel can always be done much more accurately than the other channels as there are twice

Appendix A. Analysis and implementations of different Bayer Demosaicing and Automatic White Balancing Methods

as many pixels which measure the green channel.

Table A.1 – PSNR comparison of the Kodak image set 19th image for different Bayer CFA interpolation methods

Method	PSNR R [dB]	PSNR G [dB]	PSNR B [dB]
Bilinear	17.39	22.21	17.63
Hybrid	23.33	27.71	22.89
Adam & Hamilton	26.95	28.86	27.09
Colour Correlation	27.27	29.31	27.46

Even though these values show the same behavior as predicted by literature, they are not exactly the same as in the papers where the same image has been tested. There are mainly two reasons for this. First, the authors did not specify how their systems behaved at the borders at all. The wrapping around at the borders might be one possible reason for the different behavior. Second, the bit width provided by the reference image was only 8-bits although the system has been designed to work with 10-bits. At the output, the internal signals, which are mostly wider than 10-bits have been cropped again to 8-bits. Therefore there are multiple sources where precision could have been lost. The most important one is at the input: if the authors of the other papers had the 10-bit Bayer values, it is clear that they produce better results. Furthermore, the resolution used for their outputted RGB was not specified anywhere and could be bigger than the 8-bits used in this work.

A.8.3 Resource Usage

Table-A.2 shows the resource usage of the different algorithms for Bayer-to-RGB conversion implemented on the FPGA. The first column indicates the name of the algorithm, the second column indicates the size of the Block Ram used, the third column indicates the number of D-Flip-Flops used, the fourth column indicates the number of Look-Up-Tables (LUTs) used, the fifth column indicates the number of slices (usually a combination of typically 4 LUTs and 4 DFFs) used and the sixth columns indicates the number of DSP-slices used, which are hard coded multipliers in this case.

Table A.2 – Resource usage of the different Bayer-to-RGB algorithms

Method	BRam [kB]	DFFs	LUTs	Slices	DSPs
Bilinear	4	224	285	143	0
Hybrid	8	501	561	291	4
Adam & Hamilton	20	1174	1200	669	0
Colour Correlation	20	1324	2181	898	0

The increasing size of the used block ram can be explained by the increase of either the working window (bilinear to hybrid) and the increase in the number of used windows (one to two from hybrid to gradient based methods). The number of DFFs scales in a similar fashion and the number of used slices as well. However, the number of used LUTs is nearly doubled

when comparing Adam and Hamilton's method and its improvement, which uses the color correlation to improve the gradient. The reason for this is that the calculation of the color correlation is costly as the sums of 16 elements needs to be calculated twice. The use of the DSP slices was not intended (as the bit width of the operands in multiplications is rather small) but the synthesizer decided to use them in one case.

A.9 Automatic White Balancing

Due to the lack of a adequate metric for evaluating automatic white balancing, only an optical analysis has been done to test the performance of the different algorithms.

A.9.1 Optical Analysis

Fig.A.9 shows the image of a baby, photographed with a tungsten lamp as light source. The original image (Fig.A.9a) appears strongly blueish due to the color heat of the illumination source. The white patch based white balancing method (Fig.A.9b) using a 4×4 sized low-pass filter is able to remove this blueish tone partially but not completely. In contrary, using the gray world method in Fig.A.9c, the blue tone of the image is nearly completely removed and the image appears without any distortions. This leads to the conclusion, that the gray world criterion has a bigger impact on our visual perception than the white patch criterion and most state-of-the-art algorithms for white balancing algorithms rely on this criterion. However, as shown in the following, this poses a major problem for our application.

The gray world method generally performs quite well (and any other method which is based on this criterion, which corresponds to a big portion of modern algorithms) however, this is only the case when no color takes up a big part of the picture. For example, all gray world based methods struggle when a big part of the image is taken by the blue sky.

Unfortunately, the scenario of the endoscopy represents such a case where the gray world assumption is not fulfilled any more: inside the human bowels the environment contains dominantly red color. Fig.A.10 shows the live image of an endoscopy which already has been white balanced in Fig.A.10a. Ideally, applying the algorithm at this stage should not alter the image any more by a lot as it already is white balanced. However, as seen in Fig.A.10c, the gray world method fails in this case and rends the image to its assumption: it makes it gray. On the other hand, the white patch based algorithm does not alter the image visibly, therefore this algorithm seems to be more robust.

As a result, for the application in an endoscopy, the gray world method can be discarded. The white patch method might work, especially as it acts on reflections of light which are usually present during an endoscopy, but it is not as efficient and powerful as the gray world method, e.g. it is only able to correct for slight unbalancing, as soon as there are big distortions, the white patch method is not able to correct them.

Appendix A. Analysis and implementations of different Bayer Demosaicing and Automatic White Balancing Methods



(a) Original image

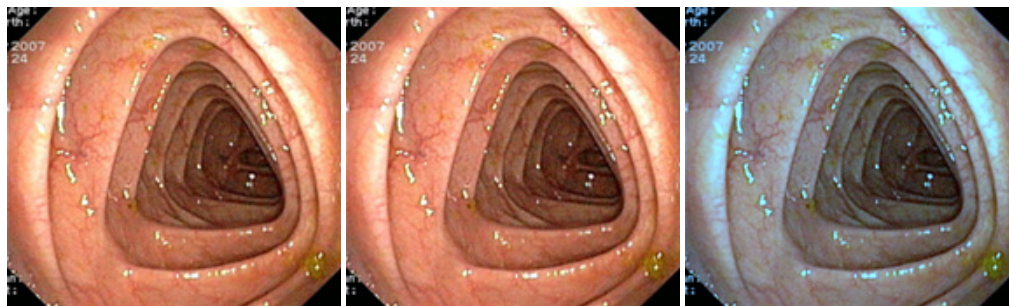


(b) White patch 4 × 4



(c) Gray world

Figure A.9 – White balance comparison for images taken under a tungsten light source



(a) Original image

(b) white patch 4 × 4

(c) Gray world

Figure A.10 – Image taken in human guts with balanced light

A.9.2 Resource Usage

Table-A.3 shows the resource usage of the different algorithms for automatic white balancing implemented on the FPGA. The first column indicates the name of the algorithm, the second column indicates the size of the Block Ram used, the third column indicates the number of D-Flip-Flops used, the fourth column indicates the number of Look-Up-Tables (LUTs) used, the fifth column indicates the number of slices (usually a combination of typically 4 LUTs and

A.10. Visual Results from the implemented single camera system

4 DFFs)) used and the sixth column indicates the number of DSP-slices used, which are hard coded multipliers in this case.

Table A.3 – Resource usage of the different AWB algorithms

Method	BRam [kB]	DFFs	LUTs	Slices	DSPs
Gray World	10	975	1890	743	43
White Patch 4×4	22	1460	1498	668	16

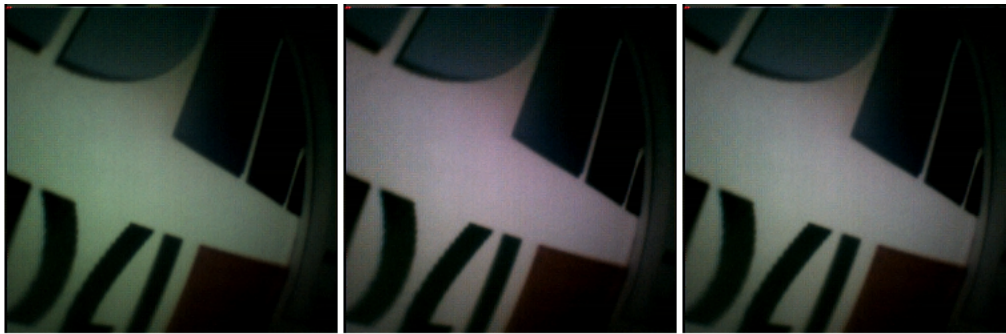
Comparing the two methods in terms of resources used, it can be stated that the gray world method needs more resources for computations while the white patch method needs more memory. 16 DSP slices are used for the division to calculate the gains, e.g. one of those divider uses eight DSP slices. The additional 27 are used to calculate the mean of each channel of the picture for the gray world method (9 DSP slices per divider), because the image size does not represent a power of two where the division could be done easily by shifting. The additional block ram for the white patch method is used for the low-pass filter at the beginning, which needs a frame buffer.

A.10 Visual Results from the implemented single camera system

The system using no automatic white balancing and using a bilinear interpolator for Bayer-to-RGB conversion compared to a system which uses both automatic white balancing methods and the color correlation aided gradient based method for Bayer-to-RGB conversion.

Fig.A.11 shows a comparison between the different methods using a sample image taken with the actual system. When no automatic white balancing is performed (Figure A.11a), the image have green channel dominance. Fig.A.11b shows the resulting image when the gray world method is used for automatic white balancing. Fig.A.11c shows the image produced using the white patch method with a 4×4 low pass filter. This method was perceived to work the best out of the three, as it is able to remove most of the greenish tone without introducing any new distortions like the gray world method did.

Appendix A. Analysis and implementations of different Bayer Demosaicing and Automatic White Balancing Methods



(a) Image without AWB

(b) Gray world AWB

(c) White patch 4×4 LPF AWB

Figure A.11 – Image taken with a live system with different AWB methods

B Details of sub blocks FPGA to ASIC conversion

B.1 Details of the dual clock FIFO design

The challenging part about this design is about generating FIFO pointers and supplying a reliable way to determine full and empty status [101, 102]. No increment-decrement counter can be utilized in an asynchronous FIFO because two different clocks would be needed to control it. Instead, write and read pointers need to be compared: the first one always points to the following word to be written whereas the second always points to the current to be read. The FIFO then can be:

- **empty** on reset or when the read pointer catches the write pointer, having read the last word.
- **full** when both pointers are equal but the write pointer has wrapped around and caught up the read pointer.

In order to make a differentiation between these two situations an extra bit is supplemented to each pointer: when one of them wraps around, the MSB is incremented while all other bits are set to 0. Thus, the FIFO is **empty** when both pointers are equal, including the MSBs, while it is **full** when both pointers, except the MSBs, are equal.

Clock domain crossing issues

Metastability, data loss and incoherency are the three leading issues which can occur in a clock domain crossing, i.e. the transfer of a signal from a flip-flop driven by a clock to one driven by another clock [103].

Appendix B. Details of sub blocks FPGA to ASIC conversion

Metastability It is the kind of problem which takes place when a transition on a signal initiates from the first clock domain which happens very close to the edge of the second clock. This process leads to set up or hold violations at the destination flip-flop. Thus, the sampled output is unstable and it may or may not settle down to a stable value before the next clock edge arrives.

Consequences might be: high current flow, high propagation delays or entering in unknown states. A common solution for this issue is the use of **multi-flop synchronizers** as shown in Fig.B.1 in the destination domain, that allow sufficient time for the oscillations to settle down and ensure a stable output.

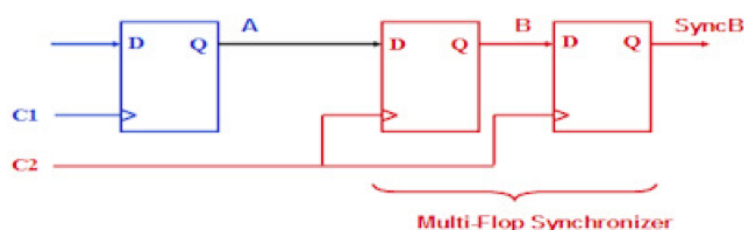


Figure B.1 – Multi-flop synchronizer [103].

Data loss As a result of metastability whenever a transition on the source takes place, it may not be sampled by the destination domain at the first clock. Data loss occurs when a transition on a source signal is not captured by the destination clock.

In order to prevent data loss, at least one destination clock edge with no setup or hold violations has to be available after every source signal transition.

Data incoherency Data incoherency is related to the situations when the transfer of signals occur from one clock domain to another and each of these signals synchronized by utilizing a multi-flop stage. When multiple signals changes simultaneously and at those times the two clock edges are close together, at the destination domain some of these signals may be sampled during first clock cycle whereas the others during the second. Such an incident comes true due to metastability. As a result of metastability, an invalid combination of signals is produced which lead to data incoherency as illustrated in Fig.B.2.

To handle this problem, the design of the circuit needs to be arranged in such a way that while moving from one state to another, there should be only one bit change. Thus, the bus will change into a new value or continue with the original one.

In order to accomplish this, the bus is required to be Gray-encoded, and as a result of this Gray counters have been accommodated in the developed FIFO design.

B.2. Details of serial communication interface choice

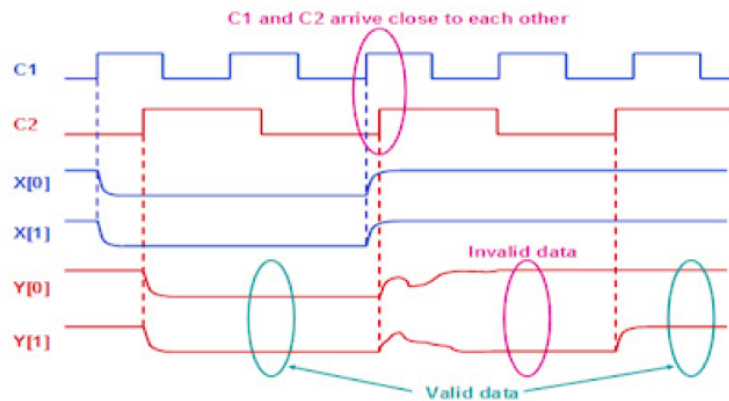


Figure B.2 – Example of data incoherency [103].

B.2 Details of serial communication interface choice

Serial interfaces are typical in embedded system peripherals and are preferred to parallel ones due to their low pin counts, so they require less space and they are cheaper to be implemented. Furthermore, it is often likely to drive the serial links with a faster clock rate than the parallel ones to attain a higher data rate, thanks to the reduced clock skew and crosstalk issues.

The main terms related to serial protocols are listed below [104]:

- *Synchronous* buses send data with clock while *asynchronous* ones does not.
- On a bus, one device, the master, controls one or more slaves. Usually *master/slave buses* are synchronous and the master supplies the clock. Multi-master buses are possible, but an arbitration scheme is necessary to solve conflicts when more than one master attempt to access the bus at the same time.
- In a *point-to-point* interface no masters or slaves are present (peer relation) and they are usually asynchronous.
- In a *multi-drop* interface there are several receivers and one transmitter while a *multi-point* interface describes a bus with several transceivers. It is different from the multi-drop since it allows bidirectional communication on the same set of wires.
- On a *full-duplex* bus, data can be sent and received simultaneously while on a *half-duplex* one, data can be sent or received but not at the same time.

A comparison between the selected serial communication interfaces (SCI) is shown in Table B.1.

Appendix B. Details of sub blocks FPGA to ASIC conversion

Table B.1 – Comparison for different serial communication interfaces

SCI	Type	Sync/Async	Signaling
RS-232 (UART)	point-to-point	async	single-ended unbalanced line
RS-422 (UART)	multi-drop	async	differential balanced line over twisted pair
RS-485 (UART)	multi-point	async	differential balanced line over twisted pair
I ² C	multi-master	sync	single-ended unbalanced line
SPI	multi-master	sync	single-ended unbalanced line

SCI	Pins required	Duplex	Max data rate
RS-232 (UART)	2 (TX,RX)	full	20 Kbps
RS-422 (UART)	2 (D+,D-)	half	10 Mbps
RS-485 (UART)	2 (D+,D-)	half	10 Mbps
I ² C	2 (SDA, SCL)	half	3.4 Mbps
SPI	4 (SCLK,MOSI,MISO,SS)	full	1 Mbps

SCI	Communication distance	Advantages	Disadvantages
RS-232 (UART)	off-board	Good solutions since allow the connection ASIC - PC.	Async serial ports require hardware overhead.
RS-422 (UART)	off-board		
RS-485 (UART)	off-board		
I ² C	on-board	Simple implementation, both in HW and SW. No specific connectors. Possibility to perform clock stretching.	Limited range of speeds supported. Open-drain transistors necessary to pull high the lines.
SPI	on-board	Full duplex communication. HW can be a simple shift register.	Number of wires and pins required increases with multiple slaves.

B.3 The memory generator used for custom memory blocks for ASIC design

An example view of the ARM Artisan memory generator GUI.

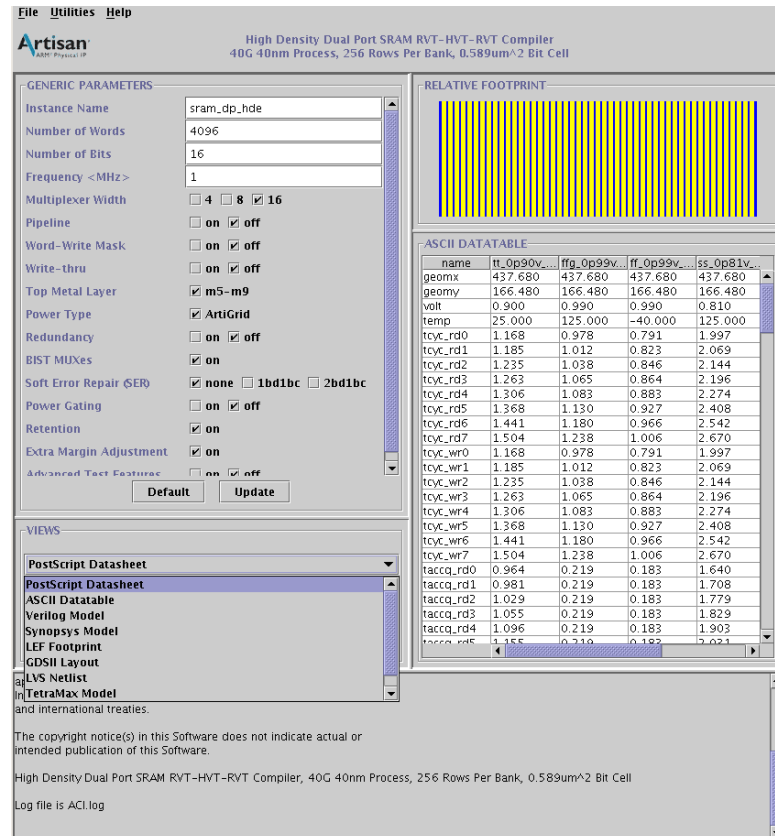


Figure B.3 – ARM Artisan Physical IP GUI.

B.4 I²C Communication Protocol

The main bus rule implies that when data or addresses are sent, the SDA signal is allowed to change only when the SCL line is low, and when the clock is high, data should remain stable. Each I²C communication introduced by a master begins with a start condition and ends with a stop condition, special messages that break this rule. Effectively, both transitions on SDA happen when the clock is at logic 1: the start is a falling transition while the stop is a rising transition. As illustrated in Fig.B.4 the communication proceeds as follow [105–107]:

1. A master sends a **start condition** on SDA and provides on SCL the clock signal, used by all the ICs as a reference time to sample the data at the rising edge. The bus is considered as busy.

Appendix B. Details of sub blocks FPGA to ASIC conversion

2. The master provides in serial form the **7-bit address** of the slave. It aims to communicate with and a **data direction bit**, telling whether it wants to write (0) or read (1) data.
3. The slave sends an **acknowledge bit** (SDA low) to give a feedback that it has recognized its address and it is ready to communicate.
4. Data is transferred as **8-bit words** without limitations on the number but after each byte the expectation of the transmitter is an acknowledge bit from the receiver. Both the address and the words are transmitted with the MSB first.
5. The communication ends when the master sends a **stop condition** on SDA and the bus is considered as free. On the other hand, the bus stays busy if a **repeated start condition** is generated instead. In this way the master can change the data transfer direction or begin the communication with another slave.

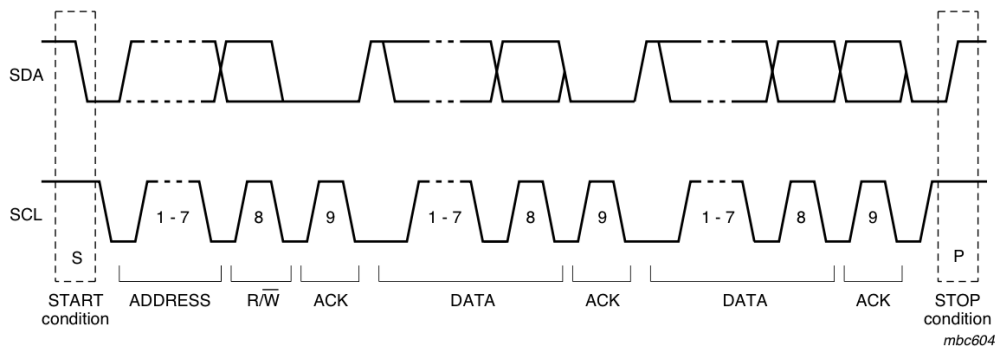


Figure B.4 – I²C data transfer [106].

C Example video output links for the miniaturized compound eye imaging system

In this part, the link for two example video from the system is presented. The videos are at 1080x1080p resolution, 24 fps. The field of view is $190^\circ \times 190^\circ$ to show the boundaries of the visible area by the system. In the first video, it can be observed that even small polyps behind the folds can be captured by the proposed system. The second video is starting from outside of the polyp and goes through the bowel. Both of the videos are 30 seconds long. The name of the videos are: for the first one: "bowel190x190_1080x1080_1.avi" and for the second video: "bowel190x190_1080x1080_2.avi"

The link for the folder contains the two videos:

<https://drive.google.com/folderview?id=0BxWB5nAMFJ8sVVpiRDJtWk1HT0k&usp=sharing>

Bibliography

- [1] Pointgrey ladybug5. <http://www.ptgrey.com/ladybug5-360-degree-usb3-spherical-camera-systems>. Accessed: 2015-09-06.
- [2] Nokia ozo camera. <https://ozo.nokia.com/>. Accessed: 2015-09-06.
- [3] Altia Systems panacast 2 camera. <http://www.getpanacast.com/>. Accessed: 2015-09-06.
- [4] Serveball squito throwable camera. <http://www.serveball.com/>. Accessed: 2015-09-06.
- [5] Panano panono camera. <https://www.panono.com/#/en/home#at-a-glance>. Accessed: 2015-09-06.
- [6] Photosiphoneauto360. <http://photoshipone.com/auto360/>. Accessed: 2015-09-06.
- [7] Gopano gopano micro. <http://www.gopano.com/products/gopano-micro-iphone5>. Accessed: 2015-09-006.
- [8] Kogeto kogeto-dot. <http://kogeto.com/dot.html>. Accessed: 2015-09-006.
- [9] Bubblepix bubblescope. <https://bubblepix.com/>. Accessed: 2015-09-006.
- [10] N. Franceschini. Small brains, smart machines: From fly vision to robot vision and back again. *Proceedings of the IEEE*, 102(5):751–781, May 2014.
- [11] Simon Thibault, Jocelyn Parent, Hu Zhang, and Patrice Roulet. Design, fabrication and test of miniature plastic panomorph lenses with 180 field of view. In *International Optical Design Conference*, pages 92931N–92931N. International Society for Optics and Photonics, 2014.
- [12] Lindsey A. Torre, Freddie Bray, Rebecca L. Siegel, Jacques Ferlay, Joannie Lortet-Tieulent, and Ahmedin Jemal. Global cancer statistics, 2012. *CA: A Cancer Journal for Clinicians*, 65(2):87–108, 2015.
- [13] Ian M. Gralnek, David L. Carr-Locke, Ori Segol, Zamir Halpern, Peter D. Siersema, Alan Sloyer, Jay Fenster, Blair S. Lewis, Erwin Santo, Alain Suissa, and Meytal Segev. Comparison of standard forward-viewing mode versus ultrawide-viewing mode of

Bibliography

- a novel colonoscopy platform: a prospective, multicenter study in the detection of simulated polyps in an in vitro colon model (with video). *Gastrointestinal Endoscopy*, 77(3):472 – 479, 2013.
- [14] Moshe Rubin, Konika P. Bose, and Sang H. Kim. Mo1517 successful deployment and use of third eye panoramic™ a novel side viewing video {CAP} fitted on a standard colonoscope. *Gastrointestinal Endoscopy*, 79(5, Supplement):AB466 –, 2014. {DDW} 2014ASGE Program and Abstracts {DDW} 2014ASGE Program and Abstracts.
- [15] Nazia Hasan, Seth A. Gross, Ian M. Gralnek, Mark Pochapin, Ralf Kiesslich, and Zamir Halpern. A novel balloon colonoscope detects significantly more simulated polyps than a standard colonoscope in a colon model. *Gastrointestinal Endoscopy*, 80(6):1135 – 1140, 2014.
- [16] Nathan Gluck, Sigal Fishman, Alaa Melhem, Sharon Goldfarb, Zamir Halpern, and Erwin Santo. Su1221 aer-o-scope™, a self-propelled pneumatic colonoscope, is superior to conventional colonoscopy in polyp detection. *Gastroenterology*, 146(5, Supplement 1):S-406 –, 2014. 2014 {DDW} Abstract.
- [17] Toshio Uraoka, Shinji Tanaka, Takayuki Matsumoto, Takahisa Matsuda, Shiro Oka, Tomohiko Moriyama, Reiji Higashi, and Yutaka Saito. A novel extra-wide-angle-view colonoscope: a simulated pilot study using anatomic colorectal models. *Gastrointestinal Endoscopy*, 77(3):480 – 483, 2013.
- [18] Michael F. Land and Dan-Eric Nilsson. *Animal eyes*. Oxford University Press, 2012.
- [19] Michael F. Land. Visual acuity in insects. *Annual review of entomology*, 42(1):147–177, 1997.
- [20] EPFL laboratory of intelligent systems- curvace. <http://lis.epfl.ch/curvace>. Accessed: 2015-09-06.
- [21] Luke P. Lee and Robert Szema. Inspirations from biological optics for advanced photonic systems. *Science*, 310(5751):1148–1150, 2005.
- [22] Sudipta N. Sinha. Pan-tilt-zoom (ptz) camera. In Katsushi Ikeuchi, editor, *Computer Vision*, pages 581–586. Springer US, 2014.
- [23] Kenro Miyamoto. Fish eye lens. *JOSA*, 54(8):1060–1061, 1964.
- [24] Pascal Ferrat, Christiane Gimkiewicz, Simon Neukom, Yingyun Zha, Alain Brenzikofer, and Thomas Baechler. Ultra-miniature omni-directional camera for an autonomous flying micro-robot. In *Photonics Europe*, pages 70000M–70000M. International Society for Optics and Photonics, 2008.
- [25] Patrice Roulet, Pierre Konen, Mathieu Villegas, Simon Thibault, and Pierre Y Garneau. 360 endoscopy using panomorph lens technology. In *BiOS*, pages 75580T–75580T. International Society for Optics and Photonics, 2010.

-
- [26] Edward H Adelson and John Y. A. Wang. Single lens stereo with a plenoptic camera. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (2):99–106, 1992.
- [27] Bennett Wilburn, Neel Joshi, Vaibhav Vaish, Eino-Ville Talvala, Emilio Antunez, Adam Barth, Andrew Adams, Mark Horowitz, and Marc Levoy. High performance imaging using large camera arrays. *ACM Trans. Graph.*, 24(3):765–776, July 2005.
- [28] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 31–42, New York, NY, USA, 1996. ACM.
- [29] et al. Tanida, Jun. Thin observation module by bound optics (tombo): concept and experimental verification. *Applied Optics*, 40(11):1806–1813, 2001.
- [30] Andreas Brückner, Jacques Duparré, Peter Dannberg, Andreas Bräuer, and Andreas Tünnermann. Artificial neural superposition eye. *Opt. Express*, 15(19):11922–11933, Sep 2007.
- [31] Jun Tanida, Keiichiro Kagawa, Keita Fujii, and Ryoichi Horisaki. A computational compound imaging system based on irregular array optics. In *Frontiers in Optics 2009/Laser Science XXV/Fall 2009 OSA Optics & Photonics Technical Digest*, page CWB1. Optical Society of America, 2009.
- [32] Andreas Brückner, Jacques Duparré, Robert Leitel, Peter Dannberg, Andreas Bräuer, and Andreas Tünnermann. Thin wafer-level camera lenses inspired by insect compound eyes. *Opt. Express*, 18(24):24379–24394, Nov 2010.
- [33] Kartik Venkataraman, Dan Lelescu, Jacques Duparré, Andrew McMahon, Gabriel Molina, Priyam Chatterjee, Robert Mullis, and Shree Nayar. Picam: An ultra-thin high performance monolithic camera array. *ACM Trans. Graph.*, 32(6):166:1–166:13, November 2013.
- [34] Jaeyoun Kim Jeong, Ki-Hun and Luke P. Lee. Biologically inspired artificial compound eyes. *Science*, 312(5773):557–561, 2006.
- [35] Lei Li and Allen Y Yi. Development of a 3d artificial compound eye. *Optics express*, 18(17):18125–18137, 2010.
- [36] et al. Song, Young Min. Digital cameras with designs inspired by the arthropod eye. *Nature*, 497(7447):95–99, 2013.
- [37] et al. Floreano, Dario. Miniature curved artificial compound eyes. *Proceedings of the National Academy of Sciences*, 110(23):9267–9272, 2013.
- [38] Geoffrey P Luke, Cameron HG Wright, and Steven F Barrett. A multiaperture bioinspired sensor with hyperacuity. *Sensors Journal, IEEE*, 12(2):308–314, 2012.

Bibliography

- [39] Stéphane Viollet, Stéphanie Godiot, Robert Leitel, Wolfgang Buss, Patrick Breugnon, Mohsine Menouni, Raphaël Juston, Fabien Expert, Fabien Colonnier, Géraud L'Eplattenier, et al. Hardware architecture and cutting-edge assembly process of a tiny curved compound eye. *Sensors*, 14(11):21702–21721, 2014.
- [40] John. Palka. Diffraction and visual acuity of insects. *Science*, 149(3683):551–553, 2006.
- [41] H-Y Shum and Richard Szeliski. Construction of panoramic image mosaics with global and local alignment. In *Panoramic vision*, pages 227–268. Springer, 2001.
- [42] Richard Szeliski and Heung-Yeung Shum. Creating full view panoramic image mosaics and environment maps. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 251–258. ACM Press/Addison-Wesley Publishing Co., 1997.
- [43] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.
- [44] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.
- [45] Matthew Brown and David G Lowe. Recognising panoramas. In *ICCV*, volume 3, page 1218, 2003.
- [46] Luo Juan and Oubong Gwun. Surf applied in panorama image stitching. In *Image Processing Theory Tools and Applications (IPTA), 2010 2nd International Conference on*, pages 495–499, July 2010.
- [47] Yanju Liang, Qing Li, Zhenzhen Lin, and Dapeng Chen. A panoramic image registration algorithm based on surf. In Zhihong Qian, Lei Cao, Weilian Su, Tingkai Wang, and Huamin Yang, editors, *Recent Advances in Computer Science and Information Engineering*, volume 128 of *Lecture Notes in Electrical Engineering*, pages 473–478. Springer Berlin Heidelberg, 2012.
- [48] Feng-Cheng Huang, Shi-Yu Huang, Ji-Wei Ker, and Yung-Chang Chen. High-performance sift hardware accelerator for real-time image feature extraction. *Circuits and Systems for Video Technology, IEEE Transactions on*, 22(3):340–351, 2012.
- [49] Jie Jiang, Xiaoyang Li, and Guangjun Zhang. Sift hardware implementation for real-time image feature extraction. *Circuits and Systems for Video Technology, IEEE Transactions on*, 24(7):1209–1220, 2014.
- [50] Yuan Xu, Qinghai Zhou, Liwei Gong, Mingcheng Zhu, Xiaohong Ding, and Robert KF Teng. High-speed simultaneous image distortion correction transformations for a multicamera cylindrical panorama real-time video system using fpga. *Circuits and Systems for Video Technology, IEEE Transactions on*, 24(6):1061–1069, 2014.

- [51] Xiaoming Peng, Mohammed Bennamoun, Qingbo Wang, Qian Ma, and Zhiyong Xu. A low-cost implementation of a 360° vision distributed aperture system. *Circuits and Systems for Video Technology, IEEE Transactions on*, 25(2):225–238, 2015.
- [52] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 81, pages 674–679, 1981.
- [53] et al. Aldalali, Bader. Flexible miniaturized camera array inspired by natural visual systems. *Journal of Microelectromechanical Systems*, 22(6):1254–1256, 2013.
- [54] Ryusuke Sagawa, Takurou Sakai, Tomio Echigo, Keiko Yagi, Masatsugu Shiba, Kazuhide Higuchi, Tetsuo Arakawa, and Yasushi Yagi. Omnidirectional vision attachment for medical endoscopes. In *The 8th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras-OMNIVIS*, 2008.
- [55] Yingke Gu, Xiang Xie, Guolin Li, Tianjia Sun, Qiang Zhang, Ziqiang Wang, and Zhihua Wang. A new system design of the multi-view micro-ball endoscopy system. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pages 6409–6412. IEEE, 2010.
- [56] Sakib F Elahi and Thomas D Wang. Future and advances in endoscopy. *Journal of biophotonics*, 4(7-8):471–481, 2011.
- [57] Roy Chih Chung Wang, M Jamal Deen, David Armstrong, and Qiyin Fang. Development of a catadioptric endoscope objective with forward and side views. *Journal of biomedical optics*, 16(6):066015–066015, 2011.
- [58] J. Liu, B. Wang, W. Hu, P. sun, J. Li, H. Duan, and J. Si. Global and local panoramic views for gastroscopy: An assisted method of gastroscopic lesion surveillance. *Biomedical Engineering, IEEE Transactions on*, PP(99):1–1, 2015.
- [59] Chun-Hsiang Peng and Ching-Hwa Cheng. A panoramic endoscope design and implementation for minimally invasive surgery. In *Circuits and Systems (ISCAS), 2014 IEEE International Symposium on*, pages 453–456, June 2014.
- [60] Ian M. Gralnek. Emerging technological advancements in colonoscopy: Third eye® retroscope® and third eye® panoramictm, fuse® full spectrum endoscopy® colonoscopy platform, extra-wide-angle-view colonoscope, and naviaidtm g-eyetm balloon colonoscope. *Digestive Endoscopy*, 27(2):223–231, 2015.
- [61] Vincent Kristian Dik. Prevention of colorectal cancer development and mortality: from epidemiology to endoscopy. 2015.
- [62] CapsoVision capsocam. <http://www.capsovision.com/index.php/capsocam#capsocam>. Accessed: 2015-11-01.
- [63] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

Bibliography

- [64] H. Afshari, V. Popovic, T. Tasci, A. Schmid, and Y. Leblebici. A spherical multi-camera system with real-time omnidirectional video acquisition capability. *Consumer Electronics, IEEE Transactions on*, 58(4):1110–1118, November 2012.
- [65] et al. Afshari, Hossein. The panoptic camera: a plenoptic sensor with real-time omnidirectional capability. *Journal of Signal Processing Systems*, 70(3):305–328, 2013.
- [66] et al. Popovic, Vladan. Image blending in a high frame rate fpga-based multi-camera system. *Journal of Signal Processing Systems*, 70(3):1–16, 2013.
- [67] K. Seyid, V. Popovic, O. Cogal, A. Akin, H. Afshari, A. Schmid, and Y. Leblebici. A real-time multiaperture omnidirectional visual sensor based on an interconnected network of smart cameras. *Circuits and Systems for Video Technology, IEEE Transactions on*, 25(2):314–324, Feb 2015.
- [68] H. Afshari, L. Jacques, L. Bagnato, A. Schmid, P. Vandergheynst, and Y. Leblebici. Hardware implementation of an omnidirectional camera with real-time 3d imaging capability. In *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2011*, pages 1–4, May 2011.
- [69] Franz Aurenhammer. Voronoi diagrams – a survey of a fundamental geometric data structure. *ACM Comput. Surv.*, 23:345–405, September 1991.
- [70] H. Afshari, A. Akin, V. Popovic, A. Schmid, and Y. Leblebici. Real-time fpga implementation of linear blending vision reconstruction algorithm using a spherical light field camera. In *Signal Processing Systems (SiPS), 2012 IEEE Workshop on*, pages 49–54, Oct 2012.
- [71] Truman E. Sherk. Development of the compound eyes of dragonflies (odonata). iii. adult compound eyes. *Journal of Experimental Zoology*, 203(1):61–79, 1978.
- [72] Qiang Du, Vance Faber, and Max Gunzburger. Centroidal voronoi tessellations: applications and algorithms. *SIAM review*, 41(4):637–676, 1999.
- [73] Lili Ju, Qiang Du, and Max Gunzburger. Probabilistic methods for centroidal voronoi tessellations and their parallel implementations. *Parallel Computing*, 28(10):1477–1500, 2002.
- [74] O. Cogal, V. Popovic, and Y. Leblebici. Spherical panorama construction using multi sensor registration priors and its real-time hardware. In *Multimedia (ISM), 2013 IEEE International Symposium on*, pages 171–178, Dec 2013.
- [75] Yihong Gong, Guido Proietti, and David LaRose. A robust image mosaicing technique capable of creating integrated panoramas. In *Information Visualization, 1999. Proceedings. 1999 IEEE International Conference on*, pages 24–29. IEEE, 1999.

- [76] Jiajun Zhu, Greg Humphreys, David Koller, Skip Steuart, and Rui Wang. Fast omnidirectional 3d scene acquisition with an array of stereo cameras. In *3-D Digital Imaging and Modeling, 2007. 3DIM'07. Sixth International Conference on*, pages 217–224. IEEE, 2007.
- [77] Joshua Gluckman, Shree K Nayar, and Keith J Thoresz. Real-time omnidirectional and panoramic stereo. In *Proc. of Image Understanding Workshop*, volume 1, pages 299–303. Citeseer, 1998.
- [78] Alan Brunton and Chang Shu. Belief propagation for panorama generation. 2006.
- [79] Marshall F Tappen and William T Freeman. Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 900–906. IEEE, 2003.
- [80] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient belief propagation for early vision. *International journal of computer vision*, 70(1):41–54, 2006.
- [81] Richard Szeliski, Ramin Zabih, Daniel Scharstein, Olga Veksler, Vladimir Kolmogorov, Aseem Agarwala, Marshall Tappen, and Carsten Rother. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(6):1068–1080, 2008.
- [82] Wayne Brown and Barry J Shepherd. *Graphics File Formats; Reference and Guide*. Manning Publications Co., 1994.
- [83] Sean R Stanek, Wallapak Tavanapong, Johnny Wong, Jung Hwan Oh, and Piet C De Groen. Automatic real-time detection of endoscopic procedures using temporal features. *Computer methods and programs in biomedicine*, 108(2):524–535, 2012.
- [84] R. Forster. Manchester encoding: opposing definitions resolved. *Engineering Science and Education Journal*, 9(6):278–280, Dec 2000.
- [85] Olivier Losson, Ludovic Macaire, and Yanqin Yang. Comparison of Color Demosaicing Methods. *Advances in Imaging and Electron Physics*, 162:173–265, 2010.
- [86] C RajaRao, Mahesh Boddu, and Soumitra Kumar Mandal. Single Sensor Color Filter Array Interpolation Algorithms. In *Information Systems Design and Intelligent Applications*, pages 295–307. Springer, 2015.
- [87] Ke Liu, Lei Shen, Jun Yu, Zhiqiang Sun, Chenyu Wang, and Yuan Yao. An Improved Demosaicing Algorithm by Adopting Color Correlation Aided Gradients. In *Signal Processing, Communications and Computing (ICSPCC), 2014 IEEE International Conference on*, pages 859–862, Aug 2014.
- [88] H.S. Malvar, Li-Wei He, and R. Cutler. High-Quality Linear Interpolation for Demosaicing of Bayer-Patterned Color Images. In *Acoustics, Speech, and Signal Processing, 2004*.

Bibliography

- Proceedings. (ICASSP '04). IEEE International Conference on*, volume 3, pages iii–485–8 vol.3, May 2004.
- [89] Xilinx virtex-7 fpga family overview. <http://www.xilinx.com/products/silicon-devices/fpga/virtex-7.html>. Accessed: 2015-05-01.
- [90] G. Ciuti, A. Menciassi, and P. Dario. Capsule endoscopy: From current achievements to open challenges. *Biomedical Engineering, IEEE Reviews in*, 4:59–72, 2011.
- [91] A. Akin, O. Cogal, K. Seyid, H. Afshari, A. Schmid, and Y. Leblebici. Hemispherical multiple camera system for high resolution omni-directional light field imaging. *Emerging and Selected Topics in Circuits and Systems, IEEE Journal on*, 3(2):137–144, June 2013.
- [92] Omer Cogal, Abdulkadir Akin, Kerem Seyid, Vladan Popovic, Alexandre Schmid, Beat Ott, Peter Wellig, and Yusuf Leblebici. A new omni-directional multi-camera system for high resolution surveillance. In *SPIE Sensing Technology+ Applications*, pages 91200N–91200N. International Society for Optics and Photonics, 2014.
- [93] Luigi Bagnato. *Omnidirectional light field analysis and reconstruction*. PhD thesis, École Polytechnique Fédérale de Lausanne (EPFL), 2012.
- [94] Sebastian Nowozin. Autopano-sift, making panoramas fun, 2006. [Online] Available: <http://user.cs.tu-berlin.de/nowozin/autopano-sift/>.
- [95] Luigi Bagnato, Yannick Boursier, Pascal Frossard, and Pierre Vanderghyest. Plenoptic based super-resolution for omnidirectional image sequences. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 2829–2832. IEEE, 2010.
- [96] Pointgrey. ladybug. [Online]. Available: <http://www.ptgrey.com/products/spherical.asp>.
- [97] Dragomir Anguelov, Carole Dulong, Daniel Filip, Christian Frueh, Stéphane Lafon, Richard Lyon, Abhijit Ogale, Luc Vincent, and Josh Weaver. Google street view: Capturing the world at street level. *Computer*, 43(6):32–38, 2010.
- [98] John F. Hamilton Jr. and James E. Adams Jr. Adaptive color plan interpolation in single sensor color electronic camera., May 1997.
- [99] Bahadir K Gunturk, Yucel Altunbasak, and Russell M Mersereau. Color plane interpolation using alternating projections. *Image Processing, IEEE Transactions on*, 11(9):997–1013, 2002.
- [100] Edmund Y. Lam and George S.K. Fung. Automatic White Balancing in Digital Photography. In *Single-Sensor Imaging: Methods and Applications for Digital Cameras*, page pp. 267–294. CRC Press, 2008.
- [101] Clifford E. Cummings. Simulation and Synthesis Techniques for Asynchronous FIFO Design. *SNUG*, Rev. 1.2, 2002.

- [102] Clifford E. Cummings. Clock Domain Crossing (CDC) Design & Verification Techniques Using System Verilog. *SNUG*, Rev. 1.0, 2008.
- [103] Understanding clock domain crossing issues, S. Verma and A.S. Dabare, 2007.
- [104] Serial Protocols Compared, J. Patrick.
- [105] I2C Bus Specification.
- [106] NXP Semiconductors. UM10204. *I2C-bus specification and user manual*, Rev. 6, April 2014.
- [107] NXP Semiconductors. AN10216-01 I2C MANUAL. *APPLICATION NOTE*, March 2003.

Omer Cogal

Curriculum vitae

Ch. de la Chiesaz, 10
1024 – Ecublens (CH)
+41 (76) 2673723
omer.cogal@epfl.ch
people.epfl.ch/omer.cogal

WORK EXPERIENCE

NOVEMBER 2011 – PRESENT
EPFL, Micro Systems Laboratory

Researcher, HW/SW Engineer

Design of FPGA based embedded systems and image processing circuits for Real-time Multi-camera Imaging Systems

FEBRUARY 2011 – NOVEMBER 2011
EPFL, Micro Systems Laboratory

Research Intern, HW/SW Engineer

FPGA Implementation and Verification of a e200 PowerPC Based 32-bit SoC for High Temperature Operation

AUGUST 2005 – FEBRUARY 2011
The Scientific and Technological Research Council of Turkey, National Research Institute of Electronics and Cryptology

Researcher, HW/SW Engineer

Prototype Design, PCB design, Digital Circuit Design, Micro-controller Firmware, Linux based embedded system design, Smart card reader

EDUCATION

2011 – PRESENT **Electronics Engineering**
PHD
EPFL, Lausanne

2006 – 2009 **Electronics Engineering**
MSc
Bogazici University, Istanbul

2000 – 2005 **Electronics Engineering**
BSc
Istanbul Technical University

PROJECTS

2011-2015 **Multi-Camera Real-time Imaging Circuit and Systems Design** (<http://www.rts.ch/emissions/court-du-jour/6390143-les-cameras-panoptiques.html>)
System Design, Opto-Mechanical Design, Image Processing Algorithm SW, FPGA Circuit Design, Microblaze Embedded SW Design, PCB Design, SolidWorks, MATLAB, VHDL, C, Xilinx Virtex-5/Virtex-7, ISE/EDK, Altium

2011 **FireBird:FPGA Design and Verification of a e200 PowerPC Based 32-bit SoC for High Temperature Operation**
FPGA Implementation, SoC System Architecture Design, ASIC migration, Linker Script, C, Assembly, VHDL, Xilinx Virtex-5, ISE/EDK

2007-2011 **Card Access Reader** (bilgem.tubitak.gov.tr/en/icerik/kec-card-access-device)
ARM920T, Embedded Linux cross-compiling, Driver Hacking, ARM-7 STM32 Based MCU, Teridian 8051/2 Based MCU, Keil 8051 Compiler, USB Slave Smart Card Reader Device Design, PCB Design, Altium

2005-2007 **Embedded system design for Electrooptical Devices** (<http://bilgem.tubitak.gov.tr/en/urunler/optics-and-laser-systems>)
16-bit PIC based embedded system design, C, discrete component based analog circuit design, PCB design, Protel99, Altium

SOFTWARE SKILLS

LANGUAGES VHDL/Verilog, C/C++, MATLAB

TOOLS Xilinx ISE/EDK, Altium, Solid Works, gcc, Linux

FPGAs Xilinx Virtex5, Virtex7, Spartan3

MCUs/MPs Microblaze, PowerPC e200
ARM7, ARM9, PIC, 8051/2,
8085/8086 173

PUBLICATIONS

PATENT

WO2015128801 A2 , **Large Field of View Multi-Camera Endoscopic Apparatus with Omni-Directional Illumination**, Publication Date: 03.09.2015, EPFL, Omer Cogal, Yusuf Leblebici

JOURNAL/CONFERENCE

K. Seyid, V. Popovic, **O. Cogal**, A. Akin and H. Afshari et al. **A Real-time Multi-aperture Omni-directional Visual Sensor Based on Interconnected Network of Smart Cameras**, in IEEE Transactions on Circuits and Systems for Video Technology, vol. 25, num. 2, p. 314-324, 2015.

V. Popovic, K. Seyid, E. A. Pignat, **O. Cogal** and Y. Leblebici. **Multi-Camera Platform for Panoramic Real-Time HDR Video Construction and Rendering**, in Journal of Real-Time Image Processing, 2014.

O. Cogal, A. Akin, K. Seyid, V. Popovic and A. Schmid et al. **A New Omni-Directional Multi-Camera System for High Resolution Surveillance**. SPIE Defense and Security Symposium, Baltimore, Maryland, United States, Proceedings of SPIE, 2014.

V. Popovic, K. Seyid, A. Akin, **O. Cogal** and H. Afshari et al. **Image Blending in a High Frame Rate FPGA-based Multi-Camera System**, in Journal of Signal Processing Systems for Signal, Image and Video Technology-, vol. 76, num. 2, p. 169-184, 2013.

O. Cogal, V. Popovic and Y. Leblebici. **Spherical Panorama Construction using Multi Sensor Registration Priors and Its Real-time Hardware**. IEEE International Symposium on Multimedia (ISM2013), Anaheim, California, USA, 2013.

A. Akin, **O. Cogal**, K. Seyid, H. Afshari and A.

Schmid et al. **Hemispherical Multiple Camera System for High Resolution Omni-Directional Light Field Imaging**, in IEEE Journal of Emerging and Selected Topics in Circuits and Systems, vol. 3, num. 2, p. 137 - 144, 2013.

R. Cojbasic, **O. Cogal**, P. A. Meinerzhagen, C. C. S. D. Senning and C. J. Slater et al. **FireBird: PowerPC e200 Based SoC for High Temperature Operation**. IEEE Custom Integrated Circuits Conference (CICC), San Jose, California, USA, 2013.

ACTIVITIES

ENTREPRENEURSHIP	Several courses taken at EPFL, Leading Business Plan Development Team EndoPano Project, EndoPano Start-up Pitching in front of expert juries, National Competitions(links available upon request)
SOCIAL	Board Member of TURQUIA 1912 The Association for Turkish Students in Switzerland
SPORTS	Swimming
ART	Folk Dance (Aegean Region)

COMMUNICATION SKILLS

TURKISH	Native speaker
ENGLISH	Oral: good – Written: good
FRENCH	Oral: basic

REFERENCES

Available upon request