Developing a Flexible Range Sensing System for Industrial Inspection Applications

by

Yoshen Hou

Thesis submitted to the Faculty of the

Bradley Department of Electrical Engineering of

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Electrical Engineering

APPROVED:

Dr. Richard W. Conners, Chairman

Dr. D. Earl Kline, Co-Chaiman

Dr. A. Lynn Abbott

Dr. Daniel L. Schmoldt

June 15, 1993

Developing a Flexible Range Sensing System for Industrial Inspection Applications

by
Yoshen Jason Hou

Dr. Richard W. Conners, Chairman
Dr. D. Earl Kline, Co-chairman

Electrical Engineering

ABSTRACT

This thesis describes the development of a range sensing system. The goal was to create a range sensor that is robust and flexible so that a number of applications within the forest products manufacturing environment can be addressed. Features of the system include: the capability of producing spatially registered image pairs of range and intensity, the ability to generate both range and intensity very quickly, the applicability to a wide variety of industrial applications, the ability to handle large depth-of-field range sensing problems, the ability to do real-time data processing, and the capability to do extensive system diagnostics under complete software control.

A triangulation based plane-of-light optical method is employed to extract range information. The research shows that this method suits range sensing applications where conveyor belts are involved. An in-depth study of the triangulation method is included. In the study it shows that this method also supports large depth-of-field range sensing. A dedicated signal processing hardware, built on the Micro Channel interface, performs pipelined image processing and generates range and intensity images in a spatially registered form. The hardware is designed to support several modes of operation, for the

purpose of facilitating optical adjustments and calibrations. The hardware self-diagnostic facility is also included in the design.

A memory management scheme is provided that facilitates real-time data processing of the range and intensity images. The experiments show that this scheme provides a real-time environment for software processing. This thesis also contains a theory exploring the limitations of the measurement accuracy of the range detection algorithm employed in the prototype system.

The maximum data generation rate of the prototype system is 380 range/intensity lines per second at 128 range/intensity pixels per line. Several proposals toward future work are included that aim at improving the speed as well as the measurement accuracy of the prototype system.

Acknowledgements

I would like to express my thanks to Dr. Richard Conners, who as the chairman of the committee offered a great deal of advice during the research and the writing of the thesis. My thanks go to Dr. D. Earl Kline for his support and his confidence in my abilities, and for his assistance in the thesis writing. Also thanks to Mr. Philip A. Araman for his constant encouragement that makes this project possible. I also thank Dr. A. Lynn Abbott and Dr. Daniel L. Schmoldt for their time as members of my committee and their assistance in the development of the thesis.

In addition, thanks to:

- Tomas Drayer for his guidance in hardware design.
- Lichung Kuo for her assistance in software programming.
- Harold Vandivort for supporting the mechanical design.
- Ron McIlwain for his help in improving the writing.

I dedicate this thesis to my brother, Suen Hou, for his unflagging faith in me.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Background

The forest products industry employs more than 1.6 million people, has a total annual payroll exceeding $34 billion, and generates over $83 billion annually in value-added manufacturing. However, the industry faces severe economic and technical problems that are limiting its growth. The increasing cost of a decreasing supply of high-quality hardwood timber resources along with labor-intensive manufacturing methods has pushed manufacturing costs close to unprofitable levels. Furthermore, competitive pressures from foreign companies are threatening these industries. Changes in the raw material supply and global economy are only a few of the large variety of factors that interact to adversely affect the industry. If the industry is to survive and grow under such changes, the industry must be able to increase their production efficiency.

Current wood processing operations examine lumber manually to locate and identify the types of defects that need to be removed to assure a quality product. In general, the forest products manufacturing employee is typically an entry level position with pay at minimum wage. Manufacturing facilities are dusty and noisy, processing decisions have to be made quickly, and the work is tiring. Under such conditions, human performance can be much less than optimum (Regalado, et al [1]; Huber et al. [2]). Therefore, the development and installation of automatic inspection systems will be

instrumental to the future success of the forest products manufacturing industry.

State-of-the-art manufacturing technologies that are available for the metal working, pharmaceutical, or electronics industries seemingly have a great deal to offer the forest products industry. However, directly utilizing these technologies is difficult due to the unique physical properties of wood. Wood as a raw material is very heterogeneous with many variable properties. Physical properties in wood such as moisture content, weight, density, location of defects, shape, color and grain orientation have a large effect on the quality and cost of the finished product. The random variability in the location of features present in wood such as wane, knots, splits, and warp has a substantial effect on the way wood is converted from one form to another. Most of these features are treated as defects to be removed in the manufacturing process.

To provide uniform and consistent wood products to the customer at a low cost, the properties of wood must be reliably detected. There are three main categories into which features in hardwood lumber may be classified. These are: 1) visual surface features (e.g. knots holes, splits, decay, color, grain orientation), 2) geometric features (e.g., warp, crook, wane, thickness variations, voids), and 3) internal features (e.g., moisture content, density, honeycomb, decay). Surface, geometric, and internal features have been categorized in this way because they often require different detection methods that depend on where they appear.

The most popular sensors that have been applied for detecting features in wood include microwaves, capacitance sensors, x-ray imaging, laser profiling systems, and optical scanning systems. Each of these sensing modalities is capable in detecting certain types of features and not others. Generally, it has been recognized that no single sensing modality can be used to automatically detect all categories of features in wood (Szymani and McDonald [3]). Conners et al. [4] have been developing a general purpose machine

vision technology for the forest products manufacturing industry that will include several different sensing modalities. This general purpose machine vision system includes only a color scanning system for detecting visual surface features. This system has been successful in automatically detecting objectionable features in wood such as wane, splits, checks, knots, and stain that need to be removed in the manufacturing process. In addition to removing objectionable features, another important function of the manufacturing process is to mold and shape wood into a value-added product. This function requires that geometric features in wood can be accurately detected. Hence, a general purpose machine vision system will need to incorporate additional sensing modalities that can detect these geometric features.

Unfortunately, there has been very limited research in automatic inspection for geometric features in wood. For example, low resolution board profile detection systems have been developed in lumber manufacturing industry to increase lumber yield (Grove [5]). However, little research and experimentation has been done in testing the application of sensing geometric features for a wide variety of forest products manufacturing applications. Hence, research is needed in developing and testing sensing modalities for geometric features in wood. Furthermore, research is needed to integrate these sensing modalities with other modalities such as optical scanning systems.

1.2 Objectives

The overall objective of this research is to design a robust and flexible automatic inspection technology for the forest products manufacturing environment. This research will focus on automatically detecting both surface and geometric features in wood by integrating laser ranging information with gray-level intensity information. The goal is to

create a general purpose system that can address a variety of problems in the forest products manufacturing industry. The specific tasks of this research include:

a) Design and build a general purpose laser ranging/intensity sensing system that can address a number of industrial applications
b) Design a stand-alone prototype hardware system that provides not only all the needed functionality but also provides a great deal of flexibility.
c) Develop software to perform real-time image processing and to control all of the operation of the prototype system hardware.
d) Determine errors and limitations of the prototype system.
e) Demonstrate the capabilities of the system.

The development of a range sensing system that is capable of producing both range and intensity data that are perfectly registered spatially is something that has long been pursued ([6]). An obvious advantage of the approach proposed in this research is that a single sensor module can function as a combination of sensing modalities: 1) range profile sensing and 2) gray-level intensity sensing. Furthermore, the flexibility that will be designed into the system will be able to accommodate various different optic configurations and resolutions so that a wide variety of industrial applications can be explored.

1.3 Rationale and Significance

1.3.1 System Overview

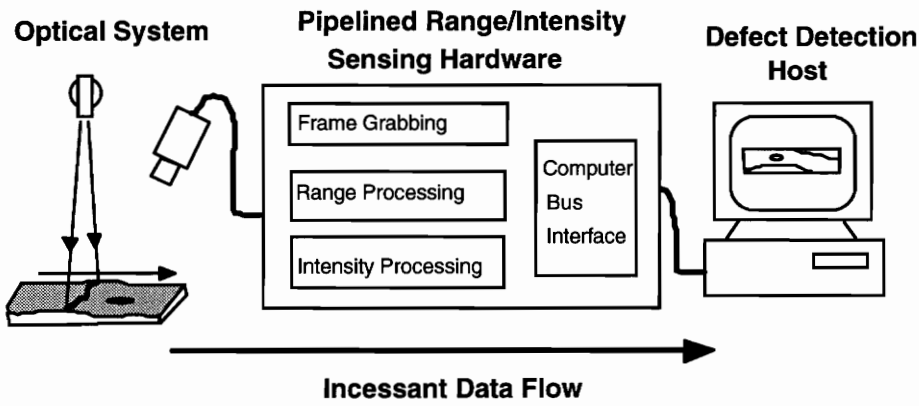


Figure 1.1 System configuration of the prototype.

The prototype system employs the triangulation technology in extracting range profiles. This technology is deemed most appropriate for the industrial profiling purpose. As shown in Figure 1.1, a video camera views a plane-of-light illuminated object at an angle. The image signals produced by the video camera is sent to a dedicated hardware system to produce both thickness profiles and intensity profiles of the object surface. The hardware system performs four functions: Frame grabbing, range processing, intensity processing, and provides an interface to a host computer bus. The frame grabbing capability allows users to obtain information about the viewing conditions of the video camera. Optical adjustments and calibration can be performed by invoking this function. The range processing module produces an array of range profiles of an object's surface. The intensity processing module generates intensity images of an object's surface. The computer interface is designed to transfer the two arrays, the range and intensity images, to the host computer memory for further processing and defect detection.

1.3.2 System Design Features

Automatic inspection problems in wood processing encompass a wide variety of parameters. The size of the material being processed can range from very large to very small. The shape of the material can range from simple rectangular pieces to very intricate and detailed pieces. The required levels of resolutions can range from very coarse to very fine depending on the particular manufacturing process. Therefore, a great deal of flexibility as well as robustness must be designed into any sensing system that is to serve as an test facility for all of the parameters encountered in wood processing.

Although video cameras are commonly available at low cost, a stand-alone intensity-based sensor is not flexible enough to explore the range of applications that are typically encountered in a forest products manufacturing environment. The thrust of this research is to add range profile information of the viewing scene. By adding range profile information with intensity information provided by the video camera, the research of computer vision can advance to a stage of providing visual capability similar to that of human vision.

The prototype range and intensity sensing system can be divided into three subsystems: 1) the optical subsystem (described in Chapter 3), 2) the hardware subsystem (described in Chapter 4), and 3) the software subsystem (described in Chapter 5). The optical subsystem was designed such that very uniform and focused lighting conditions could be created and controlled. The optical subsystem considers imaging optics that are flexible enough to test many range measurement configurations. These configurations include settings of sensor working distance and viewing angle, sensing width, and depth of range sensing.

A dedicated hardware subsystem was designed to perform the required frame

grabbing. The frame grabbing hardware facilitates the optical adjustments and calibration. This hardware subsystem also condenses each video frame captured into two 128-byte arrays, one containing 8-bit range data and the other containing 8-bit intensity data. The unique feature of this hardware is that it generates range and intensity images that are perfectly registered in a pipelined manner. The hardware subsystem also uses a timing format in controlling processing units so that most video cameras can be accommodated. Furthermore, the hardware subsystem implements the "Center-Pixel" algorithm developed in this research for range detection. Other features included in the hardware subsystem are real-time range data conversion using a hardware look-up table and a built-in fault detection facility that allows the host computer to perform hardware diagnostic tests.

The software subsystem was developed to provide the user access to all hardware features. The software system manipulates thirteen I/O registers to perform hardware control. A memory management scheme is developed for a software environment that allows real-time image processing of range and intensity images. Finally, the software includes a set of utility subroutines that allow easy and flexible way for user programs to control the hardware prototype.

A range detection theory is developed to analyze the accuracy performance of the system (presented in Chapter 6). The conclusion drawn from the theory states that the range measurement accuracy of the system is twice the diameter of the laser beam. To achieve better measurement accuracy, we can reduce the beam diameter by further focusing the laser beam using laser scan lenses. Doing so, we can build a system capable of performing range measurements with accuracy on the range of microns. Chapter 8 presents several proposals for improving measurement accuracy, enhancing data generation rate, and widening application scope. By way of this study we can build a range and intensity sensing system flexible for a wide variety of industrial applications.

Chapter 2
Literature Overview

2.1 Forest Products Manufacturing

The major consumption of high and medium grade hardwood timber lies in the manufacture of furniture, cabinets, flooring, millwork, and moldings. To produce final products, two steps are involved in processing hardwood logs: primary processing and secondary manufacturing. Primary processing of hardwood logs includes transporting logs from woodlot to sawmills. Logs are classified into two major categories: veneer logs and saw logs. They are then processed in different mills. The veneer logs are sawn into flitches and then sliced to produce veneer in veneer mills. The saw logs are broken down into lumber of standard thickness. Lumber and veneer serve as the intermediate results toward final products. A secondary remanufacturing facility obtains the boards and converts them into furniture/cabinet parts, flooring, or moldings. Final cutting, milling, gluing, staining, and assembly are among the categories of secondary remanufacturing of lumber boards. There are many plants dedicated to specific final products.

2.1.1 Primary Processing

Figure 2.1 shows the typical components and material flow for a hardwood sawmill in the Southeastern region of the United States. The figure indicates that a

general sawmill includes a debarker, a headrig, an edger, a resaw, and a trimmer (Kline et. al. [7]). The debarker removes bark and debris from logs to avoid prematurely dulling saw blades. Debarked logs are queued on a log deck and sent to a headrig. These logs are broken down at the headrig into cants and flitches. A gang resaw is used to assist in cutting log cants into lumber. The resaw reduces the number of headrig passes on a given log and thus increases headrig production. To increase lumber grade, boards are routed to an edger for wane and edge defect removal. The edger is in general placed behind the head saw.

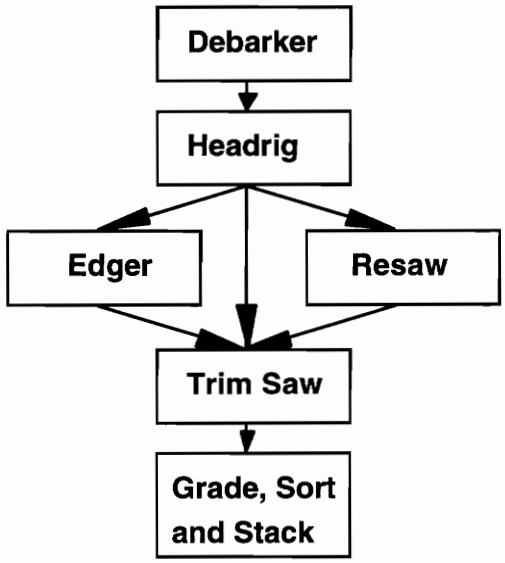


Figure 2.1. Basic components of a hardwood sawmill [7].

Boards coming out of the headrig, edger, and gang resaw are sent to the trim saw to remove end defects. The principal objectives of trimming are to cut the ends of boards so that they will be square and parallel with each other, to cut the board into one or more pieces suitable for commercial lengths, and to eliminate defects to create more valuable boards. Finally, edged and trimmed lumber is graded, sorted and stacked.

It is always desirable to achieve maximum lumber value production. Lumber value depends on balancing between lumber volume and lumber grade. Several elements in

primary processing affect hardwood lumber value. Among these elements, edging and trimming practices have one of the greatest effects on lumber value. Many factors influence the performance of current edging and trimming operations for sawmills employing manual processing. Since human operators are involved and edging and trimming processes requires laborious operations, human factors presides over the performance. These factors include: (a) operator knowledge of hardwood grading rules, (b) operator skill, (c) processing speed the operator must maintain, and (d) operator fatigue (Kline et. al. [7]). Besides human factors, other factors include machinery maintenance and raw material quality. Regalado et. al. [8] performed a study on the value losses on manual operation of edging and trimming. The study shows that value recoveries ranged from 62 to 78 percent at various different sawmills. It was concluded that substantial increases in value can be expected from edging and trimming optimization.

In another study, Regalado et. al. [1] concludes that it is possible to obtain lumber values higher than actual sawmill output using automation systems based on limited sensory information. For example, by using wane as the only basis for determining edging and trimming solutions, the optimization process can yield an average value recovery of approximately 81 percent of the optimum value. When based on wane, knots, and decay information, the optimization can give an average value recovery of approximately 88 percent of the optimum value. If locations of all defects are given without knowing the defect type except for wane, the average value of 95 percent of the optimum value can be achieved.

The hardwood lumber manufacturers are recognizing the need to reduce edging and trimming losses so that the value lumber produced can be improved. However, very few sawmills are able to accomplish these objectives. To improve edging and trimming operations, factory automation, computer-aided processing, and operator training are

needed. A large hardwood sawmill (40 million board feet per year) in Louisiana recently upgraded its equipment to include an edging optimizer (Griffin [9]). The mill has achieved a 40 percent production gain given a 20 percent increase in capital investment. The large return on investment justifies the need to modernize hardwood sawmills. However, the investment return depends on the size of the sawmill.

Only large sawmills as illustrated above, can readily adapt to costly automation equipment that has typically been used on softwood lumber manufacturing. Until systems that are specifically developed for addressing unique requirements of hardwood lumber manufacturing, the automation equipment will remain costly. To automate hardwood edging and trimming operation, the scanning system should be able to produce information about board geometry and about as many surface features as possible that are important in determining lumber grade. Board geometry includes length, width, thickness, and wane. Other features needed to establish lumber grade include knots, holes, stain, mineral streak, decay, splits, and checks. The scanning system should also take into account the difference in features across different species of timber since a hardwood sawmill processes more than one species. Hence, the board scanning system must be as flexible and robust as possible, yet simple enough to minimize the cost of the system.

2.1.2 Secondary Processing

The secondary manufacturing of wood produces often involves four stages of processing: preparing the lumber, dimensioning the pieces, shaping the pieces into parts, and assembling the parts into finished goods. Green lumber is air- and/or kiln-dried at the lumber preparation stage before sending it to the shop floor. The dimensioning of the pieces occurs at the rough mill which involves cutting the boards, planing them to achieve

uniform thickness, cross cutting and ripping the boards to produce defect-free pieces, gluing up the pieces, and trimming the glued up pieces to ensure smooth, straight ends (Kline et. al [7]). The shaping stage consists of all the processes that occur between the rough mill and the assembly stage. At this stage, basic shapes (internal as well as external) are cut from the standard dimension pieces. For these operations, planers, routers, mortisers, lathes, tenorners, shapers, profiling machines, drills, carving machines, and boring machines may be used. Once the pieces have their basic shapes, the edges are shaped, sanded and inspected. In the fourth stage, the different components of the product are assembled, inspected, stained/finished, the hardware is attached and one final inspection is performed.

The operations involved in secondary manufacturing have remained basically the same in the last century (Huber et al. [2]) and, hence, rely heavily upon the skills of human operators. These operators must be able to see defects, calculate the proper position of a wood piece during the machining process, manually position the board, and remain attentative to quality of the machining process at all times. Consistently performing these tasks is very difficult for humans in an industrial environment. In a previous study by Huber et al. [2], it was found that human operators performed at about 68 percent efficiency. This low efficiency points to the need for updating the secondary manufacturing mills with new automated inspection and processing technologies.

To provide performance efficiencies that exceed that of human operators, automated inspection and processing systems must be able to reliably detect surface features on wood. Therefore, the system must be able to detect the same features described in primary processing operations. These features include geometric shape, knots, holes, stain, mineral streak, decay, splits, and checks. Additional features that need to be detected in secondary manufacturing operations include color, grain patterns, and

internal features that may be uncovered in subsequent machining such as hidden knots, honeycomb, and decay. Since detail becomes more important in determining the value of the finished product, the automated system must be able sense much information at very high resolutions.

In summary, improving the performance of the many critical operations involved in both primary and secondary manufacturing must involve some degree of automation. While automation technologies are readily found in the softwood lumber industry, they are very scarce in the primary and secondary manufacturing of hardwood products. Due to the unique nature and the diversity of hardwood processing operations, automatic inspection technologies targeted specifically for these operations are in their infancy and much more basic research is needed.

2.2 Status of Automatic Inspection Systems for Hardwood Processing

As indicated earlier, the forest products manufacturing industry relies heavily on skilled human operators to perform critical processing operations. Often, these operators do not have the necessary skills to adequately perform these operations. Even highly skilled operators have a difficult time working at peak efficiency since the forest products manufacturing employee must make decisions very rapidly in a dusty, noisy, and tiring environment. To increase their overall productivity and to remain profitable, the industry must be able to update their processing equipment and install automatic inspection systems.

Automated manufacturing technologies that are used in other industries such as the metal working, pharmaceutical, or electronics industries can be used to help automate the forest products manufacturing industry. However, these technologies cannot be directly

used due to the unique physical properties of wood. Wood is a very heterogeneous material, with many variable properties. The random location of growth features present in wood has a substantial effect on the way wood is converted from one form to another. An automatic inspection system must be able to reliably detect those features that have the most important influence on the quality or value of the final product.

Over the past five years, research at Virginia Tech has been directed at developing automatic inspection technologies to address a variety of different applications for primary and secondary hardwood manufacturing. Substantial progress has been made in developing hardware and software systems for color imagery to locate and identify surface features on hardwood lumber (Conners et al.[4]; Conners et al. [10]). This automatic inspection system based on color imagery has been found to work equally well for a number of wood species including red oak, white oak, cherry, maple, walnut, poplar, hickory, and white pine. The recognition software has been developed to identify splits, checks, knots, holes, and wane. Work is continuing to further develop these algorithms to recognize other surface features.

Although the ability to sense color information is very important in evaluating the quality of hardwood, color imagery sensors alone are not adequate to sense all of the important features present in wood. It is generally recognized that no particular sensing technology can distinguish all features for all applications (Szymani and McDonald [3]). Different sensing technologies being applied today for defect detection are: microwaves (King [11]; Portala and Ciccotelli [12]), capacitance sensors (McDonald and Bendtsen [13]; Steele et al. [14]), x-ray imaging (Kenway [15], Portala and Ciccotelli [12]), optical scanning (McMillin et al.[16]); Conners et al. [4]), and laser scanning. Each of these sensing modalities is adept at detecting certain types of features present in wood and not others.

With the above research efforts, it can be concluded that a multiple sensing system is necessary for automatic inspection systems to be broadly useful for a variety of hardwood processing applications. As an example, it is difficult to locate and identify all surface features on hardwood lumber using only color image data, regardless of the computational complexity of the algorithms employed (Conners et al. [10]). Knots can be the same color as clear wood, and hence may be misclassified as clear wood. By combining information from color cameras and other sensors such as x-ray sensors and laser-based ranging camera systems, the accuracy and speed of lumber scanning systems can be much improved.

With the many different applications found within primary and secondary processing operations, a flexible automatic inspection technology is needed that can be used to test various different sensing technologies. Present research at Virginia Tech involves developing a multiple sensor machine vision prototype that can be used to study a number of these applications (Araman et al. [17]; Conners et al. [10]). The prototype will integrate color camera systems, x-ray scanners, and laser-based ranging systems with a materials handling system, a computer system, and application software. Work has been completed on integrating a flexible color scanning system into the machine vision prototype. The research reported in the subsequent chapters describes the development of a flexible laser-based ranging system and the design features that will allow it to be integrated into the multiple sensor machine vision prototype. The next section briefly summarizes range sensing techniques as background material for this research effort.

2.3 Range Sensing techniques

Optical ranging methods can roughly be divided into two main categories: (1) active techniques that require the use of an auxiliary light source, and (2) passive range detection techniques that use only the ambient light (see, e.g., [18]). The auxiliary light source used in active techniques is designed so that the problem of extracting range information is made as simple as possible. Passive methods include using stereo image pairs or the detection of relative motion of an object's surface. However because of the complexity of determining scene disparities in stereoscopic image pairs and in image sequences, the use of passive methods is limited to very simple scenes.

2.3.1 Ratio Image Range Sensing

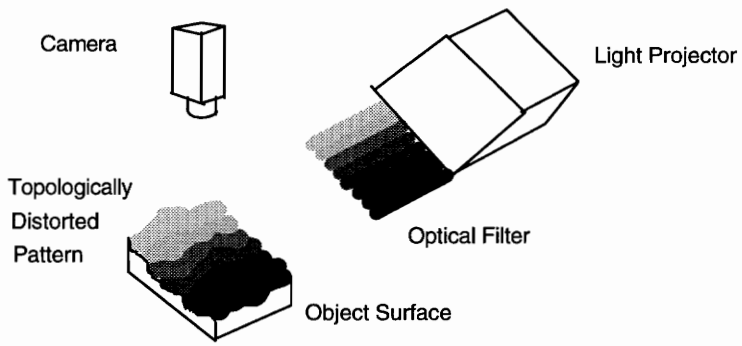


Figure 2.2 Ratio image sensing technique.

The range sensing technique using ratio images depicts a way to "link" pixel intensities of an image to the depth values of the scene. In its simplest form of implementation (J. Schwartz[19]), a camera takes two pictures of the object using two light patterns for illumination. The imaging geometry is illustrated in Figure 2.2. The first picture is taken when the object is illuminated with a uniform light beam. This picture contains the intensity outlook of the object surface. The second picture is then taken

when the object is illuminated with a light beam of a known intensity pattern. As shown in the figure, the pattern can be a ramp. Different patterns can be created by projecting a uniform light beam through optical filters coded with desired transmittance functions.

The acquired ratio image can be deemed a topologically distorted image of the projected light pattern (known pattern), whereas the degree of distortion is determined by the height of the object surface. Triangulation is the source of distortion. Figure 2.2 clearly shows that the degree of distortion is directly linked to the surface shape. The purpose of using a known light pattern is to establish a direct mapping between height values and the pixel intensities of the ratio image. In other words, after obtaining both the ratio image and the second projection pattern, a data processing host must be able to establish a mapping relationship such that intensities recorded in pixels of the ratio image can be traced back to the location where the light originates from on the optical filter. Aided with the triangulation geometry, a ratio image can then be converted into a range image by way of data recovery.

The ratio image represents a structure light technology that requires complex data computations. The basic assumption is having a "static" operating environment that allows non-disturbed consecutive exposures. This technology obviously can not be adopted in a highly motional sawmill. Besides, a highly complicated computing machine has to be devised in order to achieve real-time operation. This technology is too costly to develop.

2.3.2 Moiré Effect Range Sensing

Moiré interferometry is a well established technique in the study of displacement measurement. It is extensively used for material analysis such as stress deformation.

Although it is capable of achieving displacement resolution of $\lambda/2$ (λ is the wavelength of the light being used), it can not be implemented in environment such as the lumber inspection station. A simple reason is that a high precision optical grating has to be placed close to the material in order to create diffraction patterns. Besides, to find contours, a dedicated hardware has to count interference fringes to find out surface values. For situations where there is a big gap on the surface, the continuity of fringes is lost and results in unrecoverable errors. A thorough study of this technique can be found in [20].

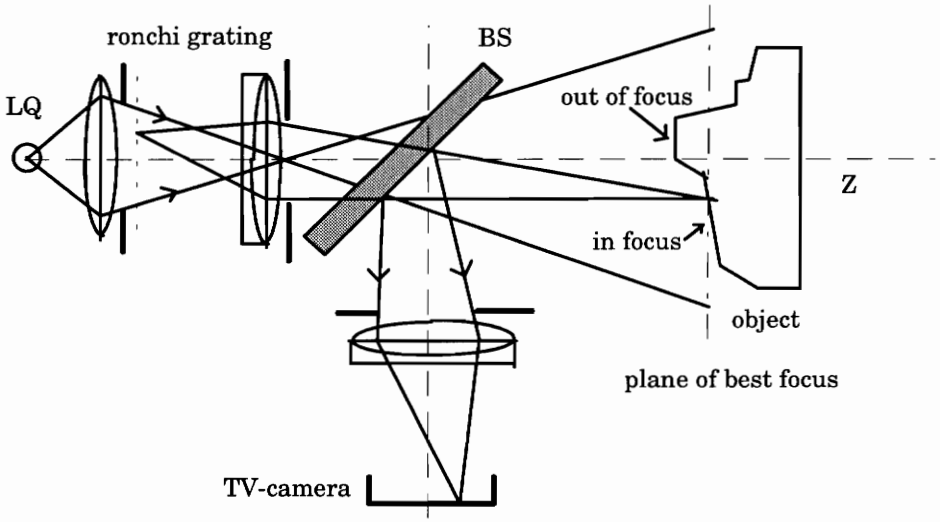


Figure 2.3 Moiré effect focus sensing technique[22].

Nevertheless, several techniques employing moiré geometry have been proposed and implemented. Kai Engel Hardt proposed a focus sensing technique in 1988 [21]. The optical structure is shown in Figure 2.3. Parallel light lines are projected onto the object surface. These lines are diffraction beams created by focusing incoherent light coming out of a ronchi grating (an optical filter carved with high frequency equidistant fine slits) onto the object surface. A very small depth-of-focus lens is used to focus these beams. The purpose of using a small depth-of-field lens is that only surface locations with distance to the lens equal to the focal length have smallest projected line width. If a dedicated system

can identify locations where thinnest lines exist, their distance to the lens is equal to the focal length. Using a very small depth-of-field lens helps narrow down the difficulty of identifying nicely focused light lines.

A video camera is installed to record projection images. A beam splitter duplicates the image of object surface. The focusing lens of the video camera is exactly the same as the one used for projecting diffraction patterns. The principle is to obtain a completely "duplicated' environment to be shown onto the image sensor, where the distance of light traveling path is the same for both the transmitting and returning paths, for the purpose of retrieving locations where light lines are nicely focused onto the object surface. To measure the depth values of object surfaces, a dedicated electronic circuit finds locations of best focused light lines using camera signals. Sequentially moving the light/camera module in the depth axis with known increments and performing best focusing computations, a complete 3-D information can be acquired. Hence the name focus sensing.

The resolution performance of this technology depends on the mechanical precision of incrementing the light/camera module. This of course can be improved by switching to high precision translation stages. However, this technique requires stationary objects during the period of depth directional motions. It therefore can not accommodate moving lumber.

2.3.3 Time-of-flight Range Sensing

Time of flight range finders use the same concept as a conventional electromagnetic wave radar. The principle is measuring the time interval needed for a round trip propagation between the transmitted light and its return signal. An alternate

name is called laser radar. The motivation behind this scheme comes from the fact that lasers very low beam divergence even after traveling for a long distance. With the help of precision time measuring electronics, a laser radar can achieve better directionality, indetectability, and range precision than a conventional electromagnetic wave radar (see, e.g., [23]).

The laser radar meets the needs for long distance range sensing. For indoor range sensing, where distance resolution often requires less than 1 cm, the laser radar is costly to build due to the required high precision timing electronics and the short elapsing time between the transmitted and the return signal (may be less than .1 nS). Accuracy of the laser radar depends on the accuracy of electronics to tell the time difference between the transmitted and the return signals. For a circuit that can only tell 1 nS difference, the range uncertainty would be 1 nS x 3x10(8) m/S or 30 cm. To achieve better resolution, expensive equipment is needed.

2.3.4 Plane-of-light Range Sensing

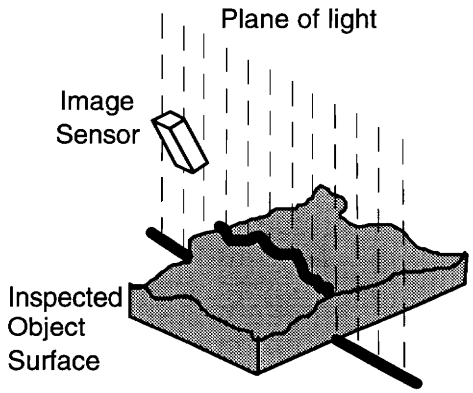


Figure 2.4 A typical plane-of-light range sensing structure.

The most frequently used technique is an architecture consisting of a plane of light and a video camera. Figure 2.4 illustrates the imaging geometry. The mathematics behind

this scheme is basically the principle of triangulation, formed by the plane of light and the stripe-of-light image observed by the camera. What occurs with this method inherently, however, are problems such as occlusion and shadowing.

Marc Rioux proposed a laser range finder based on synchronized scanners[24] in 1984, which is implemented in a planar form. Two scanners in synchronization allow a linear position sensor to be used for surface topography measurement. This laser range finder employs the principle of triangulation and uses a lateral effect photodiode to generate range information. This type of configuration promises high speed generation of range data due to the frequency response characteristic of the lateral effect photodiode. The lateral effect photodiode can also produce intensity images that are in a perfect registration with the 3-D range images The experimental results show a response time shorter than a microsecond using a lateral effect photodiode as a position sensor. Figure 2.5 illustrates the imaging geometry.

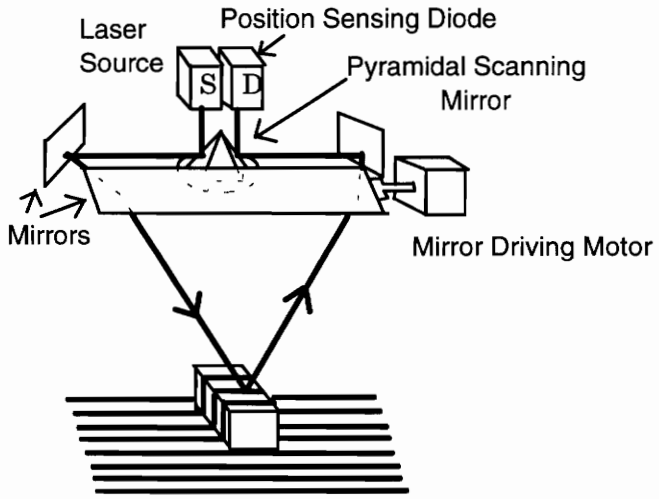


Figure 2.5 Synchronized triangulation based range finder[24].

An interesting range sensing technique, proposed by M. Rioux and F. Blais in 1986[25], depicts a modified architecture of flying spot range sensing. This range sensor consists basically of a mask with two apertures, a camera lens, and a standard solid state

compact CCD video camera. In a practical implementation, the double aperture mask replaces the iris of a standard camera lens. This architecture provides a very compact 3-D range sensor, ideal for robotics applications. Figure 2.6 shows the basic principle of this device.

The basic principle of this approach is as follows. Assume the camera has been adjusted such that a point A on the reference plane is focused on the CCD photodetector at point A'. Point B on an object located farther than the reference plane will be focused at B' in front of the photodetector. Because of the double aperture mask, two points b' will be measured on the CCD. The distance between these two points is a function of the object distance to the camera (z-coordinate) while the geometrical center gives the x- and y- coordinates. The speed of acquisition is mainly limited by the integration time of the CCD, the number of simultaneous projected lines, and the image processing system.

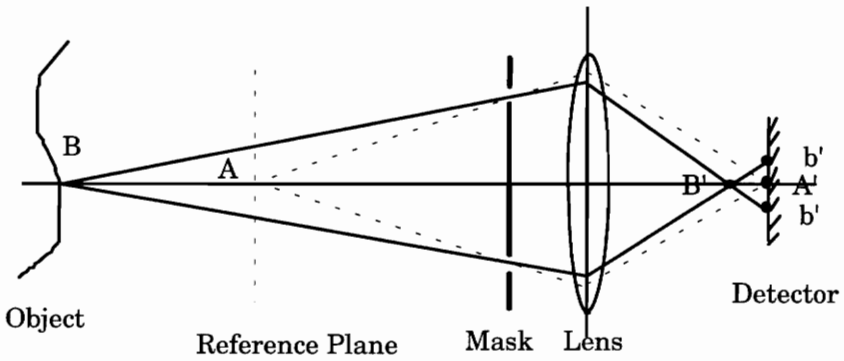


Figure 2.6 Compact planar range sensor[25].

For high speed processing, the double aperture mask limits the amount of light coming into the camera module. The light intensity issue is very important if speed is of concern. In the lumber inspection application, operating speed is critical. If this technique is to be realized, the intensity of the laser beam must be so intense that it can endanger the inspected objects.

2.4 Summary

The variety of features that must be detected for the various primary and secondary manufacturing of hardwood products necessitates the use of multiple sensors. Previous research in automatic inspection systems for hardwood processing indicates the need for both color and range sensing technologies. Very little research has been done in the area of developing a range sensing technology that can be integrated with other sensory information and, hence, this is the focus of this study.

The study of the processes involved in primary and secondary manufacturing of hardwood products provides us with views of the operation environment for the installation of range sensors. In general, the range sensor should be chosen so that it fits well into the overall milling system. The other point of emphasis is speed. The speed of the sensor has to be consistent with the cutting process. Based on considerations of operating environment and a review of available range-sensing techniques, the nature of a plane-of-light triangulation technique appears to be best suited for forest products applications. This technique provides a good structure for objects that are long in length but short in width. The resolution performance of the plane-of-light technique is very good. The plane-of-light technique is an inexpensive technology that can perform range sensing operations in real-time. The research reported in the subsequent chapters describes the development of a flexible laser-based prototype ranging system utilizing the plane-of-light technique. The design features will allow it to be integrated with information from other sensing modalities.

Chapter 3
System Overview

The discussion of ranging techniques presented in Chapter 2 indicates that, to acquire profiling information of industrial objects, the plane-of-light triangulation method is clearly the method that provides the most flexibility and the greatest speed of operation. In this chapter the discussion focuses on the implementation aspects of the triangulation method.

A profiling sensor is often referred to as a range sensor, or range finder, in the literature (see, e.g., [18]). In this thesis the term range sensor will be used in most of the discussion instead of profiling sensor. Range, depth, and thickness are used interchangeably depending on applications. For instance, the term thickness profile is better suited for applications involving the profiling of thin and flat objects such as lumber. A range image, which is a gray scale image with pixel intensities representing distances of the object surfaces from the viewer, is sometimes called a depth map. Every row (or column) of a range image can also be treated as a thickness profile with gray scale intensities providing heights.

This chapter discusses several important aspects of image formation. In the discussion we will find that the triangulation method can perform large depth-of-field range sensing. This capability is a unique feature of the plane-of-light triangulation method over the other range sensing methods. The theory of geometric data correction will also be discussed. Data correction is essential when performing large depth-of-field

range sensing. Signal processing schemes for range and intensity generation are also explained. The theoretical aspects of the algorithm employed for extracting range information are contained in Chapter 6. The hardware implementation of both the range and intensity algorithms are described in Chapter 4.

3.2 General Introduction

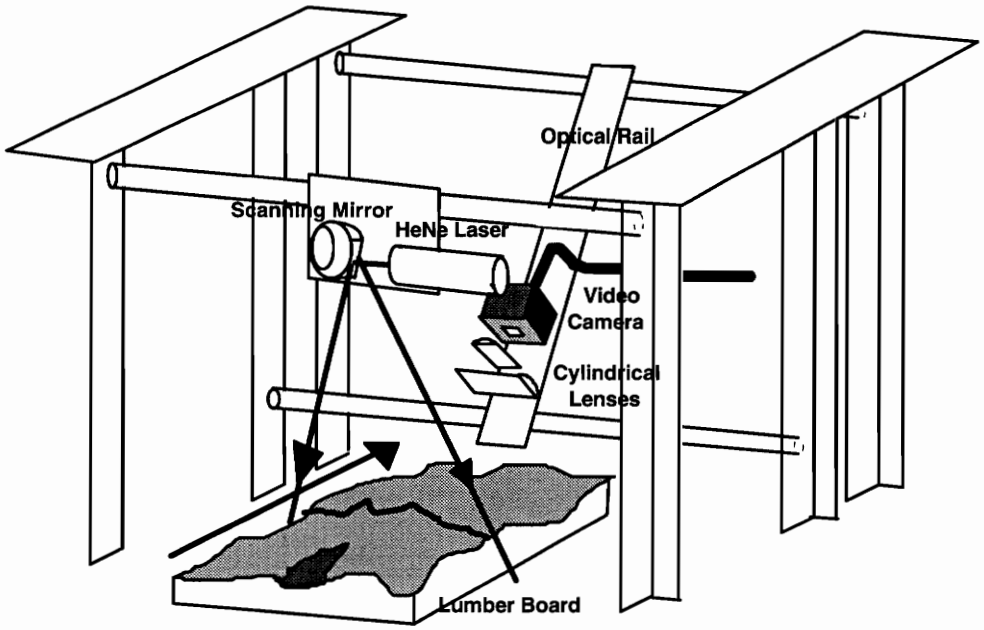


Figure 3.1 Imaging geometry of the range sensing system.

The imaging geometry used by the range sensing prototype is illustrated in Figure 3.1. In this geometry, the video camera is the agent used for retrieving profile information. The triangulation geometry is formed by the plane of light and the video camera used in the prototype. The beam of laser light shines on a fast spinning multi-facet mirror to create a plane of light. This plane of light illuminates the object to be scanned for range and intensity profiles. The video camera is laterally offset at an angle to view the stripe-of-light image of the cross-section profile "intercepted" by the plane of light. The

shape of the stripe-of-light image is therefore a direct measure of a thickness profile of an object.

To obtain thickness profiles of the whole object, a materials handling system carries the object at a constant speed while the video camera continuously captures the stripe-of-light images. A dedicated range sensing hardware system receives these contiguous images and converts them into thickness and intensity profiles at the camera's frame rate. These thickness and intensity profiles are then buffered in First-In-First-Out (FIFO) memory to be transferred to the computer. Figure 3.2 illustrates the data flow of the data processing.

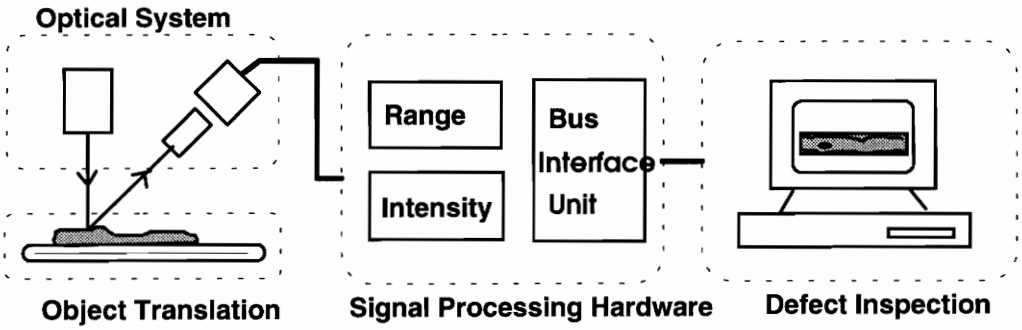


Figure 3.2 System configuration.

The profile sampling rate is the same as the camera's frame rate. This rate directly affects the speed at which the material handling system can operate. For a higher frame rate camera the profile sampling rate is higher. The material handling system can therefore carry objects at a faster speed. Currently two kinds of video cameras can be used on the ranging prototype. A RS170 camera operates at 30 frames per second, while MC9128 camera can operate at maximum of 380 frames per second. The maximum profile sampling rates are therefore 30 and 380 thickness and intensity profiles per second respectively.

There are several parameters pertaining to a range sensor that indicate the system performance. They are profile sampling rate, range and intensity data rate, dynamic range of depth measurement, measurement resolution, number of pixels in each profile line, and the viewing width of a profile line. For a range sensor using the implementation scheme mentioned above, the profile sampling rate is equal to the camera's frame rate, and the number of pixels in a profile line is equal to the number of video lines in a video frame. As a result, the range and intensity data rate is equal to the product of the profile sampling rate and the number of pixels in each profile line. The video camera used determines each of these parameters.

As to the dynamic range of depth measurement, measurement resolution, and the viewing width of a profile line, the optics plays an important role. To begin with, the optical system has to provide a plane of light with uniform beam intensity along the profiling direction. Having a uniform light source along the profile line markedly reduces hardware complexity, e.g., shading correction is not necessary. Secondly, the imaging lens on the video camera should allow a good deal of flexibility for adjusting the dynamic range of depth measurement. The dynamic range of depth measurement, in its literal meaning, represents the range of depth measurements required of the range sensor. For a video camera equipped with a regular lens, the viewing width is the same as the viewing height. For the geometry adopted in this project, this in turn means that the dynamic range is as wide as the profile viewing width. This is bad because the video camera has a limited number of pixels to provide thickness detection and when the dynamic range is so wide, the "fine" variation of thickness is lost.

Section 3.2.3 contains the design of the imaging lens module. Two cylindrical lenses are involved in forming images on the image sensor. The greatest benefit in using this lens configuration is that we can flexibly choose the dynamic range of depth

measurement without much concern about the settings of the physical viewing width of the profile line. The depth measurement resolution is therefore equal to the ratio of the dynamic range to the number of imaging pixels of each video line.

The following describes the design of the optical system and the hardware range detection scheme. Detailed descriptions of the hardware implementation can be found in Chapter 4.

3.2 Optical System Design

Two components constitute the optics of the triangulation range sensing method: the plane-of-light generating mechanism and the video camera. In the next section the mathematical formulation of triangulation method is presented. The geometry shown in Figure 3.3 corresponds to the side view of Figure 3.1. Note that the image sensor is tilted at an angle from the optical axis. The theory will show that by changing the way the image sensor is aligned with the optical axis, the triangulation range sensing method can increase the dynamic range of depth measurement. The theory also presents a new calibration method for the large depth-of-field range sensing applications.

3.2.1 Triangulation Method

Figure 3.3 illustrates the geometry of triangulation range sensing method in a planar form. A laser emits a beam to illuminate objects in its path. Points A and B indicate the positions where the laser beam encounters different objects. An image focusing module collects the light reflected from these points and forms images on the

image sensor. The positions where images of points A and B locate on the image sensor indicate the distance values of A and B from the camera module.

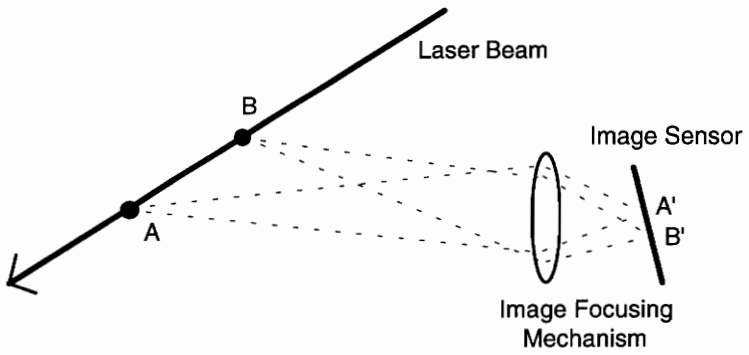


Figure 3.3 Triangulation method.

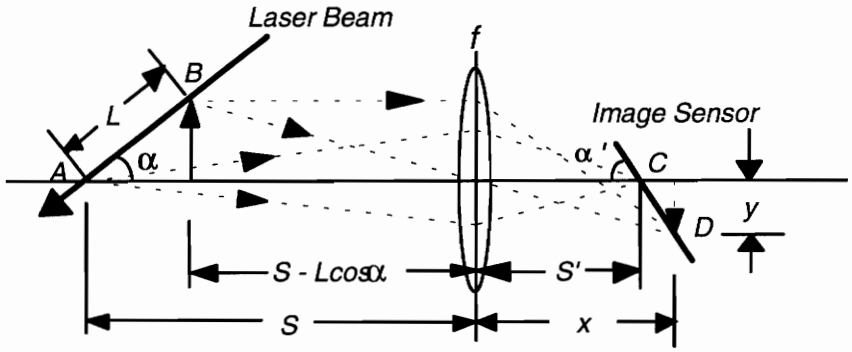


Figure 3.4 Geometry of image formation.

Figure 3.4 illustrates the geometry of image formation. This figure has the same structure as in Figure 3.3. Assume the laser beam intercepts the optical axis at point A and the virtual image of A is located at point C (on the optical axis). According to the law of image focusing, we can establish the following relationships.

$$\frac{1}{f}=\frac{1}{S}+\frac{1}{S'} \tag{3.2.1.1}$$

$$\frac{1}{f}=\frac{1}{S-L\cos\alpha}+\frac{1}{x} \tag{3.2.1.2}$$

where L is the distance of the point B to point A and α is camera's viewing angle. Solving for S' and x, we get

$$S' = \frac{fS}{S - f} \tag{3.2.1.3}$$

and

$$x = \frac{f(S - L\cos\alpha)}{S - L\cos\alpha - f} \tag{3.2.1.4}$$

The object point at B and its image D comply with the law of image magnification, which states

$$\frac{y}{L\sin\alpha} = \frac{x}{S - L\cos\alpha} \tag{3.2.1.5}$$

Substitute x using Equation (3.2.1.4), we can get the value of y as

$$y = \frac{fL\sin\alpha}{S - L\cos\alpha - f} \tag{3.2.1.6}$$

To find out the slope of the virtual line image, we find that x-S' is represented as

$$\begin{aligned} x - S' &= \frac{f(S - L\cos\alpha)}{S - L\cos\alpha - f} - \frac{fS}{S - f} \\ &= \frac{f^2 L\cos\alpha}{(S - f)(S - L\cos\alpha - f)} \end{aligned} \tag{3.2.1.7}$$

The slope of the line segment CD therefore is

$$\begin{aligned} \tan\alpha' &= \frac{y}{x - S'} = \frac{fL\sin\alpha}{S - L\cos\alpha - f} \times \frac{(S - f)(S - L\cos\alpha - f)}{f^2 L\cos\alpha} \\ &= \frac{(S - f)}{f}\tan\alpha \\ &= Const. \end{aligned} \tag{3.2.1.8}$$

As shown in the equation, the slope is a constant since S, f and the viewing angle α are fixed values. In other words, the slope is independent of L. Therefore the virtual image of the laser beam is a straight line.

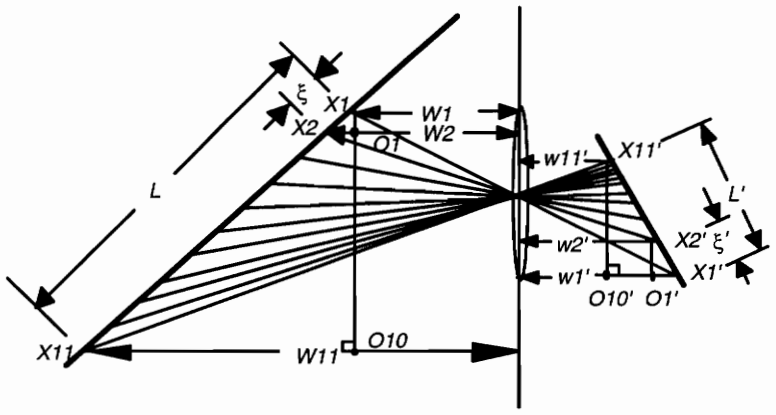


Figure 3.5 Depth measurement geometry.

While it is possible to calculate the distances of illuminated points by measuring the viewing angle of the camera and the image positions of these points, this method is not recommended. Figure 3.5 shows a calibration geometry that requires only three physical measurements of distances. Assume that the object points X_1, X_2 and X_{11} lie on the plane formed by the laser beam, and their images are perfectly focused on the image sensor at X_1', X_2' and X_{11}' respectively through an imaging lens with focal length f. The following equations describe the relationships of these object points and their image positions.

$$\frac{1}{f} = \frac{1}{W_1} + \frac{1}{W_1'} \tag{3.2.1.9}$$

$$\frac{1}{f} = \frac{1}{W_{11}} + \frac{1}{W_{11}'} \tag{3.2.1.10}$$

$$\frac{1}{f} = \frac{1}{W_2} + \frac{1}{W_2'} \tag{3.2.1.11}$$

Assume three distance measurements are made on W_1 (the near end of the dynamic

range of depth measurement), W_{11} (the far end of the dynamic range), and L (the physical distance between object points X_1 and X_{11}). Three position recordings are performed on images of these object points: X_1' (the imaged position of X_1), X_{11}' (the imaged position of X_{11}), and L' (the imaged distance between X_1' and X_{11}'). The measurements of X_1', X_{11}', and L' can be done by invoking a frame grabbing facility of the image sensor, and therefore are given the unit of video pixels.

The triangles $\Delta\ X_1X_{11}O_{10}$, $\Delta\ X_1X_2O_1$, $\Delta\ X_1'X_2'O_1'$, and $\Delta\ X_1'X_{11}'O_{10}'$ in Figure 3.5 establish the following equations.

$$\frac{\xi}{L}=\frac{W_2-W_1}{W_{11}-W_1} \tag{3.2.1.12}$$

$$\frac{\xi'}{L'}=\frac{W_2'-W_1'}{W_{11}'-W_1'} \tag{3.2.1.13}$$

where ξ is an arbitrary increment starting at X_1 along the laser beam, and ξ' is the corresponding image segment. The relationships of Ws and their image positions are described in Equation (3.2.1.9) to (3.2.1.11). Replacements of W's lead to the following equation.

$$\frac{\xi'}{L'}=\frac{W_2'-W_1'}{W_{11}'-W_1'}$$

$$=\frac{W_2-W_1}{W_{11}-W_1}\frac{W_{11}-f}{W_2-f}=\frac{W_{11}-f}{W_2-f}\frac{\xi}{L} \tag{3.2.1.14}$$

The transformation of an arbitrary increment ξ to its image segment ξ' is thus established as

$$\xi'=\frac{W_{11}-f}{W_2-f}\frac{\xi L'}{L} \tag{3.2.1.15}$$

where

$$W_2=\frac{\xi(W_{11}-W_1)}{L}+W_1 \tag{3.2.1.16}$$

As explained earlier, W_1, W_{11}, L are measured quantities and f is a known focal length of the imaging lens. The length of ξ' has the same metric unit as L', which is

measured through the frame grabbing facility of the image sensor. Through computations using Equation (3.2.1.15) and (3.2.1.16), we can build a transformation table of the physical depth values from pixel coordinates of the recorded image points. The distance measurement based on the triangulation method is therefore complete.

There is an important implication in the above derived equations: an almost unlimited potential in the size of the dynamic range of depth measurement. If the image sensor is positioned nicely osculating the straight line of laser image, the range sensor can perform a very wide range of depth sensing.

3.2.2 Laser Scanning Mechanism

Several techniques are available for creating a plane of light. Among these is the scanning laser beam technique. It is the one most frequently used. The advantage of using a laser beam is its small spot size and its superb low beam divergence. An alternate way of creating a plane of light is using optical fiber bundles. The light is generated by a powerful light source such as a tungsten halogen bulb and is focused on and guided into a fiber bundle. The other end of these optical fibers is spread out in a long stripe to create the line of light. The outcoming light is then focused by a cylindrical lens to create a light plane. The disadvantage of this scheme is that the light plane has a shallow depth-of-field due to the use of a focusing lens.

Another way of creating a light plane is by projecting a single laser beam through a cylindrical lens with a short focal length. Unlike regular lenses that focus images in both dimensions, the cylindrical lens functions one-dimensionally. In other words, the light focusing phenomenon only appears in one of the two axes that form the surface of the cylindrical lens. When a light beam passes through the cylindrical lens, the beam

converges in the focusing axis while maintaining the beam characteristic in the other. Due to the nature of the laser beam, this plane of light is not uniform. It has an intensity profile of a Gaussian curve that is intense in the middle and trails off toward the edges.

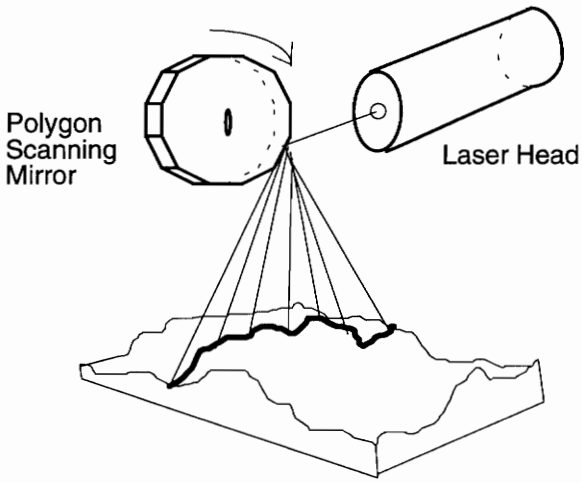


Figure 3.6 Plane-of-light optical component.

The scheme employed in this project utilizes a laser beam and a spinning multi-facet mirror. Figure 3.6 illustrates this optical component. A HeNe laser head with a small diameter beam and small beam divergence emits a beam of light (632.8 nm) that shines on a fast spinning polygon mirror. The resulting light pattern, in strict sense, is not a plane of light. Instead it is a flying laser beam. The fact that it creates the effect of a plane of light is as follows. An array camera integrates light signals during the period of a frame exposure. If the flying laser beam can scan across the scene several times during a frame exposure period, this flying beam can be deemed a plane of light. If the spinning mirror is synchronized with the camera frame rate so that only a fixed number of laser lines scan through the scene during each frame exposure period, the plane of light is almost perfectly uniform. Even without synchronization, the plane of light can be made uniform by having the laser beam sweep rate be much higher than the camera frame

integration period. For example, if the laser sweep rate is 20-21 lines per frame, the non-uniform factor is only 1/20 of the average intensity.

There are several advantages in using a multifaceted mirror to create the plane of light. Section 8.1.1 will explain a modification that reduces the laser beam size to improve the measurement accuracy (the theory is described in Chapter 6). Simply by placing a proper lens in front of the laser head, the laser beam size can be substantially reduced without changing other settings. This in turn means that we can have a very "thin" sheet of light. Section 8.5 also presents a "light flashing" mechanism for applications where objects are moving at high speed while the camera frame rate is low. By using this mechanism, the image blur caused by the moving object can be reduced. This also allows low frame rate cameras to perform depth sensing on fast moving objects.

3.2.3 Imaging Lens System

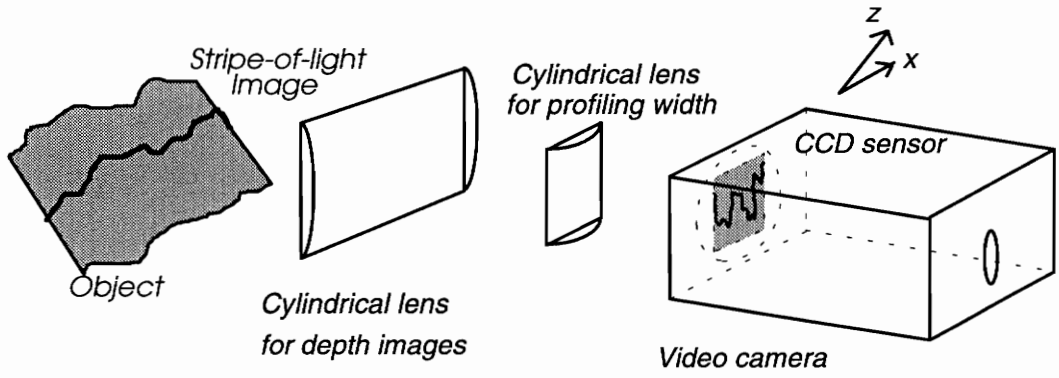
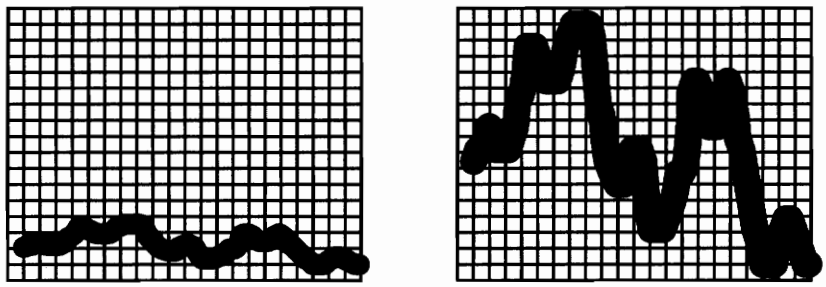


Figure 3.7 Imaging lens system.

Figure 3.7 illustrates the design of the imaging system. The imaging system utilizes two cylindrical lenses for non-isotropic image magnification. As shown in the figure, the cylindrical lens for depth images (thickness variation) stretches the image of the required range of depth measurement onto the vertical axis of the image sensor. The

cylindrical lens for viewing width focuses the image of the profile line onto the horizontal axis of the image sensor.

Figure 3.8 illustrates the design concept of the non-isotropic image magnification. Figure 3.8.(a) shows an example of the camera's view when a regular camera lens is used. The width of this figure covers the required profiling width. Depth variation is, in general, far smaller than the required profiling width. As shown in this figure, the image of depth variation only spans few rows and leaving most of the pixels unused. Detection of range positions is achieved by identifying skeleton positions of the stripe-of-light image. The a fewer rows that the depth image can cover means the fewer range positions available for depth measurement. It also means poorer measurement resolution.



a). Image on the CCD sensor using a regular lens

b). Image on the CCD sensor using cylindrical lenses

Figure 3.8 Image focusing concept.

The design goal of the imaging lens system is to increase the photocell utilization of the image sensor and therefore enhance measurement resolution. Figure 3.8.(b) illustrates the ideal image of the same scene. As shown in the figure, the thickness variation is "magnified" while the width of this stripe-of-light thickness profile stays the same. A properly chosen cylindrical lens (for depth image) can stretch the desired dynamic range of depth measurement to the image sensor's full length, i.e., the length of an entire video line. We can therefore obtain full utilization of all imaging pixels of the image sensor.

In choosing cylindrical lenses to be used, it should be noted that the selections of two lenses can be done independently. While one cylindrical lens focuses the required dynamic range of depth measurement to the vertical axis, the other cylindrical lens focuses the width of the profile line to the horizontal axis. In other words, we can flexibly change the image magnification of the vertical axis while maintaining the profiling width of the horizontal axis, and vice versa. The video camera is assumed to perform serial data readout one video line after another, and is positioned such that each video line contains at most one bright spot (the video line is therefore the vertical axis of the image sensor). The reason for using camera's lines to make range measurements is to facilitate the hardware processing. Using this method, one range datum can be generated for every outcoming video line without having to wait for the whole frame to finish as would be required if the columns were used to generate range information.

3.2.4 Guidelines for Selecting Cylindrical Lenses

The selection of each cylindrical lens is based on the law of lens focusing. Figure 3.9 illustrates the geometrical relationships of these parameters. Values of the dynamic range of depth measurement (L), profiling width (W), camera's viewing angle (α), working distance of the range sensor (S), and the physical size of the image sensor (P_1 for horizontal axis, P_2 for vertical axis) are required in the derivation. The figure shows the optical geometry of the non-isotropic image magnification in both camera's viewing axes. Figure 3.9.(A) illustrates the focusing geometry for obtaining the desired imaging width and Figure 3.9.(B) shows the geometry for obtaining depth image focusing.

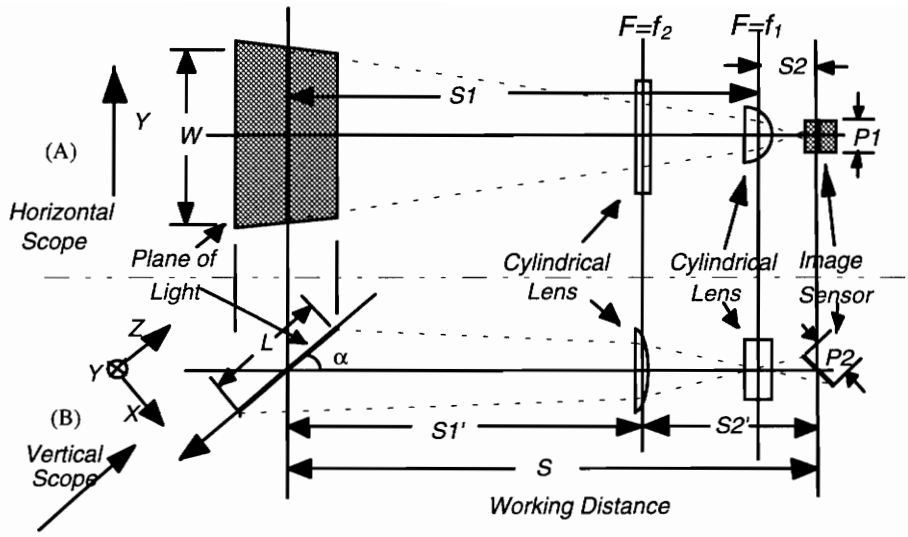


Figure 3.9 Cylindrical lens focusing.

To find the focal length to be used for focusing the profiling width W, we assume that the profiling width is defined as the center width of the sensing window (shown as the shaded area in Figure 3.9.(A). This window is the observation window of the plane of light perceivable by the image sensor at the working distance S. This window is not rectangular. Assume the size of the horizontal axis of the image sensor is P_1, we can establish the following equations

$$\frac{1}{f_1} = \frac{1}{S_1} + \frac{1}{S_2} \tag{3.2.4.1}$$

$$M_1 = \frac{\text{Object Size}}{\text{Image Size}} = \frac{W}{P_1} = \frac{S_1}{S_2} = \frac{S_1}{f_1} - 1 \tag{3.2.4.2}$$

where M_1 is the image magnification factor for horizontal focusing and f_1 is the focal length to be used to obtain the whole thickness profile. M_1 is a known value in that W and P_1 is known. As a result we can obtain the focal length to be used ($S=S_1+S_2$ is the working distance of the range sensor).

$$f_1 = \frac{SM_1}{(M_1+1)^2} \tag{3.2.4.3}$$

Equation (3.2.4.3) depicts the rule of choosing the proper focal length for horizontal focusing. To calculate the focal length for vertical focusing, the exact solution may be very complicated. Equation (3.2.4.4) illustrates the complexity of this equation.

$$P_2 \cos\alpha' = \frac{f_2^{\,2} L\cos\alpha}{S - f_2}\left(\frac{1}{S - L\cos\alpha - f_2} + \frac{1}{S + L\cos\alpha - f_2}\right) \qquad (3.2.4.4)$$

where $\tan\alpha'=\tan\alpha\times(S-f_2)/f_2$ as depicted in Equation (3.2.1.8). This equation comes from Equation (3.2.1.7). The dynamic range of depth sensing is assumed to be $(-L,+L)$ at the working distance S. P_2 is the size of the vertical axis of the image sensor as in Figure 3.9. As shown in the equation, f_2 does not have an explicit solution. A simplified derivation of the focal length f_2 is, therefore, desirable.

To simplify the calculation, we ignore the tilting factor of the image sensor and use a derivation similar to the one used for f_1. To begin with, we have the image focusing equation

$$\frac{1}{f_2} = \frac{1}{S_1'} + \frac{1}{S_2'} \qquad (3.2.4.5)$$

where f_2 is the focal length of the cylindrical lens for focusing the dynamic range of depth measurement. Likewise, we can express the image magnification factor in the vertical direction as

$$M_2 = \frac{L\sin\alpha}{p_2} = \frac{S_1'}{f_2} - 1 \qquad (3.2.4.6)$$

where L is the desired depth sensing range and p_2 is the size of the vertical axis of the image sensor. M_2 is also a known value. We therefore can get f_2 as

$$f_2 = \frac{SM_2}{(M_2+1)^2} \qquad (3.2.4.7)$$

Equation (3.2.4.7) represents a simplified solution for obtaining the focal length f_2.

This approximation does not deviate from the true value if the sensing range L is small. By using Equation (3.2.4.3) and (3.2.4.7), we can select cylindrical lenses to give the highest possible spatial resolution across the width of the imaging field and the highest possible range resolution.

3.3 Signal Processing Schemes for Range and Intensity Values

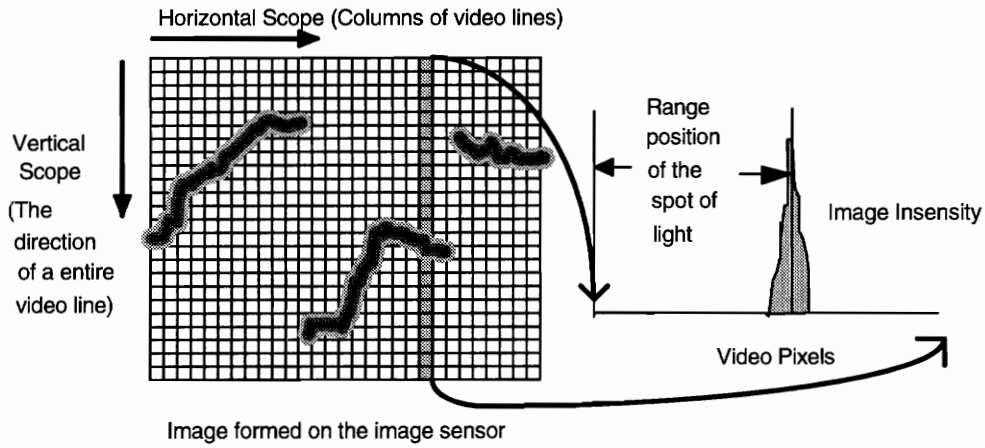


Figure 3.10 Signal processing scheme for range values.

Figure 3.10 illustrates how a thickness profile cab be obtained from an image recorded by the image sensor. As shown in the figure, the goal is to find the "skeleton" of the stripe-of-light image. The resulting profile curve is therefore an explicit function of the profiling axis, and the height of the curve at a particular point (in this case, the pixel position of the curve) is directly related to the thickness, or range, of that location. To obtain this information, we need an algorithm that can produce the "best" skeleton curves. As shown in the figure, every video line contains a segment of bright pixels. The task, then, is to find the pixel position within the video line that best represents the position of this segment of bright pixels.

Most of the range sensors employing the plane-of-light technique use either peak or centroid detection to give the skeleton required (see, e.g. [26]). Centroid detection is inherent in systems using position sensing diodes, and therefore is not considered in this project. For digital systems the peak detection is popular in that it is fairly simple to implement in hardware. Nevertheless, it has several drawbacks. As is well known image sensors have problems with signal saturation. If this happens in collecting light images, there will be large areas of saturated intensities on every video line. The peak algorithm fails in this situation. The peak algorithm also fails when there is more than one peak in a video line. Highly irregular surface reflectance and electronic noise can produce multiple peaks.

In software processing, a popular method for obtaining skeletons uses a morphological thinning operator (see, e.g. [27]). A one-dimensional thinning operator (in the direction of a video line) is implemented in the hardware prototype. We call this algorithm the Center-Pixel Detection Algorithm. This detection scheme looks for the segment of bright pixels in a video line by using a binary thresholding technique. The purpose of this threshold is to identify pixels of a video line illuminated by the laser beam. This threshold value is set slightly above the dark level of the background. This segment is a string of contiguous pixels. Once this string of pixels is found, the algorithm then searches for the position of the center pixel within this contiguous string of pixels. This position is considered to define the range point for this line of video imagery.

Figure 3.11 shows a comparison of the peak and Center-Pixel detection algorithms. As shown in the figure, the peak position will cause errors because it is highly dependent on the distribution pattern of the pixel intensities. It is believed that this dependency on an object's surface reflectance characteristics should be removed from profile (range) generation. In experiments that have been conducted, the Center-Pixel

detection algorithm has proven to be more accurate in generating range positions than the peak position algorithm. The cross section of a beam of laser light is a Gaussian curve. When the beam shines on an object, the perceived image can have a pattern as in Figure 3.11. By thresholding this pattern using a very low threshold value, we can identify the area where the spot of laser light strikes the video line. Experiments suggest that the center pixel algorithm provides a better estimate to the actual location of the range position. Based on these experimental results, we conclude that the Center Pixel algorithm is the better scheme for locating range positions.

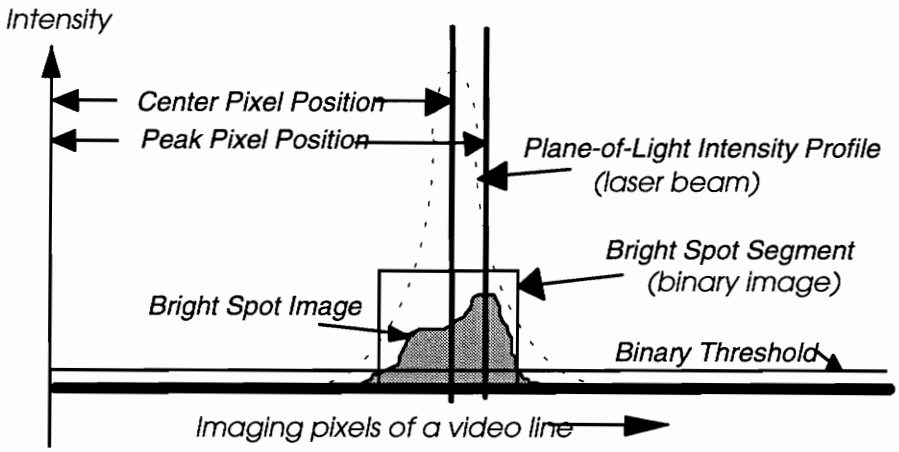


Figure 3.11 Peak and Center-Pixel Detection Algorithms.

Section 6.1 describes the theoretical basis for this processing scheme. This scheme is also robust against image blur. Section 6.3.1 describes this situation. The Center-Pixel algorithm takes advantage of the readout format of the video camera. In general, a video camera outputs image signals in a serial manner. Pixel intensities in a video line are sent out one after another continuously. A one-dimensional thinning operator can be implemented in hardware to take in this serial string of pixels representing each line of video data. The output of this thinning operator is the center pixel position of the

illuminated area of the video line. Detailed description of the hardware implementation is contained in Section 4.2.3.

The range sensor not only generates a profile image, it also creates a black-and-white image of the object as well. This black and white image is perfectly registered with the profile or range image. Two processing units are implemented in the prototype. Figure 3.12 illustrates the processing methods to accomplish this. The Center-Pixel processing unit determines the "range point" along a video line. The Total-Reflectance processing unit determines the intensity value by summing the intensities of all elements in the video line that lie above the previously described binary threshold. Doing so, one video line produces one range datum and one intensity datum (in the manner of Center-Pixel intensity or Total-Reflectance intensity). In either case both the range image and intensity image are perfectly registered in space.

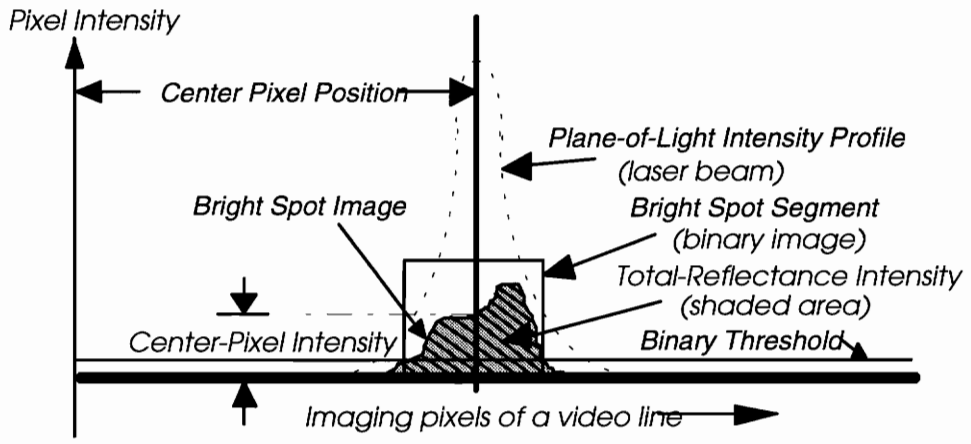


Figure 3.12 Intensity processing schemes.

3.4 Summary

This chapter described the triangulation optics devised for use in this range/intensity imaging prototype. The derivations presented an important capability of

the triangulation method: large depth-of-field range sensing. The large depth-of-field range sensing capability surpasses other techniques used for range sensing. Using the signal processing scheme presented in this chapter, the measurement accuracy is improved.

Section 3.2.1 presented the theory of triangulation for recovering range information recorded by the image sensor. As proven in this section, the virtual image of the laser beam is a straight line. An image sensor placed on this straight line can therefore perceive clear images without concerning the depth-of-field constraint of a regular lens. This implies a capability of large depth-of-field range sensing far beyond the depth-of-field limit of a regular camera head. The theory also presents a depth correction scheme as a result of the large depth-of-field range sensing.

Section 3.2.2 described a mechanism for creating a plane of light. There are several merits to this scheme. First, it preserves the optical quality of a laser beam. Second, it allows room for improving the quality of the laser beam. Section 8.1.1 provides a method for reducing the laser beam width. This is essential for improving the measurement accuracy, as the theory in Section 6.1 describes. Section 8.5 also presents a mechanism that allows a low frame rate camera to perform profile sensing on fast moving objects. All of these require slight modifications of the existing laser scanning mechanism.

Section 3.2.3 contains the imaging system design. Two cylindrical lenses provides a great deal of flexibility to the range sensing system. The goal is to provide full utilization of the image sensor even when the required profiling width is large while the height variation is small. Section 3.2.4 provides the guidelines for choosing the cylindrical lenses to be used to fulfill application needs.

Section 3.3 explained the signal processing schemes used in the prototype system. The method used for detecting range positions is the Center-Pixel algorithm. It is, in

essence, a one-dimensional thinning operator implemented in hardware. The methods for generating intensity images are also included. Two units are implemented. The Center-Pixel intensity processing unit produces center pixel intensities along with range values. The Total-Reflectance intensity processing unit produces total reflectance intensities of the corresponding range pixels. The prototype can therefore operate as a combination of a black-and-white video camera and a profiling sensor, where registered image pairs of range and intensity are required.

Chapter 4
Range Sensing Hardware

This chapter describes the design of the range sensing hardware for producing range and intensity data. Several features have been incorporated into this hardware system. These features are: (1) a simplified system control that uses only a few I/O ports, (2) a frame grabbing capability, (3) pipelined processing units for range and intensity determination, (4) a memory management scheme that facilitates real-time processing, (5) a sophisticated system diagnostic capability for pin-pointing hardware errors, (6) real-time Look-Up-Table mapping image pixel location into range data, (7) data overflow error reporting capability, (8) a capability to support two different black and white camera systems, and (9) a multiple-mode hardware operation. The design provides a good deal of software control over the system. Detailed description of hardware interconnection is therefore not included due to the signal connection complexity. Nevertheless, the functionality of the above mentioned features is explained in some detail.

4.1 General Operating Procedures

The prototype range sensing hardware consists primarily of six modules: (1) two image digitization units, (2) a frame grabbing unit, (3) a range and two intensity processing units, (4) a MicroChannel control unit, (5) a Look-Up-Table unit, and (6) a memory buffer unit. Figure 4.1 shows the system configuration of the range sensing

hardware. Currently the system is built on the MicroChannel Architecture. Future migration to other bus interface standards can be achieved by redesigning the Interface Control Unit. Three processing units constitute the heart of the system. These processing units are the Range, Center-Pixel Intensity, and Total Reflectance Processing Units. The Range Processing Unit employs the Center-Pixel range detection algorithm as described in Section 3.3. It is the basic processing unit for generating range data. The method for generating intensity data, however, is a user's option. The user can choose to ignore the intensity data during Direct Memory Access (DMA) transfers or generate intensity data using either Center Pixel intensity or Total Reflectance value described in Section 3.3.

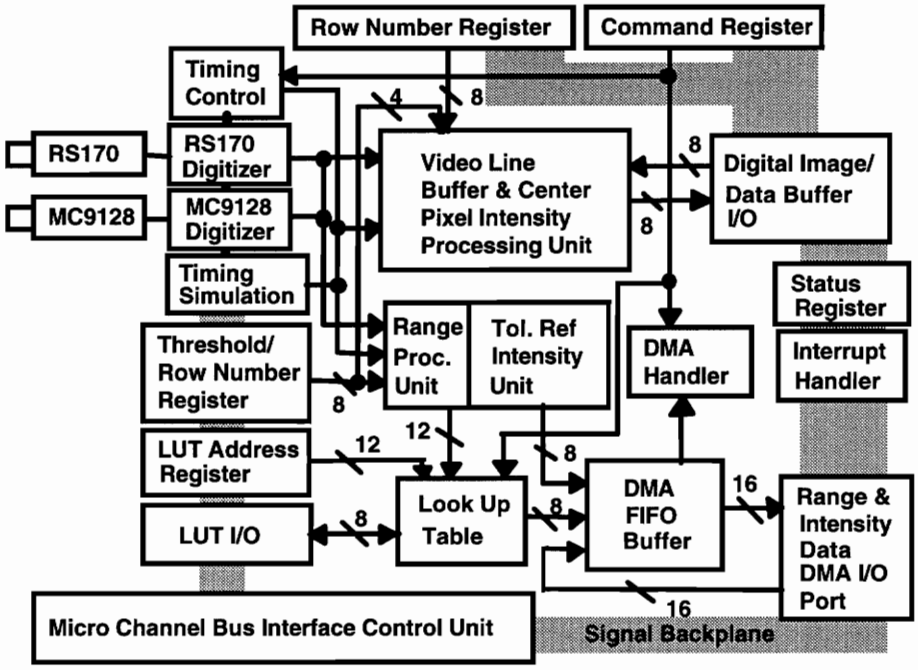


Figure 4.1 System Configuration of The Range Sensing Hardware.

From the host computer's viewpoint, this system is a module with one DMA channel, one interrupt level, and a set of I/O ports (thirteen contiguous I/O addresses). A MicroChannel configuration utility program allocates a DMA channel, an interrupt level and a I/O port address space for the hardware system during MicroChannel initialization.

The host computer then manipulates this allocated set of I/O ports for hardware control. Chapter 5 explains in detail how these I/O ports should be manipulated by the software. Thirteen I/O ports control the system behavior. These I/O registers are Command Register, Digital Image/Data Buffer, Row Number Register, Threshold/Row Number Register, DMA I/O port, Look Up Table (LUT) Address Register, LUT I/O port, Timing Simulation Register, Status Register, Camera/Intensity Processor Select Register, and Video Frame Subsampling Factor Register.

The range sensing hardware operates in one of three modes: Frame Grabbing Mode, Range Sensing Mode, or Computer Access Mode. The mode selection is achieved by issuing commands to the Command Register (the base address of the allocated I/O space). In the following section general operating procedures for performing range sensing are described. Detailed descriptions of individual units are contained in later sections.

4.1.1 Optical Adjustment and System Initialization

The first step in system initialization is to ensure proper optical settings, i.e., dynamic range of depth measurement, working distance, viewing angle, viewing width of profile lines, and good image focusing. Optical setup procedures should follow the adjustment guidelines described in Chapter 3. These optical adjustments are performed via operating the range sensing hardware in the Frame Grabbing Mode that allows the system to capture video frames recorded by the video camera. These images can then be displayed on the host computer. Manual adjustments of optical components enable the user to select proper imaging lenses, to assure good focus, and to assure proper viewing conditions.

Two video digitizers are available on the hardware system; one is for digitizing from an RS170 camera (480 video lines for the implemented digitizer) and the other is for digitizing for an MC9128 camera (128X128 pixels). The user can choose either one as the source of image signals by programming the Camera Select Register (bit 0,1). If a RS170 camera is chosen, the system will generate 30 profile lines per second and 480 range and intensity pixels per line. The range value can be one of 384 levels because the implemented RS170 digitizer can only digitize 384 pixels of a video line (see [21] for more details). Section 4.2.1 explains more about this. If the MC9128 camera is chosen, the system can have 380 profile lines per second and 128 range and intensity pixels per line. The range value can be one of 128 possible levels. These range data are converted to a 8-bit data format via the Look-Up-Table mapping.

Besides optical adjustments, the host computer has to perform several important tasks before activating a range scanning process. As explained in Chapter 3, the range detection algorithm is performed by detecting center positions of bright segments on column images (equivalent of one-dimensional thinning). It needs a threshold value to determine whether the outcoming image signal is bright (laser illuminated) or dark (background). The threshold value is set to make binary image streams. This value can be obtained by invoking the frame grabbing function. A utility subroutine (explained in Chapter 5) enables users to manually select the threshold value to be used. Whether this value is only slightly higher or much higher than the background level is up to the user's judgment. Section 6.3.2 explains how the selection should be made. The threshold value is then latched onto the Threshold Register I/O port.

Another task to be performed during system initialization is establishing a mapping table for geometric data correction. The data correction is two-fold; one is for depth value correction (described in Section 3.2.1), which is done in hardware, and the other is

for cross field correction (described in Section 3.3.3), which is done in software. If the geometric distortion is not very large, i.e., when the ratio of depth sensing range to the working distance is small, a "flat mapping table" (without any conversion) can be used to set up the hardware Look-Up-Table as well as the software mapping table. The hardware Look-Up-Table can be disabled by issuing a command to the Command Register (bit 1=0). The Look-Up-Table is a SRAM that allows a 12-bit range value to be converted to 8 bits. Chapter 5 explains the programming method used to establish entries in the hardware Look-Up-Table.

To make sure that the range sensing hardware is functioning correctly, the host computer can perform a diagnostic test of the hardware. Chapter 5 explains the procedure. A diagnostic program can tell whether the individual modules are functioning correctly and/or whether the camera is properly attached.

In addition to the hardware initialization, the host computer has to establish the software environment for data processing. The software initialization includes allocating DMA memory buffers, establishing interrupt services, and/or providing algorithms for real-time processing of range and intensity data (e.g., saving data onto peripheral diskettes). Chapter 5 explains these procedures in details.

4.1.2 Activating the Range Scanning Process

Several system parameters have to be set before activating the range sensing process. Apart from the threshold value and the Look-Up-Table entries described above, the system has to set several options: which camera source (RS170 or MC9128), which intensity processing unit (center pixel intensity or total reflectance), what frame rate subsampling factor (subsampling the frame receiving rate), and which DMA transfer mode

(single or burst transfers) are to be used. The programming methods of setting these parameters are described in Chapter 5.

As soon as system parameters are latched onto the hardware system, a range sensing command can be issued to the Command Register to activate the range sensing process once the memory management environment is allocated (DMA buffers, interrupt servicing, and real-time processing). The data transfer of range and intensity data to the host computer memory is achieved by raising DMA requests. A 16 bit First-In-First-Out (FIFO) memory (two 8 bits FIFO memory chips each 512 words deep) serves as an asynchronous buffer between the range and intensity processing units and the DMA transfer unit. The asynchronous read and write capability of FIFO memory enables the range sensing hardware to generate range and intensity data in parallel with any data processing being done on the host computer. Data transfers happen only when an adequate amount of range and intensity data is buffered in the FIFO memory, i.e., when 512 word FIFO chips are at least half full. The range sensing hardware competes for DMA services and transfers data with the aid of the DMA slave controller residing on the MicroChannel bus.

Three processing units are implemented in the range sensing hardware. The Range Processing Unit serves as a pipelined processor to extract range information from each line of image data. The range processing unit is explained in Section 4.2.3. The other two processing units compute the intensity values that correspond to the range pixel for the purpose of creating perfectly registered pairs of images (both range and intensity images). The Center-Pixel Intensity Processing Unit produces intensity values of detected range pixels (center pixel intensity) while the Total-Reflectance Processing Unit sums up pixel intensities that appear in a bright segment. Users can select either one for the generation

of intensity data before activating a range scanning process. Section 4.2.4 explains the processing methodology.

The word size used in the DMA transfer is 16 bits, or two bytes. The high byte of these 16 bits is reserved for intensity data while the low byte is for range data. The generic range data when fresh out of the Range Processing Unit can be as wide as 12 bits, depending on the image sensor used. To fit the 8 bit constraint on range data, the range information generated by the Range Processing Unit is passed through a hardware Look-Up-Table to convert it to 8-bit size. Entries of this Look-Up-Table are programmable. The intensity data when coming out of the Total-Reflectance Processing Unit can also be as wide as 12 bits. These data are divided by 8 and are truncated to 8-bit data size (ignoring the last three bits and preserving only bit 3 to bit 11). The intensity data generated by the Center-Pixel Processing Unit are exactly 8 bits and therefore do not need to undergo any conversion before the DMA transfer to the host computer.

A memory management scheme has been devised that employs DMA transfers and interrupt signaling. This scheme allows real-time data processing of both the range and intensity images concurrently with the continued acquisition of additional image data. Section 5.4 explains the processing concepts behind this approach.

4.1.3 MicroChannel Bus Management

An integrated chipset, 82C612[1], performs the interface control functions needed by the MicroChannel Bus. Once it is initialized for use on the MicroChannel Bus, the data interfacing to the host system is transparent to the programming environment. This chipset provides bus controls, DMA arbitration, and POS facilities. These functions are

[1]82C612 is manufactured by Chips and Technology Inc.

required by the MicroChannel Architecture (MCA). Using this chipset markedly simplifies interfacing the system to the MicroChannel.

Section 5.1 explains the configuration file of the range scanning hardware required by host systems conforming to the MicroChannel Architecture. Detailed description of the MicroChannel Architecture can be found in [29]-[32].

4.1.4 Data Flow Control

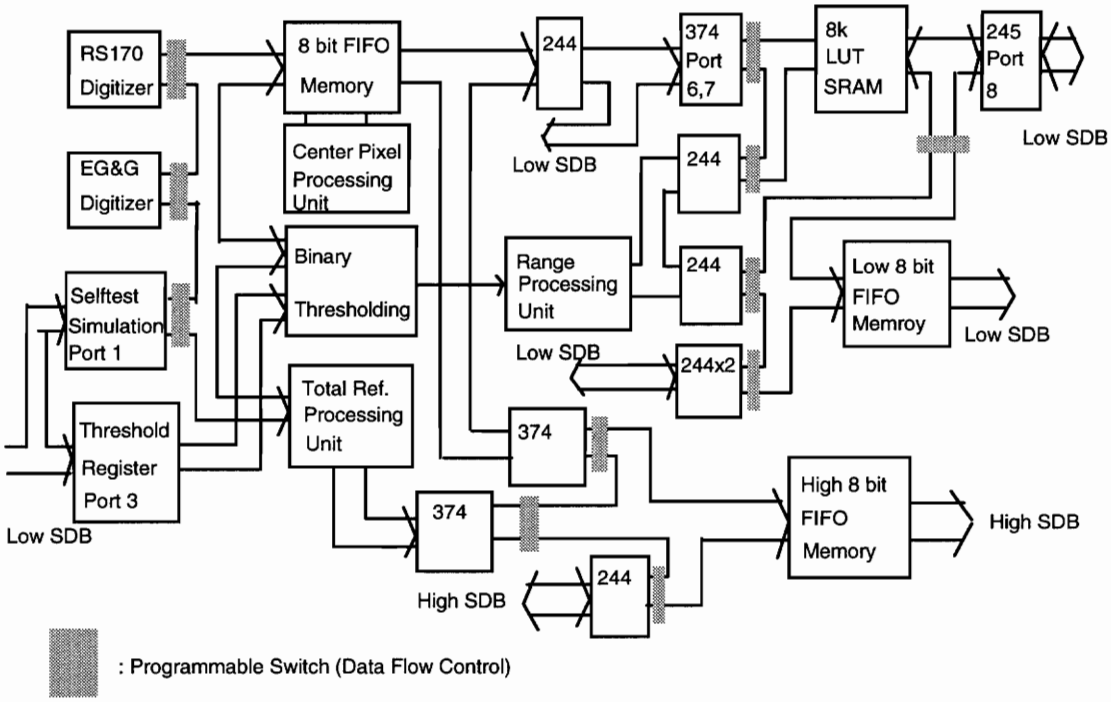


Figure 4.2 Data flow control gates.

The range sensing hardware system can operate in three modes. The Computer Access Mode allows the host system to retrieve data and to perform diagnostic tests. The Frame Grabbing Mode allows the host system to obtain images from one of the two video cameras on the prototype. This mode of operation is provided to simplify optical calibrations and adjustments for the user. The Range Scanning Mode is the main

operating mode, and is used to generate range and intensity data. Figure 4.2 shows the data flow through the range sensing hardware. The gate controls used to control the data flow are determined by bit combinations of the Command Register. Mode switching information can be found in Section 5.2.

To explain range and intensity generation, refer to Figure 4.2. After system parameters are correctly latched, i.e., parameters that determine the video camera, the intensity processing unit, the threshold value, the DMA transfer mode, entries of the Look-Up-Table, the frame rate subsampling factor, and after the software environment has been established, a range scanning command can be issued to the hardware system. The video digitizer provides image and timing signals to be used in the processing. Image signals go into the binary thresholding unit to be thresholded into binary streams. Logic 1 indicates that the image is illuminated by the laser beam. Based on this information, the Range Processing Unit detects a center pixel position in each binary stream. As soon as the Range Processing Unit locates these positions the values are pushed into the Look-Up-Table to be converted from pixel location data to range data. At the same time the image signal goes into the Center-Pixel Processing Unit and the Total-Reflectance Processing Unit to generate intensity values. The Center-Pixel Processing Unit uses a 8-bit FIFO memory as the buffer and latches the center pixel intensity once the Range Processing Unit locates the center pixel positions. The Total-Reflectance Processing Unit sums up intensity values of bright pixels using the information provided by the binary streams. As soon as a video line is finished (indicated by the Line Enable signal), the converted range and intensity data are pushed into the 16-bit FIFO memory to wait for DMA transfer to the host computer.

One after another video lines are processed and data are buffered in the FIFO memory. When the FIFO memory reaches the half full state it raises a DMA request via

the DMA handling circuit. Once the MicroChannel Manger approves the request the DMA slave controller residing on the MicroChannel Bus transfers data from the FIFO memory (a 16-bit I/O port) to the system memory according to software settings. The interrupt handling circuit detects situations when the DMA slave controller needs to be reprogrammed (detecting -TC signal) and raises an interrupt service request to the host computer. The service routine then takes over the reprogramming task and reinstates the range scanning process. The range and intensity generation is therefore completed.

4.2 Hardware Signal Processing

This section introduces the signal processing units implemented in the range sensing hardware. Circuit schematics are shown in Appendix A. These units serve as pipelined data processing units for image data that extract both range and intensity values.

4.2.1 Video Digitizers

Two modules for digitizing video are implemented on the prototype system. One is for digitizing RS170 standard images and the other is for digitizing the MC9128 images. Most of the video cameras sold conform the RS170 standard. Implementing the RS170 standard digitizer allows the prototype to take advantage of this low cost, but slow, method for collecting range data. The prototype also provides the capability of generating range data from MC9128 cameras. These cameras produce genuine timing signals of CCD sensors using the RS422 industrial standard. The advantage of using a MC9128 camera is that it is capable of operating at high frame rates and while still being relatively low in cost.

The range sensing hardware employs an interface timing format similar to the signal format of a genuine CCD sensor. Figure 4.3 shows the timing relationship of these signals. The /Frame Enable indicates that an available fresh frame is in progress (active low). The /Line Enable provides the valid period of line images in a frame. The Pixel Clock is the synchronized data clock for use in data latching and timing controls. All images are digitized to 8 bits. The design goal in using this timing format was to have resulting hardware be easily adapted to many image sensors other than just the MC9128.

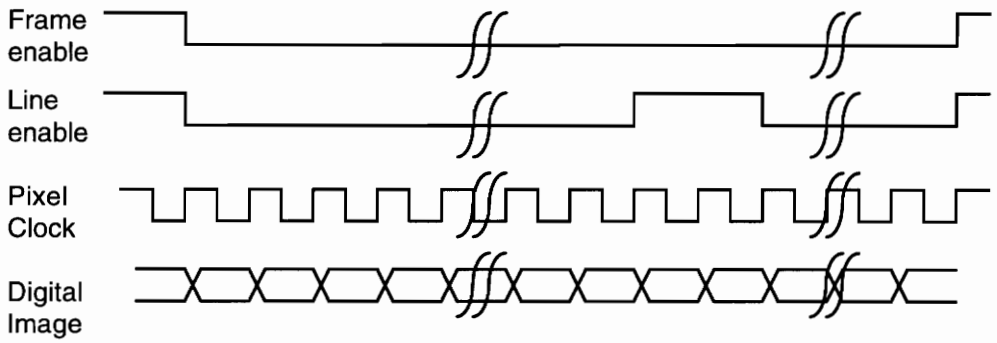


Figure 4.3 The interface timing format of camera signals used in the range sensing hardware.

4.2.1.1 RS170 Digitizer

The RS170 image digitization is accomplished using an AD9502AM[2] chip, a chip specifically designed for RS170 digitization. The pixel clock generator is built-in and is phase-locked to the incoming video signals from a RS170 video camera (7.31MHz version). At this clock rate the digitizer can produce 384 digital pixels per line with a total of 480 lines per frame. Two trimmers, UR1 and UR2 (shown in Appendix A), adjust the offset and gain of the flash A/D converter. The output signals of this chip include Vertical

[2]AD9502 is manufactured by Analog Device Inc.

Sync, Horizontal Sync, Pixel Clock, and digitized image signals. Detailed functioning of this chip is described in [33].

The RS170 digitizing process starts with the detection of a Vertical Sync signal. This signal indicates the beginning of a field, be it an even or an odd field. In this format, picture frames are interlaced with even and odd fields. An even field is composed of even-numbered video lines while an odd field is composed of odd-numbered video lines. A field detection logic determines whether it is an even or an odd field. A picture frame starts with the transfer of an even field and then an odd field. Eleven blank video lines, signaled by Horizontal Sync, is deleted from the frame grabbing process by external decoding logic. Successive video lines contain valid video pixels and can be stored in a frame memory. Horizontal Sync indicates the start of a line. However, the first 6.575 μs of each video line after the falling edge of Horizontal Sync (9.4 μs of width) is invalid and should be discarded from the frame grabbing process.

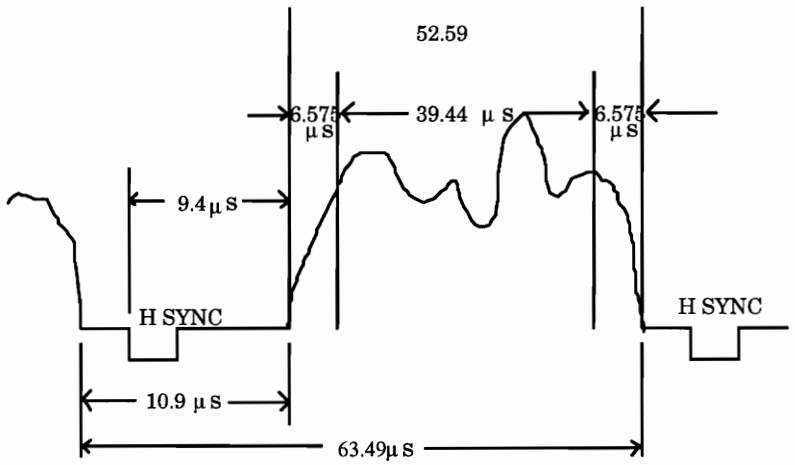


Figure 4.4 RS170 video format[28].

Figure 4.5 illustrates the even and odd field positions of RS170 format. Because of these even and odd fields, software is needed to reformat the data coming from the range sensor. This reformatting involves merging data from the even and odd fields to

create a single range image. The even field results in 240 range pixels of even numbered video lines that are output first, and odd field produces 240 range pixels of odd numbered video lines and is output afterward. An order merging operation of these range pixels has be executed to get a correct 480 range pixels per frame (merging the range pixels of even and odd fields).

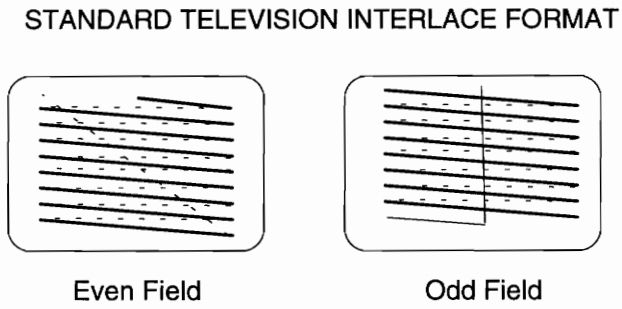


Figure 4.5 Even and Odd Field Scanning format[28].

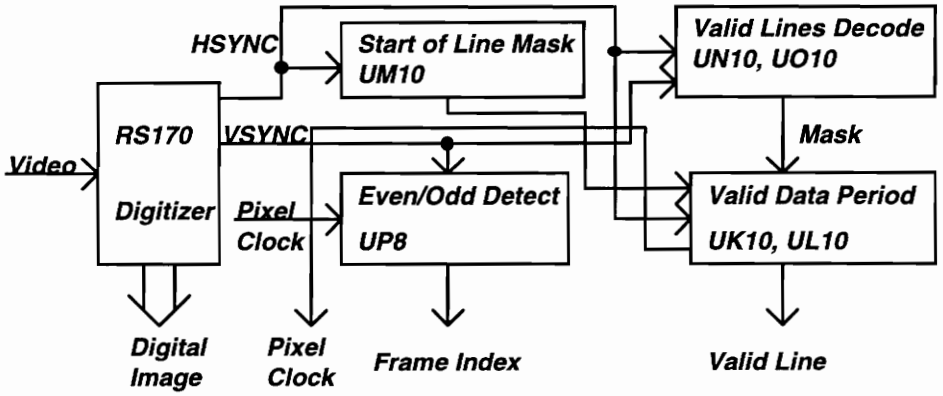


Figure 4.6 RS170 format conversion module.

The timing conversion of the RS170 video format to the interface timing format is accomplished via three sets of counters. Figure 4.6 shows the timing conversion scheme. The timing signals of RS170 format, HSYNC, VSYNC and Pixel Clock, pass through this module. The Even/Odd Field Detector generates a pulse once an even field is detected. This pulse also indicates the start of a fresh image frame. The Valid Line Decode then filters out the invalid line periods that are only used for the phase-locked-loop timing

purposes. In the period of a Horizontal Sync, not the whole line period contains valid image pixels. The Valid Data Period further provides the interval where valid image pixels are available from the RS170 digitizer.

Figure 4.7 shows the timing diagram of this RS170 format conversion module. As is shown in the diagram, the active period of the HSYNC signal does not contain all of the valid image pixels. Sheet 5 of 14 in Appendix A shows the circuit designed to convert VSYNC and HSYNC to LNEN and FREN respectively.

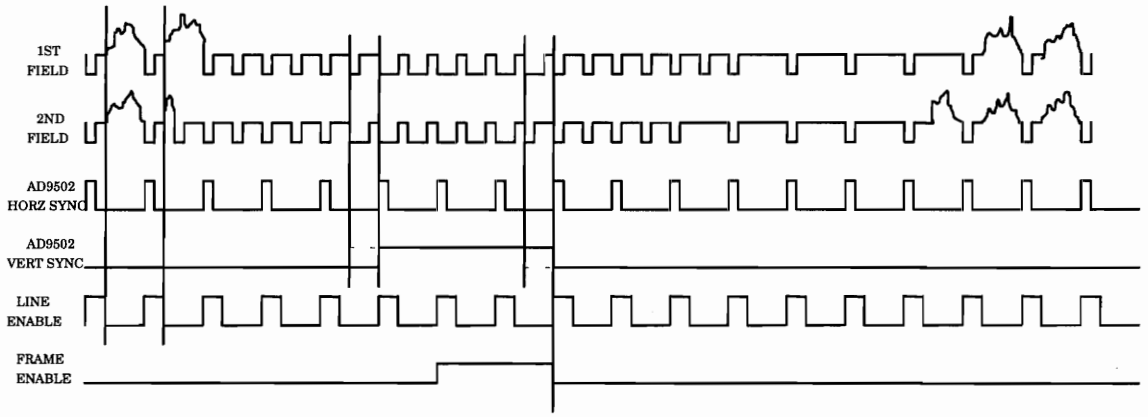


Figure 4.7 Timing diagram of the RS170 format conversion.

4.2.1.2 EG&G Reticon Camera Digitizer

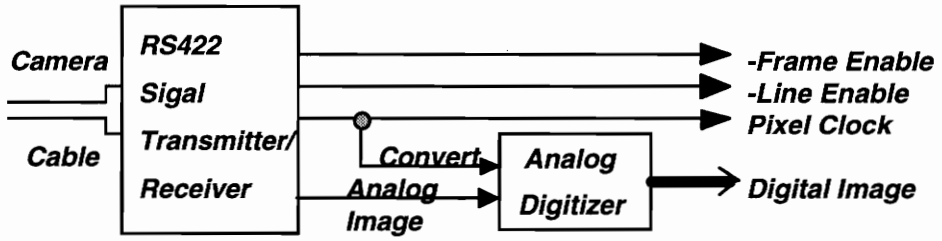


Figure 4.8 MC9128/MC9256 video camera digitizer.

MC9128/MC9256 cameras utilize VLSI CCD timing signals in the camera cable interface. These cameras employ the RS422 industrial standard for cable transmission to

reduce transmission noise. For details about this camera please refer to [28]. Sheet 6 of 14 and 7 of 14 in Appendix A show the circuit designed to recover data from these cameras. Figure 4.8 shows the function blocks of this module.

The digitizer module is composed of RS422 differential transmitters/receivers for recovering camera control signals. An AD9048[3] flash A/D converter, capable of operating at 35MSPS (Million Samples Per Second), converts analog image signals into digital signals. The outputs of this module conform to the format of the interface timing signals in Figure 4.3.

4.2.2 Frame Grabbing Unit

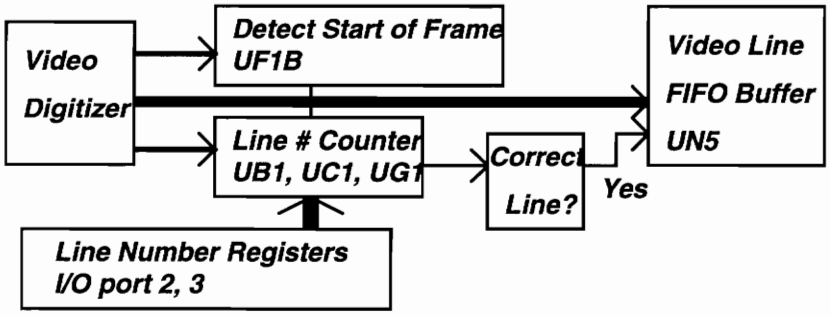


Figure 4.9 The block diagram of the Frame Grabbing Unit.

The purpose of having a frame grabbing circuit is to facilitate the adjustments of optical components. It allows a user to have a full knowledge of what the camera is actually observing so that optical components can be positioned to meet the requirements of field of view, the dynamic range of depth measurement, and image focusing of stripe-of-light images.

Figure 4.9 shows the functional blocks of the Frame Grabbing unit. Instead of grabbing a whole frame into a memory buffer, which requires a large amount of memory,

[3]AD9048 is manufactured by Analog Device Inc.

this unit grabs a single line image whose line number is specified by the host system. The image buffer of this frame grabbing module is a 512 word 8-bit FIFO memory. In order to grab a complete image frame, this unit has to be activated after the host system increments the line number from 0 to the end number of a frame. These line images constitute a complete frame of the camera's view. The Center-Pixel Intensity Processing Unit shares the same FIFO memory with this Frame Grabbing Unit. This FIFO memory also serves as a data processing agent for center-pixel intensity. This implementation helps save development cost of the range sensing hardware as well as minimizing board real estate.

After the host computer latches a line number into the Line Number Register port, a frame grabbing command issued by the host to the Command Register causes this Frame Grabbing Unit to grab a video line with the specified line number from a fresh video frame. The line image is stored in the 512 words FIFO memory. As shown in Figure 4.9, after detecting a fresh frame from the video signals, a set of counters starts looking for the line number that matches the specified number. A counter module composed of UB1, UC1, and UG1 outputs a pulse to a timing circuit. This timing circuit generates signals that enable the FIFO memory to accept image signals. After a complete video line write, this timing circuit freezes the write action and generates status bits (into the Status Register) for the host system to read. The host system polls on these status bits to determine the proper time to read a line image.

Chapter 5 explains the procedures for grabbing a single line as well as for grabbing a complete frame. Although the speed of the frame capturing is slow, it was designed to be simple and to be low in cost. Its sole purpose is provide the user a simple method for making adjustments in the optics of the range sensing prototype system.

4.2.3 Range Processing Unit

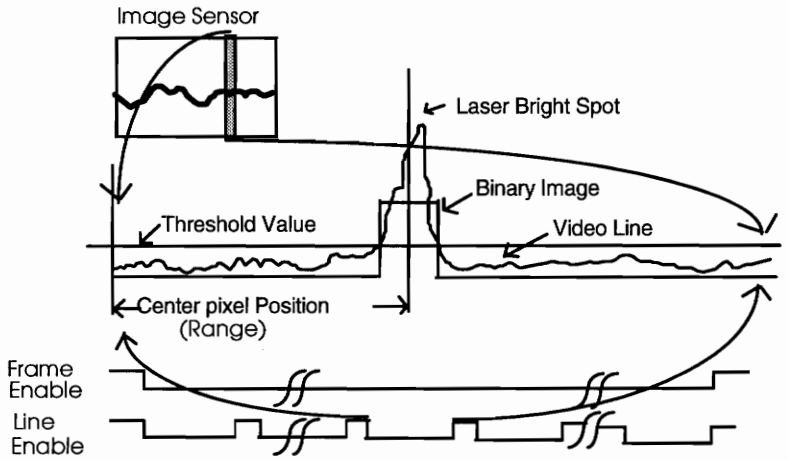


Figure 4.10 Signal processing of range positions.

The range detection algorithm, as described in Chapter 3, was devised to locate the center skeleton of a stripe-of-light image (column-wise thinning) and to determine the positions of skeleton pixels. Figure 4.10 illustrates the processing concept. From the image processing point of view, this algorithm is a one-dimensional thinning filter. A video line (column image) contains a segment that is illuminated by the laser. This video line is thresholded to form a binary stream (bright and dark stream). The Range Processing Unit locates center pixels of bright segments and produces position values to be the output. The image sensor, however, must be positioned such that sequential data output are column-wise. In other words, every output video line must contain only one bright segment. This is required in order to simplify the implementation of the one dimensional thinning.

Figure 4.11 shows the function blocks of the Range Processing Unit. Detailed circuit designs are shown in Sheet 9 of 14 and Sheet 10 of 14 in Appendix A. The heart of this processing unit is the Position Counter Timing Control that monitors pixel positions of bright segments. At the start of a line, the timing control generates pulses to

increment the position counter at the same rate as the Pixel Clock. After encountering a bright segment, the pulse rate is cut in half. When the binary image stream leaves this bright segment, the timing control unit freezes all increments until the end of a video line.

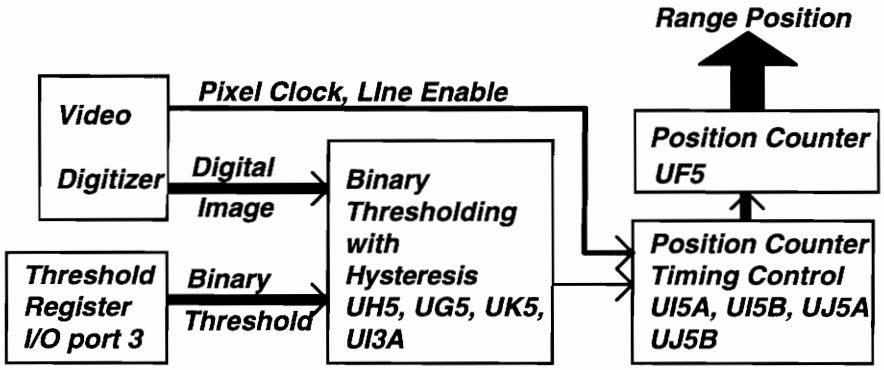


Figure 4.11 Range processing unit.

Figure 4.12 illustrates the timing diagram of this processing unit. As shown in the diagram, the bright spot image starts at pixel 4 and ends at pixel 6 (solid line), a total of 3 clock periods are covered under that bright segment. The center pixel, from visual inspection, is pixel 5. The position counter records the number 5 as it should based on this thinning algorithm. For bright segments of even pixels, however, a certain compromise must be made. The binary image stream represented by the dotted line starts from pixel 4 to pixel 7, a total of 4 clock periods are covered under the bright segment. The center pixel may either be pixel 5 or pixel 6. In this Range Processing Unit we chose the latter.

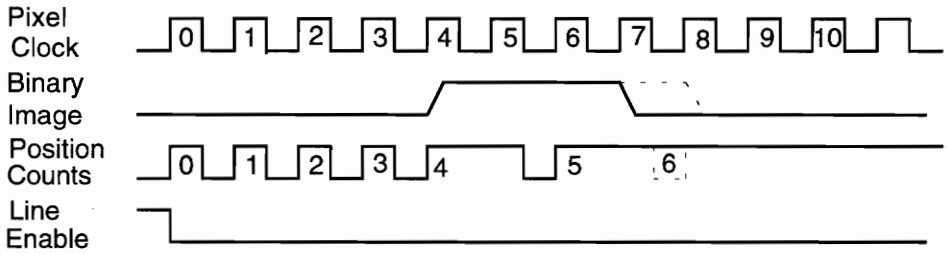


Figure 4.12 Range processing timing diagram.

Since only one bright segment is allowed in a video line, the binary jitters at edges caused by image thresholding have to be avoided. The prototype's signal processing unit performs digital hysteresis filtering. The hysteresis thresholding technique is widely used in image thresholding applications such as bar code reading (analog version). Figure 4.13 illustrate the processing procedure of this hysteresis thresholding. The size of the hysteresis window is determined by the magnitude of the noise. The use of a larger window helps reduce the amount of binary jitters but also distorts the binary image to an extent. A tradeoff has to be considered when choosing the size of this hysteresis window between binary jittering and signal fidelity. Currently we set the hysteresis window at 8 quantization levels of the image digitizer. The digital hysteresis circuit implemented in the image thresholding stage is a simplified version and can only take threshold values at discrete levels (2, 6, 10 , 14, ..., 254).

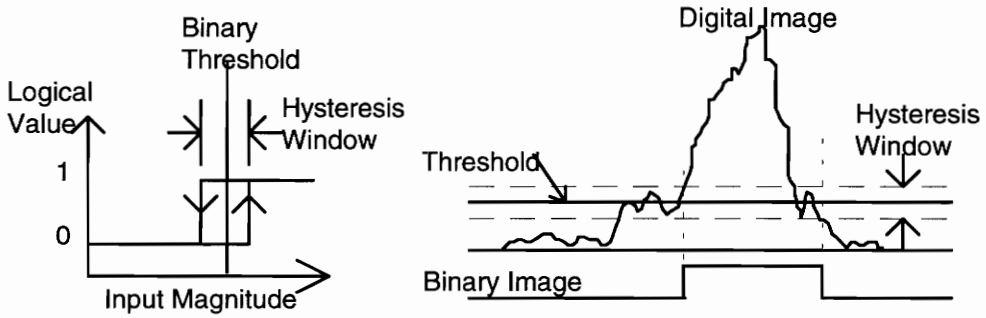


Figure 4.13 Digital hysteresis thresholding.

4.2.4 Intensity Processing Units

Apart from the Range Processing Unit, two intensity processing units are implemented in the range sensing prototype system, for creating intensity, or gray scale images in registration with range images. Having perfectly registered intensity and range

images will hopefully facilitate algorithm development for object recognition since range images contain information about object shape while intensity images provide gray scale information about object surfaces.

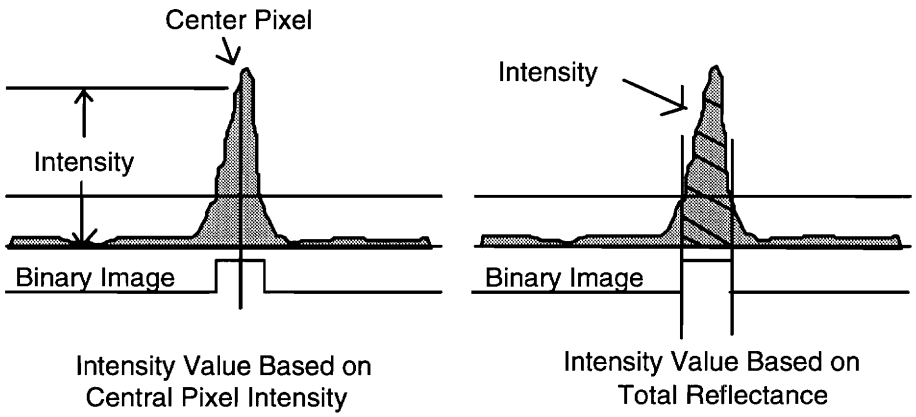


Figure 4.14 Intensity processing algorithms.

Figure 4.14 illustrates the principles of the two intensity processing units. As explained earlier in Chapter 3, center pixel positions are close to the center points of laser illuminated images even if they are slightly blurred. Due to the Gaussian intensity profile of a laser beam, the center pixel is considered to be illuminated by the brightest peak of the laser beam. It is this rationale that prompted the use of the Center-Pixel Intensity Processing Unit. However, when a bright segment covers only a few pixels, i.e., the image is under sampled, this scheme for determining intensity may cause a large variation due to small number of sampling points. It was this concern that prompted the development of another processing unit. As shown in Figure 4.14, the Total-Reflectance Intensity Processing Unit computes the total reflected light power collected by all the bright segment pixels. Selecting the processing unit that should be used depends on the size of the bright segments that appear on the image sensor. The intensity unit used is selected by programming the Camera Select Register.

As described earlier, the Center-Pixel Intensity Processing Unit shares an 8-bit FIFO memory (512 word deep) with the Frame Grabbing Unit. This FIFO memory provides a data buffering function for previously scanned pixel intensities. The length of a bright segment is not a fixed value and therefore this processing unit has to store previously scanned pixel intensities of a bright segment before being able to select the center pixel intensity. Figure 4.15 illustrates functional blocks and timing relationships of this processing unit. After the center intensity value is selected, it is pushed onto the DMA FIFO buffer for a later DMA transfer.

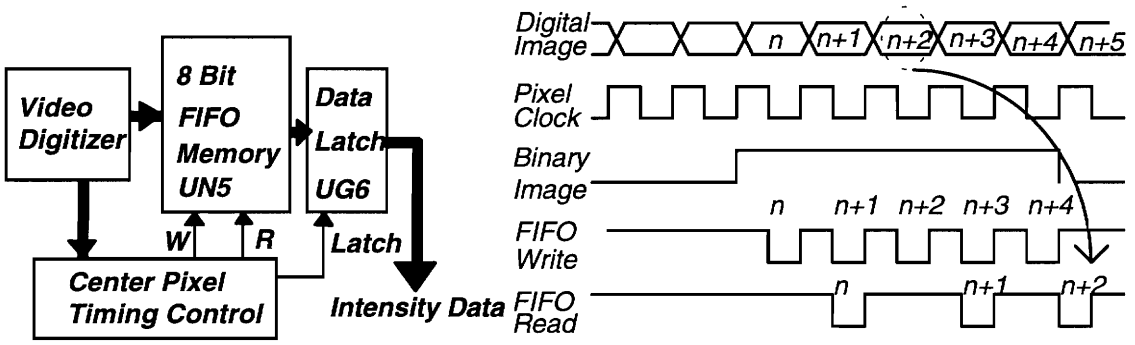


Figure 4.15 Center-Pixel Intensity Processing Unit.

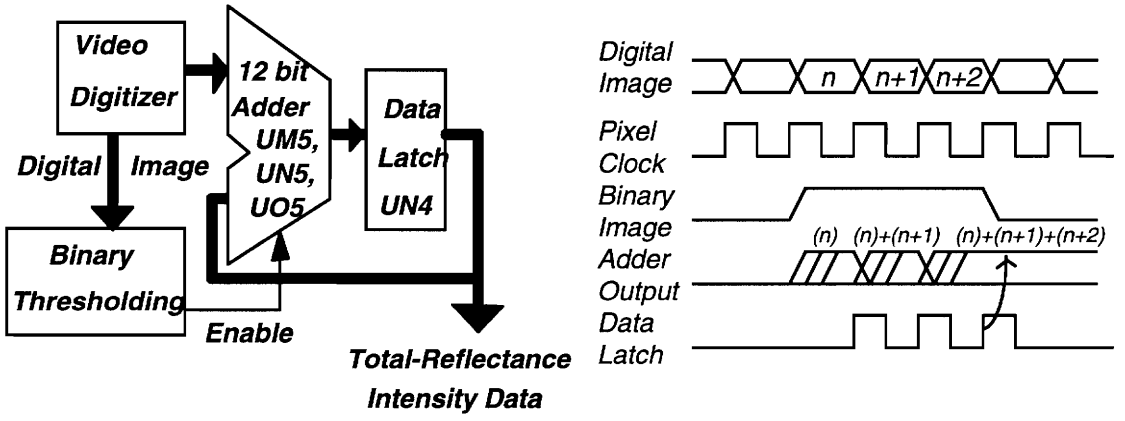


Figure 4.16 Total-Reflectance Intensity Processing Unit.

The functioning of the Total-Reflectance Intensity Processing Unit is simpler than that of the Center-Pixel Intensity Processing Unit. The basic building block is an

accumulator that sums up all intensity values of the bright pixels that appear in a bright segment. Figure 4.16 illustrates the function blocks and the timing diagram of this processing unit.

It is important to note that the speed of the adder module must be such that it can keep up with the pixel clock. Currently all cameras supported by the system have pixel clocks that are less than 10 MHz. It is therefore adequate to choose 74HCT283 as an adder building block.

4.2.5 Look-Up-Table Unit

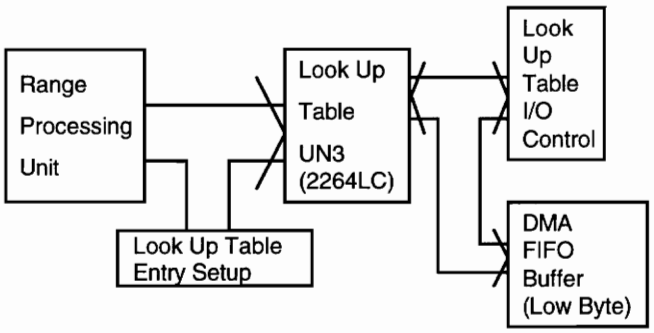


Figure 4.17 Look-Up-Table Unit.

The purpose of implementing a Look-Up-Table is for the real-time mapping of pixel location information into range data. Before a scanning process begins, the user sets up entries in the Look-Up-Table. These mapping entries may be either for geometrical correction or for a special transformation depending on user's needs.

Figure 4.17 illustrates the block diagram of this Look-Up-Table Unit. The output of the Range Processing Unit feeds into this Look-Up-Table that is constructed out of a SRAM. The Look-Up-Table generates mapped data to be pushed into the DMA FIFO buffer. The host computer may load values into this Look-Up-Table when the prototype system is operating in the Computer Access Mode. I/O operations for loading the Look-

Up-Table are explained in Chapter 5. This Look-Up-Table can be disabled by issuing a command to the Command Register (bit1=0).

4.3 MicroChannel Interface

Currently the range sensing hardware system is built on the MicroChannel Architecture for use with IBM PS/2 series computers or IBM RISC 6000 workstations. The MicroChannel Architecture consists of an address bus, a data bus, an arbitration bus, a set of interrupt lines, and a set bus control lines. It uses synchronous and asynchronous procedures for data transfers between memory, I/O devices, and bus masters. The controlling master can be a DMA controller, the system master (system processor), or a bus master. The features of the MicroChannel architecture are [29]:

1. I/O data transfers of 8-,16-,24-,or 32-bits within a 64KB address space (16-bit address width).
2. Memory data transfers of 8-, 16-, or 32-bits within a 16MB (24-bit address width) or 4GB (32-bit address width) address space.
3. An arbitration procedure that enables up to 15 devices and the system masters and slaves.
4. A direct memory access (DMA) procedure that supports multiple DMA channels. Additionally, this procedure allows a device to transfer data in bursts.
5. An optional streaming data procedure that provides a faster data-transfer rate than the basic transfer procedure and allows 64-bit data transfers.
6. Address- and data-parity enable and detect procedures.
7. Interrupt sharing on all levels.

8. A flexible system-configuration procedure that uses programmable registers.
9. An adapter interface to the channel using:
   - A 16 bit connector with a 24-bit address bus and a 16-bit data bus
   - A 32-bit connector with a 32-bit address bus and a 32-bit data bus
   - An optional matched-memory extension
   - An optional video extension
10. Support for audio signal transfer (audio voltage-sum node).
11. Support for both synchronous and asynchronous data transfer.
12. An exception condition reporting procedure.
13. Improved electromagnetic characteristics.

The range sensing hardware system uses an 82C612 chipset in decoding MicroChannel control signals and in providing necessary POS[4] registers for use on the MicroChannel Bus. For details about MicroChannel Architecture and 82C612 refer to [29]-[32]. The following section explains the control procedures of the MicroChannel bus performed by the range sensing hardware. The software bus management procedures will be described in Chapter 5.

4.3.1 MicroChannel Interface Decode

The MicroChannel management of the range sensing hardware is accomplished via the control logic provided by the 82C612. Figure 4.18 illustrates the block diagram of a standard MicroChannel interface structure using an 82C612.

[4]POS stands for Programmable Option Select. It is a salient feature of the Micro Channel Architecture that replaces hardware switches.

The range sensing hardware utilizes thirteen I/O addresses, one DMA slave channel and one interrupt level. The I/O address space allocation and DMA channel assignment are programmable through POS registers inside the 82C612. Power-on selftest procedures of the MicroChannel based host system program these POS registers and resolve interface resource conflicts. For a newly installed interface adapter, an ADF file for this adapter must be available to the MicroChannel configuration program. This is required by the MicroChannel Architecture. POS register values are saved in a non-volatile memory of the host system for later power-on MicroChannel initialization. Section 5.1 describes the format of the ADF file for the range sensing hardware. It is listed in Appendix B.

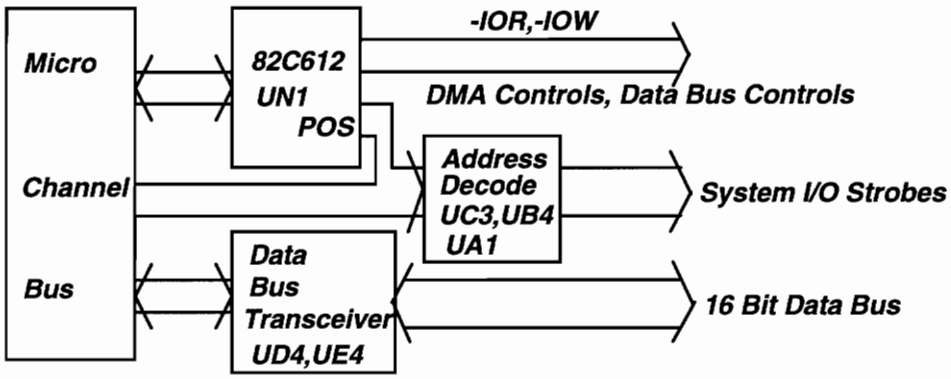


Figure 4.18 MicroChannel Interface Decode.

4.3.2 Command Register Design

The range sensing hardware utilizes thirteen MicroChannel I/O ports. of all these thirteen ports the Command Register is the most important one. It controls all the operations conducted by the range sensing hardware. If this register is improperly set the range sensing hardware may very well crash the host system. Figure 4.19 illustrates the design of the Command Register.

To avoid latching a random value into the Command Register during system power-on, a logic design disables the Command Register right after hardware power-on. A D-type flip-flop (UI3A) controls the output enable pin of this Command Register (74HCT374). The host MicroChannel system activates a CHRES pin (on the MicroChannel bus) during system power-on and at the same time disables the Command Register through clearing the flop-flop (UI3A). To enable the Command Register the host must execute an I/O WRITE operation to the Command Register that will force D1 low. A set of pull-up resisters at the output end of the Command Register ensures that all the output signals are high (value FFh) before this register is enabled. This ensures that the range sensing hardware is in a safe state since the Command Register employs negative logic (active low).

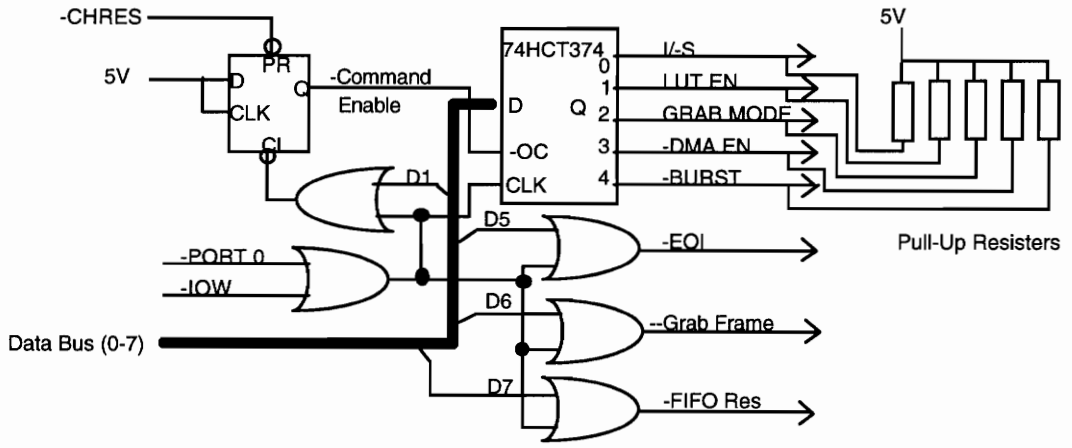


Figure 4.19 Command Register.

Five control signals using level logic (bit 0 to 4, high/low logic) and three control signals using pulse logic (bit 5 to 7, pulse active) are implemented on the Command Register. Function descriptions of these control signals are listed in Section 5.2.1.

4.3.3 DMA handling

The range and intensity data generated by the range sensing hardware is transferred to the host system's memory through DMA requests. The DMA transfers are handled by a DMA slave controller residing on the MicroChannel Bus. Transfer requests to the slave are initiated by the range sensing hardware. Proper programming of this DMA controller is necessary for correct system operations.

Two DMA transfer modes are available by issuing a correct command to the Command Register. The Burst Mode DMA Transfer performs multiple DMA transfers at one DMA request. The Single Mode DMA Transfer, on the other hand, transfers only one datum during a DMA request. This mode obviously wastes DMA channel resources and is not recommended. Nevertheless, this mode is compatible with the ISA[5] bus standard. The DMA transfer mode is selected using bit 4 of the Command Register (0 for Burst Mode, 1 for Single Mode).

The DMA slave controller must be properly programmed to perform I/O to the host system's memory. If only range data is to be transferred, an 8-bit DMA transfer mode can be programmed into the DMA slave controller. The 16-bit DMA transfer mode, on the other hand, transfers both range and intensity data to system memory. Programming methods for using either of these two transfer methods will be described in Chapter 5.

The signal used to make a of DMA request is generated by detecting status flags of the 16-bit DMA FIFO memory (512 word deep). When the range sensing hardware is operating in the Range Scanning Mode, the range and intensity DMA FIFO memory receives data at the camera frame rate. Once this FIFO memory reaches a half full state (-

[5]ISA is the interface bus standard for IBM PC/AT computers.

HF flag low), the DMA handling circuit issues a DMA request to the 82C612. This MCA[6] controller chip then takes over the work of bus arbitration. The allocated DMA channel of the range sensing hardware is programmed into the DMA slave controller. The DMA slave controller then monitors the arbitration levels and takes over DMA transfers for the range sensing hardware once the MCA controller chip gets control of the bus.

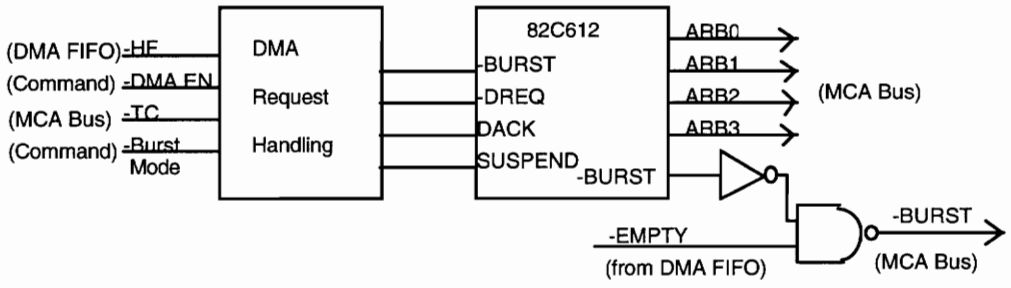


Figure 4.20 DMA handling scheme.

The 82C612's DMA request logic is was be adapted to meet the needs of the range sensing hardware. The DMA FIFO initiates a DMA request by raising the Half Full flag and terminates DMA transfers by activating the Empty flag. Unfortunately the 82C612 is unable to terminate DMA requests right after detecting the Empty flag when operating in the DMA Burst Mode due to a timing mismatch. An additional I/O READ by the DMA slave controller is performed right after deactivating the DMA request signal. This results in a DMA transfer error. Figure 4.20 illustrates the circuit modification for correcting this situation. Instead of connecting 82C612's BURST pin directly to the MicroChannel Bus, a logic substitution is performed that incorporates the empty flag (-EMPTY) of the DMA FIFO before connecting to the MCA Burst Mode pin (see Sheet 2 of 14 in Appendix B for more detains).

[6]MCA stands for Micro Channel Architecture.

4.3.2 Interrupt Handling

The occurrences of either of two events will initiate interrupt requests:(1) when the Terminal Count (-TC) of the DMA slave controller is activated during range and intensity transfers, or (2) when the DMA FIFO is full and is likely to overflow (-FULL flag activated). After the DMA slave controller signals a Terminal Count (-TC) when transferring data, the interrupt service routine should provide this controller with another starting address and count number for transmitting additional data. The starting address is directed to a blank page of a data buffer and the count number is the size of this page. If complete data transfers are finished, the interrupt service routine should terminate the scanning process and set semaphores to the main program to indicate this situation. Another function that an interrupt service routine should perform is to ensure the data integrity of the range sensing hardware. When the DMA FIFO memory is full, a full flag (-FULL) will be set by this FIFO memory. This full flag then triggers an interrupt request to the host computer and signals that data overflow is happening. The host computer then either terminates the scanning process or acknowledges this situation and ignores it.

After the host system has finished servicing an interrupt request, an End of Interrupt (-EOI) command must be issued to the Command Register to reset the interrupt request initiated by the range sensing hardware. EOI function is performed by writing a command to the Command Register with D5 low. The range sensing hardware shares interrupt level IRQ9 with other adapters and provides an interrupt ID bit on the Status Register port (bit 2) for host system identification. Chapter 5.3 describes the programming method of interrupt servicing.

4.3.3 Mode Switching Unit

The range sensing hardware provides flexible controls over the range sensing process and any optical calibration. Three modes of operations are provided in controlling this hardware system. Chap 5 describes operations that can be performed in each operating mode and the commands needed to control these operation. Switching the operating modes is accomplished using two high impedance data buffers (74HCT244s, UH2 and UI2). Sheet 14 of 14 in Appendix A shows the design of this mode switching unit.

(A) Computer Access Mode:

The Computer Access mode allows the host system to gain access to the core of the range sensing hardware. Bit 0 of the Command Register indicates the operation of this mode. The use of this mode is basically for system diagnosis. A dedicated I/O port (Computer Simulated Camera Control Register) simulates the camera control signals. Together with the Simulated Digital Image Port that provides simulated digital images, the host system can test the range sensing system's hardware processing units. Data can be tapped and checked by the host system when the range sensing hardware is operating in this mode. The diagnostic testing that can be performed includes tests of the FIFO memory chips, the Look-Up-Table memory, the system registers, and the range and intensity processing units.

(B) Frame Grabbing Mode:

This mode enables the range sensing hardware to grab a video line whose line number is specified by the host system (this number is latched into the Line Number I/O register). The design goal is to allow the host system to collect intensity data from the camera across any part of the camera or across the whole field of view, for the purpose of optical calibrations. Optical calibrations include adjusting the optical system for good image focusing, correcting the width of the field of view, correcting the dynamic range of depth sensing, and/or establishing a correct Look-Up-Table.

(C) Range Scanning Mode:

The range sensing hardware performs pipelined range and intensity processing when the host system activates this mode. The range sensing hardware captures image frames right after detecting the activation command (issued to the Command Register with bit 0 low) and processes image frames into profile lines according to the parameters set into this hardware. Every frame is processed down to a profile line. The 16-bit DMA FIFO buffers the range and intensity data and initiates DMA requests once these data reach half full count (256 words when using 512 word-deep FIFO memories). Interrupt requests of the data overflow (-FULL flag of the DMA FIFO) ensure the integrity of the processed range and intensity data.

The mode switching is controlled by bit 0 and bit 1 of the Command Register. Four combinations are available out of these two bits. A decode logic separates these combinations and feeds proper enabling signals to 74HCT244 chips that regulate the

control signals to all function blocks. Only three combinations are used in this process. The mode activation method will be described in Chapter 5.

4.4 Summary

This chapter explains the design of the range sensing hardware. The primary objective of this design is to provide a flexible device for generating range and intensity data. A multi-mode operating scheme was implemented that allows flexible manipulation of the device. The usage of the multi-mode operation is described in Chapter 5.

The range sensing hardware consists of six modules. (1) Image digitization units convert analog images into digital images for use by the hardware. (2) A frame grabbing unit enables users to obtain an image of what is in the camera's field of view. This facilitates making optical adjustments. (3) Range and intensity processing units read incoming video signals and produce range and intensity data in a pipelined manner. (4) A Look-Up-Table unit converts pixel position information into range data in real-time. The contents of this LUT are provided by the host computer. (5) A memory buffer unit stores range and intensity data prior to DMA transfer to the host computer. (6) A MicroChannel control unit handles communications between the host computer and the range sensing hardware. Three basic functions are administered by this unit: I/O operations, DMA transfers, and interrupt servicing.

Thirteen I/O ports control the range sensing hardware system. Three operation modes are provided in the design. The Computer Access Mode is designed such that the host computer can perform sophisticated diagnostic test on the range and intensity generation hardware. The Frame Grabbing Mode enables users to acquire and image of the camera's field of view and hence make many needed adjustments on the optical

settings. The Range Scanning Mode sets the hardware in a pipelined operating mode. Video signals are processed and sent to the host computer through DMA requests. With the help of interrupt servicing, real-time software processing is possible.

Chapter 5
Software Programming

This chapter describes programming methods of initializing and activating the range sensing system. System control is achieved by programming thirteen I/O registers, referred to as system registers. The memory management scheme used by the range sensing hardware facilitates the development of real-time processing algorithms that takes advantage of the low data generation rate of range scanning. Utility subroutines for controlling the range sensing system are listed in Section 5.3. This chapter also provides programming examples for optical calibration and for range scanning activation.

5.1 General Introduction

The design of range sensing hardware attempted to maximize software control over the system. Almost all of the hardware units presented in Chapter 4 are software controllable via I/O operations. Figure 5.1 shows the configuration of I/O registers residing on the range sensing hardware. Through this map it can be seen that the hardware is totally controlled by the software programming. The following section explains how I/O operations are used to control the hardware system.

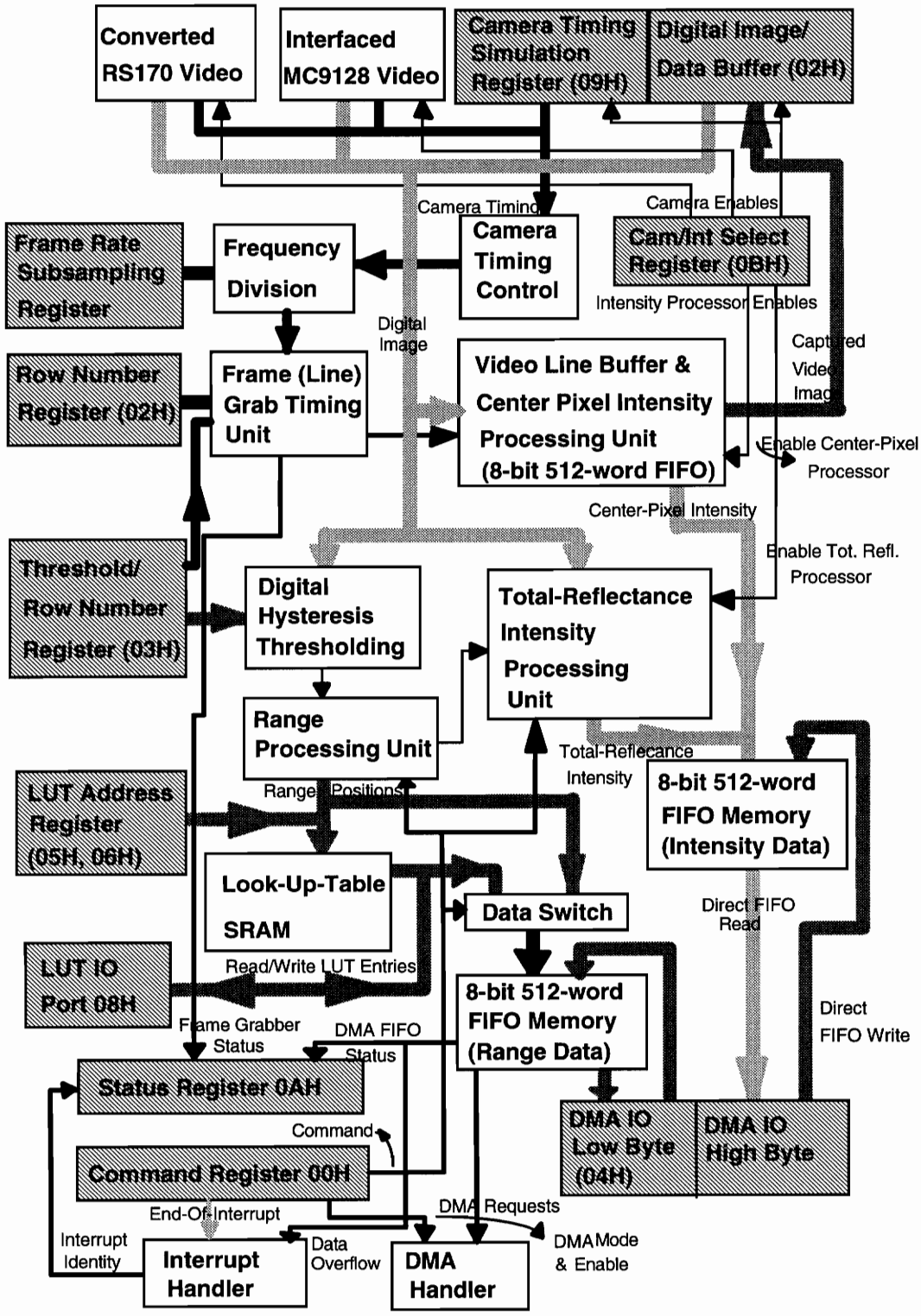


Figure 5.1 Configuration of the system I/O registers.

5.1.1 Hardware System Initialization

MicroChannel systems and adapters support a Programmable Option Select (POS) facility that eliminates the need for switches on system boards and adapters. The Programmable Option Select facility is implemented by programming a set of I/O ports. These ports can only be accessed during system initialization such as the system power-on initialization. An automatic configuration and change configuration utility is provided on the System Reference Diskette in the accessory kit of IBM PS/2 series computers Model 50 or higher. These utilities set previously configured parameters onto adapters' POS registers and set new values on newly installed adapters according to the Adapter Description File (ADF).

An Adapter Description File contains allocation choices of system resources, i.e., interrupt levels, DMA arbitration levels, I/O address spaces, and memory address spaces, that will be occupied by this adapter. An automatic configuration utility identifies unknown adapters on the MicroChannel bus and automatically resolves and allocates system resources according to their description files (each adapter carries an unique hardware ID number, as is listed on the top of its description file). A change configuration utility allows the user to change the allocated system resources. It is only after being configured that an adapter card is enabled for use on the MicroChannel system.

Appendix B lists the adapter description file of the range sensing hardware (file name @59CD.ADF). The setting of I/O addresses and a DMA channel has to be in accordance with the software programming environment. The utility subroutines use the system configuration: I/O addresses 1380H-138FH, DMA channel 6, shared interrupt level 9. These parameters can be found in the definition file *scanner.h* in Appendix C. A detailed description of the MicroChannel management can be found in [29]-[32].

As soon as the MicroChannel master assigns MicroChannel resources onto the POS registers of the range sensing hardware, the hardware is ready to interact with which ever software process that wants to use it. The base address of thirteen I/O ports, the interrupt level, and the DMA channel used by the range sensing hardware are now known information. Software can now be programmed to manipulate the hardware system.

After system power-on, the range sensing hardware is in a stationary state. A byte command with bit1=0 has to be written to the Command Register to activate the range sensing hardware. This action was explained in Section 4.3.2. The Command Register is the I/O register residing on the base I/O address allocated by the MicroChannel master. This register is the main I/O register that controls the operation of the range sensing hardware. Section 5.2.1 explains operations of this register.

5.1.2 System Diagnosis

As explained in Chapter 4, the range sensing hardware operates in one of three modes: Computer Access Mode, Frame Grabbing Mode, and Range Scanning Mode. The Computer Access Mode is basically designed to acquire hardware logic status. A byte command with bit 0 and bit 2 high written to the Command Register activates this mode. The computer can "tap" into several points on the hardware, including all memory values stored in memory chips on the range sensing hardware. Most I/O registers are bi-directional.

Table 15.1 lists all thirteen I/O registers used in the system. Figure 5.1 illustrates the usage of these thirteen registers. The testing of registers are done by writing to and reading back the values to check if the data are correct. There are exceptions. The Digital Image/Data Buffer port (I/O address offset 1) is the I/O port connected to frame

grabber's video FIFO buffer. It manages a 512-byte FIFO memory. The DMA IO port (I/O address offset 4) connects to the data ports of the 16-bit DMA FIFO. It manages a 512-word FIFO memory. The LUT Address port (I/O address offset 6,7) sets the addresses for the SRAM Look-Up-Table during system setup, and the LUT I/O port enables the host computer to read and write the SRAM memory contents. Memory checks of the FIFO memory and the LUT SRAM are done by manipulating these ports. Section 5.2 explains detailed usage of these ports.

The testing of the Range Processing Unit, Center-Pixel Intensity Processing Unit, Total-Reflectance Processing Unit, and the frame grabber are done by replacing the regular video source (either RS170 or MC9128) with simulated virtual video ports controlled by the host computer. As seen on the top of Fig. 15.1, there are two I/O ports reserved for generating the simulated video signals. The host computer generates proper video signals as described in Section 4.2.1 while activating the Frame Grabbing or Range Scanning mode. A diagnostic program then retrieves data stored in the DMA FIFO memory or the Video FIFO buffer and compares them with the desired data. Procedures of activating the Frame Grabbing Mode and the Range Scanning Mode are explained in the following sections.

The program can also test DMA transfers and interrupt servicing by driving the virtual video source. The hardware system is operating in the Range Scanning Mode while the video source is driven by the host computer. The host computer therefore can drive the hardware system as though this system were performing normal scanning operation. A diagnostic program is provided for the host computer to determine the integrity of the range sensing hardware. Malfunctioning hardware units can be identified by this program.

5.1.3 Frame Grabbing Operation

The statement in Section 4.1.1 explains the need for the capability of grabbing video frames. As stated, the goal is to acquire image data of the video camera's field of view. As explained in Section 4.2.2, the hardware implements a simplified frame grabbing unit that captures a single video line at a time rather than a whole video frame. The video line that is obtained, however, can be specified by the host computer. This number is provided by the Row Number Register (I/O address offset 2).

Before activating the frame grabbing process, the host computer has to choose which video source to use. As shown in Figure 5.1, there are three video sources available: RS170, MC9128, and simulated virtual video camera. As explained in Section 5.1.1, the simulated virtual video camera is primarily used for hardware diagnosis. Selecting the camera source is done by writing a command to the Camera Select/Intensity Processor Select Register (I/O address offset 11). A detailed description of this register can be found in Section 5.2.10.

After the row number is given, a byte command with bit 0=1 (I/-S=1) and bit 2=0 (-IG=0) issued to the Command Register activates the frame grabbing operation. The byte command should have bit 6=0 to reset the fresh video frame detector that searches for the start of a fresh video frame. After a fresh frame is detected, a status bit is set on the Status Register (bit 7=0, or -FM DET=0). The frame grabbing unit then searches for the video line whose number is given in the Row Number Register. Once detected, the unit sends image signals into the video FIFO buffer and at the same time sets bit 6 low (-LINE ON=0) on the Status Register. After the whole video line is being recorded into the video FIFO buffer, the unit sets bit 5 low (-LINE END=0) on the Status Register. During the whole process, the host computer monitors the status byte by reading the Status

Register. After detecting the completion of frame grabbing, the hardware is switched back to the Computer Access Mode by sending a byte command with bit 0=1 (I/-S=1) and bit 2=1 (-IG=1) to the Command Register. The host computer reads the Row Image Buffer port (I/O address offset 1) sequentially to get image signals recorded by the frame grabbing unit.

To acquire a complete video frame, the host computer provides the Row Number Register with each line number appearing in the video frame and performs the procedures stated above for each one of these lines. Grabbing a complete frame is slow (approximately 1 second for a 128X128 image), it, nevertheless, provides the capability for performing optical adjustments while markedly simplifying the hardware design.

5.1.4 Range Scanning

After all optical adjustments are finished, the hardware can then proceed with the process of range scanning. Section 4.1.2 explains the general procedures needed to be performed before activating the range scanning process. As explained, four tasks have to be done. First, the host computer has to choose which video source to use. This action has been described in pervious section. Second, the host computer has to choose the intensity processor to use. The selection is done by writing a byte command to the Camera Select/Intensity Processor Select Register. Section 5.2.10 describes the bit assignments of this register. Third, the host computer has to put a threshold value on the Threshold Register (I/O address offset 3). This is the value used to distinguish bright from dark pixels. Detailed description of the threshold value can be found in 6.3.2. The host computer can invoke the frame grabbing function to acquire image signals to determine the proper threshold value. Fourth, the host computer has to set up the Look-Up-Table

for range data mapping. Geometrical data correction may be needed, as described in Section 4.2.5. Look-Up-Table values can be set by latching a 16-bit address value to the LUT Address Register (I/O address offset 6) and performing memory writes to the LUT I/O port (I/O address offset 8)

During system setup, the hardware must be operating in the Computer Access Mode, a mode that allows the host computer to do hardware management. As soon as system setup is complete, the range scanning process can begin. The range scanning processing, however, needs to cooperate with the software running on the host. The host computer's system memory has to be allocated to perform data transfers from the range sensing hardware. In next section a real-time memory management scheme is presented for the range scanning process. This scheme involves the use of the host system's memory, the DMA transfer of data, and interrupt servicing to complete the work. Section 5.1.5 explains more about this.

As soon as all preparations are done (hardware setup, software memory management), the host computer can issue a byte command with bit 0=0 (I/-S=0) and bit 2=1(-IG=1) to the Command Register to activate the range scanning process. Bit 6 (-Grab) and bit 7 (-RES) are also driven low to reset the DMA FIFO and reset the fresh video frame detector. Bit 4 (-BURST) determines the mode of the DMA transfer. Section 5.2.1 explains the functioning of the Command Register.

The range scanning process then proceeds. As shown in Figure 15.1, video signals enter the frame rate subsampling unit that uniformly reduces the amount of video frames to be processed by the hardware (one for every N frames when the value N is latched onto the register). Image signals enter the digital hysteresis thresholding unit and are converted into binary image streams. These binary streams enter the Range Processing Unit and are processed to determine center pixels. These pixels then enter the Look-Up-Table to be

mapped to range data. The data coming out of this LUT are then pushed into the DMA FIFO buffer (low byte) to await a DMA transfer to the host. The intensity generated by the system follows a different route. The image signals enter the Center-Pixel Intensity Processing Unit as well as the Total-Reflectance Intensity Processing Unit from their video source. One of these two units is enabled and the other disabled by the selection command on the Camera and Intensity Processor Select Register. The intensity data created are then pushed into a DMA FIFO buffer (high byte) to await a DMA transfer to the host.

Lines of video data are processed one after another and pushed onto a 16 bit DMA FIFO buffer. After reaching the half-full state (-HF low), the DMA handling unit issues a request for a DMA transfer. The DMA slave controller then takes over the DMA operation by reading this DMA IO port (16-bit I/O address offset 4) and writing data to the host computer's system memory. The interrupt handling unit on the range sensing hardware monitors the proceeding of DMA transfers. When the DMA slave controller finishes the designated number of data transfers, the DMA process is stopped internally and the controller issues a terminal-count signal to the MicroChannel bus. The interrupt handling unit detects this event and initiates an interrupt request to the host computer. An interrupt service routine responds to the request and performs the reprogramming of the DMA slave controller. This reprogramming task includes providing a new starting address and the number of data transfers of a new page of system memory to the DMA slave controller.

When the required amount of range and intensity data are scanned and saved in the host computer memory, the host computer issues a command for restoring the range sensing hardware back to the Computer Access Mode to terminate the range scanning process. This completes the process of range scanning. There is, however, one exception

that may abort the scanning process. The data produced by the range sensing hardware is buffered in a DMA FIFO memory for data transfers to the system memory. The MicroChannel bus latency of servicing DMA requests can take too much time in which event the FIFO memory will be filled and data will be lost. When this situation occurs, the range sensing hardware issues an interrupt request and sets bit 4 low (-OF=0) on the Status Register. The interrupt service routine should check this bit to determine whether data have been lost. If so it should set a semaphore for the main scanning process. Upon detecting this situation, the main program should terminate the scanning process. The data management scheme is explained in details in next section.

5.1.5 Real-time Memory Management

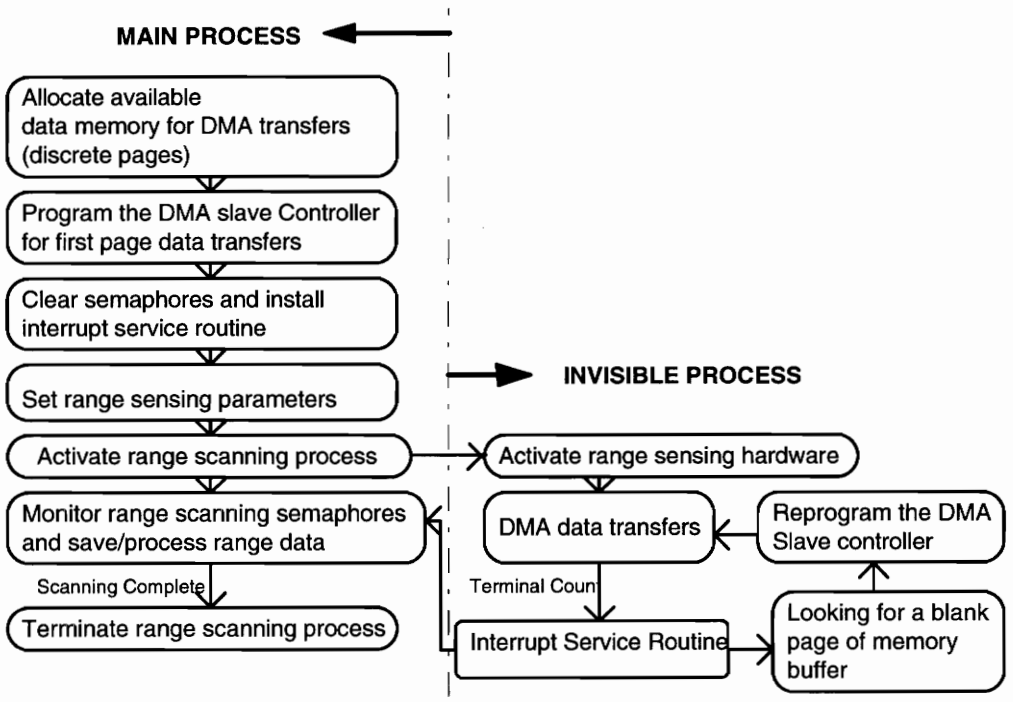


Figure 5.2 Real-time memory management of the range sensing hardware.

Since the range sensing hardware is a slave interface device, correct control procedures have to be executed by the host computer to ensure data integrity. Basically control commands either set system parameters or direct data flow within the hardware. Range and intensity data transfers between range sensing hardware and the host system's memory are performed by DMA requests and are administered by a built-in DMA slave controller on the MicroChannel. Because of the use of DMA transfers, this process is transparent to users, i.e., users do not know that this process is proceeding.

The signaling of scanning process completion or the signaling of a data loss failure to the main program is accomplished setting semaphores via an interrupt service routine. Two situations during range scanning will invoke interrupt service requests: data overflow and DMA terminal count. When the count number programmed in the DMA slave controller reaches an assigned value, a Terminal Count (-TC) pin on the MicroChannel is driven low to indicate this situation. The range sensing hardware monitors this signal when it has the control over the MicroChannel bus. It stops requesting DMA transfers when this pin is driven low, and issues an interrupt service request to allow the host to reprogram of the DMA slave controller. The interrupt service routine then checks for system memory information and error messages from the hardware (through the Status Register) and sets proper semaphores for indicating the range scanning status. Figure 5.2 shows the memory management used in range data generation.

The real-time processing of range and intensity images is illustrated in Figure 5.3. It proceeds as follows. Several pages of memory are allocated in the beginning as data buffers for range images. Semaphores accompanying these memory pages are created. These semaphores hold information about their respective memory pages such as the page number, whether the page is empty or full, page size, etc. After the range sensing hardware is activated, the interrupt service routine finds an empty memory page and

reprograms the DMA slave controller to transfer range data to this page. Once this page is filled, the appropriate page semaphores are set by the interrupt service routine. A main program monitors these semaphores and searches for filled pages that contain range data. The program then processes these data concurrently with the continued DMA transfers of additional range information. Once a page of data has been processed, this page is released and is again available as a DMA buffer. If the processing algorithm requires more execution time, these memory pages function as a reservoir for buffering range/intensity data. In using the scheme, the host can take advantage of the time in between DMA transfers and process data so that the host system's resources are fully utilized.

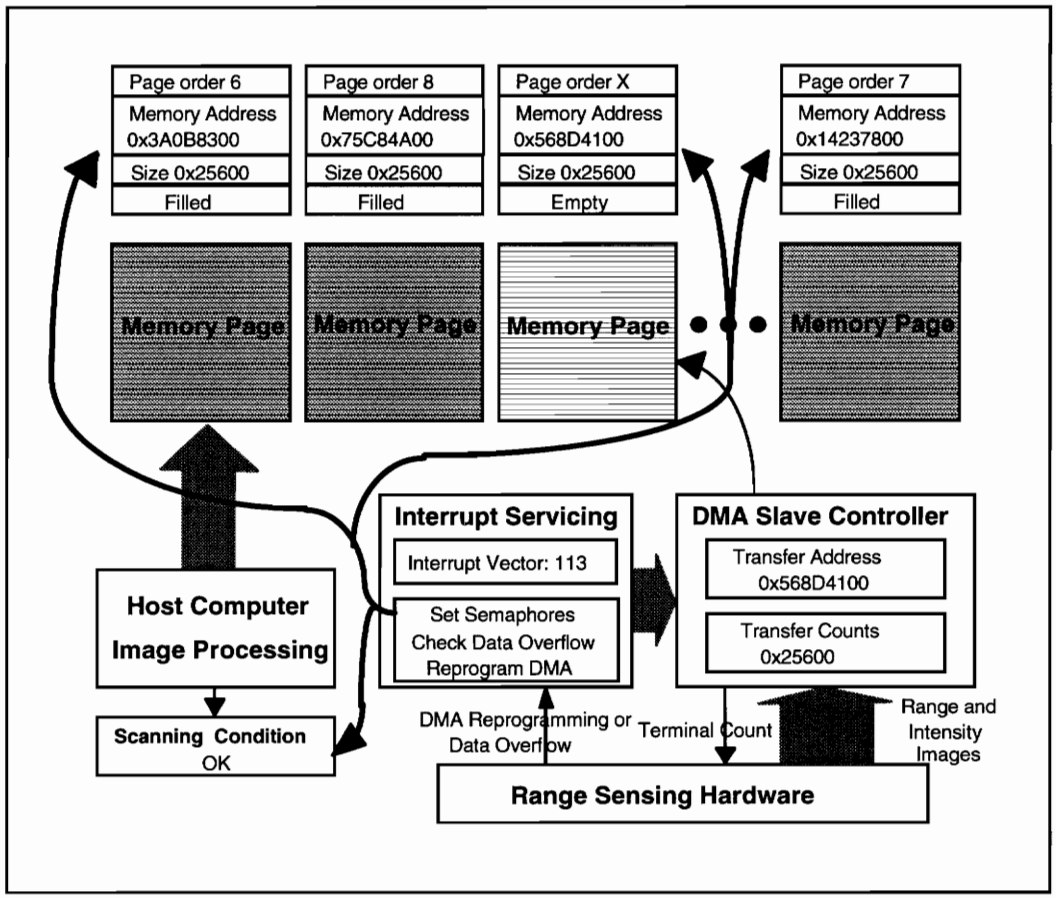


Figure 5.3 Memory management visualization.

5.2 System Registers

The range sensing hardware is a system composed of thirteen I/O ports, one 8 or 16 bit DMA slave channel, and one shared interrupt level on the MicroChannel bus. A set of POS registers configures the range sensing hardware for use on the MicroChannel bus, whereas allocated I/O registers control the system. Table 5.1 lists the offset, name, function description, and I/O port type of these I/O registers.

5.2.1 Command Register (I/O address offset 0)

The Command Register controls data flow within the range sensing hardware. This I/O register resides on the first of thirteen I/O addresses allocated by the configuration facility of POS registers. This register controls the system enable, system reset, the DMA transfer mode, look up table enable/disable, the switching of operation modes, the detection of a fresh frame, and the end-of-interrupt procedure. Detailed descriptions of control bits are listed in Table 5.2.

Bits 5,6,7 of the Command Register utilize pulse logic (one shot) and can be activated simply by writing the desired values once via an I/O WRITE operation to the Command Register without multiply toggling their registered values (to provide pulses). All bits of this register are negative logic (active low). Immediately after hardware power-on, this register is reset to a FF hexadecimal value automatically. This makes the hardware system disable all range sensing operations. To enable this register, bit 1 has to be driven low at least once via an I/O WRITE operation. The reason for this is to ensure that a random value will not be latched into this register during hardware power-on.

Table 5.1 Descriptions of system registers.

Offset	Register Name	Function Description	Type
0	Command Register	Contains the control command for the range sensing hardware.	I/O
1	Digital Image /Data Buffer	Simulates digital images when serving as an output port. Provides accesses to the video image buffer when serving as an input port in Frame Grabbing Mode.	I/O
2	Row Number Register (Low Byte)	Contains the row number in a video frame that is captured during the frame grabbing procedure.	O
3	Threshold /Row Number Register	Contains the threshold value for image thresholding. The lower four bits also serve as the upper four bits of the twelve bit row number value.	I/O
4	DMA I/O (Low Byte)	Serves as the I/O data port during DMA slave transfers. When operating in Computer Access Mode the host system has direct access to the FIFO memory.	I/O
5	DMA I/O (Upper Byte)	Same as above.	I/O
6	LUT Address (Low Byte)	Provides the Look Up Table input value (twelve bit size).	O
7	LUT Address (High Byte)	Same as above.	O
8	LUT I/O	Provides the setup values for the Look Up Table.	I/O
9	Camera Timing Simulation Register	Generates the simulated control signals for a virtual video camera.	O
A	Status Register	Contains the system status information.	I
B	Camera Select/ Intensity Processor Select Register	Selects the camera source (RS170 or MC9128) and the intensity processor.	O
C	Frame Rate Subsampling Factor Register	Contains the subsampling factor of the video frame rate.	O

Table 5.2 Command Register control bits.

Bit Number	Symbol	Function Description
Bit 0	I/-S	Low to enable the range scanning process, high for the computer accesses of I/O ports.
Bit 1	-LUT/-EN	Must be driven low once for hardware initialization. Low to disable Look-Up-Table mapping of range data and high to enable it during range scanning.
Bit 2	-IG	Low to activate the Frame Grabbing Mode.
Bit 3	-DMA EN	Low to reinstate DMA transfer requests stopped by the Terminal Count of the DMA slave controller.
Bit 4	-BURST	Low to enable the BURST DMA transfer mode, high to enable the SINGLE Mode.
Bit 5	-EOI	Low to perform End-Of-Interrupt procedure during interrupt servicing initiated by the range scanning hardware.
Bit 6	-Grab	Low to reset the fresh video frame detector for video frame grabbing.
Bit 7	-RES	Low to reset FIFO memory buffers.

5.2.2 Simulated Digital Image/Row Image Buffer (I/O address offset 1)

This I/O port serves two purposes. It can serve as an output latch for holding simulated digital image data generated by the host computer. When used in this manner the Timing Simulation Port (I/O offset 9) provides simulated camera control signals. Used together these ports provide a means for the host to diagnose possible hardware problems in the range sensing system.

It can also be used as an input port for providing the host computer a line of image data obtained while the range sensing was operating in the Frame Grabbing Mode. When used in this manner, an I/O READ operation issued by the host computer will activate the video buffer, an 8-bit FIFO memory, that is used by the frame grabbing unit to store a line

of image data. The read must be done while the range sensing system is in the Computer Access Mode. When the range sensing hardware is operated in the Frame Grabbing Mode, a fresh video line with the row number specified by the Row Number Register, is stored in the FIFO image buffer. After the hardware switches from the Frame Grabbing Mode to the Computer Access Mode, the host system can obtain this video line by sequentially reading this port.

5.2.3 Row Number Register (I/O address offset 2, part of offset 3)

A simplified version of a typical frame grabbing circuit has been designed that only allows one video line to be captured with each frame grabbing command. When the range sensing hardware is operating in the Frame Grabbing Mode, the host system provides a 12-bit row number of the video line that is to be grabbed to the Row Number Register. A frame capture command issued to the Command Register enables the hardware circuit to capture the video line with the specified row number. This frame grabbing circuit is capable of grabbing any video line in an image with the only restriction being that the camera must have less than 4,096 lines in a single frame (12 bits). The width of a single video line is limited by the depth of the FIFO memory used in the frame grabbing circuit. Currently the frame grabbing hardware is implemented using an 8 bit 512 byte deep FIFO memory. A video line that has less than 512 pixels can be stored in this memory.

The Row Number Register (I/O port 2) latches the lower 8 bits of the row number that the host system provides. The upper 4 bits of the row number are overlapped with the Threshold Register (I/O offset 3). There is no data contention problem caused by this overlapping. The row number is only valid in the Frame Grabbing Mode whereas the threshold value is only valid in the Range Scanning Mode.

The Row Number Register is a write-only I/O port. Hence, the value in the register cannot be verified by the host using an I/O READ operation. However, this verification can be performed by using simulated video signals. The host computer drives the virtual video source while activating the Frame Grabbing mode. Only the correct video line can be stored in the video frame buffer if proper video timing is provided (the timing format is described in Section 4.1.1).

5.2.4 Threshold/Row Number Register (I/O address offset 3)

A threshold value is used to convert gray scale images into binary images. Areas having a value of 1 on the binary image are the areas illuminated by the laser beam. The range processing unit receives this binary image a line at a time looking for the center pixel of the string of 1's to determine the range points. These range points are directly related to the depth values of the object's surface that is being imaged.

Theoretically the optimal threshold value is the value slightly above the dark level of the image background. The host system is responsible for calculating this value and providing it to the Threshold Register. However, setting the threshold to a value greater than the dark level can help reduce the effect of the image blur. There is, however, one drawback in using a higher threshold value. A higher value makes object surfaces with low reflectance invisible to the range sensing hardware. Section 6.3.2 discusses the consequences of using a high threshold value that can affect range generation.

The Threshold Register is both an output and input port for the host computer. Consequently the contents of this register can be verified using I/O commands. The lower four bits of this register also contain the upper four bits of the row number mentioned in last section. The number of the video line to be captured during frame grabbing is

therefore a 12-bit value (the Row Number Register is a 8-bit I/O register). This design eliminates the need for an extra I/O register to contain the extra 4 bits of the 12 bit line address.

5.2.5 DMA IO (I/O address offset 4,5)

This 16 bit I/O port holds the outcoming range and intensity data during DMA transfer when the range sensing hardware is operating in the Range Scanning Mode. The upper 8 bits hold intensity data and the lower 8 bits hold range data. An I/O READ operation to this port directly activates a 16 bit FIFO output regardless of the hardware operation mode. What happens during an I/O WRITE operation, however, is dependent on the operational mode that the hardware is in. When the range sensing hardware is operating in the Computer Access Mode, an I/O WRITE operation to this port enables the host system to write a word to the 16 bit FIFO memory. When the hardware is operating in the Range Scanning Mode, the 16 bit FIFO memory buffers the intensity and range data for DMA transfers.

This 16 bit port is mainly used for DMA slave transfers of range and intensity data. The DMA slave controller is programmed so that it performs "I/O to system memory" data transfers (I/O READ then MEMORY WRITE). The host computer can also program the DMA slave controller to perform 8 bit DMA transfers. In this way only the range image is transferred to the host computer's memory during DMA operations. Intensity images are discarded. Hence the transfer of intensity images is optional.

5.2.6 Look Up Table Address Register (I/O address offset 6, 7)

The reason for designing a hardware Look-Up-Table was explained in Section 4.2.5. This hardware unit is composed of an 8KX8 SRAM. The Look-Up-Table Address Register is used to store SRAM memory address. Pixel positions generated by the Range Processing Unit are sent to the address lines of this SRAM. The pixel positions are then directly converted to range data according to the memory contents of the Look-Up-Table.

The Look-Up-Table Address Register is a 16-bit I/O register is specifically designed to provide the SRAM with address values during Look-Up-Table setup. When the range sensing hardware is operating in the Computer Access Mode, address lines of the SRAM are connected to this I/O register. A bi-directional I/O port is connected to the data port of this SRAM that allows the host system to read and write to this SRAM at the address contained in this register. This bi-directional port is described in next section. When the hardware is operating in the Range Scanning Mode, the address lines of the SRAM is connected to the output port of the Range Processing Unit. The data lines of this SRAM is also enabled and produces the range data to be saved in the DMA FIFO buffer.

5.2.7 Look Up Table IO (I/O address offset 8)

When the range sensing hardware is operating in the Computer Access Mode, this 8 bit I/O port connects to the data port of the SRAM that serves as a hardware Look-Up-Table. When the entry address is latched onto the Look-Up-Table Address Register, I/O READ and I/O WRITE operations to the Look-Up-Table IO port perform MEMORY READ and MEMORY WRITE operations to this SRAM. The host system can therefore

set entries and verify memory contents in this way. When the range sensing hardware is operating in the Range Scanning Mode, this I/O port is disabled. The Mode Switching Unit attaches the address lines to this SRAM to the output lines of the Range Processing Unit and routes the output data lines to the DMA FIFO buffer.

5.2.8 Camera Timing Simulation Register (I/O address offset 9)

This port enables the host computer to generate simulated control signals of a virtual camera. The signal timing should follow the interface format described in Section 4.2.1. Table 5.3 shows the bit descriptions of this port. When the host system chooses to use this port as the camera signal source, it can perform circuit tests on the range sensing hardware. This port can serve as the camera source in both the Range Scanning and Frame Grabbing Mode, and can be chosen by issuing a byte command to the Camera and Intensity Processor Select Register (I/O address offset 11).

Table 5.3 Bit descriptions of the Camera Timing Simulation port.

Bit Number	Meaning
Bit 0	Pixel Clock
Bit 1	Inverted Pixel Clock
Bit 2	Line Enable (Active Low)
Bit 3	Frame Enable (Active Low)

When the host computer activates the Range Scanning or Frame Grabbing Mode, it also drives this register in a format described in Section 4.2.1. This port provides the control timing for the virtual video camera. Together with the Digital Image Register (I/O address 01) that provides the digital image signals, the host computer can input a known stream of image signals to the range sensing hardware. The output data buffers should

therefore contain processed data. The host computer can retrieve these data and compare them with correct ones. Doing so, the host computer can check the integrity of the range sensing hardware.

5.2.9 Status Register (I/O address offset 10)

This port contains the status of the range sensing hardware. Bit 0 and 1 define the status of the DMA FIFO memory. These two bits are used in the verification of DMA FIFO memory integrity. Bit 2 provides the interrupt identity of this system when several adapters on the MicroChannel bus share the same interrupt channel (IRQ 9). This function is required in the MicroChannel Architecture for interrupt sharing identification (see [29] for more details).

Table 5.4 Bit descriptions of the Status Register.

Bit Number	Name	Description
Bit 0	-EMPTY	DMA FIFO memory Empty Flag (active low).
Bit 1	-HF	DMA FIFO memory Half Full Flag (active low).
Bit 2	-IRQ	Interrupt request identity of the range sensing hardware (active low).
Bit 3	N.C.	
Bit 4	-OF	DMA FIFO Memory Overflow (active low).
Bit 5	-LINE END	Successful capturing of the Frame Grabbing Operation (active low).
Bit 6	-LINE ON	A video line being captured (active low).
Bit 7	-FM DET	Detection of a fresh video frame (active low).

Bit 4 indicates the status of data overflow in the DMA FIFO. When the interrupt latency is long and while the range sensing hardware is waiting for the interrupt to be serviced, the DMA FIFO memory may be overflowed, resulting in a loss of range data. If this situation occurs this bit will be driven low. Bits 5 to 7 carry the status of the frame

grabbing circuit. The host computer polls on this register to determine if the video buffer is ready while the hardware is performing frame grabbing. These bits tell the host computer if it should proceed with the reading of a video line by switching to the Computer Access Mode. Table 5.4 lists the bit descriptions of the Status Register.

5.2.10 Camera and Intensity Processor Select Register (I/O address offset 11)

This I/O port selects the camera source and intensity processing unit to be used in the range scanning process. Three camera sources are currently available: RS170 standard, EG&G MC9128, and the simulated virtual camera generated by the host computer (I/O port 9). Two intensity processing units are available: Center-Pixel Intensity Processing Unit and Total Reflectance Processing Unit.

A widely used RS170 camera has 30 video frames/second capability at 480 video lines per frame. The RS170 digitizer used in the range sensing hardware produces 384 pixels per line (digitizer model AD9502AM). The range generation performed by using a RS170 camera is therefore 30 profile lines per second at 480 range pixels per profile line, with 384 range levels. The data generation rate is equivalent to 14,400 range pixels per second.

The MC9000 series camera is capable of delivering high video frame rates. A MC9128 camera can generate up to 380 video frames per second, with 128X128 pixels per video frame. When it is used as an image sensor in generating range data, this camera can produce 380 profile lines per second at 128 range pixels per line, with seven bit range resolution. The data generation rate is equivalent to 48,640 range pixels per second.

Another MC9000 series camera, MC9256, can generate up to 150 video frames per second, 256X256 pixels per video frame. As with the MC9128 camera, this camera is

capable of generating 150 profiles per second at 256 range pixels per line, with eight bit range resolution. The data generation rate is equivalent to 38,400 range pixels per second.

Bit 4 of this port selects the intensity processor to be used in generating intensity images. Two intensity processing units are available in the range sensing hardware, the Center-Pixel Intensity Processing Unit and the Total-Reflectance Intensity Processing unit. Table 5.5 shows the bit values to be used in the selection of cameras and intensity processing units.

Table 5.5 Camera sources and intensity processing units selections.

Bit Position	Description
Bit 1,0 = 00	EG&G MC9128 Camera
= 01	Camera Timing Simulation Signals (Port 1,9)
= 10	RS170 Standard Camera
Bit 4 = 1	Enable Center Pixel Intensity Processor
= 0	Enable Total Reflectance Intensity Processor

5.2.11 Frame Rate Subsampling Factor Register (I/O address offset 12)

This port contains the value that subsamples incoming video frames. In applications where the object to be measured is moving fast and the requirement of the amount of range data is not stringent, a high frame rate camera such as MC9128 can capture the range images of this object with a minimum degree of blur. If obtaining all the range lines is not required, the subsampling of these video frames can reduce MicroChannel bus traffic. The function that this subsampling circuit performs is basically frequency division. With a value N latched to this port the subsampling circuit allows one video frame to pass through the range sensing hardware for every N video frames.

5.3 Utility Subroutine Listings

This section gives a brief introduction of available utility subroutines for controlling the range sensing hardware. These subroutines are written in C and can be used only on IBM PS/2 computers that have a DOS operating system. The source code of these subroutines is given in Appendix C.

Table 5.6 Listings of the utility subroutines for controlling the range sensing hardware.

Item	Name	Description
1	**void stereo_initilize(void)**	This subroutine initializes the Command Register. It must be executed before proceeding with any operations involving range sensing hardware.
2	**void stereo_close(void)**	This subroutine resets the range sensing hardware and restores it to a stationary state.
3	**void Cam_intensity_select(unsigned char cam_intensity)**	This subroutine selects the camera and intensity processor to be used in the range sensing process. The bit assignments of cam_intensity are listed in Table 5.5.
4	**void set_subsampling(unsigned char number)**	This subroutine sets the subsampling factor for reducing the camera frame rate. A value 0 set in the **number** keeps the range sensing hardware working at full speed while a value N lets the range sensing hardware process one video frame for every N frames.
5	**void interrupt interrupt_handler()**	This interrupt service routine enables the range sensing hardware to perform various kinds of memory management tasks. Three functions are performed in this service routine: checking the data overflow, reprogramming the DMA controller for different locations for image transfers, and setting semaphores to signal the status of the range scanning process (data overflow, process abnormally terminated, process successfully terminated, and order information of the received memory pages).
6	**void DMA_setup(int buffer)**	This subroutine programs the DMA slave controller for range/intensity data transfers. **buffer** represents the allocated blank memory page to be used in the DMA transferring of range/intensity data. These blank memory pages must be established before performing the range scanning.
7	**void DMA_destination (int buffer)**	This subroutine works with the interrupt service routine for DMA controller reprogramming during the memory management procedures. **buffer** is the empty memory page to be programmed as the DMA buffer.
8	**void set_threshold (unsigned char threshold)**	This subroutine sets the threshold value to be used in image thresholding.
9	**void scanner_enable(void)**	This subroutine activates the Range Scanning Mode .

10	**void FIFO_clear(void)**	This subroutine clears the FIFO memory buffers of both the range and intensity buffers, and reset the Command Register.
11	**int row_capture(unsigned char * row, int row_number, int size)**	This subroutine captures a video line with the row number specified by **row_number**. The scanned video line is stored in the **row** buffer. It is a data pointer. The size of the video line is specified in **size**. Return value 1 represents that the camera source is not ready, 0 means that the process is successful.
12	**void fgrab(int x_dim, int y_dim)**	This subroutine grabs a whole frame of images recorded by the camera and displays it on the screen with the camera frame size specified by **x_dim** and **y_dim**. The range sensing hardware is operating in the Frame Grabbing Mode when executing this subroutine.
13	**void setup_LUT(int maximum, int minimum)**	This subroutine sets Look-Up-Table entries. The **maximum** and **minimum** parameters are set according to the occupied range of bright positions in a video line. The Look-Up-Table expands this range of video positions uniformly to a full eight bits.
14	**void find_parameters(int* maximum, int* minimum, unsigned char* threshold)**	This subroutine helps users to manually select the threshold value, the Look-Up-Table range to be expanded, and the verification of hardware signal processing. The range sensing hardware is operating in the Frame Grabbing Mode when executing this subroutine.
15	**void create_buffers(void)**	This subroutine creates available memory pages to be used in range and intensity data transfers. A memory page must have contiguous memory address for DMA operations. Address locations between, pages, however, do not need to be contiguous.
16	**void free_buffers(void)**	This subroutine frees the memory buffers created by the **create_buffer** subroutine.
17	**void interrupt_setup(void)**	This subroutine resets semaphores passing between the main program and the interrupt service routine. These semaphores represent the information signaled by the range sensing hardware. Concurrent data processing of the range images can be achieved this way. These semaphores are listed in the definition file of Appendix C.

5.4 Programming Examples

The programming example in Figure 5.4 illustrates how the utility subroutines can be utilized to control the range sensing hardware. In this example the program brings up a graphic screen with several on-screen instructions for adjusting system parameters. This is achieved by calling the **find_parameters** subroutine.

```
/* This program provides adjustments of the ranging hardware parameters.
  All instructions and simulations are provided on screen */
#include <stdio.h>
#include <scanner.h>
void main(void)
{
unsigned char threshold;
int maximum, minimum;

set_subsampling(0);          /*  reset the frequency divider */
stereo_initialize(); /* initilize the range sensing hardware */
cam_intensity_select(0);  /* select MC9128 as the camera source, and the Total
                             Reflecance Intensity processor to use */
FIFO_clear();                /* clear DMA FIFO buffers */

/* show ranging parameter information on screen */
find_parameters(&maximum, &minimum, &threshold);
set_graphics_mode(3);  /* return to text mode */

/* set entries of the Look-Up-Table by expanding the value between
    maximum and minimum */
setup_LUT(maximum, minimum);

set_threshold( threshold);  /* set the threshold value chosen in the find_parameter
                                                    subroutine */
stereo_close();
   }
}
```

Figure 5.4 A programming example of utility subroutines.

The program first initializes the range sensing hardware by calling the **stereo_initilize** subroutine. The frame rate subsampling unit is disabled by writing 0 to the register. The video source chosen by the **cam_intensity_select** subroutine is MC9128 camera. This program executes the **find_parameters** subroutine that brings up a graphic screen containing on-screen instructions for defining parameters to be used. These parameters are the binary threshold value and the maximum and minimum range positions of a video line. The interval between maximum and minimum of a video line represents the range of object depth allowed in generating the range values. The actual video waveform is displayed on the screen along with simulated hardware processing results. The **setup_LUT** subroutine then sets up the Look-Up-Table by expanding this interval into full 8-bit values. All subsequent range positions in this interval are mapped to 8-bit values during the range scanning process.

The application programs given in Appendix D to illustrate the general procedures for using the utility subroutines to control the range sensing hardware. The first program activates the range sensing hardware to start collecting range and intensity data. This program demonstrates the possibility of doing real-time image processing using the memory management scheme described earlier. The range scanning process works concurrently with the data storing process (stored into the peripheral disk). As described in Section 5.4, the interrupt service routine sets semaphores on the main program when a memory page is filled with data through DMA transfers. The main process monitors these semaphores and pushes the filled page of memory into peripheral disk and then resets these semaphores.

5.5 Summary

This chapter explains programming methods for initializing and activating the range sensing hardware. Bit descriptions of system registers are described in detail. For real-time operations a memory management scheme is described and its utility demonstrated. The first programming example in Appendix D demonstrates this real-time processing capability. Range and intensity images are collected into system memory, average filtered, and stored into a peripheral diskette in real-time. Experimental results are given in Chapter 7.

The utility software routines provide procedures for controlling the operation of the range sensing hardware. These utility procedures can be used to perform most if not all the required system management functions. Hence, a user of this hardware does not have to burden himself with all the implementation details.

Chapter 6
Measurement Error Analysis

This chapter analyzes measurement errors that can occur in the range sensing system. These errors are examined from both a theoretical as well as practical points of view. The theoretical analysis assumes infinite resolving resolution of the image sensor, perfect optical focusing, and noiseless signal processing. The practical analysis discusses possible error sources that can affect the resolution performance of the range sensing system.

While the theoretical analysis places a performance limit on the range sensing technique, it also provides a foundation for possible performance improvements. The result of this analysis indicates that the optical components place physical limits on the accuracy that can be obtained. This analysis complies with the well known Shannon's sampling theorem, which states that to discretely sample an analog waveform, the sampling rate should be, at least, a factor of two higher than the frequency bandwidth of the waveform in order to avoid aliasing in the time domain (see, e.g., [34]). In the theory presented in this chapter the laser beam diameter plays the role of a sampling period and the spatial domain plays the role of the time domain. This range measurement theory, therefore, concludes that, for a range sensor using a laser beam of diameter L, the theoretical limit of measurement accuracy is $2L$.

Practical processing errors are introduced by non-ideal signal processing circuits and optical misadjustments. However, reducing these errors require a trade-off between the application requirements and system performance. The following chapter explains the

range sensing system limitations and describes possible methods for improving the performance of this system.

6.1 Theoretical Error Analysis

The basic design of an optical system for generating range information was discussed in Chapter 3. The principle of extracting range information is based on a triangulation method. Figure 6.1 illustrates the imaging geometry of the range sensing optics.

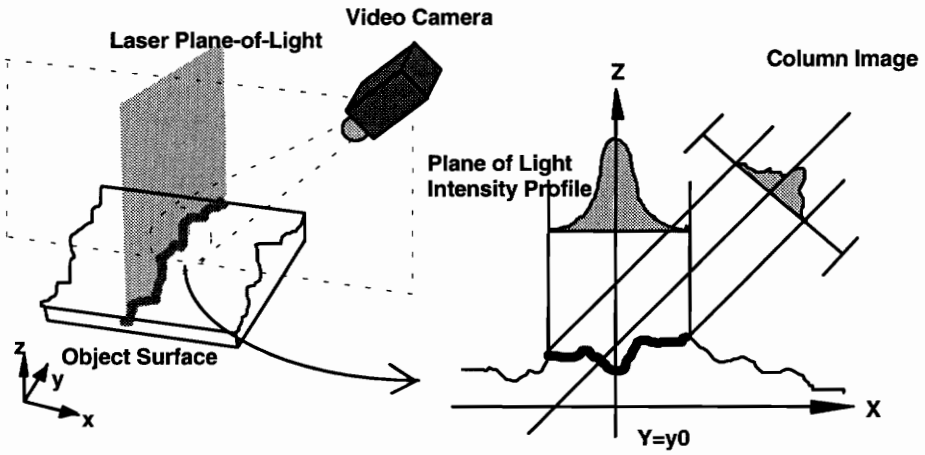


Figure 6.1 Imaging geometry of the range sensing system.

Several assumptions are made to facilitate the derivation of measurement accuracy. These include the assumptions (1) that the working environment is completely dark, (2) that the image sensor is noiseless, (3) that the measured object is continuous in shape, (4) that the image sensor has continuous photocells, and (5) that the illuminated scene is perfectly focused onto the image sensor. Figure 6.2 illustrates the center-pixel position algorithm for extracting range information. This method is implemented in hardware and forms the backbone of the range sensing system.

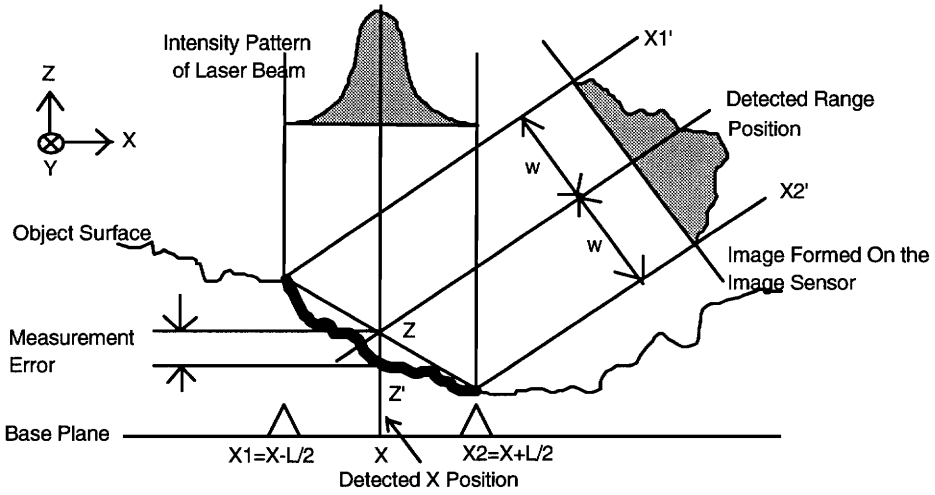


Figure 6.2 Range detection theory.

As shown in Figure 6.2, the projected laser beam serves as an acquisition medium for extracting depth information. The size of the beam, as can be seen, is a sampling window in retrieving the range information from diffuse reflective object surfaces. Assume the width of the beam is L and the shape of the object surface is described as $z=f(x,y)$, where y is the axis perpendicular to the plane shown in Figure 6.2.. Disregarding the intensity profile of this beam, the projected beam shines on a segment of the object surface from *X-L/2* to *X+L/2*. With a perfect thresholding circuit, the measured range value is expressed as

$$\hat{z}(x,y_0)=\frac{f(x+L/2,y_0)+f(x-L/2,y_0)}{2} \tag{6.1.1}$$

for a fixed y_0. The measurement error is

$$Error(x,y_0)=f(x,y_0)-\frac{f(x+L/2,y_0)+f(x-L/2,y_0)}{2} \tag{6.1.2}$$

The surface roughness of an object is indicated by the bandwidth of its spatial frequency spectrum, which is the two-dimensional Fourier transform of $f(x,y)$ using spatial frequencies instead. The spatial frequency spectrum of a rough surface has a wide

frequency bandwidth and is concentrated on the high frequency side. Since the signal processing scheme used in depth retrieval is one dimensional, we will hold y constant in the following discussion, i.e., we only process a column image (a video line) at a time. To deduce of the relationship between depth resolution and laser beam width, a spatial Fourier transform is applied to the measurement error function *Error(x,y)* in Equation (6.1.2). It is shown as follows:

$$E(\omega, y_0) = \Im\{Error(x, y_0)\} = F(\omega, y_0) - \frac{(F(\omega, y_0)e^{i\omega\frac{L}{2}} + F(\omega, y_0)e^{-i\omega\frac{L}{2}})}{2}$$

$$= F(\omega, y_0)(1 - \frac{e^{i\omega\frac{L}{2}} + e^{-i\omega\frac{L}{2}}}{2})$$

$$= F(\omega, y_0)(1 - \cos(\omega\frac{L}{2})) \qquad (6.1.3)$$

where

$$F(\omega, y_0) = \Im\{f(x, y_0)\} = \int_{-\infty}^{\infty} f(x, y_0)e^{-i\omega x}dx \qquad (6.1.4)$$

is the one-dimensional Fourier transform of the object surface $z=f(x,y)$ for $y=y_0$, and $E(\omega, y_0)$ is the spatial Fourier transform of $Error(x,y_0)$, i.e.,

$$E(\omega, y_0) = \Im\{Error(x, y_0)\} \ . \qquad (6.1.5)$$

The signal to noise ratio of this range detection scheme is given by

$$SNR = \frac{\int_{-\infty}^{\infty} F^*(\omega, y_0)F(\omega, y_0)d\omega}{\int_{-\infty}^{\infty} E^*(\omega, y_0)E(\omega, y_0)d\omega}$$

$$= \frac{\int_{-\infty}^{\infty} F^*(\omega, y_0)F(\omega, y_0)d\omega}{\int_{-\infty}^{\infty} F^*(\omega, y_0)F(\omega, y_0)(1 - \cos(\omega\frac{L}{2}))^2 d\omega} \qquad (6.1.6)$$

As shown in Equation (6.1.6), the signal to noise ratio (*SNR*) is dependent on the spectral distribution of the surface shape function *f(x,y0)*. For an object surface that is mostly smooth when compared to the laser beam width, the signal to noise ratio is high. This observation is based on the multiplication factor $(1-\cos(\omega L/2))^2$ in the denominator. For low spatial frequencies the factor is approximately zero. If the spectral distribution concentrates on the low spatial frequency, the resulting *SNR* is high as a result of this multiplication factor.

Equation (6.1.3) also infers an interesting fact: the ratio of *E(w,y0)* and *F(w,y0)* is dependent only on the spatial frequency ω. This fact indicates that the measurement error at a specific spatial frequency is proportional to the spatial component of the signal spectrum by a factor dependent only on the spatial frequency. This relationship is expressed as

$$E_{ratio}(\omega) = \frac{E(\omega, y_0)}{F(\omega, y_0)} = 1 - \cos(\omega \frac{L}{2}) \qquad (6.1.7)$$

Equation (6.1.7) agrees with a practical observation about the depth measurement accuracy. When the laser spot is large, or L is large, the measurement error is great because all high frequency variations of the object surface are "missed" by the laser beam. The beam size must be small in order to measure fine variation. This can be seen from Equation (6.1.7) by setting $\omega = \pi/L$. At this ω the measurement output is zero since $(1-\cos(\pi/2)) = 0$. When L is large, the corresponding ω is small. The physical meaning of this observation is that the larger the laser beam diameter is, the lower of the spatial frequency that the range sensor can detect.

This "sampling beam" plays the most important role in precision evaluation of range data. According to Equation (6.1.1), the measurement spectral intensity of an object surface *f(x,y)* is derived as

$$\hat{F}(\omega, y_0) = \Im\{\hat{z}(x, y_0)\}$$

$$= \Im\{\frac{f(x+L/2, y_0) + f(x-L/2, y_0)}{2}\}$$

$$= F(\omega, y_0)(\frac{e^{i\omega\frac{L}{2}} + e^{-i\omega\frac{L}{2}}}{2})$$

$$= \cos(\omega\frac{L}{2})F(\omega, y_0) \qquad (6.1.8)$$

The depth measurement can therefore be treated as a filter with $\cos(\omega L/2)$ as the transfer function and the object surface is the input.

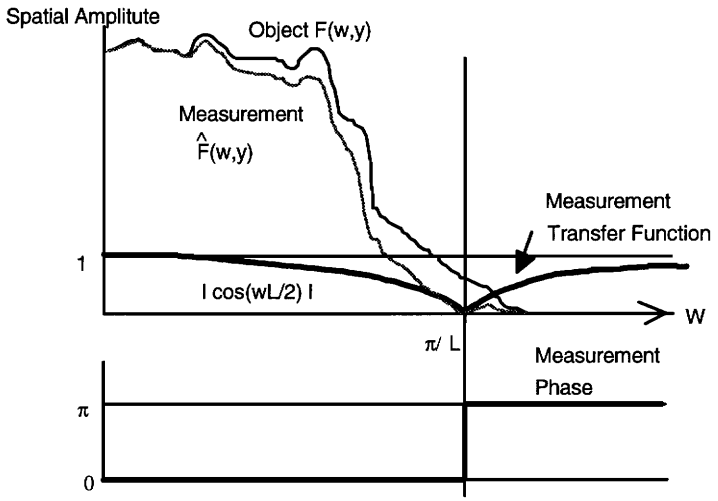


Figure 6.3 Actual and measurement spectral distributions.

Figure 6.3 depicts a phenomenon that is analogous to Shannon's sampling theorem. According to Equation (6.1.8), for a range sensing system with a laser spot size L, the critical measurement bandwidth is π/L. At this frequency the measurement output is zero. If this system is to measure a surface object that has a spectral intensity pattern exceeding this range, an alias sampling phenomenon will happen ($\cos(\omega\ L/2)$ becomes negative). The spatial wavelength corresponding to the critical bandwidth is $2L$. This is equivalent to saying that a laser beam with a spot size L has the maximum resolving capability of

detecting a surface waveform with a spatial wavelength *2L*. This result agrees with the Shannon's sampling theorem which states that in order to discretely sample a signal with frequency bandwidth *BW*, the sampling function must have a repetitive frequency greater than *2BW*.

6.2 Errors Caused by Binary Hysteresis Thresholding

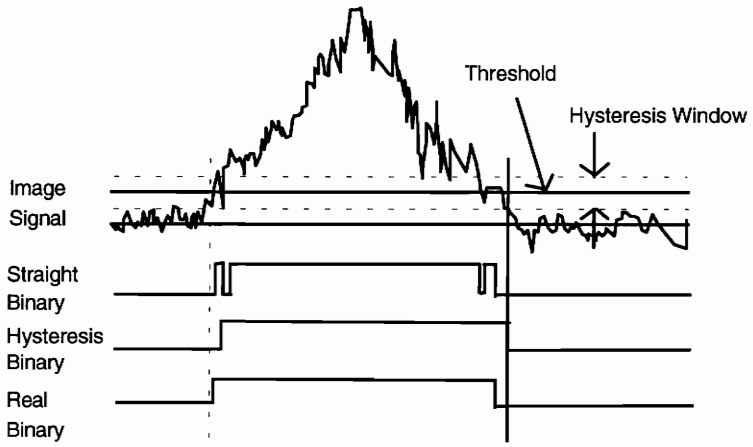


Figure 6.4 Effects of applying binary hysteresis thresholding.

To identify areas of images illuminated by the laser beam, a threshold value is provided for converting gray scale images into binary images. Section 4.2.3 described the implementation of this thresholding operation in the Range Processing Unit. This hardware unit receives gray scale images produced by the image digitization unit, generates binary streams according to the specified binary threshold, and calculates center pixel positions for range detection. To create the binary stream a binary hysteresis thresholding technique is employed. As mentioned, the processing unit only allows one bright group of contiguous pixels in each video line. This contiguous string of pixels in this string are all assigned the value 1 by the thresholding operation. If more than one string of 1's are detected, the range hardware unit only accepts the first string and

computes the center pixel position for only this string. The main reason for this is hardware simplicity. Nevertheless, it is because of this that edge jittering is not allowed in generating binary images.

Figure 6.4 explains the reason for employing binary hysteresis thresholding. As shown in the figure, the binary values around edges oscillate between 0 and 1 if a straight-forward binary thresholding is used. This is mostly caused by electronic noise inherent in the image sensor and in the analog-to-digital converter. Because of this noise, gray scale pixels with intensity values around the threshold tend to vary randomly. To overcome this phenomenon, a digital hysteresis mechanism was added to the thresholding circuit. Section 4.2.3 provides an explanation of the hysteresis window. This hysteresis filter produces binary strings with smooth edges. However, because of the hysteresis, the number of pixels in a string of 1 can be different from the ideal situation without hysteresis. In Figure 6.4, the string of 1's produced by the hysteresis thresholding is also shifted in position. The only method for alleviating this effect is to reduce the size of the hysteresis window. The size of this window also determines the noise immunity of the thresholding operation. A larger window gives better noise immunity. Hence, a window that approximately covers the maximal magnitude of the noise is the best size to use.

6.3 Errors Caused by Defocused Images in the Vertical Axis

In the discussion of the range detection theory (Section 6.1), a question may be raised: since the diameter of the laser beam is known, and since the two edges of the bright spot is the range values of these edges when there is a perfect bright detection circuit, why go to the trouble of finding the center pixel? This question, however, is based on the assumption that the laser illuminated image is perfectly focused onto the image

sensor. Blurred images, which occur in most situations, will cause serious range measurement error. Figure 6.5 illustrates this situation.

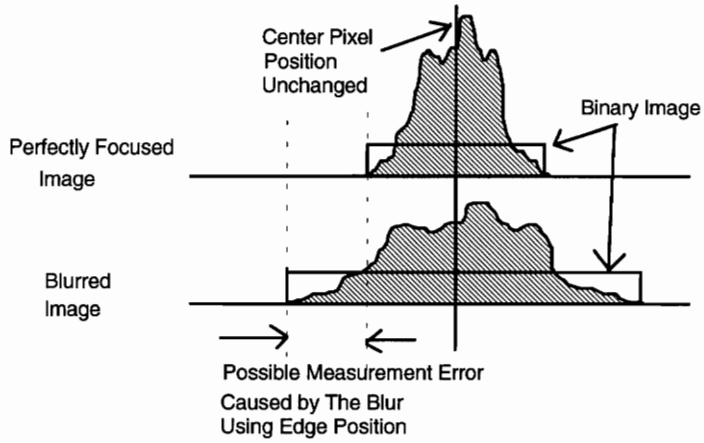


Figure 6.5 Measurement errors caused by bright edge detection.

As seen in Figure 6.5, the blur tends to spread in both directions evenly. If the bright edge position is used to determine range, blur will cause large range errors. This is illustrated in Figure 6.4 by the interval between the dotted lines. The center pixel position based range detection scheme, however, tends to stay the same regardless of the blur. This center pixel scheme can also reduce the error introduced by the hysteresis thresholding by a factor of two. If an additional bright pixel is added to the binary stream due to the noise, the measurement error is half of the pixel width. This scheme is therefore more robust than the edge detection scheme.

6.4 Errors Caused by High Threshold Levels

The extent of error caused by high threshold levels when using the Center-Pixel Algorithm depends on the roll-off rates of image signals near the ends of a illuminated spot. Figure 6.6 illustrates an example of range detection using different thresholding levels when the image signal is highly asymmetrical.

As shown in Figure 6.6, part (A) represents a case where a low threshold level is used. The detected center pixel deviates little from the true center pixel. Conceptually this is true since the lower the threshold is, the better it can determine locations where the object is illuminated by a laser beam, even when the object is illuminated by a weak beam. When a high threshold level is used, as in part (B), the deviation of center pixel from the true value depends on the difference of signal roll-off along each side of the "central" peak. For a symmetrical image signal, the difference is zero and therefore there is no deviation. However, for a signal that is highly asymmetrical, the deviation can be great and thus results in a large measurement error. It is therefore a requirement that the thresholding level be kept as low as possible.

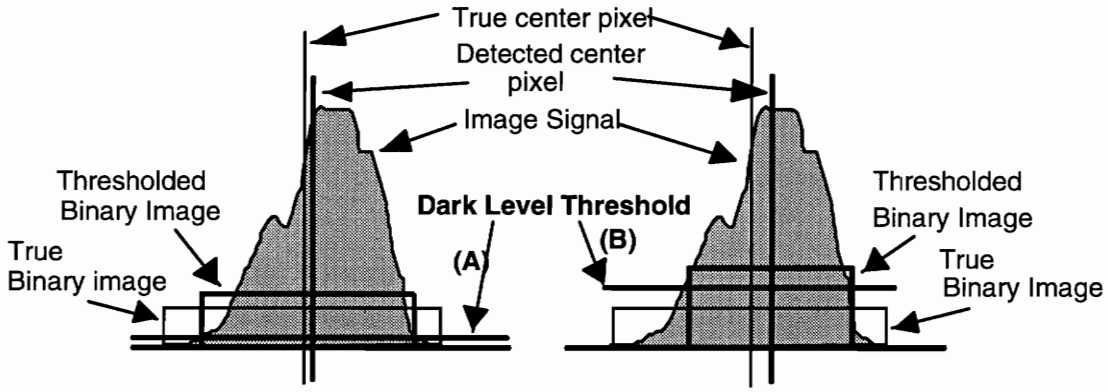


Figure 6.6 Effects of using different dark levels in range detection.

Selecting an optimal threshold level depends largely on the magnitude of electronic noise accumulated by the image sensor and its signal processing circuits. A threshold level should be able to discriminate noise from light signals. This may not be easy if the noise has a large variance. However, for objects with smooth surface reflectance (low reflectance variation for all areas with size comparable to the laser line width), the threshold level is not critical since the difference in signal roll-offs is close to zero. The

concern should be whether the system can detect locations where the reflected light is very weak.

6.5 Cross Field Image Focusing

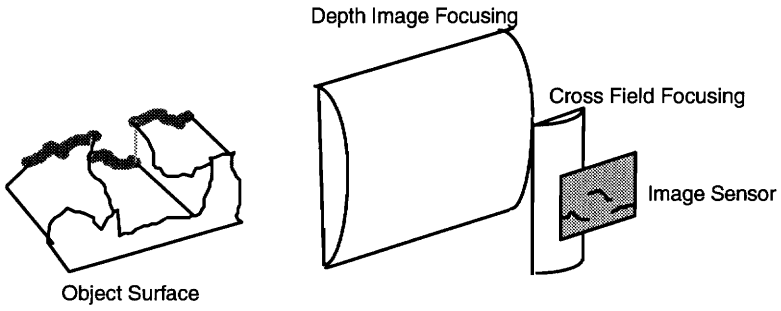


Figure 6.7 Image focusing of edges.

Section 6.3 explains the robustness of center-pixel position algorithm with regard to blurred images as compared to other algorithms. This robustness, however, is only applicable to video lines. For cross field images, the blur can protrude across video lines and result in inconspicuous edges. Figure 6.7 illustrates a situation when an object's surface has sharp edges. In this case, the first cylindrical lens magnifies and focuses depth images. The blur caused by this lens has little effect on the range detection, as explained in Section 6.3. The second cylindrical lens focuses cross field images onto the image sensor. The blur caused by this lens, however, greatly affects the system's ability to accurately detect sharp edges.

Figure 6.8 illustrates the importance of cross field focusing. This figure follows the one shown in Figure 6.7. As shown in this figure, the well-focused stripe-of-light image has clear profile "jumps" while the poorly focused image has protruding blur around the points of discontinuity. This causes detection errors since the Range Processing Unit only processes the first bright string in each video line. Using the pixel convention shown in the figure, the width of the "crack" of the object surface in (A) is five pixels while it is

only three pixels in (B). This is the reason that we cannot detect small cracks when the image is blurred in the cross-field direction.

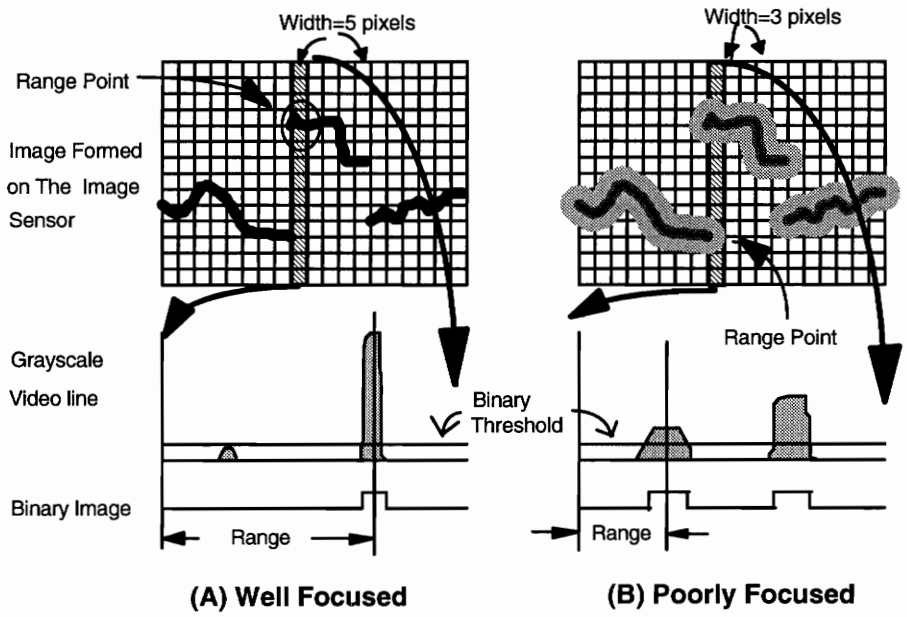


Figure 6.8 Errors caused by the blur of cross field focusing.

A way of alleviating the range detection error caused by the blur in the cross-field direction is to raise the binary threshold. The extent of blur across sharp object edges can be lessened when a greater threshold value is used. This approach, however, has one significant drawback. The range positions of surfaces with low reflectance will be ignored due to the high dark level threshold.

The best approach for solving cross field focusing is to redesign the imaging lens system. Currently the image focusing is performed by placing two cylindrical lenses in front of the camera, with each lens focusing one of the imaging axes onto the image sensor. Spherical aberration is introduced into the imaging process due to the use of a constant curvature lens. Elaborate combination lenses may be required in order to improve the focusing quality.

6.6 Summary

This chapter contains both the theoretical and practical analysis of range measurement errors related to the range sensing system described in this thesis. The theoretical analysis shows that the measurement accuracy is directly related to the projected laser line width. This relationship states that to achieve the measurement accuracy of L, the laser beam diameter has to be $L/2$. In terms of signal processing, this relationship resembles Shannon's sampling theorem. The laser beam diameter represents the sampling period in Shannon's theorem while the object surface is a two-dimensional signal.

This chapter also discussed sources of errors caused by implementation methods used in the prototype. The binary hysteresis thresholding shifts and changes range positions. As explained, this can be remedied by reducing the width of the hysteresis window. The minimum width that this window can be reduced to, however, is constrained by the maximal magnitude of the noise. To achieve good binary thresholding quality, the image digitization circuit has to be carefully designed to minimize the amount of electronic noise.

The measurement errors caused by the imaging optics is more severe than that caused the signal processing hardware. While the optics can be designed to somewhat alleviate the severity as explained in Section 6.3, the blur introduced by the imaging optics is still the main reason for not being able to detect small cracks on object surfaces. Chapter 8 presents a scheme that may improve this situation. However, to achieve large depth-of-field range sensing as described in Section 3.2.1, an imaging module composed of two cylindrical lenses is still the best choice. The task then is to elaborate on the lens combination design to improve the imaging quality of this lens combination.

Chapter 7
Experimental Results

This chapter contains experimental results produced by the prototype. Images of range, center-pixel intensity, total-reflectance intensity, and 3-D mesh are provided for comparison. These pictures were generated by the range sensor using a MC9128 camera. The camera is operating at 380 frames per second. Due to the limited speed of the translation stage available on the experimental setup, the experiment records only one out of twenty thickness profiles produced by the range sensor. A real time averaging (software) filter is implemented to smooth images by averaging 20 scanned thickness profiles into one for every 20 lines. The results are stored on the hard disk concurrently with the range and intensity data generation. Figure 7.1 shows the data flow chart of the scanning process.

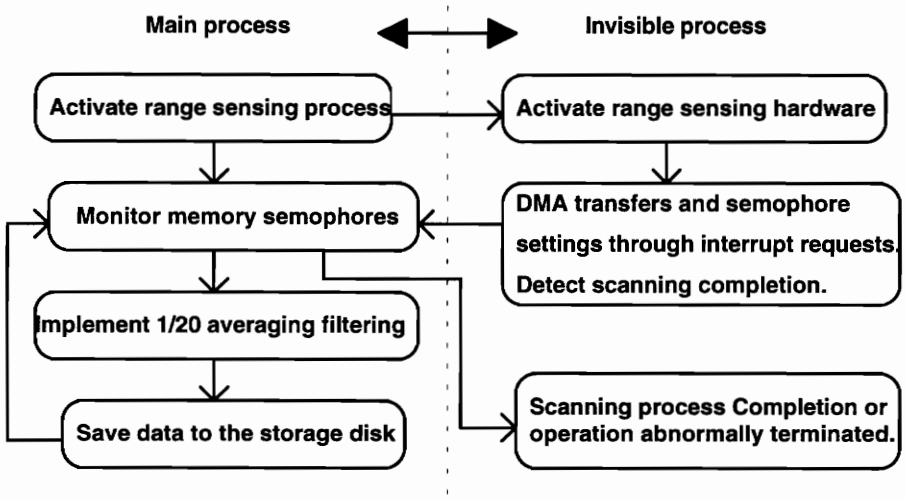


Figure 7.1 Real time filtering of the range/intensity images.

The optical imaging system has a viewing width of 4 inches. The depth measurement is adjusted to contain 1 inch depth range. The hardware Look-Up-Table is setup to perform contrast enhancement only. Geometrical data correction, as described in Section 3.2.1, is not implemented in these experiments since the ratio of the working distance of the camera to the magnitude of height variation of lumber is small.

The source code for the programs that were created to perform these experiments are given in Appendix D. Section 5.4 contains descriptions of these programs. These experiments clearly demonstrate the real-time processing capability of the memory management scheme discussed earlier. In these experiments, the range sensing hardware is installed on a IBM PS/2, Model 80 computer. The real time processing algorithm loaded in the computer during range scanning is an average filter. Section 5.1.4 explains the range scanning process. As demonstrated, the range sensor produces 380 thickness profiles per second at 128 pixels per profile. The software performs average filtering of images and at the same time saves processed images to peripheral storage disks. We also can save the processed images to a floppy disk during range scanning without losing data, if the length of the image is not too long.

Figures 7.2-5 show 3D plots of scanned objects. These mesh plots provide clear illustrations of the capabilities of this system. Figures 7.6-11 show "gray scale" versions of range imagery and the accompanying intensity imagery. These images have been smoothed by the real-time averaging filter mentioned above to improve image quality. Different samples are shown to illustrate the performance of the range sensor. Gray scale intensities of the range image indicate surface heights of the corresponding pixels.

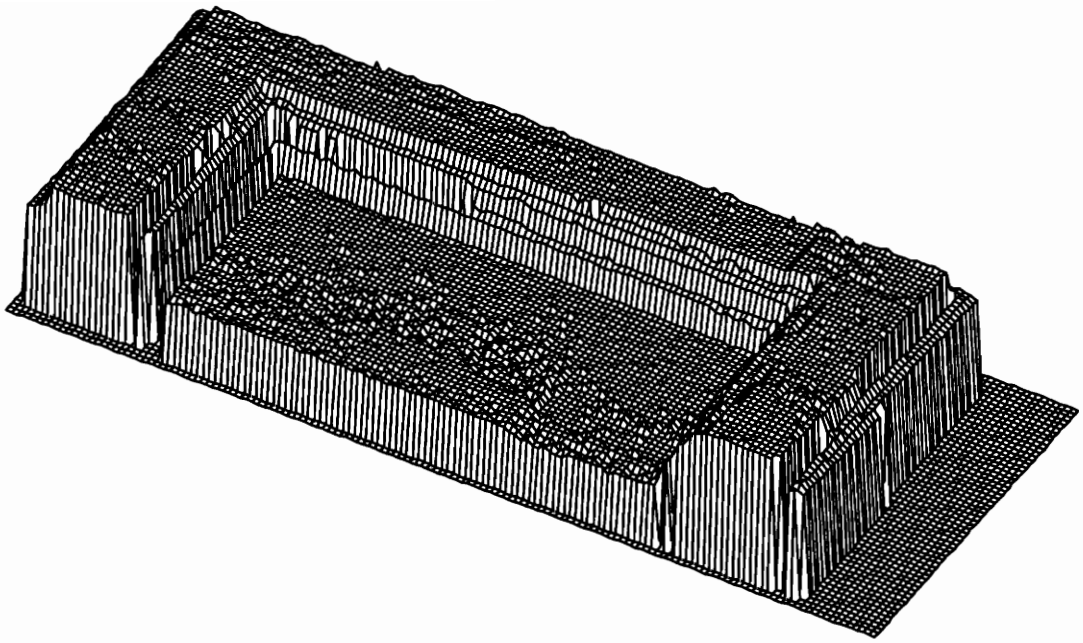


Figure 7.2 3D mesh plot of a cabinet frame.

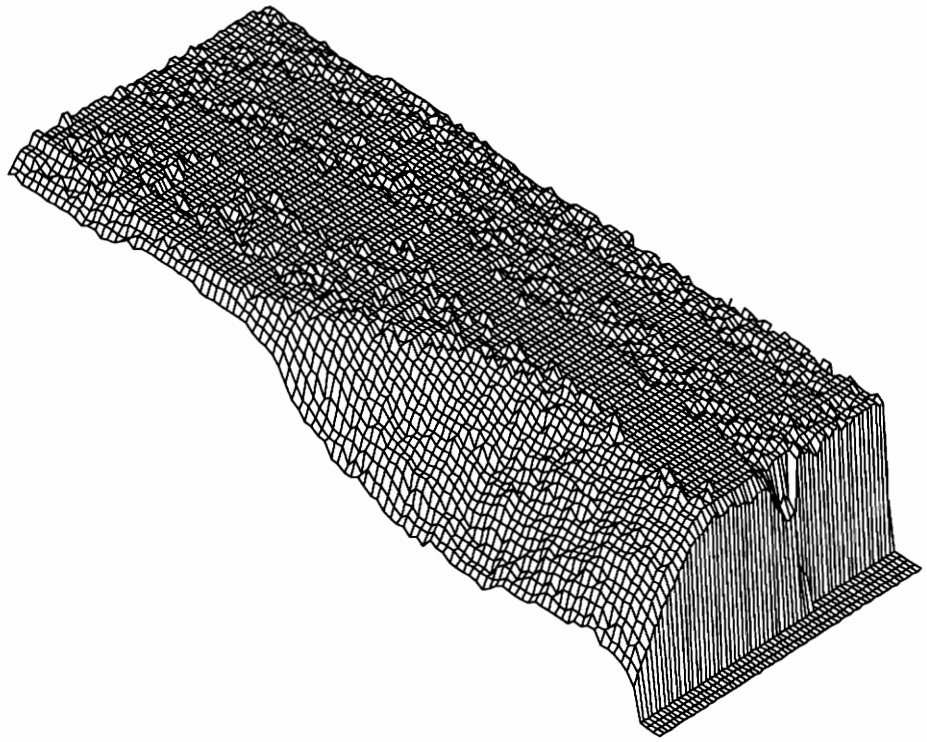


Figure 7.3 3D mesh plot of a lumber with wane.

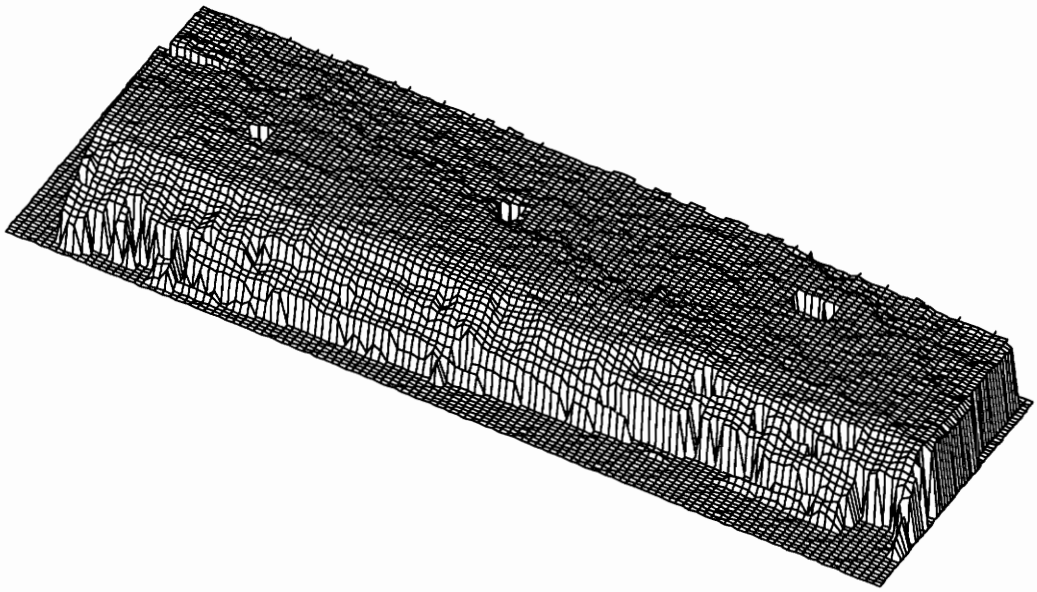


Figure 7.4 3D mesh plot of a lumber with holes, wane and a split.

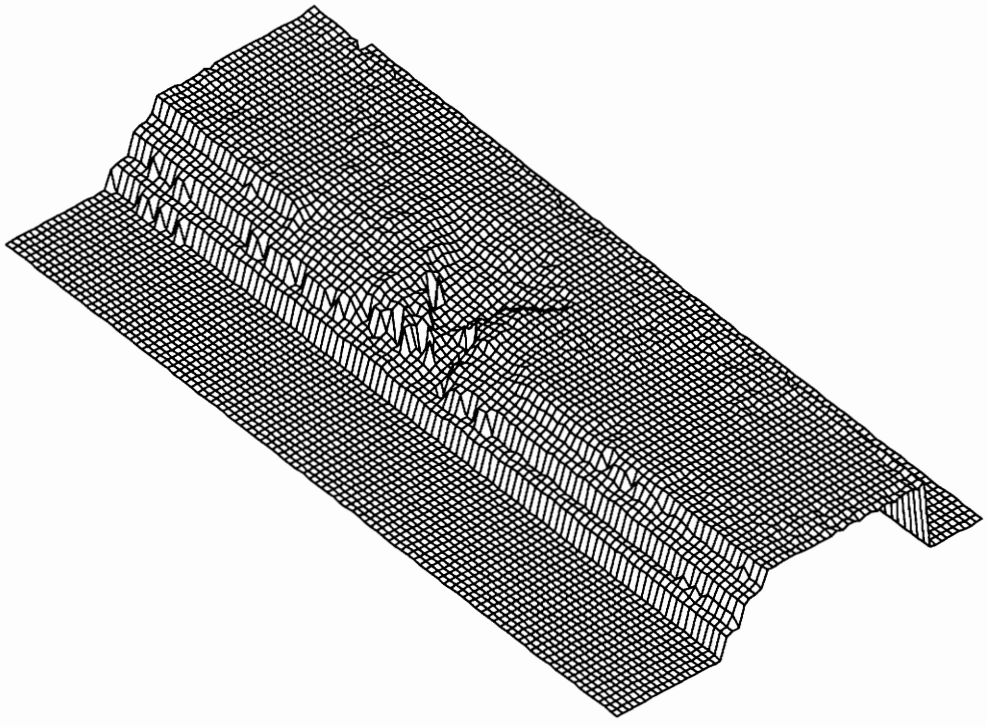


Figure 7.5 3D mesh plot of a molded lumber with defects.

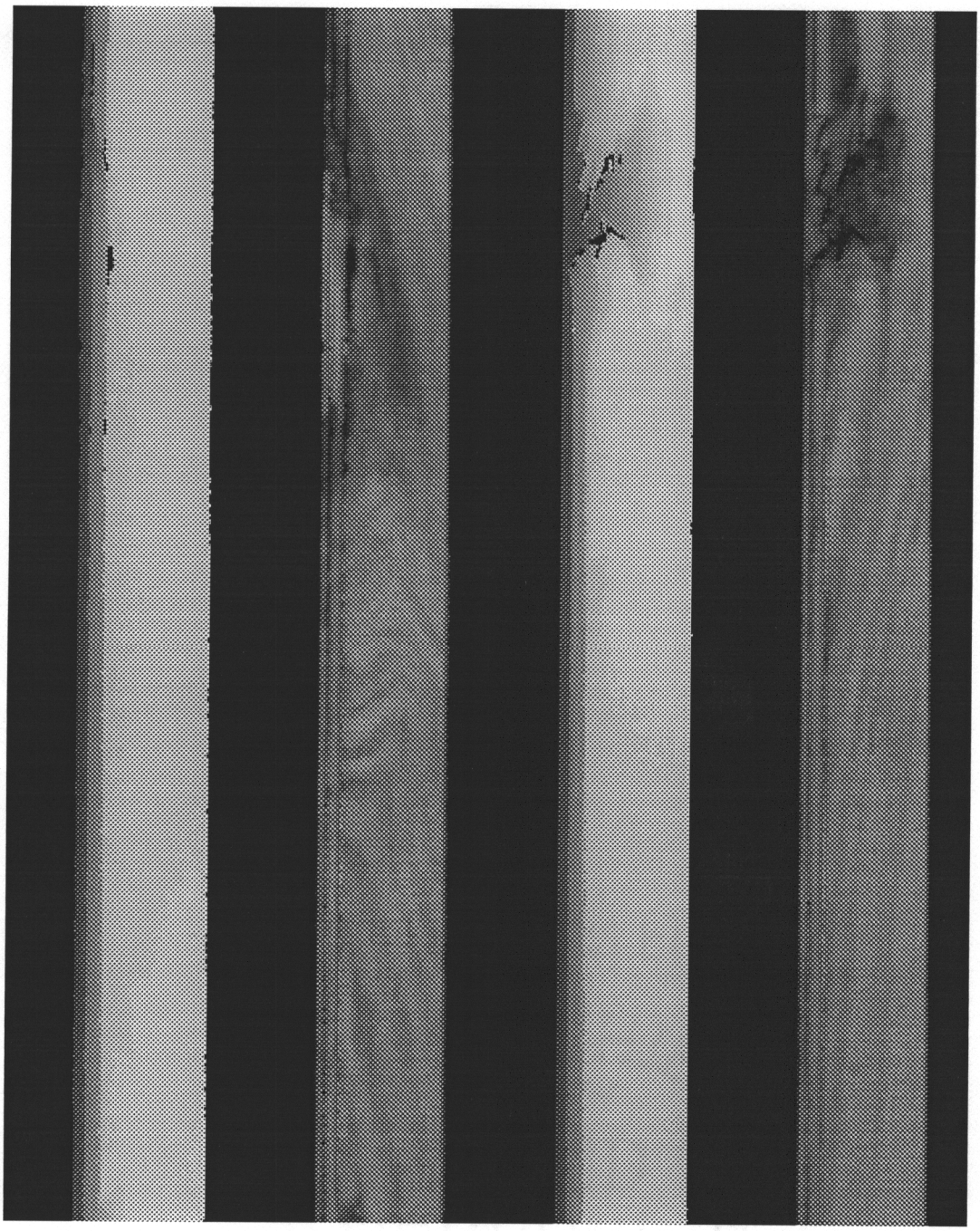


Figure 7.6 From left to right: (A) range in registration with (B) intensity, (C) and (D) are continuous parts of (A) and (B). These registration pair images are generated by the Range and Total-Reflectance Processors.

Figure 7.7 Comparisons of registered image pairs. From top to bottom: (A) range in registration with (B) intensity (Total-Reflectance), (C) range in registration with (D) intensity (Center-Pixel Intensity). Board B.

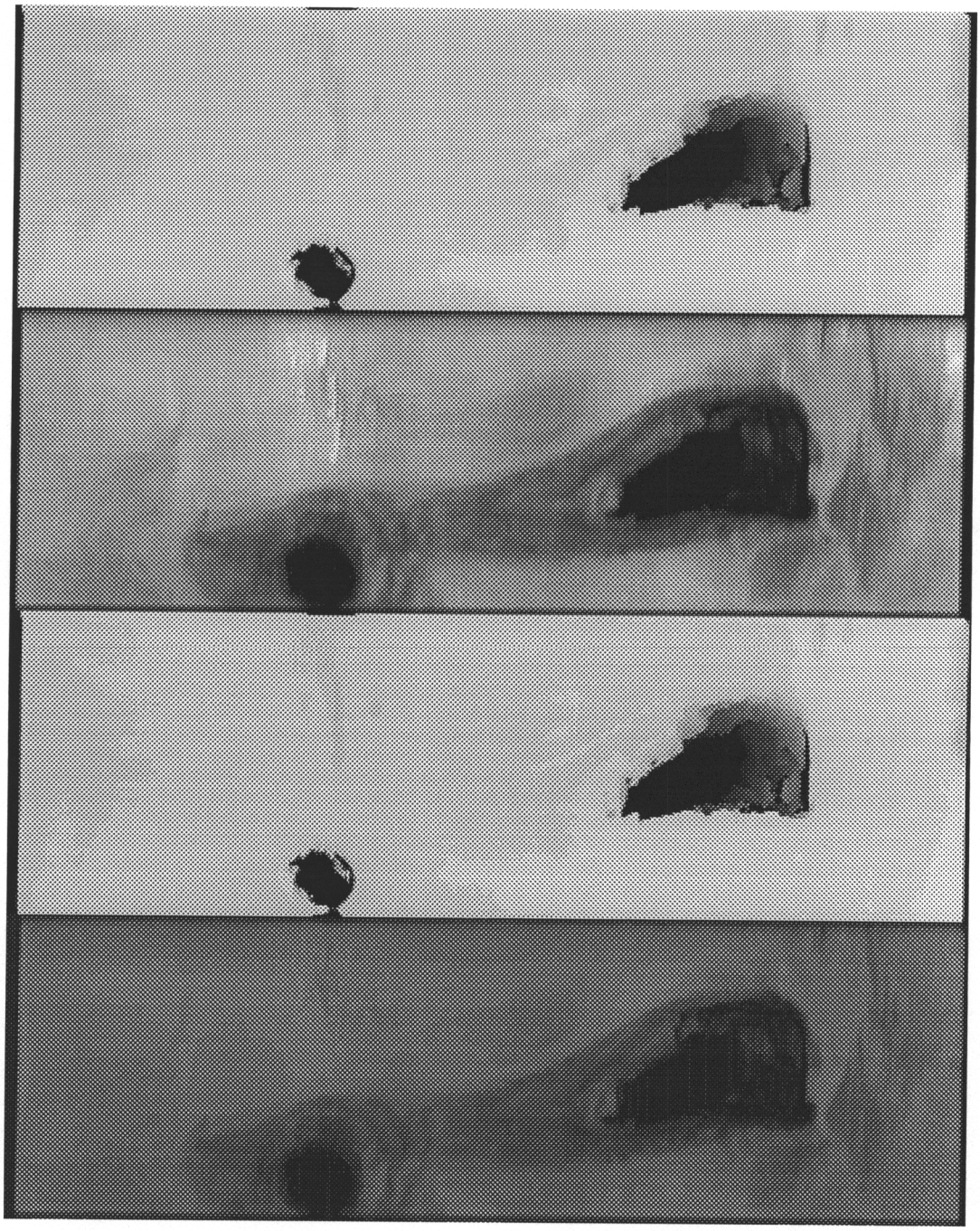


Figure 7.8 Comparisons of registered image pairs. From top to bottom: (A) range in registration with (B) intensity (Total-Reflectance), (C) range in registration with (D) intensity (Center-Pixel Intensity). Board C.

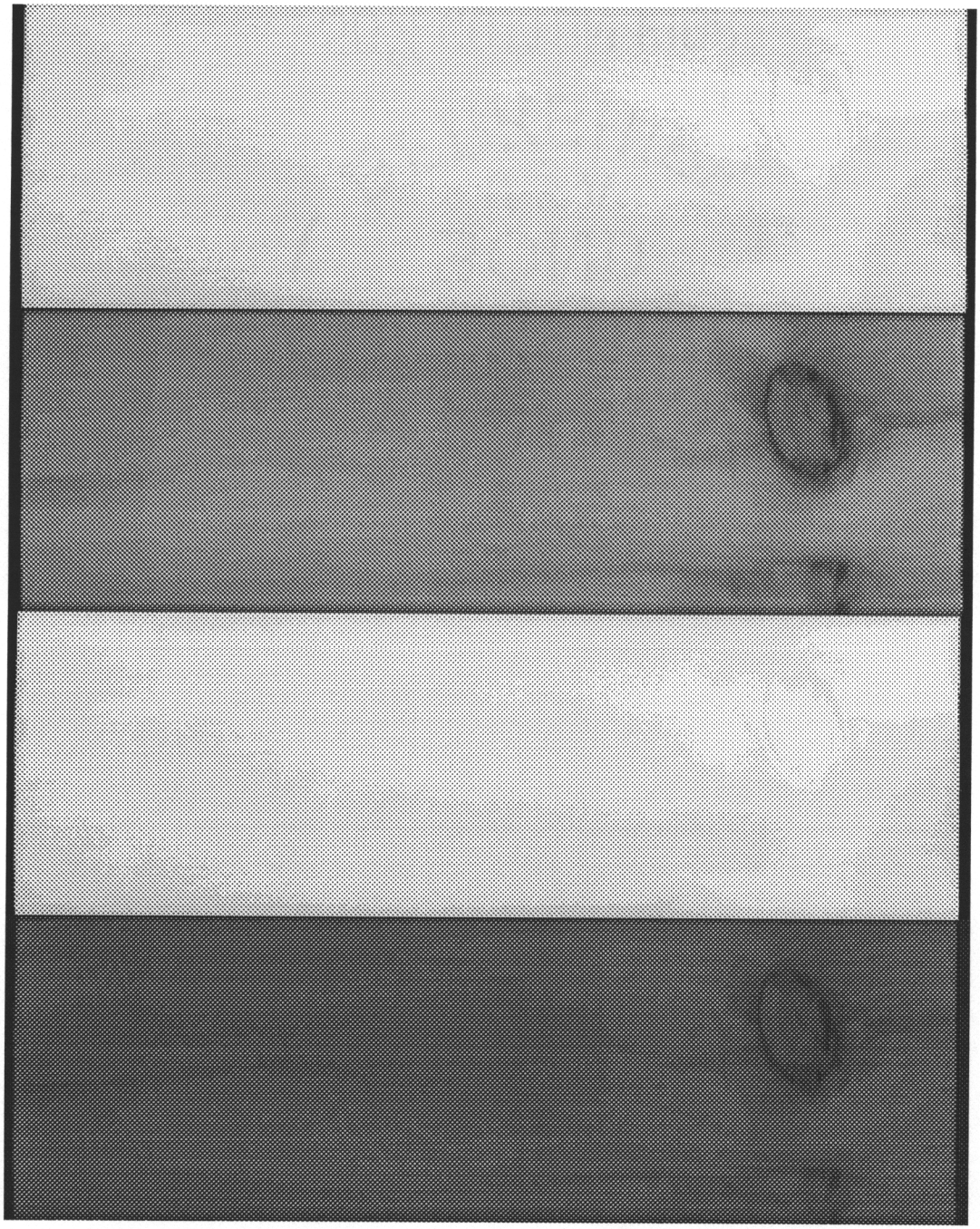


Figure 7.9 Comparisons of registered image pairs. From top to bottom: (A) range in registration with (B) intensity (Total-Reflectance), (C) range in registration with (D) intensity (Center-Pixel Intensity). Board D.

Figure 7.10 Comparisons of registered image pairs. From top to bottom: (A) range in registration with (B) intensity (Total-Reflectance), (C) range in registration with (D) intensity (Center-Pixel Intensity). Board E.

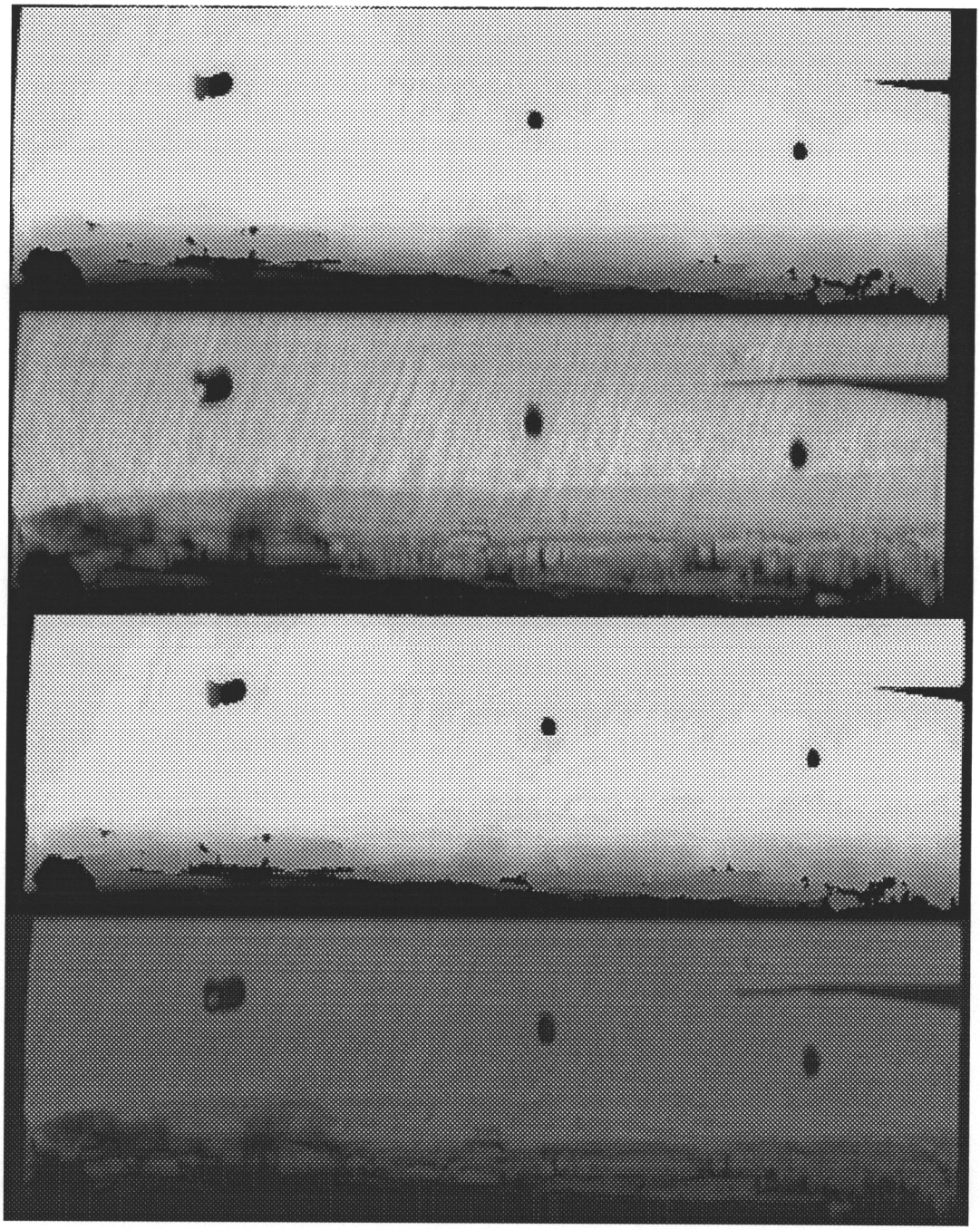


Figure 7.11 Comparisons of registered image pairs. From top to bottom: (A) range in registration with (B) intensity (Total-Reflectance), (C) range in registration with (D) intensity (Center-Pixel Intensity). Board F.

Chapter 8
Future Development

This chapter contains proposals for improving the range sensing prototype system's performance. These suggestions will focus on image sensor technologies rather than on position sensor diode technologies. The reason is that, with a video camera, a user can obtain full knowledge of the viewing scene and therefore can proceed with various optical adjustments as required. With a position sensor diode, all adjustments have to be performed beforehand. A user cannot actually know whether the image has been properly focused or not.

A system equipped with a position sensor diode, however, has the advantage of acquisition speed. Nevertheless, the trend in image sensor technology will push the sensor speed and pixel dimension into a new era. In fact, several advances have been reported (see, e.g. [35]-[36]) that image sensors can support high speed and high resolution operations. For example, [36] reported a 512X512 matrix CCD sensor that can operate at 500 frames per second. When used in a range sensing system, this camera would provide a data generation rate of 256,000 range pixels per second, 500 profile lines per second, and at 9-bit depth resolution. The performance of a range sensing system based on this image sensor technology would be better than a system employing a photo sensor diode. However, the cost of these CCD sensors are presently vary high.

Chapter 6 explains some of the performance constraints encountered in the present prototype system. Among these constraints the optics is the major issue. Hopefully by

proposing solutions of the existing optical problems, the future version of the range sensing system can be made to be more robust and can achieve better performance levels.

8.1 Optical System

While the image sensor determines the range generation speed, the optical system determines the accuracy of range measurement. The following sections describe modifications that can be made to the prototype that will improve range measurement accuracy.

8.1.1 Laser Beam Width Reduction

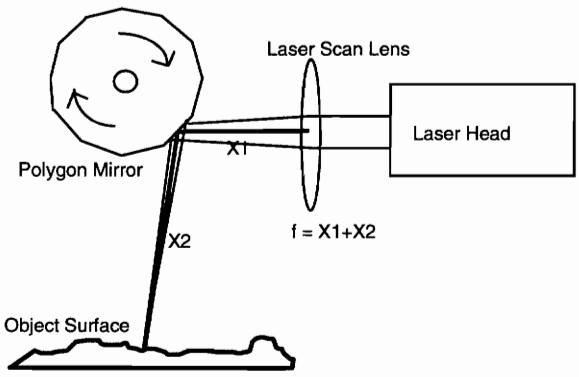


Figure 8.1 Optical structure for beam width reduction.

Section 6.1 discussed the theoretical limit of range measurement accuracy using the range detection scheme of center-pixel position. It was shown that this limit is directly related to the laser's beam diameter. The beam diameter must be small in order to obtain fine range resolution. However, the beam diameter can not be smaller than the optical wavelength of the laser beam. This is the diffraction limit commonly encountered in the optical system design.

It is nevertheless feasible to reduce the beam diameter somewhat by using optical lenses. Figure 8.1 shows a method for reducing the laser beam width simply by placing a laser scan lens in front of the laser head. This scan lens, optimized to the optical wavelength of the laser beam, has a focal length that is equal to the beam's traveling path to the object surface.

8.1.2 Depth Image Magnifier Design

As mentioned in Section 6.3.3, the quality of cross field focusing determines how well the system can detect fine edges on an object surfaces. The imaging lens optics currently implemented in the prototype uses two cylindrical lenses to obtain different image magnification factors. A significant drawback is that these lenses have a constant curvature and as a result there is spherical aberration in image formation process. Figure 7.8. in Chapter 7 shows a situation where fine surface features such as a crack can not be detected from the range image because it is blurred as a result of this spherical aberration. To improve image quality a more elaborate lens design has to be used to compensate for the spherical aberration of both lenses. In doing so, the development cost is greatly increased and may not be a practical way to approach this problem.

Figure 8.2 and 8.3 illustrate a less expensive way of solving this problem. Instead of using two cylindrical lenses to obtain different image magnifications in the two different orthogonal axes, two cylindrical lenses are placed in parallel to create a magnified virtual image (in the direction of depth). Using both a convex and a concave lenses reduces the spherical aberration somewhat because convex lenses have a positive aberration coefficient while concave lenses have a negative one. In this configuration a regular camera lens is used to obtain the desired cross field spatial resolution.

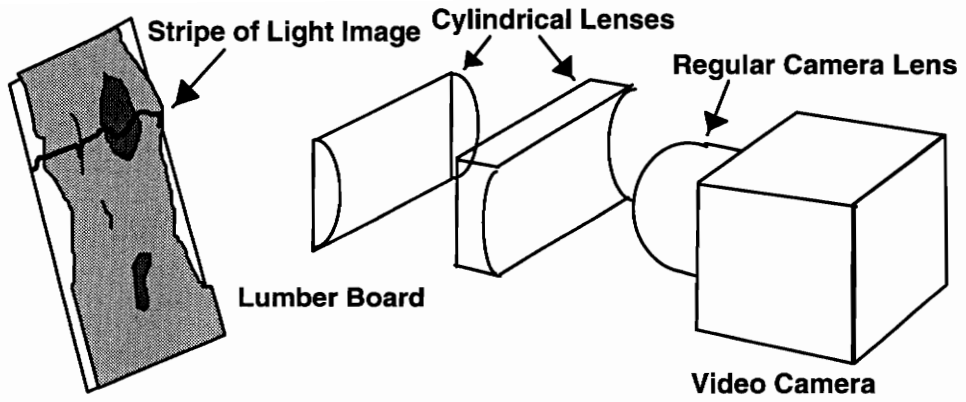


Figure 8.2 Depth magnifier design.

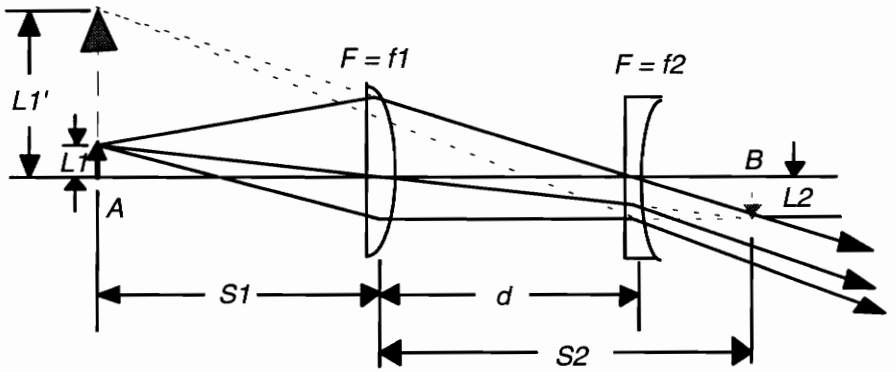


Figure 8.3 Depth Image Magnification.

The principle of this design is that two cylindrical lenses create a magnified image with the virtual image position staying at the object point. When a video camera is equipped with a regular lens and views the scene, what the camera sees is an elongated object at its original position. A module of two cylindrical lenses is used as a depth image magnifier. Figure 8.3 illustrates the principle of this design.

Figure 8.3 shows the geometry of this design. This figure shows the image focusing mechanism in the direction of depth. The other axis is intact in that cylindrical lenses perform light focusing only in a single axis.

Assume the object is placed in front of the first convex cylindrical lens at a distance S_1 (at position A in Figure 8.3). The separation between the convex and concave lenses is d. The focal length of the convex cylindrical lens is f_1 and the focal length of the concave one is f_2. The design principle is to create a virtual image with the image position at the object point. The computation follows a lens combination law.

Assume the image position formed by the first convex lens only is S_2 away (at position B in Figure 8.3) from the lens. The relationship following Gaussian's law is

$$\frac{1}{f_1} = \frac{1}{S_1} + \frac{1}{S_2} \tag{8.1.2.1}$$

If the image formed at B is again imaged by the concave cylindrical lens, the resulting virtual image should be at the position A. If this is true then the design goal is reached. The relationship is

$$\frac{1}{f_2} = \frac{1}{d - S_2} + \frac{1}{-(S_1 + d)} \tag{8.1.2.2}$$

Assume that the object at A has height L_1, the image at B has height L_2, and the resulting virtual image has height L_1'. We can obtain the following equations

$$L_2 = \frac{-S_2}{S_1} \times L_1, \text{ erect image if } L_2 > 0 \tag{8.1.2.3}$$

$$L_1' = \frac{S_1 + d}{d - S_2} \times L_2, \text{ erect image if } L_1' > 0 \tag{8.1.2.4}$$

The magnification factor $\delta = L_1'/L_1$ can be calculated with the help of Equation (8.1.2.1)-(8.1.2.4).

$$\delta = \frac{L_1'}{L_1} = \frac{-S_2}{S_1} \times \frac{S_1 + d}{d - S_2}$$

$$= \frac{f_1}{f_1 - S_1} \times \frac{(S_1 - f_1)(S_1 + d)}{d(S_1 - f_1) - f_1 S_1}$$

$$\text{or} \qquad \delta = \frac{f_1(S_1+d)}{f_1 S_1 + d(f_1 - S_1)} \tag{8.1.2.5}$$

If the magnification factor δ is a specified value and the focal lengths are of interests, Equation (8.1.2.3) can be expressed as

$$f_1 = \frac{dS_1\delta}{S_1\delta + d\delta - S_1 - d} \tag{8.1.2.6}$$

and

$$f_2 = \frac{(S_1+d)(dS_1 - df_1 - f_1 S_1)}{S_1^2} \tag{8.1.2.7}$$

As can be seen from Equation (8.1.2.6) and (8.1.2.7), there are three degrees of freedom left in choosing parameters. If the magnification factor δ is fixed for a particular application, there are still two degrees of freedom left in choosing S_1 and d. S_1 is the distance between the first convex cylindrical lens and the illuminated surface, and d is the separation between two cylindrical lenses. We can find optimal values for them for a specific application. For example, if an application requires that δ=10, S_1= 40cm, d= 4cm, then the required cylindrical lenses would be f_1= 4.04cm convex and f_2= 0.49cm concave.

In fact, this module does not need to have cylindrical lenses where one is concave and one is convex. It is possible to have two pieces of concave cylindrical lenses, depending on the parameters specified in Equation (8.1.2.6) and (8.1.2.7).

After the object image is elongated, a regular camera lens can be installed and proceed with adjusted without worrying about adjusting the cylindrical lenses. The main reason for this design is that a regular camera lens is nicely compensated for spherical aberration. However, its image magnification is the same for both axes. To reach better depth resolution the range sensing system has to obtain elongated images in the depth direction. As mentioned in Chapter 6, the cross field focusing provides accurate detection of fine features such as cracks. A regular camera lens is typically designed to accurately

compensate for spherical aberration and is therefore a better choice for a lens. The depth magnifier design presented above elongates the object height and bring the image back to the object point. This module is not free from spherical aberration. However, as mentioned in Chapter 6, the image blur in the depth direction is not critical. The Center-Pixel algorithm is robust against image blur.

8.1.3 Depth Magnifier Stability Analysis

As explained in Section 8.1.2, the purpose of depth magnifier is to elongate the depth image while keeping the virtual image at the same location as the object point. In this way the regular camera lens has no trouble focusing the elongated image onto the image sensor. However, when the object deviates from the designed point (deviate from S_1 as in Section 8.1.2), the virtual image may not stay at the same position as the real object.

The following analysis studies the stability performance when the object deviates from the designed position.

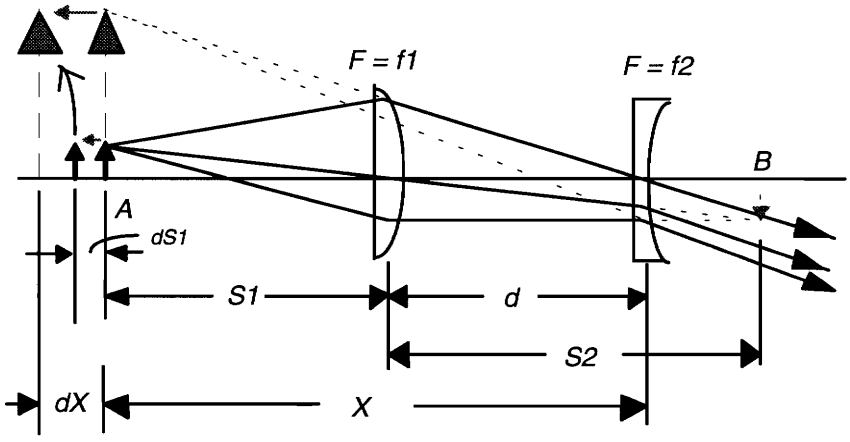


Figure 8.4 Stability analysis of depth magnifier.

As shown in Figure 8.4, the first cylindrical lens focuses the object at *A* to the image at *B*. The position variation caused by the object movement is

$$\frac{dS_2}{dS_1} = \frac{-f_1^2}{(S_1 - f_1)^2} \tag{8.1.3.1}$$

Assume the virtual image focused by the second cylindrical lens with focal lens f_2 is *X* distance away. The expression for *X* is

$$X = \frac{f_2(d - S_2)}{f_2 - d + S_2}, \quad > 0 \text{ if to the left} \tag{8.1.3.2}$$

The variation of the virtual distance *X* caused by the object position deviation is thus

$$\frac{dX}{dS_1} = \frac{-(f_2 - d + S_2) f_2 \frac{dS_2}{dS_1} - f_2(d - S_2)\frac{dS_2}{dS_1}}{(f_2 - d + S_2)^2} \tag{8.1.3.3}$$

Substitute dS_2/dS_1 with Equation (8.1.3.1), we get

$$\frac{dX}{dS_1} = \frac{S_2}{S_1} \times \frac{(S_1 + d)^2}{(d - S_2)^2} = \delta^2 \tag{8.1.3.4}$$

Equation (8.1.3.4) shows that the virtual image deviates from its position δ^2 times the distance of the object point deviation. The distance between the virtual image and the object is therefore (δ^2-1) the distance of the object point deviation. What this tells us is that the more the depth magnifier elongates the image, the more unstable the virtual image is.

If the depth magnifier is to be used in magnifying depth images, the magnifying factor and the depth sensing range should be carefully considered. As shown in Equation (8.1.3.4), the virtual image has to stay within the camera lens's depth-of-field or the image formed on the image sensor would be seriously blurred. The depth magnifier is not applicable for large depth sensing range applications. However, applications such as

lumber inspection, the thickness variation is well within 1 inch. This is small compared to the working distance of the camera. Hence the use of a depth magnifier is appropriate.

8.2 Linear CCD Version

The range sensing system developed in this project has two moving components. A linear translation stage carries inspected objects in a linear manner at constant speed for profile samplings down board, and a spinning mirror that creates a light plane for object illumination. The laser camera module has no moving component because the cross field imaging is accomplished by using a matrix array CCD sensor and a cross field focusing lens (a cylindrical lens).

If a linear array CCD sensor is to be employed as the image sensor, the optical structure has to be changed. The objective of using a linear array is to enhance the measurement resolution. Due to its capability of high packaging density, up to 4,096 photocells can be fabricated in a single chip. The resulting range resolution can therefore be 12 bits. The drawback of using a linear array sensor, however, is the slow range generation rate. A linear array sensor with 4,000 photocells using a clock rate of 20 MHz can have a line rate of 5,000 lines/second. The equivalent range generation rate is then 5,000 range pixels per second. This acquisition speed is very slow.

Marc Rioux proposed a synchronized scanning system that can employ either a linear CCD sensor or a position photodiode[24]. Section 2.1.5 describes this system and its optical structure. The disadvantage of this scheme is that the imaging area is limited by a diamond shaped scanning mirror (one facet). The light source has to be powerful for object illumination because of the small observing area provided by the diamond mirror. Figure 8.5 shows an alternate way of creating the synchronized scanning scheme. The use

of a large size scanning mirror ensures the illuminated image to be focused onto the linear CCD sensor once this system is calibrated. The motion of this mirror will not affect the image sensor observation if it is accurately calibrated with its motion being parallel to the observation plane (maintained by the mirror). It also makes the image lens capable of collecting more light for CCD detection.

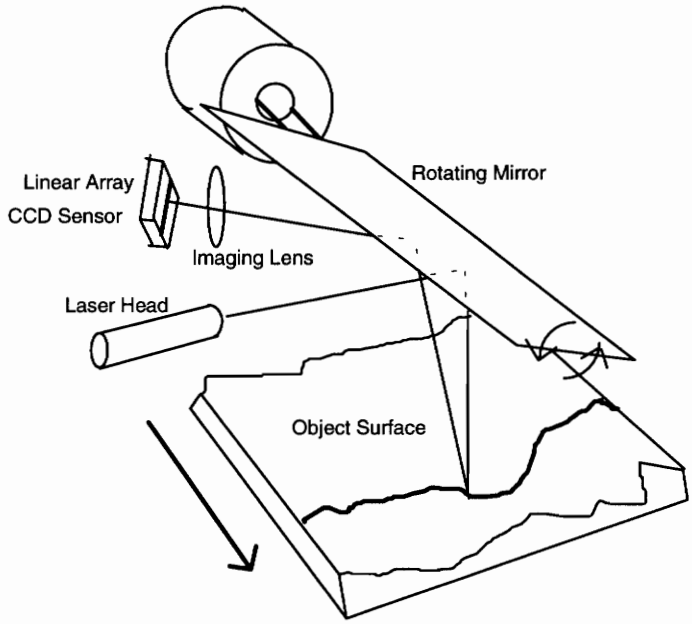


Figure 8.5 Range sensing system using a linear CCD sensor.

8.3 Sub-pixel Range Detection

As mentioned in Section 4.2.3, the Range Processing Unit detects center pixel positions of bright video segments. The output of this unit indicates the representative positions of the corresponding bright spot images in a video line. As a result, a video camera with 2^N pixels in a video line can provide 2^N possible range positions. This in turn means that the range resolution is N bits. Nevertheless, the center pixel scheme can be

modified so that a video camera with the same number of pixels can provide $(N+1)$ bits of range resolution. Figure 8.6 presents the processing idea.

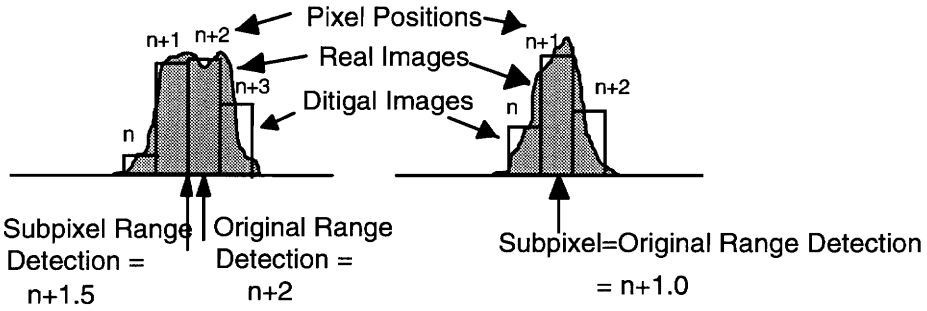


Figure 8.6 Effects of sub-pixel range detection.

As shown in the figure, the Sub-pixel range detection scheme can double the number of usable range positions and thus provide better representations of center-pixel positions. To achieve this capability requires very little modification to the current of range sensing hardware. Simply by rewiring several signals of the Range Processing Unit it can perform sub-pixel range detection.

If this scheme is used, a matrix CCD sensor with 2^N pixels can have 2^{N+1} range positions. The range data is therefore (N+1) bits wide.

8.4 Multi-head Range Sensing System

The range sensing system developed in this project is a single camera system. Due to the use of a single camera, the measurement performance is limited by the capability of the installed camera. The CCD technology provides image sensors with spatial regularity and high packaging density. However, it is the characteristics of silicon material that restrains these sensors from high speed operations. It is caused by data transfers between CCD cells. The range generation rate is directly related to the data transfer speed of CCD

cells and therefore it is limited. A way of improving the speed constraint is to use more cameras in parallel. A range sensing system consisting of several camera heads can generate more range pixels due to the parallel range sensing. Figure 8.7 illustrates the optical configuration of this system.

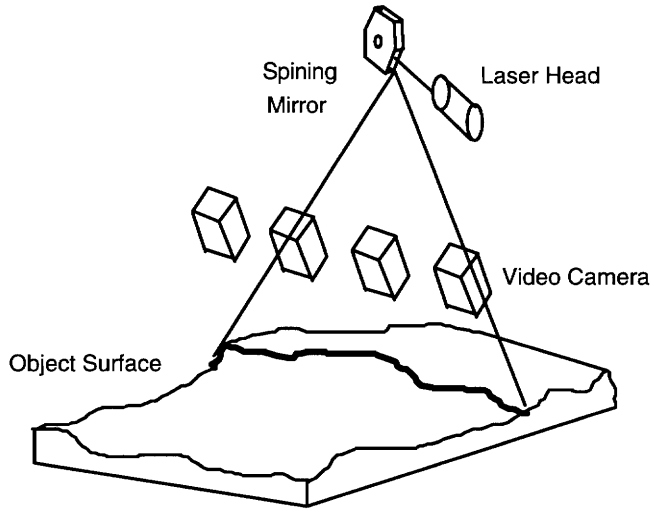


Figure 8.7 Multi-head range sensing system.

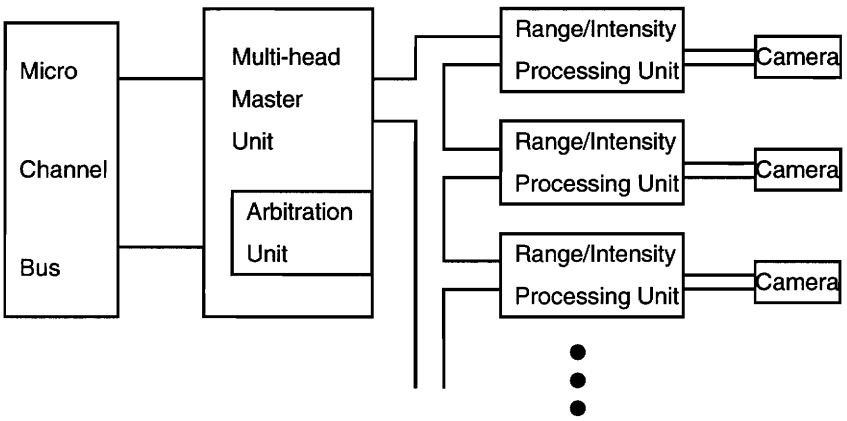


Figure 8.8 Block diagram of the multi-head range sensing system.

The hardware system for managing these cameras can be redesigned by adding a master unit for governing a local bus formed by several modules of range/intensity processing units. Figure 8.8 shows the block diagram of this system. The master unit has

an arbitration unit that can identify bus requests from the individual modules, and works in concert with the host processing system.

8.5 Light Flashing and Synchronization Mechanism

For low frame rate cameras, capturing images of fast moving objects may result in a significant blur that render these images useless. A general remedy for this situation is to generate high intensity light pulses to illuminate the scene. The camera therefore receives a short but intense pulse for every frame. This is particular useful for cameras like RS170 (30 frames per second). If this is done, the range sensing using RS170 camera can scan fasting moving objects without the blur associated with a 1/30 second integration time. Figure 8.9 illustrates the proposed optical setup.

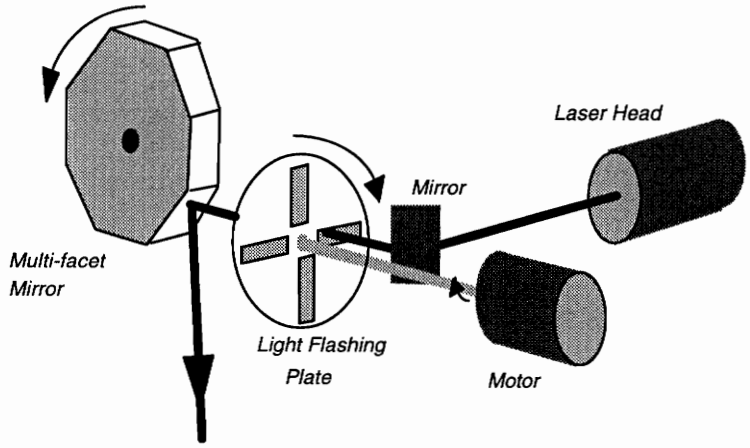
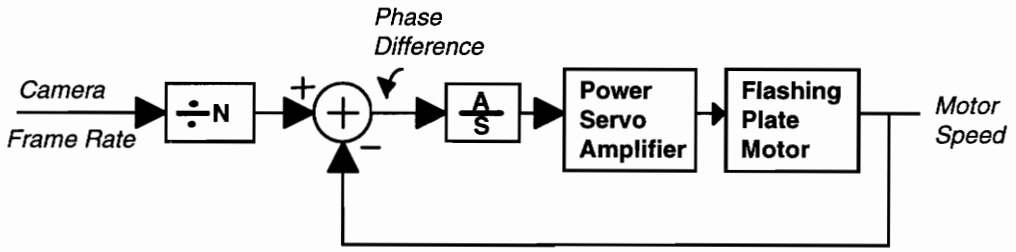


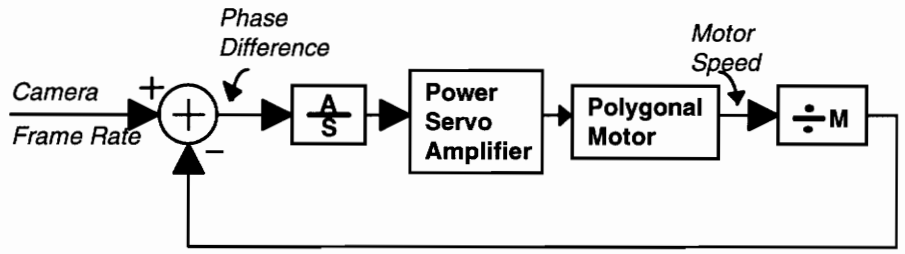
Figure 8.9 Light Flashing Mechanism.

The optical module shown in Figure 8.9 replaces the laser scanning mechanism described in Section 3.2.4. A spinning motor carries a circular plate that allows the laser beam to pass through openings on the plate at a repetitive rate. The repetition rate is therefore determined by the speed of the motor, and the flashing period of the laser beam

is determined by the size of the opening. If the repetition rate is synchronized to the camera frame rate, the camera can then receive one flash light per video frame. This requires a careful design of the motor controller. During the period of light flashing, the polygonal mirror is assumed to be rotating at a very high speed that can provide several scan lines even in the short flashing period. This may not be true when the flashing period is short. Nevertheless, if the polygonal mirror can provide a fixed integer number of scan lines in a flashing period, the created plane-of-light can still be deemed uniform even though there are only few scan lines. This requires a synchronization of the camera frame rate and the polygon spinning rate.



(A) Synchronization Scheme for Light Flashing Motor Module



(B) Synchronization Scheme for Polygonal Laser Scanning Module

Figure 8.10 Controlling schemes for motor drives.

Figure 8.10 presents the control schemes for both the flashing plate motor and the polygon motor. Basically the principle is to perform the Phase-Lock-Loop (PLL) feedback control. In (A) we assume there are N symmetrical openings on the flashing plate. The camera frame rate is divided by N to feed in the controlling module. After the

system reaches steady state, the motor is synchronized to the divided-by-N frame rate. As a result, only one opening can pass through for every video frame exposure.

The controlling scheme presented in Figure 8.10.(B) has the same principle as in (A). The only difference is in the position where the frequency division happens. The polygon spinning motor operates at a much higher speed than the camera frame rate. To synchronize the motor with the camera, a divided-by-M block is added. When synchronized, the camera can receive M scan lines for every frame exposure. The resulting plane-of-light can be deemed almost uniform. When the flashing mechanism shown in Figure 8.9 is added to the scanning module, the number of scan lines for a video frame is still a fixed number. The plane of light is therefore uniform in this situation.

8.6 Summary

This chapter presented several possible ways for improving the range sensing performance. In general the optical system is basically the main problem in the system design since it is not as flexible as the other hardware components. For most cases it is also the most costly part of the system. Image focusing quality dominates measurement accuracy. Although the cylindrical lens solution explained in Chapter 3 is the best possible optical arrangement for flexible performance, the induced spherical aberration is its main drawback. To improve this situation an elaborate composite lens system would have to be designed. This is a costly solution.

A depth image magnifier design has been proposed as the second choice. However, this magnifier does not support a wide dynamic range of depth sensing, as has been described in this chapter. For shallow depth sensing applications this solution is inexpensive and can maintain the high output capability of matrix CCD sensors.

The beam width reduction scheme is strongly recommended for improving the measurement accuracy. Structurally it is simple and flexible. It is also inexpensive in that only a regular concave lens is required.

The linear CCD version is an alternate arrangement in performing range detection. However, as explained in Chapter 2, it requires precision modules for maintaining the imaging plane. This solution probably would be more expensive as compared to the matrix CCD solution.

A hardware scheme for sub-pixel range detection is proposed. This scheme is simple and requires minimum modification of the existing prototype, while doubling the range resolution.

Using a combination of these approaches, the prototype range sensing system could be modified to get increased flexibility, better accuracy, and better capability in the industrial environment. It is the sincere hope that through the study of the prototype system the range sensing system combined with intensity generation will prove a efficient vehicle for attacking a number of industrial inspection problems.

Chapter 9
Conclusion

The goal of the project contained in this thesis was to create a range sensor that is robust and flexible so that a number of applications within the forest products manufacturing environment can be addressed. A prototype system has been built and tested. The development of this prototype focused on automatically detecting both surface and geometric features in wood by integrating laser ranging information with gray-level intensity information. Although the specific design parameters of this prototype focused on aspects within the forest products manufacturing environment, the overall system design can be adjusted for a wide variety of other industrial applications as well. This prototype features:

(a) The capability of producing spatially registered image pairs of range and intensity.
(b) The high data generation rate due to dedicated hardware signal processing.
(c) The applicability to a wide variety of applications due to the flexibility of the optical structure.
(d) The large depth-of-field range sensing capability, the real-time data processing.
(e) The hardware diagnostic capability.

A triangulation based range sensing technique was selected to be the structure of the range sensor. This technique is considered best suited for industrial applications where the system throughput rate is a great concern and where the flexibility of optics is critical

for a wide variety of specifications. A specialized imaging module was designed to achieve the optical flexibility. In this study it is also shown that, by employing the triangulation method and the specialized imaging optics, the range sensor can perform large depth-of-field range sensing, a capability crucial to robotics navigation. It is the optical flexibly that allows the range sensor to address a number of industrial applications.

A stand-alone hardware system was designed and implemented to perform pipelined image processing for extracting range and intensity information. The hardware contains (a) a frame grabber designed for acquiring camera's view for the purpose of optical adjustments and calibrations, (b) three pipelined signal processing units for generating registered range and intensity images, (c) a real-time look-up-table unit for data mapping from imaging pixels to range data, (d) a hardware diagnostic facility for pin-pointing hardware failures, and (e) a MicroChannel interface control unit. Range pixels are generated by a hardware unit employing the Center-Pixel algorithm. This algorithm ensures the data accuracy. Intensity pixels are generated by either one of two hardware units employing the Center-Pixel or Total-Reflectance intensity algorithm. A set of system registers are provided to facilitate hardware management performed by the host computer.

A memory management scheme was implemented to demonstrate the real-time data processing capability of the range/intensity sensing system. A program utilizing this scheme was demonstrated. The research also provides a set of utility subroutines for application programming.

This thesis contains a theory exploring the accuracy performance of the Center-Pixel algorithm. The study shows that the range measurement accuracy is directly related to the laser beam diameter and the camera's imaging resolution. The relationship of range measurement accuracy and the laser beam diameter complies with the well-known sampling theorem of digital signal processing. It is concluded that the limit of range

measurement accuracy is twice the quantity of the laser beam diameter. Several sources of measurement errors are also discussed. Proposals for enhancing system performance are included.

Several experimental results of the prototype system are included in this thesis. These experiments are performed by using a high speed video camera, MC9128, operating at a video frame rate of 380 frames/sec. These experiments also show that the range/intensity sensing prototype is capable of performing real-time data processing while maintaining high measurement resolution. Through proposing methods toward future development, the operating speed and measurement accuracy can be manipulated such that the range sensor can be a general purpose system for looking at many different industrial applications.

References

1. Regalado, C., D. E. Kline, and P. A. Araman, 1992, "Value of defect information in automated hardwood edger and trimmer systems," *Forest Products Journal*, 42(3):29-34.

2. Henry A. Huber, Charles W. McMillin, John P. Mckinney, "Lumber defect detection abilities of furniture rough mill employees," *Forests Journal*, Vol. 35, No. 11-12 (1985).

3. Szymani, R. and McDonald, K. A., "Defect detection in lumber: state of the art." *Forest Products Journal*, Vol 31, no. 11, pp. 34-44, 1981.

4. Conners, R. W., C. T. Ng, T. H. Drayer, J. G. Tront, and D. E. Kline. 1990. "Computer vision hardware system for automating rough mills of furniture plants." *Proceedings of the SPIE, Applications of Artificial Intelligence*, VII, Orlando, Florida, pp. 777-787.

5. Grove, L., "State-of-the-art edger and trimmer optimizers." *Improving productivity through microelectronics*, Special Publication, Forintek Canada Corp. (No. SP 12).

6. Paul Besl & Ramesh Jain, "Range Image Understanding," *Proc. IJCAI*, pp. 430-449, 1985.

7. Kline, D. E, Hou, Y. J., Conners, R. W., Schmoldt, D. L., Araman, P. A., "Lumber scanning system for surface defect detection," ASAE Meeting Presentation, Paper No. 923582.

8. Carmen Regalado, D. Earl Kline, Philip A. Araman, "Optimum edging and trimming of hardwood lumber," *Forest Products Journal*, 42(2):8-14, 1992.

9. Griffin, G., "Edger optimization vital to hardwood mill upgrade," *Forest Industries*, March 1989.

10. Conners, R. W., T. H. Cho, C. T. Ng, T. H. Drayer, P. A. Araman, and R. L. Brisbin. "A machine vision system for automatically grading hardwood lumber." *Industrial Metrology*, 2:317-342, 1992.

11. King, R. J., "Microwave electromagnetic nondestructive testing of wood," *Proceedings of the 4th Symposium on Nondestructive Testing of Wood,* Vancouver WA, pp 121-134, 1978.

12. Portala, J. F. and Ciccotelli, J., "NDT techniques for evaluating wood characteristics," *Proceedings of the 7th Symposium on Nondestructive Testing of Wood*, Washington State University, Pullman WA, pp. 97-124, 1990.

13. McDonald, K. A. and Bendtsen, B. A., "Measuring localized slope of grain by electrical capacitance," *Forest Products Journal*, vol. 36, no. 10, pp. 75-78, 1986.

14. Steel, P. H., Neal, S. C., McDonald, K. A., and Cramer, S. M., "The slope-of-grain indicator for defect detection in unplaned hardwood lumber," *Forest Products Journal*, vol. 41(1): 15-20, 1991.

15. Kenway, D., "Computer-aided lumber grading," *Proceeding of the 7th Symposium on Nondestructive Testing of Wood*, Washington State University, Pullman WA, pp. 189-206, 1990.

16. McMillin, C. Q., Conners, R. W., and Huber, H. A., "ALPS -- A potential new automated lumber processing system," *Forest Products Journal*, vol. 34, no. 1, pp. 13-20, 1984.

17. Araman, P. A., D. L. Schmoldt, T. H. Cho, D. Zhu, R. W. Conners, and D. E. Kline, "Machine vision systems for processing hardwood lumber and logs," *AI Applications*, 6(2):13-26, 1992.

18. C. Marc Bastuscheck, 1989. "Techniques for real-time generation of range images," *Proc. IJCAI*, pp. 262-268, 1989.

19. Schwartz, J. "Structured light sensors for 3D robot vision," Robotics Research Report No. 8, Courant Institute of Mathematical Sciences, New York University, 1983.

20. O. Kafri and I. Glatt, *The Physics of Moire Metrology*, Wiley Press, New York (1990).

21. K. Engelhardt and G. Hausler, "Acquisition of 3-D data by focus sensing," *Appl. Opt.* 27, 4684-4698 (1988).

22. Kai Engelhardt, "Acquisition of 3-D data by focus sensing utilizing the moire effect of CCD cameras," *Appl. Opt.*, Vol. 30, No. 11, pp. 1401-1407.

23. M. Koskinen, J. Kostamovaara, R. Myllyla, "Comparison of the continuous wave and pulsed time-of-flight laser rangefinding techniques," *SPIE Conference on Optics, Illumination and Image Sensing for Machine Vision VI*, Vol. 1614, pp. 296-305, 1991.

24. Marc Rioux, "Laser range finder based on synchronized scanners," *Appl. Opt.*, Vol. 23, Nov. 21, pp. 3837-3844, 1984.

25. Francois Blais, Marc Rioux, Jacques Domey & J.-Angelo Beraldin, "A very compact real time 3-D range sensor for mobile robot applications," *SPIE Vol. 1007 Mobile Robots III*, pp. 330-338, 1988.

26. Donald J. Svetkoff, Donald B. Kilgus, "Influence of object structure on the accuracy of 3D systems for metrology," *SPIE Conference on Optics, Illumination, and Image Sensing for Machine Vision VI (1911)*, Vol. 1614, pp. 218-230.

27. Gonzalez, R. C., Wintz, P, *Digital Image Processing*, Addison-Wesley, Reading, MA, 1987.

28. *High Speed Design Seminar*, Analog Devices, 1990.

29. *IBM RISC System/600 POWERstation and POWERserver Hardware Technical Reference-Micro Channel Architecture*, IBM, 1990.

30. *IBM Personal System/2 Seminars and Proceedings*, IBM, 1989.

31. Drayer, T. H. *A High Performance Microchannel Interface for Image Processing Applications*. Masters Thesis, Virginia Polytechnic Institute and State University, 1991.

32. *A Plain Man's View of the IBM Micro Channel*, Document Number GG24-3584-00, August 1990.

33. *Hybrid RS-170 Video Digitizer*, Analog Devices, AD9502 Data sheet.

34. Oppenheim, A. V., *Discrete-Time Signal Processing*, Englewood Cliffs, NJ, Prentice-Hall, 1989.

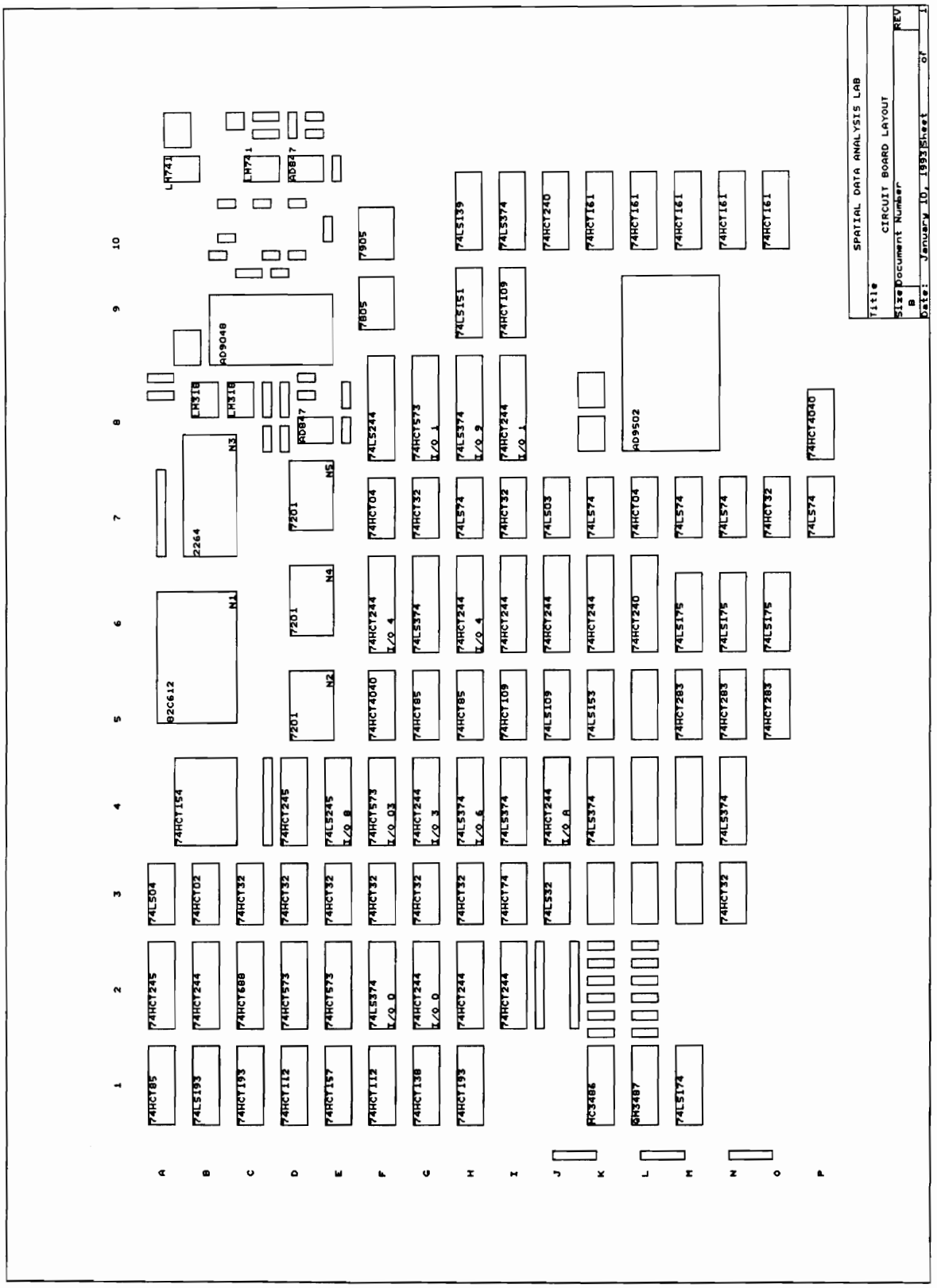
35. Rudolf Germer and W. Meyer-Ilse, "A high-speed CCD-videocamera," *SPIE 19th International Congress on High-Speed Photography and Photonics (1990)*, Vol. 1358, pp. 346-350.

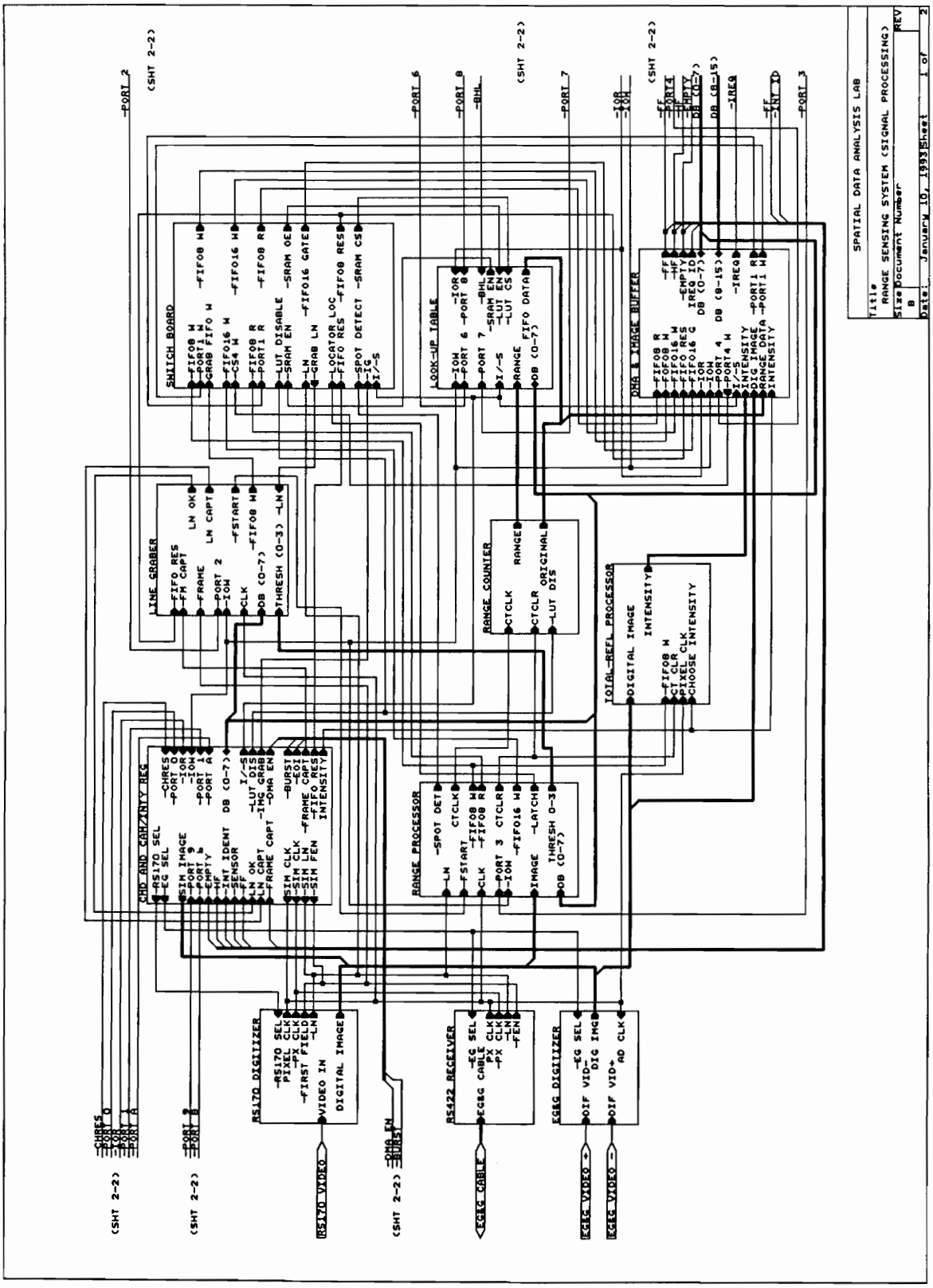
36. K. Ball, D. J. Burt, G. W. Smith, "High speed readout CCD's,", *SPIE 19th International Congress on High-Speed Photography and Photonics (1990)*, Vol. 1358, pp. 409-420.

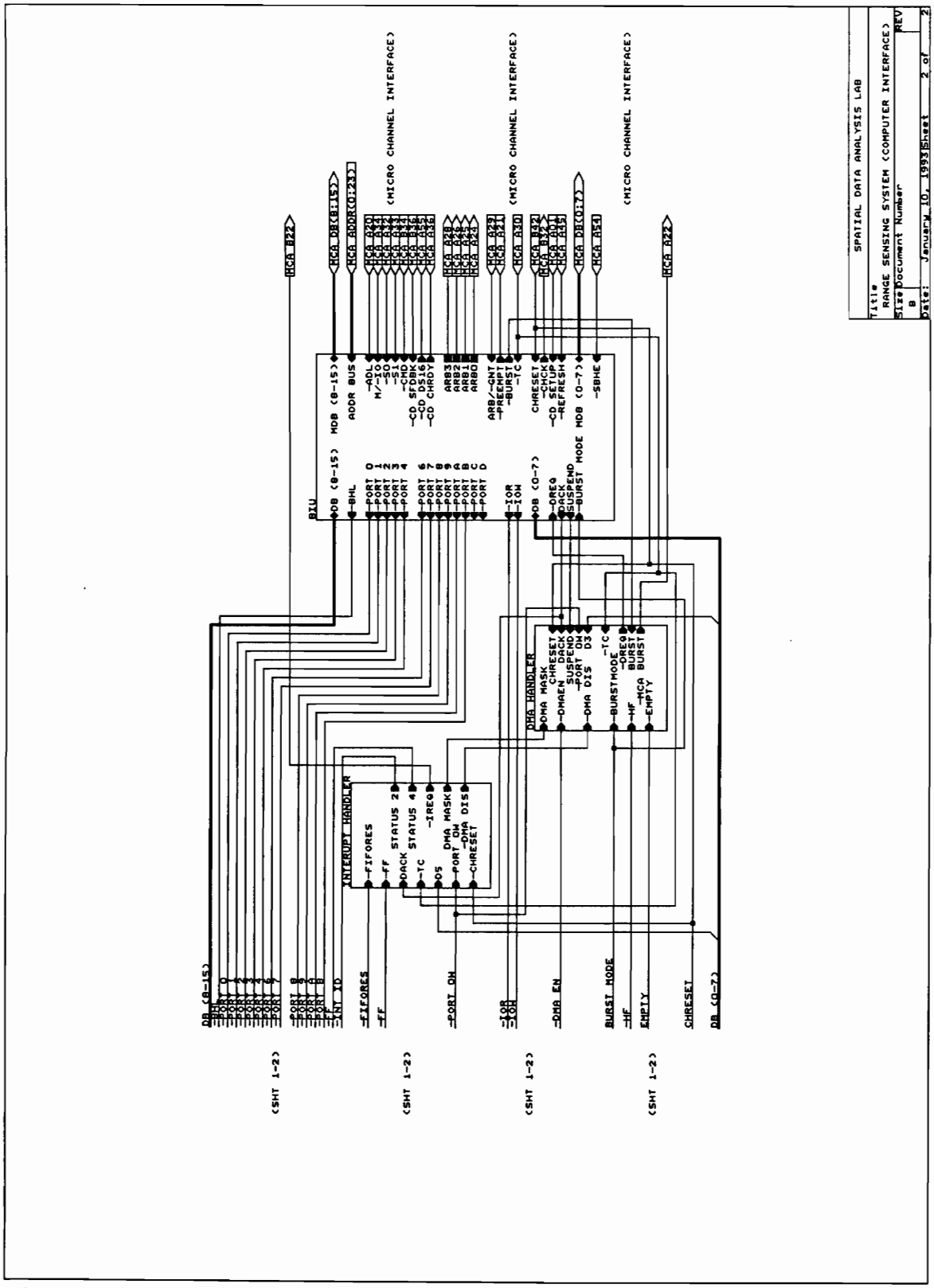
Appendix A

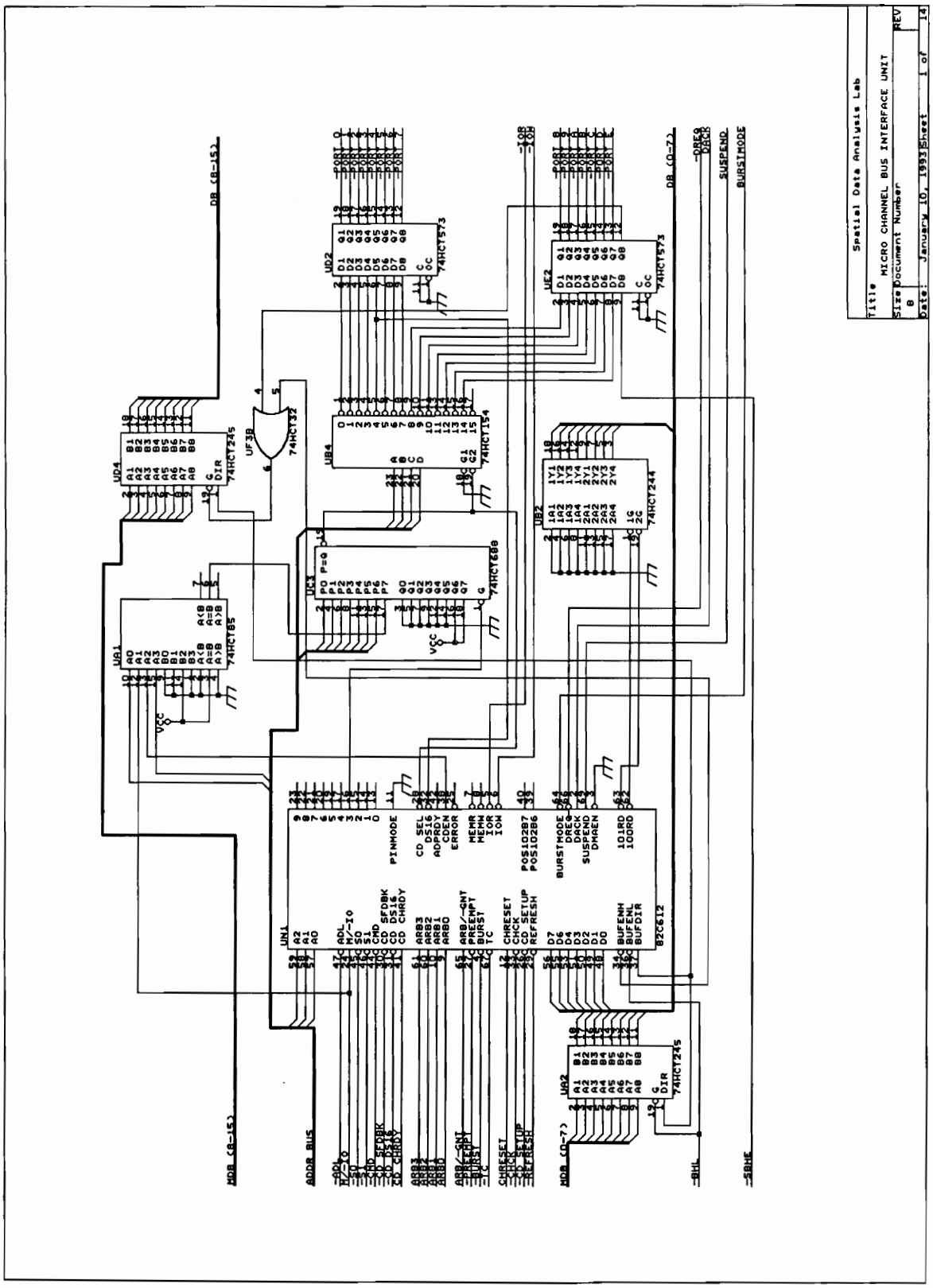
Schematic Diagram

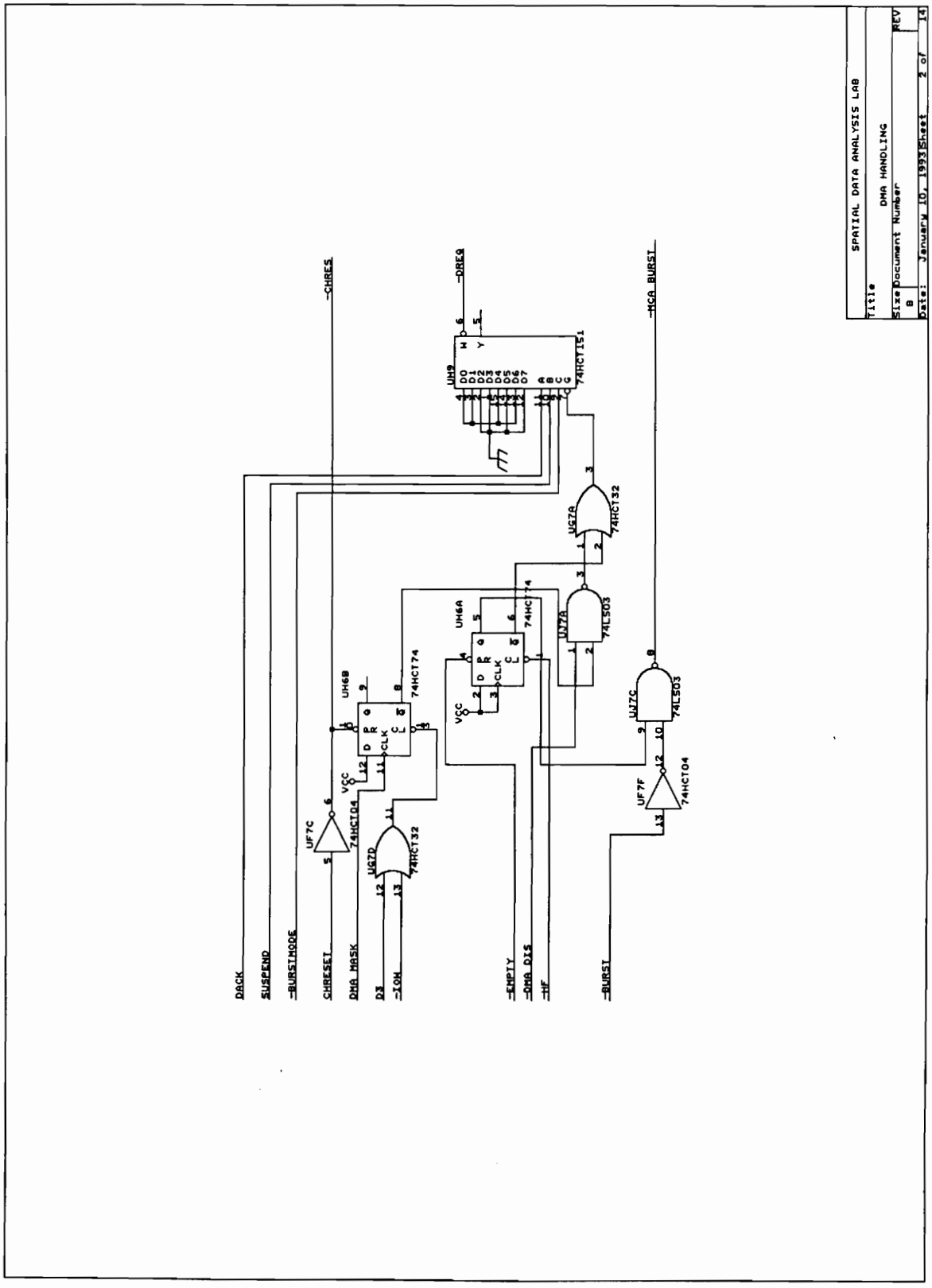
This appendix contains the schematic diagrams of the range sensing hardware. Seventeen Sheets are included to provide the complete diagram. The first two sheets contain modular level design and the next fourteen sheets contain detailed designs of the separate modules. The last sheet provides the physical chip layout of the interface board.

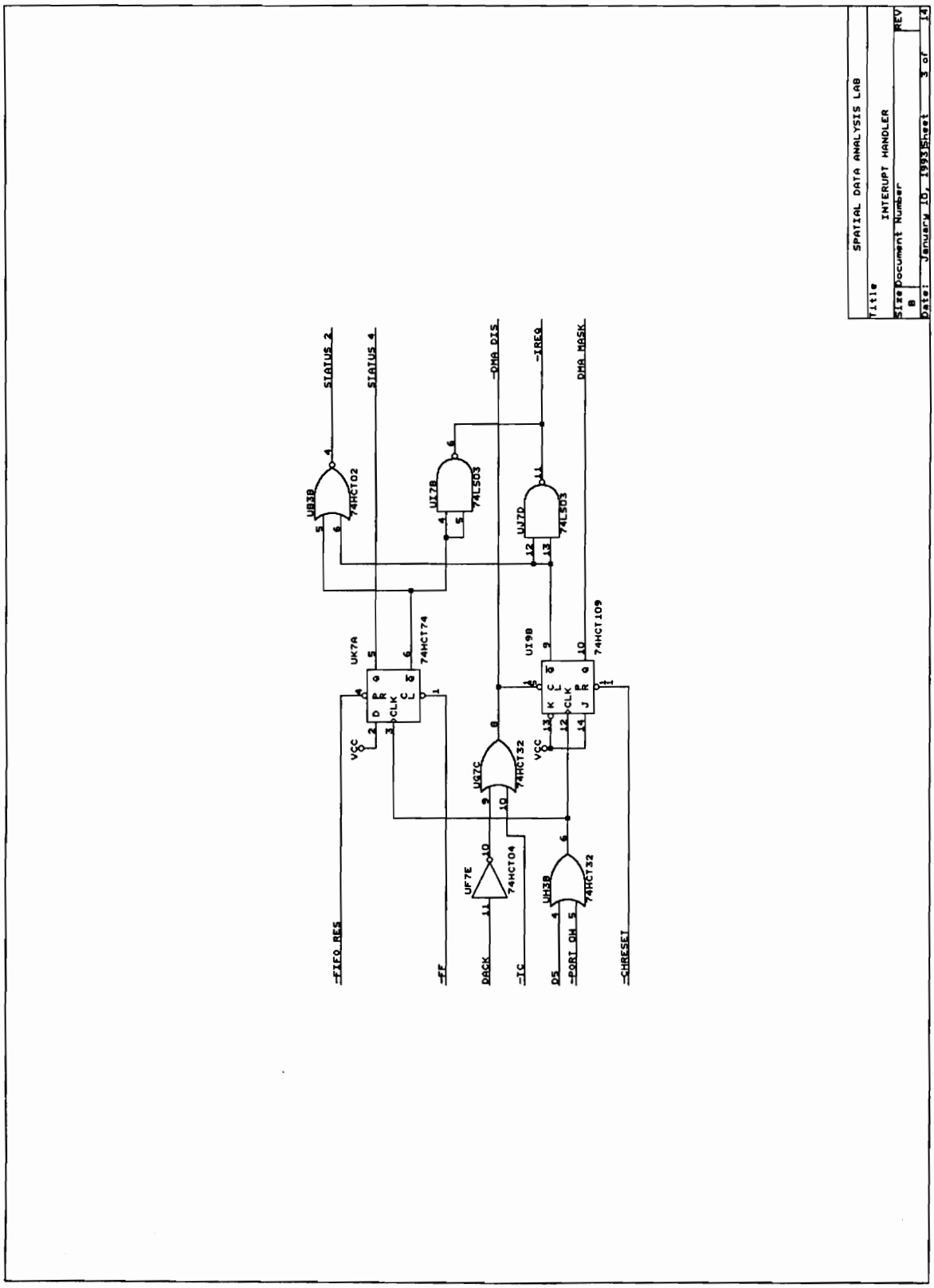

SPATIAL DATA ANALYSIS LAB
Title CIRCUIT BOARD LAYOUT
Size B Document Number REV
Date: January 10, 1993 Sheet of

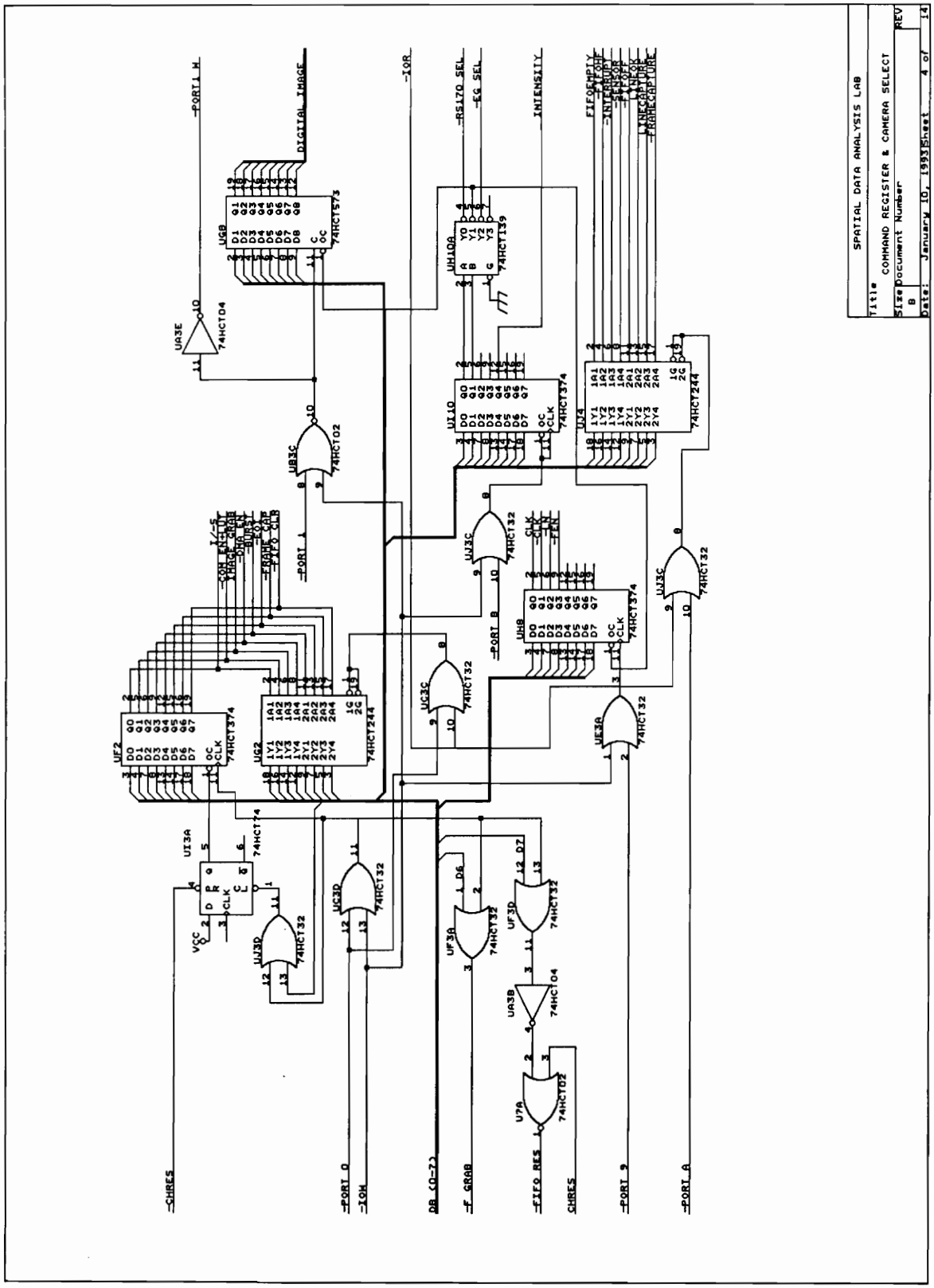

SPATIAL DATA ANALYSIS LAB
Title
RANGE SENSING SYSTEM (SIGNAL PROCESSING)
Size Document Number REV
B
Date: January 10, 1993 Sheet 1 of 2
SWITCH BOARD
LINE GRABBER
LOOK-UP TABLE
DMA & IMAGE BUFFER
CMD AND CONTNTY REG
RANGE PROCESSOR
RANGE COUNTER
TOTAL-REFL PROCESSOR
RS170 DIGITIZER
RS422 RECEIVER
EG&G DIGITIZER
RS170 VIDEO
EG&G CABLE
EG&G VIDEO +
EG&G VIDEO -
(SHT 2-2)
-PORT 2
-PORT 6
-PORT 8
-PORT 7
-PORT 3
-BHE
-IOR
-IOW
-IRQ9
-INT 10
DB (0-7)
DB (8-15)

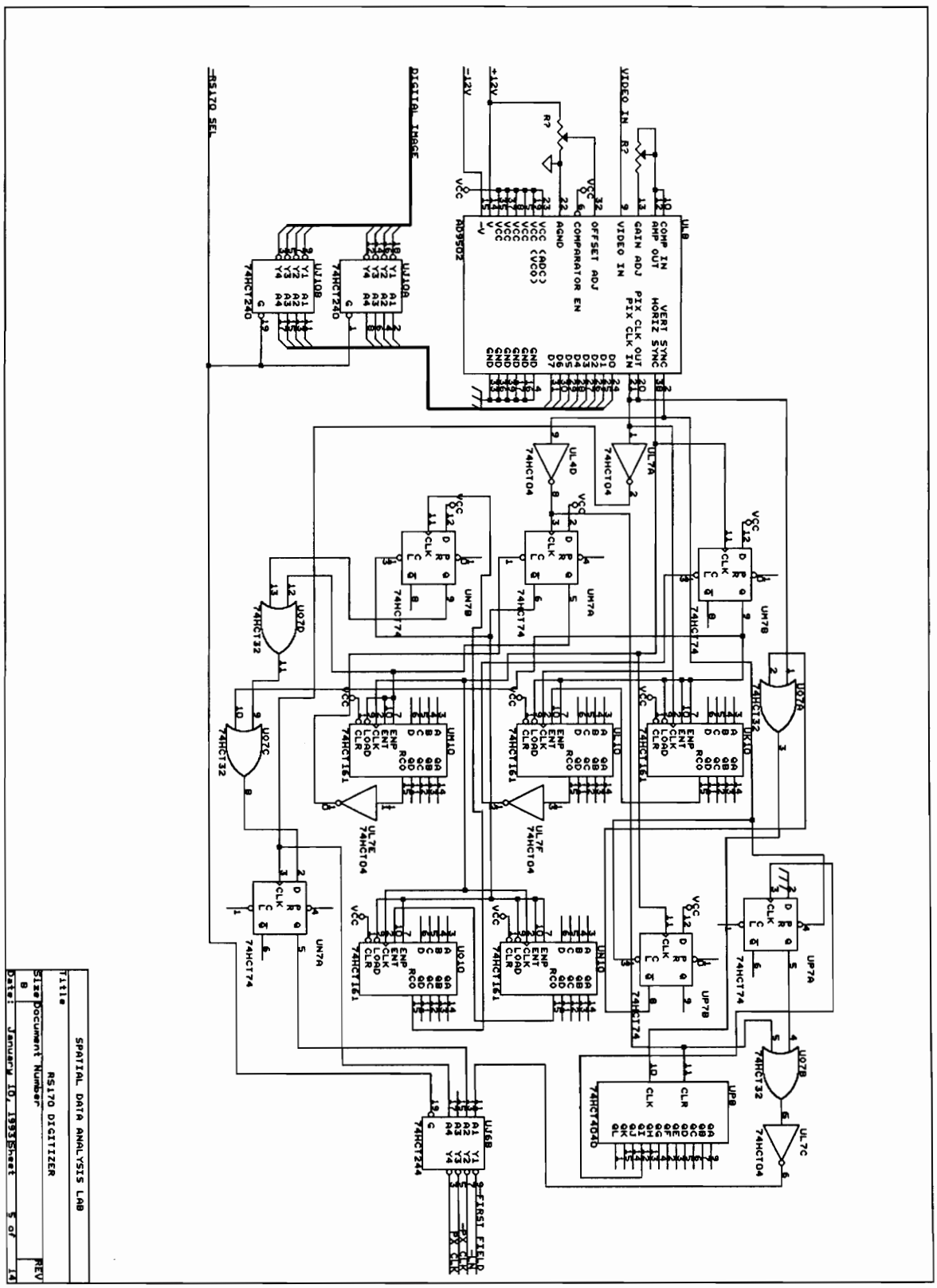

SPATIAL DATA ANALYSIS LAB
Title
RANGE SENSING SYSTEM (COMPUTER INTERFACE)
Size
B
Document Number
REV
Date: January 10, 1993
Sheet 2 of 2
(MICRO CHANNEL INTERFACE)
BIU
INTERRUPT HANDLER
DMA HANDLER
DB (0-15)
DB (0-7)
ADDR BUS
MDB (0-15)
MDB (0-7)
(SHT 1-2)
-FIFORES
-FF
-PORT DH
-DMA EN
BURST MODE
-HF
EMPTY
CHRESET

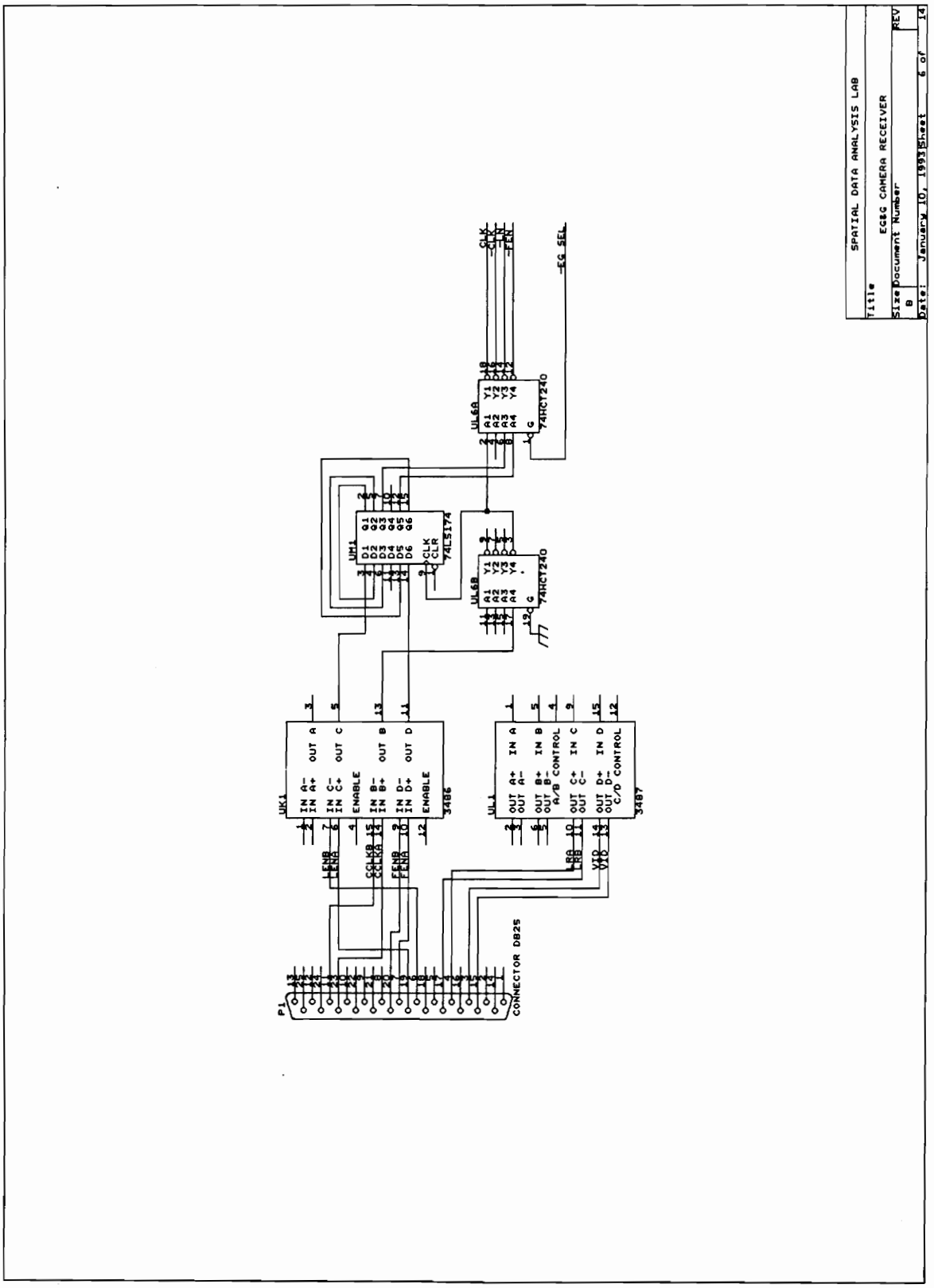

Spatial Data Analysis Lab
Title MICRO CHANNEL BUS INTERFACE UNIT
Size B Document Number REV
Date: January 10, 1993 Sheet 1 of 14
DB (8-15)
DB (0-7)
ADDR BUS
MDB (8-15)
MDB (0-7)
74HCT245
74HCT32
74HCT154
74HCT573
74HCT688
74HCT85
74HCT244
82C612
SUSPEND
BURSTMODE
-SBHE
-BHE

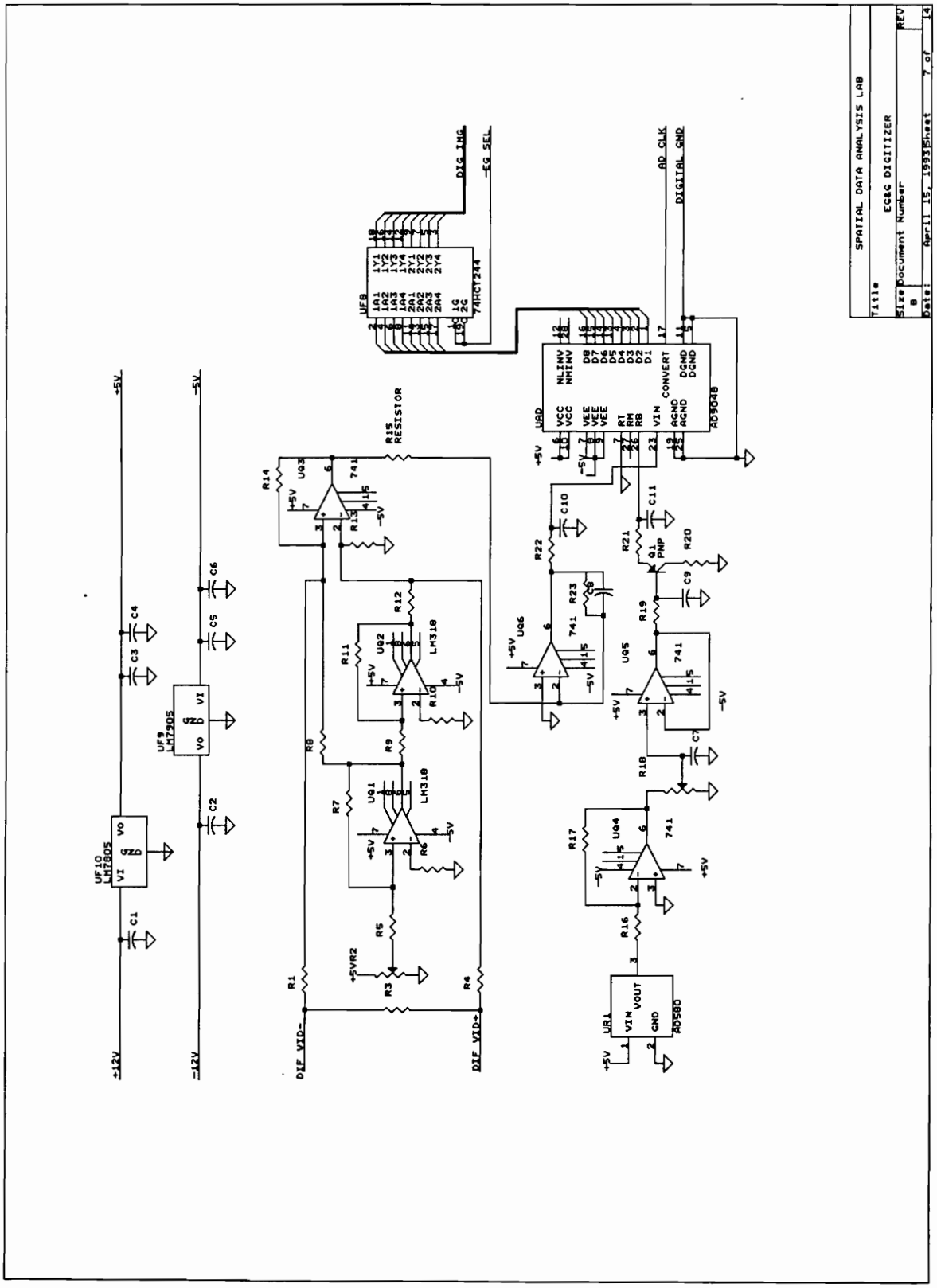

SPATIAL DATA ANALYSIS LAB
Title
DMA HANDLING
Size B
Document Number
REV
Date: January 10, 1993 Sheet 2 of 14
-CHRES
-DREQ
-MCA BURST
UH9
74HCT151
UG7A
74HCT32
UH6A
74HCT74
UJ7A
74LS03
UJ7C
74LS03
UF7F
74HCT04
UH6B
74HCT74
UF7C
74HCT04
UG7D
74HCT32
DACK
SUSPEND
-BURSTMODE
CHRESET
DMA MASK
D3
-IOW
-EMPTY
-DMA DIS
-HF
-BURST

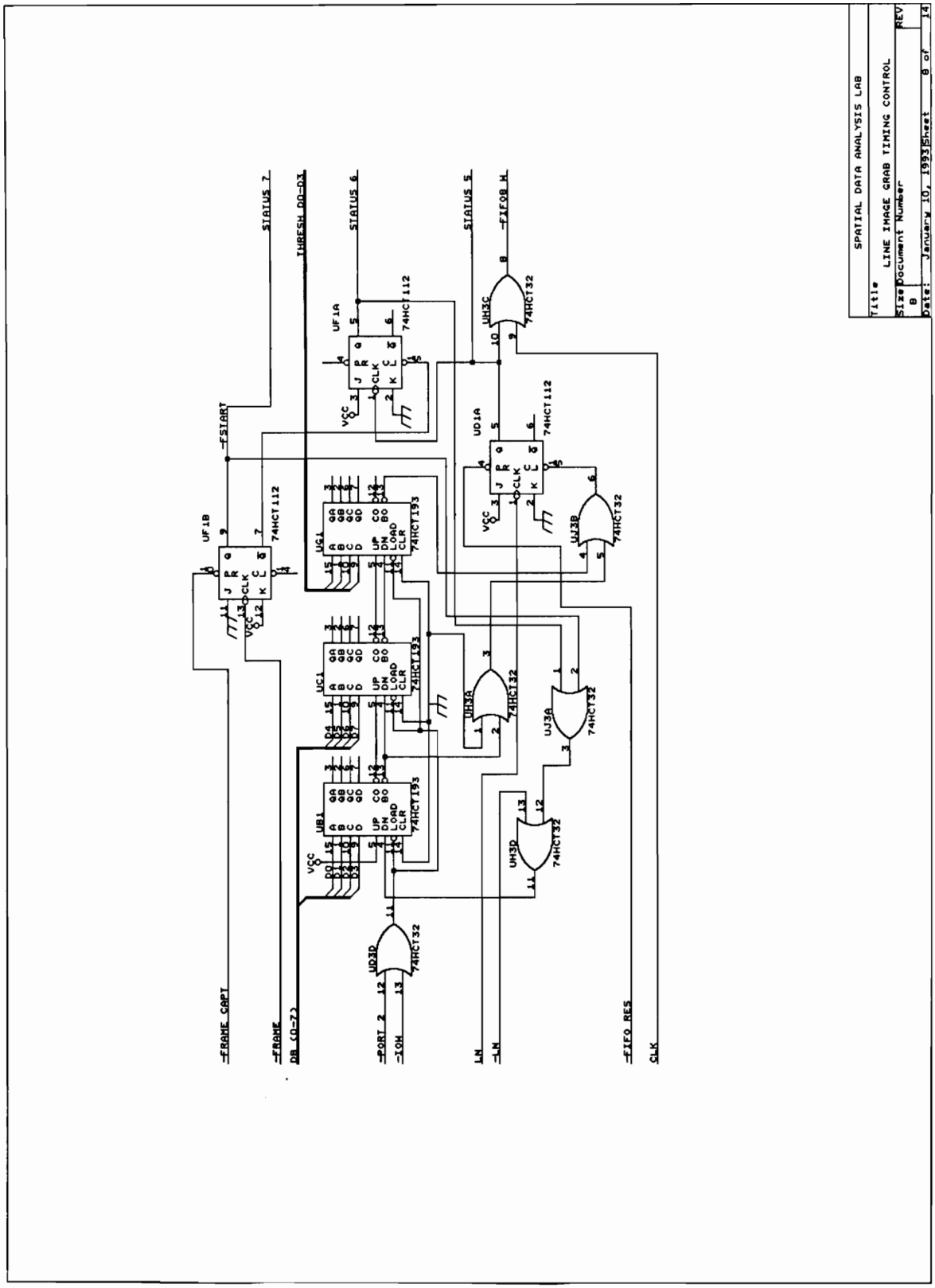

SPATIAL DATA ANALYSIS LAB
Title
INTERUPT HANDLER
Size
B
Document Number
REV
Date: January 10, 1993
Sheet
3 of
14
STATUS 2
STATUS 4
-DMA DIS
-IREQ
DMA MASK
UB3B
74HCT02
UJ7B
74LS03
UJ7D
74LS03
UK7A
74HCT74
U19B
74HCT109
VCC
UG7C
74HCT32
UF7E
74HCT04
UH3B
74HCT32
-FIFO RES
-FF
DACK
-IC
DS
-PORT SW 5
-CHRESET

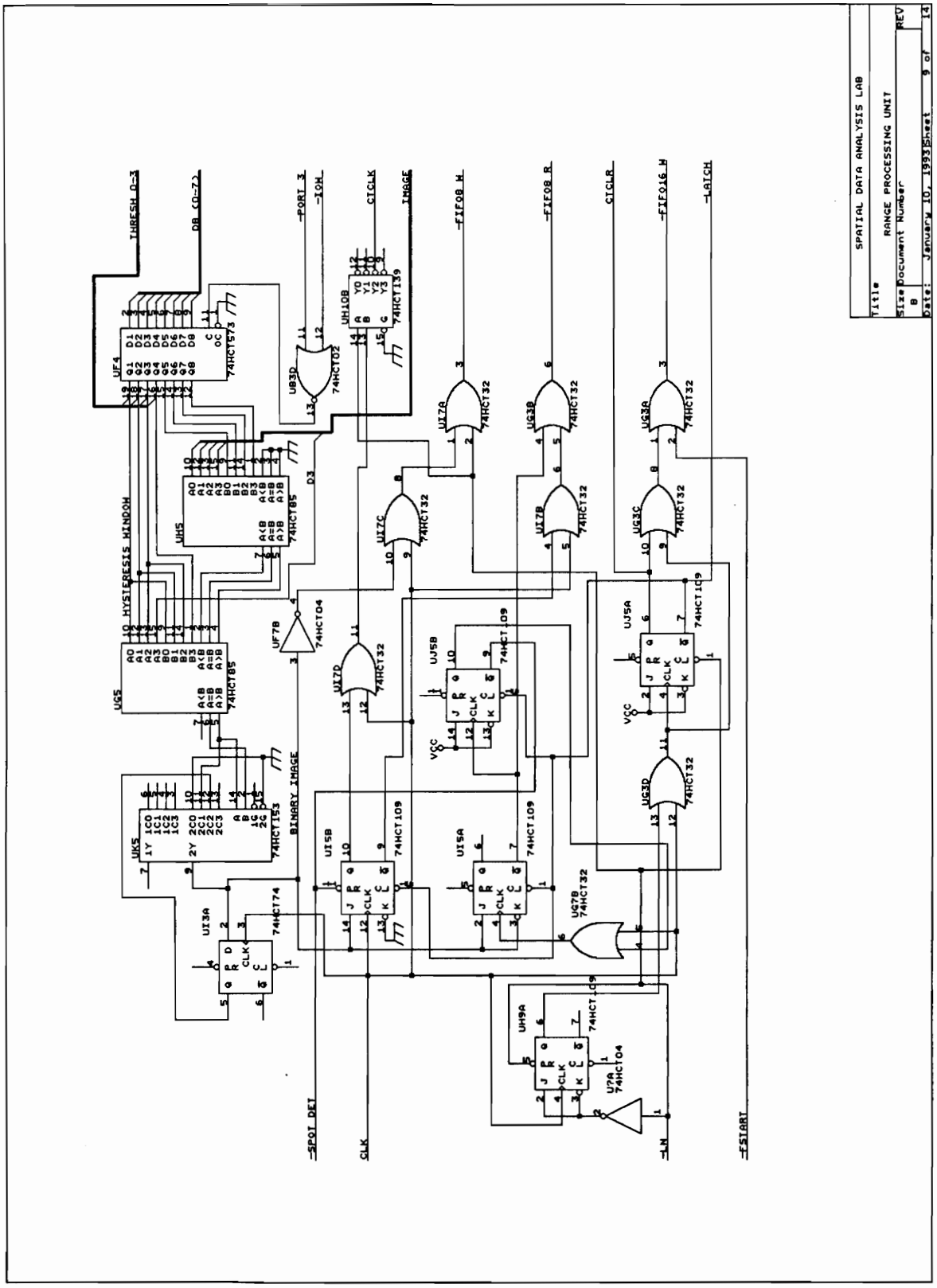

SPATIAL DATA ANALYSIS LAB
COMMAND REGISTER & CAMERA SELECT
Document Number
January 10, 1993
Sheet 4 of 14
-PORT H
DIGITAL IMAGE
-IOR
-RS170 SEL
-EG SEL
INTENSITY
FIFOEMPTY
-INTERRUPT
SENSOR
-FLOOF
LINECOUNTER
FRAMECAPTURE
74HCT573
74HCT139
74HCT04
74HCT02
74HCT374
74HCT244
74HCT32
74HCT74
-PORT A
-PORT B
-PORT J
-CHRES
-PORT D
-IOW
DB (0-7)
-F GRAB
-FIFO RES
-PORT 9

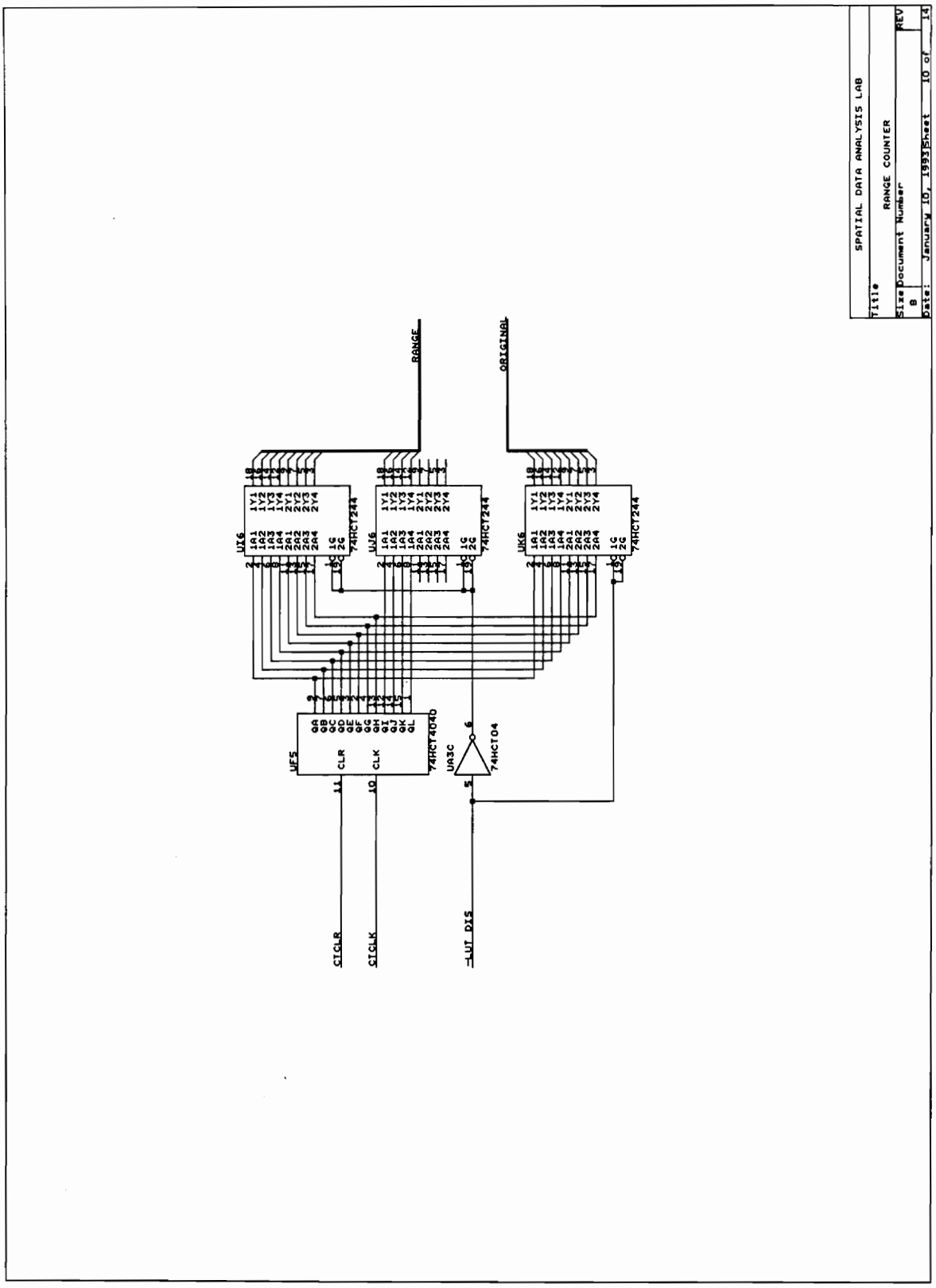

SPATIAL DATA ANALYSIS LAB
RS170 DIGITIZER
January 10, 1993

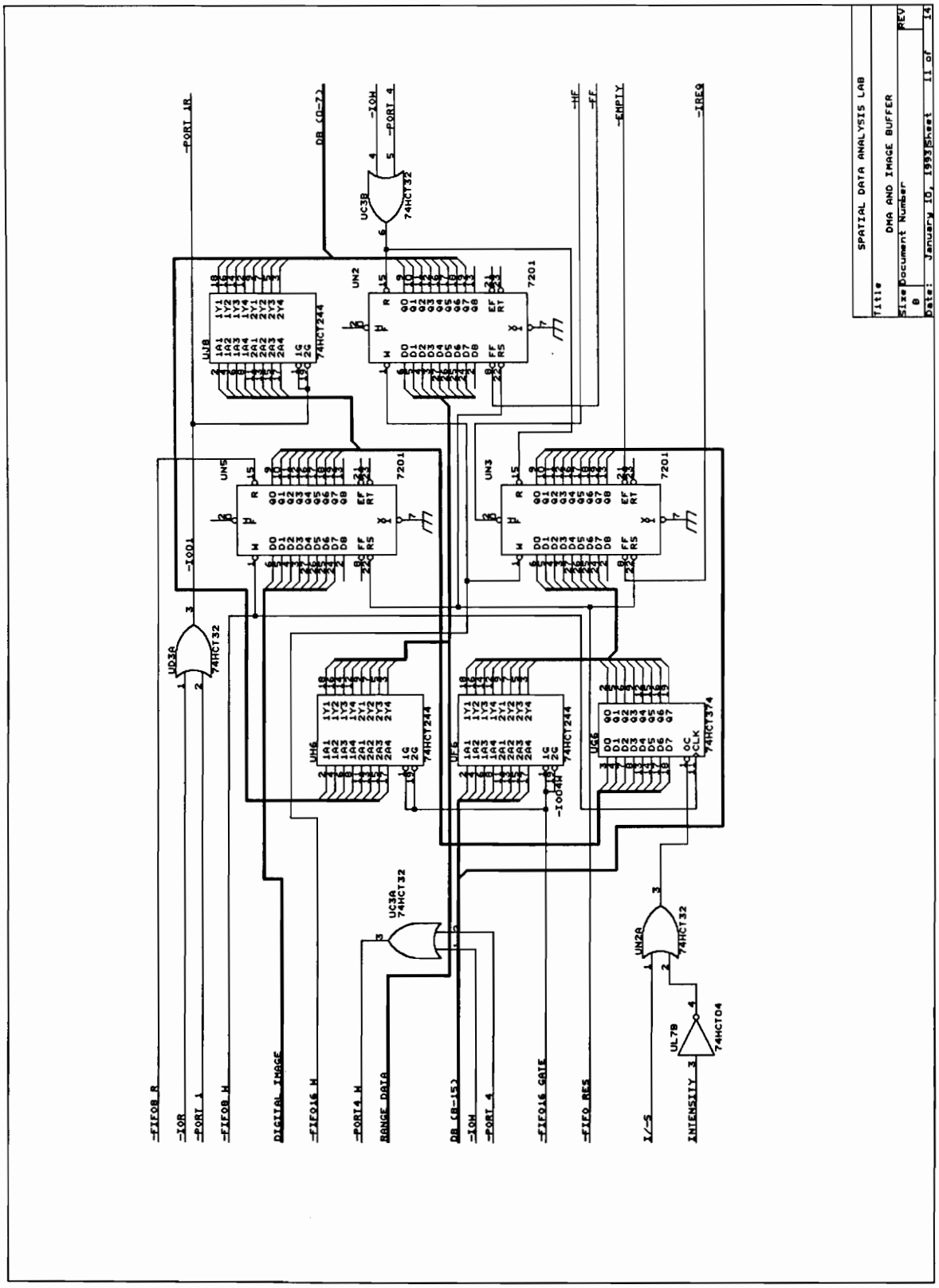

SPATIAL DATA ANALYSIS LAB
EG&G CAMERA RECEIVER
Size B
Document Number
REV
Date: January 10, 1993
Sheet 6 of 14
-EG SEL
74HCT240
74LS174
3486
3487
CONNECTOR DB25

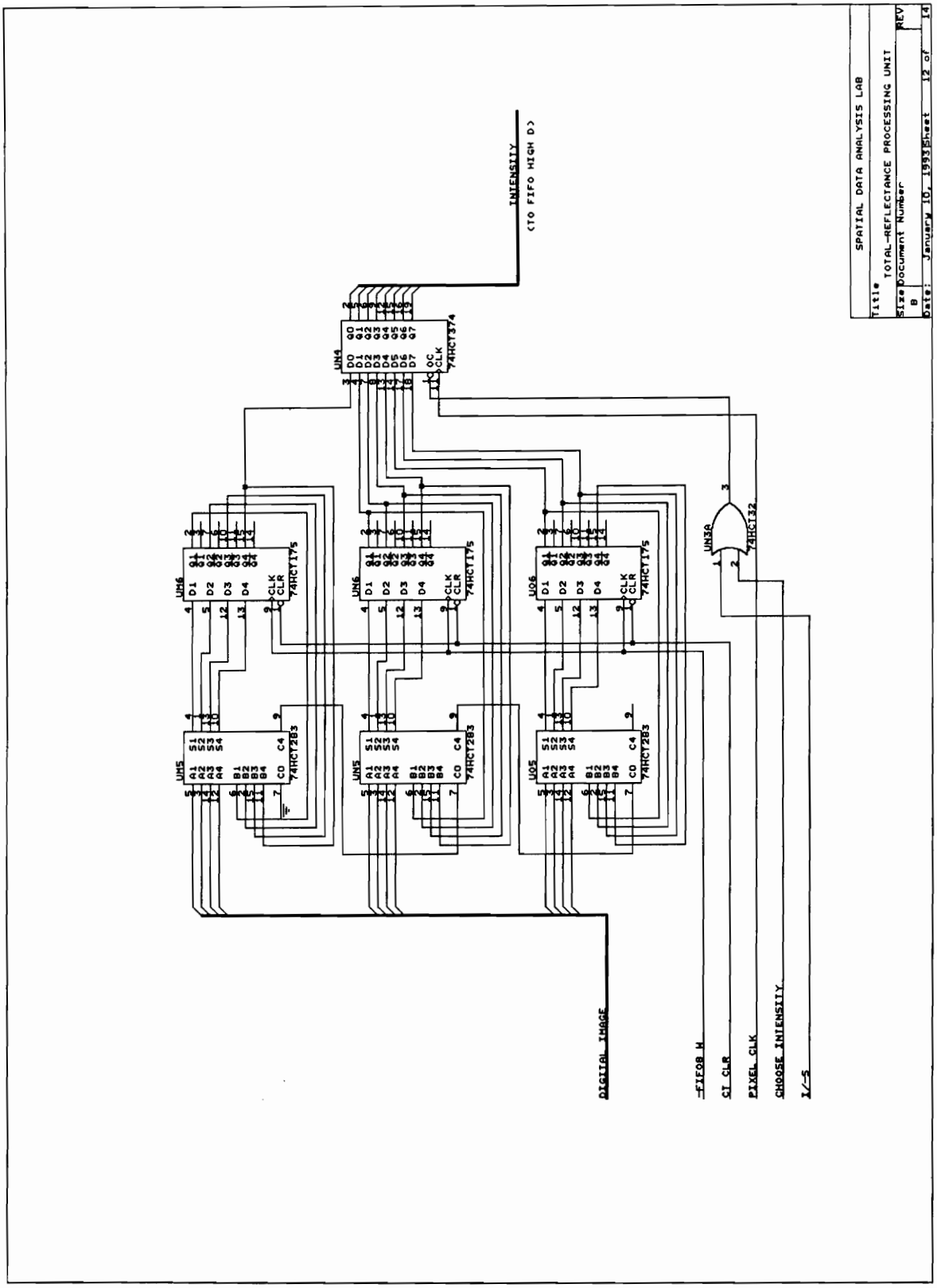

SPATIAL DATA ANALYSIS LAB
Title
E&G DIGITIZER
Size
B
Document Number
REV
Date:
April 15, 1993
Sheet
7 of
14
DIG_IMG
-EG_SEL
AD_CLK
DIGITAL_GND
DIF_VID-
DIF_VID+
+5V
-5V
+12V
-12V
R15 RESISTOR
LM318
741
AD580
AD9048
74HCT244
LM7805
LM7905

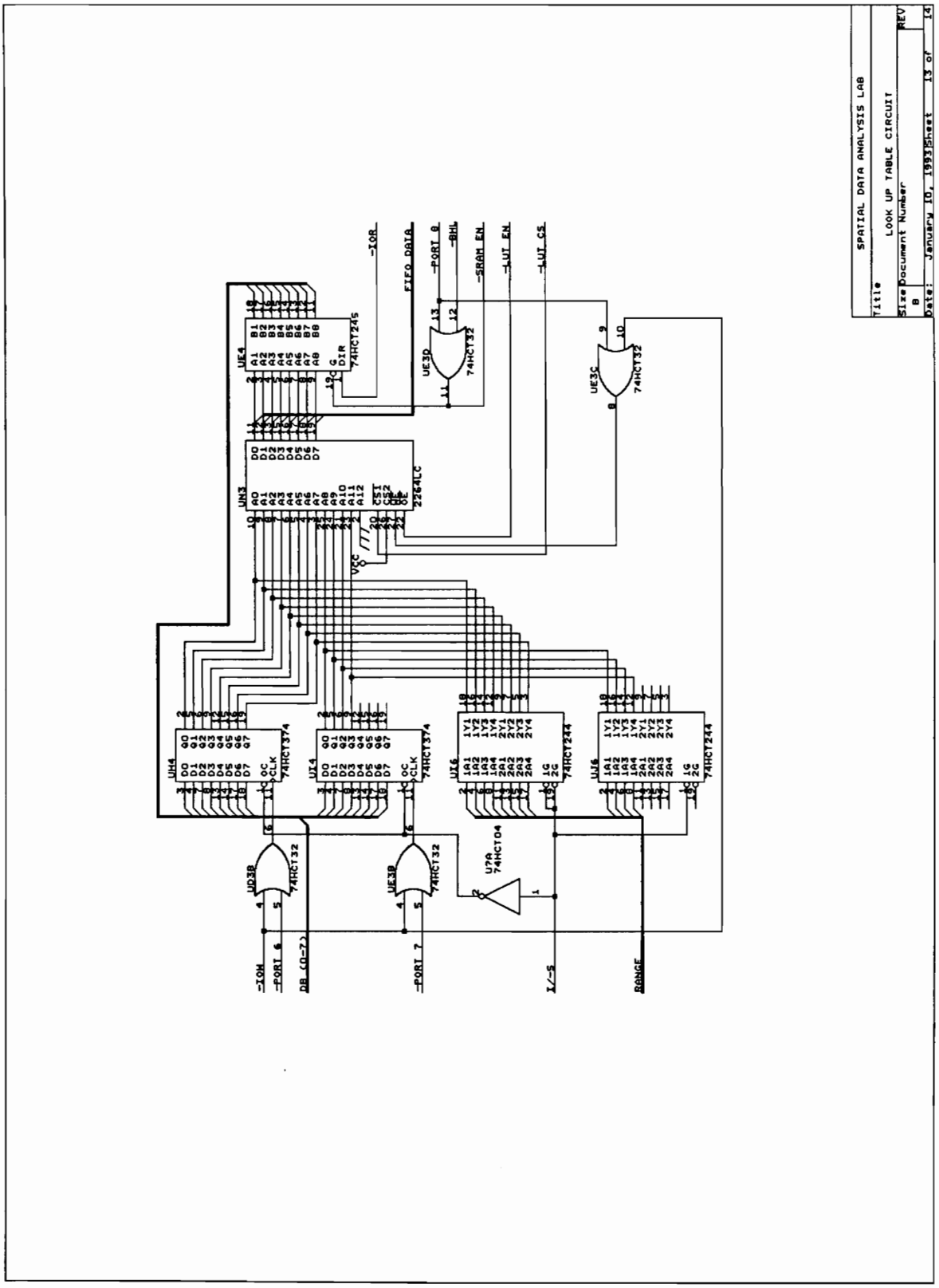

SPATIAL DATA ANALYSIS LAB
Title
LINE IMAGE GRAB TIMING CONTROL
Size
B
Document Number
REV
Date: January 10, 1993
Sheet 8 of 14
STATUS 7
STATUS 6
STATUS 5
-FIFO8 HF
THRESH D0-D3
-FSTART
UF1B
UF1A
UG1
UC1
UB1
UD1A
UH3C
UJ3B
UH3A
UJ3A
UH3D
UD3D
74HCT112
74HCT193
74HCT32
VCC
-FRAME CAPT
-FRAME
DB (0-7)
-PORT 2
-IOM
LN
-LN
-FIFO RES
CLK

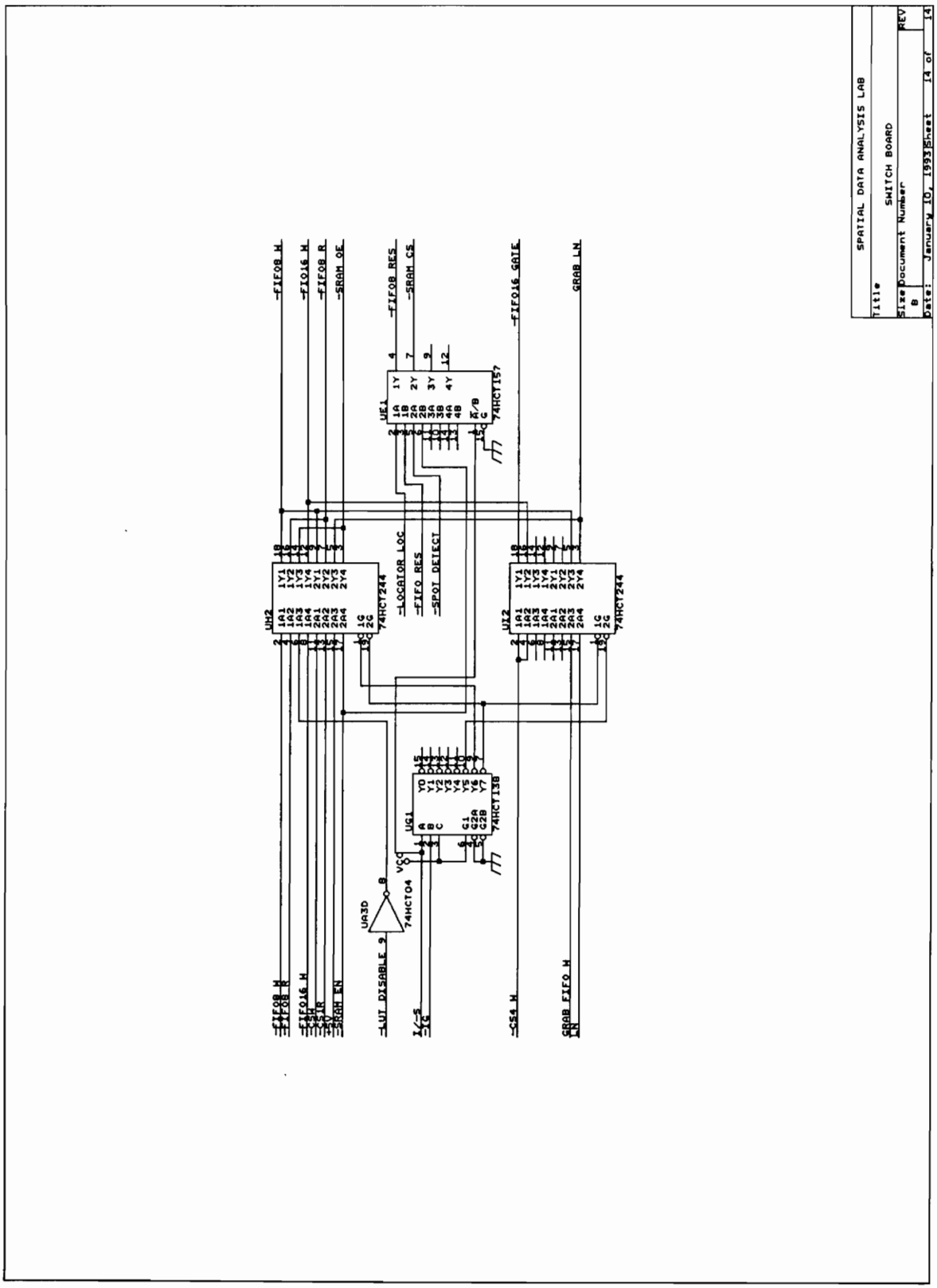

SPATIAL DATA ANALYSIS LAB
Title RANGE PROCESSING UNIT
Size B Document Number REV
Date: January 10, 1993 Sheet 9 of 14
THRESH 0-3
DB 00-23
-PORT 3
-IOH
CIGLK
IMAGE
-FIF08 M
-FIF08 R
CIGLR
-FIF016 M
-LATCH
HYSTERESIS WINDOW
BINARY IMAGE
-SPOT DET
CLK
-LN
-FSTART
VCC
74HCT573
74HCT02
74HCT139
74HCT885
74HCT153
74HCT74
74HCT04
74HCT32
74HCT109


SPATIAL DATA ANALYSIS LAB
Title
RANGE COUNTER
Size
B
Document Number
REV
Date: January 10, 1993
Sheet 10 of 14
RANGE
ORIGIN0L
UJ6
UJ6
UK6
74HCT244
UF5
74HCT4040
UA3C
74HCT04
SIGLR
SIGLK
-LUT DIS


SPATIAL DATA ANALYSIS LAB
DMA AND IMAGE BUFFER
Size Document Number REV
B
Date: January 10, 1993 Sheet 11 of 14
-PORT 1R
DB (0-7)
-IOW
-PORT 4
-HF
-FF
-EMPTY
-IREQ
-IOR1
-IOR
-PORT 1
-FIFOR H
DIGITAL IMAGE
-FIFO16 H
-PORT4 H
RANGE DATA
DB (8-15)
-IOW
-PORT 4
-FIFO16 GATE
-FIFO RES
I/-S
INTENSITY
UD3B
UC3B
UN2
UN5
UN3
UH6
UE6
UG6
UC3A
UN2A
UL7B
74HCT32
74HCT244
74HCT374
74HCT04
7201


SPATIAL DATA ANALYSIS LAB
Title TOTAL-REFLECTANCE PROCESSING UNIT
Size Document Number REV
B
Date: January 10, 1993 Sheet 12 of 14
INTENSITY
(TO FIFO HIGH D)
UN4
74HCT374
UN6
74HCT175
UN5
74HCT283
UO6
UO5
UN36
74HCT32
DIGITAL IMAGE
-FIFOB W
SI CLR
PIXEL CLK
CHOOSE INTENSITY
I/-S


SPATIAL DATA ANALYSIS LAB
Title LOOK UP TABLE CIRCUIT
Size B Document Number REV
Date: January 10, 1993 Sheet 13 of 14
-IOR
FIFO DATA
-PORT 8
-8HL
-SRAM EN
-LUT EN
-LUT CS
UE4
74HCT245
UE3D
74HCT32
UE3C
74HCT32
UN3
2264LC
UH4
74HCT374
UI4
74HCT374
UJ6
74HCT244
UK6
74HCT244
UD3B
74HCT32
UE3B
74HCT32
U7A
74HCT04
-IOH
-PORT 6
DB (0-7)
-PORT 7
I/-S
RANGE


SPATIAL DATA ANALYSIS LAB
Title
SWITCH BOARD
Size
B
Document Number
REV
Date: January 10, 1993
Sheet 14 of 14
-FIFOB_H
-FIFO16_H
-FIFOB_R
-SRAM_OE
-FIFOB_RES
-SRAM_CS
-FIFO16_GATE
GRAB_LN
UH2
74HCT244
UE1
74HCT157
UI2
74HCT244
UG1
74HCT138
UA3D
74HCT04
-LOCATOR_LOC
-FIFO_RES
-SPOT_DETECT
-LUT_DISABLE_9
-S54_H
GRAB_FIFO_H

Appendix B

Allocation Description File of The Range Sensing Hardware

This appendix provides the Allocation Description File of the range sensing hardware for use on the Micro Channel Architecture

```
AdapterId 059CDh
AdapterName "Stereo Scanner System"
NumBytes 3
NamedItem
Prompt "I/O Address Select"
choice "Command Register 0300h" pos[1]=xx000xxxb io 0300h-030fh
choice "Command Register 0380h" pos[1]=xx001xxxb io 0380h-038fh
choice "Command Register 1300h" pos[1]=xx010xxxb io 1300h-130fh
choice "Command Register 1380h" pos[1]=xx011xxxb io 1380h-138fh
choice "Command Register 2300h" pos[1]=xx100xxxb io 2300h-230fh
choice "Command Register 2380h" pos[1]=xx101xxxb io 2380h-238fh
Help "This selects the I/O address range."
NamedItem
Prompt "Arbitration Level Selection"
choice "Level 0" pos[3]=xx1x0000b arb 0
choice "Level 1" pos[3]=xx1x0001b arb 1
choice "Level 2" pos[3]=xx1x0010b arb 2
choice "Level 3" pos[3]=xx1x0011b arb 3
choice "Level 4" pos[3]=xx1x0100b arb 4
choice "Level 5" pos[3]=xx1x0101b arb 5
choice "Level 6" pos[3]=xx1x0110b arb 6
choice "Level 7" pos[3]=xx1x0111b arb 7
choice "Level 8" pos[3]=xx1x1000b arb 8
choice "Level 9" pos[3]=xx1x1001b arb 9
choice "Level 10" pos[3]=xx1x1010b arb 10
choice "Level 11" pos[3]=xx1x1011b arb 11
choice "Level 12" pos[3]=xx1x1100b arb 12
choice "Level 13" pos[3]=xx1x1101b arb 13
choice "Level 14" pos[3]=xx1x1110b arb 14
Help "This selects the DMA arbitration level."
NamedItem
Prompt "Select Fairness"
choice "Fairness Enabled" pos[3]=xxx0xxxxb
choice "Fairness Disabled" pos[3]=xxx1xxxxb
Help "This selects whether or not fairness is implemented."
```

Appendix C

Utility Subroutine Listings

This appendix contains program listings of the utility software used in the control of the range sensing hardware

```
/*****************************************************************/
/* File Name : scanner.h                                         */
/*  Programmer: Y. Jason Hou                                     */
/*                                                               */
/* Definitions for range sensing software management             */
/*****************************************************************/
#ifdef  RS170                   /* definitions of frame size of RS170 camera */
#define ROW_WIDTH       240
#define COL_WIDTH       410
#endif
#ifdef  EG_G                    /* definition sof frame size of MC9128 camera */
#define ROW_WIDTH       128
#define COL_WIDTH       128
#endif

#define ComPort         0x1380          /* system I/O register addresses */
#define ImageRowBuffer  0x1381
#define RowNum          0x1382
#define Thresh          0x1383
#define DMAIO           0x1384
#define LUTAddr         0x1386
#define LUTIO           0x1388
#define CamSim          0x1389
#define Status          0x138a
#define CamSel          0x138b
#define FreqDiv         0x138c

#define CHANNEL         6       /* DMA channel */
#define HIGH            50      /* logic high */
#define LOW             0       /* logic low */

#ifdef  DMA_8
#define BUFFER_SIZE     12800U
#endif
#ifdef  DMA_16
#define BUFFER_SIZE     25600U
#endif

typedef struct {                /* semaphore structure for interrupt routine */
   unsigned char far* buffer;
   int full;
   int number;
   unsigned char low_byte;
```

```
   unsigned char high_byte;
   unsigned char page;} DMA_buffers;

void stereo_initialize(void);
void stereo_close(void);
void cam_intensity_select( unsigned char);
void set_subsampling( unsigned char);
int  row_capture( unsigned char*, int, int);
void interrupt_setup(void);
void LUT_setup (int, unsigned char);
void DMA_setup(int);
void DMA_destination (int);
void set_threshold( unsigned char);
void scanner_enable(void);
void FIFO_clear(void);
int  create_buffers(void);
void free_buffers(void);
void row_display( unsigned char *, int, int, int);
void fgrab(int, int);
void setup_LUT(int, int);
void find_parameters(int*, int*, unsigned char*);
```

```
/*************************************************************************/
/*  Program Name:  Utility Subroutines                                   */
/*  Programmer:  Y. Jason Hou                                            */
/*                                                                       */
/* This pakage contains utility subroutines used to control the range sensing hardware. */
/*************************************************************************/

#define EG_G                    /* select MC9128 EG&G camera */
#define DMA_16                  /* choose 16 bits DMA transfers */
#include <stdio.h>
#include <alloc.h>
#include <math.h>
#include <dos.h>
#include <scanner.h>            /* definition file for software programming */

extern int scanning_done_flag;      /* flag for scanning completion */
extern int hardware_error;          /* flag for hardware error reporting */
extern int page_count;              /* numer of memory pages being filled */
extern int current_buffer;          /* the memory page buffer currently being used */
extern int total_buffers;           /* the number of memory buffers being allocated */
extern int PAGES_NEEDED;    /* the total number of memory pages to be scanned */
extern DMA_buffers buffers[20];     /* semophores of memory pages set by interrupt */

/* Enable command register and clear command register */
void stereo_initialize(void)
{
outportb(ComPort, 0x7d);            /* Enable command port and clear FIFO */
}

/* Terminate sensor operations */
void stereo_close(void)
{
outportb(ComPort, 0x0f);
outportb(ComPort, 0xff);
}

/*  Select the video source and the intensity processing unit to be used   */
void cam_intensity_select(unsigned char cam)
{
outportb(CamSel, cam);
}

/* Set slow down factor of scanning speed */
```

```
void set_subsampling(unsigned char number)
{
outportb(FreqDiv, number);
}

/* Clear system semaphores for interrupt service routine */
void interrupt_setup(void)
{
int i;
page_count=0;
for (i=0; i<20; i++) {
        buffers[i].buffer=0;          /* clear buffer number semaphore */
        buffers[i].full=0;            /*  clear page full semaphore */
        buffers[i].number=0;          /* clear page size semaphore */
        buffers[i].low_byte=0;        /* clear page starting address */
        buffers[i].high_byte=0;       /* clear page high byte address */
        buffers[i].page=0;            /* clear page number semaphore */
        }
}

/* Interrupt service routine of Stereo Scanner System using IRQ 9          */
/*              Major function :  Memory buffer assignment.                */
/*    Memory management error code :                                       */
/*           hardware_error = 1 :  DMA FIFO overflow                       */
/*                          = 2 :  Empty memory buffer not                 */
/*                                                                         */
void interrupt interrupt_handler ()
{
int i,k;
unsigned char status;

status=inportb(Status);
hard_status=status;
page_count++;                              /* Advance to a new page */
if ((inportb(status)&0x10) == 0 ) {        /* Overflow happens, terminate process */
   hardware_error=1;                       /* Error code 1 for FIFO overflow */
   outportb(ComPort, 0x0f);                /* stop scanning process */
   outportb(ComPort, 0x7f);                /* send EOI to Comman Register */
   outportb(0xA0, 0x20);                   /* interrupt controller EOI */
   outportb(0x20, 0x20);
   return;
   }
buffers[current_buffer].full=1;        /* set full flag to the current memory buffer */
```

```
buffers[current_buffer].number=page_count;  /* set page number to the current buffer */
if ( page_count == PAGES_NEEDED)  {     /* Done with required scanning pages */
   scanning_done_flag=1;                /* set scanning completion flag */
   if ((inportb(status)&0x10) == 0) {
        hardware_error=1;               /* hardware error */
        }
   outportb(ComPort, 0x0f);             /* terminate scanning process */
   outportb(ComPort, 0x7F);
   outportb(ComPort, 0xf);
   outportb(0xA0, 0x20);
   outportb(0x20, 0x20);
   return;
   }
else  {
   for (k=0; k<total_buffers;k++) {     /* Seek next empty memory page */
        i=current_buffer+k;
        if (i>=total_buffers) i=i-total_buffers;
        if (buffers[i].full == 0 )  {   /* found an empty memory page */
           DMA_destination(i);          /* reprogram DMA slave controller */
           current_buffer=i;            /* track the memory page being used */
           if ((inportb(Status)&0x10) == 0) {        /* FIFO overflow */
                hardware_error=1;       /* Error code 1 for FIFO overflow */
                outportb(ComPort, 0x0f);    /* reinitiate scanning process */
                outportb(ComPort, 0x7f);
                outportb(ComPort, 0xf);
                k=total_buffers+1;
                break;
                }
           outportb(ComPort, 0xC6);  /*  reinitiate scanning process */
           k=total_buffers+1;
           break;
           }
        }
   }

if (k==(total_buffers+1)) {
   outportb(0xA0, 0x20);
   outportb(0x20, 0x20);
   return;
   }

/* Empty buffer not found */
hardware_error=2;  /* Error code 2 for not finding empty buffers */
```

```
outportb(ComPort, 0x0f);  /* EOI and terminate scanning process */
outportb(ComPort, 0x7f);
outportb(0xA0, 0x20);
outportb(0x20, 0x20);
return;
}

/* Set up the DMA slave controller on the channel assigned by CHANNEL. */
/* Stereo scanner DMA IO port is at DMAIO                              */
void DMA_setup ( int buffer)
{
outportb(0x18,(0x90+CHANNEL) );          /* Channel mask */
outportb(0x18, CHANNEL);                 /* Clear byte pointer and set IO register */
outportb(0x1A, (unsigned char) DMAIO);   /* Lower byte of IO address */
outportb(0x1A, DMAIO>> 8);               * Higher byte of IO address */
outportb(0x18, 0x20+CHANNEL);            /* DMA IO transfer address setup */
outportb(0x1A, buffers[buffer].low_byte ); /* Low byte of buffer pointer*/
outportb(0x1A, buffers[buffer].high_byte ); /* High byte */
outportb(0x1A, buffers[buffer].page);    /* Set up page register */
outportb(0x18, (0x40+CHANNEL));          /* Set up DMA count */
outportb(0x1A, (unsigned char) ((BUFFER_SIZE-1)>> 8) );*/
outportb(0x1A, (unsigned char) ( (BUFFER_SIZE>>1) -1));
outportb(0x1A, (unsigned char) ( ( (BUFFER_SIZE>>1) -1) >>8) );
outportb(0x18, (0x70+CHANNEL)); /* Mode register function */
outportb(0x1A, 0x4d);  /* 16 bit, write mem, programmed IO, transfer */
outportb(0x18, (0xA0+CHANNEL)); /* Reset Mask bit */
}

/* Assignd DMA slave controller the memory buffer address and transfer count */
void DMA_destination ( int buffer )
{
outportb(0x18, 0x90+CHANNEL);
outportb(0x18, 0x20+CHANNEL);            /* Clear byte pointer and set Mem register */
outportb(0x1A, buffers[buffer].low_byte ); /* Low byte of buffer pointer*/
outportb(0x1A, buffers[buffer].high_byte ); /* High byte */
outportb(0x1A, buffers[buffer].page);    /* Set up page register */
outportb(0x18, (0x40+CHANNEL));          /* Set up DMA count */
outportb(0x1A, (unsigned char) ((BUFFER_SIZE>>1)-1) );  /*16 bit DMA transfers*/
outportb(0x1A, ( (BUFFER_SIZE>>1)-1) >>8);
outportb(0x18, 0xA0+CHANNEL);
}

/* Set threshold value into the threshold register */
```

```
void set_threshold ( unsigned char threshold)
{
outportb(Thresh, threshold);
}

/* Enable scanner system, enable DMA, burst mode on */
void scanner_enable(void)
{
outportb(ComPort,0x7F);
outportb(ComPort,0x1F);
outportb(ComPort, 0x06);   /* DMA burst mode */
}

/* Clear FIFO buffer */
void FIFO_clear(void)
{
outportb(ComPort, 0x7f);
}

/* Create memory buffers for DMA transfers */
int create_buffers(void)
{
unsigned long empty_memory;
unsigned long available;
unsigned long l,m;
unsigned char far *block;
int i;

empty_memory=farcoreleft();
available = empty_memory/BUFFER_SIZE;
if ( available > 30)                    /* limits the maximal number of memory pages *
   total_buffers=30;                    /* to be used */
else
   total_buffers=(int) available;
printf("\n Total_buffers before processing %d", total_buffers);
for (i=0; i<total_buffers; i++) {
   if ( NULL== (block=farmalloc(BUFFER_SIZE+20)) ) {
         printf("\n Error allocating %d buffers, %lx", i,buffers[i].buffer);
         farfree(block);                        /* memory allocation fails 8/
         total_buffers=i;
         break;
         }
   buffers[i].buffer=block;
```

```
    }
for (i=0;i<total_buffers; i++) {          /* set starting addresses of allocated memory */
    l=(unsigned char far *)buffers[i].buffer;          /* for interrupt routine */
    m= ((l>> 16)<< 4) + (unsigned int) l;
    buffers[i].page= (unsigned char) (m>> 16);
    buffers[i].low_byte=(unsigned char) m;
    buffers[i].high_byte=(unsigned char) (m>> 8);
    }
if (total_buffers == 0 ) return(-1);                    /* fail to allocate any memory */
else return(0);
}

/*  Free memory buffers to DOS after finishing scanning */
void free_buffers(void)
{
int i;
for (i=0; i< total_buffers; i++)
   farfree(buffers[i].buffer);
}

/*  The kernal function for grabing a single row in a frame.                    */
/*  Parameter description:                                                      */
/*        row       : unsigned char array of row buffer.                        */
/*        row_number : the number of row to be grabbed in a frame.              */
/*        size      : the size of the row image to be grabbed.                  */
/*  Return value   : 0 for successful return.                                   */
/*                   -1 for error return.                                       */
int row_capture( unsigned char *row, int row_number, int size)
{
long i;
int k;

outportb(ComPort, 0x7f);                          /* Clear FIFO buffer */
outportb(Thresh, (unsigned char) row_number>> 8); /* set the high 4 bits of row number
*/
outportb(RowNum, (unsigned char) row_number);   /* set lower 8 bits of row number */
outportb(ComPort, 0x3b);              /* Frame clear, image grab mode, fifo clear */

for (i=0; i<= 80000L; i++) {          /* waiting to capture a frame index */
   if ( i ==80000L ) return(-1);      /* break if not found */
   if ( (inportb(Status) & 0x80) == 0) break;
   }
for (i=0; i<= 80000L; i++) {          /* waiting to capture a line start */
```

```
    if ( i == 80000L ) return(-1);          /* break if not found */
    if ( (inportb(Status) & 0x40) == 0x40) break;
    }
for (i=0; i<= 80000L; i++) {                /* waiting for a line to finish */
    if ( i == 80000L ) return(-1);          /* break if not found */
    if ( (inportb(Status) & 0x20) == 0x20) break;
    }
outportb(ComPort, 0xff);                    /* Return to computer access mode */
for (k=0; k < size; k++)
    *(row+k) = inportb(ImageRowBuffer);  /* transfer data from hardware to computer */
return(0);
}

/* Grab a frame and show it onto the screen */
/* Has to enable 300X200 graphic mode beforhand*/
void fgrab(int x_dim, int y_dim)
{
int i,j;
unsigned char row_buffer[500];

for (i=0;i < x_dim; i++)  {
    if ( row_capture(row_buffer, i+256, x_dim) == -1 ) {/* capture lines by incrementing */
            printf("\n Camera not ready !");                    /* the row number */
            break;
            }
    for (j=0; j <y_dim; j++) {
            write_dot(i, j, row_buffer[j]+30);
            }
    }
}

/* Set up Look up Table within the range specified */
void setup_LUT(int maximum, int minimum)
{
unsigned int i;
double difference;
double increment;
double temp;
unsigned int a;

difference=(double)maximum-(double)minimum;     /* do a uniform expansion */
if (difference != 0)
        increment=256.0/difference;
```

```
if (difference != 0)
   for (i=(minimum+1); i<maximum; i++) {
         outport(LUTAddr, i);                    /* set LUT memory address */
         temp=((double)(i-minimum))*increment;
         a = (unsigned int)temp;
         outportb(LUTIO,(unsigned char) a);      /* write LUT entries */
         }
for (i=maximum; i<=3095; i++)  {
   outport(LUTAddr,i);
   outportb(LUTIO, 0xff);
   }
for (i=0; i <= minimum; i++) {
   outport(LUTAddr, i);
   outportb(LUTIO, 0xff);
   }
}
```

Appendix D

Programming Examples

This appendix contains programming examples for controlling the range sensing hardware.

```
/*********************************************************************/
/*  Program Name:  fgrab.c                                           */
/*  Programmer:  Y. Jason Hou                                        */
/*                                                                   */
/*  This program shows the image observed by the camera installed in the range  */
/*   Range Sensing Hardware.                                         */
/*********************************************************************/
#define EG_G  /* choose to use MC9128 as the range sensing camera */
#define DMA_16
#include <stdio.h>
#include <graphics.h>
#include <scanner.h>
#include <image.h>

int scanning_done_flag;  /* initilize semaphores passed between the main program and */
int hardware_error;              /* the interrupt subroutine */
int page_count;
int current_buffer;
int total_buffers;
int PAGES_NEEDED;
DMA_buffers buffers[20];

void main(void)
{
set_subsampling(0);              /*  reset frequency divider */
stereo_initialize();             /* enable the range sensing hardware */
cam_intensity_select(0);         /* select the MC9128 camera */
FIFO_clear();
enable_graphics();               /* enable graphic display */
setup_graphics();
for (;;)
        {
        fgrab(128*2, 128);       /* grab video lines and show them on the screen */
        if (kbhit())
                if (getch()==0x1b) break;       /* break if ESC is found */
        }
set_graphics_mode(3);  /* return to text mode */
}
```

```
/*****************************************************************/
/*  Program Name:  scan.c                                        */
/*  Programmer:  Y. Jason Hou                                    */
/*                                                               */
/*  This program executes real time filtering of range and intensity images and save  */
/*   the processed images to storage disks in concurrent with the scanning process.   */
/*   Usage:   scan filename                                      */
/*                 filename is the file name of the output image.                      */
/*****************************************************************/
#define EG_G          /* chose to use MC9128 as the range sensing camera */
#define DMA_16        /* enable 16 bit DMA transfers for range and intensity data */
#include <stdio.h>
#include <alloc.h>
#include <dos.h>
#include <fcntl.h>
#include <sys\stat.h>
#include <scanner.h>
#include <range.h>

/* global variable initialization */
int scanning_done_flag = 0;
int hardware_error = 0;
int page_count = 0;
int current_buffer=0;
int total_buffers = 0;
int PAGES_NEEDED = 30;
DMA_buffers buffers[20];

/* command line can specifies an output image file to be  stored*/
void main(int argc, char** arg[])
{
int i,j,k;
unsigned char subsampling[1280];
unsigned int sum,temp1;
unsigned char temp;
char D;
int page=0;
int p,q,r,line;
int count;
long l;
int file_handle;
char filename[30];
```

```
if (argc <2) {
   printf("Please specify the image file name to be saved!\n");
   exit (0);
   }
strcpy(filename,arg[1]);

printf("Specify the number of pages to be scanned. (100 lines per page)\n");
printf("Number :");
scanf("%d",&PAGES_NEEDED);
printf("Generate the grayscale images in Center-pixel or Total-reflectance?\n");
printf("T(Total-reflectance) or C(Center-pixel) :");
while(((C=getch())!='T')&&(C!='C')) printf("%c\nT or C:",C);

set_subsampling(0);                    /* reset frame rate subsampling */
stereo_initialize();
switch(C) {            /* select the camera source and intensity processor to be used */
   case 'T':
          cam_intensity_select(0);
          break;
   case 'C':
          cam_intensity_select(0x10);
          break;
   }

interrupt_setup();                     /* clear semaphores for interrupt signalling */
stereo_scanner_int(interrupt_handler);       /* install interrupt service routine */

/* open the file to be written for images */
if (-1 == (file_handle=open(filename, O_CREAT|O_WRONLY|O_BINARY,
                                       S_IREAD|S_IWRITE) ))  {
   printf("\n Cannot open image file");
   close(file_handle);
   exit(0);
   }
if (create_buffers() == -1)  {   /* Create buffers for DMA operations */
        free_buffers();
        printf("Can't create memory buffers!\n");
        exit(1);
        }
enable_graphics();             /* enable 300X200 graphic display mode */
setup_graphics();
DMA_setup( 0);                 /* program the first memory page for DMA transfers */
```

```
printf("\n Please hit any key to continue..");
getch();
FIFO_clear();
scanner_enable();
page=1;
i=0;
line=0;

/* start range scanning process  */
for (;hardware_error==0;i++) {                  /* monitor hardware status */
   if (i >= total_buffers) i=0;                 /* memory page search pattern */

   if (buffers[i].number==page)  {      /* find serial pages to process */
      page++;
      for (p=0;p<5;p++) {       /* implement 20 lines into one average filtering */
         temp1=p*20;
          for (r=0;r<256;r++) {
            sum=0;
            for (q=0;q<20;q++)
            if ((temp=*(buffers[i].buffer+((q+temp1)<<8)+r))!=255)
                sum+=temp;                  /* default background intensity is 255 */
            subsampling[(p<<8)+r]=(unsigned char)(sum/20);          /* averaging */
             }
          }

       for (p=0;p<5;p++) {                  /* display processed images on the screen */
          for (q=0;q<256;q+=2) {
              write_dot((q>>1),line+p,subsampling[(p<<8)+q]);
              write_dot((q>>1)+150,line+p,subsampling[(p<<8)+q+1]);
              }
           }
       line+=5;
       if (line>=200) line=0;

       if (0==write(file_handle, subsampling, 1280)) {          /* write processed data to */
              printf("File write error!\n");                    /* storage disks */
              hardware_error=3;
              }

       buffers[i].full=0;/* clear semaphores and release this memory page */
       buffers[i].number=0;
        }
    if (page==(PAGES_NEEDED+1)) break;/* complete scanning process */
```

```
   }

stereo_close();          /* close range sensing hardware */
close(file_handle);              /* close the image file */
interrupt_restore();             /* restore the interrupt  channel */
free_buffers();          /* free allocated mamory */
set_graphics_mode(3);            /* return to the text mode */

 /* error reporting of scanning process */
if ((scanning_done_flag==1)&&(hardware_error==0))
   printf("Scanning process complete!\n");
else if (hardware_error==1)
   printf("Data overflow!\n");
else if (hardware_error==2)
   printf("Memory Buffer not enough!\n");
else if (hardware_error==3)
   printf("File write error!\n");
}
```

Vita

Yoshen Jason Hou was born on June 26, 1965 in Junhwa, Taiwan, Republic of China. He graduated from Junhwa High School, in Junhwa, Taiwan. In the fall of 1983, Yoshen attended National Chiao Tung University in Hsinchu, Taiwan. He graduated with a Bachelor of Science in Control Engineering in June 1987. Immediately following the graduation, he joined Chinese Navy for two years as an electronic petty officer, to fulfilled the civil obligation of the Republic of China. In 1989 to 1990, he worked as an electrical engineer in the Electro-optics & Peripherals Development Center, Industrial Technology Research Institute, in Hsinchu, Taiwan. Following the completion of a project, he attended Virginia Polytechnic Institute and State University and began work on his Masters degree in Electrical Engineering in the fall of 1990.

His personal interests include playing musical instruments, poem reading, and many recreational sports.