

DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED
SCIENCES

FPGA BASED
STEREO CAMERA DEPTH MAP GENERATION

by
Celal GÜVENDİK

August, 2014
İZMİR

FPGA BASED STEREO CAMERA DEPTH MAP GENERATION

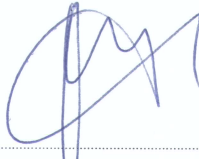
**A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Master of Science of
in Electrical and Electronics Engineering**

**by
Celal GÜVENDİK**

**August, 2014
İZMİR**

M.SC. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled “**FPGA BASED STEREO CAMERA DEPTH MAP GENERATION**” completed by **CELAL GÜVENDİK** under supervision of **ASSIST. PROF. DR. ÖZGÜR TAMER** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.



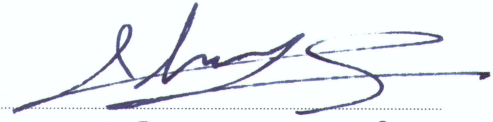
Assist. Prof. Dr. Özgür TAMER

Supervisor



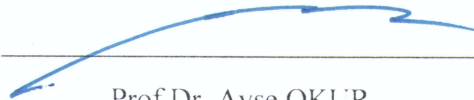
Assist. Prof. Dr. Nalan ÖZKURT

(Jury Member)



Assist. Prof. Dr. M. Alper Selver

(Jury Member)



Prof. Dr. Ayşe OKUR

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGMENTS

Foremost, I would like to express my special thanks of gratitude to my supervisor Asst. Prof. Dr. Özgür TAMER who always supported and encouraged me in all steps of the thesis. His guidance helped me in all the time of research and writing of this thesis. Besides my supervisor, I like to thank Asst. Prof. Dr. Metehan MAKİNACI for Computer Vision lessons during my Master of Science program. His lessons provided to me very useful knowledge for the thesis.

I also would like thank to my family, my mother Hülya, my father A. Cemal and my sister K. Cansu and my girlfriend Efruz for their never ending support and motivation.

This thesis is also supported under the project no. 3110560 and name “3D Video Conference TV” that is funded by TUBITAK and carried out by VESTEL Elektronik San. ve Tic. A.Ş. and I would like to thank to my colleges A. Esat Genç and Aziz Gökgöz and our manager Metin Nil for their cooperation in this project.

Celal GÜVENDİK

FPGA BASED STEREO CAMERA DEPTH MAP GENERATION

ABSTRACT

The aim of this thesis is depth based segmentation of the video frames by using a computational stereovision system. During the study, first the stereo matching method is modeled on Matlab software by the aid of two ordinary webcam employed as stereo cameras and the results are evaluated to form the depth map. The model is then modified to be appropriate for parallel processing on FPGA with real stereo camera. The final system runs on Xilinx Virtex®-5 LX50T FPGA chip, uses Digilent VmodCAM - Stereo Camera Module to capture stereo images and for the real time monitoring of the outputs. HDMI interface is used to send video to the display.

The output of the system is the video frames that are the segmented according to the test objects distance from stereo camera to exact location of these objects. The algorithm uses disparity information of the captured images and the stereo camera parameters to obtain the distance of the objects to the stereo camera system. So, if the object is on the correct distance for the current disparity value, the vertical edges of the object disappear on the display.

These disparity values were used to generate the depth map on the Matlab model automatically and on the FPGA part of this study, they are chosen manually to show, how the algorithm works and segment the depth of field.

Keywords: Computer stereovision, stereo camera, disparity, depth of field, depth map, image segmentation, FPGA, parallel processing, Verilog HDL, Matlab.

FPGA TABANLI ÇİFTLİ KAMERA DERİNLİK HARİTASI OLUŞTURULMASI

ÖZ

Bu tezin amacı, bilgisayarlı çiftli görüntüleme (computational stereovision) sistemi kullanımı ile video imgelerinde derinlik tabanlı ayrıştırma yapmaktır. Çalışma sırasında, ilk önce çiftli (stereo) eşleme metodu Matlab yazılımında modellenmiş, iki adet sıradan web kamerası yardımıyla çiftli (stereo) kamera oluşturulmuş ve sonuçlar derinlik haritası olarak değerlendirilmiştir. Ardından bu model FPGA ortamında gerçek çiftli (stereo) kameralar ile paralel çalışmaya uygun olarak düzenlenmiştir. Nihai sistem Xilinx firmasının Virtex®-5 LX50T FPGA çipinde koşturulmuş, stereo imgeleri yakalamak için Digilent firmasının VmodCAM - Stereo Camera Module kartı kullanılmış ve video görüntüsü HDMI arayüzü ile ekranda gösterilmiştir.

Sistemin çıkışı test için kullanılan nesnelerin görüntülerinin yine bu nesnelerin kameraya olan gerçek uzaklıklarına göre ayrıştırılmış olduğu video çerçeveleridir. Algoritma yakalanan imgelerin eşitsizlik (disparity) bilgisini ve stereo kamera parametrelerini kullanarak nesnenin çiftli kamera sistemine uzaklığını belirler. Eğer nesne o an ki eşitsizlik (disparity) değeri için doğru uzaklıkta bulunuyorsa, nesnenin dikey kenarları ekrandaki sonuçta kaybolacaktır.

Bu eşitsizlik (disparity) değerleri Matlab modelinde otomatik olarak derinlik haritası oluşturmak için kullanılmıştır ve çalışmanın FPGA kısmında, eşitsizlik değerleri manuel olarak değiştirilerek algoritmanın nasıl çalıştığı ve derinliğin nasıl ayrıştırıldığı gösterilmiştir.

Anahtar kelimeler: Bilgisayarlı stereo (çiftli) görüntüleme, stereo kamera, eşitsizlik (disparity), alan derinliği, derinlik haritası, imge ayrıştırılması, FPGA, paralel işlem, Verilog HDL, Matlab.

CONTENTS

	Pages
M.Sc. THESIS EXAMINATION RESULT FORM.....	ii
ACKNOWLEDGMENTS	iii
ABSTRACT.....	iv
ÖZ	v
LIST OF FIGURES	viii
LIST OF TABLES	xi
 CHAPTER ONE - INTRODUCTION	 1
1.1 Aim of the Thesis	3
 CHAPTER TWO - STEREOVISION.....	 5
2.1 Human Vision System.....	5
2.2 Computer Stereovision	7
2.3 Feature Extraction, Rectification and Stereo Matching	11
2.4 Image Segmentation with Stereo Cameras.....	11
 CHAPTER THREE - METHODOLOGY AND SIMULATION RESULTS.....	 13
3.1 Stereo Matching	13
3.1.1 Feature Detection and Matching.....	15
3.1.2 Proposed Algorithm and Image Rectification	19
3.2 Disparity and Depth Map Estimation	23
3.2.1 Sum of Absolute Difference	24
3.3 Simulation and Results	24
3.3.1 Results.....	26
 CHAPTER FOUR - FPGA IMPLEMENTATION OF DISPARTY CONCEPT	 34
4.1 Hardware Design	34

4.2 Camera Driver Block.....	36
4.2.1 Block RAM in FPGA	37
4.3 Disparity Map Generator Block	40
4.4 Clock Generator Block	43
4.5 HDMI Driver Block	44
4.6 Results of the Hardware Design	47
4.6.1 Rectangular Object, at 1 Meter	48
4.6.2 Rectangular Object, at 2 Meters	51
4.6.3 Rectangular Object, at 0.75 Meter	53
4.6.4 Circular and Triangular Objects, at 1 Meter	55
4.6.5 Changing Distance when Disparity is Constant	58
4.6.6 Stereo Field of View at 0.25 Meter	59
4.6.7 The Observed and Calculated Disparity Values	62
CHAPTER FIVE - CONCLUSION	64
5.1 Future Works	65
REFERENCES	66
APPENDICES	69

LIST OF FIGURES

	Pages
Figure 1.1 A Stereoscope in 1900s	1
Figure 1.2 Kinect for XBOX 360.....	2
Figure 1.3 The rescue robot Urbie has stereovision.....	3
Figure 2.1 Human vision system.....	5
Figure 2.2 The scene difference between right and left eyes	6
Figure 2.3 Experimental stereo cameras	7
Figure 2.4 some of intrinsic parameters of a camera	8
Figure 2.5 Focal length “f” of the a camera	9
Figure 2.6 The relation between disparity ($x-x'$), baseline (B), focal length (f) and the distance (Z) between camera and the object “X”	10
Figure 2.7 Epipolar line and epipolar plane	10
Figure 3.1 General structure of the algorithm.....	14
Figure 3.2 ‘(a)’ is the detailed presentation of a2 block (Harris Operator block) and ‘(b)’ the is the Harris Operator that finds out the corner features	16
Figure 3.3 Detailed presentation of a3 block	19
Figure 3.4 Detailed presentations of a4 and a5 blocks.....	20
Figure 3.5 $\tan(A1)$, slope of between any two features	21
Figure 3.6 a and b are the list of true common features of the images, c and d are the matrices that contain tangents of line segments that are generated from the feature points.....	22
Figure 3.7 USB2.0 stereo camera systems.....	24
Figure 3.8 Block diagram of the depth map generator system	25
Figure 3.9 The image is captured by left camera	26
Figure 3.10 The image is captured by right camera.....	27
Figure 3.11 Corner features of the left camera image.....	27
Figure 3.12 Corner features of the right camera image	28
Figure 3.13 True common feature points of the processing image.....	28
Figure 3.14 The α angle of between x axis and the line segment of two feature in left image	30

Figure 3.15 The α angle of between x axis and the line segment of two feature in right image.....	30
Figure 3.16 Disparity map of the tested images.....	31
Figure 3.17 Distance map of the test image.....	32
Figure 3.18 Distance map of the test image without rectification	32
Figure 4.1 General Structure of hardware design in FPGA	35
Figure 4.2 Signals on the image sensors as block diagram.....	36
Figure 4.3 IP symbol of the Simple Dual Port RAM that is used in this design from Core Generator of ISE.....	38
Figure 4.4 Downscale operation in this design	39
Figure 4.5 SW7 is high and the disparity value is set as $(0010111)_2=(23)_{10}$	41
Figure 4.6 SW7 is low and the rectification value is set as $(0000101)_2=(5)_{10}$	41
Figure 4.7 Verilog HDL codes for rectification and disparity values usage in equation 4.1	42
Figure 4.8 Block demonstration of DCM_ADV primitive in Virtex-5	43
Figure 4.9 The required signal to drive Chronitel 7301C	44
Figure 4.10 Timing diagrams of HDMI sync block from test bench.....	46
Figure 4.11 The test bed and the test objects	47
Figure 4.12 Stereo camera and FPGA development board in (a), the displays of the images from left camera, right camera in (c) and disparity demonstration in (b)	48
Figure 4.13 Disparity map with zero pixel disparity and zero pixel rectification.....	49
Figure 4.14 Disparity map after, right and the left images are aligned horizontally by entering rectification register $(0000011)_2$	49
Figure 4.15 The appropriate disparity value of $(0010110)_2=(22)_{10}$ for one meter distance from cameras	50
Figure 4.16 Disparity map for $(0010101)_2=(21)_{10}$ disparity level at one meter.....	50
Figure 4.17 Disparity map for $(0010111)_2=(23)_{10}$ disparity level at one meter.....	51
Figure 4.18 The appropriate disparity value of $(0000111)_2=(7)_{10}$ for two meters distance from cameras	52
Figure 4.19 Disparity map for $(0000101)_2=(6)_{10}$ disparity level at two meters.....	52
Figure 4.20 Disparity map for $(0001000)_2=(8)_{10}$ disparity level at two meters.....	53

Figure 4.21 The appropriate disparity value of $(0011111)_2=(31)_{10}$ for 0.75 meter distance from cameras	54
Figure 4.22 Another appropriate disparity value of $(0100001)_2=(33)_{10}$ for 0.75 m distance from cameras	54
Figure 4.23 Disparity map for $(0011100)_2=(28)_{10}$ disparity level at 0.75 meter.....	55
Figure 4.24 Disparity map for triangular for $(0010111)_2=(23)_{10}$ disparity level at one meter.....	56
Figure 4.25 The appropriate disparity value $(0010110)_2=(22)_{10}$ for triangular, one meter distance from cameras	56
Figure 4.26 Disparity map for circular for $(0010111)_2=(23)_{10}$ disparity level at one meter.....	57
Figure 4.27 The appropriate disparity value $(0010110)_2=(22)_{10}$ for circular, one meter distance from cameras	57
Figure 4.28 Disparity value is set $(0010110)_2=(22)_{10}$ at the distance 1.1 meter	58
Figure 4.29 Disparity value is set $(0010110)_2=(22)_{10}$ at the distance 0.9 meter	58
Figure 4.30 Stereo field of view calculation of the test setup.....	59
Figure 4.31 The big rectangular object is not fit in near field of cameras and disparity map is not generated correctly.	60
Figure 4.32 Small rectangular is used to find appropriate disparity in 25 cm from cameras and it is observed as $(1101111)_2=(111)_{10}$ pixel.....	60
Figure 4.33 At 25 cm distance, left and right images of the small rectangular is separated each other completely in disparity map.	61
Figure 4.34 Circular almost fit the common stereo vision area at 25 cm distance because it has 15.5 cm diameter.....	61
Figure 4.35 Graphical demonstrations of the observed and calculated disparity values	62

LIST OF TABLES

	Pages
Table 4.1 Clocks and frequencies of the design.....	44
Table 4.2 Values of the graph in Figure 4.35.....	63

CHAPTER ONE

INTRODUCTION

Recently, perception of depth to digital video content is one of the popular things in the film industry. However, this is not the only use of these cameras. They are also used as smart surveillance systems, droids, drones, smart robots (Netting, 2011) and the video games (Kinect, 2014). Depth segmentation of a scene provides very useful information about the objects in the image. Knowing the distance of any point in the image helps us to define the objects in the scene. There are several methods to get the depth information about the scene such as; stereoscopic visualization (Camellini, et al., 2014), one moving camera method (Holzmann & Hochgatterer, 2012) (to capture multiple images of a scene from multiple projections), one camera with IR sensor and IR emitter (Kinect, 2014).



Figure 1.1 A Stereoscope in 1900s (Davepape, 2006)

Stereoscopy is not a new subject; it is old as the humanity. Because the human eyes have the stereoscopic vision and the first studies on stereoscopy was made by Euclid in AD 280 (The Turing Institute, 1996). Also in 1558, Leonardo Da Vinci used the perception of depth in its studies (The Turing Institute, 1996). In 1611,

Kepler proposed the projection theory of the human stereo vision in his book “*Dioptrice*”. Early studies on the stereoscopy were made with the “*stereoscopes*” in 1900s which is presented in Figure 1.1. These devices were working as watching device of stereo photos (The Turing Institute, 1996).

With the contribution of the recent digital and computational developments, the phenomenon of stereoscopy evolves to stereovision technologies. For example, films are shot with stereo cameras, can be watched on 3D digital TVs in the houses or in the theatres and the video games become playable without any control device like consoles.

The background of the stereovision or multi-vision systems is based on finding the similarities and the dissimilarities of the scenes that are captured by the digital cameras positioned to special coordinates and taking information from the scene. After that, this information is used for many purposes like segmentation of the scene depth or depth map generation of the scene (InTech, 2012). Depth map of an image is the demonstration of this image as distance between cameras and the objects in the scene.

Generating the depth map of any scene may help a system that needs distance information of the objects to do its duty. For example, the Kinect (Figure 1.2) is a device, which is presented by Microsoft for XBOX 360 gaming console and provides to the user playing games without any game console. Kinect takes advantage of the depth information that is generated by itself via its RGB camera, IR sensor and IR emitter and percept the movements of the gamer during the game by using the depth of the scene. These depth data is processed in XBOX console and the reaction is transferred to the appropriate form for displaying in the monitor (Kinect, 2014).



Figure 1.2 Kinect for XBOX 360 (Evan-Amos, 2011)

Another example of the systems that uses the stereovision is the rescue robots. The rescue robot named “Urbie” has the stereo camera for stereovision (in Figure 1.3) (Netting, 2011). The Urbie is designed to help police or military forces in their rescue operations. Urbie uses stereovision to identify its path or its target by using perception of depth.

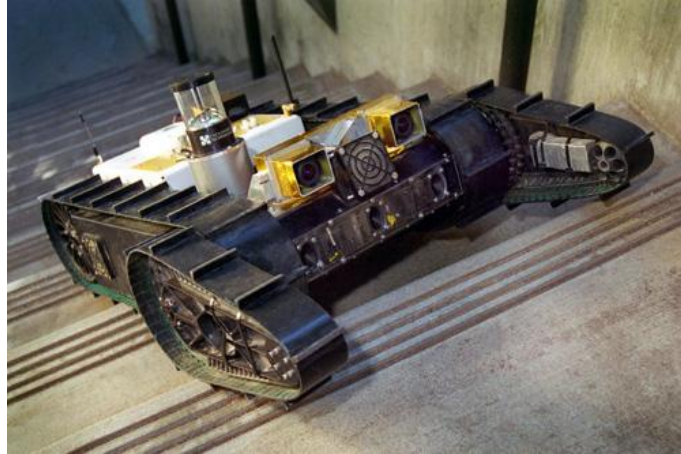


Figure 1.3 The rescue robot Urbie has stereovision (Netting, 2011)

As evident from these examples, to give perception of depth on any artificial system is an important issue to make more functional systems.

1.1 Aim of the Thesis

In this thesis, stereovision is used to segment distance of the object in a scene and contribute the extraction of depth information on image. While generation the depth information is studied, the geometrical and computational parameters of a stereovision system is examined in both software and hardware. Also by using FPGA, the hardware is designed to work parallel.

In the process of depth based segmentation (Mutto, Zanuttigh, & Cortelazzo, 2010; Cigla & Alatan, 2008), dissimilarities and similarities can be separated by using the disparity information. Disparity is the difference of the stereo images and the depth map is generated from possible disparity maps (Mattoccia, 2013). The most appropriate disparity values are chosen from possible disparities and used in the

depth map. By employing the camera system parameters with these depth information, all the distances of the objects in scene can be obtained.

In the processes of obtaining depth information of the images in Matlab, Harris operator (Harris & Stephens, 1988) is used to feature extraction from the stereo images. For stereo matching, Sum of Squared Differences (SSD) method (Okutomi, Canon Inc., & Kanade, 1991) is applied. Also an algorithm is proposed to rectify the stereo images. The algorithm aims to get the images on same epipolar lines by using the tangent information of the matching features.

In the hardware design, the distance of an object is obtained by using stereovision. To realize this, Genesys Virtex-5 FPGA development board is used with the employing appropriate stereo camera board VmodCAM™ from Digilent. It is aimed to make a design which is capable of parallel processing. The Verilog HDL is chosen to make this design in ISE from Xilinx.

In the following chapter, the stereovision will be explained. In chapter three, the simulation part of the study, depth map generation and its results will be given. In chapter four, the depth segmentation will be studied with its result as FPGA implementation of a stereovision system. Conclusion of the thesis will be given in chapter five.

CHAPTER TWO

STEREOVISION

Stereovision is the visualization of any scene by using two cameras. It is a term that is used in stereoscopy. Stereoscopy is a very old concept that is based on 1800s and its purpose is the imitation of human vision to make an illusion to generate perception of depth on images (The Turing Institute, 1996). Previously, analog photography techniques were used to make stereoscopic images but at the present time, digital cameras and the computers are used to generate stereoscopic images.

First, human vision system will be explained shortly. After that, stereovision techniques will be mentioned.

2.1 Human Vision System

We experience objects consisting of three dimensions with our eyes; width, length and height. Two dimensional images occur because of the reflections from the objects, and human vision system provides depth perception by using these two 2D images. In other words, perception of three dimensions is the perception of depth. This system is called stereopsis and helps us sensing the world in 3D.

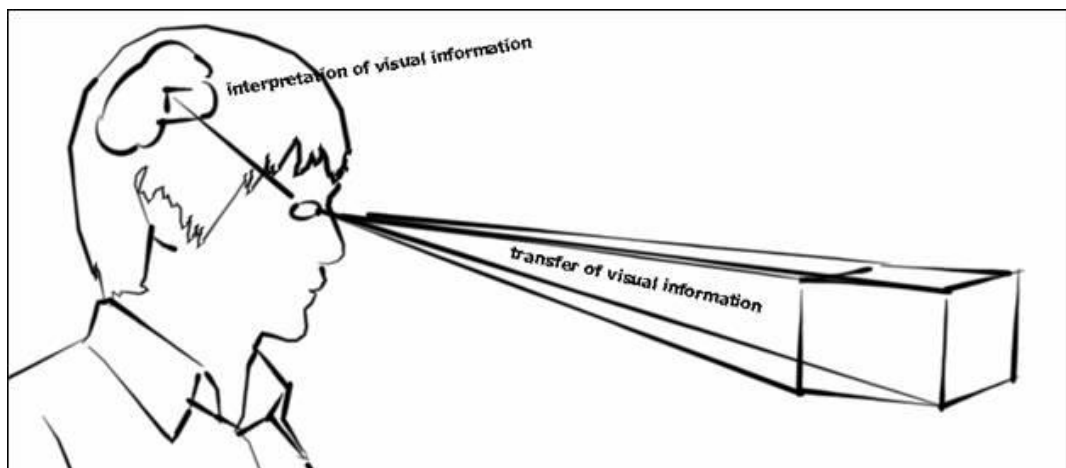


Figure 2.1 Human vision system (eLADwiki, 2010)

There are four main information help occurring perception of depth;

Eye Lenses Information (focus and convergence): Visual information is transferred to the retina in an appropriate manner with help of our eye lenses. These lenses focus the image on the retina. Also, the alignment of the eyes is important for perception of depth information at this point. Convergence is the inward motion of eyes according to the distance of eyes line of sight (when viewed as an object much nearby location, your eyes is in squint status). Through focus and converge, the two compatible images are transmitted to the brain via our eyes.

Stereoscopic Information: There is approximately 6.3 cm distance (interocular distance) between human eyes (Dodgson, Woods, Merritt, Benton, & Bolas, 2004). This means the right eye and the left eye see different images from the scene. In other words, the images occurring on the right and the left retinas are not the same. This is the result of the horizontal separation of the eyes. This difference is called interocular (binocular) disparity. After interpretation of the images on the brain, 3D perception is evaluated.

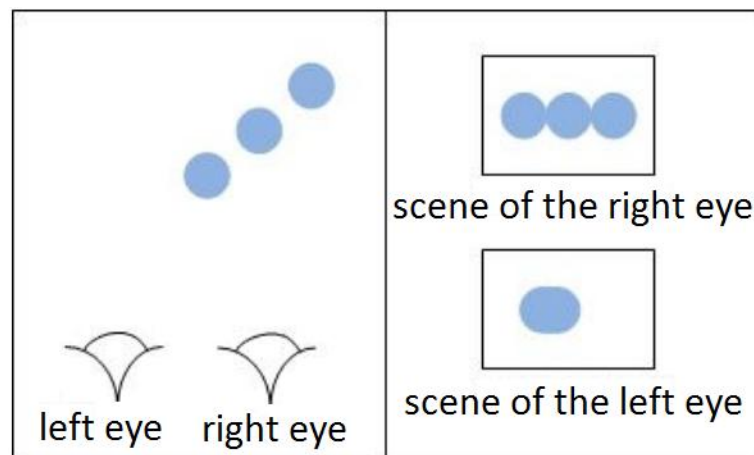


Figure 2.2 The scene difference between right and left eyes

Dynamic Information: The dynamic information is about Parallax effect. Parallax effect is the difference of the scenes that are observed from two different lines of sight (Benbow, 1989). Parallax causes the formation of these two different images. The dynamic information is used to interpret the images in our brain again.

Pictorial Information: Previous information about our perception of depth is geometric parameters that contribute to the formation the depth perception. Apart

from these three parameters, pictorial information (color tone, perspective, shadows etc. ...) is also effective in creating a sense of depth. Even in two-dimensional images or videos, the only information that will give depth perception is the pictorial information. Depth is not detected by stereoscopy; it is only constructed by using pictorial information of our previous visual experience in our brains' depth perception.

2.2 Computer Stereovision

To imitate the human eyes, stereo cameras are used in computer stereo vision systems. The goal is the same with human vision; generating the 3D structure of a scene using two images, each are acquired from a different viewpoint in space. Also the images can be obtained by one moving camera or multiple cameras. In this thesis, the subject is studied with employing stereo camera (two cameras) system.

Stereo Camera: In a stereovision system, one of the most important elements is the stereo camera. While images are processed and depth is calculated, the intrinsic and the extrinsic parameters of the camera system are needed.



Figure 2.3 Experimental stereo cameras

Intrinsic parameters characterize the transformation of image plane coordinates to pixel coordinates in each camera. Resolution, FOV (field of view), pixel dimension of the image sensor, focal length of the camera lenses are examples of the intrinsic parameters (Figure 2.4).

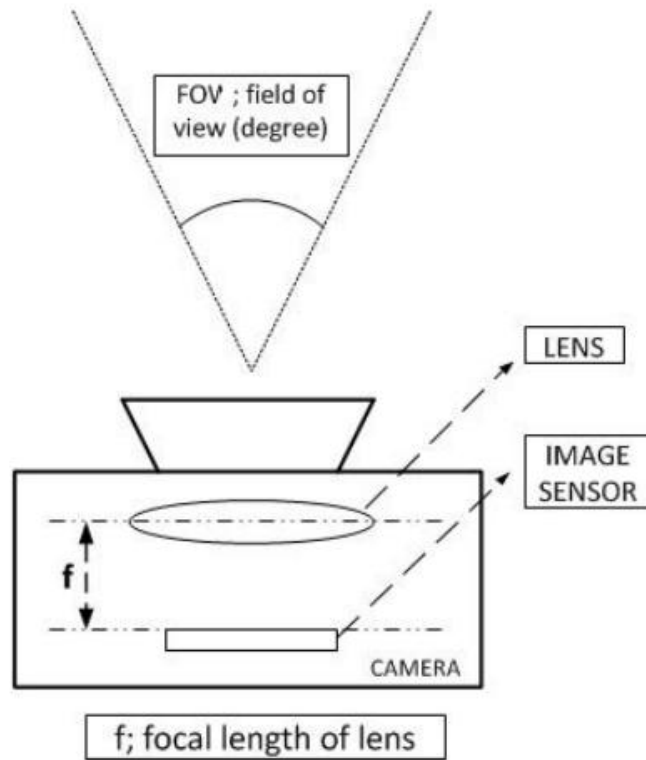


Figure 2.4 Some of intrinsic parameters of a camera

Extrinsic parameters describe the relative position and orientation of the two cameras. For example, the distance between the center of the lenses and the relative angles of the cameras are some of the extrinsic parameters. Variation of any of these parameters, affect the system outputs

Baseline: The distance between the centers of the projection axes is called the baseline (Mattoccia, 2013). Like human interocular distance of the eyes, there must be an appropriate distance between cameras in the stereovision system. This distance is generally 6.3 cm (Dodgson, Woods, Merritt, Benton, & Bolas, 2004) in order to imitate human vision.

Focal Length (f): The focal length of the camera lens is the distance between the lens and the image sensor of the camera when the viewed object is focused. It is usually stated in millimeters. It is one of the most important parameters while calculating the disparity.

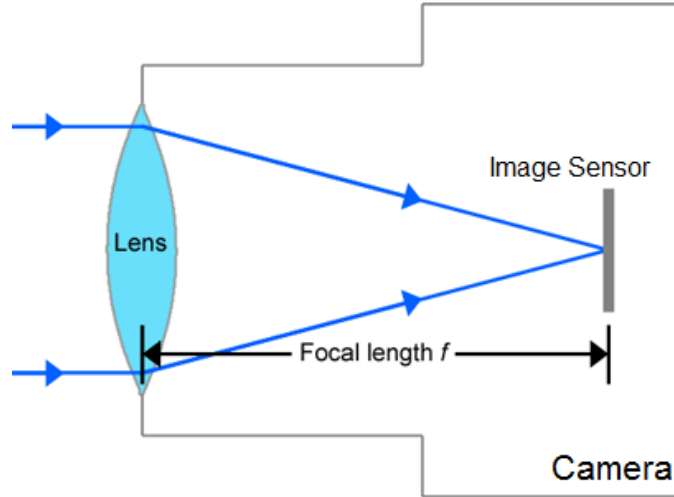


Figure 2.5 Focal length “f” of the a camera

Disparity: Disparity can be determined as the dissimilarity of the right and the left images that are acquired by a stereo camera system (Mattoccia, 2013). Each camera of the stereo system has different two dimensional frames and this difference generates the third dimension. In Figure 2.6, x and x' are the distances between the points in image plane that correspond to the scene point 3D and their camera center. B is the baseline of the stereo camera and f is the focal length of camera.

$$disparity = d = x - x' = \frac{Bf}{Z} \quad (2.1)$$

Here, d is disparity, B is baseline, f is focal length and the Z is the distance from camera to object. Thus, the depth of a point in a scene is inversely proportional to the difference in distance of corresponding image points and their camera centers. So with this information, the depth of all pixels in an image can be found.

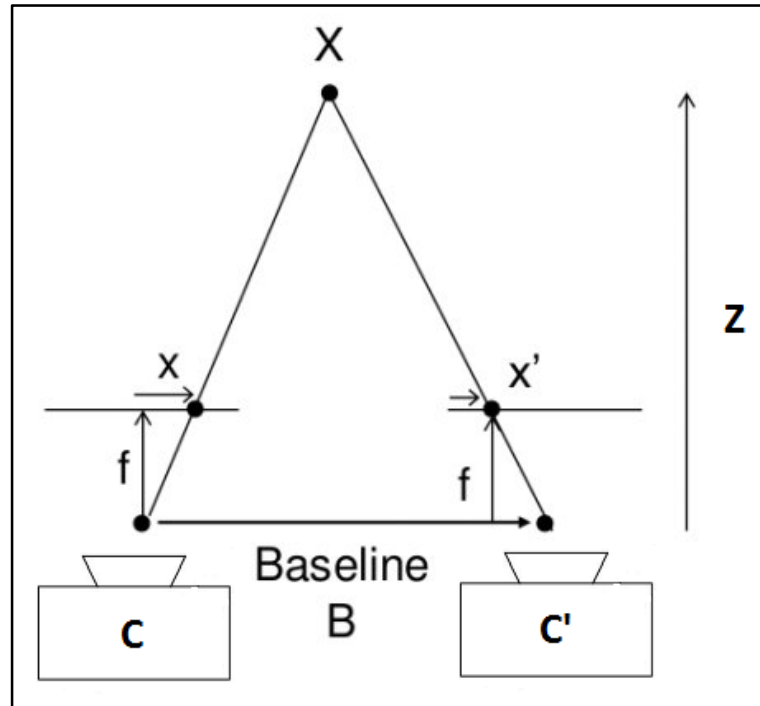


Figure 2.6 The relation between disparity ($x-x'$), baseline (B), focal length (f) and the distance (Z) between camera and the object “ X ”

Epipole: Epipole is the image that is captured by a camera, is on the projection center of the other camera (Bebis, 2004).

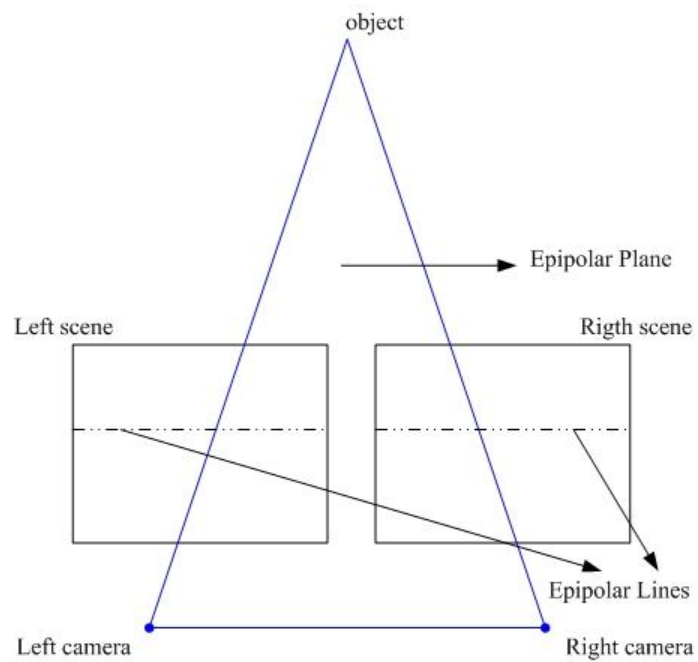


Figure 2.7 Epipolar line and epipolar plane

Epipolar line and epipolar plane: In Figure 2.7 epipolar line and the epipolar plane is demonstrated, the virtual triangle that has the corner of object, left camera center and the right camera center is called epipolar plane and epipolar line is the intersection of the image plane and the epipolar plane that is formed between projected object and the projection centers of cameras (Bebis, 2004).

Depth map: Depth map of a scene can be defined as the image which is segmented according to distance information of the objects that are on this scene. In stereovision, disparity information is used to generate depth map.

Those concepts are mentioned above are used in stereo matching problems that will be mentioned in Section 3.1. Stereo matching problem is the correspondence problems that aim to solve the finding common points in the stereo images. It is necessary to rectify the stereo images and to render in same epipolar lines.

2.3 Feature Extraction, Rectification and Stereo Matching

Stereo matching is the process of finding correspondence of the stereo images to serve other purpose like depth map generation. To solve stereo matching problems, the images are subjected to some preprocesses like feature extraction and rectification.

The all possible features of the images are extracted to find which features are common at first. These features can be textures, corners, edges etc. After that these feature are matched and commons of the features are kept to use in rectification process. In the rectification process it is aimed to locate the images in same epipolar lines.

2.4 Image Segmentation with Stereo Cameras

Segmentation of an image is separating the individual pixel groups from each other according to their corresponding specialties like colors, textures, depth. The separation of images on the foreground from background, obtaining a special object or a part of the object in an image and segmentation of human hands from whole

human body can be given as examples (Rahmat, Al-Tairi, Saripan, & Sulaiman, 2012).

There are many image segmentation methods. In this thesis the stereo image segmentation is studied and the scene is segmented according to the depth information by using stereo cameras (Cigla & Alatan, 2008; Mutto, Zanuttigh, & Cortelazzo, 2010). The stereo camera systems capture the images from different line of sight angle. Thus the objects which are in these scenes have different projection from each other. The disparity gives information about the depth of the object in the images by using these differences of the scenes.

CHAPTER THREE

METHODOLOGY AND SIMULATION RESULTS

Similar to the human eyes, images are obtained via a pair of cameras on the same horizontal axis. Although, the camera pair is placed properly on the horizontal axis, due to external and internal factors, left and right images might shift. In this case, the vertical shifting amount between the stereo images must be calculated to create proper depth information. Then, one of the stereo components must be shifted as much as the calculated shifting value. Thus, the same spots of the object in images will be aligned horizontally.

This alignment is important to accept these images as stereo images because, the most of the time, the success of stereo image processes relies on this matching process. In this study, depth map of the scene is estimated after the alignment process.

3.1 Stereo Matching

In the literature, stereo matching problems are solved by area or feature based methods (Mattoccia, 2013). Area based image matching is based on looking the neighbors of each pixel in the image to compare with other pixels in the image. Feature based image matching is based on comparing two image by the extracted features from the mapping model. Such as; Edge based, Coarse-to-fine, Adaptive windows, Dynamic programming, Markov Random Fields, Graph cuts, Multi-baseline are area or feature based methods which include mathematical modeling. According to these models, problem is resolved based on the extracted features or image pixel neighborhood. These solutions have the disadvantages of slow processing time because of the complexity of the operations.

We propose a new algorithm to match stereo images in a more simple and practical way to find true common features. Also, the algorithm finds vertical shifting amount for epipolar line rectification in parallel camera situation. It is practical because it requires less computation.

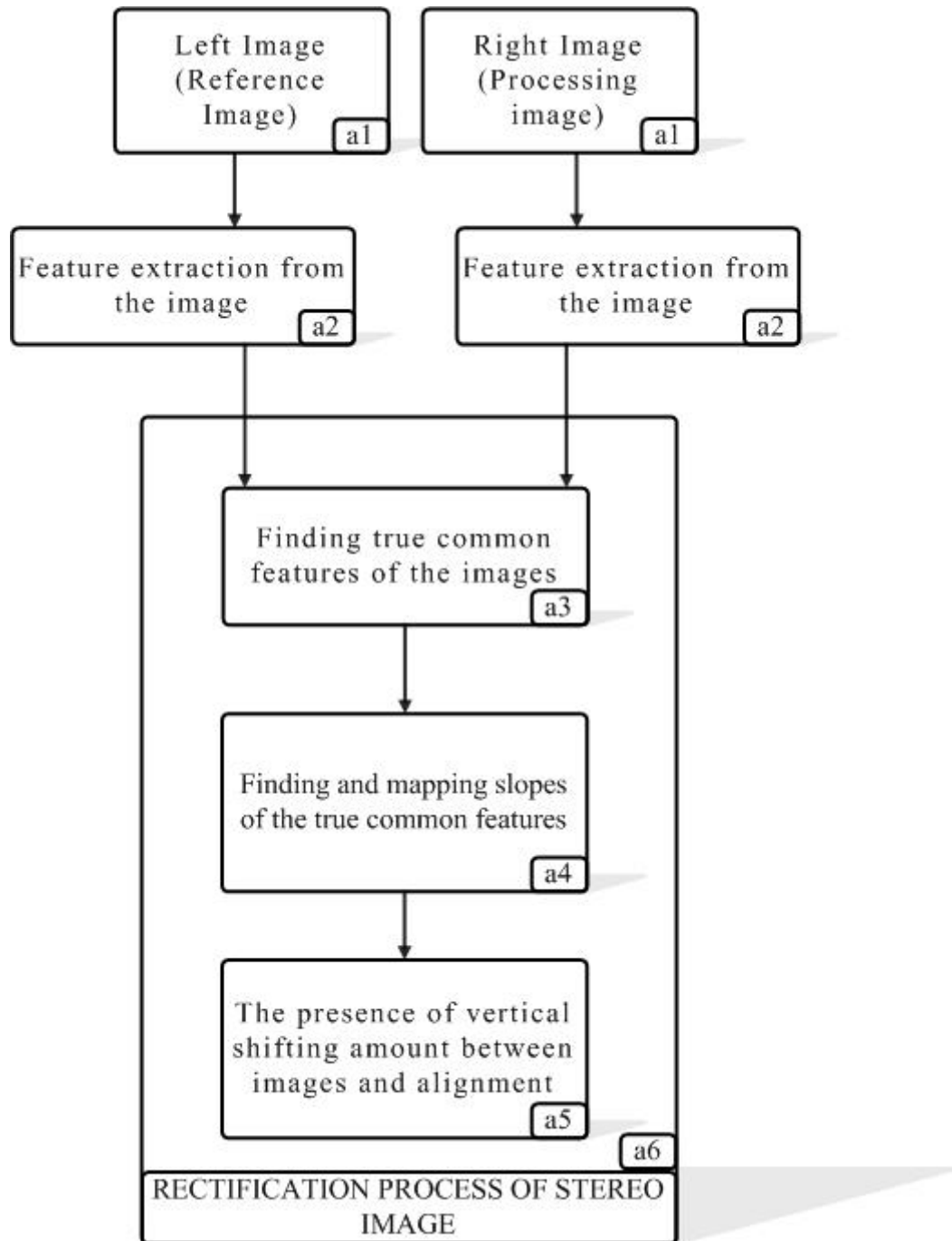


Figure 3.1 General structure of the algorithm

We obtain the image properties by using the Harris Operator Which is an area-based method and the sum of squared difference (SSD) has been applied to find real common features. Accuracy of a common feature and also the matching of the feature were realized by checking the slope between the features. In this thesis, the

slope checking procedure to match the common feature is the original addition to the algorithm.

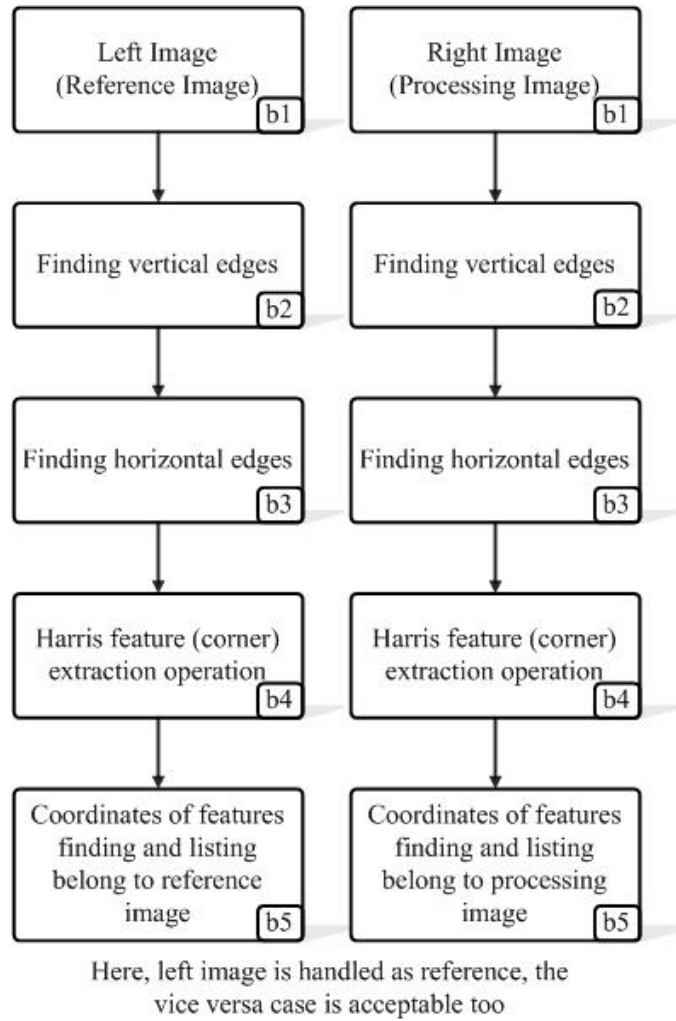
The methodology we propose is given previous page in the Figure 3.1 as the General structure of the algorithm.

In the flow chart which is given in Figure 3.1, corner features of left and right images (entering images; one of them is used as reference and the other as processing) from stereo cameras (a1) are extracted using the Harris Operator (a2). Finding property blocks (a2) are identical and independent from entering the images. However, according to the characteristics of the entering images, more features can be found by changing the parameters of the Harris operator. In this stage, it is necessary to find common features between these two images. An area-based method, (a3) Sum of Squares of the Differences (SSD), is given in equation 3.13, is used to find which features in the images are common. This method is also used to improve the accuracy and speed of the matching process. Before the feature matching, the true common features of stereo images are obtained in a matrix. In the next phase (a4 and a5) features that are common are matched, by finding slopes (tangents) between each true common feature. In this method, the features are handled according to their geometrical information and checking if they are actually common.

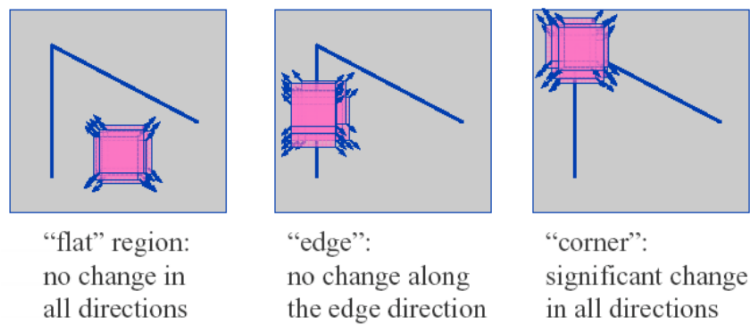
3.1.1 Feature Detection and Matching

As mentioned in the previous sections, Harris corner detection operation is used to extract features of the images (Harris & Stephens, 1988). Harris operator is first used by Chris Harris and Mike Stephens in 1988.

In stereo images or multiple image pairs, it is important to find common features of the images to process these images together. These features can be several things like colors, edges, corners, textures, shapes of the objects in the images. Harris Operator helps us to find corner features of the images which are suitable features for stereo matching process.



(a)



(b)

Figure 3.2 ‘(a)’ is the detailed presentation of a2 block (Harris Operator block) and ‘(b)’ the is the Harris Operator that finds out the corner features (Collins, 2007)

In Figure 3.2(a), feature detection process is given. Here, in b2 and b3 blocks, vertical and horizontal edges are found by using Sobel edge detection operator. Sobel operator is a gradient operator. Besides, In Harris Operator, it is given as derivation.

The purpose of derivation operation is to find significant changes on the images. Because there will be no changes on flat areas and along the edge directions (Figure 3.2(b)). It is expected that the corner areas will give these changes.

The Harris operator is given with the following mathematical expressions (Harris & Stephens, 1988) ;

$$I_x = G^x \otimes I \quad (3.1)$$

$$I_y = G^y \otimes I \quad (3.2)$$

In equations (3.1) and (3.2) I is the image that we will be detecting features. G^x and G^y are the gradient (Sobel) operators along the x and y directions, respectively. Here G^x and G^y are given in equations (3.3) and (3.4);

$$G^x = (-1, 0, 1) \quad (3.3)$$

$$G^y = (-1, 0, 1)^T \quad (3.4)$$

The equations (3.5), (3.6) and (3.7) are the products of the gradients in all directions. They will be used to express small movements of the image.

$$I_{xx} = I_x \cdot I_x \quad (3.5)$$

$$I_{yy} = I_y \cdot I_y \quad (3.6)$$

$$I_{xy} = I_x \cdot I_y \quad (3.7)$$

After that, these expressions are convolved with a filter “W”. The filter may be chosen as Gaussian or binary. In this thesis, Gaussian filter is chosen because it helps to eliminate noises in images.

$$S_{xx} = W \otimes I_{xx} \quad (3.8)$$

$$S_{yy} = W \otimes I_{yy} \quad (3.9)$$

$$S_{xy} = W \otimes I_{xy} \quad (3.10)$$

In the final stage of Harris corner detection operation, all the pixels are expressed in a matrix is given as $H(x, y)$; equation (3.11) is below. Matrix $[R]$, equation (3.12), is the corner response function that is equal subtraction of determinant and trace of the matrix “ H ”.

$$H(x, y) = \begin{bmatrix} S_{xx}(x, y) & S_{xy}(x, y) \\ S_{xy}(x, y) & S_{yy}(x, y) \end{bmatrix} \quad (3.11)$$

$$R = \det(H) - k(\text{tr}(H))^2 \quad (3.12)$$

Response matrix “ $[R]$ ” gives us the coordinates of the corners, edge and flat regions (Figure 3.2B). For this result, matrix $[R]$ must be the threshold value with appropriate numbers to find corners. Corners are represented with positive numbers; edges are represented with negative numbers and the flat areas are represented very small numbers in matrix $[R]$.

After, detection of the features for both images, they must be matched. For matching process, Sum of Squared Differences (SSD) algorithm is used. SSD is a correlation based similarity algorithm (Okutomi, Canon Inc., & Kanade, 1991). SSD formula is given below as equation (3.13);

$$F(x, y) = \sum_{(i,j) \in W} (I_1(i, j) - I_2(x + i, y + j))^2 \quad (3.13)$$

Here I_1 and I_2 are the images and the SSD looks at the feature points one by one with reference to one of the images. Result of the SSD algorithm for the features that belongs to same area in the images is small because of the similarity and they are matched each other.

In the previous stage, it is possible to find different number of features for the images with Harris Operator and some of them are not common and those are must not be matched. Here SSD helps to find possible common features. However, this matching process cannot be one hundred percent true for all features. It is possible to find false matching features.

In the next stage, the suggested method helps to rectify by finding vertical shifting amount to put the images in same epipolar lines if it is necessary. Also the effect of the false common features to rectification process is minimized with this method. The method uses tanject of the feature points and examines the features again in geometrical manner.

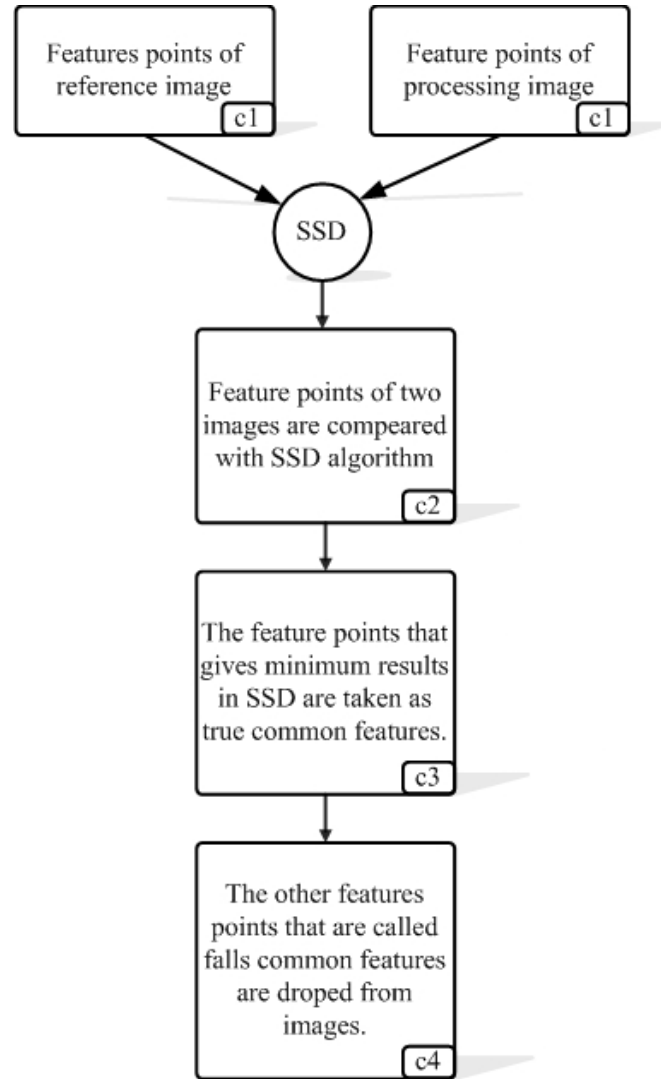


Figure 3.3 Detailed presentation of a3 block

3.1.2 Proposed Algorithm and Image Rectification

This method is proposed for parallel located cameras with normal lenses. The method examines the stereo images geometrically. The procedure is explained in Figure 3.5 briefly in the flowchart.

The suggested algorithm uses the matching features of the previous stage. The coordinates of these features are listed as “a” and “b” in Figure 3.7. Here F1 and F2 are the matched features that belong to left and right images. As said before, they are matched by using SSD but the features must be checked if they are on the same epipolar line and if not they must be positioned on the same epipolar line.

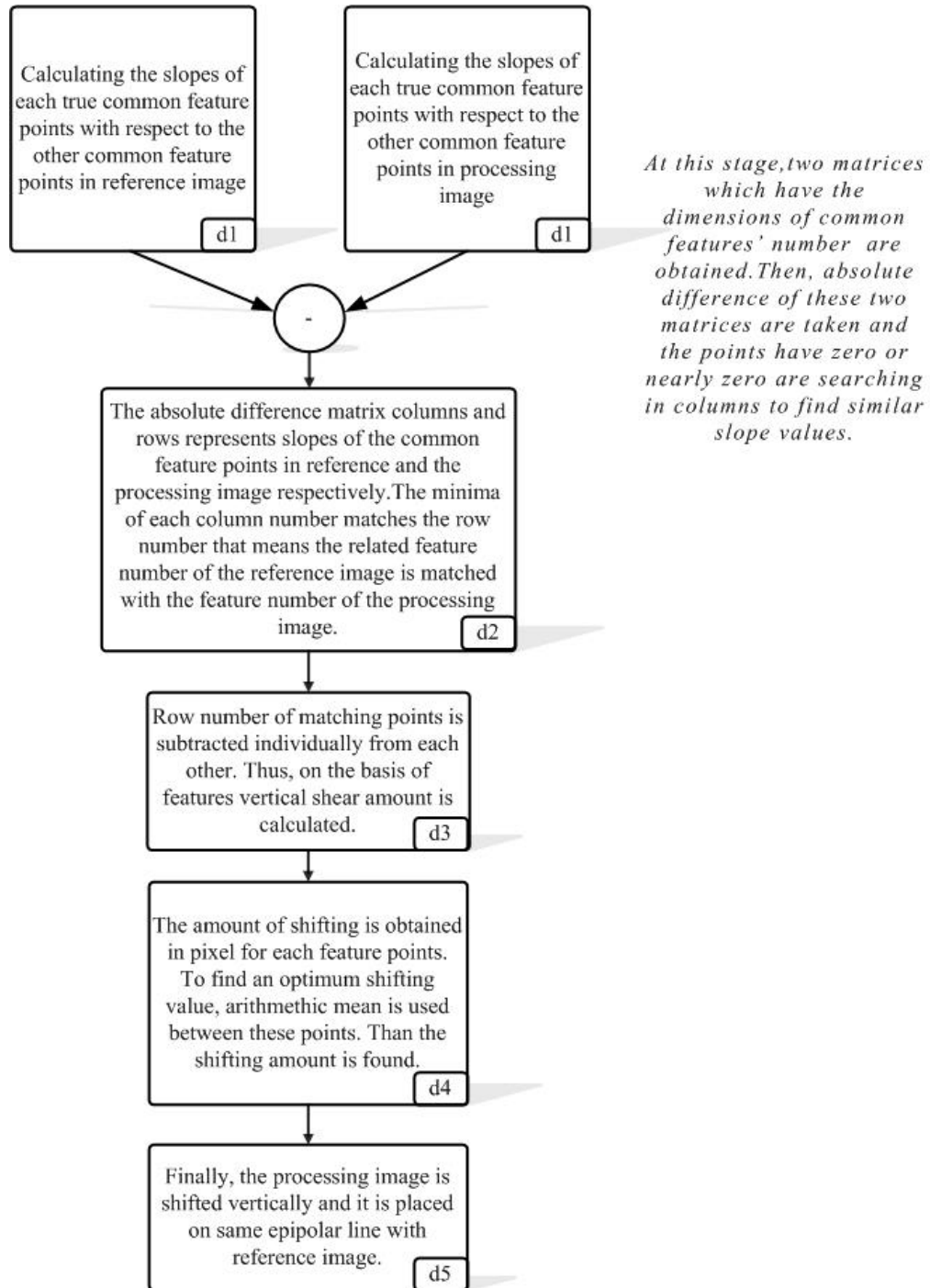


Figure 3.4 Detailed presentations of a4 and a5 blocks

To rectify the images, feature points are used to get geometrical information about the images. This info is the tangents of the angle that is between the x axis of the image and the line segments which are generated between these feature points.

In Figure 3.6 the calculation of tangents of A_1 angle between the feature line segments and the x axis of the image is demonstrated. Let (a,b) and (c,d) be the coordinates of the first and second feature of the first image, respectively, given as F(1,1) and F(1,2) in list “F1” in Figure 3.7.

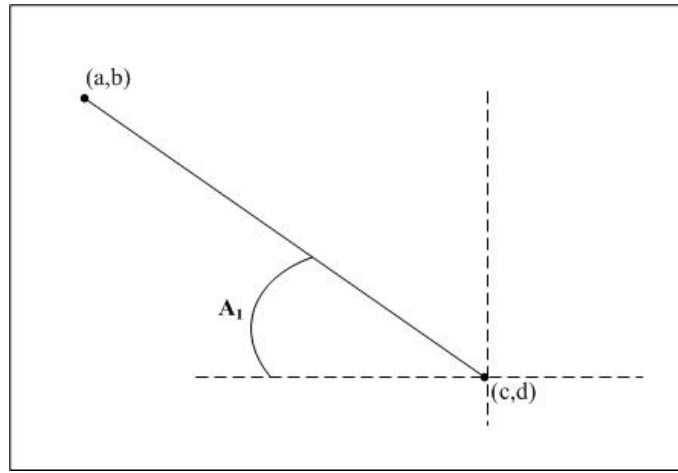


Figure 3.5 $\tan (A_1)$, slope of between any two features

To find $\tan (A_1)$ (a,b) and (c,d) coordinates;

$$\tan (A_1) = \frac{b - d}{c - a} \quad (3.14)$$

It is a very simple equation and gives us the tangent value of the angle. This tangent calculation is done between each feature point for both images and they are listed in two dimensional matrices [T1] and [T2] as shown in Figure 3.7 c and d respectively. These matrices contain all possible information for the line segments that can be drawn between feature points and also the listing is in an order for both images. Thus, in rows and columns for each image, there must tangent values that is the same or almost same each other. To find this “tangent matching” [T1] is subtracted from [T2] and the zeros or the minimum values are searched in each row of the absolute value of subtraction, [T] as shown in equation (3.15);

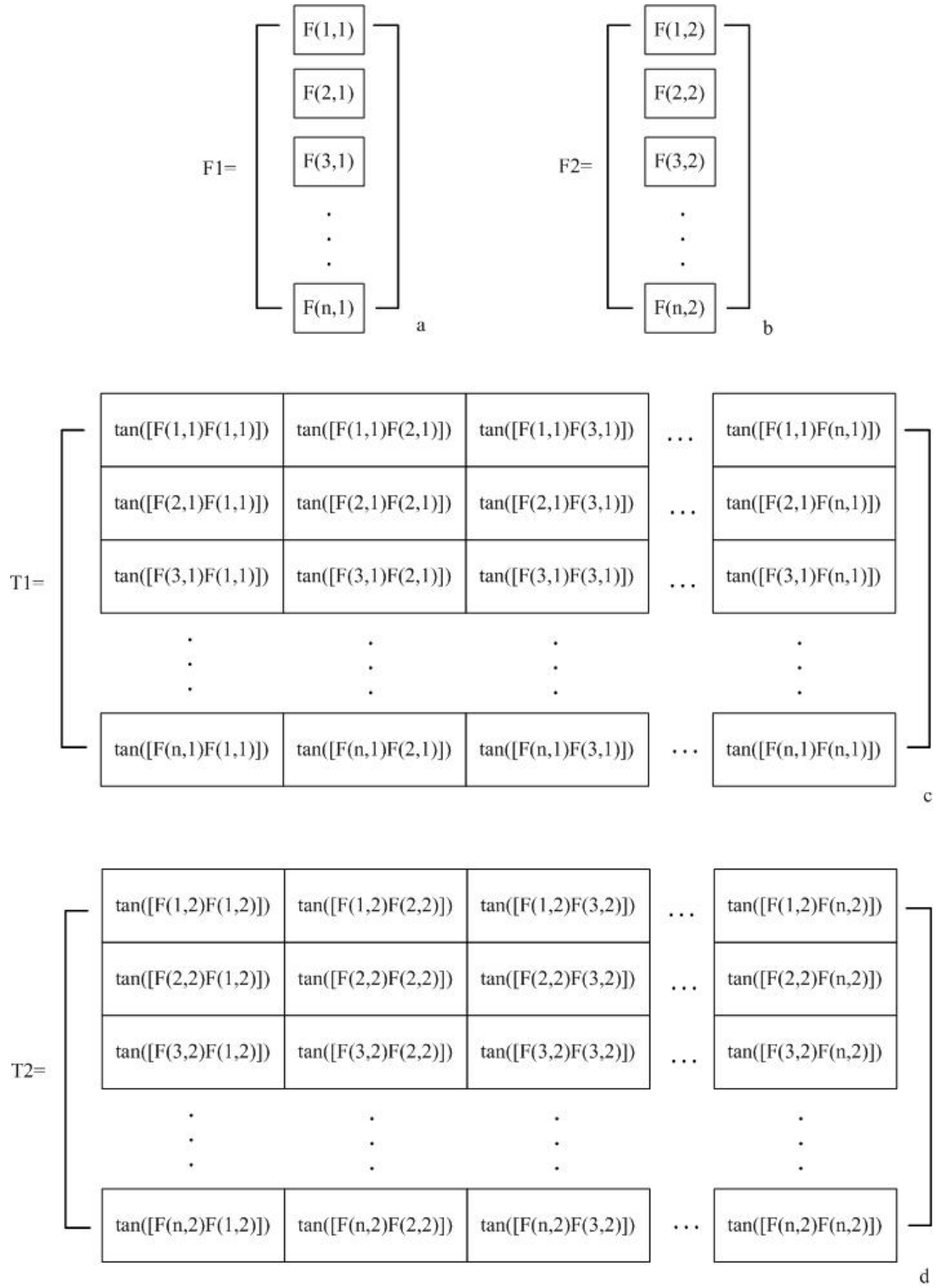


Figure 3.6 a and b are the list of true common features of the images, c and d are the matrices that contain tangents of line segments that are generated from the feature points

$$T = |T1 - T2| \quad (3.15)$$

In the matrix [T], the minimum values or zeros in each row number that corresponds column number gives us a matched feature. For example, if there is a zero or close to zero value in 4th row and 5th column in T, it means that the 4th feature of the first image matches the 5th feature of the second image.

After this matching process, row values of the first image features are subtracted from the second image and the arithmetic average values of these differences are evaluated. Finally, according to the sign of this result, processing image is shifted upward or downward (if the result has positive sign, shifting direction of the processing image will be upward, if the sign is negative, and the direction will be downward).

3.2 Disparity and Depth Map Estimation

After matching and rectification process, the rectified images are used in calculation of scene's depth. To estimate depth map from stereo image, disparity information of the stereo images are used.

In digital image processing, disparity expresses the differences of left and right images as mentioned before in Chapter two. Each object that is in different distance from the cameras has unique disparity value, thus the disparity can represent depth of the scene.

First of all, the possible disparity values of the stereo images are found and kept. Disparity values are evaluated keeping one of the images constant, and shifting the other horizontally. After each shifting operation, the images are subtracted from each other and these subtraction values are stored as cost values.

After finding all possible disparity values the minimum cost is searched between same coordinates of the costs matrices. However, these possible disparity values are not enough to generate the depth map. They must be examined with their neighbors. For this reason, a similarity based function must be used for good results. In 3.2.2 this function is mentioned.

3.2.1 Sum of Absolute Difference

For finalization of the procedure to generate depth map, a similarity matching algorithm must be used like SSD algorithm as mentioned before. Here, SAD (Sum of Absolute Differences) is preferred. It is similar with the SSD algorithm but less complex for computation. SAD is given in equations (Kamencay, Breznán, Jarina, Lukac, & Zachariasova, 2012);

$$S(x, y) = \sum_{(i,j) \in W} |I_1(i, j) - I_2(x + i, y + j)| \quad (3.16)$$

SAD is applied as windows “W” like 9x9, 15x15 and similarity is searched for disparities and which disparity level is appropriate for center coordinate in the window region is decided.

3.3 Simulation and Results

The aim of the simulation is to obtain distance of the objects from captured frames by two cameras. In order to do that depth map of the frames are created and then the distance of each pixel is calculated by using parameters such as pixel size, focal length of the cameras. Stereo matching is supported with rectification processes mentioned before in section 3.1.



Figure 3.7 USB2.0 stereo camera systems

The method, as shown in Figure 3.8, is tested on the images taken by placing two identical USB 2.0 RGB cameras placed a specified horizontal distance (about 15 cm). After the stereo rectification process results are used to generate depth map. Cameras have the image sensors with the resolution of 640x480 pixels and 3.85 mm focal length lenses. These parameters are used to estimate the distances of the objects from the stereo camera. Algorithm has been run on Matlab 2011 A and the results are obtained.

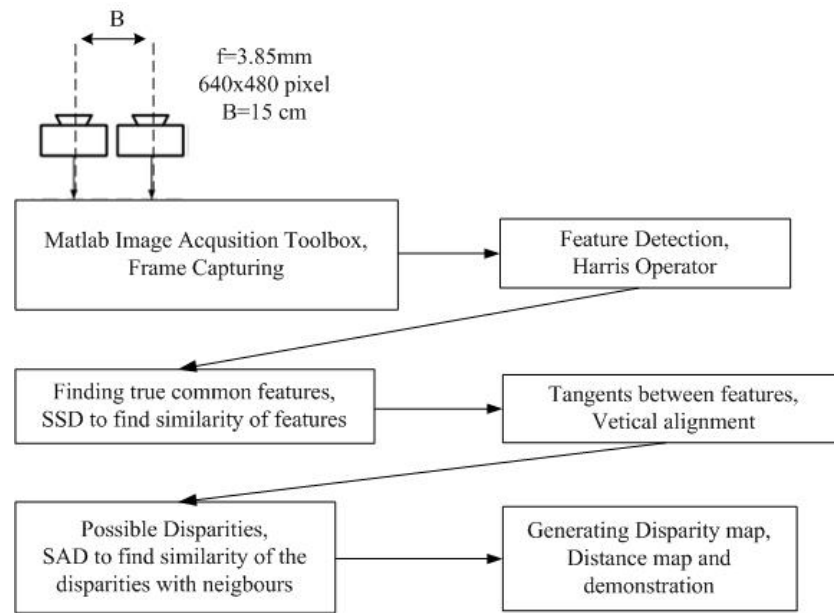


Figure 3.8 Block diagram of the depth map generator system

In Figure 3.9, the complete system is represented in block diagram. First operation is the capturing of the stereo images. In this section the Matlab Image Acquisition Tool Box is used to get these frames and some initializations are coded in the scripts.

After getting these image pairs, some feature information about the images is searched because these image pairs may not be on the same epipolar line and they must be aligned in same horizontal plane for exact matching. Here Harris corner feature detection method is used and features are obtained for each image.

After that these features are matched by using the SSD algorithm. It is then decided how many rows the images must be shifted to align in same horizontal plane and the shifting is done.

The next block is the depth map generation operation. In this stage, true disparity values are obtained from the difference matrixes. To do this, the algorithm searches the minimum difference values in the matrices and obtains which disparity matrix contains this value. The true disparity value writes a new image matrix and the depth map is generated accordingly.

After depth map generation, the real distances from cameras to the objects can be calculated with the formula given in (2.1). Finally, approximated real distances are obtained and demonstrated in new figure in Matlab.

3.3.1 Results

The image in Figure 3.10 is captured from the left camera and the Figure 3.11 is the image taken from the right camera. These two pictures are created as stereo image pair by using “Image Acquisition Toolbox” in Matlab. In Figure 3.10 and Figure 3.11, at first glance, the objects may seem to be located in the same vertical axis (it is assumed that same epipolar line for parallel oriented cameras). However, it will cause shifting errors in the matching process. Also these images are converted from RGB to grayscale images.



Figure 3.9 The image is captured by left camera



Figure 3.10 The image is captured by right camera

Firstly, the images are evaluated individually. The features of left and right images are extracted and the pixel coordinates of the features are listed and saved. Figure 3.12 and Figure 3.13, shows the feature points (with white square) that are extracted from the left and right images. The feature points are expressed in white square on the images by using color features (in here, features are formed by the corners, the horizontal and vertical edges) to these points respectively.



Figure 3.11 Corner features of the left camera image



Figure 3.12 Corner features of the right camera image



Figure 3.13 True common feature points of the processing image

If we look at the pictures in Figures 3.12 and 3.13, the common features on the two images can be noticed. However, many features are not common in both images. These are all the features were found individually for the left and the right image at the previous, -feature extraction- stage. However at this stage, the goal of suggested algorithm is to implicate common features as much as possible. In other words, the algorithm matches the common features in these two images. For example, in Figure 3.12, the feature point indicated by number 1 is not present in the region of number 1

in Figure 3.13. In contrast, the regions marked by 2 indicate real common features for each image, and both were found for the same object in the images.

As seen in Figure 3.14, real common feature points are selected in the processing image. Many feature points in Figure 3.13 has been eliminated. However, there are also some false common features in the result. They are ignored because, to find vertical shifting amount, an average value of absolute difference value of common feature points is calculated. Thus, the weights of these false common feature points are negligible.

After extraction of common feature points from the left and right images, the geometrical characteristics of the points are examined in themselves in order to ensure matching. Here, the means of geometrical characteristics is slopes (tangent) of the line segments that are generated from feature points with respect to each other. In Figure 3.15 and 3.16, a couple of the line segments from the common feature point on the door to other points are demonstrated. Tangents of these line segments is found and saved. This procedure is done for all other common feature points and tangents values are kept in two individual matrices for left and right images (For example, the tangent of the line segments between the first feature point of the left image and the second feature of the left image is written on first row, second column of this matrix). In these matrices, the row numbers of the coordinates that have the same tangent value are the order numbers of the feature in processing image common feature list (Figure 3.16) and the column numbers of these coordinates are the order numbers of the features in reference image common feature list. Thus, the coordinates that have same slope value give matching features from the lists.

After these operations, the vertical shifting amount is calculated by evaluating the difference of row values of these feature points. In this simulation, the vertical shifting value was found as 3 pixels for the processing images. So it means the processing image is shifted downward vertically by 3 pixels in order to align the objects in these images. After that, these results are used to form depth map.

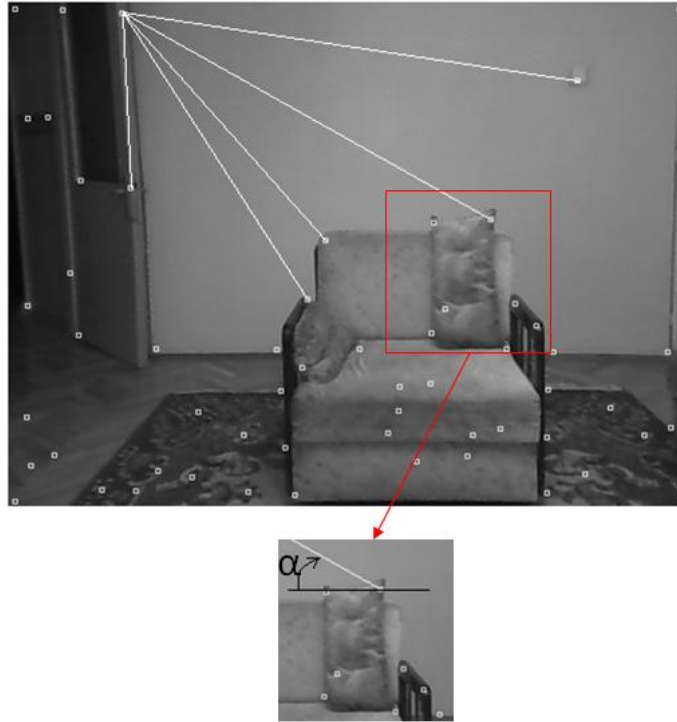


Figure 3.14 The α angle of between x axis and the line segment of two feature in left image

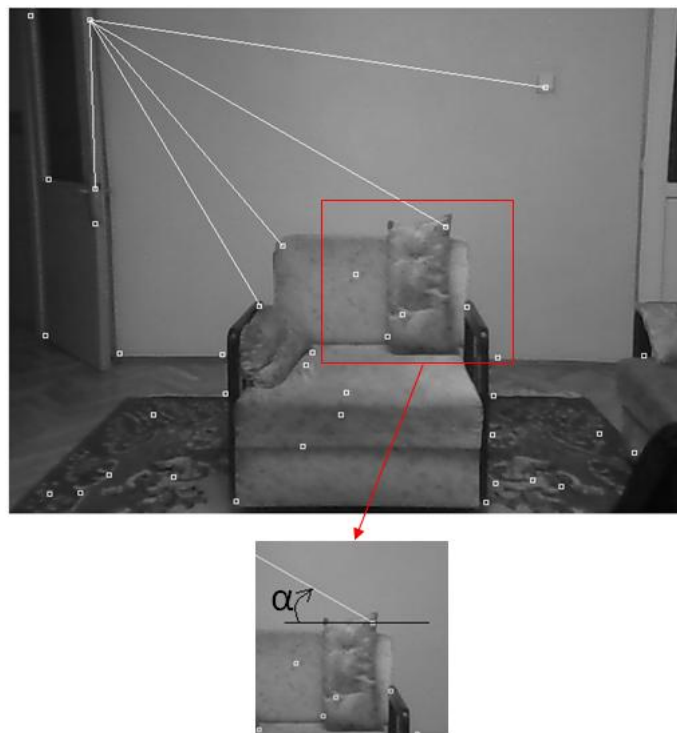


Figure 3.15 The α angle of between x axis and the line segment of two feature in right image

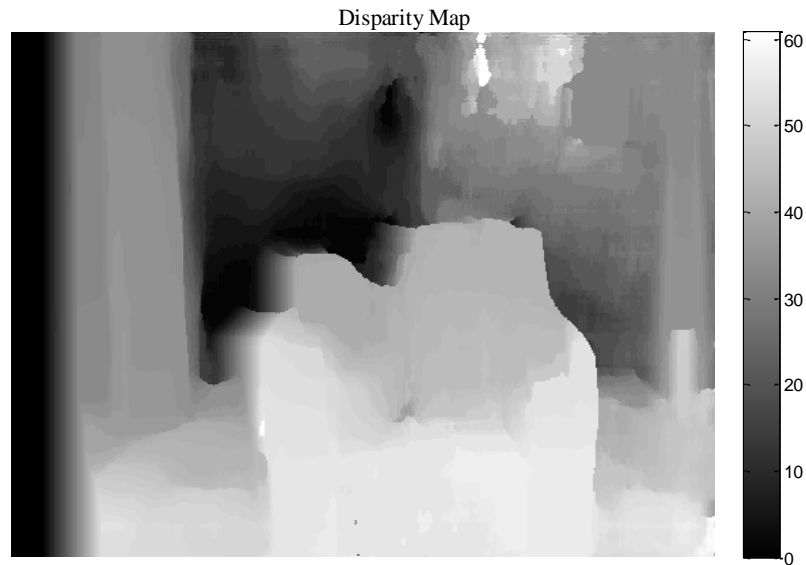


Figure 3.16 Disparity map of the tested images

After rectifications the depth map is found as given in Figure 3.17. It is expected that the disparity map intensity levels will be light nearer to the cameras and dark far from the cameras. So in Figure 3.17, the objects which are closer to the cameras have greater disparity values and the far objects from the cameras have smaller disparity values. It can be observed from the level bar that is on the right of the figure.

It should be mentioned here that disparity is inversely related with the depth map. To estimate the distance (or depth) of the object to the camera, focal length, pixel length and the distance between each cameras are also important parameters.

Cameras are separated 15cm from each other. The focal length is 3.85 mm which is labeled on the cameras and the pixel dimensions is calculated as 0.041 cm. By using these parameters, we can calculate the distance of the objects by using the disparity formula given as in (2.1). Thus the obtained distance or depth map is given in Figure 3.18;

As shown in the depth (distance) map, the far field has approximately 5 meters distance where the nearest object (the armchair) has about 250 cm- 320 cm distance from the cameras. The result is approximately same as the real distances.

However, the results in some areas on Figure 3.19 have inaccurately estimated disparity values. These defects result from the original images qualities. For example, a change of illumination across the views introduces ambiguities.



Figure 3.17 Distance map of the test image

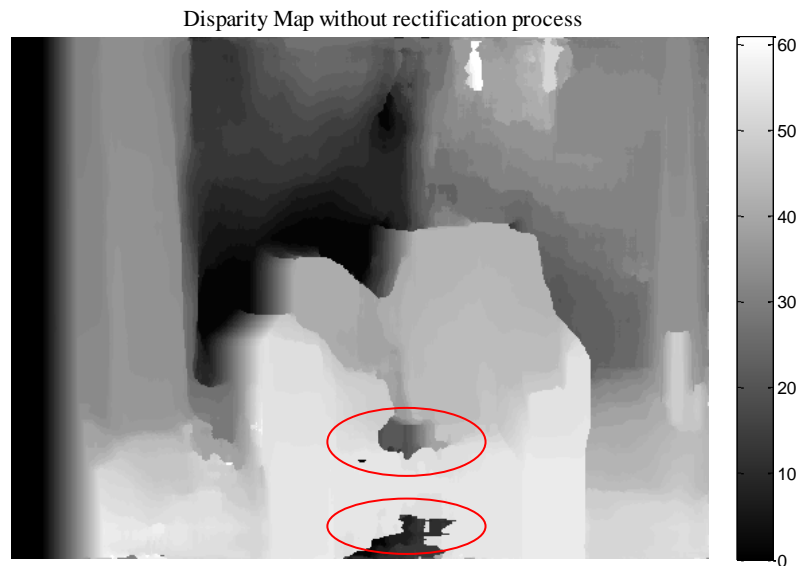


Figure 3.18 Distance map of the test image without rectification

After generating the depth map by using rectification value, also it is tired that disparities are calculated without any rectification process to show the benefits of

rectification process in same conditions (same method with same tools, same SAD window size (30x30)). It can be seen that in Figure 3.19 the some disparities on the armchair are calculated wrong. This is because a result of the unmatched pixel that is included in the disparity calculations. On the other hand these defective zones can be recovered by using bigger windows in SAD method but this time, some depth information can be lost. So the window sizes must be trained to find the optimum value.

While evaluating the depth map, there are wrong calculations in the background because of the illumination of the environment. If the images are inspected carefully, the wall on the background has different color tone from left to right. The images are captured in the room which has left side of the scene so the illumination is provided left hand side of the scene, which causes differences in the disparity at the background. For this reason the background seems to have a depth perception from left to right.

As a result, the depth calculation and the stereo matching method can work well in the good illumination of the scene because of the image quality. Also the stereo images can be processed to eliminate the noises before the feature extraction. The distances of the objects can be calculated as meter with this method by using depth map and the stereo camera system parameters. The proposed rectification method helps the rectification process and gives reliable results, but it can be improved by using other geometrical features (geometrical shapes can be searched instead of slopes between feature points) of the extracted features points.

CHAPTER FOUR

FPGA IMPLEMENTATION OF DISPARITY CONCEPT

In this chapter, the disparity phenomenon that is mentioned in the previous chapters is implemented on the FPGA (Field Programmable Gate Array) and tested in a test bed by using a stereo camera. Firstly the hardware design will be given briefly and then the results are evaluated from the test bed.

4.1 Hardware Design

The hardware is designed using Verilog HDL on Xilinx ISE 14.5 and implemented on Xilinx Virtex-5 XC5VLX50T FPGA chip. The FPGA development board is Genesys™ and the stereo camera board is VmodCAM™. Both of them have the Digilent brand. From references (Digilent, VmodCAM™ Reference Manual, 2011) and (Digilent, Genesys™ Board Reference Manual, 2013), additional technical information about FPGA board and the camera board can be accessed.

In Figure 4.1, the general structure of the hardware design is given as a block diagram. The first block is the camera driver block. The stereo camera board is connected to this block directly which initializes the camera on I²C bus and reads the synchronous signal of the cameras while sending clocks to the cameras. The camera board has two independent cameras that are integrated. The frames from left and the right cameras are buffered in the block RAMs.

The second block is the Disparity Map calculator block. In this block, the frames are stored as left and right in block RAMs. The first block is used to generate the disparity map with the disparity value that is given externally and the generated disparity map is buffered in the block RAM. The third block is the clock generator block. It generates necessary clock for the other blocks by using the 100 MHz system clock. Since some computations are done concurrently the design has the parallel processing ability.

The last block works as the HDMI driver. This block initializes the HDMI transmitter IC and sends the contents of left camera data, right camera data and the

disparity map to the IC. Also this block has a pixel clock and digital video synchronous signal generator to drive the HDMI chip.

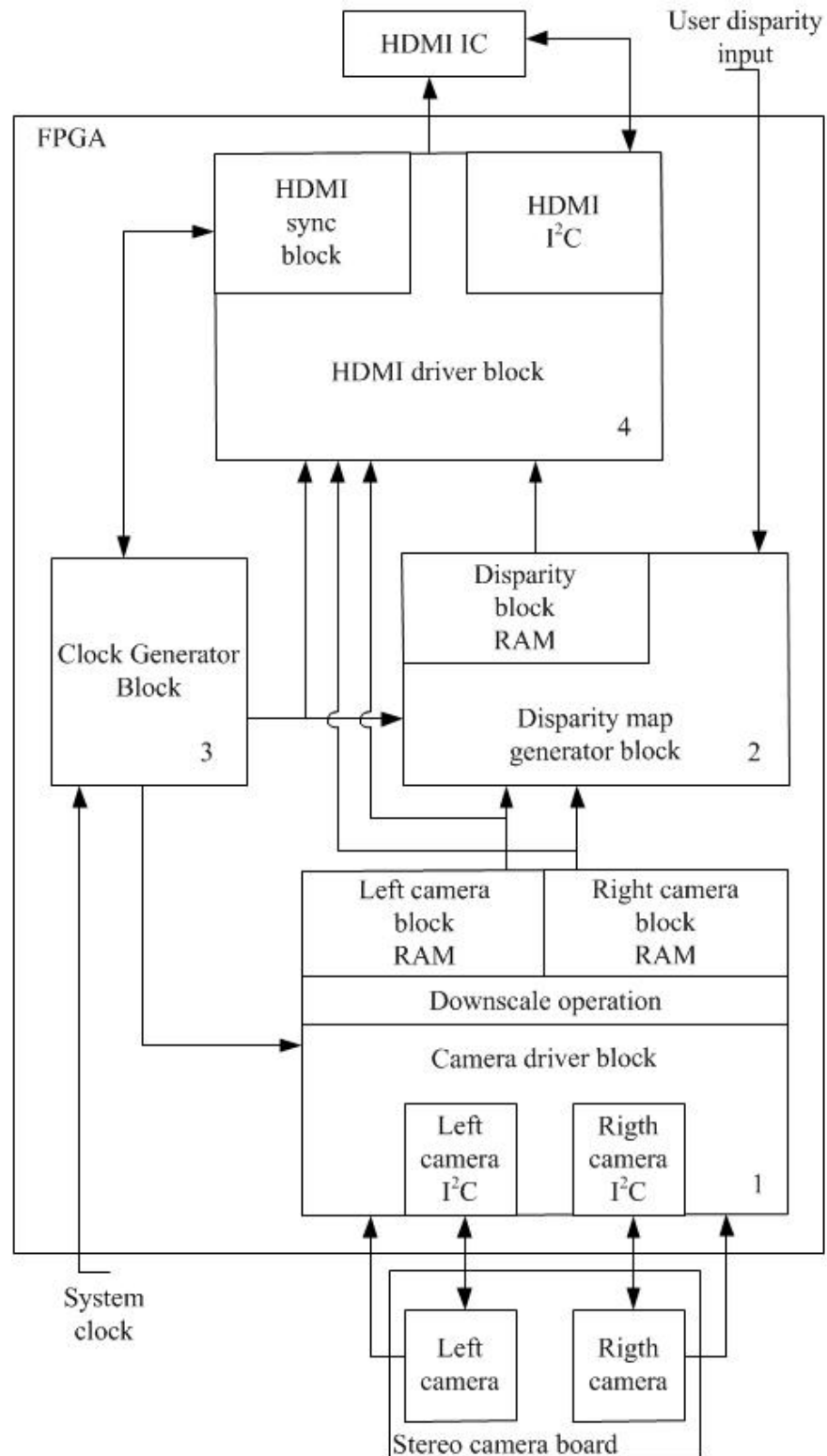


Figure 4.1 General Structure of hardware design in FPGA

4.2 Camera Driver Block

In this block, the raw image data that are sent by cameras are stored as one frame for right and one frame for left camera. To start and give initial parameters to each camera, block contains I²C master blocks for both of the cameras.

This block is connected to the stereo camera board with the signals that are shown in Figure 4.2. SDA and SCLK signals are the data and clock signals of the I²C bus, respectively. FPGA clock signal frequency is 24 MHz and is the main clock that is generated by clock generator block in Figure 4.1 for the image sensor IC to run.

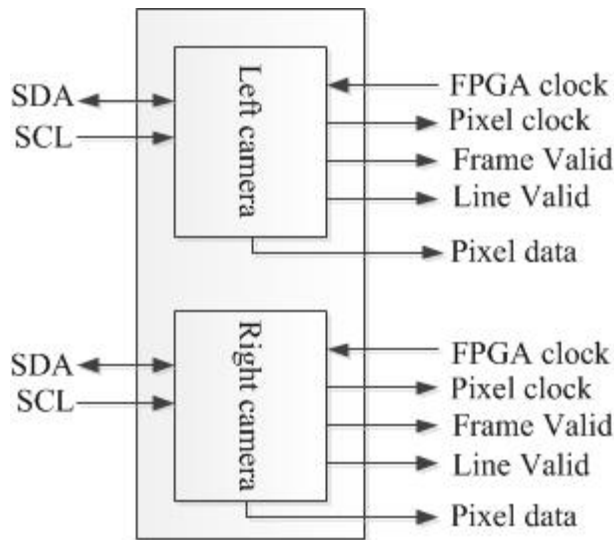


Figure 4.2 Signals on the image sensors as block diagram

The VmodCAMTM stereo camera board has two independent Aptina MT9D112 CMOS digital image sensors. These sensors have 1600x1200 maximum resolution at 15 fps and they can provide RGB or YUV image data format on output. In initialization stage, it was chosen as RGB 565 format as the output.

The image sensors have 8 bit parallel data output bus, but they can provide 16 bit raw pixel data in byte by byte. Least significant byte of the 16 bit raw pixel data is sent at the time interval of the even byte and the most significant byte is sent at the odd byte period. Each bit of the chosen format RGB 565 is given to the output with the positive edge of pixel clock as shown in Figure 4.3.

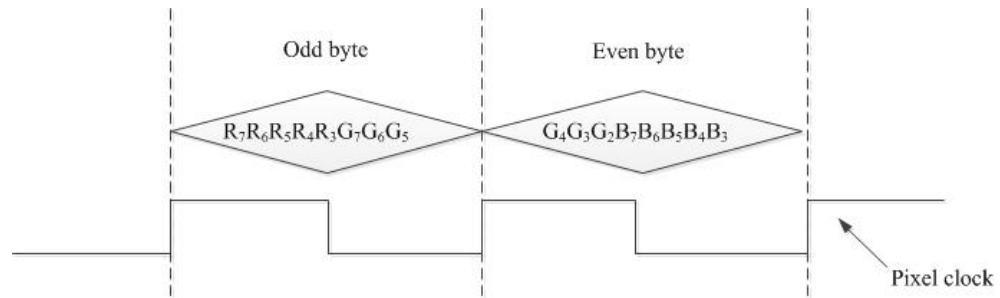


Figure 4.3 RGB 565 raw pixel format

Odd byte contains most significant 5 bits of red and the most significant 3 bits of green components. The most significant 5 bits of blue component is in the least 5 significant bits of even byte and the most significant 3 bits of even byte has the forth, third and the second bit of the green. Thus, the component pixel as RGB 565 format is generated. Finally, each component 16 bit in the block RAM of camera driver block.

The line valid and the frame valid signals are given to the output with the pixel clock to obtain rows and columns and the end of the frames. Thus, the column number of the image can be obtained by counting the pixel clock. However, as mentioned before, to collect bits of one pixel takes two clock sequences so the columns are counted in every two pixel clocks. On the other hand, the line valid signal is always high when the pixel data is being sent to output. At the end of the each row the line valid signal becomes low and waits for a couple clock cycles. Thus, like the pixel clock, if the line valid is observed and counted, the row number of the pixel at that time can be found. During each frame and at the end of the each frame it becomes low and waits for a couple of pixel clock cycles. In this way, the end of the frames can be observed. These coordinates information is important to form the images correctly in the block RAMs of each camera.

4.2.1 Block RAM in FPGA

Block RAMs can be generated easily by using Core Generator tool in ISE software. Different versions of FPGA chips can support various types and capacities

of block RAMs. For example, the Virtex-5 XC5VLX50T chip supports Single Port RAM, Simple Dual Port RAM and the True Dual Port RAM while having 2160 Kb maximum block RAM of data capacity.

In this design, Simple Dual Port RAM is preferred since both reading from RAM and writing to the RAM processes is needed. In Figure 4.4 symbol of Simple Dual Port RAM is shown.

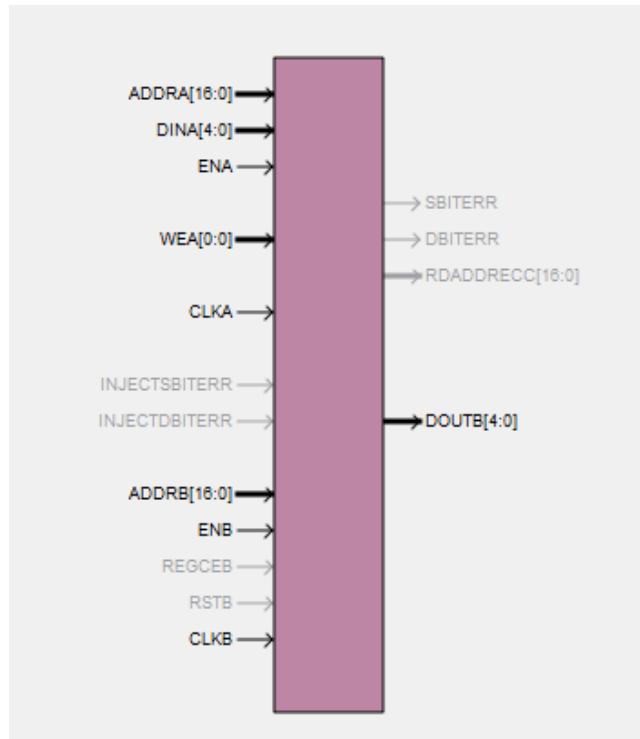


Figure 4.3 IP symbol of the Simple Dual Port RAM that is used in this design from Core Generator of ISE

The usage of this block RAM is very simple. In the first block in Figure 4.1, two of them are integrated to buffer the video frames coming from the cameras. The clocks and the synchronization signals are observed to obtain the pixel coordinates as presented earlier. This coordinate information is used to obtain RAM addresses to keep pixels in order. The linear equation used to map row and columns linearly to RAM is given as;

$$A = (y \times x_{max}) + x \quad (4.1)$$

Here, x and y represents the column and row numbers of the image, respectively where A is the address of the RAM and maximum of the x is the maximum width of the image.

As shown in Figure 4.3, the block RAM has two ports as A and B. A is the writing port and the writing operation is done with each acquired pixel. The writing operation clock cycle (CLKA in Figure 4.3) is the half of the pixel clock cycle so, as soon as any pixel becomes ready, it is written to the RAM to the correct address.

The read operation of the RAM occurs in different clock cycle but same way of addressing. Reading operation is triggered by blocks 2 and 4 in Figure 4.1 at the pixel clock of the 60 Hz full HD display.

However, the capacity of the block RAMs of this FPGA chip is not enough for the maximum resolution (1600x1200) of the two cameras in this design. For this reason, the frames of the both cameras are downscaled to the 400x300 resolution in this block before they are written to block RAMs. To do this, a simple method is used. First one of each row and the columns is chosen to represent others. It is show in Figure 4.4.

(1,1)	(1,2)	(1,3)	(1,4)	...
(2,1)	(2,2)	(2,3)	(2,4)	
(3,1)	(3,2)	(3,3)	(3,4)	
(4,1)	(4,2)	(4,3)	(4,4)	
⋮				

Figure 4.4 Downscale operation in this design

In Figure 4.4 only the (1, 1) pixel is written to represent the other fifteen pixels and itself. Even though the data is lost it provides efficiency on memory usage with the ratio of 1/16. After that another necessary efficiency improvements of memory is done and 16bits of each pixel is turned into grayscale by mean operation. So that each pixel has 5 bits color depth in gray scale. After this operation, an efficiency with the 5/16 ratio is provided again. After a gray scale stereo image is taken in FPGA block RAMs, 120K pixel is stored for each camera. It means 1200Kb ($5\text{bits} \times 2 \times 120\text{K}$) of data is kept and this is %56 of total block RAM capacity of the FPGA. These frames are sent to the disparity map generation block to be processed.

4.3 Disparity Map Generator Block

Disparity map was mentioned in the previous chapters and the depth map generation with the stereo cameras is based on searching the correct disparity value from the possible disparity values. In this block this concept is realized manually and the disparity maps are generated with according to user inputs. Also the rectification process is done in this block manually to show the correct disparity in correct distance from cameras. The aim in this block is to provide that the objects in different distance from cameras can be segmented by using stereovision with the disparity information.

This block (block 2 in Figure 4.1) works as the image processing block to generate the disparity map with the stereo image data that is coming from the Camera Driver Block. The Disparity Map Generator Block runs in synchronization with the pixel clock of the 60 Hz full HD display which is 148.25 MHz.

In block 1 (in Figure 4.1), the left and right frames with 400x300 resolution and 5bit gray scale are stored in the block RAMs. As mentioned before, B port of the Simple Dual Port RAM is for reading and works similar with the writing port. Here B port runs with 148.25 MHz and addressing formula is the same with equation (4.1). But this time the coordinates are generated by the synchronization signal of the HDMI.

Disparity map generator block takes the disparity values and the vertical rectification value from user as a binary number via 8 slide switches (from SW7 to SW0) that are on the FPGA development board. SW7 works as the selection button of the rectification or the value. The other seven bits are used to enter the value. When SW7 is high as in Figure 4.5, the other seven bits are written to a register as the disparity value and when it is low as in Figure 4.6, this value is written to another register as the rectification value. These two register values are used in the disparity map generation process. As mentioned in Chapter 3, the disparity map is generated by keeping one of the frames constant, and shifting the other as much as the given value horizontally. After the shifting operation, the images are subtracted from each other and these subtraction values are stored. This difference frame is the disparity map of the scene for the given disparity value.



Figure 4.5 SW7 is high and the disparity value is set as $(0010111)_2 = (23)_{10}$



Figure 4.6 SW7 is low and the rectification value is set as $(0000101)_2 = (5)_{10}$

To calculate the disparities, right image frame is shifted vertically as the value of the disparity register in hardware. The shifting operation is done with the same memory mapping formula which is given as equation (4.1).

Let the given disparity value be 23 in decimal number. This time the x variable of the equation (4.1) is shifted 23 pixels. At the same time, the right image is taken from the block memory without any shifting. Thus, the disparity can be calculated with the pixels that are not shifted, which are from the right image and shifted pixels which are from the left image. In Figure 4.7 the Verilog HDL code is presented for this function. In this code it is shown that the addresses are mapped with the same formula with the equation 4.1.

To make the result of subtraction positive (because disparity map is formed by the absolute differences of pixels) the code checks if the first value is bigger than second value in subtraction. If it is not, first value is made subtrahend and the second value is made minuend.

After calculation of the disparity of a pixel in a frame, it is not written directly to the disparity block RAM because of the clock signal makes the video data very noisy and the disparity map may not be generated on the screen, clearly. Because of this reason, the subtraction values are limited with the threshold value and new values are given for the intervals of threshold values.

```
address_reg_b_L<=400*(y_coor+rectification_level [6:0])+(x_coor+disparity_level [6:0]);
address_reg_b_R<=400*(y_coor)+(x_coor);
```

Figure 4.7 Verilog HDL codes for rectification and disparity values usage in equation 4.1

The rectification process is repeated with the same equation in (4.1) but this time the shifting is vertical. Thus, the value y in one of the images is shifted with the value in rectification register (Figure 4.7). This value is adjusted by observing the monitor until the objects in right and left are placed on the same epipolar line.

After these operations evaluated values are written to the disparity block ram.

4.4 Clock Generator Block

In the clock generator block, the necessary clocks are created for other blocks because the design has different processes that need different clocks. These clock are given in Table 4.1. The 3.3 V, 100 MHz clock is from the crystal oscillator on the development board is used as the source clock and the other clocks are generated from this clock by using the DCM_ADV (Digital Clock Manager with Advanced Features) primitive (primitives are the small blocks in Xilinx's FPGAs (Xilinx, 2009)) in Virtex-5. It supports various types of clock management features.

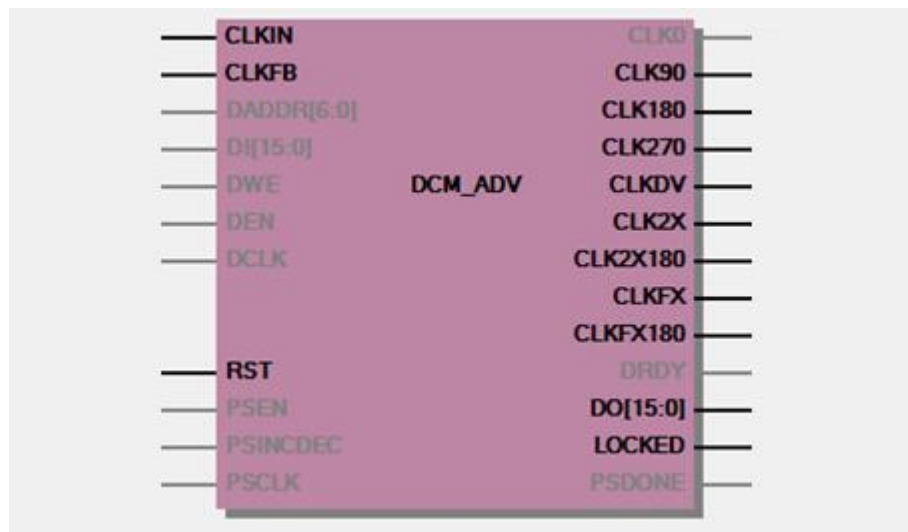


Figure 4.8 Block demonstration of DCM_ADV primitive in Virtex-5 (Xilinx, 2009)

For this design, CLKFX and CLKDV outputs which are presented in Figure 4.8 are used. CLKFX and CLKDV outputs are clock multiplexer and divider outputs, respectively. CLKIN is the 100 MHz clock that is used as the reference clock in this design.

The clock management functions are controlled via the parameters that can be changed by the designer. For example, to generate 24 MHz clock for cameras, 100 MHz clock is divided by 25 than multiplied 6 by writing this values in parameters of DCM_ADV primitive instantiation.

Table 4.1 Clocks and frequencies of the design

Clock	Clock Frequency	Clock Type
Clock Generator Block (in)	100 MHz	Single Ended
Camera Driver Block (out)	24 MHz	Single Ended
Disparity Map Generator Block (in)	148.25 MHz	Single Ended
Disparity Map Generator Block (out)	148.25 MHz	Single Ended
HDMI Driver Block (in)	100 MHz	Single Ended
HDMI Driver Block (out)	148.25 MHz	Differential

4.5 HDMI Driver Block

In this block, required signal are generated for the HDMI transmitter IC of the FPGA board (Chrontel 7301C) and the video data is sent to the LCD monitor via the HDMI interface. In Figure 4.9, the required signals to drive the IC are shown in block diagram.

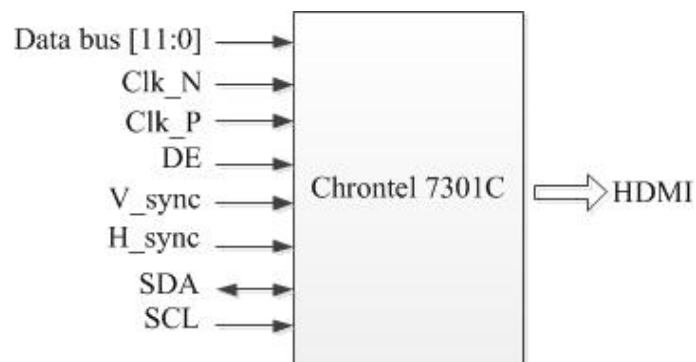


Figure 4.9 The required signal to drive Chrontel 7301C

The I²C control is necessary to initialize and start the Chronitel 7301C. There is an I²C master block in the HDMI driver block which starts the IC with its default register settings.

This IC needs a differential clock to run. As mentioned before, the block is feed by a 148.25 MHz single ended clock and it is converted differential form via OBUFDS primitive in Virtex-5.

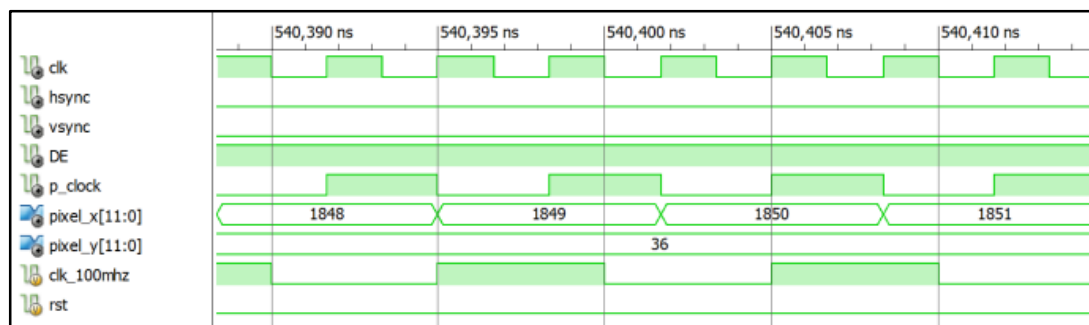
OBUFDS is a simple primitive, and abbreviation of differential output buffer. It can be added to the code with a very short instantiation; and it has one input and two outputs. The outputs are the differential signal form of the input signal.

The HDMI signal contains 24 bits RGB pixel data. But the data input bus of IC is 12 bits. The IC uses internal dual clock frequency to take 24bit RGB data inside. Here to send the pixel data 148 MHz clock is multiplied by 2 in DCM_ADV and 296 MHz clock is used to send the pixel data.

The other important signals are the synchronization signals V_sync, H_sync and DE. They are important because they must give the exact information about coordinates of the image. Similar to line valid and frame valid signal of the image sensor, they count the x and y coordinates of the image with desired resolution. In this design, they are set for 1920x1080 pixel resolution.

The “ISim” test bench tool of the ISE is used to observe these blocks’ synchronization signals. Some examples from the timing diagram of the test bench are presented in Figure 4.11(a), Figure 4.11(b), Figure 4.11(c) and Figure 4.11(d).

(a)



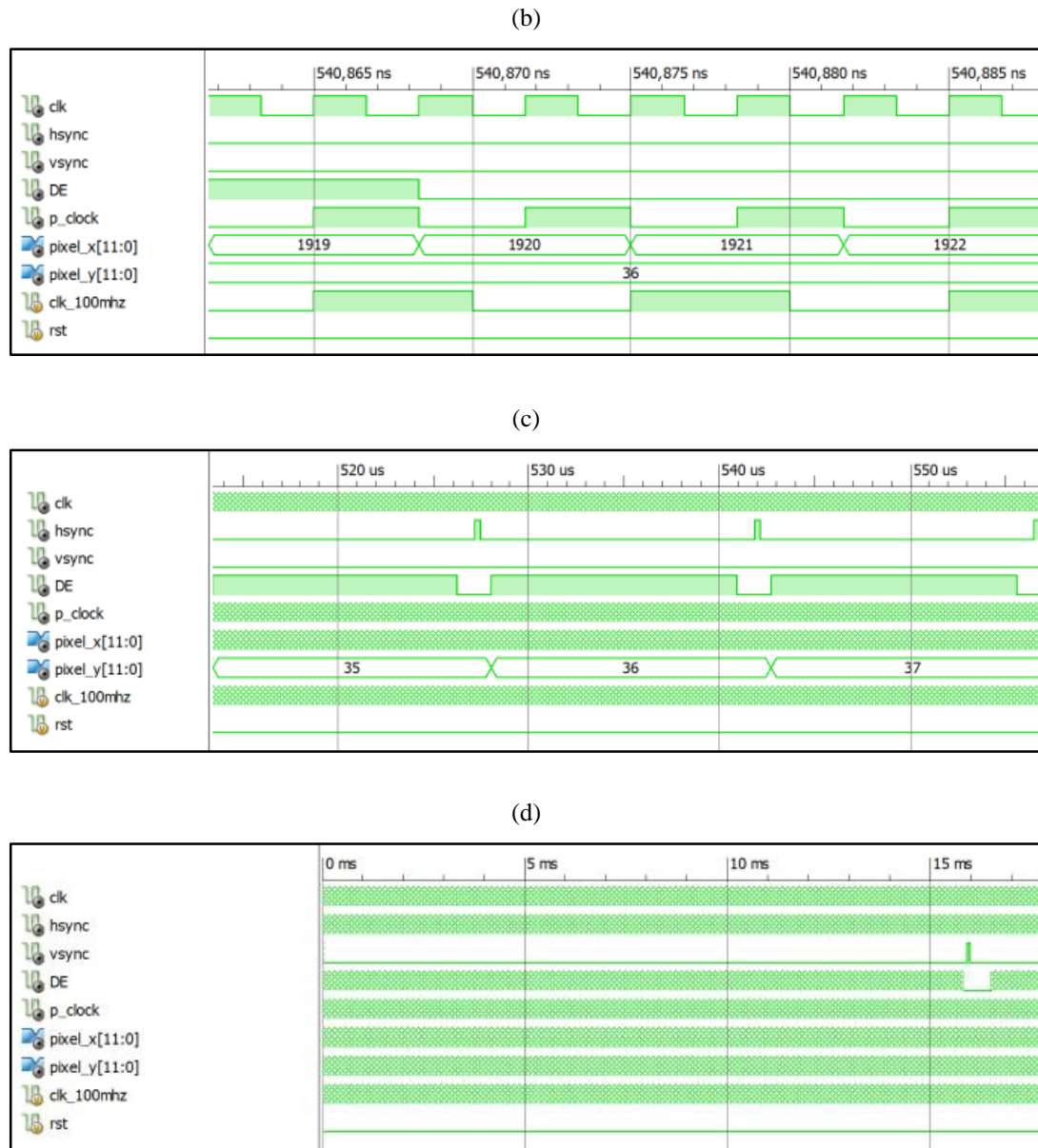


Figure 4.10 Timing diagrams of HDMI sync block from test bench

In Figure 4.10(A) the timing diagram is zoomed in to show pixel clock (148 MHz) is presented. The 296MHz “clk” signal can be seen here. It’s generated to provide the synchronization with the dual clock of HDMI IC.

In Figure 4.10(B) the ending moment of the x count for display is presented. Here DE signal becomes low after counting up from 0 to 1919. However, it continues to count because it is necessary for the wait interval before the next row and H_sync signal.

The H_sync signals can be seen in Figure 4.10(C). They become high on the wait interval of rows to activate the next row of the display. The “pixel_y” counting and “DE” signals can be seen

After enough zooming out of the timing diagram, the H_sync signal can be shown in Figure 4.10(D). This signal has a period of 16 ms.

4.6 Results of the Hardware Design

This hardware design aims to show the relationship between disparity and the distance of objects from the stereo camera. To do this a test bed and the objects are made and the hardware is tested. Figure 4.11(a) and 4.11(b) shows the test bed and the objects, respectively (in B, the blue ruler is 30 cm). Here background of the test bed is made of white fabric to make the objects visible.

During the test, these objects are positioned at several distances from the cameras and the disparity value is changed and the result are observed and noted. The results are displayed on the HDMI.

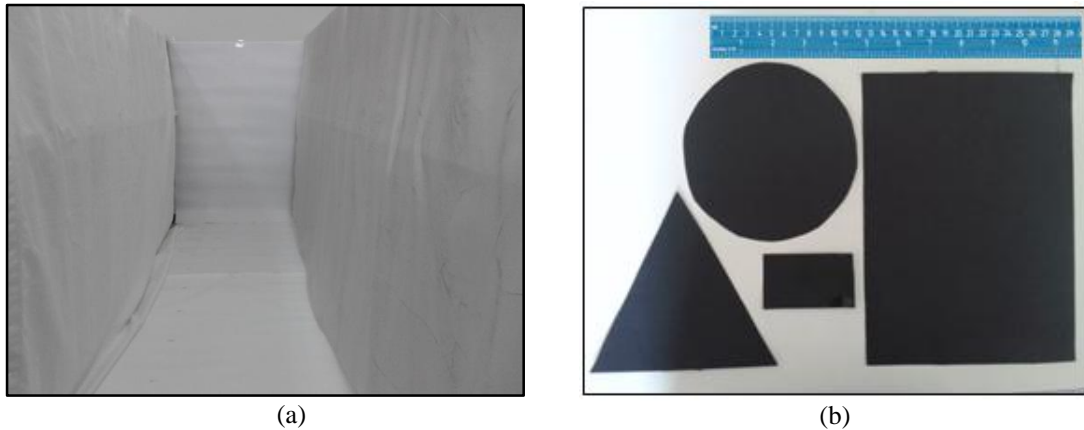


Figure 4.11 The test bed and the test objects

In Figure 4.12(a), (b), (c), the test setup is shown. In Figure 4.12(a) the Genesys development board is shown with stereo camera board. Here, the images are sent to HDMI monitor and displayed as left camera and right camera image in Figure 4.12(c) and the disparity map is in Figure 4.12(b).

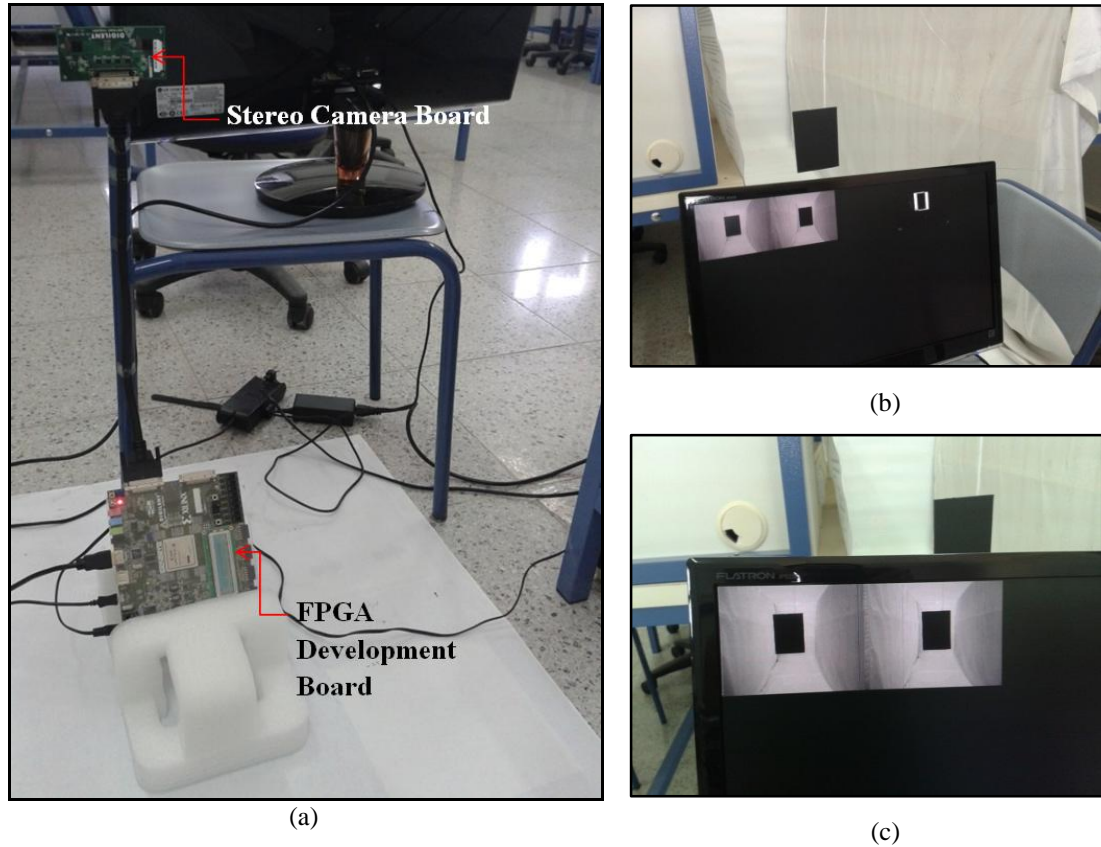


Figure 4.12 Stereo camera and FPGA development board in (a), the displays of the images from left camera, right camera in (c) and disparity demonstration in (b)

4.6.1 Rectangular Object, at 1 Meter

The rectangular object is positioned at 1 meter from the stereo camera. Firstly, the registers of disparity and rectification are zero so; the result stream is not aligned and is not at the correct disparity level, as seen in Figure 4.13.

After that the rectification value is four pixels is written to rectification register and alignment is provided like in Figure 4.14.



Figure 4.13 Disparity map with zero pixel disparity and zero pixel rectification



Figure 4.14 Disparity map after, right and the left images are aligned horizontally by entering rectification register $(0000011)_2$

The next operation is to find correct disparity level of the object at one meter from the cameras. In Figure 4.15 it is shown that the disparity value of $(0010110)_2 = (22)_{10}$ is the appropriate value for one meter distance ($(0010110)_2$ is an binary number and it equals 22 in decimal) . In Figure 4.15 it can be seen that the slide switch values are

set as SW7 is high for the selection of disparity register and the other seven bits are the disparity value.

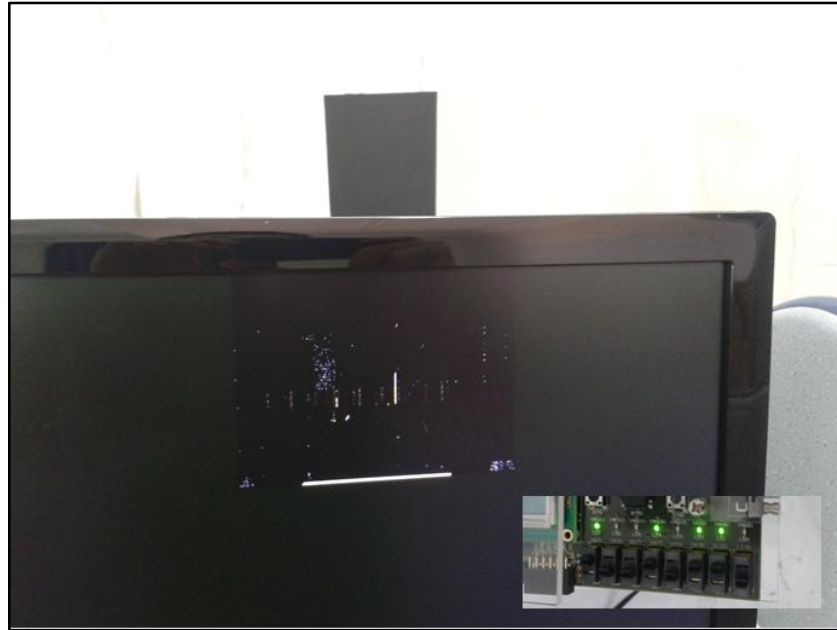


Figure 4.15 The appropriate disparity value of $(0010110)_2 = (22)_{10}$ for one meter distance from cameras

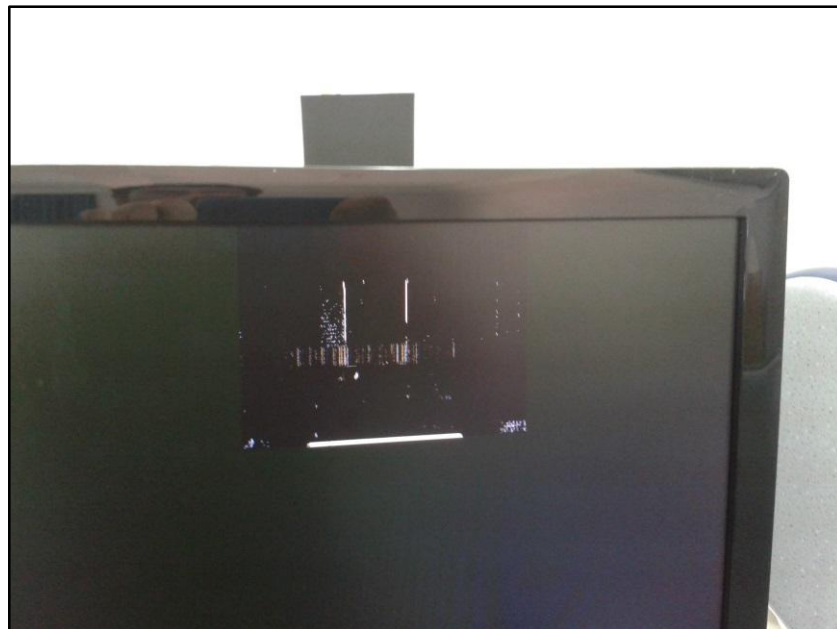


Figure 4.16 Disparity map for $(0010101)_2 = (21)_{10}$ disparity level at one meter



Figure 4.17 Disparity map for $(0010111)_2=(23)_{10}$ disparity level at one meter

As seen in Figure 4.15, if the disparity value is correct for the object, the left and the right images of the object almost overlap totally in the disparity map and it almost disappears. Also the disparities $(0010111)_2=(21)_{10}$ and $(0010101)_2=(23)_{10}$ are tried as shown in Figure 4.16 and Figure 4.17, respectively. It can be observed that when these disparity levels are written to the register the vertical edges of the object starts to appear.

4.6.2 Rectangular Object, at 2 Meters

In this test, the relationship between the distance and the disparity level is observed at two meters distance from the cameras. In Figure 4.18 it is shown that the disparity value of $(0000111)_2=(7)_{10}$ is the appropriate value for two meters distance and it can be seen that the slide switch values are set as SW7 is high for the selection of disparity register and the other seven bits are the disparity value.

As seen in Figure 4.18, if the disparity value is correct for the object at the set distance, the left and the right images of the objects almost overlap totally in disparity map and it almost disappears. Also the disparities $(0000101)_2=(6)_{10}$ and $(0001000)_2=(8)_{10}$ are set to show previous and next level of the disparity map in Figure 4.19 and Figure 4.20, respectively. It can be observed that when these disparity levels are selected, vertical edges of the object starts to appear.

There are also horizontal edges in these figures for two meters distance, although the rectification was set before for one meter test. This behavior is based on the downscaling operation that is done at the first block. The representation of the pixels is not very efficient because one pixel starts to represent sixteen pixels after downscaling. This issue will be discussed in the next chapter.



Figure 4.18 The appropriate disparity value of $(0000111)_2 = (7)_{10}$ for two meters distance from cameras

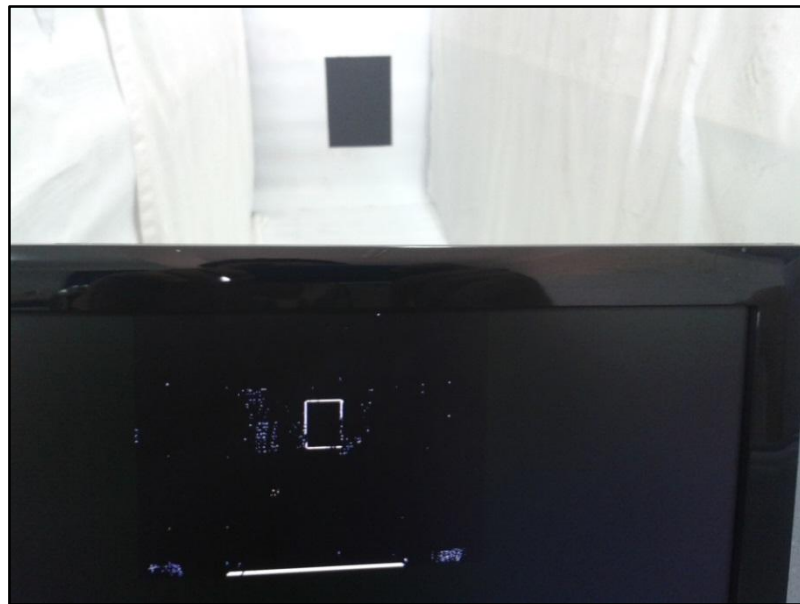


Figure 4.19 Disparity map for $(0000101)_2 = (6)_{10}$ disparity level at two meters

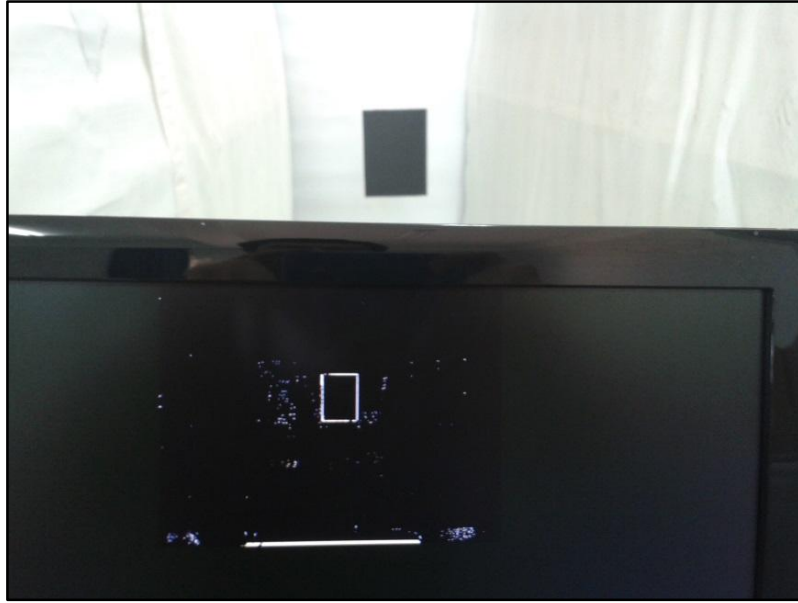


Figure 4.20 Disparity map for $(0001000)_2=(8)_{10}$ disparity level at two meters

4.6.3 Rectangular Object, at 0.75 Meter

Same procedure explained in sections 4.6.1 and 4.6.2 is applied when the object is located at a 0.75 cm distance from the cameras. In Figure 4.21, $(0011111)_2=(31)_{10}$ is observed for the appropriate disparity value. However, the effect of the down sampling is mentioned in the previous chapter, is seen here on disparity changes horizontally while the object is being closer to the cameras. For example in Figure 4.22, it can be seen that the disparity value $(0100001)_2=(33)_{10}$ still seems to be appropriate.



Figure 4.21 The appropriate disparity value of $(0011111)_2 = (31)_{10}$ for 0.75 meter distance from cameras

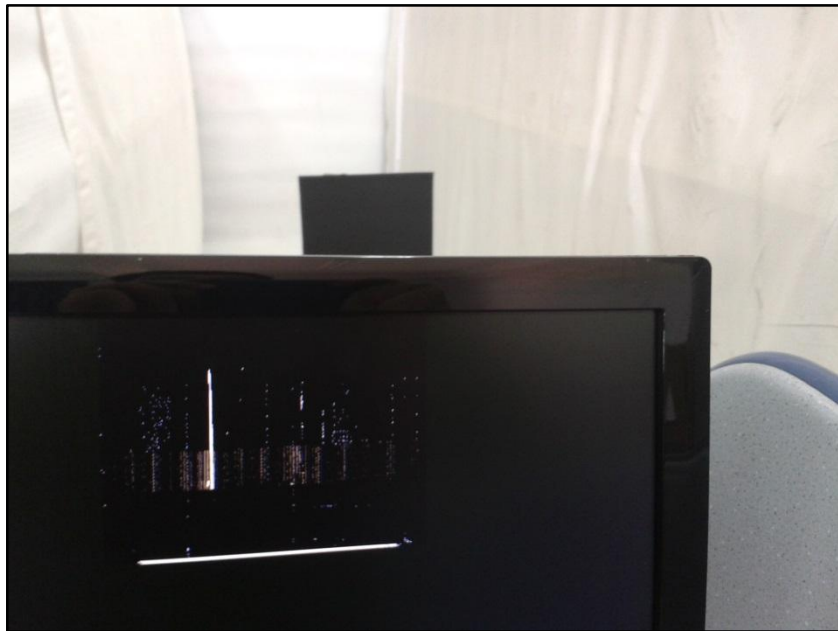


Figure 4.22 Another appropriate disparity value of $(0100001)_2 = (33)_{10}$ for 0.75 m distance from cameras

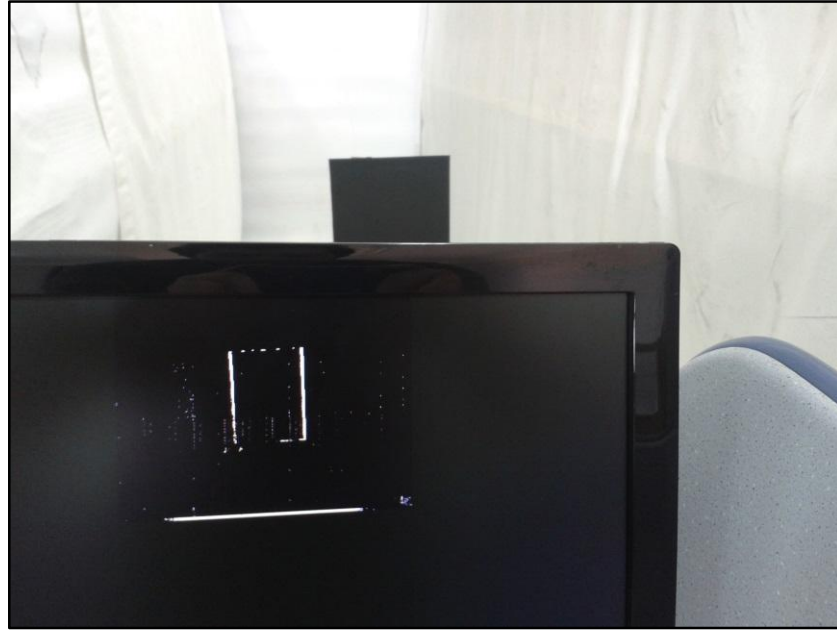


Figure 4.23 Disparity map for $(0011100)_2 = (28)_{10}$ disparity level at 0.75 meter

Likewise, in Figure 4.23 the disparity is given as $(0011100)_2 = (28)_{10}$ to show the vertical edges clearly on screen. The intermediate values of the disparity on this distance from 31 to 28 do not show the vertical edges.

This situation can be named as the system tolerance which corresponds to 8 pixels (max. 4 pixels from right and 4 pixels from left image) because of the 1/16 downscaling of the original image. The calculated and the observed values will be given in the same graph in Section 4.6.7.

4.6.4 Circular and Triangular Objects, at 1 Meter

The previous tests were done with the same rectangular object. Tests are performed at one meter from the cameras with the rectangular and the circular object and it is observed if any effect on the relationship between disparity level and the distance from camera occur.

It can be seen in figures from the 4.24 to 4.27 that, the disparity levels and the disparity maps are the same with the disparity maps of the rectangular at one meter. Similar with rectangular shape, the circular and triangular shapes almost disappear in

disparity map for 22 pixel disparity level at one meter from the cameras and the edges start appear for the next disparity value as seen in Figure 4.24 and 4.26.



Figure 4.24 Disparity map for triangular for $(0010111)_2 = (23)_{10}$ disparity level at one meter

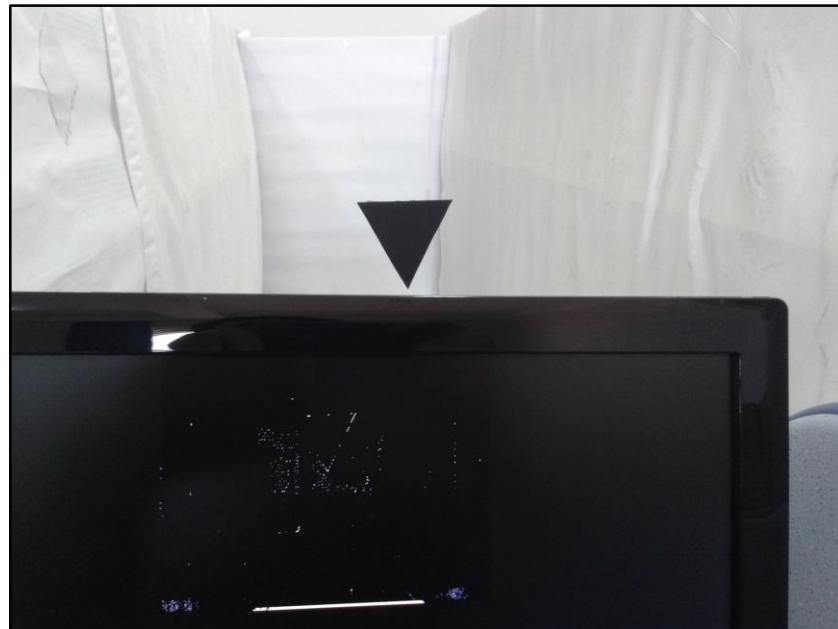


Figure 4.25 The appropriate disparity value $(0010110)_2 = (22)_{10}$ for triangular, one meter distance from cameras

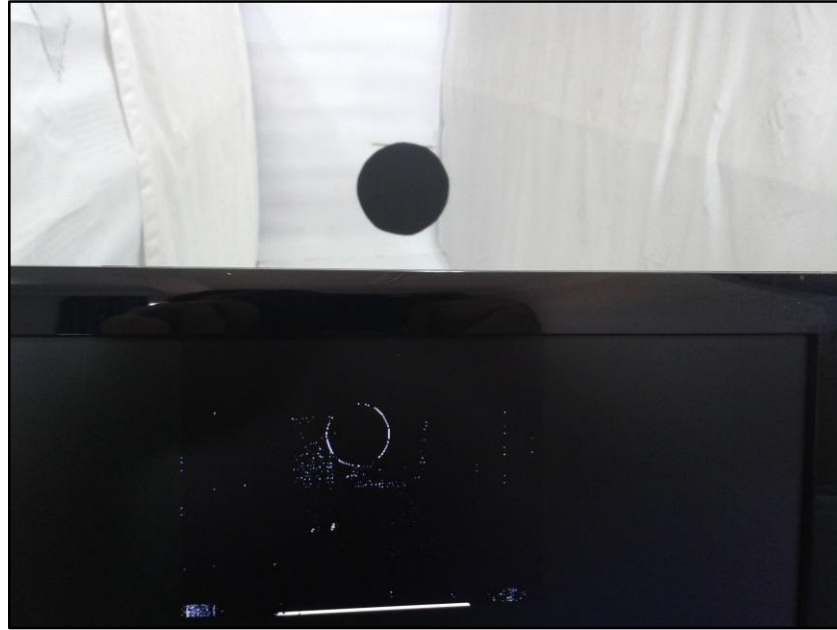


Figure 4.26 Disparity map for circular for $(0010111)_2 = (23)_{10}$ disparity level at one meter



Figure 4.27 The appropriate disparity value $(0010110)_2 = (22)_{10}$ for circular, one meter distance from cameras

4.6.5 Changing Distance when Disparity is Constant

This test is to check that the disparity map reaction while the object is moved in forward and backward direction when the disparity is kept constant. In Figure 4.15 it was given that the $(00101110)_2 = (22)_{10}$ disparity value is appropriate for one meter. This time the object is placed at 1.1 meter and 0.9 meter for 22 pixel disparity as it is shown in Figure 4.28 and 4.29 respectively and the result is observed. As it is expected the vertical edges appear for both movements.

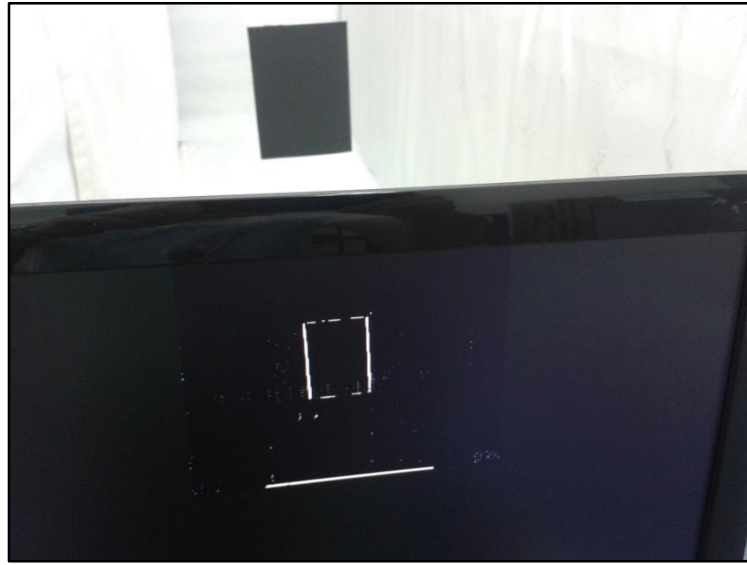


Figure 4.28 Disparity value is set $(00101110)_2 = (22)_{10}$ at the distance 1.1 meter

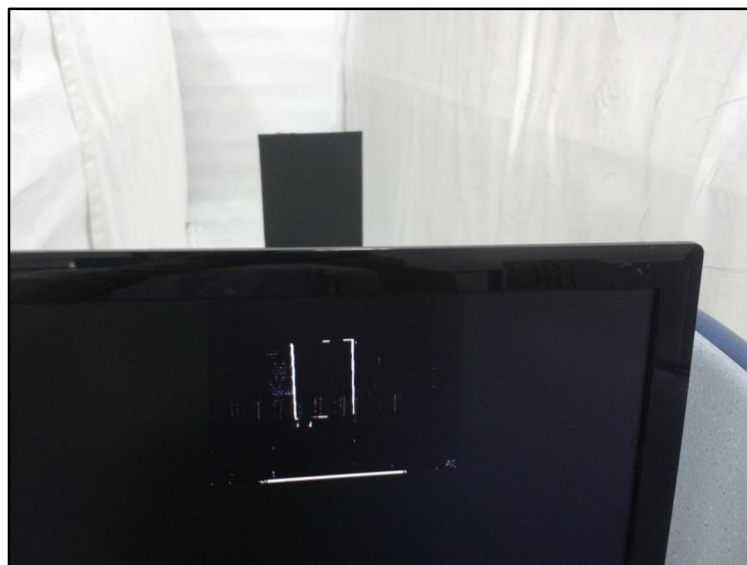


Figure 4.29 Disparity value is set $(00101110)_2 = (22)_{10}$ at the distance 0.9 meter

4.6.6 Stereo Field of View at 0.25 Meter

Stereo field of view (fov) is an important issue to reach the correct disparity value and it starts to become a problem in near field of the cameras. The lenses of the cameras have approximately 42° fov (field of view) and the starting point of the common stereovision area can be found by using this fov information.

In Figure 4.30, “A” field is the common stereo vision area and starts after approximately 6.9 cm (it is the height of the isosceles triangle which has 6.3 cm base edge and 50° vertex angle and it can be calculated with tangent theorem.) from the cameras. It means that the rectangular object does not fit in A when it is located at 25 cm from the cameras, because its width is 19 cm and the width of the common area is approximately 17 cm. In Figure 4.34 it can be seen that the circular object can almost fit the common stereo vision area at 25 cm distance because it has 15.5 cm diameter.

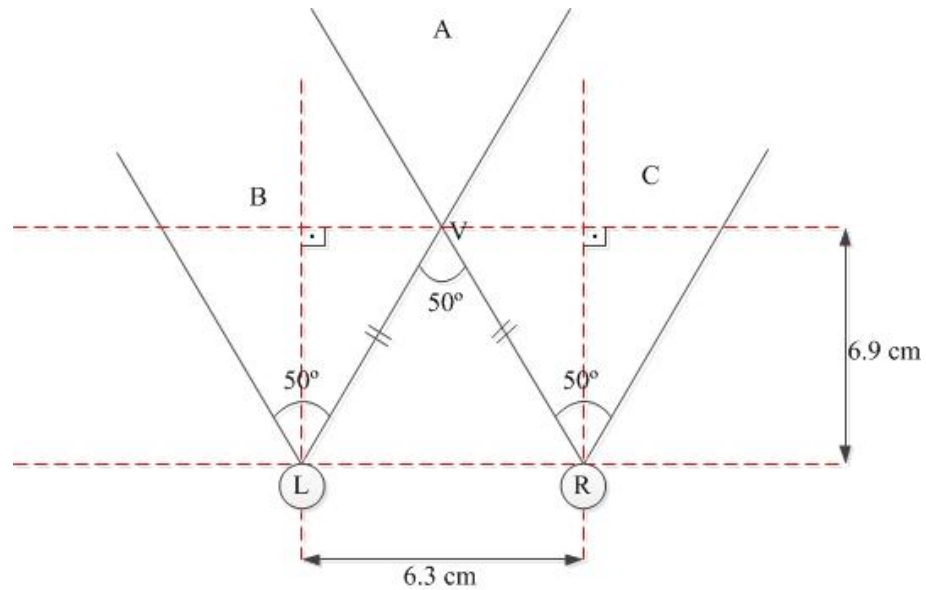


Figure 4.30 Stereo field of view calculation of the test setup

In Figure 4.31 it can be seen that the rectangular object does not fit in the common area and the disparity map is wrong. In Figure 4.32, the small rectangular shape can be seen for testing the disparity in 25 cm distance from the cameras. Here, the appropriate disparity level is observed as $(1101111)_2 = (111)_{10}$ pixels as it is presented in Figure 4.32.

In Figure 4.33 it is considered that at zero disparity level, the left and the right images are separated completely in the disparity map.



Figure 4.31 The big rectangular object is not fit in near field of cameras and disparity map is not generated correctly.



Figure 4.32 Small rectangular is used to find appropriate disparity in 25 cm from cameras and it is observed as $(1101111)_2 = (111)_{10}$ pixel

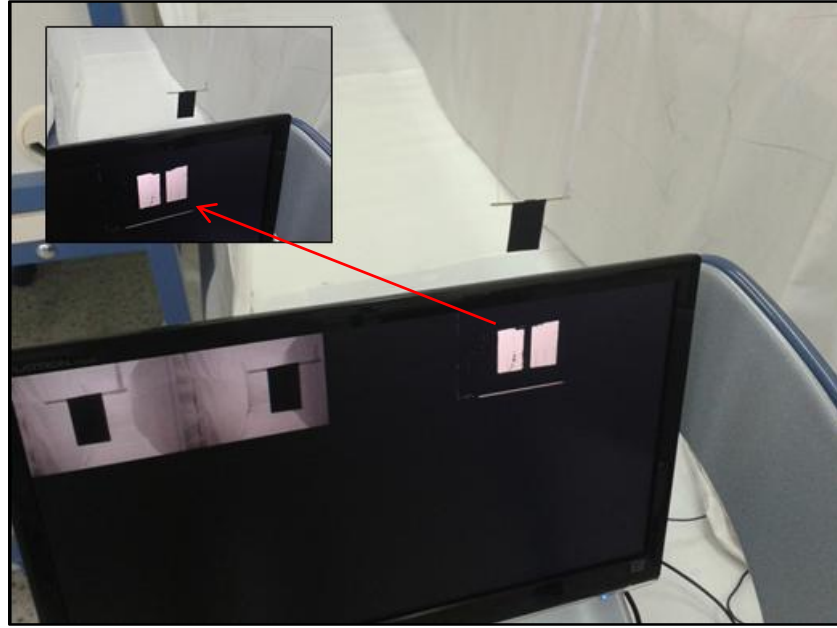


Figure 4.33 At 25 cm distance, left and right images of the small rectangular is separated each other completely in disparity map.

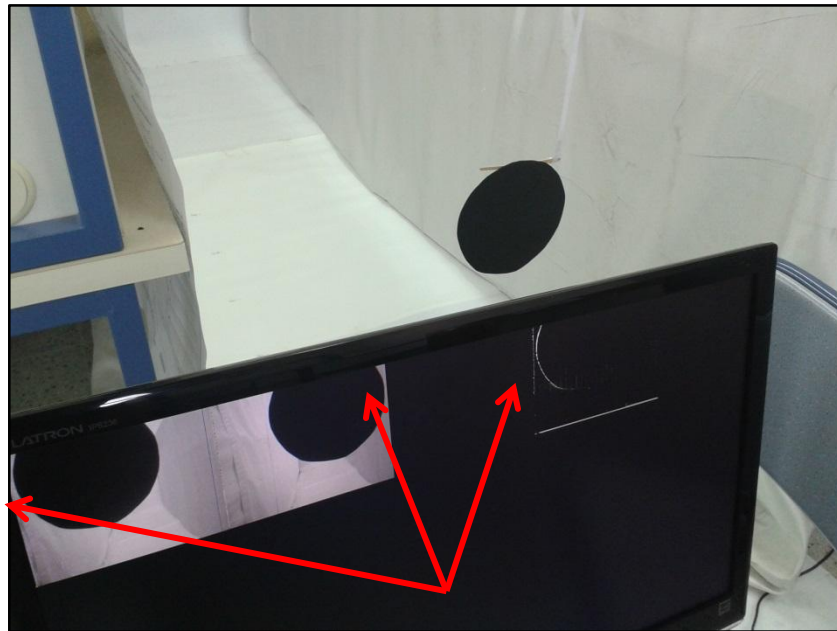


Figure 4.34 Circular almost fit the common stereo vision area at 25 cm distance because it has 15.5 cm diameter.

4.6.7 The Observed and Calculated Disparity Values

This setup is tried from 25 cm to 3 m in each 25 cm and the appropriate disparity levels are observed. And these disparity levels are compared with the calculated disparity levels.

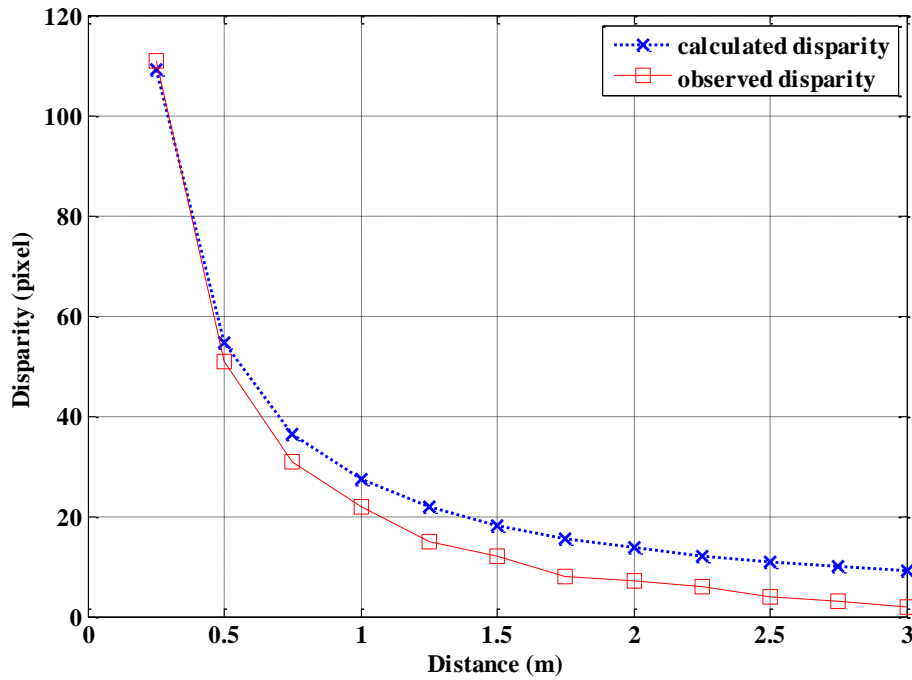


Figure 4.35 Graphical demonstrations of the observed and calculated disparity values

As it is seen from Figure 4.35 and Table 4.2, the observed and the calculated disparity values have no difference bigger than 8 pixels disparity that is mentioned before as tolerance of the design.

The calculation is done with the disparity formula given as equation (2.1). Here, the lenses of cameras have 3.81 mm lens (f in equation (2.1)) and distance between cameras is 6.3 cm as similar with human eyes. The formula gives the disparity result in cm when the other parameter is in cm. These disparity values are changed to pixel value by dividing them to pixel width of the image sensor. The image sensors have $2.2 \mu\text{m} \times 2.2 \mu\text{m}$ dimensions in the test setup and the pixels were downsampled. Thus the pixel width is assumed here, 4 times of real value ($2.2 \times 4 = 8.8 \mu\text{m}$) because of the 1/16 representation of pixels.

Table 4.2 Values of the graph in Figure 4.35

Distance from cameras (m)	Calculated Disparity (pixel)	Observed Disparity (pixel)
0.25	109	111
0.50	54	51
0.75	36	31
1.00	27	22
1.25	22	15
1.50	18	12
1.75	15	8
2.00	13	7
2.25	12	6
2.50	11	4
2.75	10	3
3.00	9	2

Although the differences are almost stable after 0.5 m from the cameras, the differences in near zone (in 0.25 m and 0.5 m) are smaller than the others. This situation is about the inverse ratio between the disparity and the distance. In near zones the disparity values are bigger and it means that the distances are represented with more pixels than they are represented in far zones from the cameras. Thus the disparity values can be observed more similar with calculated values in near zones than they are in far zones. On the other hand, it must be noticed that the observed values depend on the observer and they can vary a few pixels.

CHAPTER FIVE

CONCLUSION

In this thesis, the relationship between disparity and the distance of an object in a scene is examined and the images are segmented with the depth information. The advantage of stereovision is used to obtain the depth information. Also the stereovision system is implemented on an FPGA.

Firstly, Matlab is used to simulate the methodology. In this study, after feature extraction and matching process, the tangents between the features are used for rectification of the stereo images. After that, depth map for the stereo image images is generated by using the disparity of the stereo images in accordance with the real distance of the object in the scenes. In following study, disparity is implemented on an FPGA and the relation between the disparity and the distance of an object from the cameras is tested in a test bed and the test object is segmented according to the distance.

In simulation, it is observed that the depth map is generated more correctly when the suggested rectification is used. This rectification method uses simple geometrical information in the image. The tangents between the feature points are used to rectify stereo images. The suggested method is tested on the stereo images that are captured by parallel located cameras with normal lenses and the method works well for such images and gives the vertical shifting amount for an image in stereo pair. The rectification is done with this value.

The depth map, in simulation is not generated perfectly and some defected areas are observed in the depth map. However, the demo images are captured by ordinary USB webcams in an ambiance that does not have a good lighting and images are not modified before any stage of general methodology. Although, the objects in depth map is segmented correctly and the distance map is generated by using the camera parameters gives the correct distances.

In the hardware implementation stage, the design is made to make functions work concurrently in FPGA. Capturing processes are done at the same time from

independent stereo cameras. And this video stream is buffered, processed and sent to the display at the same time. The process is done in the 60 Hz full HD refresh time. For this reason the process can be called real-time. Also some design in hardware is done for the efficient usage of the block memories of the FPGA. To do this the video stream coming from the cameras are down scaled.

In hardware design, the rectification process is done manually with the user input. The segmentation of the test object is realized with correct disparity information at the correct distance.

The appropriate disparity levels in test points are kept and compared with the theoretical calculated disparity values. In this comparison, a tiny amount system tolerance is observed because of the downscaling operation in the FPGA. This tolerance is commented that the representation of any point in original image is not enough in downscaling the image.

5.1 Future Works

Some mentioned methods in this thesis are open for improvement. The proposed tangents method that is mentioned for the rectification of stereo images is a new concept. The performance of this method can be improved.

In the design of the FPGA, block memories are used in processes but they are not enough for bigger resolution and more complex task like depth map generation. However the skeleton of the hardware design is given in the thesis and by improving the memory options, the design can be developed to process more complex task and big images. To do these, SDRAMs can be used with an appropriate hardware design.

In this thesis, the disparity is one of the mainly discussed subjects. And it is one of the outputs of the stereovision a system. The disparity information is used the generate depth information in such system. After interpretation of the depth information, this information can be used to improve the abilities of another system by integrating the stereovision system to these systems like robots. Thus the any robotic system can have the perception of depth.

REFERENCES

- Bebis, D. G. (2004). *Dr. George Bebis*. Retrieved April 21, 2014, from University of Nevada:<http://www.cse.unr.edu/~bebis/CS791E/Notes/EpipolarGeonetry.pdf>
- Benbow, T. J. (1989). *Oxford English Dictionary (Second Edition ed.)*. Oxford University Press.
- Camellini, G., Felisa, M., Medici, P., Zani, P., Gregoretti, F., Passerone, C., et al. (2014). 3DV — An embedded, dense stereovision-based depth mapping system. *Intelligent Vehicles Symposium Proceedings* (1435-1440). Michigan, USA: IEEE.
- Cigla, C., & Alatan, A. (2008). Depth assisted object segmentation in multi-view video. *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video* (185-188). İstanbul: IEEE.
- Collins, R. (2007). *CSE/EE486 Computer Vision I*. Retrieved May 5, 2014, from <http://www.cse.psu.edu/~rcollins/CSE486/lecture06.pdf>
- Dodgson, N. A., Woods, A. J., Merritt, J. O., Benton, S. A. & Bolas, M. T. (2004). Variation and extrema of human interpupillary distance. *Proceedings of SPIE: Stereoscopic Displays and Virtual Reality Systems XI*. San Jose, California.
- Davepape. (2006). *Stereoscope*. Retrieved June 6, 2014, from Wikipedia: http://en.wikipedia.org/wiki/Stereoscope#mediaviewer/File:Holmes_stereoscope.jpg
- Digilent. (2011). *VmodCAM™ Reference Manual*. Retrieved December 20, 2013 Digilent:http://www.digilentinc.com/Data/Products/VMODCAM/VmodCAM_rm.pdf
- Digilent. (2013). *Genesys™ Board Reference Manual*. Retrieved December 20, 2013, from; Digilent:http://www.digilentinc.com/Data/Products/GENESYS/Genesys_RM_VC.pdf

- eLADwiki. (2010). Retrieved May 20, 2014, from eLADwiki: http://elad.su-per-b.org/index.php?title=File:Human_visual_system.jpg
- Evan-Amos (2011). *Kinect*. Retrieved June 6, 2014, from Wikipedia: <http://en.wikipedia.org/wiki/Kinect#mediaviewer/File:Xbox-360-Kinect-Standalone.png>
- Harris, C., & Stephens, M. (1988). A combined corner and edge detector. *The British Machine Vision Conference (BMVC)* (147-152). Manchester: BMVA.
- Holzmann, C., & Hochgatterer, M. (2012). Measuring distance with mobile phones using single-camera stereo vision. *Distributed Computing Systems Workshops (ICDCSW)* (88-93). Macau: IEEE.
- InTech. (2012). *Current advancements in stereo vision*. (A. Bhatti, Ed.) Retrieved from <http://www.intechopen.com/books/current-advancements-in-stereo-vision>
- Kamencay, P., Breznan, M., Jarina, R., Lukac, P., & Zachariasova, M. (2012). Improved depth map estimation from stereo images based on hybrid method. *Radioengineering*, 70-78.
- Kinect. (2014). *Kinect*. Retrieved June 6, 2014, from Wikipedia: <http://en.wikipedia.org/wiki/Kinect>
- Mattoccia, S. (2013). *Stereo Vision: Algorithms and Applications*. Retrieved January 12, 2014, from: <http://vision.deis.unibo.it/~smatt/Seminars/StereoVision.pdf>
- Mutto, C., Zanuttigh, P., & Cortelazzo, G. (2010). Scene segmentation by color and depth information and its applications. *Streaming Day*. Udine: University of Udine.
- Netting, R. (2011). *Spacepalce*. Retrieved June 6, 2014, from NASA: <http://spaceplace.nasa.gov/stereo-vision/en/>

- Okutomi, M., Canon Inc., K. J., & Kanade, T. (1991). A multiple-baseline stereo. *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91* (63 - 69). Maui, Hawaii: IEEE.
- Rahmat, R., Al-Tairi, Z., Saripan, M., & Sulaiman, P. (2012). Removing shadow for hand segmentation based on background subtraction. *Advanced Computer Science Applications and Technologies (ACSAT)* (481 - 485). Kuala Lumpur: IEEE.
- The Turing Institute. (1996). *The history of stereo photography*. Retrieved May 25, 2014, from http://www.arts.rpi.edu/~ruiz/stereo_history/text/historystereog.html
- Xilinx. (2009). *Virtex-5 libraries guide for hdl design*. Retrieved January 20, 2013, from;Xilinx:http://www.xilinx.com/support/documentation/sw_manuals/xilinx11/virtex5_hdl.pdf

APPENDICES

Appendix A

Matlab Codes of the Proposed Rectification Algorithm

(Mentioned in Section 3.1.2)

% rectification function with tangents of the feature points of stereo images
function [r,c,x]=common_features(Size_PI,PIP11,PIP2,I1,I2) %PIP11 and PIP2
feature points of the images and Size_PI is size of the feature list; r,c, are the
coordinates of the rectified features, x is the tangent values

PIP1x=zeros(Size_PI,Size_PI);%x coor. of first image's features
PIP1y=PIP1x;%y coor. of first image's features
phiPIP1=PIP1y;%tangents of first image's features

PIP2x=PIP1x;%x coor. of second image's features
PIP2y=PIP1x;%y coor. of second image's features
phiPIP2=PIP1x;%tangents of second image's features

for c=1:Size_PI
for t = 1:Size_PI

PIP1x(c,t)=PIP11(c,2)-PIP11(t,2);
PIP1y(c,t)=PIP11(c,1)-PIP11(t,1);
phiPIP1(c,t)=PIP1y(c,t)/PIP1x(c,t); % calculation of possible lines slopes in
first image

PIP2x(c,t)=PIP2(c,2)-PIP2(t,2);
PIP2y(c,t)=PIP2(c,1)-PIP2(t,1);
phiPIP2(c,t)=PIP2y(c,t)/PIP2x(c,t);% calculation of possible lines slopes in
second image

end
end

x=(phiPIP1)-(phiPIP2); % comparison of tangents (slopes)
[r,c]=min(abs(x),[],1);

ISE RTL Schematic Demonstration of the FPGA Design (Mentioned in Chapter Four)

