University of Florida

Department of Electrical Engineering

EEL 5934

Intelligent Machine Design Laboratory

A CCD based Image Perception Sensor for Mobile Robots

Syed Raza Hasan

TABLE OF CONTENTS

TABLE OF FIGURES

ABSTRACT

This project investigates the feasibility of application of vision sensing devices like CCDs to mobile robots. Emphasis is laid upon the simplicity and cost effectiveness of hardware and software to implement simple image processing onboard the robot. The key issue to be explored is that ‘what kind of image processing ability is required by a mobile robot’.

The concept of patch-wise correlation[1] has been explored and partially implemented on a micro-controller based system.

INTRODUCTION

Simple mobile robots have to rely on simple hardware for mobility and perception of environment. Limitations imposed by battery life and power available for locomotion, rule out the possibility of using complicated electronics hardware. Therefore a mobile robot has to depend on the simplest sensors and single chip computers to accomplish the desired tasks.

Depending on the requirements, the robot can have different sensors to ‘feel’ its environment. Vision is considered a very useful tool to perceive the environment. It may not be necessary in a vast majority of applications to acquire and analyze detailed visual information. Most of the tasks expected of mobile robots are very simple. In fact, some of the most useful robots are the ones which perform repetitive tasks in a very limited environment. Using an imaging system to gather only the necessary information is therefore a key step in simplifying the system, e.g., for detection of IR emitting objects, a sensor which can only detect infra-red radiation can drastically reduce the time and effort required to extract the same information from a more elaborate sensor covering a wider radiation spectrum.

The purpose of this project however is to try to acquire elaborate visual information, screen the information deemed unnecessary for a mobile agent and draw conclusions about different parameters like size, shape, and speed of different objects relative to the robot. All this is expected in a package which is inexpensive, light, and has a low power consumption.

1.1 System Overview

The image sensor's[a] electronics consist of two modules:

a) The controller to grab and process the image. It can also control actuators on a robot, like motors etc., in short it can act as a brain for the robot.

b) The camera board with the CCD and the latching circuitry to interface the digital signals to the controller.

1.1.1 Control Computer

The controller board is based on the Siemen's C166 micro-controller. This controller is powerful enough to grab the images from the CCD and perform simple image processing in real time. It can also simultaneously acquire information from other sensors attached to it and run motors or other actuators. The controller has a 10 channel multiplexed 10-bit A/D converter which can be interfaced to a variety of additional sensors like IR, temperature etc.. The micro-controller board currently has 64Kbytes of RAM for program and data storage. It can easily be expanded to 128Kbytes or 256Kbytes if necessary. 64Kbytes of RAM is sufficient to store the program and two complete 8-bit 160x160 pixel images (25Kbytes).

The controller is programmed using a serial link to the PC for downloading and debugging the program. The debugged program can be burned into an EPROM which resides on the board. When the program has been burned into the EPROM, the program

[a]The term sensor will be used throughout to refer to the CCD, and the micro-controller board as one system.

starts executing upon application of power to the board and the whole setup is referred to as an embedded controller.

1.1.2 Vision

The vision system relies upon a VVL1070 CCD chip[b] to acquire the images. This CCD is different from most of the other CCDs in having an 8-bit A/D on board. It also has automatic exposure control and variable frame rate controlled by digital inputs.

The CCD was purchased from the Optical Systems Division of Marshall Industries (Tel: 310-390-6608). It is packaged as a 44 pin Lead-less Chip Carrier (LCC) in a ceramic package (a PQFP version is also available for about $40). The unusual packaging requires a 44 pin LCC socket which was purchased from Newark Electronics.

The remaining hardware like the IDC connectors, capacitors, and resistors were purchased from JDR Microdevices (Tel: 1-800-538-5000).

1.1.3 CCD-Controller Interface

The CCD puts out digital data for each pixel. Each pixel value is accompanied by a pulse referred to as the pixel clock on a dedicated pin. The pixel clock is used to latch the data in a latch and is also connected to an input pin on the micro-controller. A polling routine has been used to read the data from the latch upon every pulse of the pixel clock.

The CCD has a digital input to enable its output, along with other digital inputs to control the frame size and rate. These parameters are controlled by the micro-controller using its digital output ports.

---

[b]VVL1070 from Marshall Industries for $10 in quantities of 10,000 or more. $80 in small quantities for the ceramic 80 pin LCC version for prototyping.

1.1.4 Circuit Schemetics

Even though the CCD was accompanied by Gerber plot files from Marshall Industries for an evaluation board, this PCB was designed to have 4 layers. Fabricating this kind of a board in IMDL is not possible, therefore the idea to fabricate the evaluation board was dropped. It is much easier to draw the Schemetic diagram for the evaluation board from scratch and generate the Gerber files to fabricate a single or two layer PCB. The Schemetic diagram is attached at the end of the report.

A block diagram of the system is shown in Figure 1

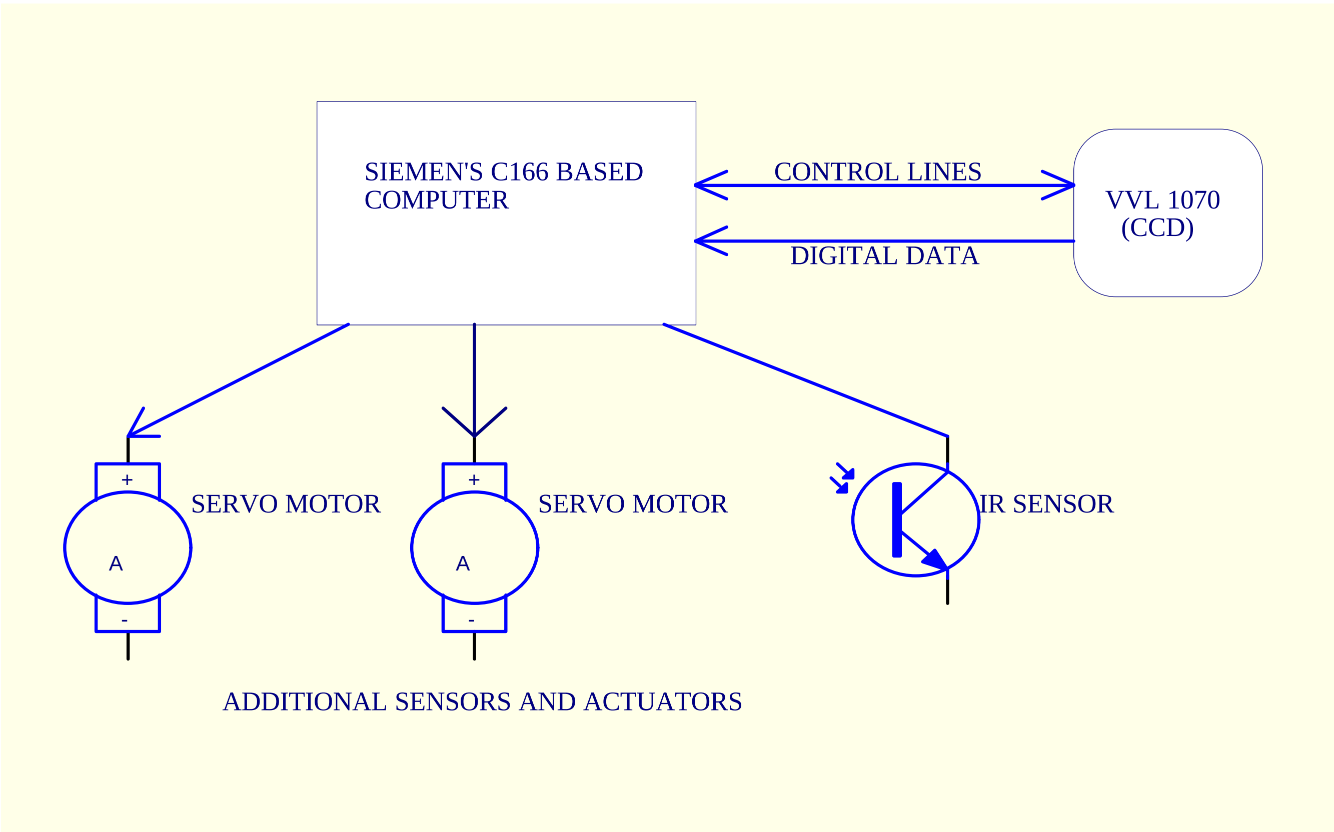

Figure 1: Proposed system with possible extensions

An IR sensor is shown in the Figure 1 as it might prove a worthy companion to the CCD by providing distance information to the target being viewed.

1.1.5 Mobile platform

A mobile platform has not used as the setup has been developed to evaluate rudimentary image processing. The results of the processing can however be used in any robot whether it is controlled by the Siemen's controller board or some other. The whole setup can act as a smart sensor by informing the main controller board of any impeding objects, or movements in view. The main controller can use this information in any way to implement a behavior. A serial link between the image sensor and the main controller will be ideal to exchange information, especially as the link will not be more than a few inches long (assuming the sensor and the controller are mounted on the same robot). The sensor setup is generic enough to be incorporated into any robot to enhance its sensory abilities without modifying the existing control hardware setup.

1.2 Possible behaviors for a 'Seeing Robot'

1.2.1 Collision Avoidance

One of the most important requirements for a mobile robot is to avoid collisions with obstacles. This is generally accomplished by using infra-red sensors and sonar devices. These devices are already sufficiently developed to provide accurate information to the robot about impending collisions well ahead of time. This concept can easily be extended to avoiding 'enemies' using CCDs. The robot might be able to recognize the 'enemies' as programmed in its 'genes' (the program) or it might be able to recognize 'enemies' with which it had an unfavorable encounters in the past. The robot can store image patterns of different objects and information about their behavior in its memory as a small data-base as it moves around in its environment. Upon 'seeing' an object, it can match it to the patterns stored in the memory and the stored information about that object's behavior can be used by the robot to advance or retreat.

1.2.2 Perception of Motion and Speed Control

The robot can be programmed to make use of images to record the speed of its travel. The concept of Patch-wise Correlation[1] has been used to extract information about movement of objects in images. As the algorithm uses changes in the image along the direction of travel, the rate of these changes can be used to calculate the velocity of the robot with respect to the observed object.

In this case, the distance to the object is an important parameter and the robot can use IR sensors to measure this distance. Based on the displacement of a certain pattern in successive frames and the distance of camera from the object, the robot can calculate its own speed with respect to that object.

The same above technique may be used to steer the robot in certain directions along with avoiding obstacles.

DESIGN AND ASSEMBLY OF CAMERA

2.1 System Electronics

2.1.1 CCD Description

The VVL1070 CCD is a 160x160 pixel area CCD. It can be read out in any combination of 160 pixels and 120 pixels, i.e., it can be read as a 160x120, 120x120, 120x160 or 160x160 pixel CCD. Only the pixels which lie in the specified combination are read out.

The CCD has an analog as well as a digital output. The digital output can be read in parallel as well as serial format. The maximum frame rate is 22.7 fps with a 12 MHz clock, configurable down to 2.8 fps.

All the above mentioned parameters can be configured with digital inputs. Further details can be obtained from the data sheet provided by Marshall Industries (Tel: 310-390-6608) who are the distributors of VVL products.

2.1.2 CCD biasing

The CCD needs some external components to bias different parameters such as black and white pixel thresholds. The circuit diagram in Schemetic section shows all the components with their values.

2.1.3 Frame grabbing circuitry

A Siemen's 80C166 based single board computer has been used to grab and process the image. The 16-bit architecture of this processor makes it more suitable for processing the image, though with careful design an 8-bit controller like a 68HC11 or an 8051 can be made to perform the same functions albeit a little slowly.

The pixel data from the CCD is accompanied by a pulse on the PCLK (pixel clock) pin. The data is valid for a short time and is therefore latched into buffers so that the u-Controller can read it even after the CCDs internal data buffers have been disabled. This circuitry is shown in the Schemetic section at the end of the report.

Initially, it was planned to map the CCD in the memory of the u-Controller. This idea was dropped because the u-Controller has sufficient ports available for a direct interface. The latching circuitry utilizes a 74LS373 latch. The data is latched into the buffer by a 35ns pulse which is generated by the PCLK. The reason for generating this pulse is to ensure that even the slowest TTL 74LS373 can be used as the PCLK itself can be very short depending on the clock input of the CCD (Half of 74LS123 was available anyway). The PCLK is also used to generate a 800ns pulse for the controller because the counter input port which polls for the pixel clock has to have a pulse valid for at least 400ns to be recognized.

A PAL has been used in the design even though currently it is only being used as a single inverter. It was included in case any more logic functionality was required. Its logic description is shown on the Schemetic. It can be replaces by a single inverter as is evident from the Schemetic.

2.1.4 Power supply

It is essential to run the u-controller and the camera off a single supply as there is no GND pin on the interfacing connector between the controller and the camera. The camera has two power connectors - one for regulated 5V, the other for unregulated 10V.

If it is supplied with unregulated 10V, the 5V connector can be used as a regulated 5V output to power the u-controller board. If the 5V connector is used as input, nothing other than a regulated 5V should be connected to it.

2.1.4.1 Camera electronics performance

The u-controller board worked without missing any pixel data upto 5.7 fps. For higher frame rates, it might be possible to use DMA to grab the image. Even though no real-time processing is currently being performed, it is possible to perform image additions or subtractions at frame rates upto 5.7 fps.

2.2 Camera Optics

The camera optics proved to be a much bigger problem than initially anticipated for the following reasons:

a) The camera electronics were prototyped with wire-wrapping. The size of the circuit board was therefore much larger than a well designed PCB would have been. Thus it was impossible to fix the circuitry in any suitable housing - like that of a disposable camera.

b) The CCD is packaged as a 44 pin LCC and therefore needed a socket with 100 mil spaced leads suitable for prototyping boards. Most of the sockets available had 50 mil zip type footprint for the dual rows on each side which is not suitable for 100 mil spaced holes of a prototyping board. The socket which was finally used dictated against the use of pin-hole lenses supplied by Marshall Electronics.

c) The active area of the CCD is 1.7mm x 1.7mm, thus making it very difficult to align lenses on top of the CCD at the exact spot. A basic pin-hole setup will not work as the hole has to be small enough to create an image but should not be too far from the CCD for an acceptable angle of view thus making the task of aligning even more difficult for a prototyping design assembled in a piece of sanitary pipe fitting (the fitting houses the camera). Images were grabbed using this setup but angle of view was very small[c] and the view was not straight either. The final optic setup consisted of a Fujinon 1:1.4/50 lens which is more of a telephoto lens than a wide angle lens. The angle of vision is not very wide but fairly decent images can be obtained with this setup. A wide angle lens is still desirable for use on an autonomous agent.

Following images of a portion of a page of text were grabbed using the Fujinon lens:

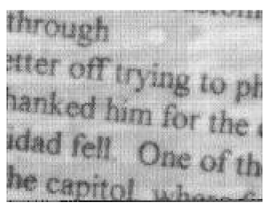

Figure 2: Image of a page of text in size 12 font

Figure 3: Image of a colored picture

[c]A similar project undertaken by Andrew S. Gavin[4] used a 3mm wide-angle CCD lens after evaluating several asepheric and pin-hole lenses. Gavin’s lens provides a 60-70 degrees angle of view.

These objects were placed at a distance of 36 inches from the camera lens. It is clear from the images that a wide angle lens will certainly solve the optical problems.

IMAGE PROCESSING

3.1 Movement recognition

Different people have tried different approaches to detect movement using image sequences. Most of these techniques however are computation-intensive and therefore not suitable for a u-controller based system on a small robot. The technique of single dimension patch correlation is fast and easier to implement on a u-controller and has therefore been tested in this project.

3.1.1 Patch-wise correlation in a single dimension

The concept of patch-wise correlation is covered in detail by Ancona[2] and Mubeen[1]. The basic idea is to take a single dimensional patch from an image which has been averaged vertically, and to move it horizontally across the second image at about the same vertical position where it was originally created in the first image and try to find the region which most closely matches the patch. This is accomplished by using the following equation:

$$\min_{\Delta x} \sum_{i=0}^{N-1} | A(i_0 + i, j_0) - B(i_0 + i + \Delta x, j_0)|$$

where i_0 and j_0 are the co-ordinates of one corner of the patch in the first image. The best match is the horizontal co-ordinate of the region in the second image most closely matches the patch.

3.2 Programming of patch-wise correlation algorithm

The concept of patch-wise correlation seems simple from the above equation and its implementation is relatively much easier on powerful computers with huge memory spaces. To implement it on a 286-class u-controller with only 64K of RAM without any higher level language is easier said than done. Assembly language routines had to be created even for such simple operations as accessing array elements. All the routines had to written for 2-dimensional arrays considering the application on hand.

Following is a brief description of the routines written to accomplish the most fundamental of tasks:

Setup_exp: This routine is executed during initialization. It looks at the central 16x16 pixel region of an image and adjusts the exposure settings of the CCD until this region shows medium brightness.

Exp_low: This routine lowers the exposure by a specified amount.

Exp_hi: This routine raises the exposure by a specified amount.

Grab_one: This routine waits until the start of a frame and then grabs and stores an image in the region specified to the routine.

Send_image: This routine sends the image to the computer through an RS-232 link. This routine is necessary for debugging.

Get_Pixel: This routine gets the value of a pixel from the image by using the base address of the image, the width of the image, the x-coordinate , and the y-coordinate of the pixel.

Put_Pixel: This routine writes the value of a pixel in the image by using the base address of the image, the width of the image, the x-coordinate , and the y-coordinate of the pixel.

Create_Patch: This routine reads a specified region in the image and creates a new array to store this data. This data is treated as a 'patch' in the image.

V_av_image: This routine averages the image vertically over a specified number of lines, i.e., the image is divided into a specified number of horizontal patches and then each patch is vertically averaged.

Match_patch: This routine takes a patch stored in memory and matches it across the specified image in memory. It returns the level of match and the location of match. Higher values indicate bad matches.

In addition to these routines, a variety of other routines were written to debug these routines. A 'C' program was developed for DOS to receive the image from the serial port of the PC, store it as a raw data file and display it[d]. This image transfer is very slow because of the 9600 baud rate used. A 160x120 pixel image consists of 19200 bytes of data and it takes about 20 seconds to transfer it to the PC.

3.2.1 Parameters for patch correlation

Patch correlation in horizontal direction requires averaging the image vertically, creating a suitable patch, getting the next image, averaging it as well and finding a match for the patch in the vicinity of the location of original patch. Size of averaged strips and patch are critical to the performance of the algorithm. A small program utilizing the aforementioned routines was executed and performed differently depending on the strip size and patch size.

Strip size: To perform patch-wise correlation, some parameters have to be specified depending on the image size, quality, and the kind of objects being viewed. As mentioned, an image has to be averaged vertically over horizontal strips for a horizontal patch search.

The size of the strips is very important to the performance of the algorithm. If the strip size is too small, a suitable match might not be found because the image region containing the patch might move slightly in the vertical direction, or the lightening conditions might change slightly. The strip size should also take into consideration the angle of view as a CCD without a very wide angle lens (like the case in this project) will result in large changes in the image corresponding to small movements in the scene.

Another issue is that of the averaging of strips themselves. If a certain number of strips are averaged and the next lower strip is averaged independently of the one above it, there will be a boundary between the two strips which may be undesirable. It should be possible to overcome this problem by using taller strips with the selected patch in the middle of a strip in the image. This should make the algorithm more robust in the presence of vibrations.

Patch selection and size: Selection of the patch is critical to the success of this algorithm. Depending on the objects in view, different patches will perform differently in similar environments. A small patch might work well in a cluttered environment but will probably fail in a sparse environment as it will find false matches. Similarly, a large patch will work well in sparse environments but might not find a match in cluttered environments. It might be possible to make the patch size a variable and adjust it depending on the kind of environment, e.g., a patch may be created between two vertical edges for horizontal matching. If two edges are found far apart in the image indicating a sparse environment, a large patch will be created and vice versa.

---

[d]This routine can be used to transfer the image to the PC from the robot through some kind of a wire-less link for some kind of an application even though the robot will not be able to do anything else during this time.

3.2.1.1 Algorithm performance

Extensive testing has not been done using a variety of combinations of the above two parameters but horizontal strip sizes of four rows of pixels were used with a fixed patch size of 48 pixels. The sensor found small horizontal movements fairly well which was expected due to the small angle of view of the sensor. Also, as patch re-creation upon the patch's going out of view has not been implemented yet, the sensor keeps looking for the same patch even after the scene has changed drastically finding extremely 'non-matches' as matches. Patch re-creation may be implemented before the demonstration of the project.

Edge detection in a single dimension is a feature which might be extremely useful to the algorithm as it will enable intelligent selection of patches.

3.3 Conclusion

This project changed from implementation of image processing algorithms to development of an image sensor. Some unforeseen problems delayed the implementation of actual image processing to the very end of the semester. It will be continued in the next semester to gather some experimental data and program refinement. A PCB will be developed to reduce the size, cost, and weight of the camera so that others interested in using a $100 digital camera will not have to through the trouble of assembling and debugging the hardware.

Implementation of the basic patch correlation algorithm is just scratching the surface of possibilities. It might be possible to implement patch correlation in four directions with four different patches to detect looming of objects.

This experiment provides a test-bed for implementation of various algorithms for robot-vision. The results from the brief experimentation have been encouraging considering that the sensor is small, light weight, has low power consumption, and can be powered by a battery pack. More experimentation and effort in software will certainly improve the performance of the sensor to the point where it can control different functions of the robot itself (the 10 inputs of the A/D port and 12 digital I/O lines are available for interfacing to sensors and switches). With suitable visual capabilities along with IR and bump sensors, this should make a fairly powerful platform to implement complicated behaviors for robots.

SCHEMETICS

REFERENCES

[1] John G. Harris, Mubeen Khan, and Keith L. Doty. Motion-based visual behavior for mobile robots. IMDL, Electrical Engineering Dept., University of Florida, March 25, 1995.

[2] Nicola Ancona and Tomoso Poggio. Optical Flow from 1d correlation. Application to a simple time-to-crash detector. Technical Report A.I.Memo No. 1375, MIT Artificial Intelligence Laboratory and Center for Biological and Computational Learning, October 1993.

[3] Marshall Industries. Specification sheet for the VVL1070 CCD chip.

[4] Andrew S. Gavin. A fast, cheap, and easy vision system for an autonomous vacuum cleaner. AAAI Fall Symposium series, October 1993.