MEE10:21

Depth Measurement Improvement Using Dithering Method in Sensor-shifted Stereo Cameras

Wail Mustafa

This thesis is presented as part of Degree of
Master of Science in Electrical Engineering

Blekinge Institute of Technology

February 2010

Blekinge Institute of Technology
School of Engineering
Department of Applied Signal Processing
Supervisor: Lic. Jiandan Chen and Prof. Wlodek Kulesza
Examiner: Prof. Wlodek Kulesza

** This page is left blank **

Abstract

The camera stereo system provides the mechanism for obtaining the 3D information of a target point. However, the depth estimation, in particular, suffers from the uncertainty caused by the quantization in the digital images. The dithering approach is a way to reduce the uncertainty of the depth reconstruction by applying a controlled readjustment for the stereo parameters, and estimating the depth from the images acquired before and after the readjustment. The skewed-parallel camera was proposed to be deployed as stereo of camera. The enhancement of the depth reconstruction uncertainty for this kind of stereo by the dithering approach was verified by simulation. This thesis verifies that enhancement by physical experiment. For that purpose, a prototype for the skewed-parallel camera is configured as stereo and used to measure the differential depths of some test targets. For the verification, the average error of the depth reconstruction using dithering approach is compared with the average error by the direct method. The research proves the improvement in the uncertainly of the depth reconstruction for the targets chosen for the experiment.

Keywords:

Depth Reconstruction, Dithering, Skewed Parallel, Stereo Setup, Iso-disparity surfaces

** This page is left blank **

Acknowledgements

I would like to thank sincerely my supervisor Prof. Wlodek Kulesza for his support and for his efforts in revising and guiding my thesis works.

I would like also to thank Lic. Jiandan Chen for his direct supervision, help and guidance.

Also, Dr. Samtak Khatibi has helped us with his advice and valuable comments, and I would like to acknowledge that with my appreciation.

Finally, I would like thank Dr. Fredrik Bergholm for his great contribution to this thesis.

Dedication

To my family

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

The advanced technologies may help the human extend the ability to process the visual information. This will raise the demand for autonomous systems with high performance sensors. This thesis is concerned in stereo camera and its applications in human activity monitoring. The Intelligent Vision Agent System, IVAS, is a vision and information processing system for this kind of applications [1], [2]. The IVAS gathers data in order to reconstruct 3D information in which it can be used for health care, security and surveillance applications. The system focuses on an interested part of the scene by a dynamic control of the stereo pair. Such system requires high accuracy for the reconstruction of the 3D information in order to guarantee high performance.

The human activity field is a 3D world, where the location of each point is represented by x,y,z coordinates. Therefore, it is highly demanded to be able to get all the three coordinates for the interested point in the field. However, camera can only capture a two-dimensional image where each point is represented by x and y coordinates. The stereo system provides the mechanism for acquiring the z coordinate. In this case, the z coordinate is referred to as the depth, and the process of acquiring the depth from the stereo system is called the depth reconstruction.

The stereo pair captures two images for the 3D world. The interested point needs to have a projection in each image, and this only happens when the point lies in the common field of view (FoV) of the camera pair. To be able to reconstruct the 3D information, the system needs first to find the corresponding points in the two images for the same view. This process is referred to as the matching process.

Figure 1.1 shows a point P in the common FoV of a camera pair, and its projections in the two image planes. The point position in 3D is represented as $P(x,y,z)$ where its projection in the left and right image planes is represented as $P_l(x_l,y_l)$ and $P_r(x_r,y_r)$ respectively. The objective of the 3D reconstruction process is to obtain the x, y and z values when P_l and P_r are available. The values of x and y can be easily obtained from either P_l or P_r knowing the configuration parameters of the corresponding camera. A method is needed to reconstruct the z value from the two projections P_l or P_r and the geometry of the stereo.

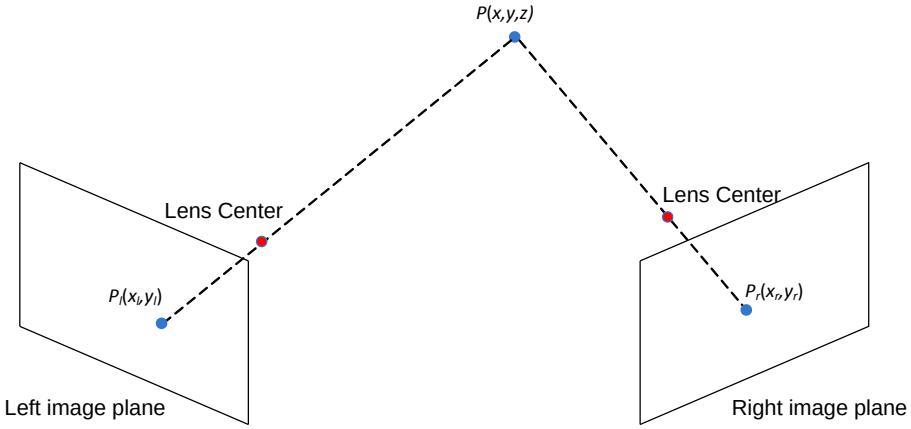


Figure 1.1: The point in space and its projections in the left and right image planes

However, digital camera quantizes the image plane into array of pixels and forms the digital image. Due to that, the projection points are approximated by the centers of the pixels on the left and right images. The difference between the exact and the discretised projections may cause error in constructing the depth and that is referred to as the depth reconstruction uncertainty.

The depth reconstruction uncertainty is a direct impact of the quantization process, and so it is related to the pixel size of the camera sensor. By decreasing the pixel size, the uncertainty can be reduced but that, on the other hand, will compromise the quality of the image as it reduces the signal-to-noise ratio.

Regarding the image resolution, the limitation of the pixel size is overcome by combing the information from slightly different low-resolution images of the same scene into an image with higher resolution. This way of enhancing the image resolution is called the super-resolution reconstruction. In a similar way, two pairs of images taken by a stereo system with two slightly different setups can be used in combination to reconstruct the depth with an enhanced uncertainty. The two different setups are resulted from readjustment of one of the stereo parameters. This method is referred to as the dithering approach [3]. Both the super-resolution reconstruction and the dithering approach work to compensate the discretization in the digital imaging.

The thesis aims to verify, by physical experiment, the improvement of the depth reconstruction through the dithering approach for a specific setup of the camera stereo. This setup deploys a novel kind of camera where the sensor can be shifted in a controlled manner. This kind of camera is called the skewed-parallel camera [4].

It is preferred to use the skewed-parallel camera setup for the 3D reconstruction instead of the conventional setups. Comparing to the convergence camera stereo setup, the skewed-parallel camera setup requires simpler processing. In addition to that, the skewed-parallel camera setup offers a wider common field of view than in the parallel camera stereo setup.

The readjustment of skewed-parallel camera stereo setup, that is required when applying the dithering approach, is done through shifting the sensors of the pair cameras. And because the readjustment requires shifting the sensor a fraction of the pixel size, the camera needs to have a micro-movement capability for the sensor in order to satisfy that.

Because the verification in this thesis is based on physical experiment, and due to the unavailability of a product for this kind of camera, a prototype for the camera is used for the purpose of this thesis.

This thesis consists of 7 chapters; chapter 1, an introduction, includes basic definitions to help the reader understand the topic. Chapter 2 presents the state of art while chapter 3 introduces the problem statement, research questions, the hypotheses and the main contributions of this thesis. Chapter 4 illustrates the iso-disparity map of the skewed-parallel camera stereo setup, its mathematical model and the implementation of the dithering approach. In chapter 5, a detailed description of the prototype of the camera is presented. The validation model and the validation experiment along with the result are shown in chapter 6. Finally, a conclusion with a recommendation for future work is introduced in chapter 7.

1.1 Definitions

1.1.1 CCD cameras

CCD stands for Charge-Coupled Device which is an image sensor technology that is used in digital cameras. Therefore, the camera that has a CCD sensor is called CCD Camera.

The CCD chip (sensor) is made of an array of microtransducers, and it replaces the film in the conventional camera. The chip has a range of size dimensions from 3 to 10 millimeters, and is referred to with a fraction that represents the size. For example, ½'' CCD ship has a size of 6.4 mm (H) x 4.8 mm (V). Each element in the sensor array is called *pixel*, and it ranges from 5 µm to 20 µm [5].

1.1.2 Primary axis and optical axis

The optical axis is the line that goes through the center of the lens and is perpendicular to the image plane, where the primary axis is the line that goes through the center of the lens and the centre of the image plane. They are identical when the middle of the image plane meets the optical axis (no shift in the image plane). Figure 1.2 shows this concept.

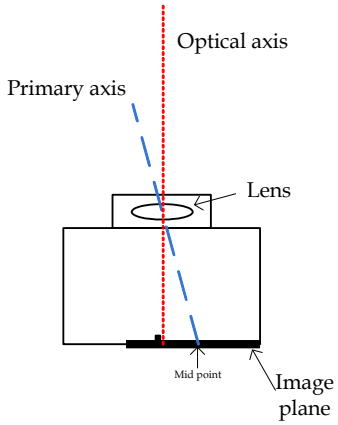


Figure 1.2: The concept of optical axis (dotted red line) and primary axis (dashed blue line)

1.1.3 Stereo configuration

Stereo configuration refers to a setup of two cameras in the 3D world with specific parameters. Figure 1.2 shows the general stereo setup in X-Z plane.

The distance between the centers of the lenses of the pair is referred to as the baseline B, and the angle between the optical axis of the camera and Z-axis is called the convergence angle, a. The focal length, f, is the distance between the lens and the image plane.

Also, the length of the sensor, m, and the pixel length, ΔD, have an importance in the configuration and the analysis of the stereo system.

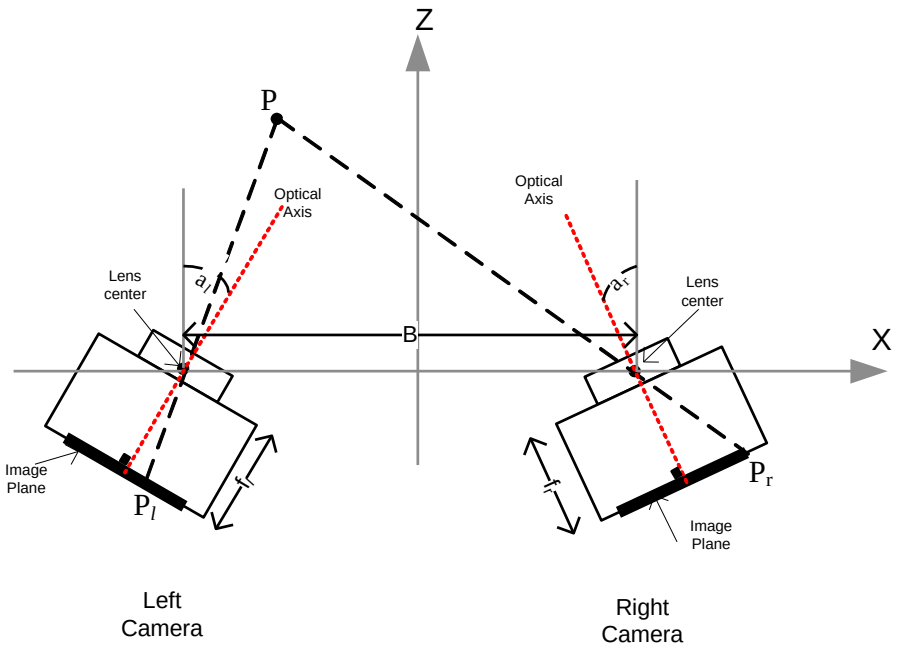


Figure 1.3: X-Z view for the general stereo setup configuration.

1.1.4 General stereo setup, parallel stereo setup and skewed-parallel stereo setup

The stereo setup for the stereo camera in Figure 1.3 is called the general stereo setup, where the convergence angle for each camera is not equal to zero, and the both the primary axis and the optical axis are identical.

When the convergence angles are equal to zero, the setup is referred to as the parallel stereo setup, Figure 1.4 (a).

Figure 1.4 (b) shows the skewed-parallel stereo setup where the primary axis and the optical axis are not identical. The angle between the two axes represents the convergence angle for the skewed-parallel stereo setup. The kind of special camera that is used in this stereo setup is called the skewed-parallel camera.

1.1.5 Fixation point

The fixation point is where the primary axes of the stereo pair converge to. The general stereo setup and the skewed-parallel setup may have a fixation point while the parallel stereo setup does not have it.

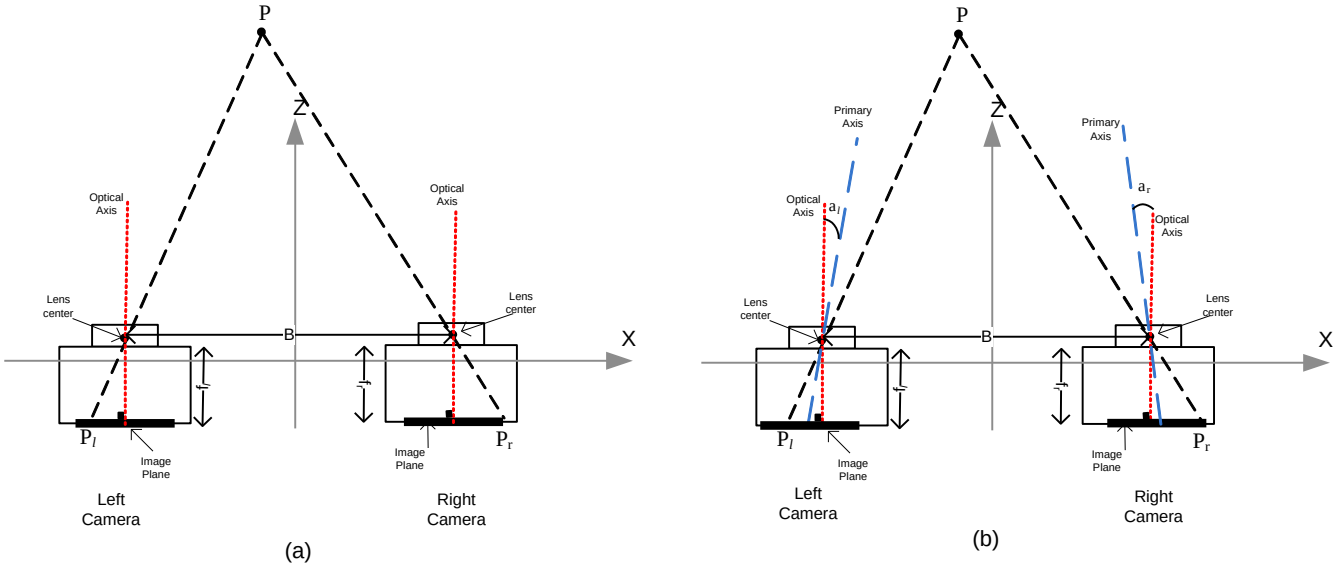


Figure 1.4: X-Z view for the parallel stereo setup (a), and the skewed-parallel stereo setup (b)

1.1.6 Disparity

The displacement of the corresponding projections, of a certain point in space, on the left and the right images is called the disparity of this point. This displacement differs from point to point, and it relates the depth of the point [6].

Due to discretization in the sensors, the disparity is usually measured in pixels and can be positive or negative. Figure 1.5 shows the disparity concept for two images taken for the same scene [7].

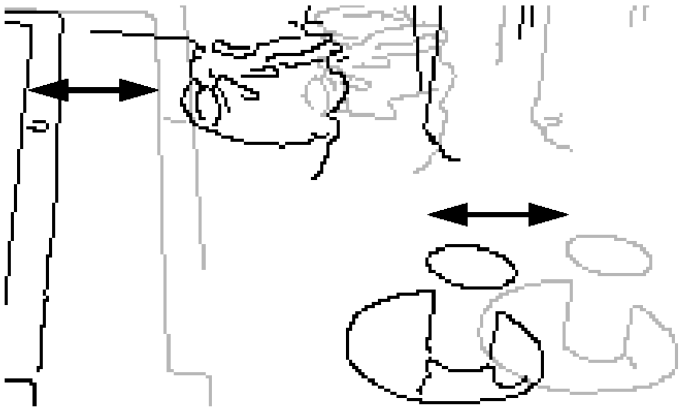


Figure 1.5: The arrows show the disparities for two different points in the same scene

1.1.7 Iso-disparity & iso-disparity surfaces

The points at the space that have the same disparities are referred to as iso-disparity. Due to discretization of the disparities, the iso-disparity surfaces are also discrete levels. Points at the same iso-disparity surface have the same depth [1].

1.1.8 Linear stage

A linear stage (translation stage) is a motion component that is used to control the movement of any object in a line with certain accuracy. The positioning of the object is typically controlled by an actuator. The kind of actuator differs and it may be manual or motorized.

Chapter 2: The State of Art

The reconstruction of the 3D information from the stereo systems has been an interested research topic in the recent years. Studying the iso-disparity surfaces in the different stereo configurations is very important for the reconstruction. M. Pollefeys and S. Sinha derived the shape of iso-disparity surfaces for general stereo cameras configuration [8]. They showed the intersection of the surfaces with epipolar planes, and proved that it consists of a family of conics. They discussed the rectification of the image and its relevance to the iso-disparity surfaces.

For planning of a multiple sensor system in indoor environments, an analysis for the effect of adjusting the variables of the stereo system on the depth reconstruction uncertainty was done by [9]. In this paper, a mathematical geometry model was presented and used to for the analysis through the aspects of the iso-disparity surfaces. The paper showed that the depth reconstruction uncertainty depends of the length of the baseline of the stereo cameras, the physical properties of the sensor and the zooming. In addition to that, the baseline length, the focal length and the distance from the baseline have more effect on the uncertainty than the convergence angle of the stereo cameras.

The accuracy of the depth reconstruction is subject to the spatial quantization which is hard to control due to the restrictions on the pixel size. That issue was discussed in [10], they studied the tradeoff of selecting the pixel size. The large size is desirable because it leads to a high signal-to-noise ratio SNR. In the other hand, the small size of the pixel is preferable since it results in a high spatial resolution which reflects in a high accuracy of reconstruction. To determine the optimal pixel size, they used a camera simulator and some quality metrics for the image. They implemented that for different processes of CMOS technology for the sensor. It was shown that the optimal size differs with the technology. This can be interpreted as a pixel size restriction for a specific technology and hence the pixel size can not be considered a factor to be change.

Improving the depth reconstruction has been studied using different approaches for the different stereo setups. In [11], Theimer and Volpel proposed a method to reduce the localization uncertainty in the stereo algorithms. The method enhances the accuracy of

obtaining the shift between the two images and thus results in reducing the uncertainty. This method is based on the idea of using Gabor filter (that is used in image processing as an edge detector) to filter the left and right images. That filtering produces a phase shift between the images which is representing the information of the spatial shift. Their way of computing the localization uncertainty is based on a perspective projection model where that uncertainty is represented as a curve segment. This segment is introduced by the intersection of the x and y iso-disparity surfaces.

An approach that is based on the micro-movement of the vergence angle of the stereo was proposed in [12].This approach relates the elements that their value of disparity is zero for a continuous sequence of vergence angles. This approach does not use the disparities to calculate the depth. Instead, it computes the depth only from the relative position of the pixel in the sensor and the vergence angle, and that provides better estimation for the depth. The approach has a support from the physiology of the human eye as the studies showed that it performs the same kind of movement.

H. Kim et al. proposed a method that controls the movement of the vergence into a stable state to improve the reconstruction uncertainty [13]. This stable state is occurred when the flux of the disparity flow is equal to zero. The disparity flow represents the change of the disparity at a given spatial location. This method introduced the idea of controlling the movement when the reconstruction uncertainty is required to be improved.

H. Sahabi and A. Basu did an uncertainty analysis for the depth reconstruction in [5]. They investigated the effect of the vergence with a uniform and with a non-uniform resolution of pixels of the sensor. The non-uniform assembles the one of humans where there is a high resolution in the center and nonlinearly decreasing resolution to the edge. The vergence has a more effect in the non-uniform resolution compared to the uniformed one. In the non-uniform resolution, we can find a vergence angle where the error is minimized while for the uniform resolution there is no such an optimal value. They also used the disparity map to analyze the different cases.

The dithering is a technique that is used in signal and image processing for different applications. One of its applications is presented in [14] where they used the dithering technique as a quantization method. In this application, the dithering was used to reduce the number of the colors in the true color image and still maintain a good quality. This reduction is important be able to deal with the image in a limited-capacity display or printing device. The technique injects an external noise to the quantization process in order to break the induced correlation in the normal quantization methods.

Adopting the dithering technique in the problem of reducing the uncertainty of the depth reconstruction was done by Chen et al. [3]. In their paper, they proposed an algorithm that estimates the depth from a few images taken from a stereo of cameras. These images are resulting from a little amount of movement in the stereo setup. The movement is controlled by a dither signal estimated from the analysis of iso-disparity planes. The algorithm is derived for the parallel stereo pair model. Also, an extension for their work was done for the skewed-parallel stereo pair model in [15].

Francisco and Bergholm introduced a new stereo model which was called skewed-parallel camera in [4]. They proposed shifting the sensor ship of the camera pair instead of changing the vergence angle. In their paper, they also discussed the benefits of using this sensor-shifted camera comparing to the vergence movement in the general stereo setup through an intensive study. To experimentally validate their model, they built a prototype for it that has a horizontally moving camera module that is controlled by a step motor.

In order to minimize the motion blur in reconstructing a high-resolution image from a set of low-resolution images, Ben-Ezra et al. proposed the jitter camera for the super-resolution video reconstruction [16]. The sensor chip of this camera is allowed to shift horizontally and vertically in a subpixel movement. They also built a prototype for the proposed camera and applied a half pixel shift for the sensor as the optimal movement for their application.

Chapter 3: Problem Statement, Research Questions, Hypothesis, Main Contributions

It is required to validate that the dithering algorithm improves the depth reconstruction for the skewed parallel camera stereo. And although that was done by simulation, it is still demanded to verify that by physical experiment. Because that kind of camera with a micro-movement capability for the sensor is not available, a prototype needs to be configured and used for this experiment. In addition to that, it is necessary to identify and obtain the parameters of the prototype that are needed to implement the experiment.

Upon above the research questions are:

1) How can a prototype for a skewed-parallel camera be built up and how can it be configured as a stereo pair to implement the experiment?
2) How to experimentally validate whether the dithering approach improves the uncertainty in the depth reconstruction for this kind of stereo?

And the hypotheses of this research are:

a) It is possible to build a prototype for the skewed-parallel camera by separating the camera body from the lens and control the required shift using a micro-movement motor. Also, this prototype can be configured as a stereo pair by using a translation stage.
b) The improvement in the depth reconstruction can be validated by configuring the prototype of the skewed-parallel camera as a stereo. Then, by comparing the average error of measuring the differential depths of some testing targets by the dithering approach, with the average error by the direct method for the same targets.

The main contributions of this thesis can be summarized as:

- Configuring a prototype for the skewed-parallel camera from an old one after testing, and replacing the components on demand. Also, the configuration parameters that are required for the experiment are estimated.
- Implementing the validation experiment.
- Analysis of the experimental results and verification of the depth reconstruction improvement by applying dithering approach.

Chapter 4: Iso-disparity Map, Mathematical Model and Dithering Approach for Skewed-Parallel Camera Stereo Setup

In many cases, the camera has to be rotated to capture a certain view that is initially out of the field of view of the camera. This rotation causes a distortion of the shapes in the captured images. Instead of rotating the whole camera, professional photographers use a technique that applies a little shift between the camera lens and the sensor. This shift provides a similar effect to the rotation in the way that it can capture the wanted view, while it avoids the distortion on the other hand.

Figure 4.1 illustrates the effects of the two methods on capturing a view [17]. Figures 4.1 (a) shows the original field of view of the camera. In this case the captured image does not contain the whole building while in figure 4.1 (b) the rotation enabled the camera to capture it. In figure 4.1 (c), the shifting enabled the camera to capture the whole build without the shape distortion seen in figure 4.1 (c).

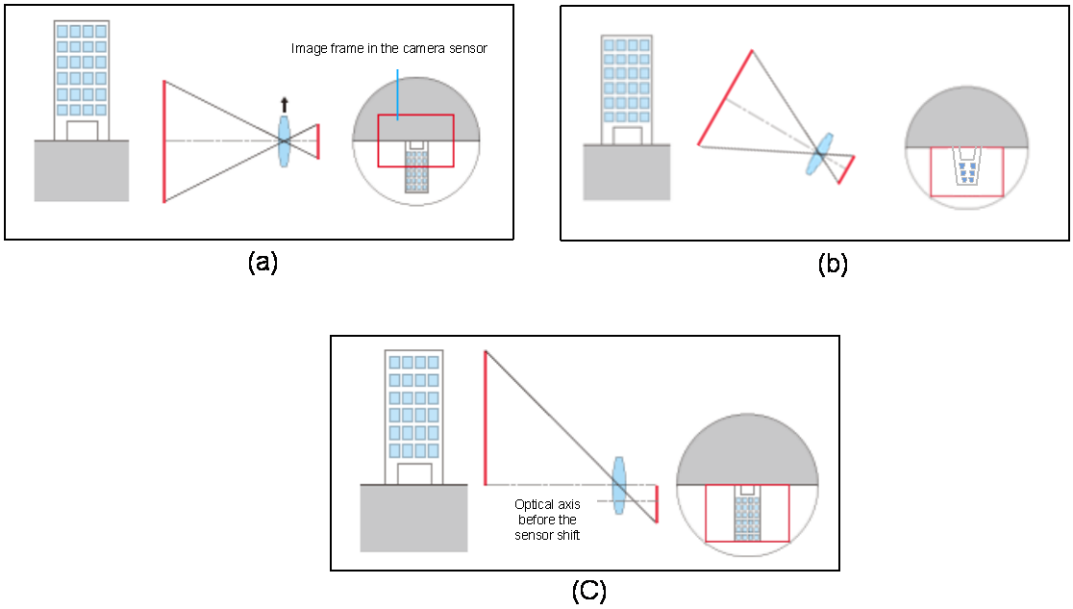


Figure 4.1: Two different ways to capture an object that is out of FOV. (a) The normal setup that captures part of the building (b) Rotating the camera (c) shifting the sensor.

In a similar way, the stereo camera needs to be rotated with a certain convergence angle to get the required common field of view (FoV). This is technique is used in the convergence stereo pair. Instead of rotating, the skewed-parallel camera can be used as a stereo pair as proposed in [15]. In this case, the required common field can be then obtained by applying the appropriate sensor shift. The advantages of using a stereo of skewed parallel camera can be listed as:

- It requires a simple matching process where the epipolar lines are parallel and horizontal.
- It can be configured to have a wider FoV compared to a parallel stereo pair with the same baseline distance.
- It has horizontal iso-disparity surfaces and that leads to simple depth reconstruction analysis and formulation.

4.1 Iso-disparity Map

The iso-disparity map is a graphical representation for the iso-disparity surfaces in the two dimensions. These surfaces appear as lines in the map and thus they are called the iso-disparity lines. The iso-disparity map can be used to model the behavior of the iso-disparity surfaces with respect to the change in the stereo model parameters. To derive the iso-disparity map, the geometrical method can be used [15].

Figure 4.2 shows an iso-disparity map for the skewed parallel camera stereo (the red lines) and how the iso-disparity surfaces behave as a response to shift the sensors away [15]. From the figure, it can be seen that the iso-disparity lines are parallel for the skewed-parallel camera stereo. When the object falls exactly in one of these lines, the depth of that object will be then reconstructed with no error. The error is introduced when the object falls between two lines.

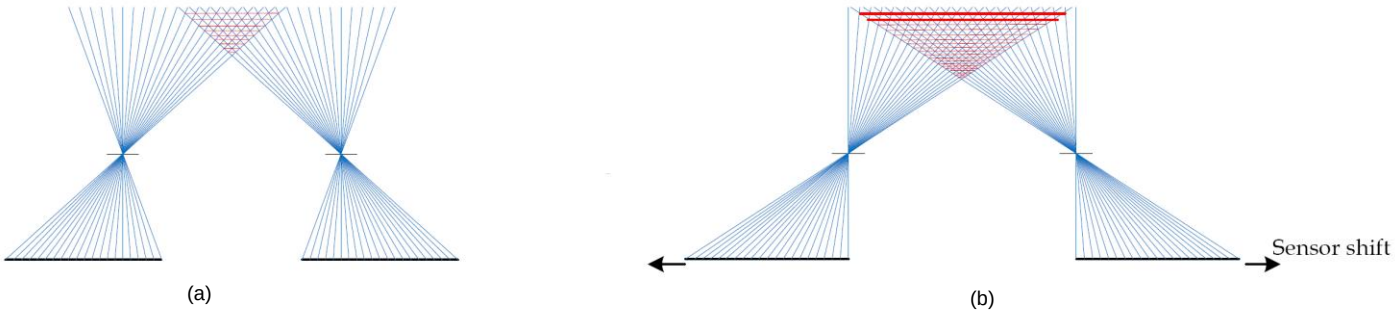


Figure 4.2: The effect of shifting the sensor on the iso-disparity map (a) before the sensor shift (b) after the sensor shift.

The figure also illustrates the effect of shifting the sensors away with respect to the initial positions. It shows that the iso-disparity lines get closer to the baseline. Reversing the direction of shifting the sensors will reverse the effect.

4.2 Mathematical Model

The mathematical model of the skewed-parallel camera stereo was derived and validated by [15]. Figure 4.3 shows the setup of the stereo in X-Z plane that was used to derive the mathematical model.

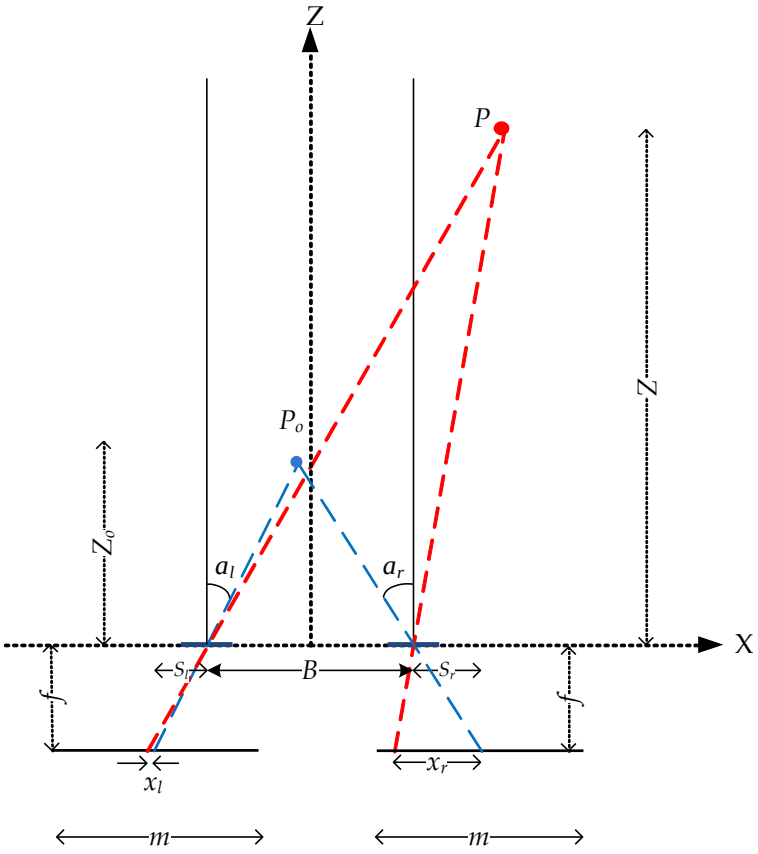


Figure 4.3: X-Z view for the skewed-parallel camera stereo setup

The center of the coordinates is in the middle between the two camera centers (lenses centers), and the sensors lie at the same horizontal line. The focal lengths of the two cameras are assumed in this thesis to have the same and thus they are denoted as *f*. The baseline is the distance between the centers of the two cameras and it is denoted as B. The shift of the sensor S, which is defined as the horizontal distance between the optical center and the sensor center, can be different in each camera. Therefore, it is denoted as Sr and Sl for the right and the left cameras respectively.

The angle between the optical axes and the primary axis is called the convergence angleα for the skewed-parallel camera. The convergence angle represents the effect of shifting the sensors of the skewed-parallel camera stereo. This shift has the same effect as rotating the convergence stereo pair in the way that it introduces the fixation point and widens the common FoV. The convergence angles of the left camera and the right camera are denoted as α_r and α_l respectively.

Using the trigonometric analysis, the convergence angle is derived as [15]:

$$\alpha = \arctan\frac{S}{f} \qquad for \ -m/2 \le S \le m/2 \qquad (4.1)$$

where m is the length of the sensor plane.

Unlike the parallel stereo setup, this skewed parallel setup has the properties of general stereo setup where there exists a zero disparity surface due to the existence of the fixation point (P_o). The depth of this point (Z_0) is found to be [15]:

$$Z_0 = \frac{fB}{(S_r + S_r)} \qquad (4.2)$$

For this stereo setup, the depth Z of any point fallen in the common FoV and with disparity D can be found as:

$$Z = \frac{fB}{(D + S_r + S_r)} \qquad (4.3)$$

Equation (4.3) is reduced to (4.2) when the disparity equal to zero.

When the quantization effect is taken into consideration, equation (4.3) can be re-written as:

$$Z_q(n) = \frac{fB}{(n\Delta D + S_r + S_r)} \qquad (4.4)$$

where Z_q is the quantized depth, n is an integer number representing the disparity, and ΔD is the length of pixel size.

4.3 Dithering Approach

Using the dithering technique, the reducing of the uncertainty of the depth reconstruction was done for the parallel stereo pair in [3]. Also, an implementation of the technique working for the skewed-parallel stereo pair model was done in [15].

For both models, the dithering signal aims at shifting the iso-disparity line to be in the middle of the two consecutive iso-disparity lines n_t and n_{t+1} in which the target is present. Then, the dithering algorithm merges the shifted disparity lines with the initial ones. Hence, the separation between the two initial lines is reduced to the half and so is the reconstruction uncertainty.

The dither signal for the skewed-parallel camera stereo was proved in [15] to be:

$$\Delta S_t = -\frac{(n_t \Delta D + S_l + S_r)\Delta D}{2\big((n_t + 1)\Delta D + S_l + S_r\big) + \Delta D} \qquad (4.5)$$

This signal was also found by simulation to be equal to $0.5\,\Delta D$ regardless the depth of the target point. That can be intereperted as that the maximum enhancement for the depth reconstruction can be obtained by shifting the sensors half the pixel length for the skwed-parallel camera stereo.

4.4 Dithering Algorithm

From the description of the dithering algorithm in [3] , and the calcuation of the dither signal for the skewed-parallel camera stereo in [15], the dithering algorithm can be then be implemented for this kind of stereo. The implementation of the dithering method can be illustrated by Figure 4.4 .

From the Figure 4.4, it can be seen that the dither signals control the position of the sensors of the left and right cameras. The quantized projections of target point are x_{Qli} and x_{Qri} for the left and right image respectively. Then, the depth information can be calculated by averaging the depths of all possible disparities d_i of the stereo pairs. The arithmetic average of all the depths constitutes an unbiased estimate of the target point depths and the depth reconstruction uncertainty is reduced by half for.

The dithering algorithm can be implemented by the four following steps:

1. Initial estimation for the depth from the disparity of the initial pair of images using equation (4.4).

2. Applying $0.5\,\Delta D$ shift in the two stereo cameras.

3. Secondary estimation for the four depths calculations corresponding to four disparity values. These disparities are calculated from the four images which consist of the initial pair and the second pair acquired after the shifting.

   Using the matrix notation, the disparity matrix **d** can be defined as:

$$\mathbf{d} = \left|\mathbf{x}_{Ql}{}^{\mathbf{T}}.\mathbf{x}_{Qr}\right| \qquad (4.6)$$

Where $\mathbf{x}_{\mathrm{Ql}}$ and x_{Qr} are extensions of the vectors $[x_{Ql1} \quad x_{Ql2}]$ and $[x_{Qr1} \quad x_{Qr2}]$ with padding of -1s and 1s, described as:

$$\mathbf{x}_{Ql} = \begin{bmatrix} x_{Ql1} & x_{Ql2} \\ -1 & -1 \end{bmatrix} \; and \; \mathbf{x}_{Ql} = \begin{bmatrix} 1 & 1 \\ x_{Qr1} & x_{Qr2} \end{bmatrix} \qquad (4.7)$$

where indices 1 and 2 correspond to the initial and the secondary pair of images respectively.

For each disparity in d, the corresponding depth is calculated using equation (4.4).

4. Calculation of the depth of the target point by averaging the four depths from step 3.

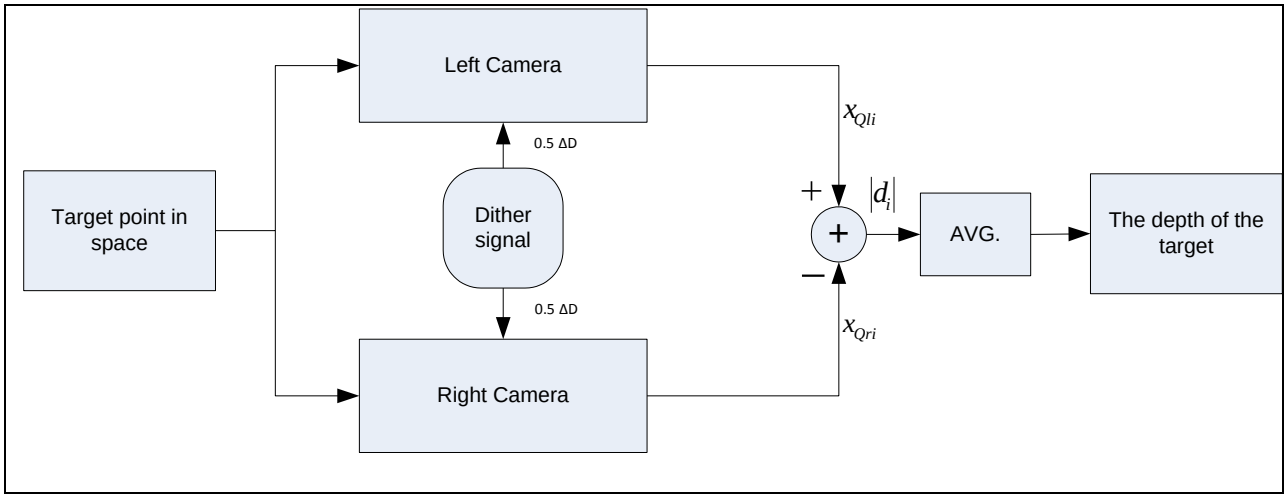


Figure 4.4: Block diagram for the dithering algorithm

Chapter 5: Skewed-parallel Camera (SPC) Prototype

This chapter aims to describe the structure of the SPC prototype that is used in this thesis. All the parts that the prototype consists of are specified with a detailed description. In addition to that, the parameters that are required to be known for the experimental part of this thesis will also be derived and presented.

One of the design considerations is to build a simple camera in the way that its components are affordable to any specialized research laboratory that may need to build such one. The main request is that the sensor of the skewed-parallel camera should have a horizontal movement in regard to the lens. In order to satisfy that, the prototype can be built by separating the camera body, which contains the sensor, from the lens that is normally attached to that body. To control the movement of the camera sensor, the camera body needs to be attached to a micro-movement mechanical device such as an x-positioner or a DC motor.

It is also required to use a stand to hold the lens close to the camera. This stand must be connected to the movement device as well. The design of the stand must also consider the exact separation distance between the sensor and the lens according to the mounting standard of that lens. Figure 5.1 illustrates the basic components of the prototype.

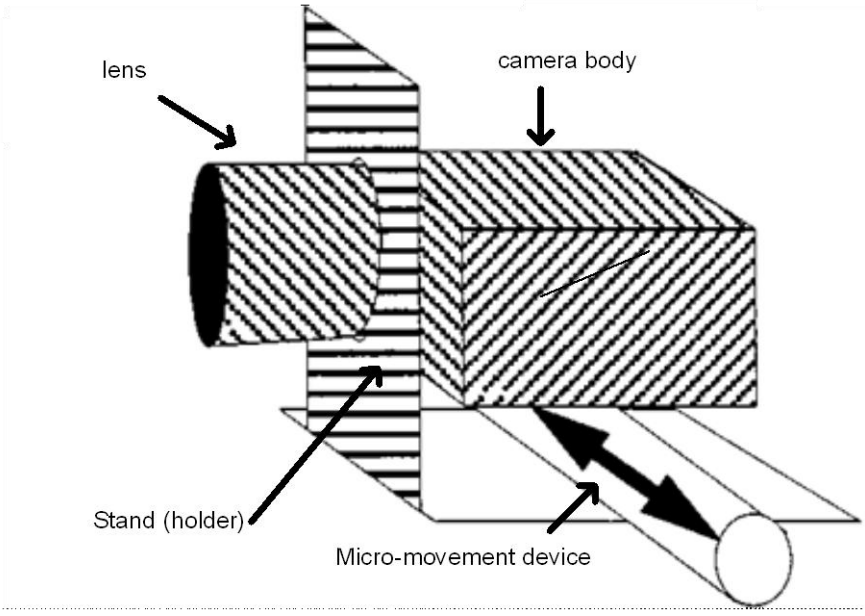


Figure 5.1: The basic components of the skewed parallel camera prototype

There was no commercially available camera that had a shifted sensor with a micro-movement, so it was demanded to design such a camera. Following the above design considerations, a skewed parallel camera prototype has been built to be used for the experimental needs [4].

5.1 Components of the Prototype

There are four main components in the camera prototype that was built for the skewed-parallel camera (SPC) model:

1. A camera module connected through a cable to a frame grabber card that is installed in a computer where the pictures are obtained, processed and saved. This module contains the CCD sensor that represents the shifted sensor in the SPC model.
2. A linear stage controlled by a DC motor connected to a computer where the user can control the motor movement through software. These parts provide the micro-movement capability for the sensor as in the SPC model.
3. A high resolution lens.
4. A metal stand that holds all the above parts.

Figure 5.2 shows front and back views for this camera prototype.

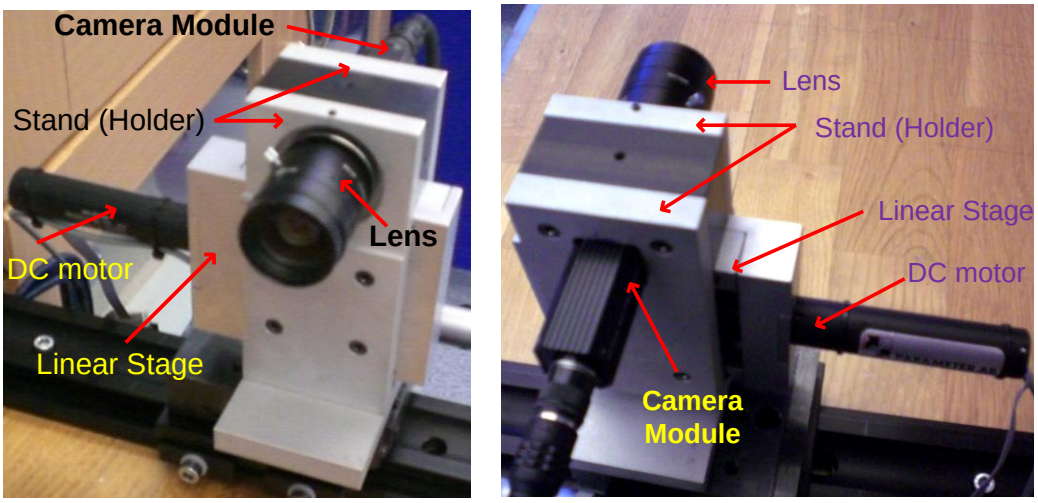


Figure 5.2 The components of the SPC prototype (a) front view (b) back view

5.2 Prototype Specification

5.2.1 The camera module

Sony XC-555P module has been used as a CCD camera. It is a module of a color video camera with a 1/2" type sensor [18]. The module is ultra compact design with dimensions of 22 (H) x 22 (W) x 75 (D) mm. It is made in one piece that suits with these applications at which the space and the weight are restricted. The module uses the NF-mount standard for the lens that can be converted into a C-mount standard. The camera module provides images with a pixel resolution of 768 (H) x 576 (V), and works with PAL (Phase Alternating Line) encoding standard.

The module has a multi 12-pin connector that connects the camera module with the other parts. To do that, a 12-pin cable (CCXC-12P02N) is used to provide the power (12 V), and it is also used to get the video signal from the module.

In its other end, the cable is connected to a junction box (JB-77). This junction box acts like a cabling hub. It is connected to the power supply and hence provides the module with the power through the designated cable pin. It is also connected to the frame grabber card through a coaxial cable with a BNC connector, and provides the card with the video signal from the cable pin of the video signal. Figure 5.3 shows the configuration of the camera module, the cables, the junction box and the frame grabber card.

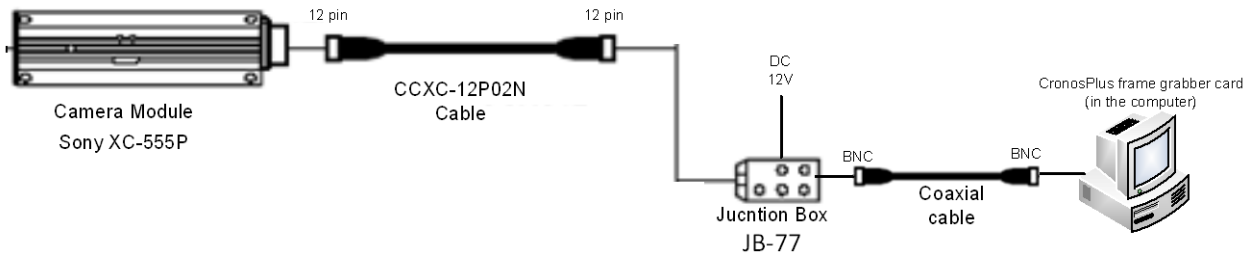


Figure 5.3 The configuration of the camera module, the junction and frame grabber card

Frame grabber is commonly used component in the computer vision system. Its function is to capture digital frames from an analogue video signal. In the camera prototype, Matrox CronosPlus frame grabber card is used which supports analogue video signal [19].

In order to manipulate the video stream in the computer, a Matlab program has been written. This program contains commands from Matlab Image Acquisition Toolbox [20]. The program shows the video stream, takes a snapshot, acquires one frame from the stream, and then returns it as a RGB color image. This image can be used for further processing regarding the experiment. The Matlab program and the camera module manual are in Appendix A and B respectively.

5.2.2 The motor

In order to provide the horizontal shift of the camera module, the prototype uses a linear stage with a DC motor. The linear stage (PI M-125.10) travels a distance up to 25 mm with a resolution of 0.06 μm /count (motor step) [21]. The DC motor (PI C-120.8) that drives the linear stage, is connected through a cable (RS connector) to the controller card (C-832) that is installed in the computer.

It is needed to have software to run the controller and then control the movement of the linear stages. The software is required to obtain the control parameters such as the number of steps, speed and acceleration, from the user, and call the corresponding instruction so that the controller can execute the movement with these parameters. In order to accomplish that, a computer program has been written in C language using TurboC compiler (Appendix C).

The program has been developed to use the code of the controller's driver that was provided with the motor. The main function of the program is to work as a user interface that prompts, from the user, the control parameter and shows the current position after the execution of the movement. The variables that contain the control parameter are passed to the driver along with movement instruction. Then, the program passes a check instruction to the driver to check whether the movement has been executed. Finally, the current position of the linear stage is returned to the user in the screen. Because the controller has 16-bit data bus without compatibility support, the program works only under DOS operating system. The manual of the controller is enclosed in Appendix E.

5.2.3 The lens:

Tamron 23FM25SP lens model has been used in this prototype. The lens is a C-mount type with a focal length of 25 mm. The focus of the lens can be adjusted for objects lie between 0.15 m and ∞ from the front of the lens [22].

Although the camera module is an NF-mount type, the section of the metal, that holds the lens and separates it from the camera module, is designed also to work as an NF to C mount adapter. The distance from the rear of the lens to the sensor of the camera module is different from mounting type to another (12 mm for NF-mount and 17.5 mm for C-mount type) [18]. Therefore, the stand was designed to hold the lens away from the camera front a distance of 5.5 mm to compensate for this difference between the two types. Figure 4.5 shows a top view sketch for the prototype. The figure illustrates the distances between the lens rear and the camera sensor for C-mount.

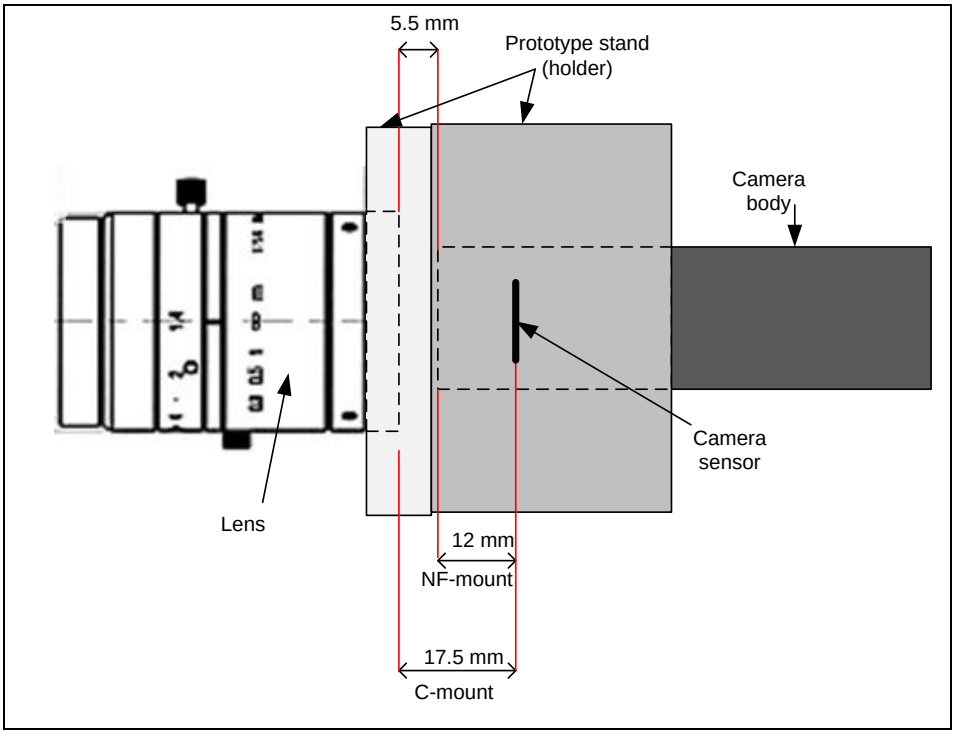


Figure 5.4: Top view for the camera prototype, illustrating how the stand acts as NF to C mount adapter for the lens

5.2.4 Using the prototype as stereo camera

In order to configure this camera prototype as stereo pair cameras, the prototype has to be placed in two different positions. The distance between the two positions represents the baseline B of the stereo cameras model. To be able to achieve that in an accurate way, the prototype has been attached on a translational stage with an accuracy of 1 mm.

5.3 Estimation of the Prototype Parameters

To use this camera prototype as skewed-parallel camera, some parameters are needed to be estimated. For the purpose of thesis, the parameters of the mathematical model that was described in chapter 4.2 are analyzed and specified.

5.3.1 Pixel size

The pixel size is the dimension (length × width) of the camera sensor element. Because this value is not specified in the camera module's manual, it is required to be calculated. The pixel size can be calculated from the size of the sensor and the image resolution as follows:

$$\text{Pixel length} = \frac{\text{The length of the camera sensor}}{\text{The horizontal pixel resoultion of the image}} \qquad (5.1)$$

$$\text{Pixel width} = \frac{\text{The width of the camera sensor}}{\text{The vertical pixel resoultion of the image}} \qquad (5.2)$$

Because these calculations disregard any up-sampling or down-sampling may be applied to the image, this value will be called the effective pixel size.

For the sensors of the 1/2" type, the size of the sensor is 6.4 mm (H) x 4.8 mm (V) [23]. From section 5.21, the image pixel resolution of the camera module is 768 (H) x 576 (V), and using (5.1) and (5.1), the effective pixel length and width will be both found equal 8.33 µm.

5.3.2 Sensor shift

The sensor shift is controlled by the motor, and hence the amount of shift is specified by the number of the motor steps to be prompted to the computer program.

To calculate the number of steps that are required to horizontally shift the sensor a fraction of the pixel length, we need to know the pixel length and the linear stage resolution and then apply the following:

$$\text{Motor steps (counts)} = \left\lfloor \frac{\rho \times \text{pixel length}}{\text{The resolution of linear stage}} \right\rceil \qquad (5.3)$$

where ρ is a fraction of pixel length, and $\lfloor x \rceil$ means rounding to the nearest integer.

This rounding operation may introduce an error in shifting the sensor. For this prototype, the pixel length is 8.33 µm and the resolution of linear stage is 0.06 µm /motor step as shown in sections 5.3.1 and 5.2.1 respectively. To shift the sensor half a pixel length (ρ =0.5), the required number of steps is equal to 69 steps with a shift error of 0.03 µm. That error represents only 0.32 % of the pixel length, and therefore, it can be neglected.

5.3.3 Focal length

The nominal value of the focal length of the lens is 25 mm. The actual value of focal length is equal to its nominal when the focus of the lens is set at infinity, ∞. Setting the focus in any other range less than ∞ will introduce a little change in the actual focal length [24]. Therefore, for this prototype, the lens focus will be kept at ∞.

Chapter 6: Physical Validation

The objective of this thesis is to prove by the physical experiment that applying the dithering algorithm will enhance the depth reconstruction for the skewed-parallel camera model. This chapter introduces a validation model that allows proving this hypothesis in an accurate way. In addition to that, it describes the experiment set up that has been done for validation. Finally, it presents the result of the experiment and its analysis.

6.1 Validation Model

The dithering algorithm is used to enhance the depth perception for the skewed-parallel camera. Trying to validate that with a known depth of a certain target will raise a difficulty, because the depth is measured as the distance between the object and the lens center, it is practically difficult to measure accurately the depth of the test target to validate the measure from the experiment.

In [3], an accurate way to validate the enhancement of the depth reconstruction was implemented. The idea is to measure the line differential depth ΔZ which is the distance between two targets along the optical axis (the axis that is perpendicular to the baseline). This method actually measures a relative depth instead of the absolute depth, and hence provides a more accurate validation because it avoids measuring the exact distance between the test target and the center of the lens since it is difficult to accurately determine the center of the lens. Figure 6.1 illustrates the difference between the two methods.

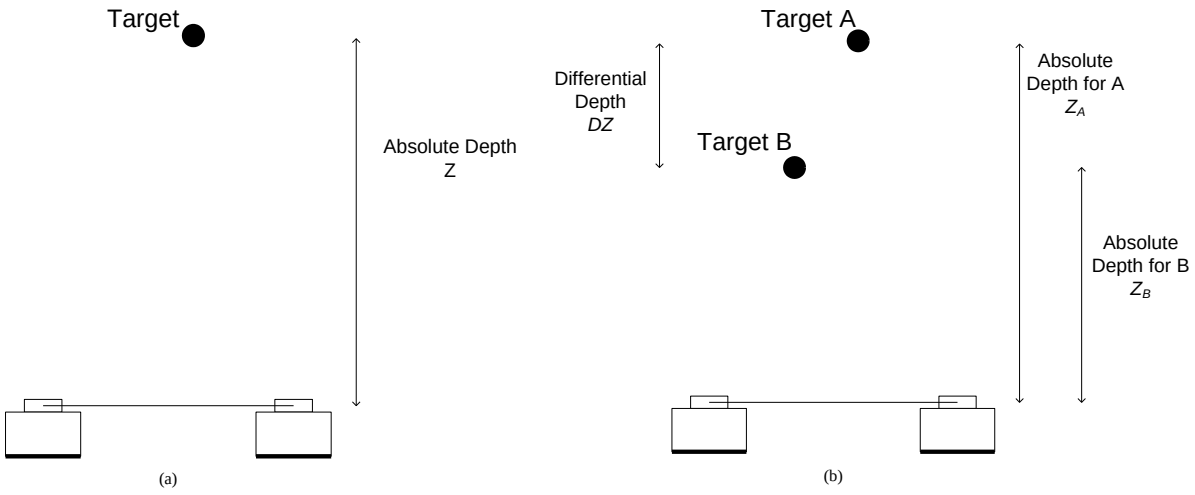


Figure 6.1: The difference between measuring the absolute and the differential depths; (a) absolute depth, (b) differential depth

ΔZ as exact value is defined by the following:

$$\Delta Z = Z_A - Z_B \tag{6.1}$$

where both $Z_A$and Z_B are the absolute depths of target A, and target B respectively.

In order to validate the enhancement of the depth reconstruction, a comparison between measuring the line differential depth by the direct method, ΔZ_α, and the line differential depth by the dithering algorithm, ΔZ_β, can be done. ΔZ_α is calculated by:

$$\Delta Z_\alpha = Z_{A\alpha} - Z_{B\alpha} \tag{6.2}$$

where $Z_{A\alpha}$ and $Z_{B\alpha}$ are the absolute depths measured by the direct method, of target A and B respectively, and they are calculated using equation (4.4). Also, ΔZ_β is calculated by:

$$\Delta Z_\beta = Z_{A\beta} - Z_{B\beta} \tag{6.3}$$

where $Z_{A\beta}$ and $Z_{B\beta}$ are the absolute depths measured by the dithering approach, of target A and B respectively, and they are calculated through the algorithm from section 4.4.

The reconstruction error is the comparison metric between the two methods. Therefore, the reconstruction relative errors for the two methods are found as:

$$\delta_\alpha = |\Delta Z_\alpha - \Delta Z| \tag{6.4}$$

$$\delta_\beta = \left|\Delta Z_\beta - \Delta Z\right| \tag{6.5}$$

where δ_α and δ_β are the reconstruction errors from the direct method and from the dithering algorithm respectively. ΔZ is the exact line differential depth of the two target points, and it must be known from the experiment setup.

Then, the enhancement in the depth reconstruction can be computed by:

$$\text{Reonstuction Enhancement } = \frac{\delta_\alpha - \delta_\beta}{\delta_\alpha} \times 100\ \% \tag{6.6}$$

6.2 Experiment Set up and Implementation

This section presents first the general setup of the experiment in the way that it shows how the prototype was positioned and how the pairs of targets were represented to follow the validation model. Also, the procedure of implementing both the dithering approach and the required calculations of the validation model are presented. Finally, this section shows how the disparity from any pair of images was experimentally calculated.

6.2.1 General setup

Figure 6.2 (a) shows the set up of the physical experiment. The camera prototype described at chapter 5 has been used as stereo cameras using a translational stage. The targets have been represented as points at a grid of lines with a spacing of 5 mm between the lines. The gird has been pasted on a board that can be easily positioned in front of the stereo cameras along the optical axis. The advantages of using the grid can be listed as:

1. It provides an easy and accurate way to represent the target points with a chosen distance between them.
2. It can be used to represent different targets at the same time at one experiment.
3. It is easy to identify the target in the image because it is represented by a black segment on a white background.

From section 6.1, to apply the validation model, it is required to have a pair of target points A and B with a known differential depth. If the grid is parallel to the baseline, A and B will have the same absolute depth and thus a zero differential depth. So, to introduce a differential depth the grid needs to be tilted with respect to the baseline as shown in Figure 6.2 (b) and (c).

Knowing the distance between the two points on the grid L, and the grid tilting angle, α, the exact differential depth can be directly obtained using :

$$\Delta Z = L \times \sin(\alpha) \qquad (6.7)$$

Different angles will lead to different differential depths. And for a specific angle, different distances L will also lead to different depths. The experiment has been done with 2 pairs of points with two different L and three different α.

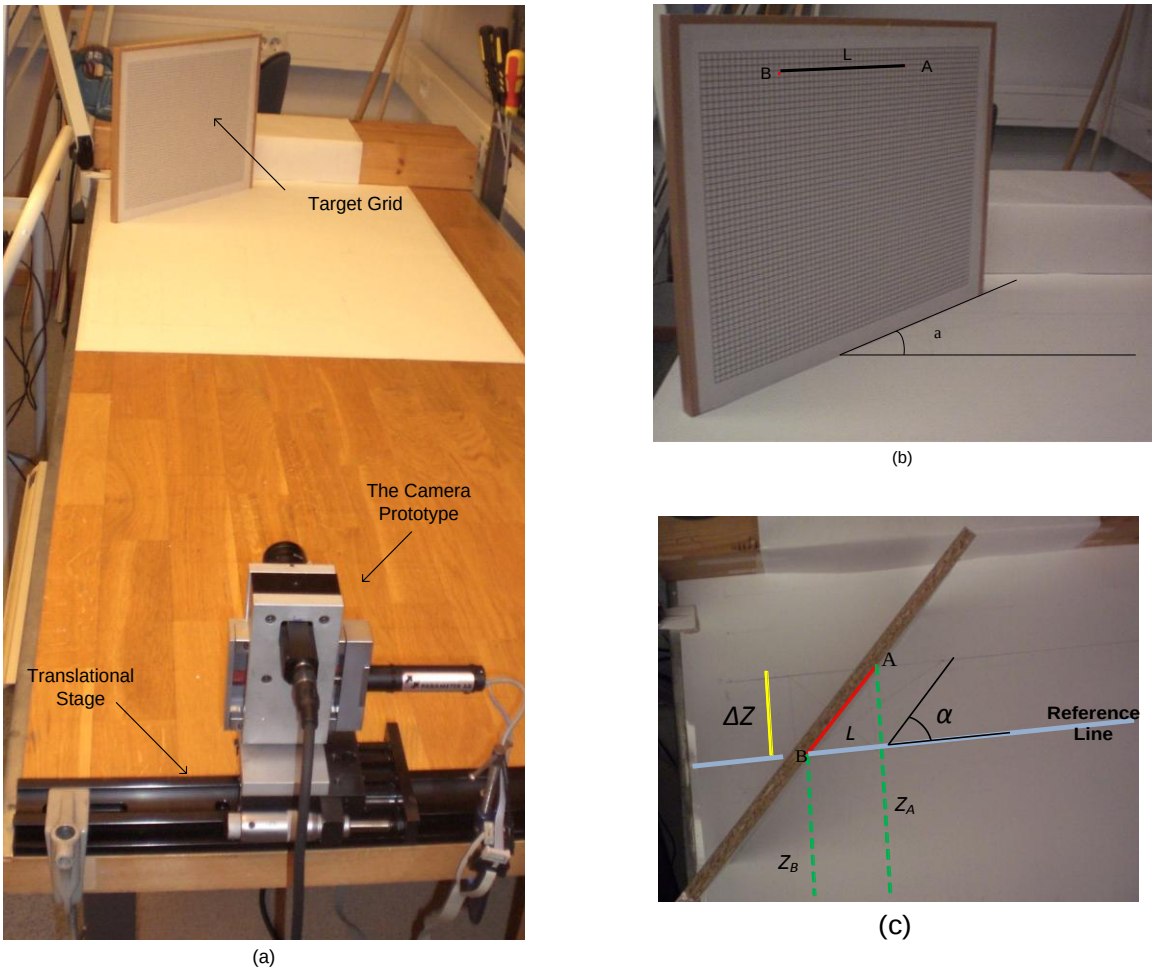


Figure 6.2 : Experiment setup; (a) a general view , (b) a front view for the target grid , (c) a top view for the target grid

6.2.2 Angle calibration

The angle α has to be calibrated to the baseline of the stereo cameras. In order to satisfy that, a line that is parallel to the base has been drawn. This line can be then considered as a reference for tilting angle of grid board as shown in figure 6.2 (c). To draw the line, a distance meter has been used to check accurately the distance of the line from the translational stage in many points. The distance meter that has been used is Fluke 416D, and it is a laser meter with accuracy of ± 1.5 mm [25].

6.2.3 Experiment procedure

In order to obtain the images of targets that are required to apply the dithering algorithm, the Matlab script mentioned in section 5.2.1 has been executed in the computer that is connected to camera module. The following steps have been implemented to obtain the images:

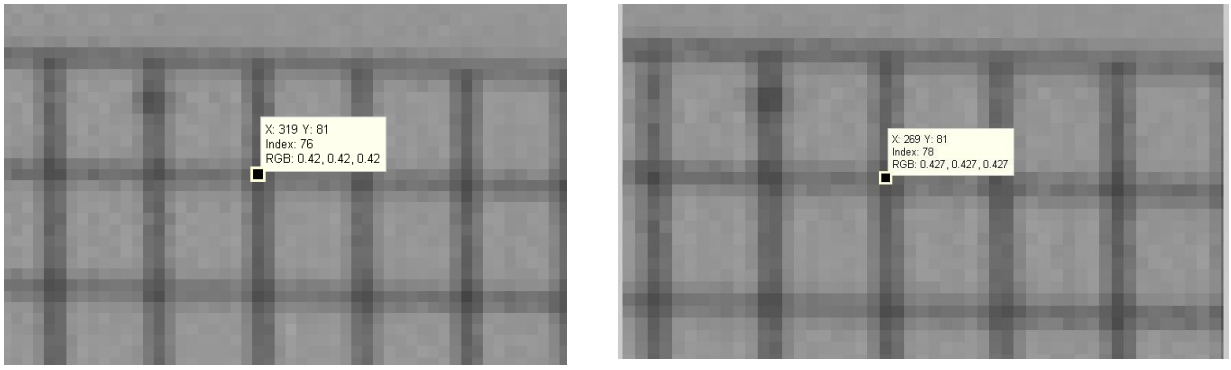
- Set the camera prototype at the position that is specified for the left camera, and then use the script to obtain a snapshot and save it as the initial left image.
- Apply the sensor shift to the left camera (0.5 pixel length) through the motor controlling program described in 5.2.2.
- Use the script to obtain a snapshot and save it as the left image after shifting.
- Set the prototype at the position that is specified for the right camera.
- Use the script to obtain a snapshot and save it as the initial right image.
- Apply the sensor shift to the right camera (0.5 pixel length).
- Use the script to obtain a snapshot and save it as the right image after shifting.

After obtaining the four images, the depth measurement can be followed. First, it is required to measure the differential depths , ΔZ_α and ΔZ_β , for each pair of targets according to equations (6.2) and (6.3) respectively. Then, the reconstruction enhancement is calculated by equation (6.6).

In order to facilitate the calculations of this experiment, a Matlab code has been written. The inputs of this function are the x coordinates values of the pair of the target point at the four images, the distance L and the angle α. The outputs are $\Delta Z,$, ΔZ_α , ΔZ_β, δ_α , δ_β and the reconstruction enhancement. The Matlab code is enclosed in Appendix D.

6.2.4 Disparity calculation

The calculation of the disparity for the target point from any pair of images has been done manually. For each image, the x coordinate value of image pixel that represents the target is read. To be able to do that, the image is zoomed in around the target. Hence, the disparity n can be obtained as the difference between the x coordinate value of the target in the left image and the corresponding value in the right image. These values are shown in figure 6.3. There, the target is the cross of the two lines where the pointer is put on.



(a) (b)

Figure 6.3: The x coordinates value of a target in the two images for the disparity calculation. (a) The left image (b) The right image

Determining which image pixel represents the point is critical for correct depth reconstruction. It can be confusing to select the corresponding image pixel of the target point among two or three horizontally adjacent pixels. Therefore, to be able to do that in an accurate way, the images have been converted from RGB standard to gray-scale. This conversion has been implemented using Matlab ***rgb2gray*** in the image processing toolbox [26]. The gray-scale standard provides one single value of comparison instead of three values. Then, the selection decision can be held based on the intensity values (indices) of the adjacent pixels. The pixel that represents the object must be the darkest one and so the pixel with the least value of intensity should be selected.

6.3 Experiment Results

The experiment has been done for three different value of the tilting angle α: 0°, 26.6° and 45°. For the two pairs of targets, the distance between the two target points on the gird L has been set to 100 mm and 150 mm for the first and the second pair, respectively. The pairs of targets are in the common field of view of the stereo. The grid of targets is set at around 1600 mm from the baseline. The parameters of the experiment are listed in Table 6.1.

Figure 6.4 shows an example of the initial left and right images where α is equal to 26.6°. Each line represents the distance L. The two target points lie at the edges of the line.

Table 6.1: Experiment parameters

Parameter	Value
Pixel Size	8.33 µm
Focal Length	25 mm
Motor Steps	69 steps
Baseline	100 mm

Table 6.2 presents the results of the validation experiment described in sections 6.1 and 6.2 for the pairs of target points. The table shows the results for the three different titling angles of the target grid. For each angle, the results for the two pairs of targets are listed where each pair is represented by the length of the line that separates the two points. The exact differential depth for each pair at each angle is also shown to be compared with reconstructed one. Since the validation experiment compares the dithering method with the direct method, the reconstructed differential depth and the error of the reconstruction are listed for each method.

From Table 6.2, it can be noticed that the reconstruction error using the dithering method is always less than the direct method. This can be interpreted as that the dithering algorithm has enhanced the depth reconstruction uncertainty for all the target pairs in the experiment.

The mean reconstruction errors, calculated from the table, are 5.1 mm and 2.6 mm for the direct method and the dithering algorithm respectively. The improvement in the depth reconstruction accuracy in this experiment is thus 49%. Theoretically, the dithering algorithm reduces the depth reconstruction uncertainty to the half or, in other words, to achieve 50% accuracy improvement. The improvement obtained in this validation experiment is quite close to the expected one.

Table 6.2 : Results of the reconstructed ΔZ for the pairs of targets showing the errors by the direct method and the dithering algorithm.

Angle [degree]	Line length L [mm]	Exact differential depth ΔZ [mm]	The direct reconstruction method		The dithering reconstruction method	
			Reconstructed $\Delta Z\alpha$ [mm]	Reconstruction error δ_α [mm]	Reconstructed $\Delta Z\beta$ [mm]	Reconstruction error δ_β [mm]
0	100.00	0.00	7.39	7.39	3.69	3.69
	150.00	0.00	7.32	7.32	3.65	3.65
26.6	100.00	44.72	47.43	2.70	43.53	1.19
	150.00	67.08	60.99	6.10	63.90	3.18
45	100.00	70.71	68.75	1.96	71.65	0.95
	150.00	106.07	111.16	5.09	108.90	2.83

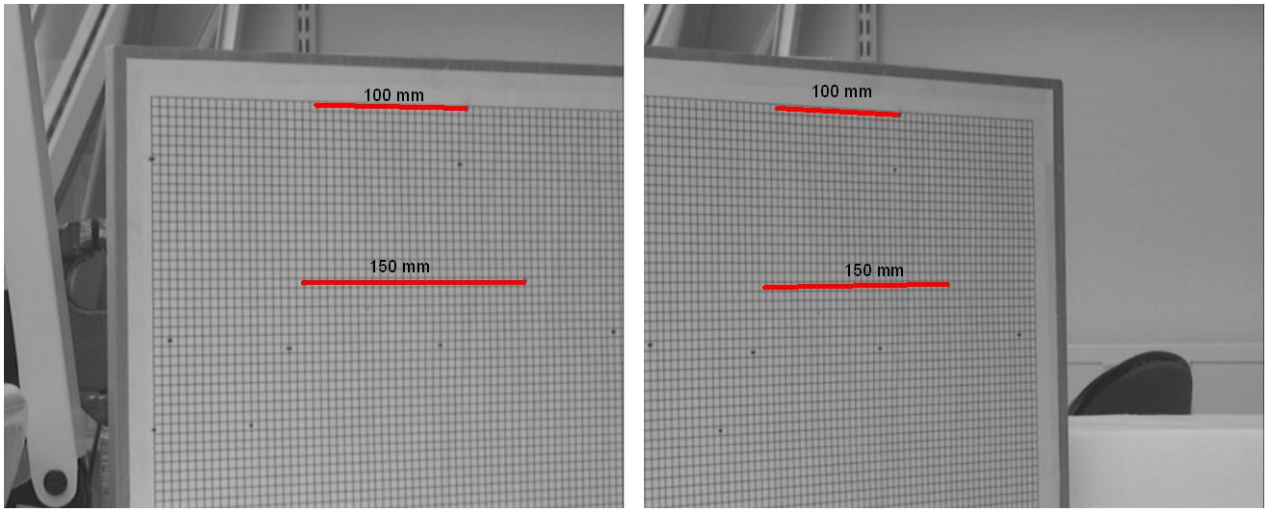


Figure 6.4: The pairs of targets presented in the results in the initial two images. (a) The left image, (b) The right image

Chapter 7: Conclusion

The thesis introduces a physical validation for the improvement in the depth reconstruction uncertainty by the dithering method when the skewed-parallel camera is deployed in the stereo system. The results show reconstruction improvement for all the test targets with an average that is found to be close to the theoretical value. Thus, the experiment proves that the depth accuracy can be improved by the dithering algorithm for this kind of camera. As an outcome of that, a real implementation for this novel kind of camera along with the dithering method would be used in the systems that involve 3D information reconstruction.

In order to conduct the experiment, a prototype for the skewed-parallel camera has been reconfigured from an old one, and setup as a stereo. The prototype is made of a camera module that is separated from its lens while the movement of the module is controlled by a micro-movement step motor. In addition to that, the experiment parameters have been estimated from that prototype.

For the improvement evaluation, the error of the depth reconstruction by the dithering approach is compared to the error from the direct method. For higher experimental accuracy, the differential depth of pair of targets is used in the validation instead of the absolute depth of a single target.

For further research, the dithering approach can be applied with more steps in order to reduce the uncertainty to less than half. The prototype used in this thesis can provide the required shifting accuracy. The prototype can also be using in other research problems that need a micro-movement for the camera sensor.

References

[1] J. Chen, S. Khatibi, and W. Kulesza, “Planning of a Multi Stereo Visual Sensor System for a Human Activities Space,” in: Proc. of 2nd International Conference on Computer Vision Theory and Applications, 2007, pp. 480-485.

[2] W. Kulesza, J. Chen and S. Khatibi, “Arrangement of a Multi Stereo Visual Sensor System for a Human Activities Space,” in: A. Bhatti (Ed.), Stereo Vision, InTech Education and Publishing, Vienna, 2008, pp. 153-172.

[3] J. Chen, S. Khatibi, and W. Kulesza, "Depth reconstrtion uncertainty analysis and improvement – the dithering approach,” Elsevier Journal of Image and Vision Computing, 2008 (submitted).

[4] A. Francisco and F. Bergholm, “On the Importance of Being Asymmetric in Stereopsis-or Why We Should Use Skewed Parallel Cameras”, International Journal of Computer Vision, 29(3) (1998), pp. 181-202.

[5] C. Cecilia, L. Chi-Chung and Y. Mak, “The CCD Camera and Its Applications in Science Teaching”, Journal of Science Education and Technology, vol. 11, issue 2, pp.145-154, 2002

[6] H. Sahabi and A. Basu, “Analysis of Error in Depth Perception with Vergence and Spatially." Computer Vision and Image Understanding, vol. 63, issue 3, pp. 447–461, 1996.

[7] “Introduction to Computer Vision” [Online]. Available: www.cs.nccu.edu.tw/~whliao/acv2008/Stereo.ppt. [Accessed: Nov. 07, 2009]

[8] S. Sinha, and P. Pollefeys, "Iso-disparity Surfaces for General Stereo Configurations." Computer Vision - ECCV 2004, vol. 3023, pp. 509–520, Springer-Verlag, 2004.

[9] J. Chen, S. Khatibi, J. Wirandi and W. Kulesza, "Planning of a Multi Stereo Visual Sensor System for a Human Activities Space - Aspects of Iso-disparity Surface," in: Proc. of SPIE on Optics and Photonics in Security and Defence, vol. 6739, Florence, Italy, September, 2007.

[10] T. Chen, P. Catrysse, A. Gamal and B. Wandell, How Small Should Pixel Size Be? In: Proc. of SPIE on Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications, Vol. 3965, 2000.

[11] B. Volpel and W. M. Theimer, "Localization Uncertainty in Area-Based Stereo Algorithms." IEEE Transactions on Systems, Man, and Cybernetics, vol. 25, issue 12 pp. 1628-1634, 1995.

[12] A. Junior, "The Role of Vergence Micromovements on Depth Perception." Philadelphia, University of Pennsylvania. Technical Report, 1991.

[13] H. Kim, S. Lee and M. Yoo, "Dynamic Vergence Using Disparity Flux." First IEEE International Workshop on Biologically Motivated Computer Vision. vol. 1811, pp. 179-188, 2000.

[14] C. Alasseur, A Constantinides and L. Husson, "Colour quantisation through dithering techniques." IEEE International Conference on Image Processing, 2003. vol. 1, pp. I-469-72, 2003.

[15] A. Siddig, “Depth Reconstruction Uncertainty Improvement for Skewed Parallel Stereo Pair Cameras Using Dithering Approach”, MSc. thesis, Blekinge Institute of Technology, Ronneby, Blekinge, Sweden, 2009. (submitted)

[16] M. Ben-Ezra, A. Zomet, and S. K. Nayar, Video Super-Resolution Using Controlled Subpixel Detector Shifts, IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(6) (2005), pp. 977-987.

[17] “Two new perspective control wide-angle lenses”, Canon Technical Hall [Online] Available: http://www.canon.com/cameramuseum/tech/report/200907/report.html. [Accessed: Sep. 22, 2009]

[18] “CCD Color Video Camera Module,” Sony Corp. ,2002 [Online] Available: http://pro.sony.com/-bbsccms/assets/files/mkt/indauto/Brochures/xc-555_techmanual.pdf [Accessed: Nov 27, 2009]

[19] “Matrox CronosPlus ,Ultra low-cost frame grabber for standard analog monochrome or color video acquisition" Matrox Electronic Systems Ltd. ,2003 [Online] Available: http:// http://www.dip.sk/ PDF/b_cronosplus.pdf [Accessed: Oct 11, 2009]

[20] “Image Acquisition Toolbox User’s Guide,” The MathWorks Inc., 2003 [Online] Available: http://www.mathworks.com/access/helpdesk/help/pdf_doc/imaq/imaq_print.pdf [Accessed: Dec 11, 2009]

[21] “Operating Manual MS 38E C-832 DC Motor Controller,” Physik Instrumente (PI) GmbH & Co., 1996. [Online] Available: http://www.physikinstrumente.net/ftpservice/Motor_Controllers/XXX__Oldcontrollers/C-832.DC-MotorController/C832.OperatingManual/MS38E280.pdf [Access: Oct 03, 2009]

[22] “2/3 25MM F/1.4 with Lock for MegaPixel Cameras” Tamron Inc. [Online] Available: http://www.tamron.com/cctv/prod/23fm25sp.asp [Access: Dec 17, 2009]

[23] “Making Some Sense Out of Sensor Sizes,” Digital Photography Review, 2002 [Online] Available: http://www.dpreview.com/news/0210/02100402sensorsizes.asp [Accessed: Jan 11, 2010]

[24] B. Dimitrov, “Focal Length Experiments,” 2002. [Online] Available: http://kmp-.bdimitrov.de /technology/focalLength [Access: Feb. 04, 2010]

[25] “421D, 416D and 411D Laser Distance Meters,” Fluke Corporation, 2008. [Online] Available: http://us.fluke.com/usen/Products/Fluke+421D+416D+and+411D.htm [Accessed: Mar 08, 2010]

[26] “Image Processing Toolbox 7 User’s Guide,” The MathWorks Inc., 1993 [Online] Available: http://www.mathworks.com/access/helpdesk/help/pdf_doc/images/images_tb.pdf [Accessed: Jan 23, 2010]

Appendices

Appendix A: Matlab script for video stream acquisition

```
vinobj=videoinput('matrox','2','pal');
preview(vinobj);
 im = getsnapshot(vinobj);
figure
 imshow(im)
I = rgb2gray(im);
figure
 imshow (I)
```

Appendix B: Sony XC-555P Camera Module Manual

SONY

Colour Video Camera Module Component/OEM

XC-555 / XC-555P

The XC-555/555P are ultra-compact, integrated 1/2 type IT CCD colour cameras ideally suited for a wide variety of applications such as machine vision, multimedia and remote monitoring.

The cameras offer a single cable solution and are designed so that a video output signal can be obtained by only providing a power supply of 12 V DC. The ultra-compact, one-piece design eliminates the need for a bulky CCU (Camera Control Unit), allowing the XC-555/555P cameras to be easily installed in space-restricted areas.
In addition, the XC-555/555P cameras use a unique, compact NF mount lens system that can be converted into a flexible C mount lens. The option to remove housing, remote CCD-block and remote rear connector circuit allows easy OEM integration. Furthermore, the XC-555/555P are highly functional DSP cameras that can be remotely controlled by RS-232C. The RS-232C interface allows the cameras to control various functions such as Shutter Speed, Red/Blue level, Pedestal, AGC and CCD IRIS. The rugged and robust construction and high image quality make the cameras suited for the most demanding applications.

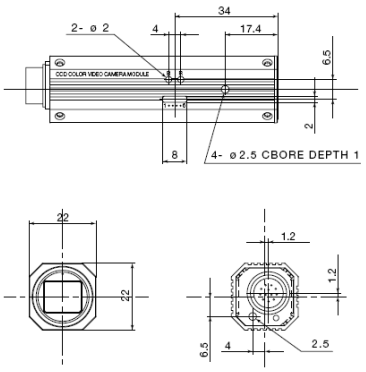
Features

- 1/2 type IT CCD
- Ultra-compact and lightweight:
  22 (H) x 22 (W) x 75 (D) mm, 60 g
  (7/8 x 7/8 x 3 inches, 2.1 oz)
- One piece camera - no bulky CCU
- Compact NF lens and lens mount
- CCD IRIS function
- VBS and Y/C outputs
- External synchronisation HD/VD, VS and CSync
- RS-232C interface to control camera functions
- Camera settings can be saved in non-volatile memory
- High shock and vibration resistance
- Single cable
- Reliable and rugged

XC-555 / XC-555P

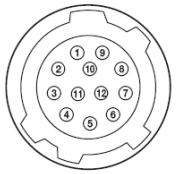
SPECIFICATIONS	XC-555 (NTSC)	XC-555P (PAL)
Image device	1/2 type IT CCD	
Colour filter	Complementary colour mosaic	
Effective picture elements	768 (H) x 494 (V)	752 (H) x 582 (V)
Lens mount	NF mount	
Sync system	Internal/External (switched automatically)	
External sync signal	HD/VD (2 to 4 Vp-p), 75 Ω VS (1 Vp-p), 75 Ω CSYNC (2 to 4 Vp-p), 75 Ω	
External sync frequency tolerance	VD/59.94 Hz ± 0.0009 Hz, HD/15734 Hz ± 0.22 Hz	VD/50 Hz ± 0.0005 Hz, HD/15625 Hz ± 0.22 Hz
Video output	VBS, Y/C (selected with the switch) VBS: 1 Vp-p, 75 Ω, sync negative Y: 1 Vp-p, 75 Ω C: C level depends on the composite video out signal	
Horizontal resolution	470 TV lines	460 TV lines
Minimum illumination	3 lx (F1.2, AGC ON)	
Sensitivity	2000 lx, F8, AGC OFF (0 dB)	
S/N ratio	48 dB (AGC OFF (0 dB))	46 dB (AGC OFF (0 dB))
Shutter speed	4 speeds slectable OFF, Flickerless, 1/1000 s, CCD iris	
CCD iris range	Auto: 1/60 to 1/4000 s RS-232C: 1/60 to 100000 s*	Auto: 1/50 to 1/4000 s RS-232C: 1/50 to 100000 s
	Max speed is not guaranteed beyond 1/4000 s.	
White balance	4 modes selectable ATW, 3200 K, 5600 K, MANUAL	
Gain control	2 modes selectable AGC, Fixed (0 dB)	
Gamma	ON/OFF *Controlled by RS-232C	
Output connector	DC IN, SYNC, VIDEO: multi 12-pin	
Power requirement	10.5 to 15 V DC	
Power consumption	2.4 W	
Operating temperature	0 to 40°C (32 to 104 °F)	
Storage temperature	-30 to 60°C (-22 to 140 °F)	
Operation humidity	20 to 80 % (no condensation)	
Storage humidity	20 to 90 % (no condensation)	
Shock/vibration resistance	70 G / 10 G (20 – 200 Hz xyz direction)	
Dimensions	22 (H) x 22 (W) x 75 (D) mm (7/8 x 7/8 x 3 inches)	
Mass	60 g (2.1 oz)	
Regulations	UL 6500 listed, FCC Class B Digital Device, CE EN61326, AS4251.1+AS4252.1	
Supplied accessories	Lens mount cap (1), Tripod adaptor (1 set), Operating Instructions (1)	

Dimensions Unit: mm



Pin Assignment (External sync.)

Pin No.	Name
1	Ground
2	+12 V
3	VBS/Y Output (GND)
4	VBS/Y Output
5	HD Input (GND)
6	HD Input
7	VD/VS/VBS Input
8	-/C Output (GND)
9	-/C Output
10	RS-232C (TXD)*1
11	RS-232C (RXD)*1
12	VD/VS/VBS Input (GND)



*1 The RS-232C switch is set to OFF at the factory. For details on how to change this setting, please refer to the Technical Manual.

Sony Contacts

ISS for Central Zone	+49 221 537 3668
(Austria, Eastern Europe, Germany, Netherlands, German-speaking Switzerland)	
ISS for Nordic Zone	+45 43 557 067
(Baltics, Denmark, Finland, Norway, Sweden)	
ISS for Other Zone	+44 1932 816 315
(UK, Ireland, Greece, Israel, South Africa)	
ISS for South Zone	+33 1 55 90 40 74
(Belgium, France, Portugal, Spain, French-speaking Switzerland)	
ISS for Italy	+39 (0) 6 334 372 27

www.sonybiz.net/vision

Appendix C: C-code for the motor controller

```
#include <stdio.h>
#include <dos.h>
#include <bios.h>

typedef unsigned char   byte;         /* 8  Bit */
typedef unsigned int    word;         /* 16 Bit */
typedef unsigned long   dword;        /* 32 Bit */

#define CHANNEL 0                  /* channel 0, connector J3 */
#define CONF_PARM 0x80               /* no board_interrupt, TTL-encoder */
#define IOADDRREG 0x210              /* board-address */
#define IODATAREG 0x211
#define INTR     0x1c               /* timer interrupt location */

#define RESET   0x00                /* LM628/629 Command Set */
#define PORT8   0x05
#define PORT12  0x06
#define DFH     0x02
#define SIP     0x03
#define LPEI    0x1b
#define LPES    0x1a
#define SBPA    0x20
#define SBPR    0x21
#define MSKI    0x1c
#define RSTI    0x1d
#define LFIL    0x1e
#define UDF     0x04
#define LTRJ    0x1f
#define STT     0x01
#define RDSIGS  0x0c
#define RDIP    0x09
#define RDDP    0x08
#define RDRP    0x0a
#define RDDV    0x07
#define RDRV    0x0b
#define RDSUM   0x0d
#define RDSTAT  0xff

#define TEST_BUSY      if (!lm628_ready(channel))\
                                              {printf("\nError: time out !"); return 0;}
#define SET_DATA_ADR    outportb(IOADDRREG,CONF_PARM | channel+1)
#define ON_TARGET      0x0400
#define LOAD_POSITION   0x0002              /* absolute position */

long    drive628(byte,byte,word,long);
void    init912(byte channel,long,long,long,long,long,long);
int     get_word();
void    put_word(int);
int     lm628_ready(byte);
void    set_watch(int);
void    interrupt timer_int();
int     parse_arg(int,char *[], char *, char *, void *);

static  volatile int time_out;
void    interrupt (*oldhandler)(void);

/*****************************************************************************
*main:         C832 programming example. This program moves then motor from
*             position 1 to 2 and 3 with a stop until you quit by ESC.
*****************************************************************PRi*190493**/
//void main (int argc, char *argv[])
void main ()
{
long i,j,kp,ki,kd,il,acc,vel,step,wait;

  kp=   1000;                          /* start values */
  ki=   100;
  kd=   1000;
  il=   1000;
  acc=  86;                           /* 10 U/s*s at 256 æs */
  vel=  1000;                         /* 50 U/s or 3000 U/min */
  step= 139;
  wait= 100;                           /* wait between movements */
  init912(CHANNEL,kp,ki,kd,il,acc,vel);       /* initialize channel 0 */

  while (!kbhit()) {
    drive628(CHANNEL,LTRJ,LOAD_POSITION,step);
    drive628(CHANNEL,STT,0,0);
    delay(1);
    while(!(drive628(CHANNEL,RDSIGS,0,0) & ON_TARGET))
      printf("\rReal Position 1: %8ld",drive628(CHANNEL,RDRP,0,0));

  }

  drive628(CHANNEL,RESET,0,0);
```

```
  printf("\n");
}

/*****************************************************************************
*drive628:     writes or reads parameter to or from LM628/629 with handshake
*****************************************************************************/
long drive628(byte channel, byte cmd, word ctrl_word, long parm)
{
  union {
                    long long_data;       /* converts long<->integer */
                    int  int_data[2];
                    } value;

  struct {
                     byte   cmd;          /* command */
                     int    report;       /* Ctrl(0)/Report(1) command */
                     int    data;         /* max. datawords */
                     int    length32;     /* 16(0)/32(1) bit valus */
                     } cmd_set[] = {
                       {RESET, 0,0,0},     /* LM628/629-commands exept RDSTAT */
                       {PORT8, 0,0,0},
                       {PORT12,0,0,0},
                       {DFH,   0,0,0},
                       {SIP,   0,0,0},
                       {LPEI,  0,1,0},
                       {LPES,  0,1,0},
                       {SBPA,  0,2,1},
                       {SBPR,  0,2,1},
                       {MSKI,  0,1,0},
                       {RSTI,  0,1,0},
                       {LFIL,  0,4,0},
                       {UDF,   0,0,0},
                       {LTRJ,  0,3,1},
                       {STT,   0,0,0},
                       {RDSTAT,1,1,0},
                       {RDSIGS,1,1,1},
                       {RDIP,  1,1,1},
                       {RDDP,  1,1,1},
                       {RDRP,  1,1,1},
                       {RDDV,  1,1,1},
                       {RDRV,  1,1,0},
                       {RDSUM, 1,1,0}
                       },
                       *dscr_ptr;         /* pointer to command description */

  dscr_ptr=cmd_set;           /* seeking command description */
  if (cmd==RDSTAT) {
    outportb(IOADDRREG,CONF_PARM | channel);
    return((long)inportb(IODATAREG));         /* status */
  }
  else while (cmd!=dscr_ptr->cmd) dscr_ptr++;

  channel <<= 1;
  TEST_BUSY;
  outportb(IODATAREG,cmd);                    /* command */

  if (dscr_ptr->data) {                       /* read/write data */
    if (cmd==LTRJ | cmd==LFIL) {
      TEST_BUSY;
      SET_DATA_ADR;
      put_word(ctrl_word);
    }
    if (dscr_ptr->report) {                   /* report command */
      if (dscr_ptr->length32) {               /* 32 Bit data */
                    TEST_BUSY;
                    SET_DATA_ADR;
                    value.int_data[1]=get_word();
                    TEST_BUSY;
                    SET_DATA_ADR;
                    value.int_data[0]=get_word();
                    return(value.long_data);
      }
      else {                                  /* 16 Bit data */
                    TEST_BUSY;
                    SET_DATA_ADR;
                    return((long)get_word());
      }
    }
    else {                                    /* control command */
      if (dscr_ptr->length32)  {              /* 32 Bit parameter */
                    value.long_data=parm;
                    TEST_BUSY;
                    SET_DATA_ADR;
                    put_word(value.int_data[1]);
                    TEST_BUSY;
                    SET_DATA_ADR;
                    put_word(value.int_data[0]);
      }
```

```
    else {                          /* 16 Bit parameter */
                TEST_BUSY;
                SET_DATA_ADR;
                put_word((int)parm);
    }
   }
  }
  return 0;
}

/*****************************************************************************
*get_word:     16 Bit input to IODATAREG
*****************************************************************************/
int get_word()
{
  union {
                int  int_data;
                byte byte_data[2];
                } temp;

  temp.byte_data[1]=inportb(IODATAREG);
  temp.byte_data[0]=inportb(IODATAREG);
  return temp.int_data;
}

/*****************************************************************************
*put_word:     16 Bit output to IODATAREG
*****************************************************************************/
void put_word(int data)
{
  union {
                int  int_data;
                byte byte_data[2];
                } temp;

  temp.int_data=data;
  outportb(IODATAREG,temp.byte_data[1]);
  outportb(IODATAREG,temp.byte_data[0]);
}

/*****************************************************************************
*lm628_ready:          test BUSY bit, returns 0 if time out
*****************************************************************************/
int lm628_ready(byte channel)
{
  outportb(IOADDRREG,CONF_PARM | channel);          /* command reg. */
  set_watch(2);
  while ((inportb(IODATAREG) & 0x01) && time_out);    /* test BUSY bit */
  set_watch(0);
  return time_out;
}

/*****************************************************************************
*set_watch:    activated or deactivated routine for timer interrupt
*****************************************************************************/
void set_watch(int cnt)
{
  disable();
  if (cnt) {
    time_out=cnt;                   /* time_out=0 after cnt * 55 ms */
    oldhandler=getvect(INTR);
    setvect(INTR,timer_int);
  }
  else setvect(INTR,oldhandler);
  enable();
}

/*****************************************************************************
*timer_int:    decremented time_out and stop interrupt routine if 0
*****************************************************************************/
void interrupt timer_int()
{
  disable();
  if (!time_out--) set_watch(0);
  enable();
}

/*****************************************************************************
*init:  basic initialisation for one C832 channel
*****************************************************************************/
void init912(byte channel,long kp,long ki,long kd,long il,long acc,long vel)
{
  drive628(channel,RESET,0,0);
  drive628(channel,RSTI,0,0);
  drive628(channel,MSKI,0,0);
  drive628(channel,LFIL,0x0008,kp);
  drive628(channel,LFIL,0x0004,ki);
  drive628(channel,LFIL,0x0002,kd);
```

```
  drive628(channel,LFIL,0x0001,il);
  drive628(channel,UDF,0,0);
  drive628(channel,LTRJ,0x0020,acc);
  drive628(channel,LTRJ,0x0008,vel);
}

/*****************************************************************************
*parse_arg:     looks for pident in the command line and converts
*              the succident value to pvalue according to pfmt.
*              Returns 1 if pident is found, otherwise 0.
*****************************************************************************/
int parse_arg(int nargs,char *pargv[],char *pident,char *pfmt,void *pvalue)
{
int i;
char *pscan = NULL;

  for (i=1; i<nargs; i++) {                 /* alle Argumente prüfen */
    if (strncmp(pargv[i],pident,strlen(pident)))
      continue;                             /* pident nicht gefunden */
    if (pvalue==NULL)
      return 1;                             /* keine Wertkonvertierung */
    if (strlen(pargv[i]) != strlen(pident))     /* blank nach pident ? */
      pscan=&(pargv[i][strlen(pident)]);        /* keine Leerzeichen */
    else
      if (i++<nargs)
                  pscan=(char *)pargv[i];               /* Leerzeichen */
    if (pscan!=NULL) {                      /* konvertieren */
      return sscanf(pscan,pfmt,pvalue) ? 1 : 0;
    }
  }
}
```

Appendix D: Matlab code for the experiment calculations

```
B=100;
Delta=0.00833;
f=25;
L=150 %;
% x poisitions
p1L1=127; %
p1L2=127; %
p1R1=75; %
p1R2=74; %
p2L1=677; %
p2L2=678; %
p2R1=630; %
p2R2=630; %

% disparities of the first point
n11=p1L1-p1R1;
n12=p1L1-p1R2;
n13=p1L2-p1R1;
n14=p1L2-p1R2;
% depths of the first point
s1=0; s2=0;
Z11=(f*B)/((n11+s1+s2)*Delta);
s1=0; s2=0.5;
Z12=(f*B)/((n12+s1+s2)*Delta);
s1=0.5; s2=0;
Z13=(f*B)/((n13+s1+s2)*Delta);
s1=0.5; s2=0.5;
Z14=(f*B)/((n14+s1+s2)*Delta);
Z1=(Z11+Z12+Z13+Z14)/4;

% disparities of the second point
n21=p2L1-p2R1;
n22=p2L1-p2R2;
n23=p2L2-p2R1;
n24=p2L2-p2R2;

%depths of the second point
s1=0; s2=0;
Z21=(f*B)/((n21+s1+s2)*Delta);
s1=0; s2=0.5;
Z22=(f*B)/((n22+s1+s2)*Delta);
s1=0.5; s2=0;
Z23=(f*B)/((n23+s1+s2)*Delta);
s1=0.5; s2=0.5;
Z24=(f*B)/((n24+s1+s2)*Delta);
Z2=(Z21+Z22+Z23+Z24)/4;

DZ=Z2-Z1;
L=300;    %%%%
alfa=(45*pi)/180;
DZR=L*sin(alfa);
Error=abs(DZR-DZ);
% DZL=DZ/cos(alfa);
DZold=Z21-Z11;
DZold2=Z22-Z12;
ErrorOld=abs(DZR-DZold);
Eimprov=(ErrorOld-Error)/ErrorOld
```

Appendix E: C-833 DC Motor Controller Manual (relevant pages)

O p e r a t i n g M a n u a l

PI
Physik
Instrumente

Operating Manual MS 38E

C-832
DC Motor Controller
ProMove Operating Software

Device and Command Reference

This Document is valid for this Product:

C-832.00 DC-Motor Controller
ProMove Operating Software Version 2.20

Release: *2.80*
Release Date: *07 May 1996*

D-76337 Waldbronn, Germany FAX: (+49)7243-604-145
EMail:info@physikinstrumente.com http://www.physikinstrumente.com

Physik Instrumente offers the following motors and stages to be used directly with the C-832:

Motors:

Order#	Power [Watt]	Encoder-Lines [lines]	Angular-Resolution [counts/rev.]
C-120.40	0.3 Watt	15 lines	60
C-120.80	2 Watt	15 lines	60
C-124.40	2 Watt	15 lines	60
C-124.50	2 Watt	500 lines	2000
C-124.51	2 Watt	100 lines	400
C-126.11	2 Watt	500 lines	59200
C-126.21	2 Watt	500 line	138400
C-128.50	6 Watt	500 lines	2000
C-128.51	6 Watt	100 lines	400

DC-Mikes:

Order#	Travel	Motor	Resolution
M-222.20	10 mm	C-120.80	0.06 µm/count
M-224.20	25 mm	C-120.80	0.06 µm/count
M-226.20	50 mm	C-120.80	0.06 µm/count

Positioning Stages:

Stage Type	Travel	Drive	Resolution
M-125.10	25 mm	M-224.20	0.06 µm/count
M-150.10	50 mm	C-120.80	0.06 µm/count
M-410.20	25 mm	C-126.11	0.017 µm/count
M-500.11	100 mm	C-128.50	0.5 µm/count
M-500.12	100 mm	C-126.11	0.017 µm/count

Positioning stages with more powerful DC-motors up to 30 Watt can be driven by the C-832 using external power amplifiers, model C-835, C-838. For further information see PI main catalog, product group MP.

2.6 Specifications C-832

Motor Types:	DC-Motors up to 6 Watt
Driver type:	PWM signal output
Interface Bus	8/16 bit PC/XT/AT 286/386/486
Addressing:	I/O mapped, address selectable
Encoder	Incremental quadrature encoder, max. 1 MHz Differential or single ended driver
Size:	115 x 160 x 20 mm (4.5 x 6.3 x .8 in)
Operating Temperature:	0 to 70 degrees Celsius
Motor Connectors:	Two DB-15 (f)
Bus connector:	16 bit PC/AT Bus can also be used in a 8-bit slot.

```
1ws
1gh
2ws
2gh
wa300
DO
  1mr-5000
  2mr2400
  1ws
  2ws
  wa450
  gh
  1ws
  2ws
LOOP 12
1sv80000
2sv160000
1mr18000
2mr24000
sv200000
gh
END
```

7. Command Reference

The following commands can be executed as single or compound commands and can be edited in the command line editor. The format is identical to that of the C-812 compatible controllers.

If a command is transferred without axis identifier or with the identifier 0, the command effects both axes. Unlike the C-812 syntax, also MA and MR may be given without axis identifier and will start both axis.

7.1 Command Survey

"a" represents the axis identifier (allowed values: 1 or 2).

Commands in alphabetic order:

[a]AB	**Abort Motion**

Stops the motor abruptly and makes the current position to the target.

CNn	**Channel ON**

This command sets the specified channel n to +5 Volts. Valid channel numbers are from 0 to 8.

If a channel number is specified, only this channel is set to HIGH. If no channel number is specified or the specified channel number is 0, all eight output channels are set to +5Volts.

Example: "CF3,CF5" sets channel 3 to HIGH. The current status of all 8 channels is reported as a binary number: "OUTPUT 00010100".

CFn	**Channel OFF**

This command sets the specified output channel n to GND (low). Valid channel numbers are from 0 to 8.

If a channel number is specified, only this channel is set to GND. If no channel number is specified or the channel number is 0, all eight output channels are set to GND.

Example: "CF3" sets channel 3 to GND. The current status of all 8 channels is reported as a binary number: "OUTPUT 00010000".

[a]DH	**Define Home**

Defines the current position as the home position and leaves the motor in the MN state.

[a]GH	**Go Home**

Moves the motor to the zero (Home) position. Leaves the channel in the MN-state.

[a]MN	**Motor On**

Activates the position control servo loop and drives the motor to the target. A move may occur if the motor was driven externally after the MF command.

[a]MF	**Motor Off**

Suspends the position control servo loop. The motor does not get any motor current, but the encoder reading is still active.

[a]MA**n**	**Move Absolute**

Moves the motor to the absolute position n. If the motor is in the MF-state, it is set ON and stays ON.

[a]MR**n**	**Move Relative**

Moves the motor relative to the current position for n counts. If the motor is in the MF-state, it's set in the MN state automatically.

aRT	**Reset**

This command resets the interrupt flags and clears all parameters. After the "RT" command, the PID parameters are set to zero. Before issuing a move command, these parameters have to be defined again or have to be reloaded with the F11

key.

[a]SLn	Set Limit withdraw step number

This command defines the number of steps the motor is moved back from the limit switch in order to release the switch.

Example: "1SL45000"

[a]ST	Motor Stop

Stops the motor with all PID filtering and the programmed deceleration slope which is the same like the acceleration value).

[a]SVn	Set Velocity

Sets the maximum velocity to n counts/s. This value can be changed on the fly.

[a]SAn	Set Acceleration and Deceleration

Sets the acceleration to the value n, given in counts/sample rate. ***ProMove*** works with a sample rate of 256 µs. The motor decelerates at the same rate.

This command should only be used if the motor is on the target.

TCn	Tell Channel

This command reports the current status of the input channels in binary values. Because all input channels are pulled to +5V on the board, the command reports "INPUT 11111111" if no lines are connected. To make one channel 0 it has to be grounded.

WAn	Wait Absolute n milliseconds

Suspends command execution for n milliseconds

WCn	Wait for Channel n

This command suspends the execution of a compound or macro command until a +5 Volts signal level is detected on channel n. Because all input channels are pulled to +5 Volts on the board, the line has to be pulled to GND externally in order to halt the command sequence. As soon as the line is released, the nect command in the row is executed.

This command does not require an axis identifier. The channel number n can range from 1 to 8 and specifies the channel number on which the trigger signal is supposed to be detected.

Example: 1MR20000,WC3,1GH

Before ececuting the command line channel #3 has to be switched to GND. Then the motor starts and goes to the target as long as channel #3 stays low. As soon as the channel is released, the motor turns back to home position.

aWS	**Wait for Motor Stop**

Suspends the command execution until the specified motor has reached its target. Not the actual motor movement is taken, but the virtual position, calculated from the velocity profile generator.

aWT	**Wait and Trigger**

This commmand can be used for triggering external devices. It works similar to the "WS" command but it generates on channel #1 a high signal for about 40 ms when the specified motor has reached the target position.

The axis identifier a is required and must not omitted.

Example: "1mr5000,1wt,rp5"

After moving 5000 steps the motor is stopped and the trigger signal is output. Then the cycle is repeated 5 times.

aFEn	**Find Edge (find reference point)**

This command starts the specified motor to a infinite position. The movement gets stopped if a reference signal is detected. Then the motor is positioned exactely at a point 10 counts before the reference trigger position.

The motor can be started in positiove and negative direction depending on the parameter n:

n=0 Motor starts in negative direction

n=1 Motor starts in positive direction

Filter Commands :

aDPn Define Proportional Term

aDIn Define Integral Term

aDDn Define Differential Term

aDLn Define Integration Limit

Report Commands :

aTP Tell position

aTT	Tell dynamic target
aTV	Tell velocity
aTY	Tell programmed velocity
aTS	Tell status
VE	Tell version number

Sequencing Commands :

DO	Start a program loop
LOOP n	Terminate a program loop and repeat the nested commands n times.
END	Terminate the program
ENDR	Restart the program