LOW-COST STEREO VISION SYSTEM FOR AUTONOMOUS MOBILE ROBOTS

A Thesis

Presented to

the Faculty of California Polytechnic State University

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

by

Connor Citron

August 2014

COMMITTEE MEMBERSHIP

TITLE:	Low-Cost Stereo Vision System for Autonomous Mobile Robots
AUTHOR:	Connor Citron
DATE SUBMITTED:	August 2014
COMMITTEE CHAIR:	Professor John Seng, Ph.D., Department of Computer Science
COMMITTEE MEMBER:	Professor Franz Kurfess, Ph.D., Department of Computer Science
COMMITTEE MEMBER:	Professor Chris Lupo, Ph.D., Department of Computer Science

ABSTRACT

Low-Cost Stereo Vision System for Autonomous Mobile Robots

Connor Citron

Something, something, robots. that

ACKNOWLEDGMENTS

I would like to especially thank my parents and family for their love and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

Introducing ...

CHAPTER 2

Background

This chapter presents some general information on stereo vision that be useful for understanding the decision that were made in developing this stereo vision system.

2.1 Computer Stereo Vision

Computer vision is concerned with using computers to understand and use information that is within visual images [9]. There are many different types of computer vision, which range from using one image to multiple images to obtain information. One image is not enough to determine the three dimensional properties of the objects within the image.

Stereo vision uses multiple images of the same scene in order to construct a three dimensional representation of the objects in the images [7]. Comparing multiple images together for their similarities and differences allows for the depth to be obtained.

Binocular stereo [11] involves comparing a pair of images. These images are normally acquired simultaneously from a scene. By searching for corresponding pairs of pixels between the two images, depth information can be determined [11]. Pixel based comparisons can require substantial amount of computational power and time. Certain assumptions are made because of the computational resources required. Camera calibration and epipolar lines [cite 14-14 and define better] are common assumptions. For example, two images of the same scene are 640 x 480 pixels in size. Each image therefore contains 307,200 pixels, which is over 600,000 pixels between the two images for one frame. For a real-time application, say 30 frames per second for example, that becomes over 18 million pixels between the two images that would need to be processed every second.

Computational requirements for real-time applications can be reduced in several ways. First, by lowering the number of pixels in the images reduces the number of pixel comparison per second. Images at a size of 320 x 240 pixels would require a quarter of the number of computations at the cost of losing some amount of detail in the images. Also, reducing the number of frames per second will decrease the amount of computing needed. Going much below 30 frames per second is noticeable to a person and can be annoying to observe a slow frame rate. A robot on the other hand, depending on its task and how fast its moving, might only need a few frames per second in order to function within a desired range. So image resolution could be more important than frames per second for a robot if details are more important than speed.

Figure 2.1 below represents a simplified illustration of binocular stereo vision. The two cameras are held at a known fixed distance from each other and are used to triangulate different the distance of objects in the images they create. The points U_L and U_R in the left and right images, respectively, are 2D represents the point P that is in 3D space. By comparing the offset of between U_L and U_R in the two images, it is possible to obtain the distance of point P away from the cameras [1].

The closer an object is to the stereo vision system, the greater the offset of corresponding pixels will be. If an object is too close to the system, it is possible

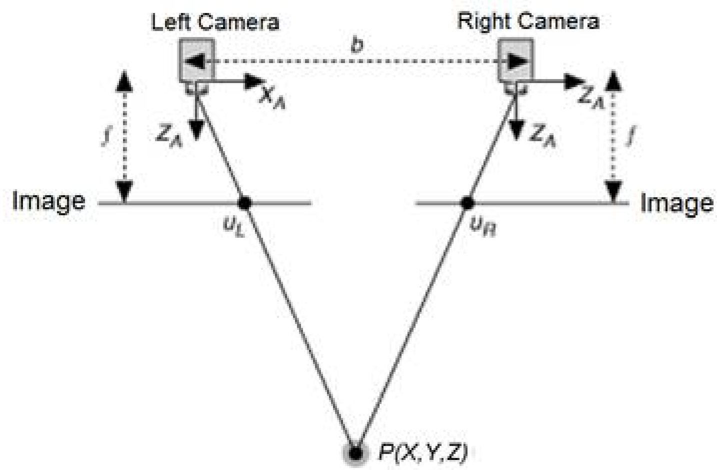

Figure 2.1: Simplified binocular stereo vision system [1].

for one camera to see part of an object that the other camera cannot. The farther an object is away from the stereo vision system, the smaller the offset of corresponding pixels will be. If an object is far enough away, it is possible for an object to be in almost the exact same location in both images. You can show this to yourself by holding a finger up close to your face, close one eye, and then alternate between which eye is open and which eye is closed. Your finger should appear to move a noticeable amount. Next, hold your finger as far away from you as you can and again alternate between which eye is open and which is closed. You should notice that your finger appears to move significantly less than it did when your finger was close to your face. That is how stereo vision works. The distance of an object is inversely proportional to the amount of offset between the two images.

2.1.1 Parallelism in Stereo Vision

Processing images for stereo vision allows for a high degree of parallelism. Locating the corresponding position of a pair of pixels is independent of finding another corresponding pair of pixels. This independent nature allows for the ability to process different parts of the same images at the same time, if there is hardware to support it.

Field Programmable Gate Arrays (FPGAs) allow for parallel processing to be implemented of the images. In section **(Implementation)** the amount of parallel processing used for the modular stereo vision system presented in this paper is discussed.

2.2 Stereo Vision Algorithms

Stereo vision algorithms can be placed into one of three different categories: pixel-based methods, area-based methods, and feature-based methods [5]. Pixel-based methods utilize pixel by pixel comparisons. They can produce dense disparity **(define!)** maps, but at the cost of higher computation complexity and higher noise sensitivity [5]. Area-based methods utilize block by block comparisons. They can also produce dense disparity maps and are less sensitive to noise, however, accuracy tends to be low in areas that are not smooth [5]. Feature-based methods utilize features, such as edges and lines for comparisons. They cannot produce dense disparity maps, but have a lower computational complexity and are insensitive to noise [5].

There are a lot of stereo vision algorithms out there [13]. In the taxonomy of [13], 20 different stereo vision algorithms were compared against each other using various reference images. Many algorithms are based on either the sum of absolute differences (SAD) or correlation algorithms [10].

An algorithm that is similar to SAD is Sum of the Square Differences (SSD). Both of these algorithms produce similar results and contain around the same amount of error [5]. SAD was chosen over the other algorithms to implement because it is simpler to implement in hardware. SSD requires squaring the difference between corresponding pixels and summing it up. Since squaring a number is the number multiplied by itself, the number will be added to itself that many times to produce the squared value. This is a lot more over head, and more hardware, than just taking the absolute difference of the difference of each corresponding pair.

2.2.1 Sum of the Absolute Differences Algorithm

SAD is a pixel-based matching method [10].

CHAPTER 3

Related Works

There are several different ways to implement a stereo vision system. Many stereo vision systems are implemented on field-programmable gate arrays (FPGAs). FPGAs allow for parallelization when processing images. Systems that use FPGAs generally can achieve a high frames per second on a decent or good image quality, but most of these systems are expensive.

FPGA Design and Implementation of a Real-Time Stereo Vision System [10] uses an Altera Stratix IV GX DE4 FPGA board to process the right and left images that come from the cameras that were attached to it. [10] uses the Sum of Absolute Differences (SAD) algorithm to compute distances. This system allows for real time speeds up to 15 frames per second at an image resolution of 1280x1024. However, the Altera Stratix IV GX DE4 FPGA board costs over $4000, [2] which makes the system impractical for non-high budget projects.

Improved Real-time Correlation-based FPGA Stereo Vision System [8] uses a Xilinx Virtex-5 board to process images. [8] uses a correlation-based algorithm, which is based on the Census Transform, to obtain the depth in images. The algorithm is fast, but there are some inherent weaknesses to it. This system can run at 70 frames per second for images at a resolution of 512x512. Unfortunately, the Xilinx Virtex-5 board costs more than $1000, [3] which is still quite expensive.

Low-Cost Stereo Vision on an FPGA [12] uses a Xilinx Spartan-3 XC3S2000 board. [12] uses the Census Transform algorithm for image processing. This allows images with a resolution of 320x240 to be processed at 150 frames per second. The total hardware for the low-cost prototype used in [12] costs just over $1000, which is a bit too pricy for a lot of projects.

An Embedded Stereo Vision Module For Industrial Vehicles Automation [6] uses a Xilinx Spartan-3A-DSP FGPA board. [6] uses an Extended Kalman Filter (EKF) based visual simultaneous localization and mapping (SLAM) algorithm. The accuracy of this system directly varied with speed and distance of detected object. The Xilinx Spartan-3A-DSP FGPA board is around $600, [4] which is fairly expensive still.

CHAPTER 4

Implementation

Architectural stuff

CHAPTER 5

Experiments and Results

Experiments and Results

CHAPTER 6

Conclusions

Concluded.

CHAPTER 7

Future Work

In the Future!

BIBLIOGRAPHY

[1] 3d imaging with ni labview. http://www.ni.com/white-paper/14103/en/, August 2013.

[2] Digi-key. http://www.digikey.com/product-detail/en/DK-DEV-4SGX230N/544-2594-ND/2054809?cur=USD, July 2014.

[3] Digi-key. http://www.xilinx.com/products/boards_kits/virtex5.htm, July 2014.

[4] Xtremedsp starter platform spartan-3a dsp 1800a edition. http://www.xilinx.com/products/boards-and-kits/HW-SD1800A-DSP-SB-UNI-G.htm, July 2014.

[5] P. Ben-Tzvi and Xin Xu. An embedded feature-based stereo vision system for autonomous mobile robots. In *Robotic and Sensors Environments (ROSE), 2010 IEEE International Workshop on*, pages 1–6, Oct 2010.

[6] P. Ben-Tzvi and Xin Xu. An embedded feature-based stereo vision system for autonomous mobile robots. In *Robotic and Sensors Environments (ROSE), 2010 IEEE International Workshop on*, pages 1–6, Oct 2010.

[7] M.Z. Brown, D. Burschka, and G.D. Hager. Advances in computational stereo. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(8):993–1008, Aug 2003.

[8] Jingting Ding, Xin Du, Xinhuan Wang, and Jilin Liu. Improved real-time

correlation-based fpga stereo vision system. In *Mechatronics and Automation (ICMA), 2010 International Conference on*, pages 104–108, Aug 2010.

[9] M. Gosta and M. Grgic. Accomplishments and challenges of computer stereo vision. In *ELMAR, 2010 PROCEEDINGS*, pages 57–64, Sept 2010.

[10] N. Isakova, S. Basak, and AC. Sonmez. Fpga design and implementation of a real-time stereo vision system. In *Innovations in Intelligent Systems and Applications (INISTA), 2012 International Symposium on*, pages 1–5, July 2012.

[11] Seunghun Jin, Junguk Cho, Xuan Dai Pham, Kyoung-Mu Lee, Sung-Kee Park, Munsang Kim, and J.W. Jeon. Fpga design and implementation of a real-time stereo vision system. *Circuits and Systems for Video Technology, IEEE Transactions on*, 20(1):15–26, Jan 2010.

[12] C. Murphy, D. Lindquist, AM. Rynning, Thomas Cecil, S. Leavitt, and M.L. Chang. Low-cost stereo vision on an fpga. In *Field-Programmable Custom Computing Machines, 2007. FCCM 2007. 15th Annual IEEE Symposium on*, pages 333–334, April 2007.

[13] D. Scharstein, R. Szeliski, and R. Zabih. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *Stereo and Multi-Baseline Vision, 2001. (SMBV 2001). Proceedings. IEEE Workshop on*, pages 131–140, 2001.