

**FOURTH OPTICAL INSPECTION OF
INTEGRATED CIRCUITS USING
IMAGE PROCESSING AND
MATHEMATICAL MORPHOLOGY**

A Report

By

EPHREM A. CHEMALY

Submitted to the Graduate College of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF ENGINEERING

Dec 1986

MAJOR SUBJECT: ELECTRICAL ENGINEERING

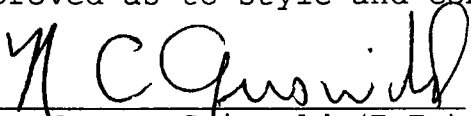
**FOURTH OPTICAL INSPECTION OF
INTEGRATED CIRCUITS USING
IMAGE PROCESSING AND
MATHEMATICAL MORPHOLOGY**

A Report

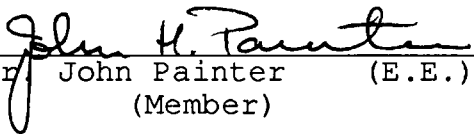
by

EPHREM A. CHEMALY

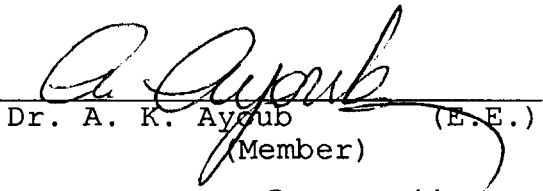
Approved as to style and content by:



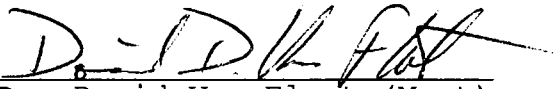
Dr. Norman Griswold (E.E.)
(Chairman of Committee)



Dr. John Painter (E.E.)
(Member)



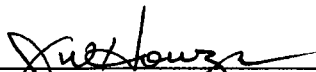
Dr. A. K. Ayoub (E.E.)
(Member)



Dr. David Van Fleet (Mgmt)
(Member)



Dr. Earl Bennet (Acct)
(Member)



Dr. Joe Howze (E.E.)
(Head of Department)



Dr. Carl Erdman
(Dean, College of Engineering)

JAN 15 RECD

Dec 1986

COPYRIGHT

This manuscript may not, in whole or in part, be copied, photocopied, reproduced, translated or reduced to any electronic medium or machine-readable form without prior consent, in writing, from Ephrem A. Chemaly.

© Copyright 1986 by Ephrem A. Chemaly.

ABSTRACT

Fourth Optical Inspection of Integrated Circuit Using Image Processing and Mathematical Morphology. (May 1986).

Ephrem A. Chemaly

Bachelor's of Science, Texas A&M University

Master's of Engineering, Texas A&M University

Chairman of Committee: Dr. Norman C. Griswold

As competition gets tighter in the industrial world, many manufacturers are switching from the classical labor intensive methods of production to more sophisticated techniques using automation. At the Microprocessor division of Motorola Inc. this approach was taken in a systematic manner. An automation group was founded with the ultimate goal of automating the testing procedures of their product.

This testing procedure consists of two major tasks. First an electrical test is performed on the microprocessors to insure that they meet the required electrical specifications. Second a visual inspection of the packages is undertaken to reject the parts with surface flows, such

as bubbles, blisters, chips, cracks, and voids.

The latter inspection will be of interest to us in this dissertation. The purpose of this work is to perform the visual inspection automatically. The method used is image processing. In particular Mathematical Morphology is viable because it is the study of shapes. As we will find in this report, defects are merely abnormal shapes of the packages.

DEDICATION

With the highest honor and respect, I dedicate this manuscript to my parents, Mr. and Mrs. Antoine Chemaly, for with their love and support I had the opportunity to achieve this higher degree of education.

ACKNOWLEDGMENT

Assistance and advice from many people have contributed to this work. The author is grateful to the whole automation team of Motorola in the Oak Hill facility for their support during this work, especially Mr. Fred Toewe for making this research project fit in a manufacturing environment.

TABLE OF CONTENTS

ABSTRACT	iii
DEDICATION	iv
ACKNOWLEDMMENT	v
TABLE OF CONTENTS	vi
CHAPTER 1. INTRODUCTION	1
CHAPTER 2. THE INSPECTION PROBLEM	13
CHAPTER 3. MATHEMATICAL MORPHOLOGY	22
CHAPTER 4. THE INSPECTION	49
CHAPTER 5. IMAGING HARDWARE	79
CHAPTER 6. MATERIAL MOVEMENT HANDLER ..	90
CHAPTER 7. CONCLUSION	110
APPENDIX A. THRESHOLD DATA COMPUTATION ..	114
APPENDIX B. CCD THEORY	119
REFERENCES.	133

CHAPTER 1

INTRODUCTION

This manuscript presents the results of a project done at Motorola Inc. in Austin, Texas. The task was to automate the visual inspection of integrated circuit packages using computer vision. This work was chartered by the automation department of the Microprocessor Product Group of Motorola.

1.1 ECONOMIC JUSTIFICATION FOR AUTOMATION

The recent widespread interest in automation in the U.S. originates from American industry's most fundamental problem: a staggering drop in productivity. From 1947 to 1965, U.S. productivity increased at an average rate of 3.4 percent a year. The growth rate decreased to 2.3 percent in the following decade, then dropped to below 1 percent in the late 1970's and down to -0.9 percent in 1980. Japan's productivity growth, the contrasting example most often cited in the literature, has been climbing at an average annual rate of about 7.3 percent [1].

Although there are many ways to influence manufacturing

productivity and product quality - regulatory, fiscal, and social - the emphasis in the following discussion is technological.

1.1.1 Production Function

In this paragraph, we focus on the economic analysis of the tradeoffs between the use of labor and automated equipment. The main interest here is the shift of isoquant and isocost curves over time. We chose the years 1980 and 1990 to illustrate this change.

Briefly, a production function is a mathematical statement of the way the quantity of output of a particular product depends on the use of specific inputs, or resources. For instance if the inputs are labor and automated equipment, we have:

$$Q = f(L, AE)$$

where Q is the maximum quantity that can be produced

L is the amount of labor

AE is the amount of automated equipment

This production function relationship can be graphed as a surface in three-dimensional space as shown in Figure 1.1.

In the long run it is possible to vary the amount of each input to obtain a certain output. Figure 1.1 shows the relationship between labor and automated equipment for three

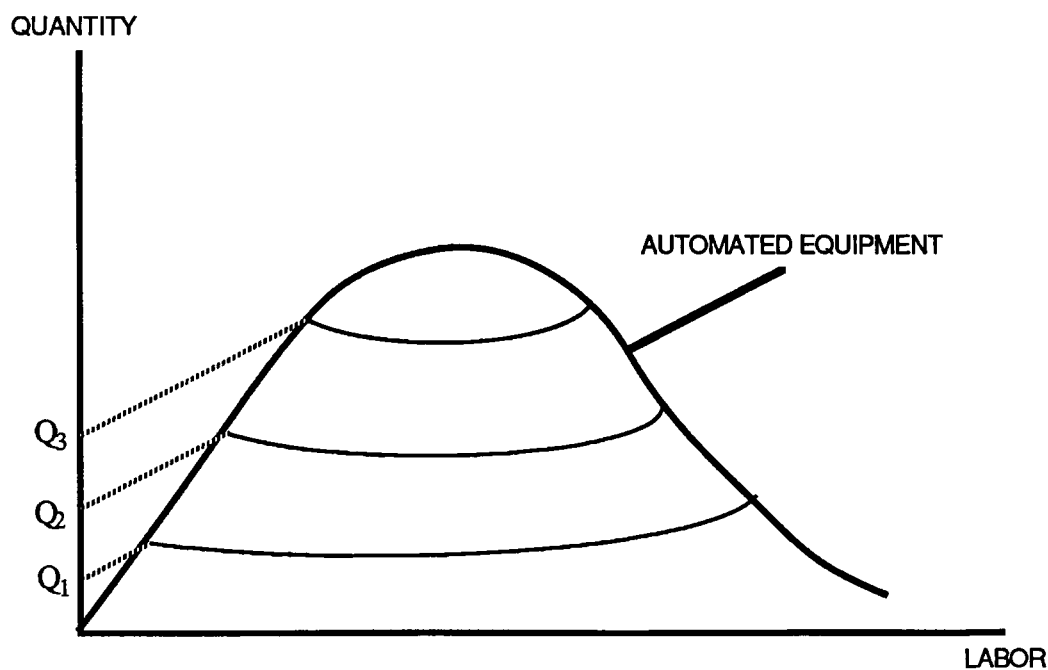


Figure 1.1. Production Function.

levels of output Q_1 , Q_2 , and Q_3 . These curves are called iso-quantity contours.

1.1.2 Isoquant and Isocost Curves.

How do the iso-quantity contours change with time? To illustrate this phenomenon we project one of the curves onto the x-y plane and call it the isoquant curve. Figure 1.2 shows such curves plotted for the years 1980 and 1990.

There are two main reasons for the changes in these curves during that decade. On one hand, the tremendous

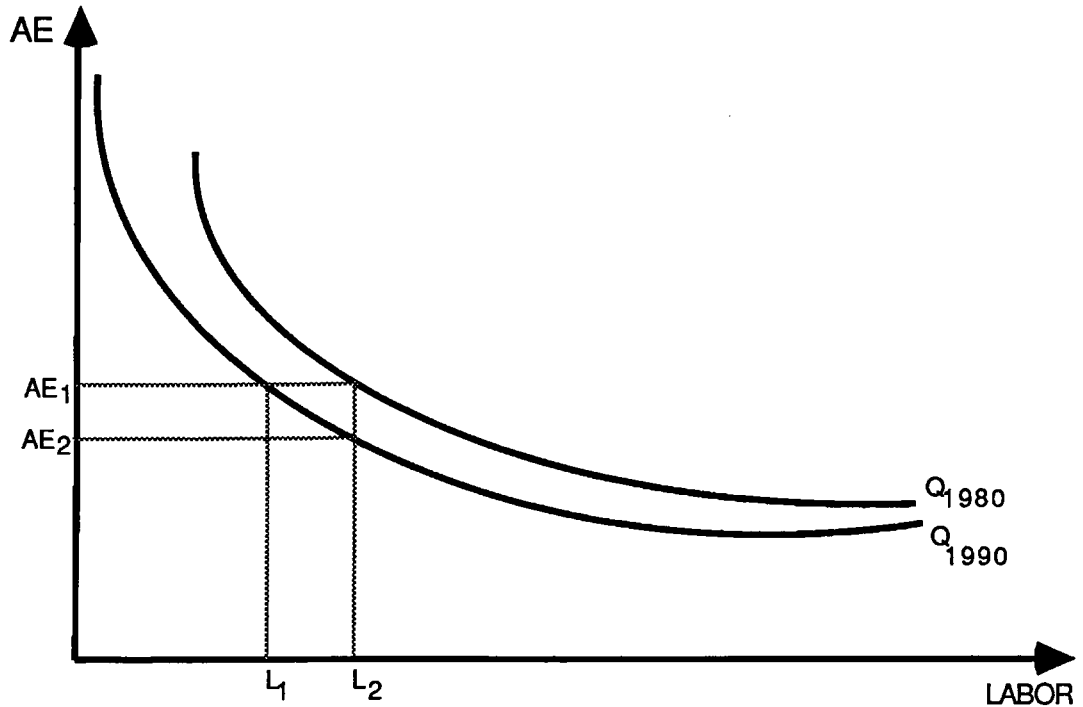


Figure 1.2. Isoquants Curves for 1980 and 1990.

advances taking place in the electronic and computer industry are making machines more capable of simulating human behavior. These machines can now perform tasks only humans were able to perform just a few years ago. Also the price of these machines is coming down at an astonishing rate. On the other hand, the increasing power of labor unions, and the rising demands of workers for higher wages and benefits, makes automation a viable alternative.

In order to make an economic decision on the optimum use of inputs, we need to consider the rate at which these

inputs can be exchanged within the firm's budget constraint. To aid our thinking in this regard, we present the concept of the isocost curves. Isocost curves show all combination of inputs that can be employed for a given dollar cost. In the case of the two inputs considered earlier, labor and automation, the equation for an isocost line is of the form,

$$C = P_L \cdot L + P_{AE} \cdot AE$$

where C is the firm total cost of inputs

P_L is the price of labor

L is the amount of labor

P_{AE} is the price of automation

AE is the amount of automated equipment

Two of those lines are plotted in Figure 1.3 for the years 1980 and 1990. The slope of these curves is given by,

$$\text{Slope} = - P_L / P_{AE}$$

As the price of labor increases and the price of automation decreases the slope of the isocost lines increases in absolute value.

1.1.3 Optimum Economic Combination of Labor and Capital Equipment.

To find the optimum combination of the two inputs we still need to define the marginal product (MP) of an input.

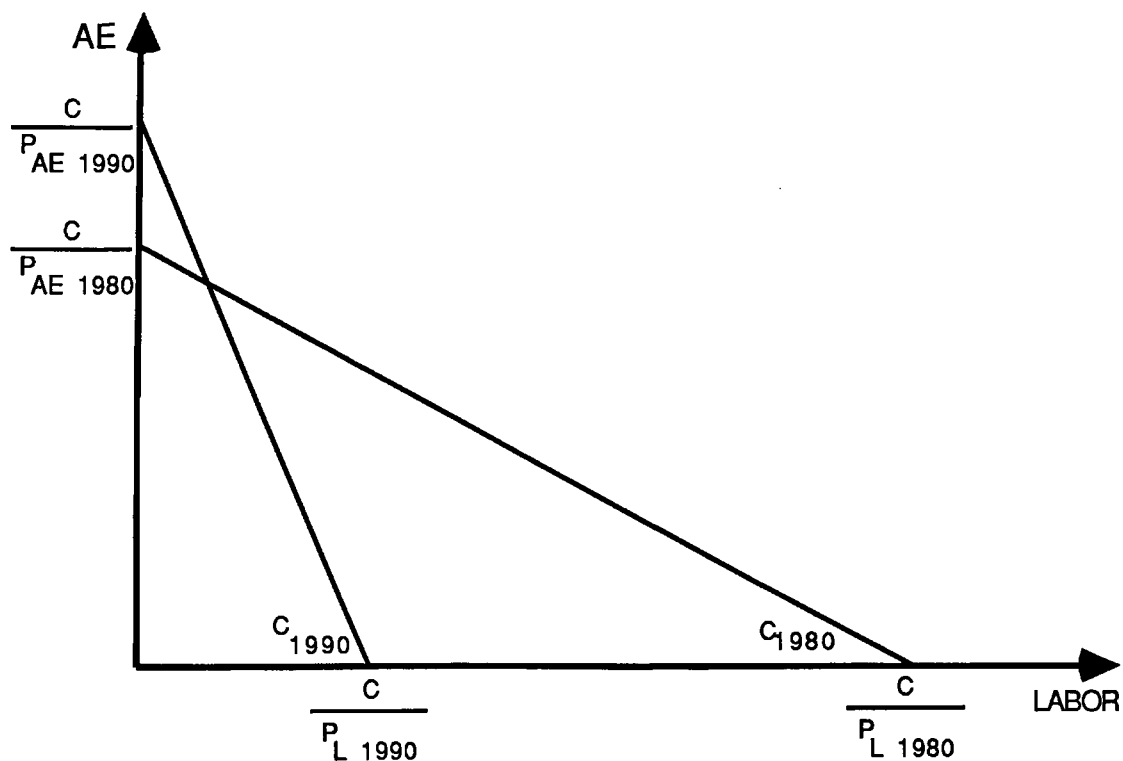


Figure 1.3. Isocost Curves for 1980 and 1990.

The marginal product is the rate of change of output (Q) resulting in the change of only one of the inputs holding everything else constant. In calculus terms, for an input a we have,

$$MP_a = dQ/da$$

To obtain the combination of inputs that will enable the firm to produce a given output at the lowest possible cost, the firm's managers must employ the two inputs in such a manner that the isocost line corresponding to the given level of expenditure is tangent to the highest isoquant

curve[2], in other words to have the following equality,

$$MP_L/P_L = MP_{AE}/P_{AE}$$

Figure 1.4 illustrates this concept for the years 1980 and 1990.

It is clear from the above discussion that the trend is to use less labor and more automation. Motorola is one

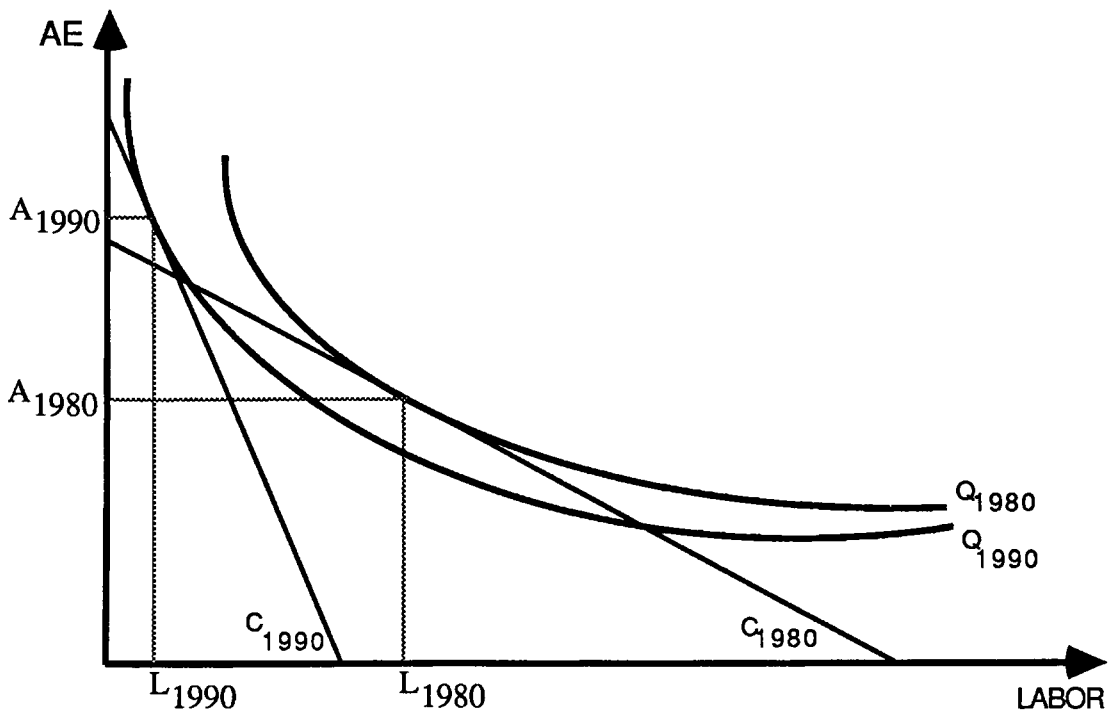


Figure 1.4. Optimal Use of Labor and Automation for 1980 and 1990.

of many companies which are automating their production process. Before defining the automation task, let us introduce the company.

1.2 THE COMPANY

Motorola is one of the world's leading manufacturers of electronic equipment and components. Motorola products include semiconductors, defense and aerospace electronics, automotive and industrial electronic equipment, and various forms of voice and data communication systems.

The Microprocessor Product Group (MPG) in Austin, Texas, is part of the Semiconductor Product Sector, headquartered in Phoenix, Arizona. MPG is a leading worldwide manufacturer of microprocessors. The innovative systems on silicon include the Motorola 68000 family and the most prominent 32 bit microprocessor, the "68020".

The production of integrated circuits entails several steps. The starting material for the fabrication of integrated circuits is silicon. Before silicon can be used for integrated circuits it must be refined to near perfect purity. Next, the silicon is "grown" into crystal ingots that are then sliced to form wafers.

The circuits are formed on the wafer using a photographic process in conjunction with chemical processes. After the circuits are finished, the dies (circuits) are separated then soldered to a metal plate (lead frame), this is called die bonding. The next step is to encapsulate these circuits in plastic or ceramic packages.

The good circuits, once packaged, are examined to insure that all of the connections have been made and no damage occurred during the packaging operation. A final test is made. The circuits that pass this final test are marked with a code number that identifies the function of the circuit now ready for shipment to the customer. This final test procedure for the microprocessor product group is performed in Austin, Texas at the Oak Hill facility.

1.3 AUTOMATION TASK

Of interest to us in this report is the automation of the final test procedure. First let us describe a typical product flow through this step of the production process. The parts are brought to the test floor from the warehouse in boxes containing many rails. A rail is usually a plastic tube that contains 7 to 15 packaged IC's. These rails are used to move the parts out of the boxes and into the different machines or workbenches used to perform different tasks on the parts. The major tasks are:

1. 25°C test: This is an electrical test performed at room temperature to assure the functionality of the product. The IC chips are loaded by operators into handlers that transport the IC to a test site. There, electrical contact with the circuit is made and a tester sends different input

signal and measures output parameters. If all the output signals meet the specified criteria the part is passed, otherwise it is failed. The operator then reloads the good parts into rails and puts them back in boxes. The rejects are also collected for further processing.

2. Marking: This step prints on each device the Motorola logo, a part number (which defines its function and sometimes some of its characteristics) and a date code. To mark the parts the operator loads them into a "marker", the parts are singulated, then marked using pad printing techniques. After the marking, the operator inspects the part for the quality of the printed characters. If, using his judgment, the mark is not legible, he will remove the part, erase the characters, and remark it. If the mark is clear the part is cured, using heat, and the mark is now permanent.

3. Pack and label: Each package is visually inspected for physical defects, then packed, boxed and labeled. The operator brings the product to his work bench. Takes the parts out of the rail, and inspects them for surface defects such as chips, cracks, surface contamination, bent leads etc. He then separates the acceptable parts from the rejects, reloads them into the rails, puts the rails into boxes, and labels the boxes.

4. QA sampling: Quality assurance sampling to verify

that the above tasks were performed properly before shipping to the customer.

To coordinate the above tasks personnel is needed for material movement, supervision, training, maintenance, etc. It is clear that the test procedure is performed in a labor intensive environment.

The first step in the automation process was to use robots to load and unload the product from the machines that perform the electrical test and the marking. Conveyor belts were used to bring the product to the robots. The second step was to perform the visual inspection using computer vision.

1.3.1 Automation Using Computer Vision

Computer vision is essentially the simulation of human vision on a computer. The principal motivation behind computer vision is increased flexibility and lower cost. The use of sensing technology to endow machines with a greater degree of "intelligence" in dealing with their environment is receiving increased attention. In the last few years a number of innovative systems have demonstrated the use of visual input and sophisticated computer processing to achieve a variety of manufacturing goals. Years ago reports of successful operation came mainly from

universities and publicly supported research laboratories. Now many of those early experiments are being converted to hardware and software operating on factory floors. The increasing interest in automatic visual inspection stems from the availability of low-cost processors, memory, and solid state imaging devices. These components have allowed implementation of image processing algorithms which were not cost effective on mini or mainframe computers. Most systems so far have been custom engineered for specific applications, but we are beginning to see the marketing of general purpose vision systems for certain classes of tasks.

CHAPTER 2

THE INSPECTION TASK

Developing an automated visual inspection system for a particular device requires careful design. First, there must be an objective, quantifiable visual test standards for the part. Currently general purpose inspection systems are not trainable by presenting them with a collection of good and defective parts and a few general words of guidance. Second, the inspection requirements are used to specify the image acquisition parameters, such as lighting, background, and the required image resolution, the most important of which is the size of the digital image sent to the processor. Third, an image analysis strategy must be designed, processing algorithms developed, and a classification scheme defined. Fourth, an appropriate parts handling mechanism is needed to present the parts to the vision system. Finally, the performance of the system must be measured by comparing its results with the independently and presumably accurate parts classification prepared from the existing manual inspection.

However the demonstration of a satisfactory method for

inspecting a part is not, by itself, a sufficient basis to use automatic inspection. For a production machine to be implemented, the inspection system must also meet cost, speed, reliability, and size constraints, as well as ease of integration with the existing production methods.

It is only when all of the above requirements have been considered that a successful implementation of an automated visual inspection system can be achieved. In this chapter we first describe the parts that need to be inspected. Second we define the inspection criteria. And finally we present an overview of the inspection task.

2.1 THE PACKAGE

The parts to be inspected are the 600 mil, plastic, dual inline package (DIP) integrated circuit shown in Figure 2.1. The package body is made of plastic using a plastic injection molding technique. The body is mounted on the metal lead frame. The leads that protrude from two of the sides are needed to make electrical contact between the circuit inside the package and the outside world.

During the molding procedure, and due to subsequent handling, some of these packages are damaged. Some of these damages are critical to the future operation of the circuit. For this, as well as for aesthetic reasons, a sample is

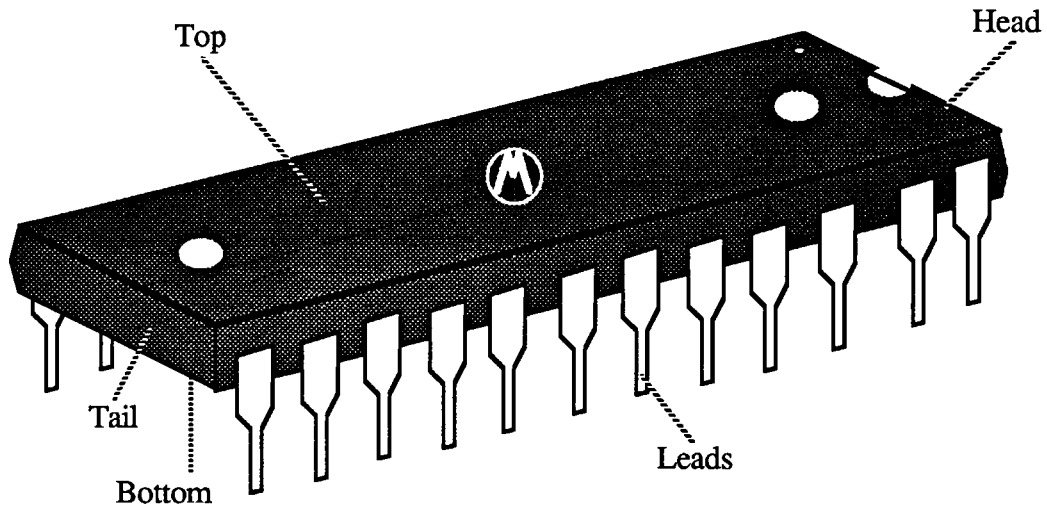


Figure 2.1. 600 Mil Dual In Line IC Package.

taken from each lot and inspected for visual defects. If any of the sample parts fail to meet the visual criteria outlined in the next section, the whole lot is inspected. The inspection entails looking at all six sides of the part to see the defects.

Figure 2.1 shows the nomenclature used to refer to all six sides of the package. The leads, also called pins, are mounted on two sides of the package. The picture of these two surfaces will be combined and called side #1. The top surface will be referred to as surface #2. The head, or surface #4, represents the "front" of the package while the tail, or surface #3, is the end of the part. The bottom surface, or surface #5, is also called the belly.

These packages come in different lengths, as shown in Figure 2.2. The four sizes shown in the figure have different lead counts on them. The smallest one has 24 leads (12 per side), the next largest one has 28 leads (14 per side), the next one has 40 leads (20 per side) and finally the largest one has 48 leads (24 per side). The mold marks are depressions on the top and bottom of the package used to eject the package from the mold after encapsulation. The pin one notch and the pin one dimple are also depressions on the top of the package used to

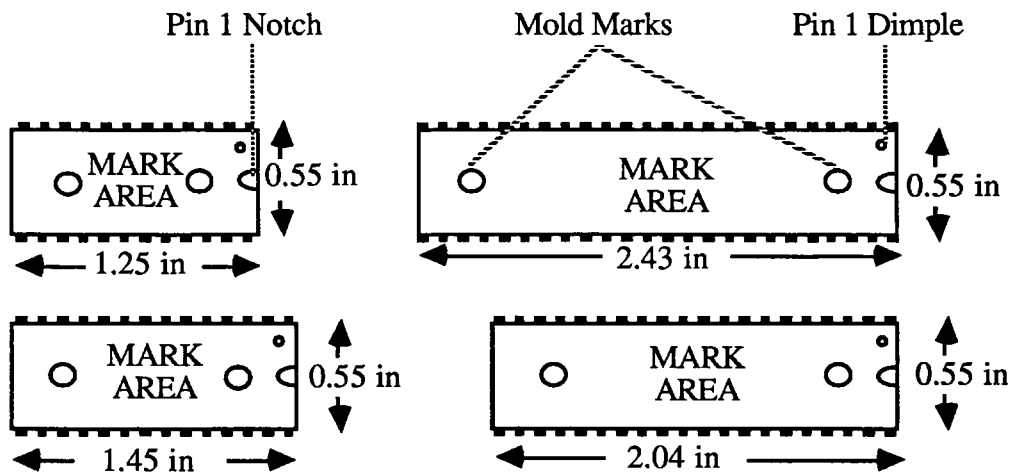


Figure 2.2 Drawing And Nomenclature of the DIP Package.

identify pin number 1 which is used to orient the parts properly. All these depressions have a glossy finish. The top and the bottom of the package have a mat finish, while

the other four sides can have a mat finish, a glossy finish, or a combination of the two. The area labeled MARK AREA is where the package is marked with a function code and a date code. This mark is made with a pad printing technique using white ink.

2.2 PLASTIC PACKAGE VISUAL CRITERIA

This paragraph presents the definitions and criteria for the visual inspection of the plastic IC packages.

2.2.1 Definitions

The definitions of the terms used to specify the criteria for the visual inspection are:

BLISTER: Blisters are circularly shaped, raised areas on the surface of the package. They most often occur on the top and bottom of the package and are the result of air being trapped within the package near the surface.

HOLE: Holes are openings on the usually flat surfaces of the package. An alternate term would be "voids". These are caused by air bubbles being trapped within the mold when the still fluid plastic is injected. Holes most often occur on the top and end surfaces of the package.

CHIP: A chip is the abnormal absence of package material

which results from an impact to or stress of the package. The ends of the package seem to be most prone to chip type defects.

CRACK: A crack is merely a chip in the making. It occurs for the same reasons, but will be found most often on the top of the package near the ends.

PIT: A pitted surface is characterized by a cluster of small depressions in the surface of the plastic. The depressions are too small to be rejectible defects individually, but taken in combination, they can be. This defect occurs most often on the top and bottom of the package.

BENT LEADS: Bent leads are those leads that are displaced from their original location.

MISSING LEADS: Missing leads are those leads that broke off the lead frame.

PIN ONE ORIENTATION: The pin one on all packages should have the same orientation while in the sleeve.

2.2.2 Criteria

The defects above (except for the last two) have to be larger than a certain size before a part is rejected. The dimension of the defects is confidential information to Motorola and cannot be disclosed. In this monograph we

refer to the size of the defects as l .

The operators usually use their own judgement during the inspection. A gauge is provided to inspect the marginal parts. Because the inspection is a simple task done repetitively, for prolonged periods of time, visual fatigue often limits the job performance. Image magnification and control of lighting are the only aids provided for the production workers.

2.3 A SYSTEMS APPROACH TO THE SOLUTION

The general problem of visual inspection is best described in terms of a system model. Figure 2.3 shows the basic components of the model and indicates the major information paths.

Overall control of the inspection procedure is performed by the process control system. It coordinates major timing functions, processes operator commands, and controls the system wide data base. Its task is to implement the normal inspection sequence and handle exceptions.

The sensing system provides the illumination source for the part, optics to control the illumination, the image optics, and the necessary support electronics to interface to the detector. This is a very important building block

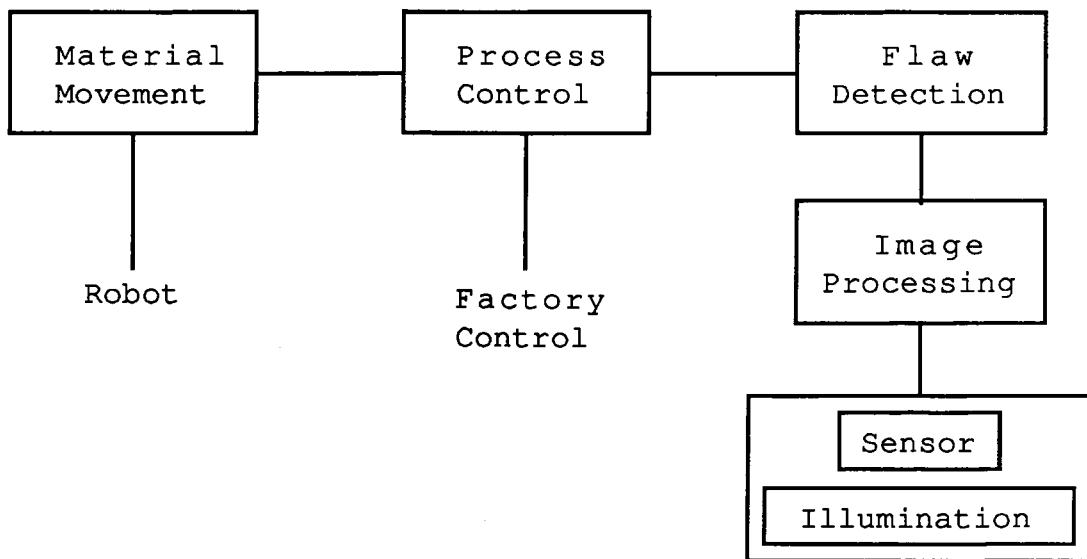


Figure 2.3. Inspection System.

for an industrial inspection system, because the image can be enhanced with the appropriate lighting at no cost in time.

The image processing system extracts pertinent information from the image data. This module in essence performs the task of isolating interesting data for further analysis. This is typically a specialized array processor that operates on the image data at very high speeds.

The flaw analysis system interprets the data identified by the image processing system and determines the part classification based on the flaw decision rule. This is usually a microprocessor based system that provides intelligence to the system.

The material movement mechanism transports the part and presents the different sides to the vision system for inspection. Reliable acquisition and accurate positioning of the part to be inspected can greatly reduce the complexity of other system processes. In image analysis, for example, if the part location is not well known, a large amount of time is needed to locate the part before the inspection can be performed. This mechanical system must operate reliably for extended periods of time.

In the next chapters the implementation of the above system for our inspection task will be presented. Chapter 3 introduces the reader to mathematical morphology - the theory behind the inspection algorithms. Chapter 4 presents the algorithms used to perform the visual inspection. Chapter 5 describes the sensor, image processing, flaw detection, and process control systems. Chapter 6 describes the material movement handler and its interface with the robot. Finally Chapter 7 concludes the manuscript with the results of the project.

CHAPTER 3

MATHEMATICAL MORPHOLOGY

Mathematical morphology was born in 1964 when George Matheron was asked to investigate the relationships between the geometry of porous media and their permeabilities, and when at the same time, Jean Serra was asked to quantify the petrography of iron ores, in order to predict their milling properties. This initial period (1964-1968) has resulted in a first body of theoretical notions, and also in the first prototype of the texture analyzer. It was also the time of the creation of the "Centre de Morphology Mathematique" on the campus of the Paris School of Mines at Fontainbleau (France). Stanley Sternberg, under the auspices of the Environmental Research Institute of Michigan, designed the Cyto computer which performed true grayscale morphology in real time.

The family tree of mathematical morphology essentially comprises the two branches of integral geometry and geometrical probabilities [3],[4]. The notion of a geometrical structure is not purely objective. Mathematical

morphology quantifies this intuition by introducing the concept of structuring elements. Chosen by the morphologist, they interact with the object under study, modifying its shape and reducing it to a sort of caricature which is more expressive than the actual initial phenomenon. The power of the approach, but also its difficulty, lies in this structural sorting.

In this chapter we start by introducing mathematical morphology in Euclidean space. Most of the properties of morphology will be given there. Then we discuss mathematical morphology in binary digital space. Finally we discuss the subject in 3-dimensional space.

3.1 EUCLIDEAN MORPHOLOGY

Let E^n denote the set of all points in Euclidean n -space and let $p=(x_1, x_2, \dots, x_n)$ represent a point of E^n . With each set A belonging to E^n is associated a binary image \mathbf{A} , an n -dimensional composition in black and white, where (x_1, x_2, \dots, x_n) is black in \mathbf{A} if and only if $(x_1, x_2, \dots, x_n) \in A$ otherwise (x_1, x_2, \dots, x_n) is white. In the sequel we will not distinguish between the image \mathbf{A} and the set A , they will both be denoted by A . A binary image in E^2 is a

silhouette, a composition in black and white which partitions the plane into regions of foreground (black) and background (white). A binary image in E^3 is a partitioning of a volume into points belonging to the surface and interior of objects (black) and points belonging to the space surrounding those objects (white). Binary images in higher dimensional spaces can be considered also.

3.1.1 Inclusion and Equality of Binary Images.

Let A and B be any two binary images. If every black point of A is black in B, then A is called a Subimage of B, or A is included (contained) in B. Symbolically, this relation is denoted

$$\text{(Inclusion)} \quad A \subset B \quad \Leftrightarrow \quad \forall \theta \in A \Rightarrow \theta \in B$$

The following are some properties of binary image (set) containment for any binary images A, B, and C:

$$\text{(Reflexivity)} \quad A \subset A$$

$$\text{(Transitivity)} \quad (A \subset B) \text{ and } (B \subset C) \Rightarrow A \subset C$$

$$\text{(Equality)} \quad A = B \quad \Leftrightarrow \quad A \subset B \text{ and } B \subset A$$

A binary image is universal if it contains every binary image under discussion. The universal image which is all black is symbolized by Ω .

The all white binary image, or null image or blank image, does not contain any other binary images and is symbolized by ϕ .

3.1.2 The Power Image

Given any binary image A , we know that the null image, ϕ , is a subimage of A and that A itself is a subimage of A . Furthermore, for any point $p \in A$, the binary image $P = \{p\}$ having a single black point is a subimage of A . For any binary image A , the collection or set of all subimages of A is called the "power binary image" or simply power image of A . The power image of A is denoted 2^A so that

$$2^A = \{X \mid X \subset A\}$$

3.1.3 Logical Operations on Binary Images

This section introduces some basic operations on binary images. Using these operations, we can construct new binary images by combining the black points of given images following certain logical rules.

The intersection of two binary images A and B , written

$A \cap B$, is the binary image which is black at all points p which are black in both A and B . Symbolically

$$\text{(Intersection)} \quad A \cap B = \{ p \mid p \in A \text{ and } p \in B \}$$

From the definition of intersection, it follows

$$A \cap B = B \cap A$$

$$A \cap A = A$$

$$A \cap \phi = \phi$$

Since $A \cap B$ is a binary image, we can consider its intersection with another binary image C ,

$$(A \cap B) \cap C = \{ p \mid p \in (A \cap B) \text{ and } p \in C \}$$

We can show that

$$\text{(Associativity)} \quad (A \cap B) \cap C = A \cap (B \cap C) = A \cap B \cap C$$

For an indexed set of binary images $A = \{A_1, A_2, \dots, A_n\}$ we write

$$A_1 \cap A_2 \cap \dots \cap A_n = \bigcap_{i=1, n} A_i$$

Similarly, the union of two binary images A and B , written $A \cup B$, is the binary image which is black at all points p which are black in either A or B . Symbolically

$$\text{(Union)} \quad A \cup B = \{ p \mid p \in A \text{ or } p \in B \}$$

From this definition we get the following properties:

$$A \cup B = B \cup A$$

$$A \cup A = A$$

$$A \cup \phi = A$$

Since $A \cup B$ is a binary image, we can consider its union with another binary image C ,

$$\text{(Union)} \quad (A \cup B) \cup C = \{ p \mid p \in (A \cup B) \text{ or } p \in C \}$$

we can show that

$$\text{(Associativity)} \quad (A \cup B) \cup C = A \cup (B \cup C) = A \cup B \cup C$$

For an indexed set of binary images $A = \{ A_1, A_2, \dots, A_n \}$ we write

$$A_1 \cup A_2 \cup \dots \cup A_n = \bigcup_{i=1, n} A_i$$

Let Ω be the universal binary image (all black) and A a binary image. The complement of A denoted A' is the binary image which interchanges the colors black and white in A . Symbolically,

$$\text{(Complement)} \quad A' = \{ p \mid p \in \Omega \text{ and } p \notin A \}$$

3.1.4 Geometric Operations of Dilation and Erosion

Let $p = (x_1, x_2, \dots, x_n)$ and $q = (y_1, y_2, \dots, y_n)$ denote two elements of E^n . The sum $p+q$ is the point

$$\text{(Sum)} \quad p + q = (x_1 + y_1, x_2 + y_2, \dots, x_n + y_n)$$

while the difference $p - q$ is the point

$$\text{(Difference)} \quad p - q = (x_1 - y_1, x_2 - y_2, \dots, x_n - y_n)$$

both in E^n . If s is a real number, then

$$\text{(Scaling)} \quad sp = (sx_1, sx_2, \dots, sx_n)$$

also belongs to E^n and is referred to as the scaling by s .

If A is a subimage of E^n and p a point in E^n , then the translation of A by p is a subimage of E^n given by

$$\text{(Translation)} \quad A_p = \{ a+p \mid a \in A \}$$

Translation of a binary image A by a point p shifts the origin of A to p .

We are ready now to define the two basic morphological operations of dilation and erosion. If $A_{b_1}, A_{b_2}, \dots, A_{b_n}$ are translations of the binary image A by the black points of

the binary image $B = \{b_1, b_2, \dots, b_n\}$ then the union of the translations of A by the black points of B is called the dilation of A by B and is symbolized by

$$\text{(Dilation)} \quad A \oplus B = A_{b_1} \cup A_{b_2} \cup \dots \cup A_{b_n} = \bigcup_{b_i \in B} A_{b_i}$$

Dilation is also called Minkowski addition [3].

Dilation can also be expressed in terms of the sums of points in binary images A and B ,

$$A \oplus B = \left\{ p \mid \exists_{b_i \in B} p \in A_{b_i} \right\} = \left\{ p \mid \exists_{b_i \in B} \exists_{\theta_j \in A} p = \theta_j + b_i \right\}$$

The operational dual of dilation is erosion. The erosion of a binary image A by a binary image B is black at a point p if and only if the translation of B to p is entirely contained in A . Symbolically,

$$\text{(Erosion)} \quad A \ominus B = \{ p \mid B_p \subset A \}$$

Erosion is also called Minkowski subtraction [3].

Dilation can also be expressed in terms of the differences of points in binary images A and B ,

$$A \ominus B = \left\{ p \mid \forall_{b_i \in B} b_i + p \in A \right\} = \left\{ p \mid \forall_{b_i \in B} \exists_{\theta_j \in A} p = \theta_j - b_i \right\}$$

The dual nature of dilation and erosion is geometric

rather than logical and involves a geometric negation as well as a logical negation. The geometric negation of a binary image is called its reflection. The reflection of a binary image B is that binary image \check{B} which is symmetric to B about the origin, that is

$$\text{(Reflection)} \quad \check{B} = \{ -p \mid p \in B \}$$

The geometric duality is expressed as

$$(A \oplus B)' = A' \ominus \check{B}$$

and

$$(A \ominus B)' = A' \oplus \check{B}$$

so that dilation and erosion can be expressed as

$$A \oplus B = \{ p \mid \check{B}_p \cap A \neq \emptyset \}$$

and

$$A \ominus B = \bigcap_{x \in \check{B}} A_x$$

3.1.5 Properties of the Geometric Operations

In the preceding section, we introduced the binary operations on binary images, dilation and erosion, and a unary operation, reflection. In this section, we will examine some properties which arise in the geometric

operations.

First let us present some basic identities for the geometric operation of dilation. For any binary images A , B , and $C \in E^n$ we have:

$$\text{(Commutativity)} \quad A \oplus B = B \oplus A$$

$$\text{(Associativity)} \quad (A \oplus B) \oplus C = A \oplus (B \oplus C) = A \oplus B \oplus C$$

There exists a distinguished binary image $I \in E^n$ such that for any $A \in E^n$:

$$\text{(Identity image)} \quad A \oplus I = I \oplus A = A$$

The identity image is black only at the origin of E^n . Dilation distributes itself over union

$$A \oplus (B \cup C) = (A \oplus B) \cup (A \oplus C)$$

but not over intersection.

So dilation is commutative, associative, distributive over union and possesses an identity element. Erosion is not commutative. Since the erosion of A by B indicates the containment of B in A . But both $A \subset B$ and $B \subset A$ implies $A = B$. Hence, erosion does not commute except in the special instance when $A = B$. But we have the following property,

$$(A \ominus B) \ominus C = (A \ominus C) \ominus B$$

so that the order in which the erosion of A by B and C is

applied is immaterial. More significant is the statement that erosion by a binary image which is itself the dilation of two binary images is equal to the sequential erosions by the factors of the eroding image. Symbolically,

$$A \ominus (B \oplus C) = (A \ominus B) \ominus C$$

With respect to the distribution of erosion with the logical operations, we have:

$$A \ominus (B \cup C) = (A \ominus B) \cap (A \ominus C)$$

and

$$(B \cap C) \ominus A = (B \ominus A) \cap (C \ominus A)$$

Finally, there exist a right identity element I for erosion,

$$A \ominus I = A$$

but not a left identity.

3.1.6 Geometric Operations of Opening and Closing

We have from the preceding paragraphs definitions of the geometric operations on binary images which involves translating or sliding a reference binary image B called a structuring element or its reflection over a binary image A , the binary image under investigation or active binary image, for which any black point of the reflected binary image \check{B} overlaps any black point of A (dilation) or noting the

positions of the origin of B for which the entire black region of B fits inside the black region of A (erosion). From these geometric interpretations of erosion and dilation, we draw definitions for two new binary operations on binary images which are expressible in terms of erosion and dilation.

First, the opening of a binary image A by a binary image B is the union of all translations of B which are included in A. Symbolically denoting the opening of A by B as A_B , we can write

$$A_B = \bigcup_{p \in (A \ominus B)} B_p$$

since $p \in (A \ominus B)$ if and only if $B_p \subset A$. Now the union of all translations of a binary image B by the black points of a binary image $(A \ominus B)$ is by definition the dilation of B by $(A \ominus B)$. Since dilation is commutative, we can then write

$$\text{(Opening)} \quad A_B = (A \ominus B) \oplus B$$

Closing is the dual operation of opening and is most easily expressed in terms of the complement of A and the reflection of B. The complement of the closing of A by B is the union of all translations of \check{B} which are included in A'. Denoting the closing of A by B symbolically as A^B , we

can write

$$(A^B)' = \cup_{p \in (A' \ominus \check{B})} \check{B}_p$$

since $p \in (A' \ominus \check{B})$ if and only if $\check{B}_p \subset A'$. Now the union of all translations of a binary image \check{B} by $(A' \ominus \check{B})$ is the dilation of \check{B} by $(A' \ominus \check{B})$. Since dilation is commutative, we can then rewrite the above equation as

$$(A^B)' = (A' \ominus \check{B}) \oplus \check{B}$$

or simply

$$\text{(Closing)} \quad A^B = (A \oplus B) \ominus B$$

The duality of opening and closing is most easily expressed by the pairs of identities

$$(A_B)' = (A')\check{B}$$

and

$$(A^B)' = (A')\check{\check{B}}$$

In algebra, an application is said to be an opening if it is anti-extensive, increasing, and idempotent. The operation of opening in the image algebra satisfies these three axioms. First, X_B is anti-extensive, i.e.,

$$\text{(Anti-extensive)} \quad X_B \subset X$$

since every point $x \in X_B$ belongs to at least one translation of B included in X . Second, the opening is increasing, i.e., if X is included in Y then X_B is included in Y_B ,

$$\text{(Increasing)} \quad X \subset Y \Rightarrow X_B \subset Y_B$$

and finally, we can show idempotence, i.e.,

$$\text{(Idempotent)} \quad (X_B)_B = X_B$$

By duality, the closing X^B is extensive, increasing and idempotent.

$$\text{(Extensive)} \quad X \subset X^B$$

$$\text{(Increasing)} \quad X \subset Y \Rightarrow X^B \subset Y^B$$

$$\text{(Idempotent)} \quad (X^B)^B = X^B$$

3.1.7 Compositions of Openings and Closings

Dilation was shown to be increasing and extensive. Closing adds the property of idempotence. When we dilate a set more than once, we do not get the same set, but a larger dilated set. To analyze the effects of successive openings and closings, we must first characterize the families of open and closed sets with respect to a given reference B .

1. The family of binary images open with respect to B is exactly the family of binary images $C \oplus B$ where C is any binary image in E^n .

Utilizing this characterization of opening leads to a further property of ordered set closing and openings.

2. If C is open with respect to B , then the openings X_B and X_C and the closings X^B and X^C of any binary image $X \in E^n$ satisfies the following inclusion:

$$X_C \subset X_B \subset X \subset X^B \subset X^C$$

3. If C is open with respect to B , then for every $X \in E^n$ the following is true:

$$(X_B)_C = X_C$$

if B is open with respect to C , then:

$$(X_B)_C = X_B$$

3.1.8 Convex Binary Images

To this point, we have developed the concept of unary and binary operations on the set of all binary images in E^n . We now introduce the set of convex binary images in E^n and

develop some additional identities for this important case.

A binary image A in E^n is convex if it contains the whole of every line segment joining any two of its points. If a_1 and a_2 are two points in A , then points x belong to the line segment joining them have the form

$$x = sa_1 + (1-s)a_2 \quad (3.1)$$

where s is a scalar in the closed interval $0 \leq s \leq 1$. If there exists an s in the closed interval $[0,1]$ such that the equation above holds true, then we say that x is a convex combination of the points a_1 and a_2 . Thus a binary image A is convex if and only if for every point $x \in A$ there also exists points a_1 and $a_2 \in A$ such that x can be expressed as a convex combination of a_1 and a_2 .

This definition of a convex set is easy to understand and visualize. But because it has no counterpart in the digital space where we replace the continuum \mathbf{R}^n by a lattice \mathbf{Z}^n a second definition of convex binary image can be extended to digital space.

A binary image X is convex if and only if it is the intersection of all half-spaces that contain it. Recall that a half-space Π of E^n are points x which satisfy the relationship

$$c_0 + c_1 x_1 + \dots + c_n x_n \leq 0$$

Denoting the set of all convex binary images as \mathbf{C} , we can formalize these two definitions of convexity as

$$X \in \mathbf{C} \Leftrightarrow x_1, x_2 \in X \Rightarrow [x_1, x_2] \in X$$

where $[x_1, x_2] = \{ x \mid x \text{ is a convex combination of } x_1 \text{ and } x_2 \}$

and

$$X \in \mathbf{C} \Leftrightarrow X = \bigcap_i \Pi_i ; \Pi_i \supset X$$

where the Π_i are the halfspaces of E^n .

3.1.9 Convex Hull

If A is a binary image in E^n , then the convex hull of A is that binary image in E^n whose black points are those that can be expressed as convex combinations of the black points of A . A convex combination of the m points a_1, a_2, \dots, a_m of a binary image A is a point p given by

$$p = \sum_{i=1, m} s_i \theta_i \quad (3.2)$$

where the scalars s_i are real numbers in the range $0 \leq s_i \leq 1$

and $\sum_{i=1, m} s_i = 1$. The expression of equation 3.1 is a

special case of Eq 3.2 for $m=2$. Thus the convex hull of a binary image is black at a point p if and only if there exists black points $a_1, a_2, \dots, a_m \in A$ for some $m > 0$ and scalars s_1, s_2, \dots, s_m in the interval $[0, 1]$ whose sum is unity such that p is expressible in the form of Equation 3.2 above. The convex hull of a binary image A will be denoted $C(A)$. Symbolically,

$$C(A) = \{ p \mid p = \sum_{i=1, m} s_i a_i, 0 \leq s_i \leq 1, \sum_{i=1, m} s_i = 1, m > 0, a_i \in A \}$$

We can also say that the convex hull is the intersection of all the half-spaces that contain A . For both definitions, the convex hull is the smallest convex set that contains A , and is identical to A if A is convex.

If A is a binary image and B is a bounded binary image, then

$$A^B \subset C(A)$$

If A is convex, then it is identical to its closure:

$$A \in \mathbf{C}, B \text{ bounded} \Rightarrow A^B = A$$

3.1.10 Pseudo-Convex Hull

Now we have defined the convex hull of a binary image A as the intersection of all the binary halfspaces of E^n that

contain A , and more precisely as the intersection of all the supporting halfspaces of A in E^n . Denote by H the halfspace of E^n which satisfies the equation

$$\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n \leq 0$$

that is, H is a halfspace whose boundary with H' is the hyperplasia

$$\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n = 0$$

Now if $\{H_i\}$ is a finite set of such halfspaces containing n elements, then the pseudo-convex hull of a binary image A is

$$C_p(A) = \bigcap_{i=1, n} A \oplus H_i$$

A regular n -hull of a binary image A in E^2 can be expressed as the intersection of the $A \oplus H_i$'s, where the H_i 's are translations of halfspaces defined from hyperplasias drawn through the origin and a vertex of a regular n -gon centered at the origin.

3.2 DIGITAL MORPHOLOGY

The implementation of our image processing methods is digital. This means that the images we are studying are represented by a finite number of sample values regularly

displayed over the space. They may take the form of vertices or points of a grid, or they may be contiguous elements which cover the space as a mosaic of picture elements (pixels). The transition from Euclidean space to Digital space must be approached with caution. It is not enough merely to replace Euclidean figures by pixels. We also have to replace the Euclidean translations, rotations, convex hull, etc., by analogous transformations on grids of points.

3.2.1 Modules

Let \mathbf{Z}^1 be the set of all integers, positive, negative, and zero, and \mathbf{Z}^2 the set of all ordered pairs of elements of \mathbf{Z}^1 :

$$x \in \mathbf{Z}^2 \Leftrightarrow x = (x_1, x_2); \quad x_1, x_2 \in \mathbf{Z}^1$$

For every $x = (x_1, x_2)$ and $y = (y_1, y_2)$ belonging to \mathbf{Z}^2 , we define addition as

$$x+y = (x_1 + y_1, x_2 + y_2)$$

Scalar multiplication is defined by

$$\lambda x = \lambda(x_1, x_2) = (\lambda x_1, \lambda x_2); \quad \lambda \in \mathbf{Z}^1, x \in \mathbf{Z}^2$$

\mathbf{Z}^2 is called a module. \mathbf{Z}^2 can be generated from any basis

made up of two elements of \mathbf{z}^2 . The module \mathbf{z}^n is the set of all n-tuples of integers satisfying the axioms above.

We now examine the basic operations on modules. The sets A, B, X, etc. are digital binary images.

3.2.2 Digital Binary Images

Let x denote an n-tuple (x_1, x_2, \dots, x_n) whose elements x_i belong to \mathbf{z}^1 . Denote by \mathbf{z}^n the Cartesian product $\mathbf{z} \times \mathbf{z} \times \dots \times \mathbf{z}$ of n terms containing all n-tuples. Consider a set of n basis vectors $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_n$ of n-dimensional space. To each n-tuple x in \mathbf{z}^n corresponds a vector (point) in n-dimensional Euclidean space E^n given by

$$\vec{v} = x_1 \vec{v}_1 + x_2 \vec{v}_2 + \dots + x_n \vec{v}_n$$

Let A denote a set of n-tuples in \mathbf{z}^n . With each set A of \mathbf{z}^n is associated a digital binary image in n-space which is black at a point (vector) \vec{v} if and only if

$$\vec{v} = x_1 \vec{v}_1 + x_2 \vec{v}_2 + \dots + x_n \vec{v}_n \quad \text{for } (x_1, x_2, \dots, x_n) \in A$$

3.2.3 Elementary Operations (defined on \mathbf{z}^2)

To perform the computations on a digital computer, we

use the following functions.

$$\text{Complementation} \quad X \in X' \Leftrightarrow x \in Z^2, x \notin X$$

$$\text{Inclusion} \quad X \subset Y \Leftrightarrow \forall x \in X \Rightarrow x \in Y$$

$$\text{Intersection} \quad X \cap Y \Leftrightarrow z \in X \cap Y \Rightarrow (z \in X \text{ and } z \in Y)$$

$$\text{Union} \quad X \cup Y \Leftrightarrow z \in X \cup Y \Rightarrow (z \in X \text{ or } z \in Y)$$

$$\text{Translation} \quad X_b = \bigcup_{x \in X} \{x + b\}, \quad b \in Z^2$$

$$\text{Reflection} \quad w \in \check{B} \Leftrightarrow -w \in B$$

$$\text{Erosion} \quad X \ominus B = \bigcap_{w \in \check{B}} X_w = \{p \mid B_p \subset X\}$$

$$\text{Dilation} \quad X \oplus B = \bigcup_{w \in B} X_w = \{p \mid \check{B}_p \cap X \neq \emptyset\}$$

3.3 GRAYSCALE MORPHOLOGY

Let $y = f(x)$ denote the grayvalue of a given point $x = (x_1, x_2, \dots, x_n)$ in n -space, where the x_i are the projected lengths of x on the basis vector $\vec{v}_i, i=1, 2, \dots, n$. The given point $[x, f(x)]$ can be represented in an augmented space for which the set of basis vectors \vec{v}_i is extended by \vec{v}_{n+1} . The

given point is represented by an $(n+1)$ -tuple $x^* = (x_1, x_2, \dots, x_n, x_{n+1})$ where $x_{n+1} = y$, the grayvalue of the point x in n -space.

If A is a set of points in n -space and if $f(x)$ is a function which assigns a grayvalue y to each point x of A , then A is a grayvalued image having a binary image representation in $(n+1)$ -space, that is, the grayvalued image A in n -space corresponds to a binary image A^* in $(n+1)$ -space, where $x^* = (x_1, x_2, \dots, x_n, x_{n+1})$ is an element of A^* if and only if $x = (x_1, x_2, \dots, x_n)$ is an element of A and the grayvalue $f(x) = x_{n+1}$. We say that x^* has the value 1 if $p^* \in A^*$ and zero otherwise in keeping with binary image convention.

Thus a grayvalued image in 2-space can be denoted by a binary image in 3-space. The binary image in 3-space consists of black "sheets" not necessarily connected, representing the single valued function of grayvalue. In this form, the binary image is referred to as a grayvalued surface.

We could apply the functions of image algebra directly to grayvalued surfaces, but to do so would be analogous to restricting ourselves to line drawings in 2-space. Since we

prefer to deal with foreground-background representation in 2-space where foreground, or black region, has solid form, we also prefer to deal with the binary image representation of a grayvalued image in solid form. For this reason, we introduce the concept of an umbra.

If A is a grayvalued image in n -space and A^* is its grayvalued surface in $(n+1)$ -space, then the umbra of A , denoted $U(A)$, is a binary image in $(n+1)$ -space where

$$U(A) = A^* \oplus \check{V}_{n+1}$$

where the binary image \check{V}_{n+1} is the reflection of V_{n+1} , the binary image representation of the positive \vec{V}_{n+1} axis. Thus a point $q=(x_1, x_2, \dots, x_n, x_{n+1})$ is black in the umbra representation of a grayvalued image if and only if the grayvalue of the point $x=(x_1, x_2, \dots, x_n)$ in the corresponding grayvalued image is less than or equal to x_{n+1} . For $n=2$ we have

$$p=(x,y,z) \in U(A) \quad \Leftrightarrow \quad z \leq f(x,y)$$

The significance of the umbrae for image processing is that they remain umbrae under the usual morphological transformations of union and intersection, dilation and erosion. The equivalence of grayscale images and umbrae

allows us to speak morphologically of dilating or eroding a grayscale image by three dimensional structuring elements, because the dilation or erosion of an umbra by any set of 3-space is always an umbra. The operations of opening and closing a grayscale image can be represented by the sliding of 3-dimensional structuring elements over gray level topologies describing image density or brightness.

Morphological transformations are always expressible in terms of the operation of intersection (or union), complementation, and translation. Intersections and unions of umbrae are umbrae. The height of the umbra union is the maximum of the individual umbrae. Symbolically, if $a(x)$ and $b(x)$ represent two gray scale functions then

$$U[a] \cup U[b] = U[\max(a,b)]$$

The points of the grayscale image union are then the maxima of the individual grayscale images. Similarly, the height of the umbra intersection is the minimum of the individual umbrae. Symbolically,

$$U[a] \cap U[b] = U[\min(a,b)]$$

The points of the grayscale image intersection are then the minima of the individual grayscale images.

The complement of an umbra is an umbra where the scene of up and down is reversed. Thus, in the grayscale morphology we simply replace the function $f(x)$ by $-f(x)$.

Finally, of course, umbrae remain umbrae under translation.

The notion of the inclusion of an umbra is also treated by the relationship of the magnitude of the grayscale function:

$$U[a] \subset U[b] \Leftrightarrow a(x,y) \leq b(x,y)$$

Dilating or eroding an umbrae by a structuring element yields an umbrae. This result comes directly from definitions for dilation and erosion. The dilation of $U[a]$ by B is the union of translations of $U[a]$ by the points of B . At a given location (x,y) the gray levels $d(x,y)$ of the dilation are determined by the maxima of the translated umbrae $U[a]$. Although $U[a]$ is translated by all the points of B , only those points on the surface of B can create maxima. The dilation of a grayscale image is computed then, as the maximum of the sum of gray levels of $b(x,y)$, the function describing the surface of B , with each of the points of $a(x,y)$,

$$d(x,y) = \max_{i,j} [a(x-i,y-j) + b(i,j)]$$

The erosion of $U[a]$ by B is likewise determined as a difference of the gray levels of $b(x,y)$ and the points of $a(x,y)$, the gray levels of the eroded image $e(x,y)$ determined by the minimum over all the points of $b(x,y)$,

$$e(x,y) = \min_{i,j}[a(x-i,y-j) - b(i,j)]$$

The expression for grayscale dilation and erosion bear a marked similarity to the convolution integral frequently encountered in image processing, with sums and differences replacing multiplication and minimum and maximum replacing summation. Unlike the linear transformations of grayscale images, dilation and erosions are characterized by their non-invertibility. Grayscale dilations and erosions, like their binary counterparts, remove grayscale information. The transformations are referred to as increasing in that they preserve containment relationships, rather than linear, as they are not invertible. Image processing through iterative morphological transformations is a process of selective information removal where irrelevant image content is irrecoverably destroyed, enhancing the contrast of the essential image features.

CHAPTER 4

THE INSPECTION

The reader should now be familiar with most of the theory involved in performing the inspection of the integrated circuit chips. This chapter describes the method used to carry out this task. Due to the speed at which the parts go through the production line, it was estimated that all six sides of the package need to be inspected in less than two seconds. This is a fairly high speed considering that a total of one and a quarter million bytes of data need to be analyzed for every part. To overcome this drawback lighting techniques are used to enhance the image.

In this chapter, for each side of the part, we first describe the physical characteristics of that surface. Second, we present the lighting scheme designed to enhance the image and simplify the inspection. Third, we present the learning algorithm needed to extract the characteristics of the lot being inspected from a few samples, and finally we present the inspection algorithm per se.

Before detailing the inspection technique, let us formally define surface texture which is used extensively in

this chapter. Surface texture is the repetitive or random deviations from the nominal surface which forms the three-dimensional topography of the surface. The roughness of a surface consists of the finer irregularities of the texture. The profile of a surface is the contour of that surface in a plane perpendicular to that surface.

4.1 INSPECTION OF THE LEADS

The leads are inspected to insure that none are bent or missing. This is subsequently referred to as inspection #1. With special lights, the bottom view provides a good image for a fast and accurate inspection. The lighting scheme is shown in Figure 4.1. The light beam passes through a pin-hole and illuminates just the tip of the leads. This presents to the camera an image with bright spots at the location of the leads and a dark area everywhere else. To fit this light assembly in the material movement handler (as described in the next chapter), another, more sophisticated, light assembly had to be designed. This design is shown in Figure 4.2. The diameter of the holes through which the light passes is only 20 mils in diameter. To get enough light to reflect from the tip of the leads which are 10x20 mils in size, a high intensity light is needed. To meet this requirement, two 3 inch long neon light tubes that emit

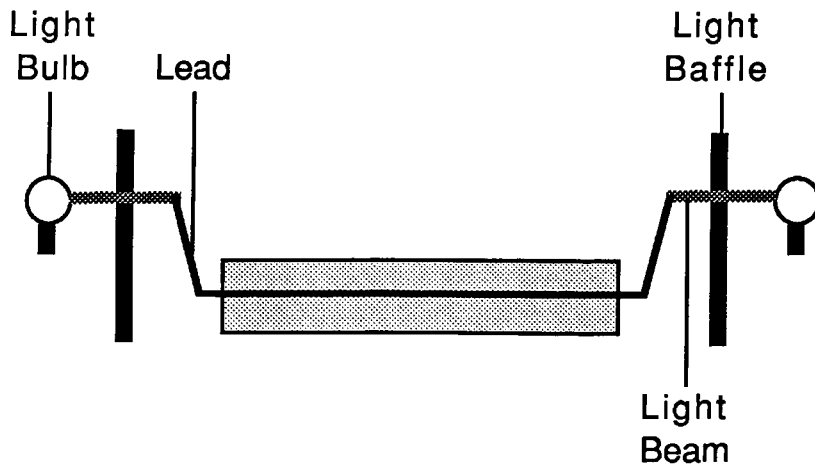


Figure 4.1 Lighting of the Leads.

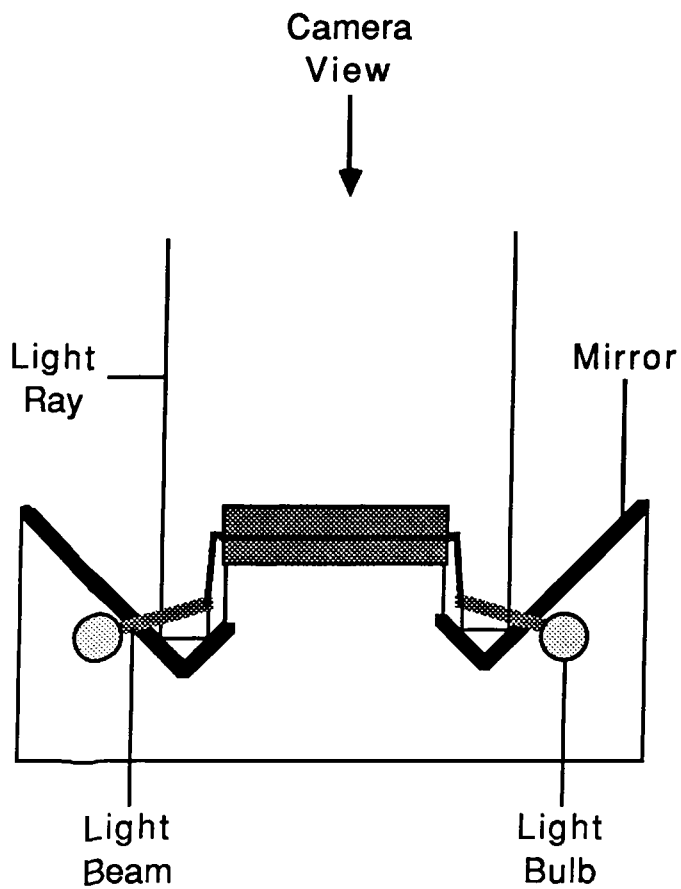


Figure 4.2 Practical Light Assembly for Lead Inspection

light from 400 nm to 720 nm are used. This spectrum is compatible with the frequency response of the camera and provides enough reflected light intensity from the tip of the leads. The light reflected by the tip of the leads is reflected again by the mirrors toward a camera placed directly above the part. This method shows both rows of leads to a single camera.

The umbra of a lead, as seen by the camera, is shown in Figure 4.3. The center of the lead is a bright spot that decreases in intensity as we move away from that center.

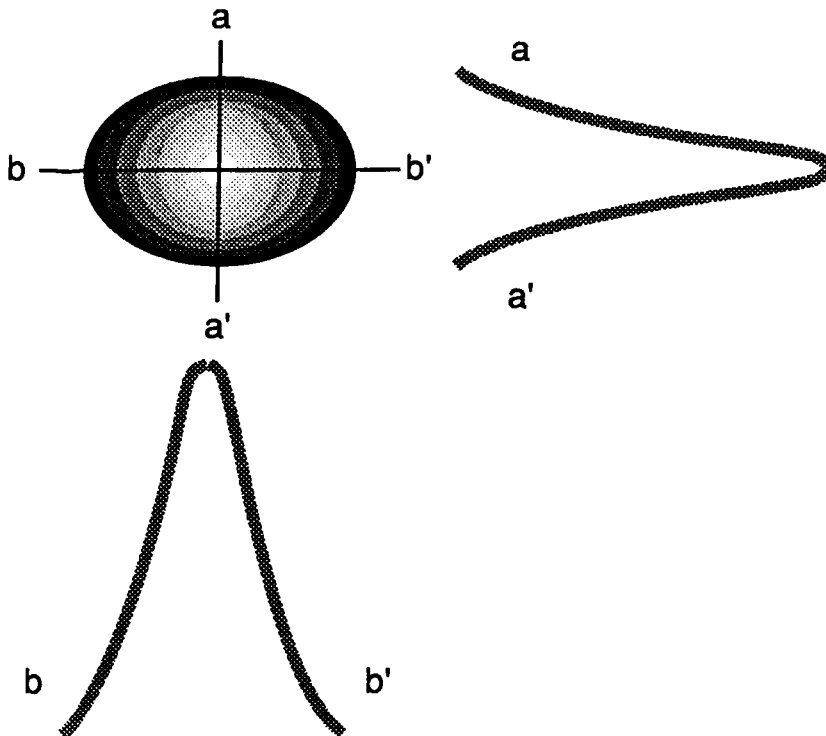


Figure 4.3 Umbra of the Tip of the Leads.

4.1.1 Learning the Leads

Before inspecting the leads, a few parts (7 to 15) are presented to the vision system and the average location of each lead is calculated. The average location is computed to create the mask used during the inspection.

To locate the leads, the highest grayscale value of each lead umbra is isolated by Eq. 4.1.

$$Y = [(X - c) \oplus B] - (X - c) \quad (4.1)$$

where

X is the original image snapped by the camera

B is a three dimensional structuring element

c is a constant

By subtracting the constant c from the original image X , we eliminate the background noise and isolate the umbra of each lead. The constant c should be larger than the peak levels of the camera noise and background lighting n_{\max} and smaller than the lowest gray scale value of the umbra tip l_{low} . The random variables n_{\max} and l_{low} cannot be directly controlled, but their mean and standard deviations can. To minimize the error in finding constant c , the mean m_n and standard deviation s_n of the camera noise and background light have

to be minimized while the mean of the peak values of the umbrae m_1 has to be maximize and the standard deviation s_1 minimized.

The values of m_n and s_n are minimized by using a high quality camera, shielding all the video cables, and protecting the inspection station from any stray light. To maximize m_1 , the highest practical light intensity is used as explained in the previous paragraph. To minimize s_1 a uniform light intensity is needed to light all the leads with the same intensity. This is achieved by using the uniform source (the neon light tube) and drilling the pin holes with a tight tolerance.

The second operation in Eq 4.1 is the dilation of (X-c) by a three dimensional structuring element B. B has the shape of a brick of length and width 1, and height 0. This structuring element flattens the tip of the umbrae and duplicates the peak values in an area of dimensions 1x1. The new image umbra created, differs from image (X-c) at every point except at the tip of the leads where the gray scale values are equal as shown in Figure 4.4. By subtracting the two umbrae we get a value of zero at the lead tip and values strictly greater than zero everywhere else. A new, two dimensional binary image is created from Y

of Eq. 4.1 by replacing all zeroes by ones and all other values by zeroes. The location of the ones are given by (x_{ij}, y_{ij}) where i is the number of the lead learned and j is the number of the part learned.

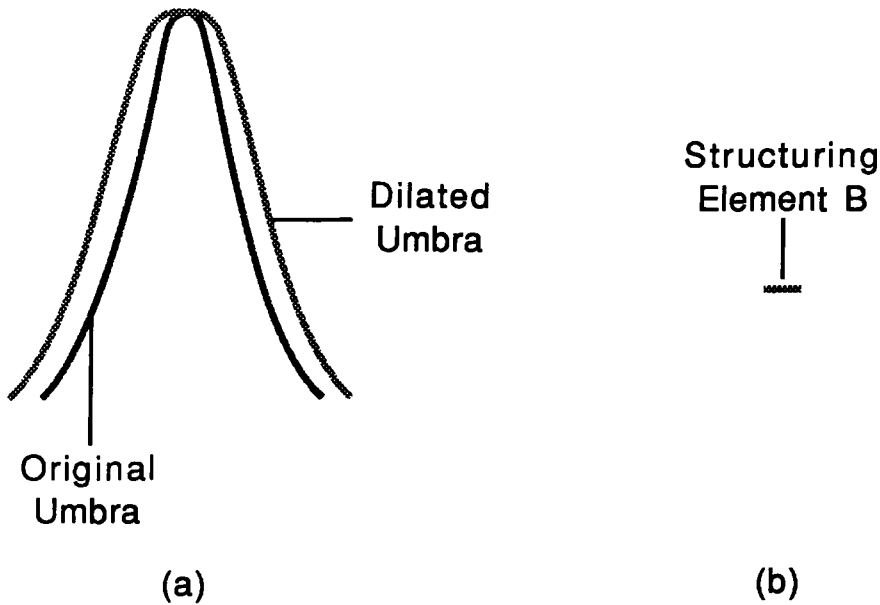


Figure 4.4. (a) Profile of the Original and Dilated Umbrae of a Lead and (b) Profile of the Structuring Element B.

After learning a few parts the average location for each lead is computed by

$$(x_i, y_i) = \sum_{j=1, n} (x_{ij}, y_{ij}) / n \quad (4.2)$$

Where

n is the number of parts learned.

The image with a value of one at locations (x_i, y_i) and zero everywhere else is called L . The mask M_1 for the inspection of the leads is computed by Eq 4.3.

$$M_1 = (L \oplus B)' \quad (4.3)$$

where

B is a two dimensional structuring element.

In Eq. 4.3 each point (x_i, y_i) is dilated by B , which is a square of size l , where l is the maximum deviation of the lead allowed by the inspection criteria. The dilated image is negated and stored in image M_1 . Figure 4.5 shows the operation of Eq 4.3 for one of the leads.

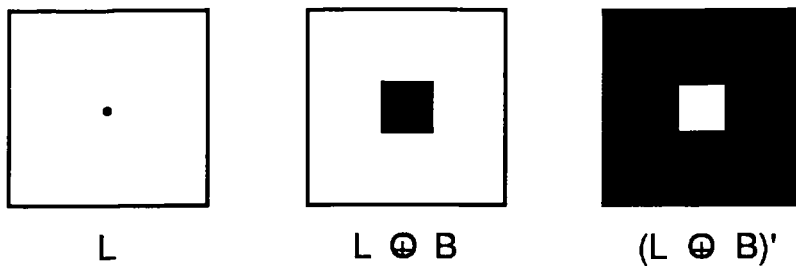


Figure 4.5. Creation of the Mask for the Lead Inspection.

5.1.2 Inspecting the leads

To inspect the leads an image is snapped. The highest gray scale value for each lead is found using the same algorithm as in the learning phase. The resulting image, with a value of one at the lead tips and zero everywhere else, is stored in Y . To insure that all the leads are present, the number of ones in Y are counted. If they are equal to the number of leads per package, all the leads are present. Otherwise some leads are missing and the part is rejected.

The second step in the lead inspection is the detection of bent leads. Image Y is transformed by Eq 4.4.

$$Z = Y \cap M_1 \quad (4.4)$$

If $Z = \emptyset$ none of the leads are bent and the part is passed, otherwise the number of ones in Z equals the number of bent leads and the part is rejected. This operation is shown in Figure 4.6 for a good lead and a bent lead.

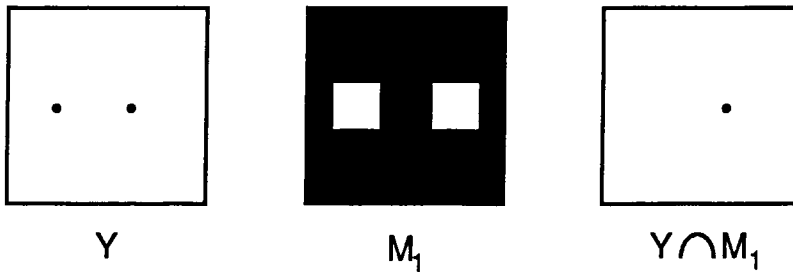


Figure 4.6. Inspecting the Leads.

4.2 INSPECTION OF THE TOP AND BELLY

This section presents the method used to inspect the top (inspection #2) and the belly (inspection #5) of the IC packages. These two inspections are similar in many ways. We will discuss the inspection of the top and point out any differences for the inspection of the belly. Figure 4.7 shows the right half of the top surface with two profiles along aa' and bb'. The profile along aa' shows the pin one dimple which is an optional depression used to indicate the location of pin 1 (this dimple is not present in inspection #5). The profile along bb' shows the white mark, the mold marks, and the pin one notch (this notch is not present in inspection #5). The surface depressions (mold marks, pin one notch, and pin one dimple) are smooth black surfaces. The mark is a series of characters stamped with white ink on top of the package. The remaining surface has a rough texture. These are usually all the features seen on the top and bottom of the package, sometimes though, dust and white powder can also be found.

To minimize the inspection time we need a simple segmentation technique that separates the defective areas from the normal areas. It has been shown [4],[5],[6], that low angular lighting (dark field) is optimum for surface defect inspection. This agreed with our experimental results.

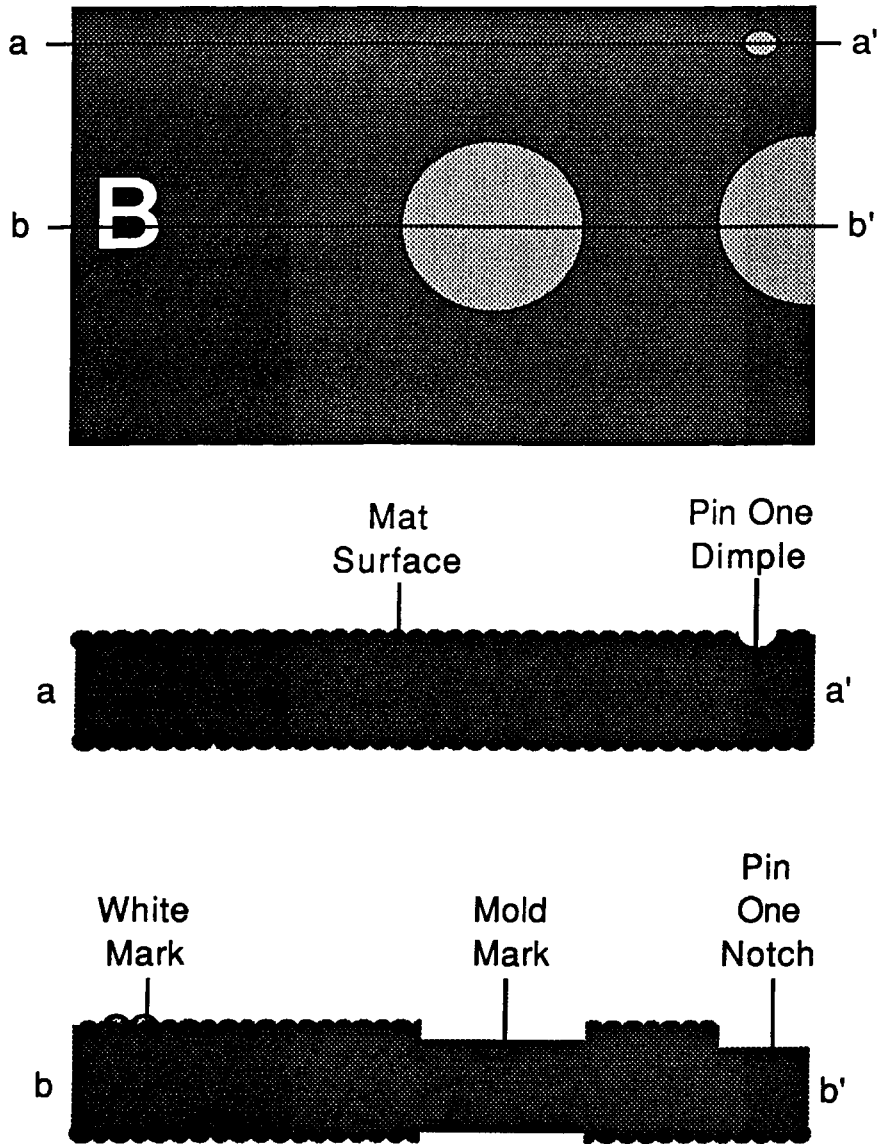


Figure 4.7. Profile of the Top Surface.

One of the drawbacks of low angular lighting is that light intensity decreases as the inverse square of the distance of propagation. To solve this problem, the light is focused with a convex lens which creates a uniform

parrallel beam. Also two light sources are used which add up to form a semi-uniform light across the package. Theoretically the light sources need to be at infinity to create a constant beam, but because of the resolution and noise of the imaging system, 12 inches are enough to create uniform light. This configuration is shown in Figure 4.8.

A total of at least 24 inches are needed to place the light sources on each side of the part. This becomes a problem when we try to fit this assembly in the material handling mechanism (see next chapter) because two feet of track are needed to accommodate this set up. The solution is to use mirrors to create virtual images of the lights where

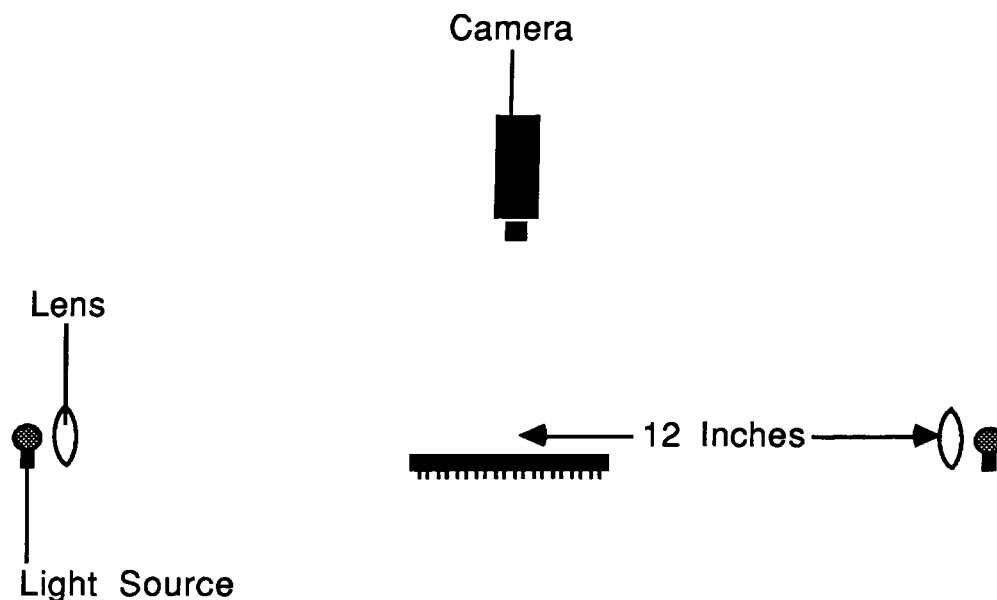


Figure 4.8. Optimum Uniform Low Angular Lighting.

it is impossible to physically mount them. This also reduces the number of light sources to only one as shown in Figure 4.9 but produces a second degree of approximation for light uniformity across the package. The fiber optic link shown in Figure 4.9 creates a somewhat coherent light beam which compensates for the second approximation.

With this set up, how are the top surface features seen from a camera placed at 90 degrees above the part? The white mark is a smooth surface that virtually reflects all light frequencies in all directions, it behaves like a Lambertian surface. The intensity of the reflected light is high and the mark is seen by the camera as a bright area.

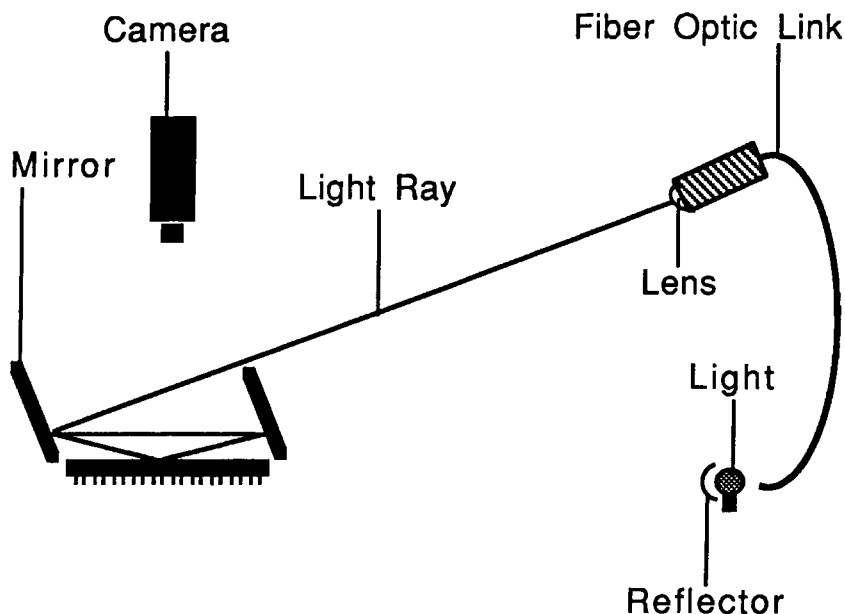


Figure 4.9. Actual Light Assembly.

The mold marks, pin one dimple, and pin one notch are smooth areas, and behave like specular surfaces (mirror like), where light is reflected at the same angle as the incident light. With low angular lighting, little intensity is reflected into the camera and these features are seen as dark areas. In addition, the depressions are shaded by the surrounding surface, this makes them even darker. Finally the remaining surface is neither Lambertian nor specular, the light reflected depends on the angle of viewing as well as the angle of incidence. On a microscopic level the surface looks like Figure 4.10. Surfaces A and B are specular surfaces that reflect low angular light from the

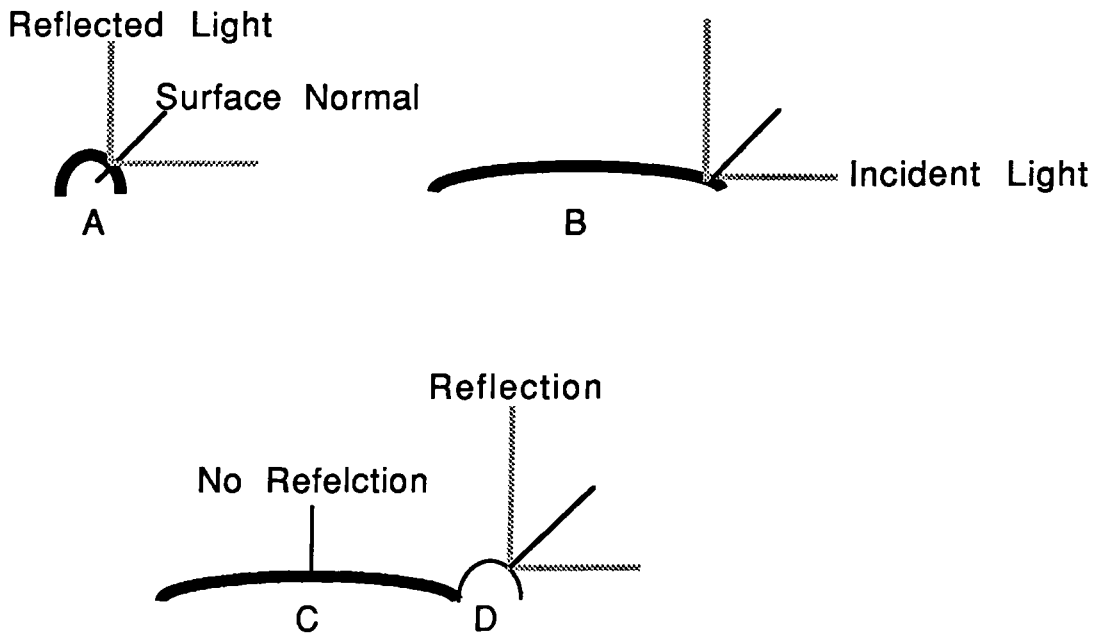


Figure 4.10. Microscopic View of the Top Surface Profile.

point where the surface normal is at 45° . Surface C on the other hand does not reflect any light because of its location next to the higher surface D. On the average, the whole surface has a reflectance value between those of the white mark and the depressions.

In summary, with low angular lighting, the mark is seen by the camera as a bright area, the depressions are seen as dark areas, and the remaining surface is of intermediate brightness. For comparison purposes using a gray scale between 0 and 255, the mark has a value around 200, the depressions have a value around 70, and the remaining surface has a value around 150. Figure 4.11 shows a profile along aa' of the dark field umbra for the top surface.

Now what do the defects look like from the camera with this light setup? A chip, a hole, and a pitted surface are depressions shaded by the surrounding areas, they are seen as dark features like the mold mark, pin one notch, or pin one dimple, discussed previously. A blister is a raised specular shape also seen as a dark area except for a ring inside of it. A grease mark is also a specular area seen as a dark area. Figure 4.12 shows the umbrae of the defects.

In summary, with low angular lighting all of the defects are seen as dark areas from a camera placed at 90° above the surface to be inspected. The remaining features of the surface (dust and powder) are seen as bright spots.

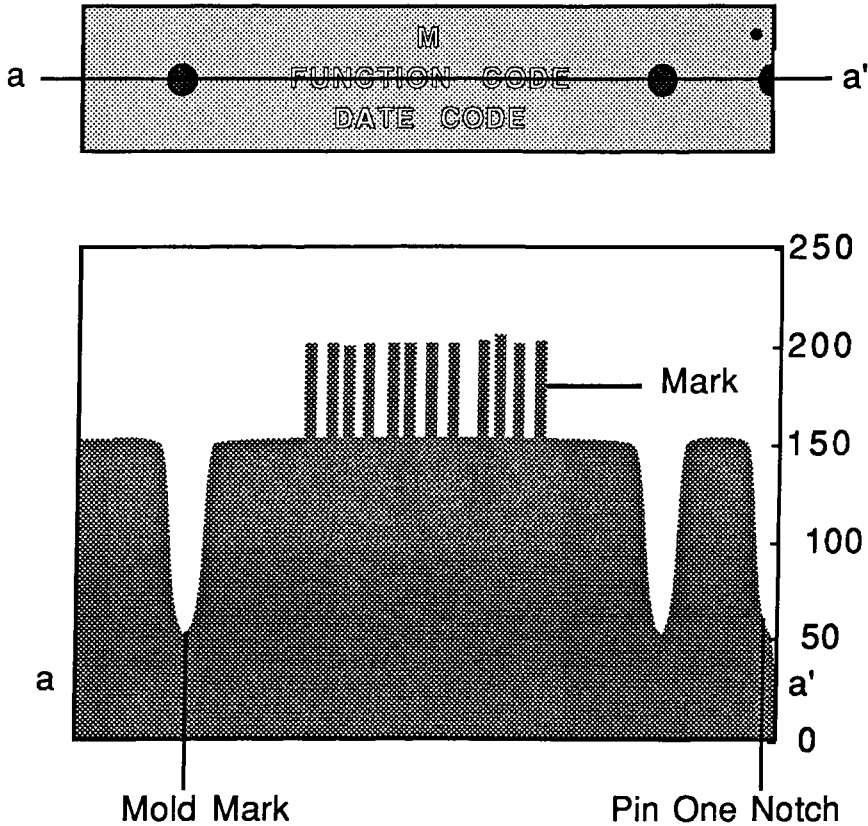


Figure 4.11. Profile of the Dark Field Umbra of the Top Surface.

With this setup, a simple threshold will separate the defects and the surface depressions from the other features. However, a second segmentation is needed to separate the defects from the depressions. Instead of taking this approach, the surface depressions are transformed to look like the regular surface before segmenting. To achieve this a mask has to be created during the learning phase.

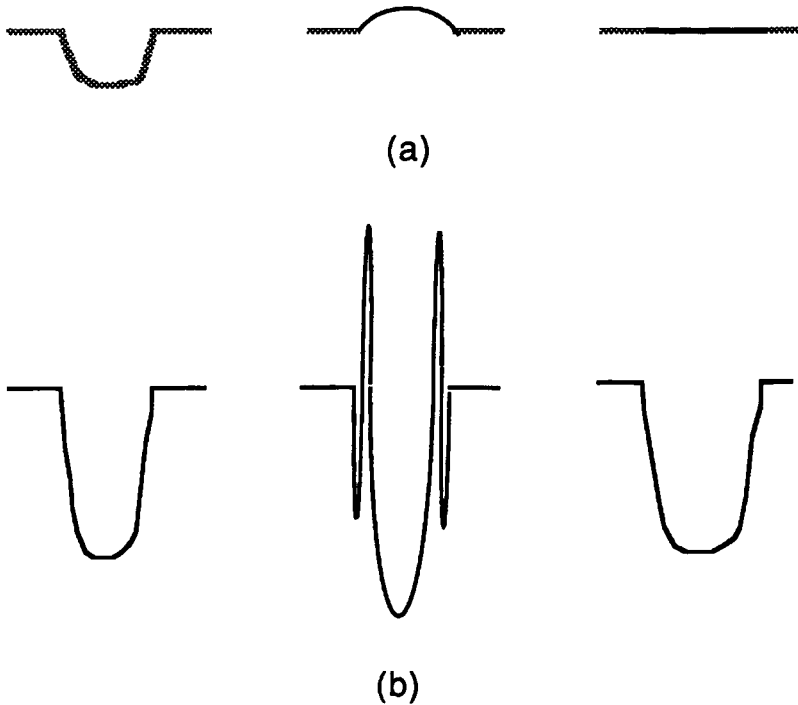


Figure 4.12. (a) Profile of Hole, Blister, and Grease Mark. (b) Umbrae of the Corresponding Profiles.

5.2.1 Learning the Top and the Belly.

Just like for the leads, before inspecting the parts, a few samples are presented to the vision system to create a mask and compute the threshold value.

Let X_i denote the i^{th} image learned. The X_i 's are transformed by Equation 4.5, to get the mask M_2 .

$$M_2 = \left[\bigcup_{i=1,n} \{ ((X_i)' - b)^A \} \right]_B \quad (4.5)$$

Where

b is a constant

A and B are three dimensional structuring elements

n is the number of parts learned

In Eq. 4.5, X_i is first negated, then the highest value b, in the mark area, is subtracted from X_i' to isolate the mold marks, pin one notch, and pin one dimple. The next operation is a closing with structuring element A. Recall that a closing is a dilation followed by an erosion with the same structuring element A. In this case A is a brick of length 20, width 5, and height 0. This closing eliminates the noise inherent to the image and flattens the tip of the mask. This is done on all the parts learned. When all these images are obtained, their union is taken to include their maximum values. Another closing is then performed on that union with structuring element B to create an envelope around the image because closing was shown to be extensive. B is a ball of radius 15.

Figure 4.13 shows the resulting mask. The profile of the umbra along aa' shows the negative of the mold marks and the pin one notch. It also shows the negative of the area

in front and behind the package. The umbra along bb' shows the negative of the mold mark again as well as the negative of the sides of the package. The sides of the package start with the same gray scale value as the top surface and get darker as we move away from that surface to the area around the part. The mask will level off these areas for the inspection.

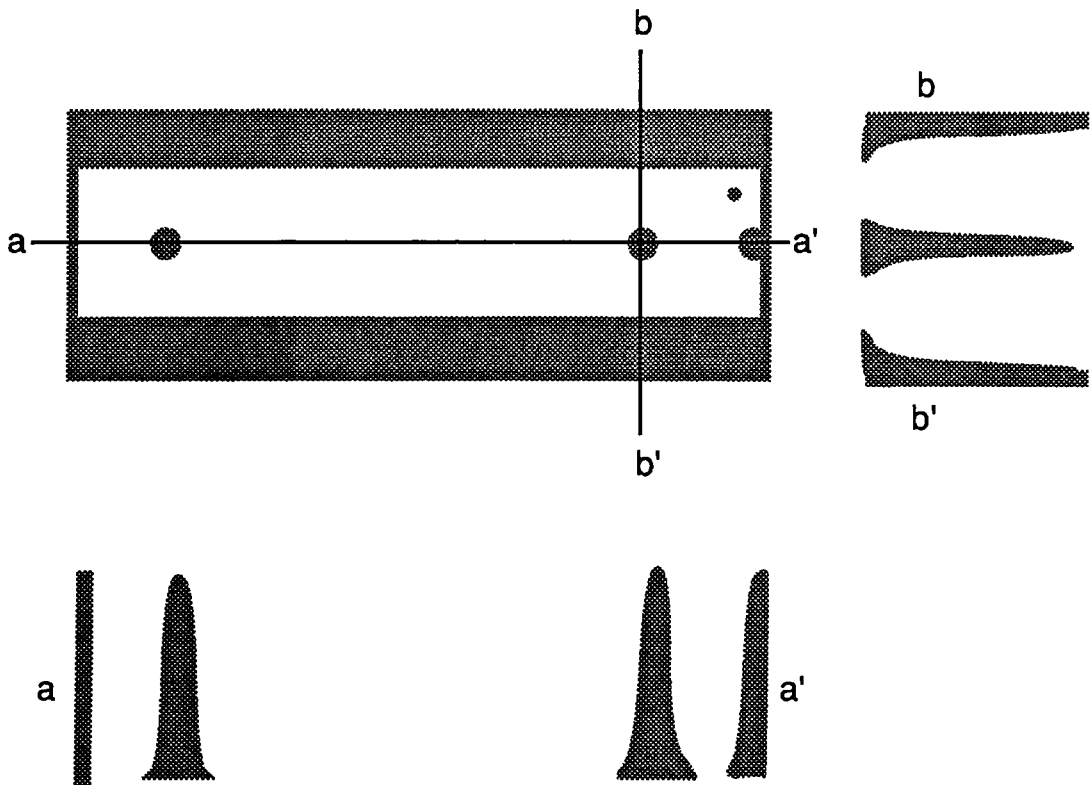


Figure 4.13. Umbra of the Mask of the Top Surface.

Another advantage of this method is of practical importance. If the mirrors or the light source are moved by

a small amount, the light intensity across the package will not be uniform, and a slightly darker area will be created to the right or left of the mark area. To compensate for these lower grayscale values, the mask image creates small grayscale values where the surface is darker. Note, however, that if the mirrors or the lights are moved by a large amount, the dark area will be inside the mark area. This will create an unusually low threshold (as computed in the next section) which signals a problem.

To calculate the threshold, the histograms $h_{mi}(x)$ of the mark area for all the learned parts is computed. The mean $m(x)$ and the standard deviation $s(x)$ for each gray scale value x of the histograms are calculated by Eq 4.6 and 4.7 respectively.

$$m(x) = [\sum_{i=1,n} h_{mi}(x)]/n \quad (4.6)$$

$$s(x) = \left\{ \sum_{i=1,n} [h_{mi}(x) - m(x)]^2 / (n-1) \right\}^{1/2} \quad (4.7)$$

where

n is the number of parts learned

x is the gray scale variable (x varies in $[0,255]$)

$h_{mi}(x)$ is the histogram of the mark area of the i^{th} part learned

The solution of Eq 4.8 is the value of the threshold t used during the inspection.

$$\sum_{x=1,t} s(x) = k \quad (4.8)$$

where

k is a constant determined experimentally.

t is the threshold

Appendix A gives a sample calculation.

5.2.2 Inspecting the Top and the Belly

To inspect the top of the parts, image X is snapped. X is added to M_2 from Eq 4.5. Note that if the part is not oriented properly, the pin one notch and pin one dimple will be on the wrong side. These features will not be added to the proper values of the mask and remain dark areas. They are detected like all the other defects. The resulting gray scale image is then transformed into a two dimensional binary image B using Eq 4.9.

$$b(x,y) = \begin{cases} 1 & \text{if } f(x,y) \leq t \\ 0 & \text{otherwise} \end{cases} \quad (4.9)$$

Where

$b(x,y)$ is the binary function of image B

$f(x,y)$ is the gray scale image function

t is the threshold from Eq. 4.8

This transforms the gray scale image function $f(x,y)$ into a binary function $b(x,y)$ where the defects have a value of 1 and the regular areas have a value of 0. If $B = \phi$, the top surface is not defective, and the part is passed. This is a fast algorithm that takes 150 ms to run. If defects are found as shown in Figure 4.14 then additional analysis needs to be performed.

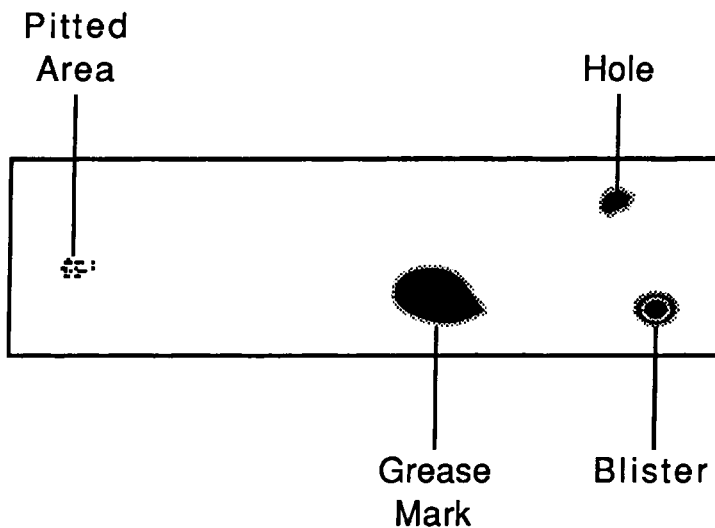


Figure 4.14. Binary Image Showing Defects.

For each defect, B will contain anywhere from a few non-connected points (e.g. pitted surface) to a solid area with a jagged edge (e.g. grease mark). Recall that a defect has to be larger than 1 before the part is rejected. To do this measurement we need to create the regular 4-hull of

each set and measure its height and width. But because the inspection only requires us to insure that the defects are larger than 1 before rejecting the part, we use Eq 4.10 to transform the binary image into a simulated regular 4-hull.

$$Y = B^C \quad (4.10)$$

where

B is the original binary image

C is a two dimensional structuring element

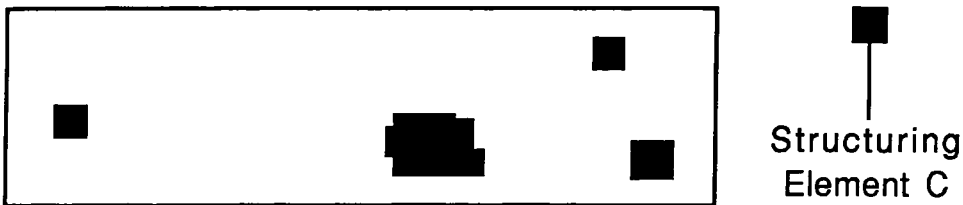


Figure 4.15. Simulated Regular 4-hull.

By closing the binary image with structuring element C (a square of size 1) all the points in a neighborhood of size 1 are connected as shown in Figure 4.15. Now if any of the defects are larger than the criteria specifies the part is rejected.

To measure each connected component, the image is eroded as shown in Eq 4.11.

$$Y = (X \ominus B_1) \cup (X \ominus B_h) \quad (4.11)$$

Where

B_h is a horizontal line of length l

B_v is a vertical line of height l

Image Y is shown in Figure 4.16.

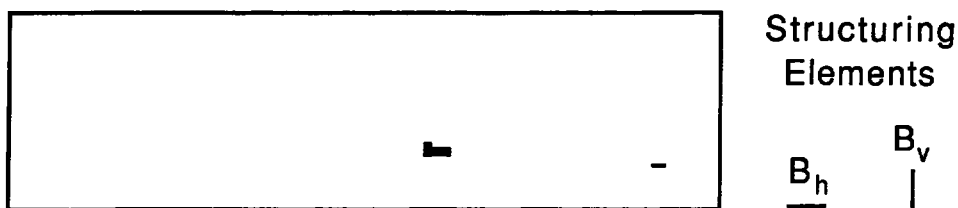


Figure 4.16. Defect Size Measurement.

If $Y \neq \phi$ then at least one of the defect is larger than the criteria specifies and the part is rejected. Otherwise the part is passed.

This additional processing adds up to 650 ms to the inspection time.

5.3 INSPECTION OF THE TAIL AND HEAD

The tail and head of the part have a different profile than the top surface but are similar to each other. We will discuss the inspection of the tail (inspection #3) and note any differences for the inspection of the head (inspection

#4). Figure 4.17 shows the tail and a profile along aa'. The area referred to as plastic injection area is a very rough and irregular surface, it is where the hot liquid plastic is injected into the mold. The mold flash is where

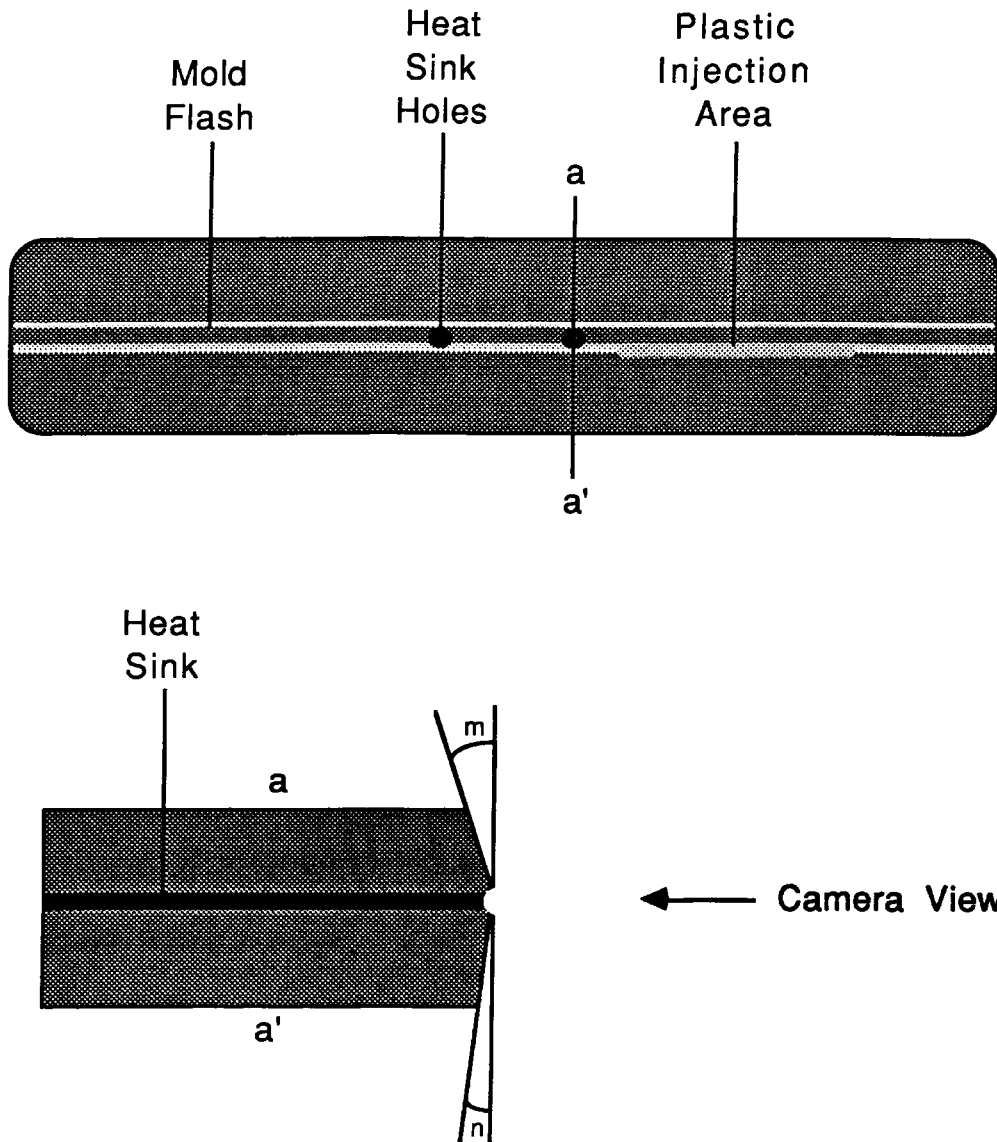


Figure 4.17. Tail of the Package with a Profile.

the top and the bottom halves of the mold meet when the parts are made. The heat sink holes are formed by the heat sink inside the part. In the profile view, the angles m and n have different values. These angles also differ from lot to lot. The remaining area can be either rough or smooth.

The variations on the end of the parts are irregular because they are not controlled properly during the molding process. For the defects to show up easily, low angular lighting is again needed. The light assembly used is shown in Figure 4.18. In this case, again, the light decreases with distance from the source, but this is compensated for during the learning process.

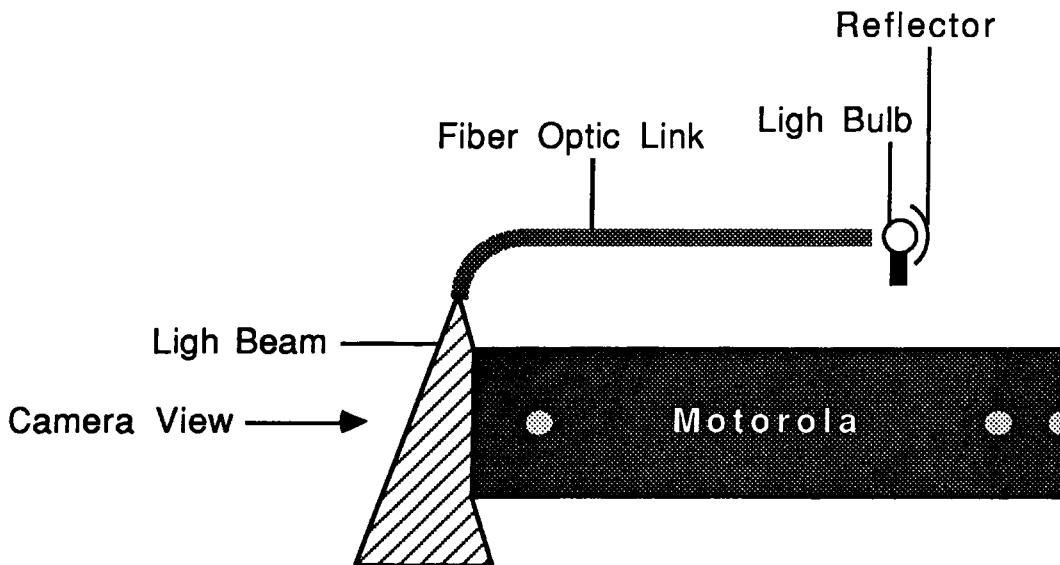


Figure 4.18. Lighting of the Tail of the Package.

With this lighting the profiles of the image are shown in Figure 4.19. Note the non-uniformity of the light across the image in both directions. The light reflected from the plastic injection and the flash is always bright. On the other hand, very little light is reflected from the heat sink areas. The reflectance from the remaining area depends on the finish as well as the distance from the light source. If it is a rough finish, the intensity is larger than if it is a smooth finish. For comparison purposes using the gray scale, the injection area and the flash have gray scale values of 230, the heat sink holes have values around 90. The light intensity on the remaining surface is not uniform along the width of the package and varies from 120 to 200.

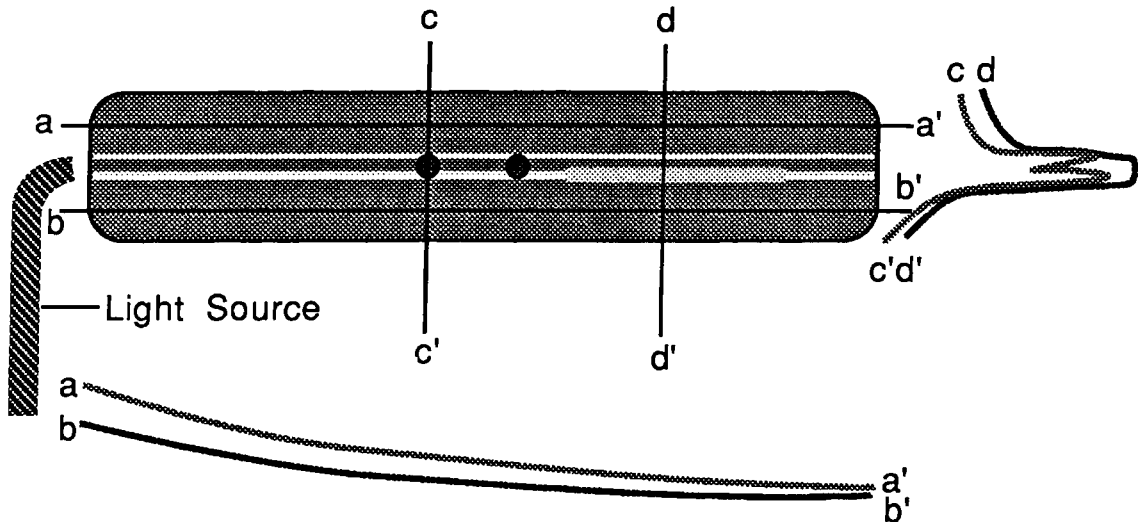


Figure 4.19. Umbra of the Tail of the Package.

The kinds of rejects that occur on the tail and head are similar to those described for the top of the package except for a chip defect which occurs on the ends of the part. A chip defect is a very rough surface, which is seen as a bright area.

5.3.1 Learning the Tail and the Head

Learning the tail follows the same pattern as learning the top. A few parts are presented to the vision system, a mask and two thresholds are calculated, the high threshold is calculated for the chip defects and the low threshold for the other defects.

Let X_i denote the i^{th} image learned. The X_i 's are transformed by Eq. 4.12. to create the mask M_3 .

$$M_3 = \left[\bigcup_{i=1,n} [(X_i)_A] \right]_B \quad (4.12)$$

Where

A and B are three dimensional structuring elements

n is the number of parts learned

The raw image X_i is opened with structuring element A in a shape of a ball of radius 3. The opening again creates an

envelope around X_i . The resulting image is then negated, and the union of all the learned images is taken. The final image is opened again with structuring element B (a ball of radius 5) that creates the envelope of the union of the images.

The two thresholds are t_h for the chips, and t_l for the other defects. Once more, the mean $m(x)$, and the standard deviation $s(x)$ of the histograms are calculated using Eq 4.13 and Eq 4.14 respectively.

$$m(x) = [\sum_{i=1,n} h_i(x)]/n \quad (4.13)$$

$$s(x) = \left\{ \sum_{i=1,n} [h_i(x) - m(x)]^2 / (n-1) \right\}^{1/2} \quad (4.14)$$

Where

$h_i(x)$ is the histogram of $U_{j=1,i-1} [(X_j)_A]' + (X_i)'_A$

The histogram is taken from the image $[(X_i)']_A$ added to the background of the previously learned parts. The high threshold is found by solving Eq 4.15 for t_h

$$\sum_{i=t_h,255} h(x) = k_h \quad (4.15)$$

while the low threshold t_l is the solution of Eq. 4.16.

$$\sum_{i=0, t_1} h(x) = k_1 \quad (4.16)$$

where k_h and k_l are constant determined experimentally.

5.3.2 Inspecting the Tail

Inspecting the tail is similar to inspecting the top. An image is snapped and added to the background M_2 from Eq. 4.12. The two thresholds are used to transform the gray scaled image function $f(x,y)$ into a binary image function $b(x,y)$. Eq. 4.17 is used for this segmentation.

$$b(x,y) = \begin{array}{ll} 0 & \text{if } t_l \leq f(x,y) \leq t_h \\ 1 & \text{otherwise} \end{array} \quad (4.17)$$

The algorithms of the top and bottom are repeated to find the defect and measure their sizes.

From the above we should notice that the lighting and the learning techniques are more elaborate and therefore more time consuming (in the case of the software) than the inspections. This was purposefully done because the learning algorithm is performed on a few parts, 7 to 15, while the inspection is done on thousands of parts for every lot.

CHAPTER 5

THE HARDWARE

Recall from Chapter 2 the system needed to solve the inspection problem. Figure 2.3 is repeated in Figure 5.1 for completeness. The previous chapter presented the illu-

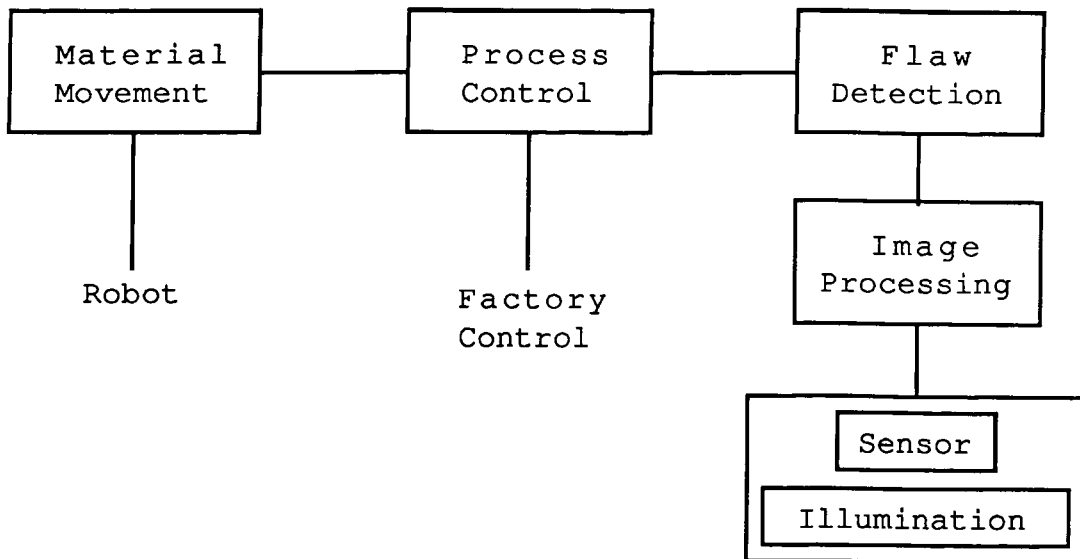


Figure 5.1. Inspection System.

mination for each of the inspections. This chapter starts by describing the sensor used to input the data to the system. Next the image processing hardware is described,

followed by the flaw detection hardware. The last section describes the process control system. The material movement handler is described in the next chapter.

5.1 Sensor

The sensor used for the inspection system is a Charge Coupled Device (CCD) video camera. Appendix B presents the theory of CCD operation. In this paragraph we present a summary of that appendix. The video camera module adopts an interline-transfer organization in which precisely aligned photosensors and vertical shift registers are arrayed interlinearly. A horizontal shift register links the vertical shift registers, as shown in Figure 5.2. Light is sensed by photosensors, which generate electronic charges proportional to the light intensity. The generated charges are fed into the vertical shift registers all at once. The charges are then transferred from the vertical shift register to the horizontal shift register successively to reach the output amplifier. The amplifier transforms the electrical charges into an electrical signal conforming to the RS-170 standard. This analog signal is sent to the image processing system.

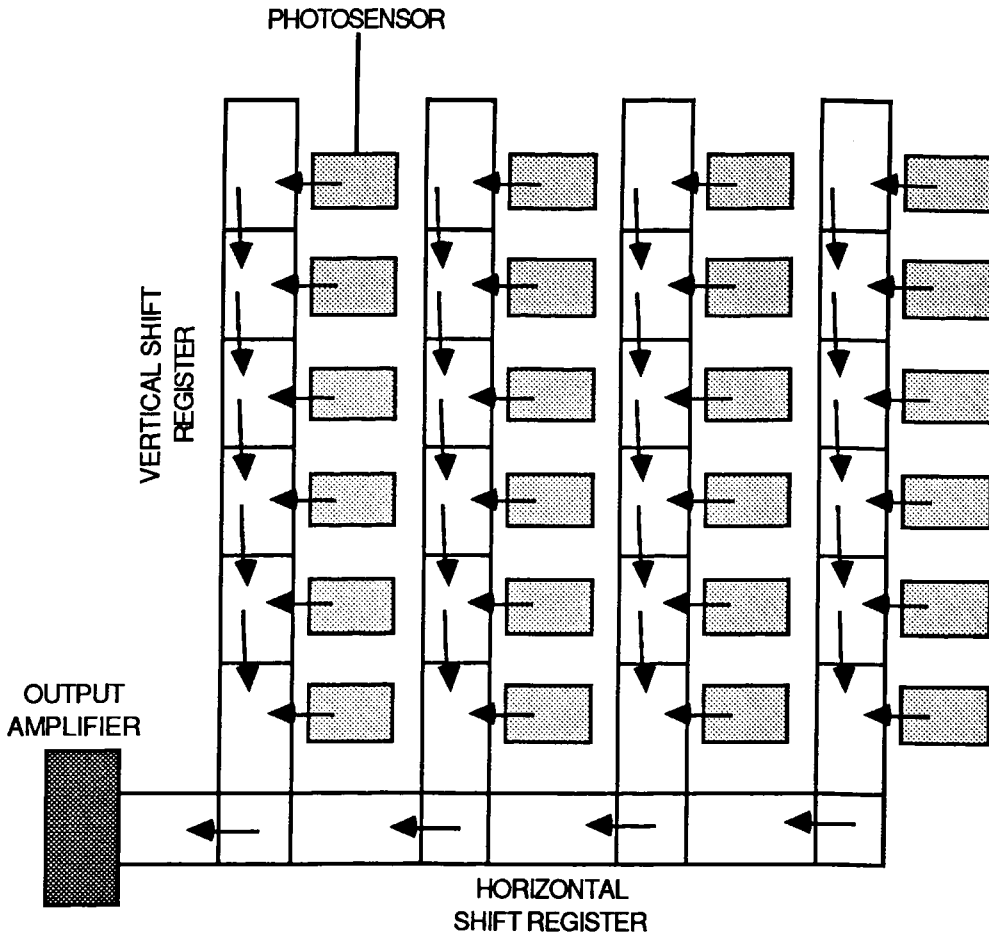


Figure 5.2 CCD Array Arrangement.

5.2 Image Processing System

The image processing system consists of an Image Flow Computer (IFC) manufactured by Machine Vision International in Ann Arbor, Michigan. A block diagram of the system is shown in Figure 5.3.

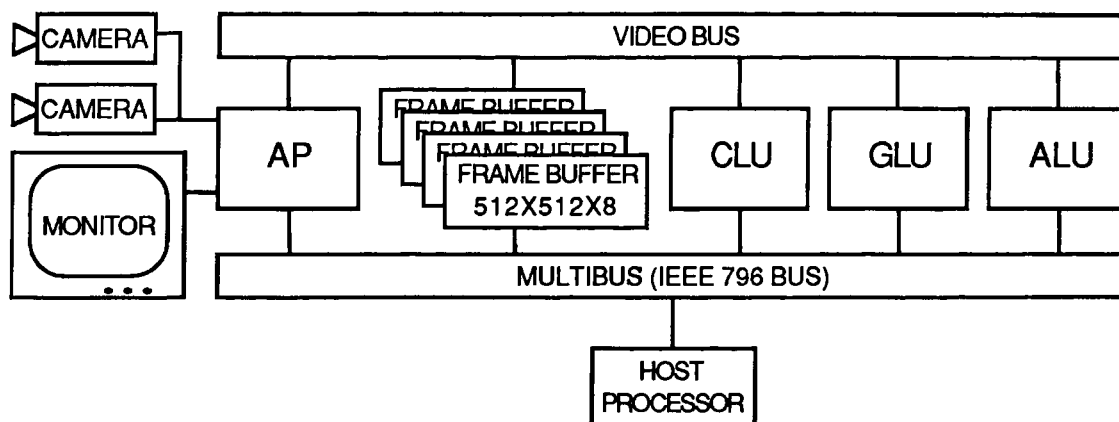


Figure 5.3 Image Flow Computer Hardware.

The RS-170 video signals from the cameras are received by the Analog Processor (AP). The AP board digitizes the input signals from as many as 8 cameras into 8 bit bytes serial data. The digitization is not necessarily linear, any lookup table can be used as a digitization transfer function. The digital data is then stored in one of 4 frame buffers. The frame buffers are fast 8 bit memories configured to store the images in a 512 by 512 array. The AP also sends analog signals to the color monitor in an RGB format to display black and white or pseudo-color images from the cameras or the frame buffers.

The main processor is a pipeline processor based on the Cyto computer [7],[8], developed at the Environmental Research Institute of Michigan. It consists of a Count and Locate Unit (CLU), a Geometric and Logic Unit (GLU), and an

Arithmetic and Logic Unit (ALU).

The CLU performs histogram calculations as well as all count and locate operations. The GLU and ALU run programs that apply sequences of neighborhood transformations to the digitized video images. Image processing operations are based on the concepts of cellular automata. Every cell or picture element (pixel) of an image is subjected to an identical sequence of time-discrete transformations, the transformed value of a pixel being determined by the values of a finite group of cells that make up its "neighborhood." Each neighborhood transformation is performed in an individual cytocomputer processing element referred to as a processing stage.

A cytocomputer is a serial pipeline of programmable processing stages, in which each stage performs a single transformation of the processing sequence on an entire image. Images are entered into a cytocomputer in a line-scanned format and progress through the pipeline of processing stages at a real time rate. Following an initial delay to fill the pipeline, images can be processed at the same rate they are scanned (33 milliseconds) provided enough stages are available.

Digital images and cellular automata have a common conceptual framework. Each pixel of a digital image can be thought of as a cell transition function on a digital image,

then we can apply the transition function to modify or transform the configuration of cell states into new configurations. Of critical importance then is whether neighborhoods and transition functions exist that will cause images to be transformed in a predictable and useful manner when subjected to long sequences of neighborhood operations.

The image processing language - mathematical morphology - differs from conventional approaches in that the basic manipulative unit is pictorial, and operations deal with images as wholes. Image processing is treated as a computation involving images as variables in algebraic expressions. These expressions may combine several images through both logical and geometrical relationships.

Cytocomputer operations are implemented in highly efficient cellular computer architectures, and the computations are very fast. Image processing algorithms are either variables representing images or image operations. The image being processed is referred to as the active image. Other images referred to in an image-algebraic expression are called structuring elements. In an image processing algorithm, we can modify the active image by probing it with structuring elements or combining it with other active images.

The neighborhood in a cellular space determines the set of structuring elements that can be employed in a single

neighborhood transformation. All structuring elements used in a neighborhood transformation must be sub-images of the window.

Consider a two-dimensional cellular array, where each cell of the lattice has connection with a finite collection of other cells that make up its input. The geometric pattern of the cell's input to a given cell is the same as the pattern of the points in the neighborhood. Figure 5.4 illustrates a cellular array with the connection pattern for a 3x3 window configuration.

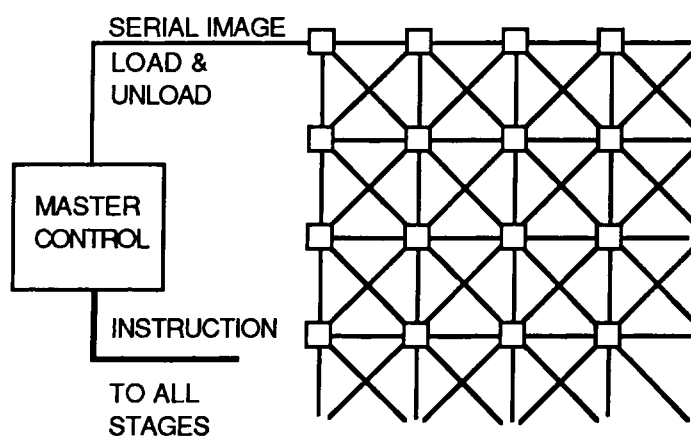


Figure 5.4. Array similar to cellular automata of identical cells connected to their nearest neighbor for iterative neighborhood processing of digital images.

In Figure 5.4, each cell of the array consists of a register for storing the state of the cell and a transition

module that computes the new value of the cell state as a function of the states of the cells in the window. When a common clock pulse is applied to each cell in the array, all cell state registers pass from their previous state to a new state as determined by the programming of the transition logic module.

A cytocomputer consists of a serial pipeline of commonly clocked neighborhood processing stages. Shift registers within each stage store two contiguous n pixel scan lines, and window registers hold the nine neighborhood pixels that constitute the 3×3 input to the neighborhood transition logic module as shown in Figure 5.5. All neighborhood transformations and data transfers are computed within a single clock

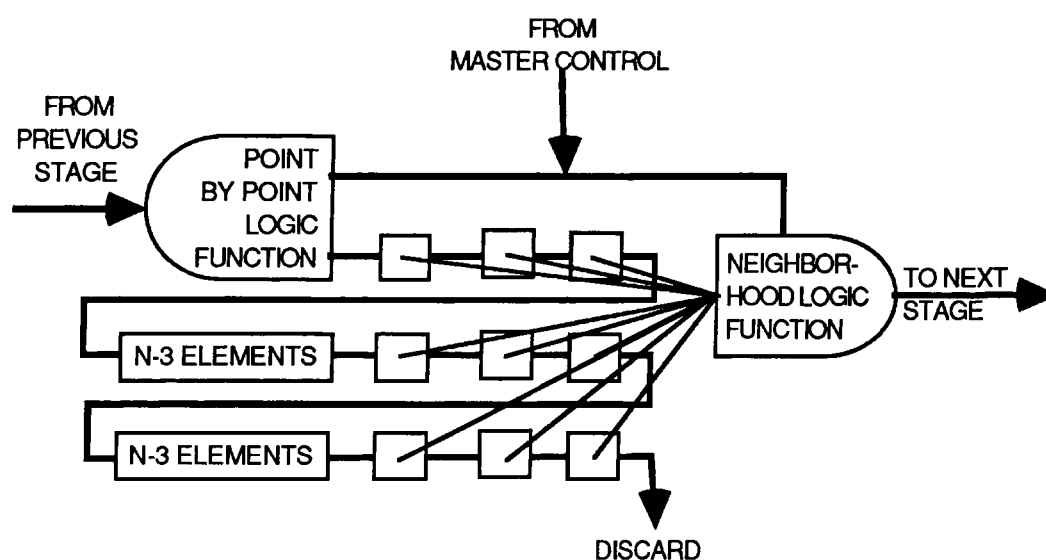


Figure 5.5 Geometric and Logic Unit.

period. In each discrete clock period, a new pixel is clocked into the stage, and simultaneously the contents of all shift register are shifted by one element. In addition, operations that do not involve the states of the pixel's neighbors, such as gray-value scaling and bit setting, are performed in a separate point transition logic module (the Arithmetic and Logic Unit - ALU) to simplify the neighborhood transition logic circuit. Because stage input and output occur at the same rate, stages can be cascaded, with all stages operating in parallel. We can visualize a series of 3x3 windows following each other across the image, each processing the previous stages output as shown in Figure 5.6. If not enough stages are available the image can be piped from the last stage back into the first stage until all the processing is completed.

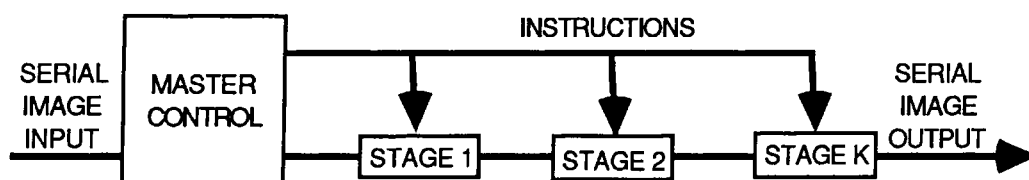


Figure 5.6. Cascading GLU's.

Transformations implemented in a cytocomputer stage fall into two main categories. In the first category, called the silhouette, or 2-D, transformation, image algebra operations

are applied to planar binary images. The second category, called the umbra, or 3-D transformation, the same set of operations are performed on gray-scale images, where the gray value represents the brightness of a pixel.

The image data is transferred between the different stages via a wide bandwidth (10MHz) video bus. To set up the different stages before processing, a microprocessor is linked to the IFC via a multibus.

5.3 Flaw Detection System.

The flaw detection system is the same as the host processor used to control the IFC stages. It consists of an Intel 8086 microprocessor, an 8087 coprocessor, and 1 MByte of RAM. The peripherals to the main processor are a 20 Mbyte hard disk drive, a serial port, and a special parallel port that links the flaw detection system to the process control hardware.

5.4 Process Control Hardware

The process controller is an IBM PC/AT. The main processor is an 80286 microprocessor with an 80287 co-processor. The system is equipped with a 20 MByte winchester disk drive, a 1.2 Mbyte floppy disk drive, 512k

of RAM, a special parallel port for communication with the IFC, and a parallel port for communication with the material movement handler.

Since the process controller is a general purpose computer, it was used to develop the inspection software as well as the process control software.

CHAPTER 6

THE MATERIAL MOVEMENT HANDLER

This chapter describes the operation of the material movement handler and the interface with the robot and the vision system.

6.1 THE HANDLER: AN OVERVIEW

The material movement handler is required to present all six sides of each integrated circuit to the cameras. At every inspection station, ambient light must be blocked out so it does not interfere with the special lighting described previously. The IC path should be free of static electricity. The input and output have to be within the workspace of the Unimation Puma 760 robot for the handler to be loaded and unloaded by that robot. Special signals are needed to communicate with the robot as well as with the vision system.

A market search revealed that no standard handler could meet these requirements. It was decided that a custom

machine had to be designed and built. A request for proposal was sent to custom handler houses. After receipt of the proposals and a few meetings with the different vendors a decision was made to have the Equipment Group of Motorola in Mesa, Arizona, design and build the material movement handler. The final design agreed upon is shown in Figure 6.1.

Here is an overview of the operation of this machine. The parts are loaded by a robot or operator into the input track. The first part is released into inspection station #1. This station performs the inspection of the leads. The part is then released into station #2 where camera #2 sees the top of the package. It is then released into Picker Wheel 1 where it is rotated by 120 degrees counterclockwise and the tail of the package is shown to camera #3. This wheel is then rotated by another 120 degrees in the same direction and the part falls into the second Picker Wheel. This wheel then rotates by 60 degrees counterclockwise and presents the head of the part to camera #4. It then rotates by 180 degrees in the direction of the arrow and releases the part on its back into the short track which shows the belly of the part to camera #5. The part is then released and slides into the third Picker Wheel. If the part is a reject, wheel 3 rotates counterclockwise by 150 degrees and releases the part into the Bad Bin Output, otherwise it

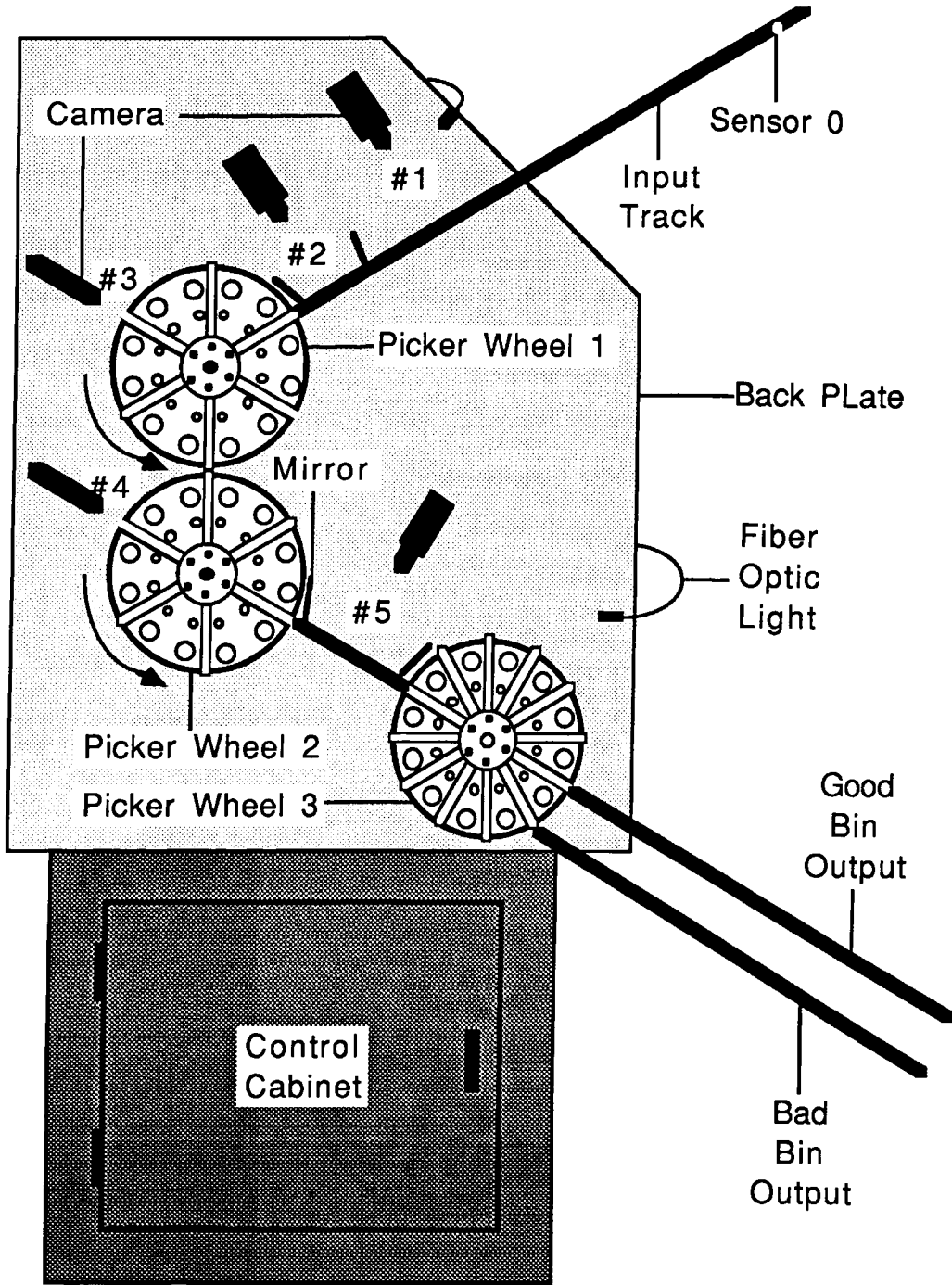


Figure 6.1. Material Movement Handler.

rotates clockwise by 180 degrees and releases the part into the Good Bin Output.

One problem encountered with this Picker Wheel arrangement is that the shortest three parts (40, 28, and 24 lead parts) fall deep in the wheel if the longest part (48 lead package) is to fit in the wheel. To overcome this problem a Picker Wheel Stop was designed as shown in Figure 6.2. For the 48 lead packages the stop is not needed. For the other three sizes the stop is screwed into the appropriate

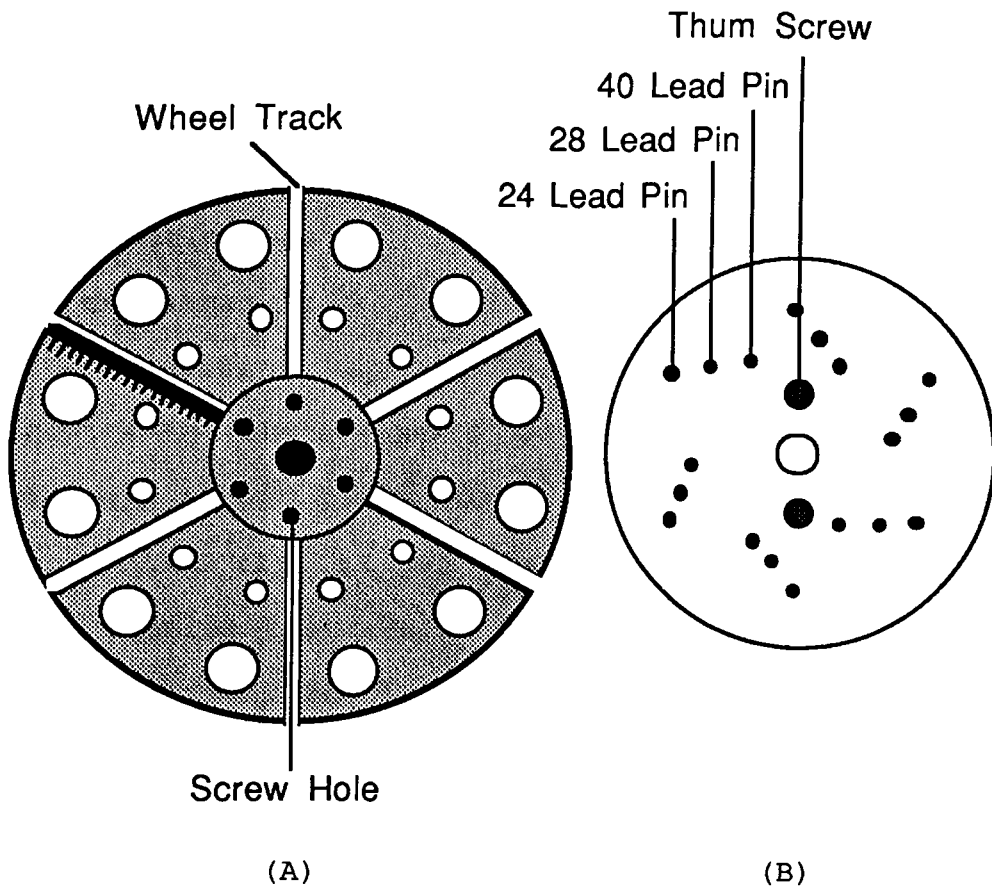


Figure 6.2. (A) Picker Wheel and (B) Picker Wheel Stop.

ate holes of the picker wheel using the thumb screws. The proper pin fits in the wheel track and stops the part from sliding all the way in the wheel. The stops need to be readjusted every time a different size package is run through the handler.

The Picker Wheels, tracks, cameras, etc. are mounted on a quarter inch steel plate called the Back Plate. This is the main support of the top half of the handler. A side view of that machine is shown in Figure 6.3 (left side view of Figure 6.1). Notice that the top half of the machine is tilted by 10 degrees from the horizontal. This causes the parts to always rest on one of their sides. This technique places the parts at the same position every time they stop at an inspection station. This is obviously needed for the reasons described in Chapter 4.

To keep ambient light out of the inspection stations a door covers the whole assembly. This door can be opened to fix jams and adjust the handler. The control panel is a hex numeric keypad that allows the operator or technician to manually control the machine and find problems. The three motors drive the three Picker Wheels. The control cabinet houses the power supplies and the card cage. Four boards are used to control the machine. The main board is a single board computer based on the 68000 microprocessor. It controls the main logic of the handler's operation as well

as the other three cards. Another 68000 board controls the three motors. The third board is a logic board that controls the sensors, and finally the fourth board controls the solenoids and the I/O of the handler.

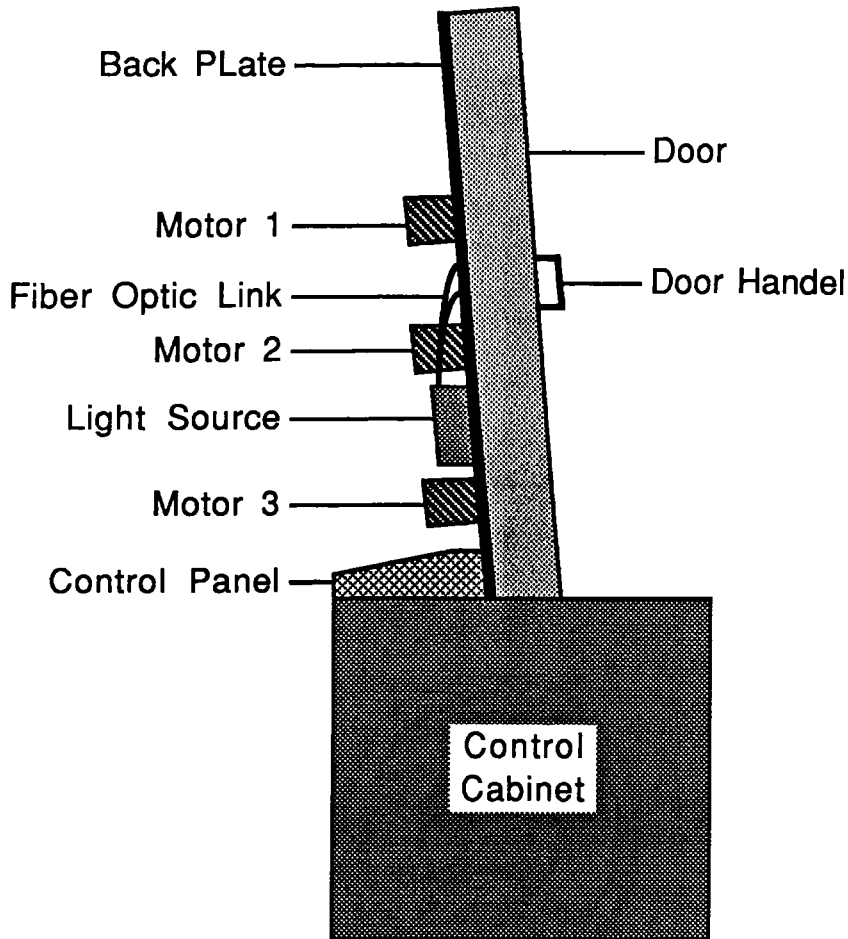


Figure 6.3. Side View of the Handler.

This machine is a gravity feed system. In other words it relies on gravity to move the parts from one station to

another. The parts are stopped using a solenoid as shown in Figure 6.4. The solenoid is normally extended. When the part slides to this location it is stopped by the plunger. When it needs to be released, the solenoid is retracted, and the part slides over the plunger.

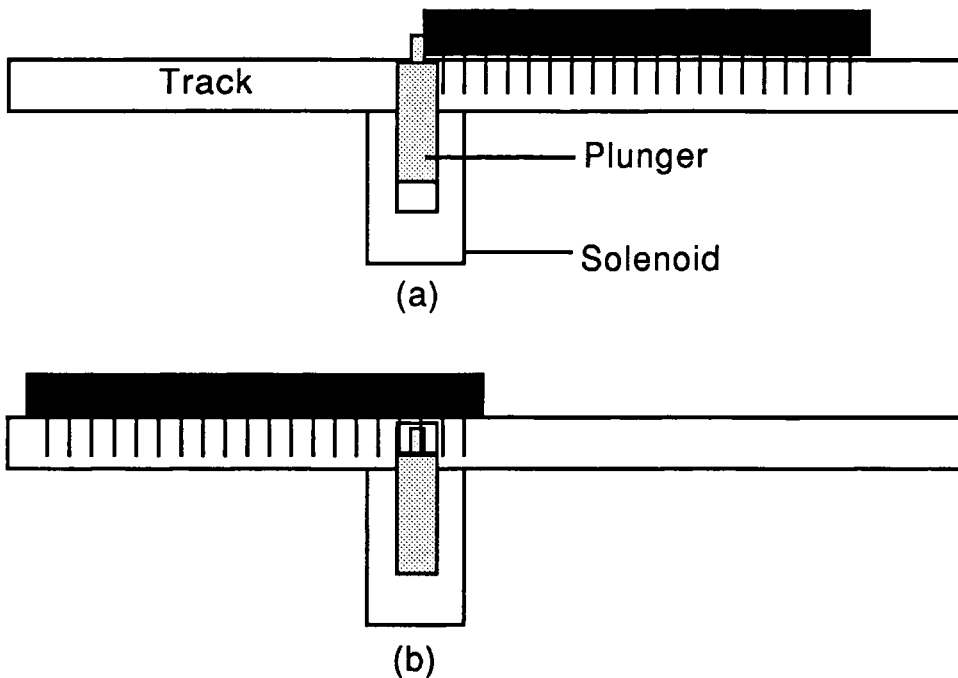


Figure 6.4. (a) Solenoid in extended position. (b) Solenoid in retracted position.

To confirm the presence or absence of a part, an emitter-receiver pair of diodes is located on each side of the track as shown in Figure 6.5. The emitter is an infrared diode while the receiver is a visual/infrared diode. When the part reaches the solenoid, the light beam

is blocked and the processor detects this event. After the solenoid is retracted the processor waits for the beam to be restored before extending the solenoid.

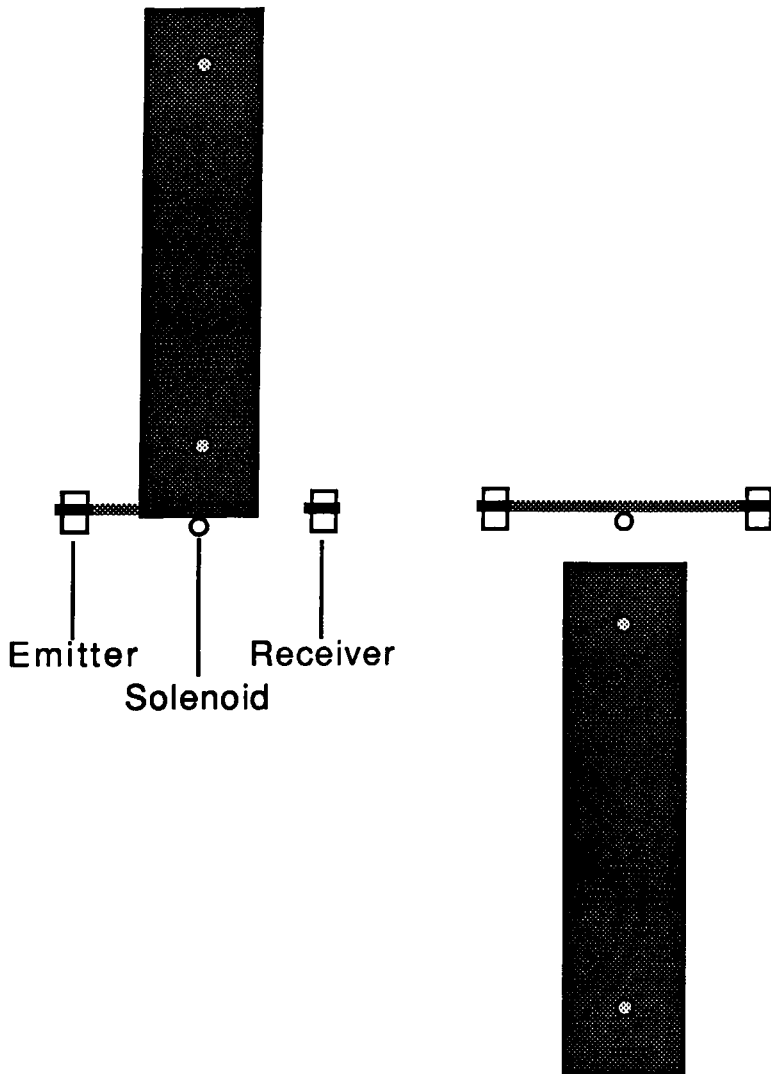


Figure 6.5. (a) Light beam is broken by a part in position. (b) light beam is restored after the part is released.

6.2. COMMUNICATION BUSES

While the parts are going through the handler, information needs to be sent to the vision system and to the robot to inform these systems of its status. Two separate buses were designed, one to communicate with the robot and another one to communicate with the vision system.

6.2.1 Communication With the Robot

The first bus is used to inform the robot of the status of the input and output bins as well as the overall status of the handler. Figure 6.6 shows all the signals used to communicate with the robot. These are 7 parallel open collector TTL signals that go high (+5 Volts) when the signal is asserted and low (0 Volt) when the signal is negated. This is a unidirectional bus from the handler to the robot.

When the input track is full of parts, a sensor located at the input is blocked (Sensor 0 in Figure 6.1), the handler then sends the Input Full signal to the robot to acknowledge proper loading of the input. When the input track is empty, a sensor detects this event (sensor 1 in Figure 6.8) and an Input Empty signal is sent to the robot to load more parts into the input track. When the good bin

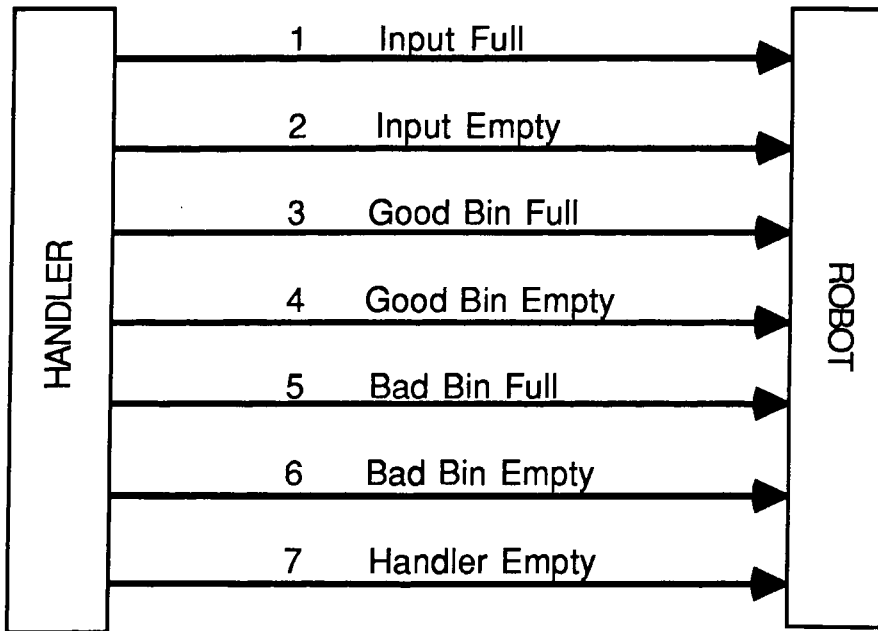


Figure 6.6. Communication Bus Between the Handler and the Robot.

is full of parts the handler asserts the Good Bin Full signal to tell the robot to empty that bin. When all the parts are out of the bin the Good Bin Empty signal is asserted so the robot can move the sleeve away from that bin. The Bad Bin Full and Bad Bin Empty are similar to the signals just described for the good bin.

The Handler Empty signal is needed at the beginning and the end of each lot. Before the lot is started the robot checks this signal to make sure that no parts were left over from the previous lot. At the end of the lot, when the robot does not have any more parts to load into the handler

it will wait for this signal to be asserted before disposing of the lot.

6.2.2 Communication with the Vision System

The communication between the handler and the vision system is more sophisticated. Two unidirectional buses link the handler to the vision system. The first one sends data and protocol signals to the vision system while the other one receives similar information. A custom protocol was developed to optimize the communication overhead.

As shown in Figure 6.7, the lowest six bits of each unidirectional bus are used to transfer data. This allows the transmission of 64 different messages. The upper two bits are protocol lines that control the proper transmission and acknowledgment of these messages.

When data is ready to be transmitted, it is placed on the data bus. Then the Data Ready line is toggled (if it was high it is changed to low, and if it was low it is changed to high). When the receiving system detects the Data Ready signal it reads the data and toggles its Data Receive Acknowledge line. Before the transmitting system sends new data, it reads the Data Receive Acknowledge signal and confirms that it has been toggled since the last write before putting a new byte on the bus.

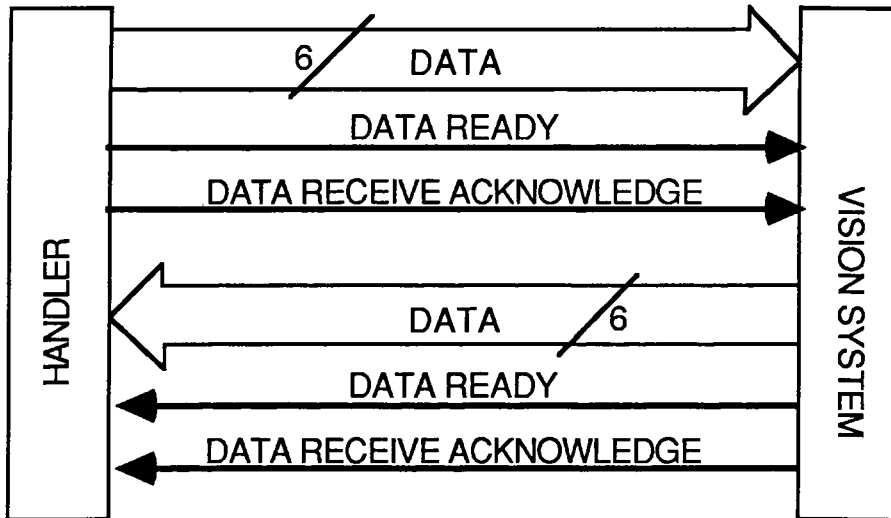


Figure 6.7. Bus Between the Handler and the Vision System.

The data transmitted by the handler is shown in Table 6.1 while the data transmitted by the vision system is shown Table 6.2. In the following, the data values of the PART AT STATION IS GOOD is the same as SNAP ACKNOWLEDGE because no conflict occurs if these two signals have the same values. A description of this data is given as we describe the operation of the handler.

6.3 OPERATION OF THE HANDLER

To start a lot, the vision system sends the VISION SYSTEM OFF signal. This resets the communication protocol.

Table 6.1. Data Transmitted by the Handler.

Data Value	Signal Description
0	HANDLER OFF
1	PART AT STATION #1
2	PART AT STATION #2
3	PART AT STATION #3
4	PART AT STATION #4
5	PART AT STATION #5
6	HANDLER JUST INITIALIZED
7	DOOR OPENED
8	DOOR JUST CLOSED
9	WHEEL 1 TIMED OUT
10	WHEEL 2 TIMED OUT
11	WHEEL 3 TIMED OUT
12	PART IS STUCK IN WHEEL 1
12	PART IS STUCK IN WHEEL 2
12	PART IS STUCK IN WHEEL 3
20	JAM AT STATION #1
21	JAM AT STATION #2
62	STAND BY
63	HANDLER OFF

The vision system then sends the number of leads per package. The operator or the robot then fills the input track with parts. These parts are singulated using two different solenoids for the different size packages as shown in Figure 6.7. Solenoid 1 is used for the 48 and 40 lead packages while solenoid 2 is used for the 28 and 24 lead packages. Unlike all other solenoids, the singulation

Table 6.2 Data Transmitted by the Vision System.

Data Value	Signal Description
0	VISION SYSTEM OFF
1	SNAP 1 ACKNOWLEDGE
2	SNAP 2 ACKNOWLEDGE
3	SNAP 3 ACKNOWLEDGE
4	SNAP 4 ACKNOWLEDGE
5	SNAP 5 ACKNOWLEDGE
6	PART AT STATION #1 IS BAD
7	PART AT STATION #2 IS BAD
8	PART AT STATION #3 IS BAD
9	PART AT STATION #4 IS BAD
10	PART AT STATION #5 IS BAD
11	48 LEAD PACKAGES
12	40 LEAD PACKAGES
13	28 LEAD PACKAGES
14	24 LEAD PACKAGES
15	ONE VISION SYSTEM
16	TWO VISION SYSTEMS
17	THREE VISION SYSTEMS
18	FOUR VISION SYSTEMS
19	FIVE VISION SYSTEMS
20	VISION SYSTEM INITIALIZED
63	VISION SYSTEM OFF

solenoids have a rubber pad mounted on the end that holds the package at its center instead of holding it from the front as explained previously.

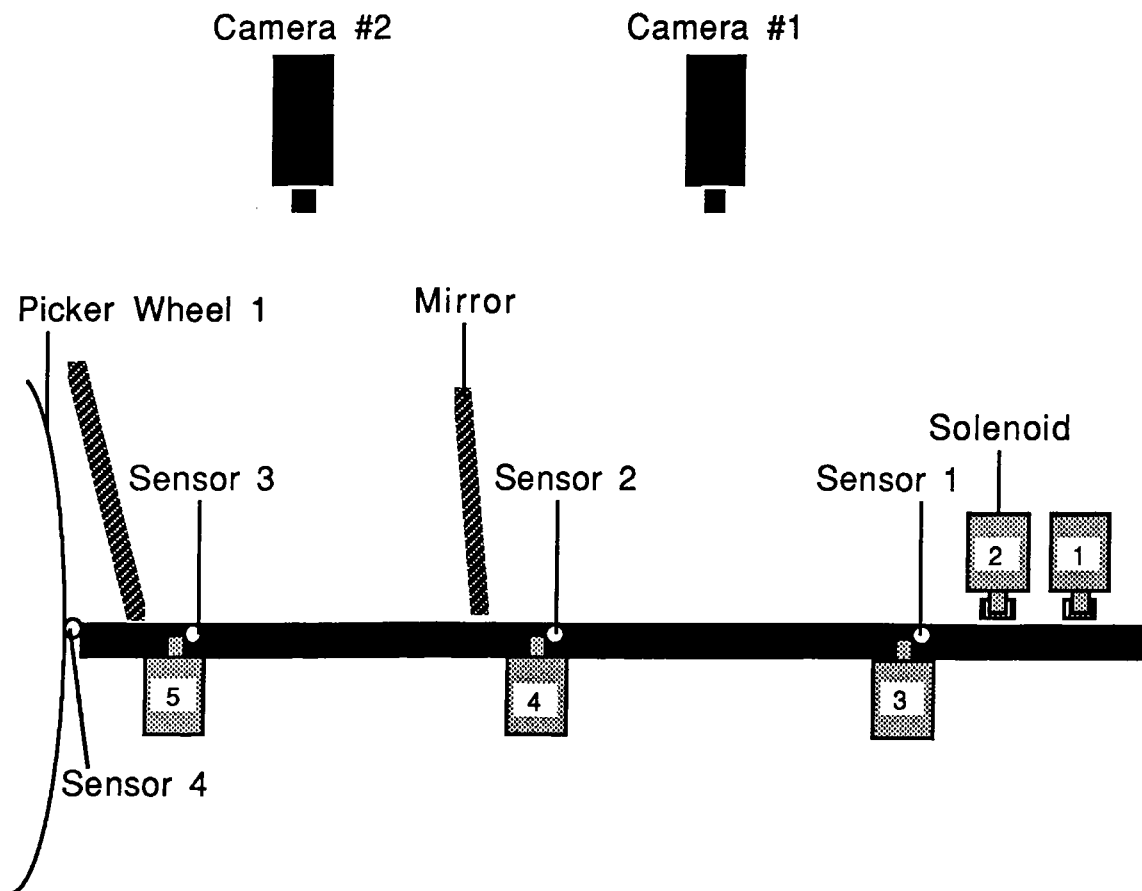


Figure 6.8. Track for Inspection #1 and #2.

Initially solenoids 1 and 2 are retracted while solenoid 3 is extended. The parts are loaded into the input rail and slide until the first one reaches solenoid 3 where they are stopped. A sensor detects the presence of the first part (sensor 1 in Figure 6.8). If the handler was previously told by the vision system that these were 48 or 40 lead packages solenoid 1 is extended and stops the second part from sliding when solenoid 3 is retracted, otherwise

solenoid 2 is extended. After solenoid 3 is retracted the first part slides and is stopped by solenoid 4. Sensor 2 confirms the presence of the part. If it does not, a solenoid jam message (JAM AT STATION #1) is sent to the vision system which alerts the operator. Usually the part drops and is ready to have its leads inspected, this is inspection #1. The handler sends a PART AT STATION #1 signal to inform the vision system that a part is ready to have its lead inspected. The vision system takes a picture and sends a SNAP 1 ACKNOWLEDGE signal acknowledging the completion of the snapping of image #1. While this was taking place, solenoid 3 was extended, then solenoid 1 or 2 were retracted and the parts in the input track slide down until the first one reaches solenoid 3. Again solenoid 1 or 2 is extended to stop the parts from sliding when solenoid 3 is retracted. Now that the vision system has a picture of the leads it can start processing the image to find defective leads. While it is performing this inspection the part at station #1 is not needed any more. Solenoid 4 is retracted and the part falls into station #2 where it is stopped by solenoid 5 and detected by sensor 3. Also a part is positioned at station #1 as described previously. Now we have parts at station #1 and station #2 ready for inspection. When the vision system completes the lead inspection of the part that is now at station #2, it sends

the result (PART AT STATION #1 IS GOOD or PART AT STATION #1 IS BAD) to the handler. The handler then sends a PART AT STATION #2 message. The vision system again snaps a picture of the part and sends a SNAP 2 ACKNOWLEDGE signal to the handler. Solenoid 5 is then retracted and the part falls into Picker Wheel 1. Sensor 4 confirms that the part is completely into the wheel. If it is not, the wheel vibrates for about 15 seconds trying to get the part to slide in. If it does not succeed, it will send a WHEEL 1 JAM signal to the vision system, which in turn will request help from the operator. If the part slides through properly, the wheel rotates by 120 degrees counterclockwise and the tail of the part is presented to camera #3 for inspection as shown in Figure 6.9.

After the handler receives the result of inspection #2, it sends a signal to the vision system telling it that a part is at station #1, and the previous routine is repeated until the vision system sends the SNAP 3 ACKNOWLEDGE signal. Now wheel 1 is rotated by 120 degrees counterclockwise and the part falls into wheel 2. Sensor 5 again confirms that the part is completely in wheel 2. This wheel is then rotated by 60 degrees to present the head of the part to camera #4. Once more all the previous parts are inspected until the SNAP 4 ACKNOWLEDGE signal is received from the vision system. Then wheel 2 rotates by 180 degrees counter-

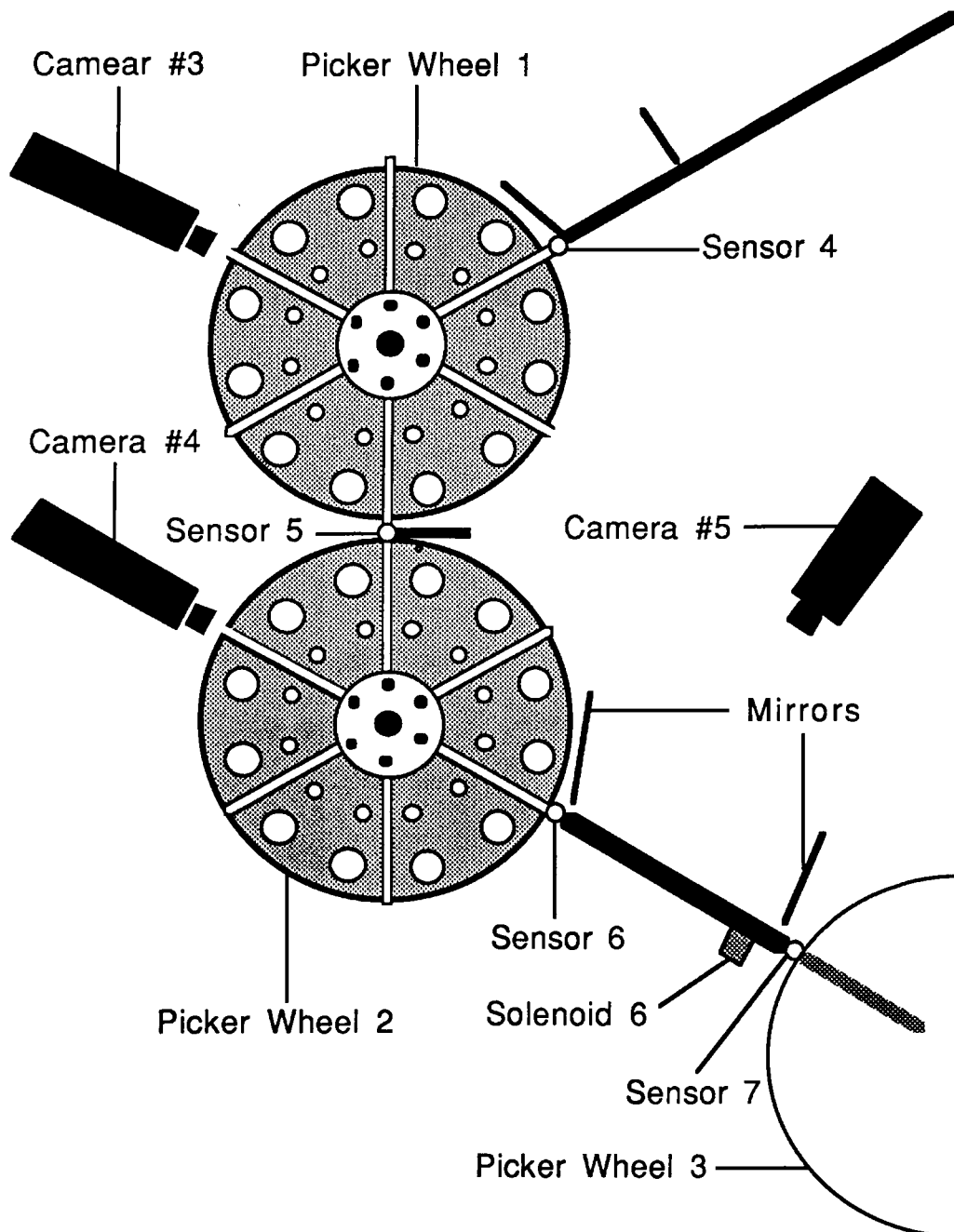


Figure 6.9. Picker Wheel Assembly.

clockwise and the part slides onto the short track with its belly presented to camera #5. Solenoid 6 stops it at the

proper location. After all the previous stations have been inspected and the SNAP 5 ACKNOWLEDGE signal is received, solenoid 6 is retracted and the part slides into wheel 3. Sensor 7 confirms that the part is completely into the wheel. When the result of station #5 is received, the handler's main processor analyzes all previous results for this particular part. If they were all good, wheel 3 rotates by 180 degrees clockwise and the part is released into the Good Bin Output. If one or more of the results for this part was bad, the Picker Wheel rotates by 150 degrees counterclockwise and releases the part into the Bad Bin Output (See Figure 6.10). Sensors 8 and 9 confirm that the parts are out of the wheel. When the Good Bin Output is full, solenoid 7 extends and a Good Bin Full signal is sent to the robot. While the robot is moving to empty the bin, two more parts can be released in the Good Bin Output. To empty the bin, the robot places a sleeve at the output and a signal retracts solenoid 8, the parts slide out of the handler and into the sleeve. Sensor 10 confirms that all the parts are out of the bin before extending solenoid 8. Solenoid 7 is retracted and the two parts that were stored in the output bin are released. The same holds true for the Bad Bin Output.

Note the efficiency with which the handling and the inspections are performed. Full advantage is taken of the

two systems to get as much parallel processing as possible. Also note the extensive use of sensors to insure the proper operation of the handler and to easily locate problems in case of a jam.

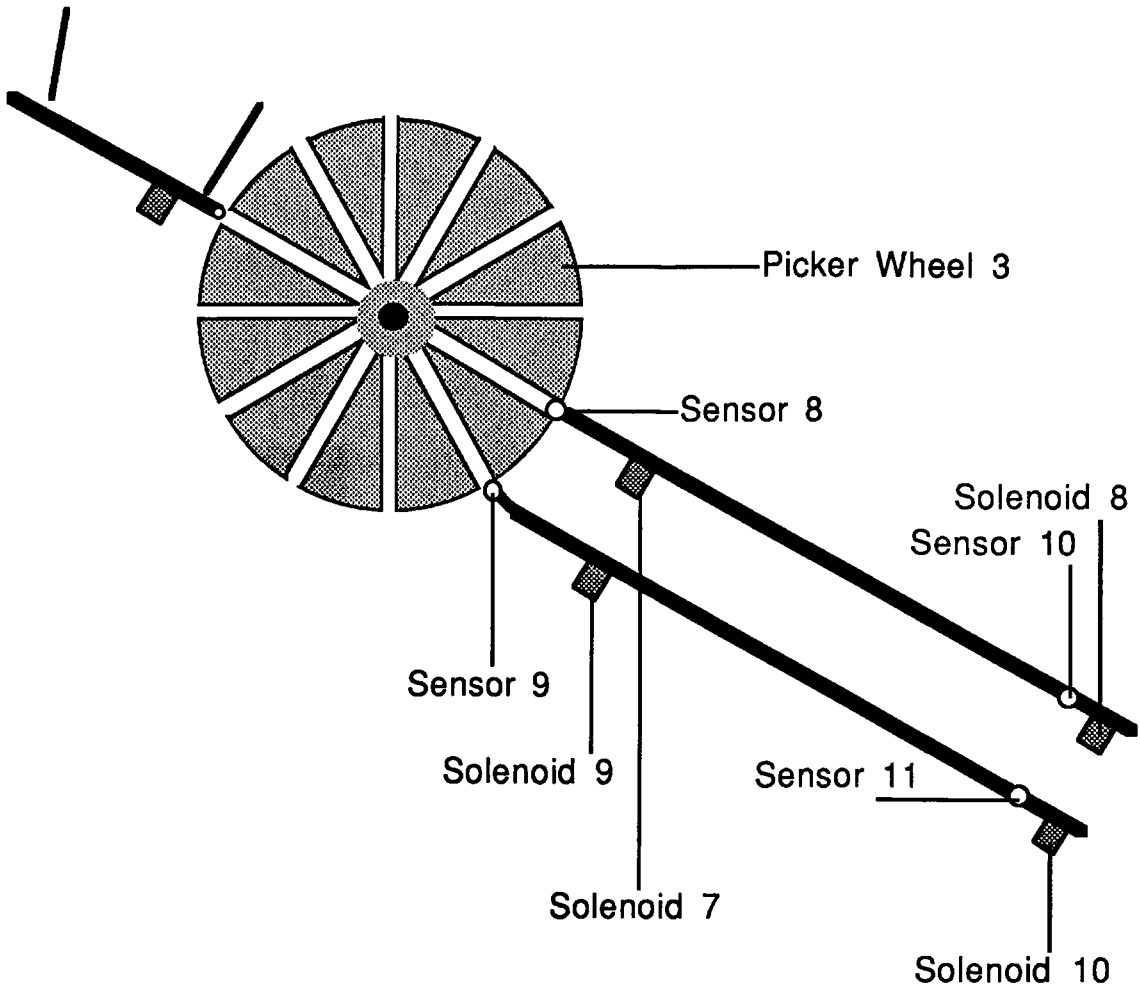


Figure 6.10 Output Tracks.

CHAPTER 7

CONCLUSION

7.1 OPERATOR INTERFACE

To integrate the inspection system with the current manufacturing process it is of prime importance to make the operator interface as simple as possible. The operation of the handler is menu driven. The machine works with the operator interactively to set up the system and inspect the product.

To start a lot the operator types "n" on the keyboard to signal the start of a new lot. The system responds by displaying "Are you sure you want to start a new lot? (y/n): " on the CRT monitor. The operator types a "y". The system then prompts for the number of leads per package with "Enter the number of leads per package (24/28/40/48): ". The operator types the number of leads per package, e.g. 40. The system then reminds the operator that the picker wheels might need to be adjusted by "Are the wheels adjusted for 40 lead packages (y/n): ". If the wheels are not adjusted, the operator will adjust them and then types "y". Now the

operator starts loading the machine with parts or instructs the robot to do so if a robot is available. The system learns the first few parts then starts the inspection. This is all the setup required. In case a jam occurs during the inspection, the machine alerts the operator, and pinpoints the location of the problem.

Training a new operator to run the machine takes only fifteen minutes as compared to the traditional two day training in a classroom environment and 9 days under close observation by a supervisor. The operators judged the machine as being easy to operate, and comfortable to work with.

7.2 PROJECT SCHEDULE

This is the only machine in operation known to the author that inspects the IC chip packages. So no previous work was available to base the research on. The project schedule was as follows:

Month Task

- 1-5 Learned the inspection task.
 Market search for the vision system.
 Purchase of the Genesis 2000 vision system.
- 6-7 Learned the UNIX operating system.
 Learned the programming language C.

Learned BLIX the image processing language.

Learned mathematical morphology.

8-14 Designed the lighting hardware.

Design of software used during development.

Preliminary design of software for the inspections.

Market search and contracting for the design of the
handler

15-19 Exchange the Genesis 2000 for a Genesis 4000 system.

Learned the operation of the new system (MS-DOS, New
programming methods).

Reprogrammed the Genesis 4000 with the development
software as well as the inspection software.

20-26 Received the material movement handler.

Wrote communication software with the handler.

Wrote control software of the whole system.

Designed and built the lighting hardware for the
handler.

Tested the operation of the software on sample
parts.

Redesigned the lighting scheme and the software for
the leads.

Wrote the operator interface software.

27-30 Tested the operation of the system and compared the
results with the manual inspection.

Even though both software and hardware were needed to perform the inspection, the software was an original creation specific for this task. Over 90,000 lines of code were written, where 60,000 lines were inspection specific and 30,000 line were for development and debugging purposes.

Once all the software and hardware was tested in a laboratory environment, the system was turned over to production. By the time this monograph was completed 10,000 parts had been inspected by the machine. These parts were also inspected by the production workers and the results of the two inspection methods were compared.

The machine never passed a single defective part, but did reject about 200 parts that should have passed the inspection. These units were not perfectly good parts though. Most of them had lost their rough surface texture due to extensive handling. The polished finish reflected light in a specular manner and "fooled" the system. All the inspection parameters were chosen to achieve a zero Average Outgoing Quality (AOQ). The 0.2 percent error is preferable to any small percent AOQ because the operators always inspects the rejects and select the good parts.

APPENDIX A

THRESHOLD DATA COMPUTATION

This appendix present a sample data for the computation of the low and high thresholds. Let i be the gray scale of the image, $h_x(i)$ the histogram of the specified area of the picture, $m(i)$ the mean for each value of the histograms, and $s(i)$ the standard deviation. The following table presents a sample data.

Table A.1. Sample data for Threshold Computaion.

i	$m(i)$	$s(i)$	$h_1(i)$	$h_2(i)$	$h_3(i)$	$h_4(i)$	$h_5(i)$	$h_6(i)$	$h_7(i)$	$h_8(i)$	$h_9(i)$
0	0	0	0	0	0	0	0	0	0	0	0
:	:	:	:	:	:	:	:	:	:	:	:
97	0	0	0	0	0	0	0	0	0	0	0
98	0	0	0	2	0	0	0	0	0	0	0
99	0	1	0	3	0	0	0	0	0	0	0
100	1	4	0	14	0	0	0	0	0	0	0
101	3	10	0	31	1	0	0	0	0	1	0
102	8	23	0	72	0	1	0	0	2	4	0
103	18	44	0	135	3	1	0	0	15	13	0

104	23	57	0	176	7	3	0	0	8	14	0
105	36	87	1	269	25	4	0	0	14	11	0
106	87	203	0	622	109	12	0	0	22	22	0
107	99	211	1	647	165	42	0	0	19	19	1
108	175	338	12	1019	381	93	1	3	30	37	3
109	325	585	34	1709	854	202	3	7	55	52	17
110	270	441	60	1249	769	251	7	13	36	24	24
111	499	738	164	2021	1438	571	28	67	53	57	68
112	410	550	178	1454	1233	507	45	92	49	59	78
113	1201	1259	918	3261	3189	1864	285	409	195	239	449
114	879	707	891	1786	2021	1491	384	472	182	217	467
115	1495	897	1780	2385	2796	2504	999	1050	402	477	1068
116	1298	567	1334	1584	2009	2029	1213	1055	498	544	1093
117	1723	569	2271	1620	2109	2486	2044	1616	844	913	1608
118	2544	746	3391	1610	2336	3238	3584	2574	1723	1786	2655
119	1192	364	1544	615	909	1384	1777	1262	952	942	1349
120	2009	665	2495	823	1209	1974	3005	2379	1831	2040	2326
121	2028	750	2319	666	976	1724	2883	2433	2410	2338	2503
122	1674	706	1765	443	712	1244	2210	2049	2325	2296	2026
123	864	398	778	220	333	624	1067	1079	1270	1315	1094
124	1911	1014	1676	420	642	1188	2085	2394	3179	3214	2408
125	861	521	672	201	269	458	815	1090	1658	1546	1048
126	1131	735	796	241	347	581	956	1478	2292	2103	1390
127	707	456	485	169	259	365	572	891	1440	1344	845
128	404	260	179	111	141	204	283	540	785	785	508

129	435	280	271	135	162	218	306	556	848	849	571
130	275	158	198	85	106	159	214	398	513	472	332
131	258	137	208	112	113	157	169	334	455	459	321
132	181	91	130	81	103	100	131	229	316	311	230
133	184	821	147	87	109	112	133	259	306	265	245
134	113	40	79	76	76	63	115	137	159	160	156
135	119	38	112	76	84	96	76	158	168	149	157
136	125	37	124	77	69	108	108	168	166	164	141
137	82	14	74	75	64	64	80	101	89	89	105
138	84	26	58	55	72	55	82	105	108	125	102
139	69	19	74	69	46	36	63	96	77	77	88
140	100	18	86	87	69	113	93	99	113	126	117
141	44	9	35	39	35	41	46	51	63	43	51
142	78	14	71	72	63	90	53	83	93	98	86
143	76	10	72	72	87	66	66	73	96	83	75
144	70	7	55	77	72	76	67	68	76	75	67
145	50	5	51	47	48	51	45	45	63	52	48
146	76	11	58	65	94	78	65	82	83	82	82
147	63	6	58	64	60	62	53	64	66	76	64
148	81	12	67	77	84	100	79	78	101	67	80
149	54	7	53	52	55	64	44	52	56	65	45
150	64	8	68	63	78	61	52	76	63	68	55
151	66	7	66	70	60	59	56	65	73	80	70
152	113	12	108	142	111	104	111	110	118	117	100
153	37	7	53	39	40	40	40	28	28	36	36

To compute the lower threshold we solve for t_1 in the following equation:

$$\sum_{i=0, t_1} s(i) = 10$$

The approximate solution is $t_1 = 102$. To solve for the high threshold t_h we use the following equation:

$$\sum_{i=t_h} s(i) = 10$$

The approximate solution is $t_h = 168$.

APPENDIX B

CHARGED COUPLED DEVICE

For the past few years there has been a great availability of solid-state cameras at reasonable prices. Most of these cameras use a Charge Coupled Device. In this appendix we develop the theory of operation of a CCD circuit then present its applications to imaging systems and finally we give some practical advice to the use of CCD imagers.

B.1 CHARGE COUPLED DEVICE

Although the charge-coupled device shares much the same technological base with the transistor, it is a functional concept that focuses on the manipulation of information rather than an active concept that focuses on the modulation of electric currents. Charge coupling is the collective transfer of all the mobile electric charge stored within a semiconductor storage element to a similar, adjacent storage element by the external manipulation of voltages. The quantity of the stored charge in this mobile "packet" can

vary widely, depending on the applied voltage and on the capacitance of the storage element. The amount of electrical charge in each packet can represent information.

Perhaps the easiest way to visualize the operation of a charge-coupled device is through the use of a mechanical analogy. Imagine a machine consisting of a series of three reciprocating pistons with a crankshaft and connecting rods to drive them (Figure B.1). On top of one or more of the pistons is a fluid. Note that rotating the crankshaft in a clockwise manner causes the fluid to move to the right, whereas rotating the crankshaft in a counterclockwise manner would cause the fluid to move to the left. Since it takes three pistons to repeat the pattern, this arrangement is called a three-phase system. If it is desired to move the fluid in one direction only, a two-phase system can be devised by imposing an asymmetry on the piston design. Regardless of the direction of rotation, the fluid advances to the right.

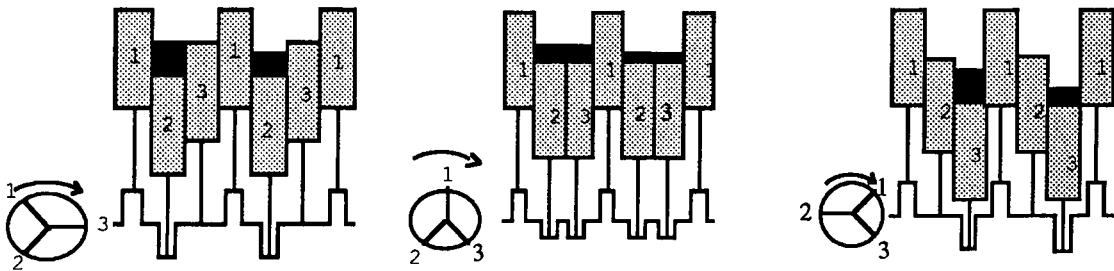


Figure B.1 Mechanical Analogy of CCD device.

Analogous charge-coupled devices can be fabricated of silicon (Figure B.2). The devices consist of a "p-type"

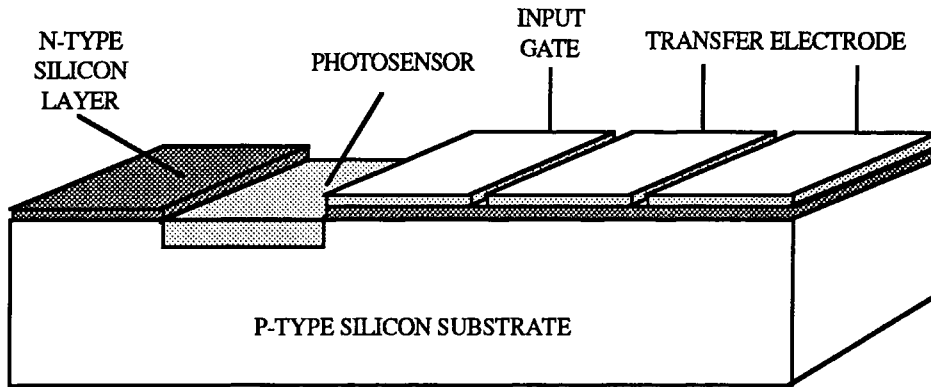


Figure B.2 CCD Circuit.

silicon substrate (in which electrons are normally the signal carriers) with a silicon dioxide insulating layers on its surface. An array of conducting electrodes is deposited in turn on the surface of the insulator. The electrodes can be interconnected to establish either two-phase or three-phase operation. Underlying the insulator and within the bulk of the silicon can be selectively altered to form "n-type" material (in which not electrons but electron "holes" are normally the signal carriers). The correspondence with the machine in the mechanical analogy is realized by supposing that the fluid represents an accumulation of electrons, that the pistons represent the potential energy associated with the voltage applied to the

electrodes and that the crankshaft and connecting rods represent the driving voltages and their relative timing.

When a periodic waveform called a "clock" voltage is applied to the electrodes, some of the electrons in the vicinity of each electrode will form a discrete packet of charge and move one charge-coupled element, or unit cell, to the right for each full clock cycle. The packets of electron charge therefore move to the right as a result of the continuous lateral displacement of the local "potential well" in which they find themselves. They are thus - or so it seems - always falling.

The creation of the necessary potential well in the semiconductor substrate deserves some elaboration because of its central importance to the charge-coupling concept. In this context a potential well is a localized volume in the silicon that is attractive to electrons; in other words, it is the most positive place around and hence is a desirable location from the point of view of the negative electron. Potential wells are formed in a charge-coupled storage element by the interaction of the different conductivity-type regions of the silicon. This interaction forms a well for electrons such that the higher the clock voltage, the deeper the well. Any electrons in the well will move with the clock voltages.

Now, if two or more wells of different depths are placed

close to one another, the wells will overlap and charge may be "coupled," or transferred, from one storage element to the next as the depth of the well is altered by the clock voltage. Thus the external clock voltages on the electrodes cause the electrons to move in packets through the semiconductor in a potential energy trough known as a channel. This mode of electron transfer is the essence of a charge-coupling. Unfortunately not all the electrons advance with the packet on each transfer, and the residual charge appears in a trailing packet. The magnitude of such "charge-transfer efficiency" is a function of the design of the device and the frequency of operation. Transfer efficiency imposes a fundamental limitation on the speed and number of transfers for a practical charge-coupled device because of the resulting attenuation of the charge packet as it is moved through the device from one region to the next. Recent advances in technology, however, have significantly reduced the seriousness of the problem. In general the charge-transfer efficiency must approach one part in 10,000 to be considered acceptable for most practical applications. In spite of this requirement, devices that can be operated at frequencies of up to 100 megahertz are possible if the structures are made small enough. With modern microelectronic manufacturing techniques it is possible to design and build a charge-coupled unit cell with dimensions

of less than 15 microns on a side.

Unit cells of such small dimensions are possible because of the simple nature of the charge-coupled structure, which does not require direct contact with the silicon in the array region. This arrangement is to be contrasted with conventional transistor technology, which in general requires several contacts per functional cell. Contacts consume a significant amount of valuable silicon because of the contact area needed to form a good electrical connection. From the manufacturing viewpoint it is this feature more than any other that makes charge-coupled devices so attractive. We are starting to see on the market CCD arrays of more than five million unit cells packed on a single die.

B.2 CCD IMAGERS

Silicon, the semiconductor material of which charge-coupled devices are generally fabricated, is highly sensitive to visible and near-infrared radiation (Figure B.3). In other words, when light falls on a silicon substrate, the radiation is absorbed, which results in the generation of electrons in a quantity proportional to the amount of incident light. If there is present an array of potential wells such as the one formed by charge-coupled

devices, these electrons will fill the wells to a level corresponding to the amount of light in their vicinity.

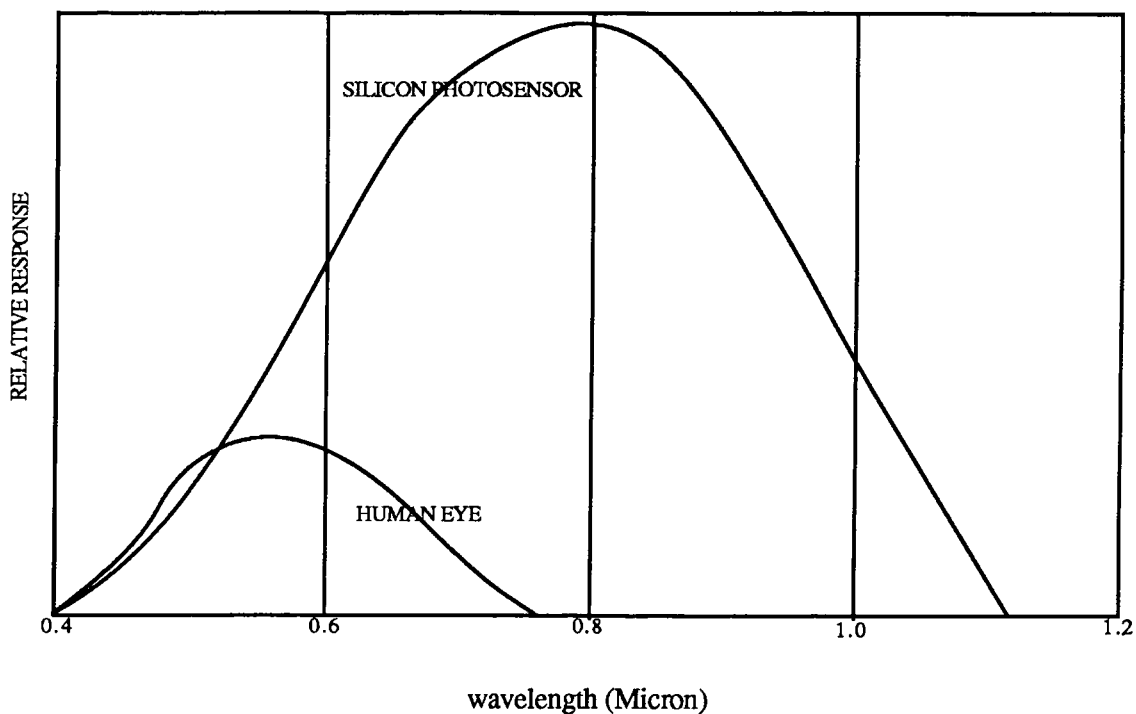


Figure B.3. Relative Spectral Response of a Charged Coupled silicon Photosensor element and the human eye.

This "electro-optic" creation of electrons represents an input to the charge-coupled device. The packet of electrons generated by the light can be moved to a point of detection and converted to an electrical signal representative of the optical image incident on the device.

An area image sensor consists of a photosensor array for accumulating the photocharge pattern plus an associated charge-coupled shift register with one charge-coupled

element for each photosensor element in order to move the resulting charge packet to an output point as shown in Figure B.4. The elements of the photosensor array are individual charge-coupled storage elements with a common electrode called a photogate. They are electrically separated from one another by a highly concentrated p-type region called a channel stop. The photosensor array is separated from the charge-coupled shift register by a region over which there is an electrode called the transfer gate.

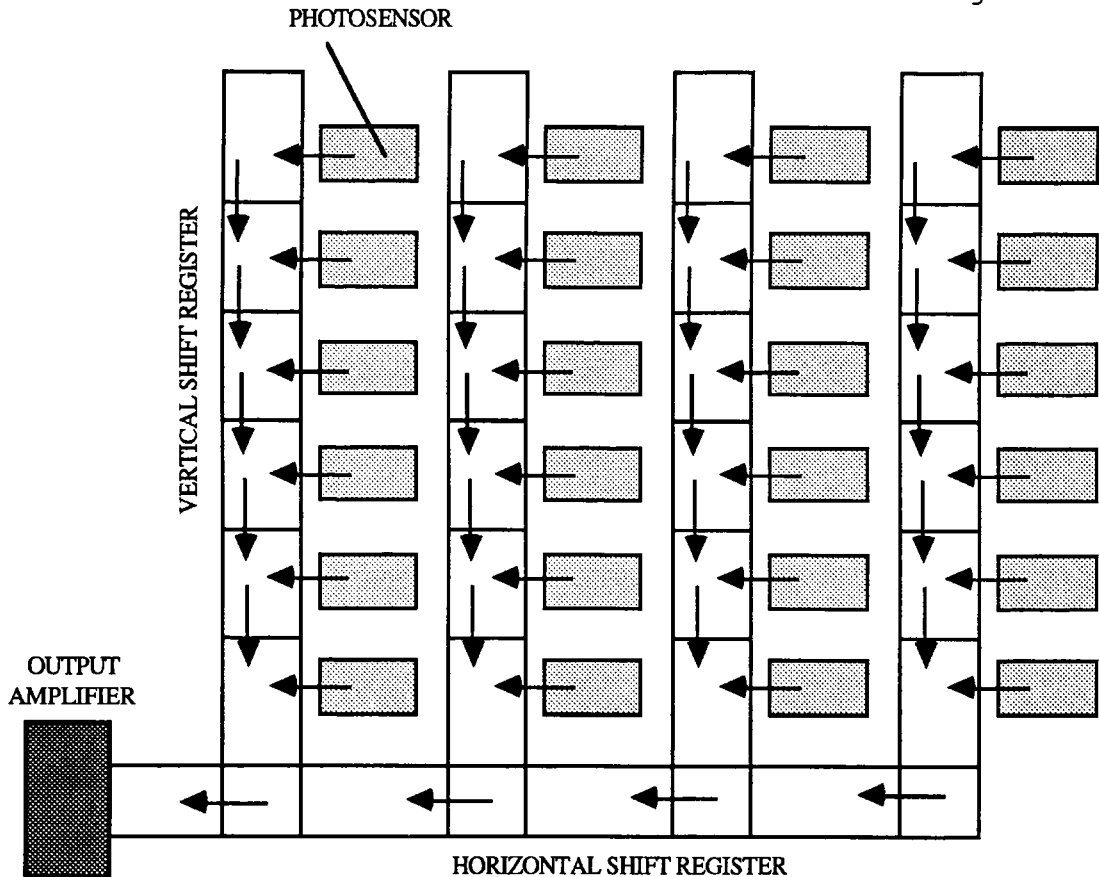


Figure B.4. Interline Transfer Organization of the CCD Area Image Sensor.

In operation the photogate voltage is held high and the charge generated by the incident radiation (the photocharge) is collected by an individual photosensor element. At the end of the integration time (typically $1/60^{\text{th}}$ second) the transfer-gate voltage is raised from its normally low voltage condition. The charge-coupled shift-register electrodes adjacent to the photosensor elements are also brought to a high-voltage state. The photogate voltage is then lowered and the accumulated photocharge transfers to the shift register. After that is accomplished the transfer-gate voltage is lowered and the photogate voltage is brought back to its normally high state for another integration period. Meanwhile the charge-coupled shift register is clocked for the purpose of reading out the charge pattern. That signal, after some conditioning, can be displayed on a cathode ray tube or digitized and stored in a computer memory. In this way a charge-coupled device can become the heart of a video camera.

In the most common area image sensors, the image is obtained by scanning the array mosaic line-by-line. As in standard broadcast television, the image is read out in two separate fields by first reading all the even-numbered photosensor elements in each column and then all the odd numbered photosensor elements in each column.

B.3 PRACTICAL USAGE OF CCD IMAGERS

Some of the significant advantages of charge-coupled image sensors over vacuum-tube sensors are:

1. Precise knowledge of the photosensor locations with respect to one another. In a camera tube the video image is "read" from a photosensitive material by a scanning electron beam. The position of the beam is never precisely known because of the uncertainty in the sweep circuit resulting from random electrical noise. In a charge-coupled sensor the location of the individual photosensor sites is known exactly, since it is determined during the manufacturing of the component. Such metric accuracy is important for doing precise measurements.

2. Low lag and high resistance to image burning. Compared with the lag of conventional cameras which use a pickup tube, the lag of the CCD video camera is considerably reduced, so a clear picture can be obtained when shooting a rapidly moving object or shooting in a place where light intensity is very low. Since resistance to image burning is very high, it is possible to shoot a bright object and to hold the camera on it for a long time. It must be noted that a "smear phenomenon" will occur when shooting a very bright object as described later.

3. Miniaturized and light weight. The adoption of the

CCD image sensor in the video camera module and the development of a small power supply make it possible to produce a very small and lightweight camera.

4. High Resistance to vibration and mechanical shock. If the CCD imager is subject to strong or continuous vibrations and shocks, noise in the picture will be negligible. The use of surface mount devices helps in this endeavour.

5. Long Life and High Stability. The CCD solid state image sensor allows the camera to maintain a high performance level indefinitely. It requires almost no maintenance.

6. Quick Start up. Shooting can be done within a second after turning on the power.

7. High Sensitivity. The sensitivity of the CCD video camera module equals that of a high-sensitivity pickup tube camera (minimum illumination = 3 lux). The CCD video camera can detect images that are close to the infrared range.

8. Low power consumption. Power consumption of the camera module used in this project for example was only 2.3W.

9. Shooting in a strong magnetic field. Since the CCD imager is not influenced by a magnetic field, it is possible to obtain a stable picture even when shooting in an area subject to a strong magnetic field.

When shooting a strong light source or a very bright object, a bright belt is produced above and below the object as shown in Figure B.5, due to the high sensitivity of the CCD imager to the near-infrared range. This phenomenon is

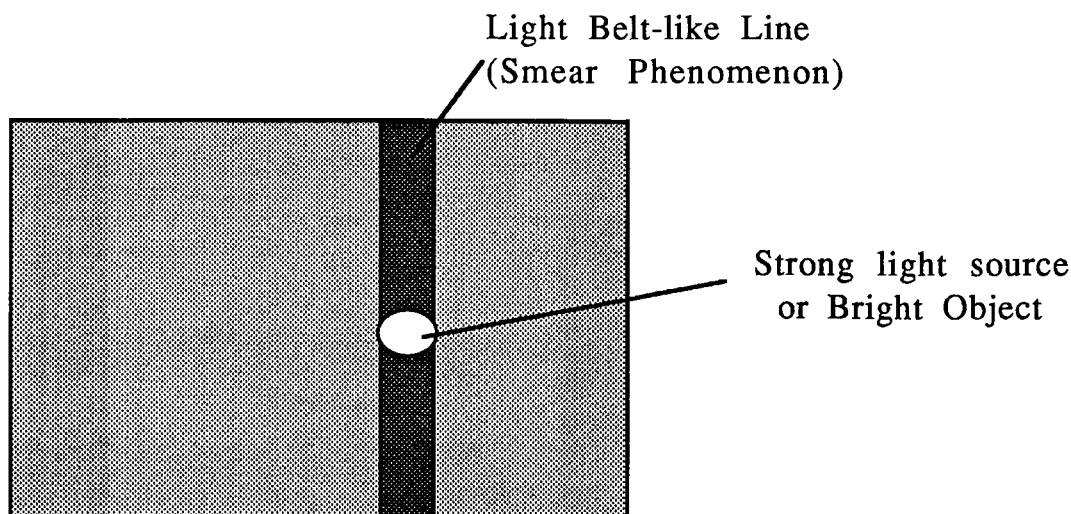


Figure B.5. Smear Phenomenon.

called "smear". Normally, as is described above, the electronic charges generated at the photosensors are fed into the vertical shift registers all at once every $1/60^{\text{th}}$ of a second. However, when a light with a long wavelength, such as an infrared light, enters the CCD imager, it passes through the photosensors and generates a charge underneath them. The generated charge is sometimes fed directly into the register in the vertical blanking interval and this causes the smear phenomenon. To prevent smear, an infrared

cut filter can be attached to the lens. It should be noted, however, that a ghost image may appear on the picture when an infrared cut filter is used.

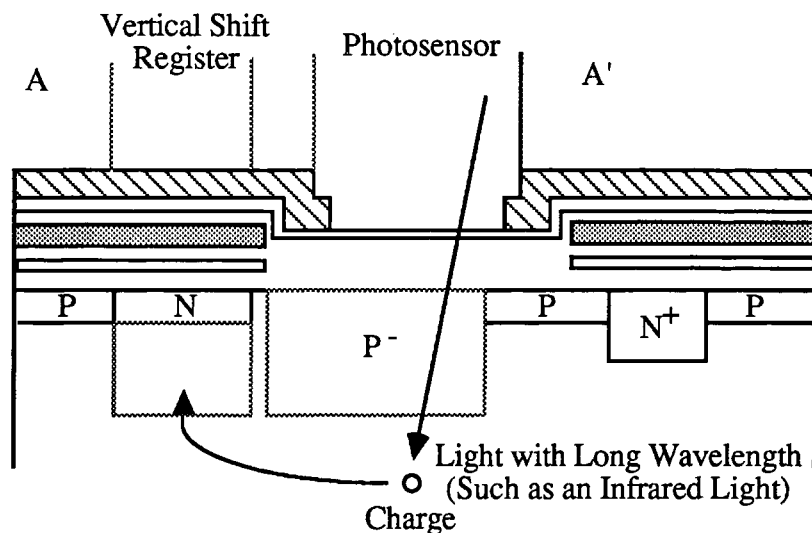


Figure B.6. Cross Sectional View of a Unit Cell.

B.4 POTENTIAL USE OF CCD ARRAYS

The ability to generate, move about, and detect many separate packets of electrons in a small piece of semiconductor material suggests that the charge-coupling principle can be applied to fulfill a number of information-processing requirements. In particular the highly ordered manipulation of charge packets characteristic of the operation of charge-coupled devices favor uses such as

intelligent image sensors, computer-memory operation, and sampled-signal processing. In each case the function is achieved by a proper combination of charge-coupled unit cells that operate individually on each data element. This can give video cameras, or memory chips processing power.

REFERENCES

- [1] O. Friedrich, "The Robot Revolution," *Time*, Vol. 116, No. 23, pp. 72-83, 1980.
- [2] L. Truett and D. Truett, *Managerial Economics*. South-Western Publishing, 1984
- [3]. Matheron, George, *Random Sets and Integral Geometry*, Wiley, 1975.
- [4]. Serra, Jean, *Image Analysis and Mathematical Morphology*, Academic Press, 1982.
- [4] J.L.Mundy and G.B. Porter, "Visual Inspection of Metal Surfaces," *IEEE 1980, CH1499-3/80*.
- [5] H. Berthold, "Obtaining Shade from Shading," *The Psychology of Computer Vision*, McGraw-Hill, New York.
- [6] J. C. Solinsky, "Clever Imaging for Modern Factory," *Photonics Spectra*, Vol. 20, Issue 9, Sept 1986. [7] S.R. Sternberg, "Biomedical Image Processing," *IEEE, COMPUTER*, Jan 1983.
- [8] P. Kendall, "Cellular Logic Arrays for Image Processing," *Handbook of Pattern Recognition and Image Processing*, Academic Press, 1986.