

10-31-1992

## Automatic surface mount solder joints inspection

Zhu Liu

*New Jersey Institute of Technology*

Follow this and additional works at: <https://digitalcommons.njit.edu/theses>



Part of the [Computer Engineering Commons](#)

---

### Recommended Citation

Liu, Zhu, "Automatic surface mount solder joints inspection" (1992). *Theses*. 2315.  
<https://digitalcommons.njit.edu/theses/2315>

This Thesis is brought to you for free and open access by the Electronic Theses and Dissertations at Digital Commons @ NJIT. It has been accepted for inclusion in Theses by an authorized administrator of Digital Commons @ NJIT. For more information, please contact [digitalcommons@njit.edu](mailto:digitalcommons@njit.edu).

## **Copyright Warning & Restrictions**

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be “used for any purpose other than private study, scholarship, or research.” If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of “fair use” that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

**Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation**

Printing note: If you do not wish to print this page, then select “Pages from: first page # to: last page #” on the print dialog screen

The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.

**ABSTRACT**  
**Automatic Surface Mount**  
**Solder Joint Inspection**

by  
**Zhu Liu**

This thesis reports the research results on automatic inspection of solder joints on printed circuit boards. The previous work on this subject has been advanced significantly in the following three aspects. With the support of AT&T Bell Lab, the most updated surface mount solder joints are inspected in this work instead of larger simulation solder joints or traditional through hole solder joints in the previous work. A small set of features is extracted for surface mount solder joints in both infrared and visual light inspection. A new image processing software named Khoros has been applied to improve the quality of images. It has been demonstrated that infrared imaging technique can identify solder joints of surface mount printed circuit boards according to their solder volume. The correct classification rate was found to be in the range of 89% to 100%.

For the sample joints provided by AT&T Bell Laboratory, reasonably good inspection results have been obtained. The experimental results demonstrate that infrared imaging technique can be utilized to discriminate solder joints on surface mount printed circuit boards with different solder volumes quite reliably.



# **AUTOMATIC SURFACE MOUNT SOLDER JOINTS INSPECTION**

by  
**Zhu Liu**

**A Thesis  
Submitted to the Faculty of  
New Jersey Institute of Technology  
in Partial Fulfillment of the Requirements for the Degree of  
Master of Science  
Department of Electrical and Computer Engineering  
October, 1992**



**APPROVAL PAGE**  
**Automatic Surface Mount**  
**Solder Joint Inspection**

**by**  
**Zhu Liu**

---

**Dr. Yun Q. Shi, Thesis Adviser**  
**Assistant Professor**  
**Department of Electrical and Computer Engineering, NJIT**

---

**Dr. Edwin Hou, Committee Member**  
**Assistant professor**  
**Department of Electrical and Computer Engineering, NJIT**

---

**Dr. Walter F. Kosonocky, Committee Member**  
**Distinguished Professor**  
**Department of Electrical and Computer Engineering, NJIT**

# **BIOGRAPHICAL SKETCH**

**Author:** Zhu Liu

**Degree:** Master of Science in Electrical and Computer Engineering

**Date:** October, 1992

## **Undergraduate and Graduate Education:**

Master of Science in Chemical Engineering  
New Jersey Institute of Technology, Newark, NJ, 1992

Bachelor of Science in Electrical Engineering,  
Zhejiang University, Hangzhou, Zhejiang Province  
People's Republic of China, 1988

**Major:** Computer Engineering

This thesis is dedicated to  
Yaling Zhou

## ACKNOWLEDGEMENT

The author wishes to express his gratitude to the many people who have helped and contributed significantly, in the bringing about of this thesis. In particular, the author would like to thank Dr. Yun Q. Shi, the thesis advisor, for his ever present support and follow up, both technically, and interpersonally. The author is also thankful to Dr. Walter F. Kosonocky, for his valuable advice and assistance, in completing this thesis.

Special thanks to Dr. Frank P. Higgins and many other researchers at the AT&T Bell Laboratory for their significant assistance. This work could not be completed without their help.

And finally, the author would like to extend his deep appreciation to his parents, whose care and encouragement have always been a major inspiration for the author.

# TABLE OF CONTENTS

	Page
1 INTRODUCTION .....	1
1.1 Automatic Solder Joint Inspection Survey.....	1
1.2 Previous Work of This Project .....	4
1.2.1 The Small Scale Image Processing System .....	4
1.2.2 Visual Light Inspection Experiment .....	6
1.2.3 Infrared Inspection Experiment .....	7
1.2.4 Other Valuable Works .....	8
2 SURFACE MOUNT TECHNIQUE.....	9
2.1 Introduction .....	9
2.2 Technical Procedure .....	11
2.2.1 Pasting .....	12
2.2.2 Reflowing .....	12
3 THE IMAGE PROCESSING SYSTEM.....	14
3.1 Thermal Imaging Theory.....	14
3.1.1 Introduction .....	14
3.1.2 Basic Concepts .....	14
3.2 The CCD Infrared Camera .....	15
3.3 The Infrared Inspection System Configuration .....	18
4 STATISTICAL PATTERN RECOGNITION.....	22

4.1 Pattern Recognition for Solder Joint Inspection .....	22
4.1.1 Basic Concepts of Statistical Pattern Recognition.....	22
4.2 the Sunview Window Tool for Solder Joint Inspection .....	25
4.2.1 Solder Joint Segmentation: .....	26
4.2.2 Mean Features Learning Or Training:.....	28
4.2.3 Classification: .....	29
4.2.4 Classification Of Fusing Visual Light And Infrared Images.....	31
5 EXPERIMENT AND RESULTS.....	33
5.1 The Sample Surface Mount Circuit Board.....	33
5.2 The Experiment Procedure .....	35
5.2.1 Heating Procedure .....	35
5.2.2 Taking Images.....	36
5.2.3 Image Processing.....	37
5.2.4 Pattern Recognition .....	39
5.3 Experiment Results .....	41
5.3.1 Solder Volume Inspection.....	41
5.3.2 Electrical Connectivity Inspection .....	46
6 VISIBLE LIGHT INSPECTION EXPERIMENT.....	47
6.1 Visual Light Inspection System .....	47
6.2 Illumination Method .....	48
6.3 The Experiment Procedure.....	49
6.4 Experiment Result .....	51
7 CONCLUSIONS AND RECOMMENDATION.....	54



APPENDIX.....	56
REFERENCES .....	119

## LIST OF TABLES

<b>Table</b>	<b>Page</b>
1 Pattern Recognition Results For 50-mil J-lead ICs.....	44
2 Pattern Recognition Results For 25-mil Gull-wing ICs.....	44
3 Pattern Recognition Results For 50-mil Gull-wing ICs .....	45
4 Pattern Recognition Results For 50-mil J-lead ICs.....	52
5 Pattern Recognition Results For 50-mil Gull-wing ICs.....	52
6 Pattern Recognition Results For 25-mil Gull-wing ICs .....	56

# LIST OF FIGURES

<b>Figure</b>	<b>Page</b>
1.1 Structure Light Method for 3-D Surface Reconstruction.....	3
1.2.1 The Original System .....	5
1.2.2 The System That Huang Used.....	6
2.1 Gull-wing-lead Surface Mount Solder Joints .....	10
2.2 J-lead Surface Mount Solder Joint.....	11
2.3 50-mil Surface Mount IC .....	12
3.1 The IR-CCD Infrared Camera .....	16
3.2 Infrared Inspection System .....	18
3.3 The Datacube Image Processing System.....	19
4.1 Pattern in Euclidean Space.....	25
4.2 Solder Joint Inspection Tool Window.....	27
5.1 The Experimental Circuit Board.....	34
5.2 Plate Heating Procedure .....	36
5.3 Histogram Modification.....	39
5.4 Name of the Selected Experiment Region .....	42
5.5 the Infrared Image of IC16.1:i16.1.....	43
6.1 Visual Light Inspection System.....	47
6.2.1 Top View of the Illuminating Position.....	48
6.2.2a Reflected Light on Normal Solder Joint Surface.....	49
6.2.2b Reflected Light on Insufficient Solder Joint Surface .....	50

# **Chapter 1**

## **INTRODUCTION**

This thesis is a part of a long term project on the application of machine vision technique, which is one of the tasks of the Machine Vision Laboratory, Electronic Imaging Center, under general direction of Dr. Walter Kosonocky, ECE Dept. of NJIT.

The research work described in this thesis is automatic inspection of solder joints on surface mount printed circuit board, using a small scale image processing system and unsupervised statistical pattern recognition technology. The purpose of this research is to research the potential of applying infrared imaging technology in the automated inspection of advanced solder joints--surface mount solder joints.

### **1.1 Automatic Solder Joint Inspection Survey**

In most mass-production manufacturing facilities, an attempt is often made to achieve 100% quality assurance of all parts, subassemblies, and finished products. One of the most efficient and difficult task in this process is that of automated machine inspection -- an inspection that seeks to identify both functional and cosmetic defects automatically done by the machine itself. Undoubtedly, the automation of inspection will increase productivity and improve product quality.

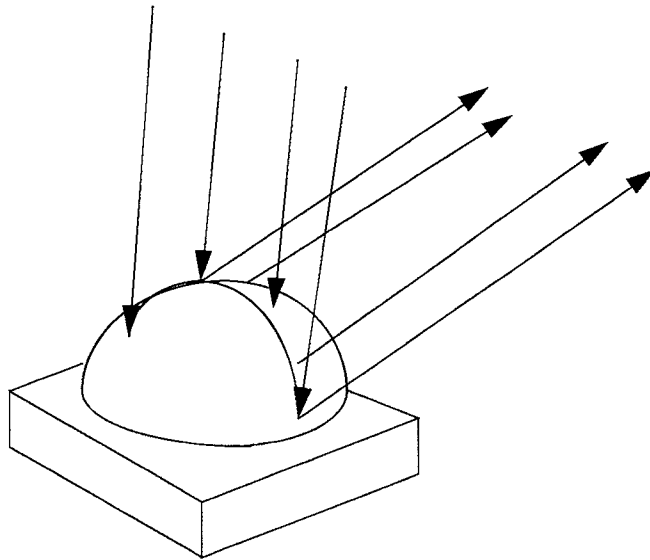
The automated inspection process involves observing the same type of object repeatedly to detect anomalies. The process starts with imaging the object to be inspected by a sensors (or sensors) from which visual data are

collected and sent to the processor for analysis. Features representing the object are then extracted and matched to a predefined model. This feature-to-model matching process is the most common technique for detecting defects. It is realized by finding features in the given object that match the model's definition of defects, or by verifying all extracted features in the given object to be normal, expected features as those defined by the model. After the detection, decisions are made to classify the object into a defect type.

Advances in computer technology, sensing devices, image processing, and pattern recognition have resulted in better and more inexpensive industrial visible light inspection equipment. Automated inspection for manufacturing is now of broad and rapidly expanding commercial use. The electronics industry is the most active one in applying automated inspection to such products as printed circuit boards, integrated circuit chips, photomasks, and hybrid circuits. This industry has been described as a \$9 billion business which scraps \$3 billion each year. The losses can be attributed to several causes, including the defective raw circuit boards, improperly mounted components and poor quality solder joints. Although a great deal of work has been done in the area of printed circuit board inspection, very little research has addressed the inspection of solder joints(10).

Unlike the printed circuit board patterns which are planar in nature, solder joints are volumetric objects. Therefore, those methodologies applied in printed circuit board inspection is not adequate for the three-dimensional solder joints. In most of the proposed methods, properties of the surface geometry are used as the criteria, since they are based on the visible light inspection. In one feasibility study, solder joint features such as those associated with brightness, size, volume, surface area, and surface curvature, extracted from the gray-level intensity image, are used in a statistical decision

rule to classify joints into various types of defects. Another system is based on a structured lighting approach to obtain shape information. Stripes of light are generated and focused onto the joint's surface. The surface reflectance containing range information is detected, strip by strip, for the construction of the surface, as illustrated in Figure 1.1. Defects are detected by analyzing the sequential signal.



**Figure 1.1** Structured Light Method for 3-D Surface Reconstruction

The electrical conductivity and the mechanical strength are the two functions of a solder joint of most concern. However, no matter what kind of method is used in visible light solder joint inspection, the system can only predict the quality of a solder joint by information associated with the solder joint surface. Some other methods such as X-ray and infrared detection have been developed during the past ten years for observing the internal nature of the solder joints.

Infrared inspection of solder joints has some valuable advantage compared with visible light inspection. Because of the similar nature of

thermal conductivity and electrical conductivity, a solder joint has good thermal conductivity usually has good electrical conductor as well. Therefore, an infrared image of a solder joint showing heat conduction characteristics is useful for inspection of the integrity of solder joints. On the other hand, infrared images may also identify some internal solder defects which may have no vestige on the solder surface, and makes it possible to inspect those solder joints which can not be seen directly by the visible light, such as J-lead surface mount joints.

## **1.2 Previous Work of This Project**

Two fellow students, Mr K. K. Zhou (5) and Mr. Jiang Huang (13) did this research and finished their master thesis in May 1990 and May 1991 respectively. A brief description of their work is given below for reference.

### **1.2.1 The Small Scale Image Processing System**

Beginning from the fall of 1988, several professors at the Electronic Imaging Center of Electrical and Computer Engineering Department, NJIT, initiated a new image processing system. Because of the advances in the computer industry, such a system may only cost tens of thousand dollars, however, it can be expended to fairly sophisticated high-speed, real-time system in the future.

The basic function of such a small scale system are:

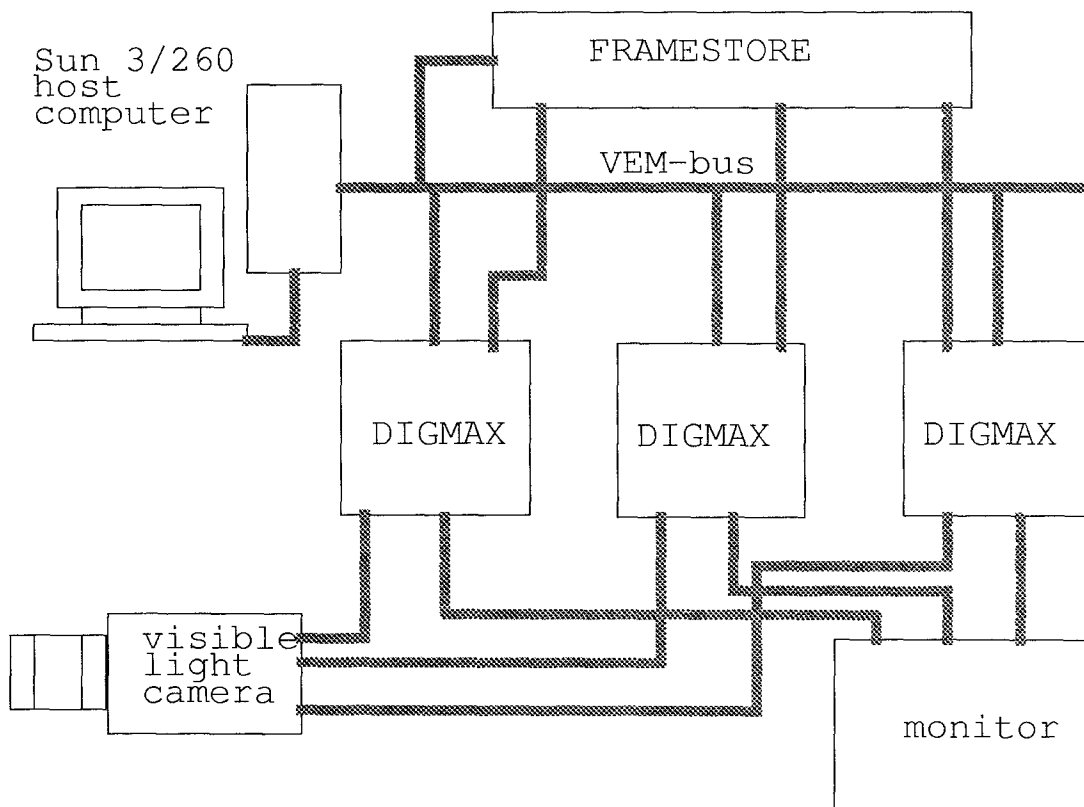
1. digital image frame acquisition from camera;
2. display digital image on the screen;
3. processing and storing the digital image in the host computer;
4. conduction theoretical research and developing applied software

utility on the system.

Zhou's work involved installing and testing five Datacube circuit boards, mounting operating software and modifying the kernel of SunOS operating system. The whole setting task was completed by June 1989. Another set of image I/O operating subroutines were installed around August 1989.

The original system is given in Figure 1.2.1. After its first set up, the system modified several times according to various application. One of later hardware configuration used by Jiang Huang is shown in Figure 1.2.2.

Now the system has been upgraded by Dr. Hou. A new Sun SPARC 300

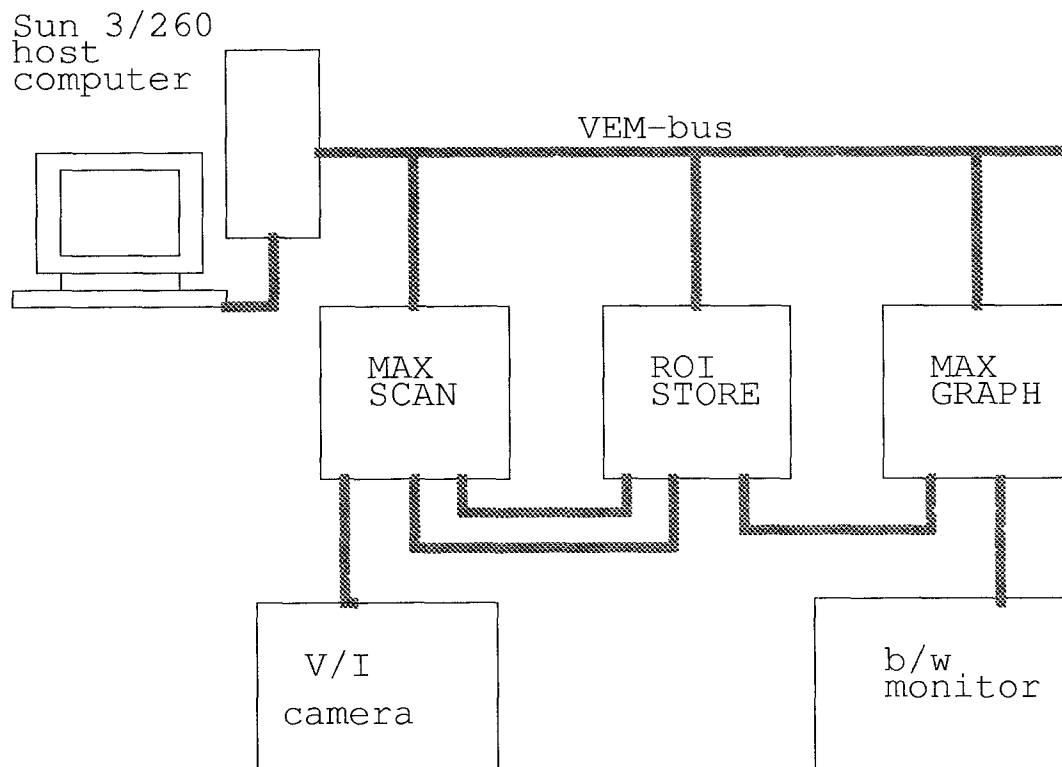


**Figure 1.2.1** The Original System

station has replaced the obsolete Sun 3/260 system, and more hardware and



software has been added. For more details about the image processing system we used in this project, please see section 3 of Chapter 3.



**Figure 1.2.2** The System used by Huang

### 1.2.2 Visible Light Inspection Experiment

Zhou began his experiment at a traditional printed circuit board with hand-made, through hole type solder joints on it. The joints on the printed circuit board were of real size for dual in-line integrated circuit chip socket. Four fundamental types of solder joints were made. They were excess solder, normal, insufficient solder and void. A visible CCD camera was used to take the visible light image of the solder joints. The original lens had a focal length of 16 mm and a minimum photo distance of 0.3 m. Then a microlens with a focus of 75

mm and a minimum photo distance of 1 meter was added. The size of solder joint subimage on the picture was 10 pixels in X direction and 14 pixels on Y direction. His best results were that the good/bad rate achieved 78.7% (85 from the total 108 joints), and the correct classified rate (the rate of correctly classifying the solder joints into one of the four categories mentioned above) was 62.96% (68 from the total 108 joints).

Jiang Huang did his work on the same circuit board, but improved the experimental results by both enlarging the solder joints more than 6 times and modifying the illuminating condition. Huang used an extra microlens to enlarge the solder joint subimage to more than 18 pixels in both X and Y direction. A circular fluorescent lamp was used for illuminating instead of projecting illumination. Besides, some pre-processes on the circuit board such as removing the rosin by alcohol were also taken by Huang. The best results claimed to be improved to that the good/bad rate was 100% (50 from total 50 joints) and the correct classified rate was 98% (49 from total 50 joints).

### **1.2.3 Infrared Inspection Experiment**

Because of the insufficient resolution of our CCD camera, Zhou did his infrared inspection on an simulating circuit board for experimental purpose. He made a experimental circuit board with 20 enlarged, artificial simulating solder joints on it, each in diameter of 15mm. Four different quality classes were included, each with 5 joints. The best results achieved were that the good/bad rate was 85% and the correct classified rate was 80%, with solder joint subimage size of 34x46 pixels.

The results of Jiang Huang represented a significant advancement in the experimental technique for infrared inspection. First, he successfully found an approach to enlarge the infrared image, making it possible for the

infrared camera to obtain an image of the solder joint with an acceptable resolution. More importantly, he introduced a new set of feature for the statistical pattern recognition process. He also developed some useful image processing technique for the infrared image enhancement, such as interpolation and histogram equalization. These efforts were fruitful. The best good/bad rate achieved 96% (48 from total 50 joints), and correct classified rate, 94% ( 47 from total 50 joints).

#### **1.2.4 Other Valuable Works**

The fusion inspection of visible and infrared images was investigated by both K. K. Zhou and Jiang Huang to see if it can achieve better improvement.

The feature distance of this method is from the weighted sum of the infrared inspection feature distance and the visible light inspection feature distance. By applying the minimum-distance principle, the tested pattern is classified to one of the four classes.

Both of their experiments show that the fusion of infrared inspection and visible light inspection can give out better results than that of using only either infrared inspection or visible light inspection. The improvement of the result depends on the selection of the weighting index. If the weighting index are chose properly, in both of their experiments, both the good/bad rate and the correct classified rate can achieve 100%.

## **Chapter 2**

# **SURFACE MOUNT TECHNIQUE**

Surface mount is not a new soldering technique in the industry. It has been applied in both single component and integrate circuit products many years ago. Recently, the application of this soldering technique grows very fast. It is reported that more than 50% of the solder joints in industry are surface mount solder joints. In this chapter, a brief introduction about what surface mount soldering is and its technical procedure is given.

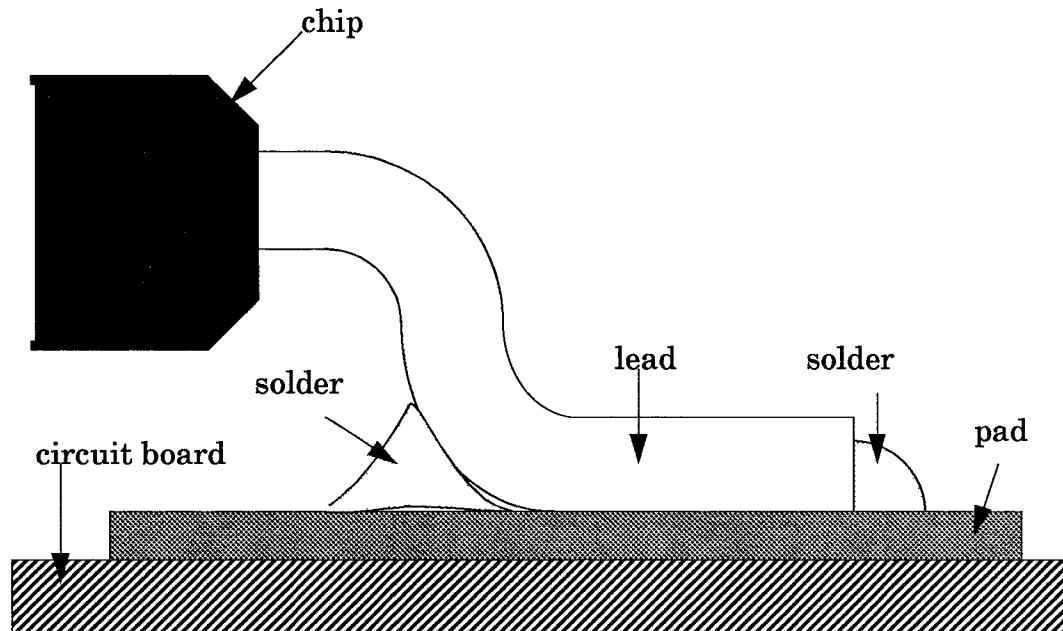
### **2.1 Introduction**

Because of the advances of the integrated circuit technology, the integrated scale of ICs on the market becomes larger and larger. The number of leads in one chip tends to be larger and larger. To reduce the size of chips, the size of each lead and the distance between leads must be reduced.

The traditional through hole soldering technique is becoming obsolete for this purpose, because it is very difficult to drill high density holes on printed circuit board, and the soldering process of through hole joints is too complicated. Through hole soldering has the advantage of high mechanical strength. However, as the number of joints in one chips increases, the required mechanical strength of each joint decreases significantly.

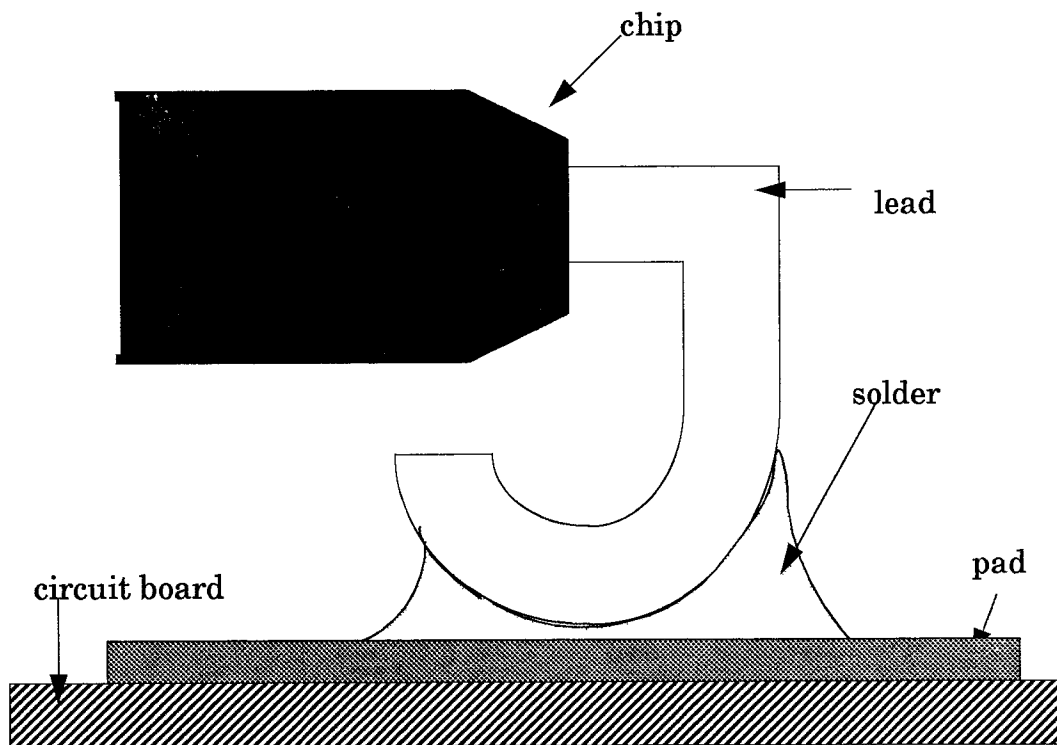
Surface mount soldering results from the needs of high-density and relatively low-strength. The word *surface mount* means that the integrated circuit chips are mounted on the surface of the circuit board, rather than drill

through holes on the circuit board and insert the chip leads through the holes. The chip leads are soldered directly on the conducting pads of the circuit board. Figure 2.1 and Figure 2.2 illustrated the two types of typical surface mount technique which are widely used in the industry.



**Figure 2.1** Gull-wing-lead Surface Mount Solder Joint

Surface mount technique has the advantage of saving space. The size of a surface mount solder joint is normally specified by the distance between two adjacent leads, as illustrated in Figure 2.3. The typical size of surface mount chips on the current market is 50 mil and 25 mil, i.e., one mil is a thousandth of one inch. Smaller size surface mount chips such as 20 mil and even 15 mil have already been made in some research institute, and will be put into commercial market in the near future. Recently, more than 50% of the through hole soldering market has been replaced by surface mount soldering. It is expected that surface mount will substitute the traditional through hole technique as the dominant soldering technique in the printed circuit board manufacturing industry.

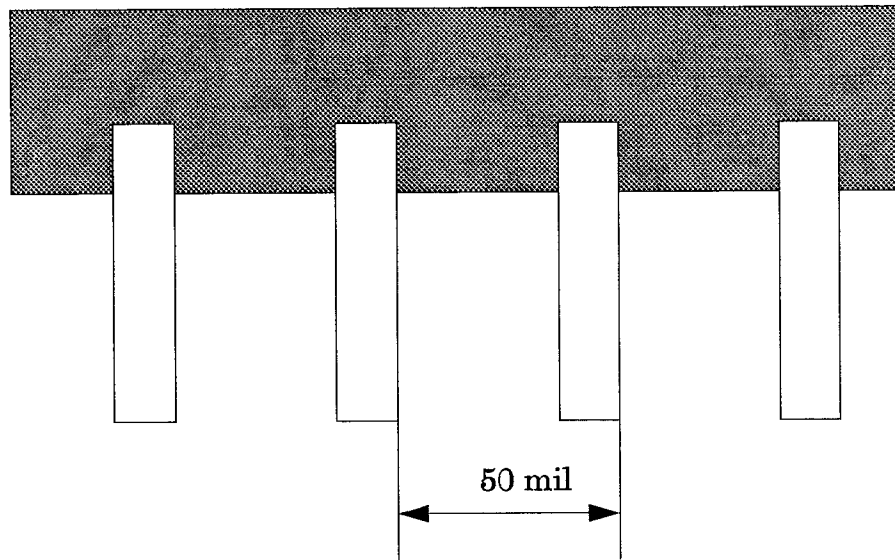


**Figure 2.2** J-lead Surface Mount Solder Joint

There are two types of surface mount leads widely used in the industry. One is called Gull-wing, as illustrated in Figure 2.1, the other is called J-lead, as illustrated in Figure 2.2. Compared with the J-lead, the Gull-wing solder joints have the advantage of easy to solder, inspect and desolder. On the other hand, the J-lead joints can use less space than Gull-wing. Because of the above advantages of the Gull-wing, it is becoming more popular than the J-lead solder joints. The smallest J-lead type joints is in 50-mil size. All of the large-scale, high density surface mount chips such as 20 mil and 15 mil are expected to implemented only with Gull-wing type leads.

## 2.2 Technical Procedure

The soldering procedure of a surface mount solder joint can be divided into two



**Figure 2.3** 50 mil surface mount IC.

procedure: pasting and reflow.

### **2.2.1 Pasting**

Before solder is heated, a stencil is placed on the circuit board. A special paste material, which is a mixture of very small solder grains, soldering-assistant material and special glue, is pasted on the pads which is going to be soldered through the holes of the stencil. The chip is then gently pressed on the paste and is stuck there. Then, the circuit board with paste and chips stuck on it is transferred to an infrared oven.

### **2.2.2 Reflowing**

Inside the infrared oven, the circuit board is passed slowly through a hot air knife created by infrared radiation. When paste is passing the hot air knife, the solder grains and the solder layer on the pads and leads are reflowed again. A new solder joint forms when the reflowed joint is cooled.



# **Chapter 3**

## **THE IMAGE**

### **PROCESSING SYSTEM**

In this chapter, the image processing system we use in our research is introduced, including the CCD infrared camera and the Datacube image processing system.

### **3.1 Thermal Imaging Theory**

#### **3.1.1 Introduction**

Thermal imaging is a technique for converting a scene's thermal radiation pattern, which is invisible to the human eye, into a visible image. Development in the field of thermal imaging have been rapid in the past few years. The current availability of commercial systems has enabled the application of thermal imaging techniques to be carried into a wide range of the areas.

#### **3.1.2 Basic Concepts**

All objects are continually emitting radiation at a rate and with a wavelength distribution that depends upon the temperature of the object and its spectral emissivity,  $\varepsilon(\lambda)$ .

A black body is an object that absorbs all incident radiation and, conversely, is a perfect radiator. The energy emitted by a black body is the maximum theoretically possible for a given temperature. The radiative power ( or number of photons emitted ) and its wavelength distribution are given by the Planck radiation law:

$$W(\lambda, T) = \frac{2\pi hc^2}{\lambda^3} \left[ \exp\left(\frac{hc}{\lambda kT}\right) - 1 \right]^{-1} \text{Wcm}^{-2}\mu\text{m}^{-1}$$

$$P(\lambda, T) = \frac{2\pi c}{\lambda^4} \left[ \exp\left(\frac{hc}{\lambda kT}\right) - 1 \right]^{-1} \text{photons} \cdot \text{s}^{-1} \cdot \text{cm}^{-2} \cdot \mu\text{m}^{-1}$$

where  $\lambda$  is the wavelength,  $T$  is the temperature,  $h$  is Planck's constant,  $c$  is the velocity of light and  $k$  is Boltzmann's constant.

As the temperature increases, the amount of energy emitted in anywhere increases, and the wavelength of peak emission decreases. The latter is given by Wien's displacement law:

$$\lambda_{mw} T = 2898 \mu\text{m} \cdot \text{K}$$

for maximum watts, or

$$\lambda_{mp} T = 3670 (\mu\text{m} \cdot \text{K})$$

for maximum photons.

### 3.2 The CCD Infrared Camera

The infrared camera used in this project was built by a group of graduate students of the E.E Department, NJIT, at the David Sarnoff Research Center (DSRC), in Princeton, New Jersey, under the direction of Professor Walter F. Kosonocky, NJIT, Chairman of the Electronic Imaging Center.

The camera is composed by two main section, the head section ( lens

and sensors) and the electronic box (A/D convertor for 12-bit digital output port and D/A convertor for RS-170 display). They are shown in Figure 3.1. The left-hand-side unit is the power supply and primary signal processing unit, while the right-hand-side unit is the head unit.



**Figure 3.1** The IR-CCD Infrared Camera

The camera operates with a 100-mm  $f/1.4$  lens, a  $f/1.3$  cold shield and a long-pass filter with a  $3.4 \mu\text{m}$  cut-off frequency. The characteristic of this IR-CCD camera are summarized below:

#### FPA STRUCTURE

- Number of Pixels:  $320 (H) \times 244 (V)$
- Pixel Size:  $40\mu\text{m} (H) \times 40\mu\text{m} (V)$

--Chip Size:  $40\mu\text{m (H)} \times 40\mu\text{m (V)}$   
 --Imager Architecture: Monolithic Silicon ITCCD  
 --IR Sensor Element: PtSi Schottky-barrier Detector

#### CAMERA ELECTRONICS

--Frame Rate: 30 frames/s with 2 vertically  
interlaced 320 x 122 fields/frame  
 --Horizontal Line Readout:  
63.5  $\mu\text{s/HL}$  with imager signal on  
alternate horizontal TV lines  
 --Uniformity Correction:  
one-point of-set-type averaging  
up to 16 reference frames

#### OPTICS

--Lens: 100 mm, f/1.4  
 --Cold Shield: f/1.3  
 --Infrared Filter: 3.4  $\mu\text{m}$  long pass

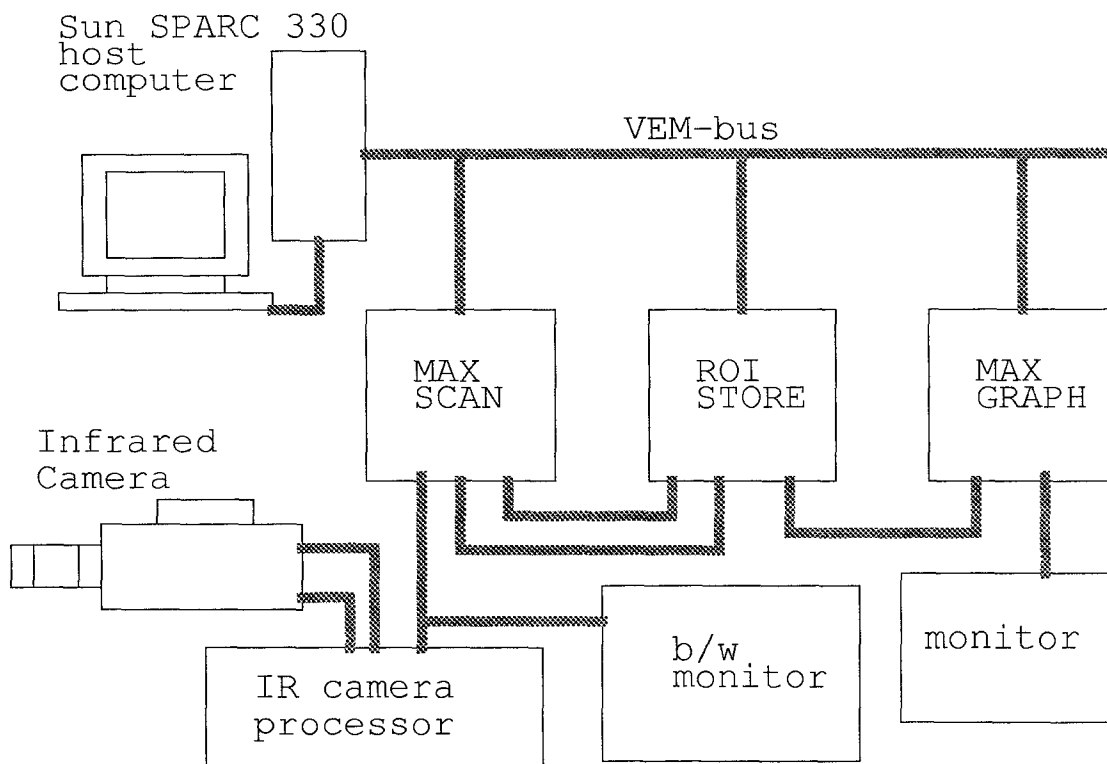
#### SYSTEM PERFORMANCE

--Operating Temperature: 77K  
 --Imager Responsivity: 57.9 mV/K (FPA output)  
34.5 mv/K (camera output)  
 --NE $\Delta$ T at FPA output: 0.034K  
 --NE $\Delta$ T at Camera Output: 0.0579K  
 --Thermal Noise:  
FPA output: 2rms mV, 772elec./pixel

camera output: 2.544rms mV,982elec./pixel

### 3.3 The Infrared Inspection System Configuration

The system diagram is shown in Figure 3.2. A Datacube Image System connects the IR-CCD camera to the VME bus of the host computer. The host computer is a Sun SPARC station 330. SPARC station is a new generation of the Sun workstations series. It implement a new type of CPU with the Reduced Instruction Set technology, which greatly improve the CPU speed.



**Figure 3.2** Infrared Inspection System

The Datacube Image System is a real time image processing system. Figure 3.3 shows the outlook of the Datacube system.

The MaxVideo modules, made by Datacube Incorporation, are used to



**Figure 3.3** The Datacube Image Processing System

construct a special image processor to perform some function like video data acquisition and image displaying. Each module presents some individual function of image processing. They can work compatibly with each other to construct different systems have their own purpose and performance, also for particular application requirement and investing scope.

In the Datacube System, there are different operating functions for every MaxVideo modules. Only three of them are used in our work. They are MAX-SCAN, ROI-STORE, and MAX-GRAPH. The MaxVideo modules group uses an open modular structure to provide the real time image manipulating, acquiring, storing and processing functions. These are briefly described below.

(1) MAX-SCAN is a complex, programmable asynchronous input module which can acquire either analog or digital input data with a variety of resolution and frame rate. Only its analog I/O port (P12) which can operate at a data rate up to 10 MHz is used. MAX-SCAN can digitalize the output of low resolution, fast frame rate sensors, or high resolution, slow frame rate sensors. It creates a 10 MHz stream of scan sequential digital video data, compatible to a wide range of variety of the MaxVideo modules. The digitized video signal is download from the MAXbus Digital Output (P5) to the Digital Input (P9) of ROI-STORE in our construction.

(2) ROI-STORE is a region-of-interest (ROI) memory and a MaxVideo compatible framestore module which supports user programmable video resolutions and processes the region-of interest image. ROI-STORE accommodates video signal from RS-170 and non-RS-170 MAXbus sources, and transfers data in and out of its MAXbus ports at standard MAXbus rate(10MHz) or 7.16 MHz. ROI-STORE module can store images or create regions of images in its memory, with any horizontal and vertical resolution up to 4K. The P4 of ROI-STORE connected with P4 of MAX-SCAN is for the ROI timing.

(3) MAX-GRAPH is a graphics generation module. It displays graphics in RS-170 format as well as in non-interlace format. It also supports mouse and light pen interfaces and RGB color output, and is compatible with 10 MHz digital video data streams.(for  $512 \times 512$  systems) In our system, the MAX-

GRAPH accepts signal from the Digital Output (p5) of ROI-STORE, and output analog signal at the Analog Output (p12) to a TV monitor.

In the small system, the VMEbus is used as the data channel between MaxVideo modules and the host computer. B bus adapter and cable are connected to the VMEbus of host computer, as shown in Figure 3.1.

The functions of the Datacube Image Processing System are:

1. displaying the thermal images from the IR-CCD camera on a TV monitor;
2. freezing the real time camera image;
3. storing and loading the digital image into the host computer Sun SPARC station 330.

Driving programs for individual module offered by Datacube Incorporation can be mounted into the kernel of operating system of the Sun workstation. The operating of modules could be put into a standard library for the convenience of users. If a user want to do a module operation, the only thing he has to do is call the appropriate procedure in the library.



# **Chapter 4**

## **STATISTICAL PATTERN RECOGNITION**

The main software work of our project is to develop a software to perform the classification of solder joint images. In this chapter, some basic concepts of the statistical pattern recognition is introduced, and brief description and manual of the software we developed for the pattern recognition work needed in our project, namely the Solder Joint Inspection Sunview Window Tool, is given.

### **4.1 Pattern Recognition for Solder Joint Inspection**

#### **4.1.1 Basic Concepts of Statistical Pattern Recognition**

Statistical pattern recognition method is utilized in this work for automatic solder joints inspection. A computer program is written in C language to specifically implement the pattern recognition.

Recognition is regarded as a basic attribute of human being. A pattern is the description of an object. Pattern recognition means classifying a class of objects from any other classes by computer systems instead of human beings. In statistical pattern recognition a pattern is described by a series of mathematics quantities, termed as feature vector, or simply, features. An object or a pattern could be distinguished from others only if it has distinct features from the others. In this thesis only the photographed object is concerned . In other words, the visible light and infrared phonographs of solder joints on PCB are needed to be recognized.

Only the concept of pattern classification by distance functions, the

simplest and most intuitive approach to the statistical pattern classification, is discussed here briefly. Prior to a more quantitative discussing some notations are given next.

### Symbol Explanation

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} \quad \text{vector; using lowercase bold}$$

$x_j$           subscripted vector

$$\|x\| = \sqrt{\sum_{j=1}^n x_j^2} \quad \text{Euclidean norm or magnitude of vector } x$$

$n$               dimensionality of pattern vectors

$E^n$             n-dimensional Euclidean space

$w_i$             ith pattern class

$m_i$             mean vector of class  $w_i$

$I$                matrix; using uppercase bold

Consider the simplest situation where only two classes,  $w_i$  and  $w_j$  are concerned. Furthermore, each is described by a 2-dimensional feature vector. These two feature vectors can be positioned in a 2-dimensional Euclidean space, i.e., a planed as shown in Figure 2.3. Every point with the two contours is a feature vector, which is computed from certain object belonging to  $w_i$  or  $w_j$ . Assuming that

$$o_1, o_2, \dots, o_s \in w_i$$

$$p_1, p_2, \dots, p_t \in w_j$$

The mean vectors  $m_i = \{m_{1i}, m_{2i}\}$  and  $m_j = \{m_{1j}, m_{2j}\}$  can be computed as

$$m_{1j} = \frac{1}{t} \sum_{k_1=1}^s p_{k1}$$

$$m_{2j} = \frac{1}{t} \sum_{k_2=1}^s p_{k2}$$

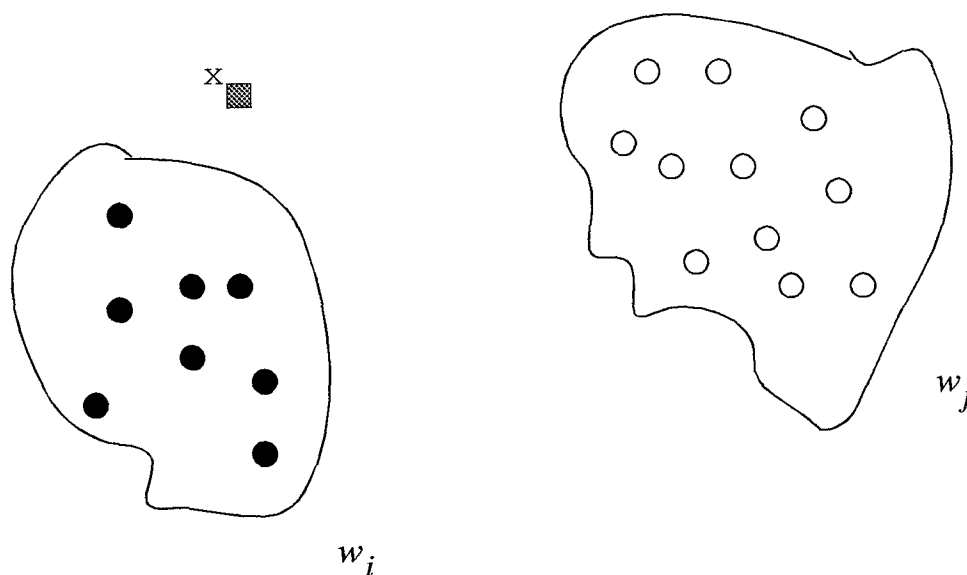
$$m_{1i} = \frac{1}{s} \sum_{k_1=1}^s o_{k1}$$

$$m_{2i} = \frac{1}{s} \sum_{k_2=1}^s o_{k2}$$

A distance between two vectors  $a = \{a_1, a_2\}$  and  $b = \{b_1, b_2\}$  is defined as

$$\|a - b\| = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2}$$

Now consider an unknown object  $X$  that needs to be classified. Its features can be computed, then it is positioned on the plane, as shown in Figure 4.1, noted as  $x$ . In Figure 4.1, the distance from  $x$  to  $w_i$  is obviously less than that of  $x$  to  $w_j$ .  $x$  is therefore classified to  $w_i$  according to the minimum-distance classification algorithm.



**Figure 4.1** Pattern in Euclidean Space

Note the space in Figure 4.1 can be imagined as an  $n$ -dimensional Euclidean space  $E^n$ , every point is an  $n$ -dimensional vector. Same algorithm can be used as a classifier. In the space  $E^n$ , distance between two vectors  $a$  and  $b$  has been defined previously in the table about symbols.

If there are more than two classes in the  $E^n$  space, the same algorithm is still working as a classifier. That is the statistical pattern recognition algorithm used in this automatic solder joint inspection system.

## 4.2 The Sunview Window Tool for Solder Joint Inspection

The Solder Joint Inspection Sunview Window Tool is a complete application software package written in Sunview window. Each step of unsupervised statistical pattern recognition is done by pushing certain software button using the mouse of the Sun workstation. The C source code of the whole software is

shown in Appendix D. The Figure 4.2 shows the Solder Joint Inspection Tool Window.

While totally four kinds of windows are offered by Sunview, only two different kinds of windows, Canvas and Panel are used in this tool. Canvas is an environment for graph and image displaying. Panel is one of the main methods for man-machine communication, such as keying-in file names, selecting functions, setting parameters etc.. This tool, when working on the Sun workstation screen, is that shown in Figure 5-1, where Image 't' is showing on the Canvas. Whole procedure is described next step by step.

#### **4.2.1 Solder Joint Segmentation:**

1. Typing-in image data file name: move the arrow cursor near by the line prompted with "Draw data file1:", press the left button of the mouse, a small triangle cursor will flash next to the ":", type in the data file name, denoted as filename1 below.

2. Typing-in position file name: Position file is used to tell the computer where a certain class of solder joints is. Do the same thing as Stop(1), let small triangle flash on the second line, prompted with "Position file1:", type in the initial name of position file, denoted as name2 below. An extension ".pos" is assumed.

3. To show the image: push the Panel button "New", that is done by moving arrow cursor onto the Panel button "New" and pressing the left button of the mouse. Through out this thesis this concrete operating procedure will not be repeated when a term *push Panel button "(specific function name)"* is mentioned.

4. To position the solder joints: When arrow cursor is moved onto Canvas, it becomes square with a cross tail, horizontal and vertical, which is

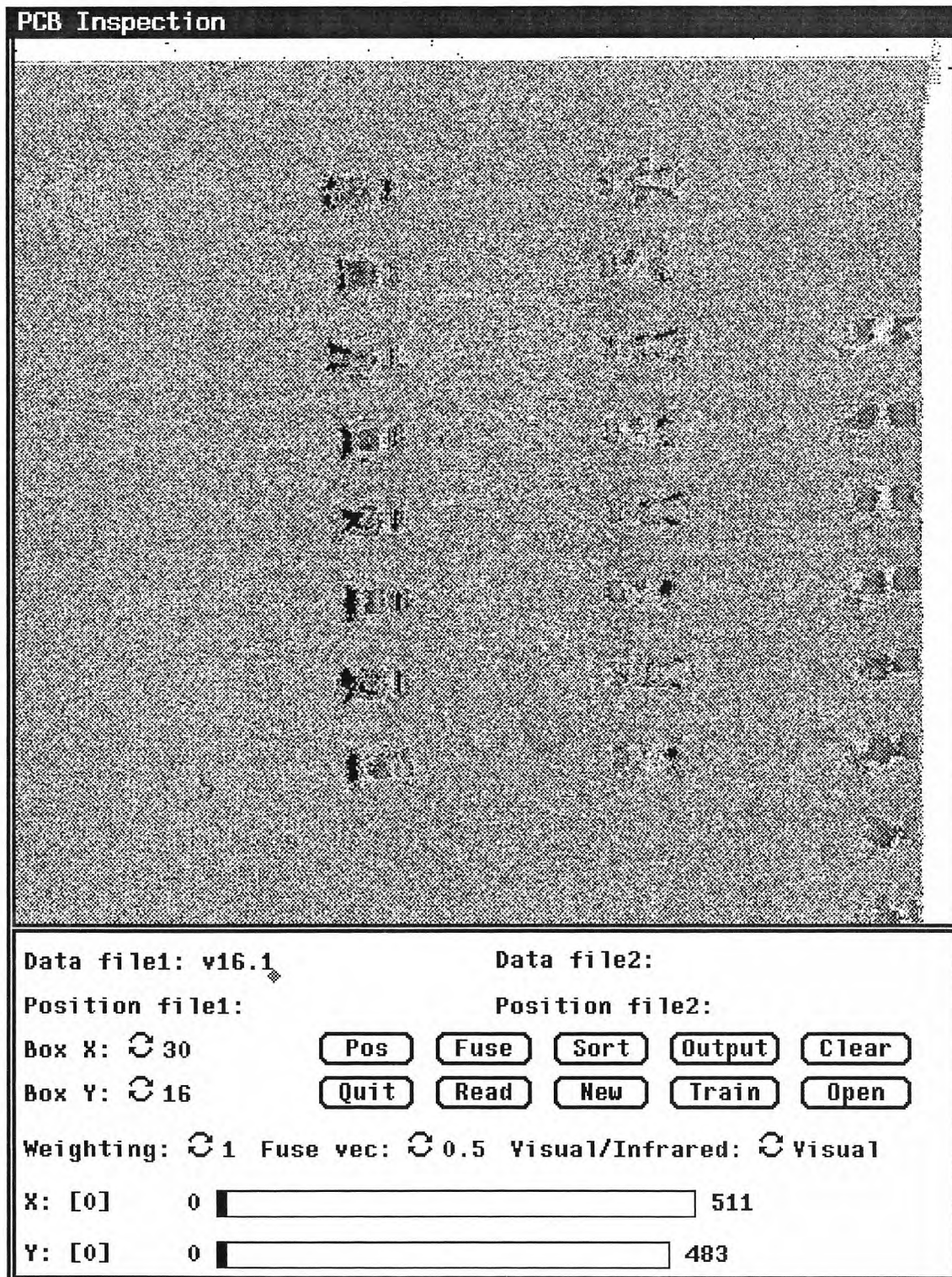


Figure 4.2 Solder Joint Inspection Tool Window

helpful to position adjacent solder joints. The keys “W, Z, A, S” on the keyboard are used to move the square cursor one pixel “up, down, left, right”, respectively. If the center of the cross seems accurately in the center of the solder joint which belongs to the class to be positioned, press the key “E” on the keyboard, a small frame box will appear surrounding the center of the solder joint and a pair of X-Y coordinates is recorded in memory. The size of the box, either X side or Y side, should be adjusted by changing the Panel items “Box X” and “Box Y:”, that will be explained below in detail.

5. To erase an input position: If a box seems not accurately positioned on the center of a solder joint, move square cursor onto the box and press key “Q” to erase the old box together with the position record in memory. Use “W, Z, A, S” to adjust the square cursor and press “E” when the position seems satisfactory. Repeat this step until the box is located on the center of the solder joint.

6. To create a position file: When a certain class of solder joints are partially or totally positioned, push the Panel button “Output”, which will save all records onto the position file name2.pos.

7. To modify a position file: After the Steps (1), (2), (3), push the Panel button “Read”, every position defined in the old position file will be called back and boxes are shown on the solder joints. Boxes can be erased, added, modified. After everything is ready, push “Output” again. The old position file will be covered by a new one with the same name.

Repeat Steps (2)-(7) to create different position files for different classes of solder joints. This completes the segmentation phase of the inspection.

#### **4.2.2 Mean Features Learning Or Training:**

Before training an important preparation is to decide the processing size of the

solder joint subimage. The size is set by Panel items "Box X:" and "Box Y:". One way to change them is moving the arrow cursor near the circle symbol and Pushing the left button of the mouse. Size will be changed to presetting value circularly for every click. Another way, a pull-down menu can be selected by pushing the right button of the mouse, moving mouse to highlight certain setting then releasing the button. Training steps are:

1. The first step here is the same as Step (1) of Phase A.

2. To begin a new round of training: push the Panel button "Open" for beginning a new round of training, name1.lnm and name1.fea will be created or renewed. Because the new data will be appended to these files while different classes of solder joints are used for training one by one.

3. Same as Step (2) of Phase A.

4. To train: push the Panel button "Train", the class in the name2.pos will be used for training and statistical mean features are recorded.

Repeat Steps (3), (4) until every position fill for the image name1 goes through. This finishes the training phase. It is recommended that always keep the same order of the position files for different rounds of training. Mean feature vectors are ready for the next phase "Classification:", in file name1.fea. Another output file name1.lnm is minimum and maximum values for every feature and every class. It is used to calculate the maximum value needed to normalize all features. That is also useful for analysis purpose. After normalization features are in the range of 0 to 1 and have equal weight to the distance decision. In the meantime a set of files name2.fea are output for analysis purpose. The standard deviation and relative error for every kind of feature are useful to evaluate the feature.

### **4.2.3 Classification:**



Before starting to classify, a weighting vector should be ready. As the initial value, a vector has its each component being 1. Text editor could be used to create a file named as `weight1.vec`, where only 1s separated by space exist. The number of 1s is the same as the number of features to be trained. Other weighting vectors could be put on `weight2.vec`, `weight3.vec`,.... Using panel circle selecting item "weighting :" to decide which vector should be used. During this procedure it is necessary that the same box size is used in one round of training and classifying. Classifying steps are described in the following:

1. The same as Step (1) of Phase A.
2. Selecting weighting vector: use Panel circle item "weighting :".
3. The same as Step (2) of Phase A.
4. Classifying: push the Panel button "Sort".

Repeat Steps (3),(4) for every class of solder joint by changing position files in the same orders when they are used for training . In classification phase two kinds of files are generated. One is `name1.sot`. This is an appended file. Sorting every position file will produce one line message. Classes are named as 1 to 4, following the order in which they are trained and sorted. The number shown after each class number means how many solder joints have been classified into this class. Because first line shows the results of the classification of Class 1, a largest number in the first line of the group is expected after the integer 1. The largest number in the second line of the group is expected after the integer 2, and so on. Two main error rates, Good/bad inspection rate and correct classified rate, are used to evaluate the performance of the classifier. Since only one class of solder joints is known as "good", Class 2 in the experiment here, the Good/bad inspection rate is computed by plus how many solder joints of Class 2 are classified into Class 2 and how many solder joints of other classes are not recognized as Class 2, then

divided by total number of solder joints in this image. The correct classified rate is the number of solder joints which are classified into the right classes divided by total.

Another kind of output files after sorting is a set of filename2.out files. The X-Y coordinates of every solder joint, the normalized distances between the feature vector associated with the solder joint and the mean feature vector of each class are printed in one line. At the end of this line is the class number to which the solder joint is assumed to belong. That is, it has the least distance to this class comparing to others. These files give the information about why a certain solder joint is sorted to a wrong class, how far it is away from its own class, or how far it is away from other classes when it is sorted correctly.

A group of experiment, training or sorting, means going through all position files associated with one image, when all parameters kept unchanged. A round of experiment means beginning from a group of training and ended by one or more groups sorting. For example, changing the weighting vector, more than one group of sorting can be implemented and different results could be obtained while the mean features are kept unchanged. Changing the size of the solder joint subimage, by changing the "Box X:" and "Box Y:" setting, and changing the typical sets of solder joints used for training, more rounds of training and sorting could be tested and more different sorting results could be obtained.

#### **4.2.4 Classification of Fusing Visible and Infrared Images.**

There are two set of features for training and sorting, one is for visible image, the other is for infrared image. Use the Panel item "Visible/Infrared" to make your choice. After choosing a set of feature, you can proceed with the process as described in the previous subsections. Pushing the Panel item "Visible/

Infrared" will switching you from one set of features to the other.

To make fusion experiment of visible and infrared images, you should process one visible image and one infrared image respectively to obtain the output files which have the extension ".out". After that, you should put the visible image file name which has been processed at the Panel item, "Data file1:" and its position file name at the Panel item "Position file1:", and put the infrared image file name at the Panel item "Data file2", again, its position file at the "Position file2:". The next step is choosing the weighting vector for the position filename1 from the Panel item "Fuse vec", and the 1-vector will become the weighting vector of the position filename2. Finally, push the Panel button "Fuse", the output will be stored in the files with extension ".fus". You can do fusion inspection for different position files by changing the position filename1 and the position filename2, and repeat the process described above.

## **Chapter 5**

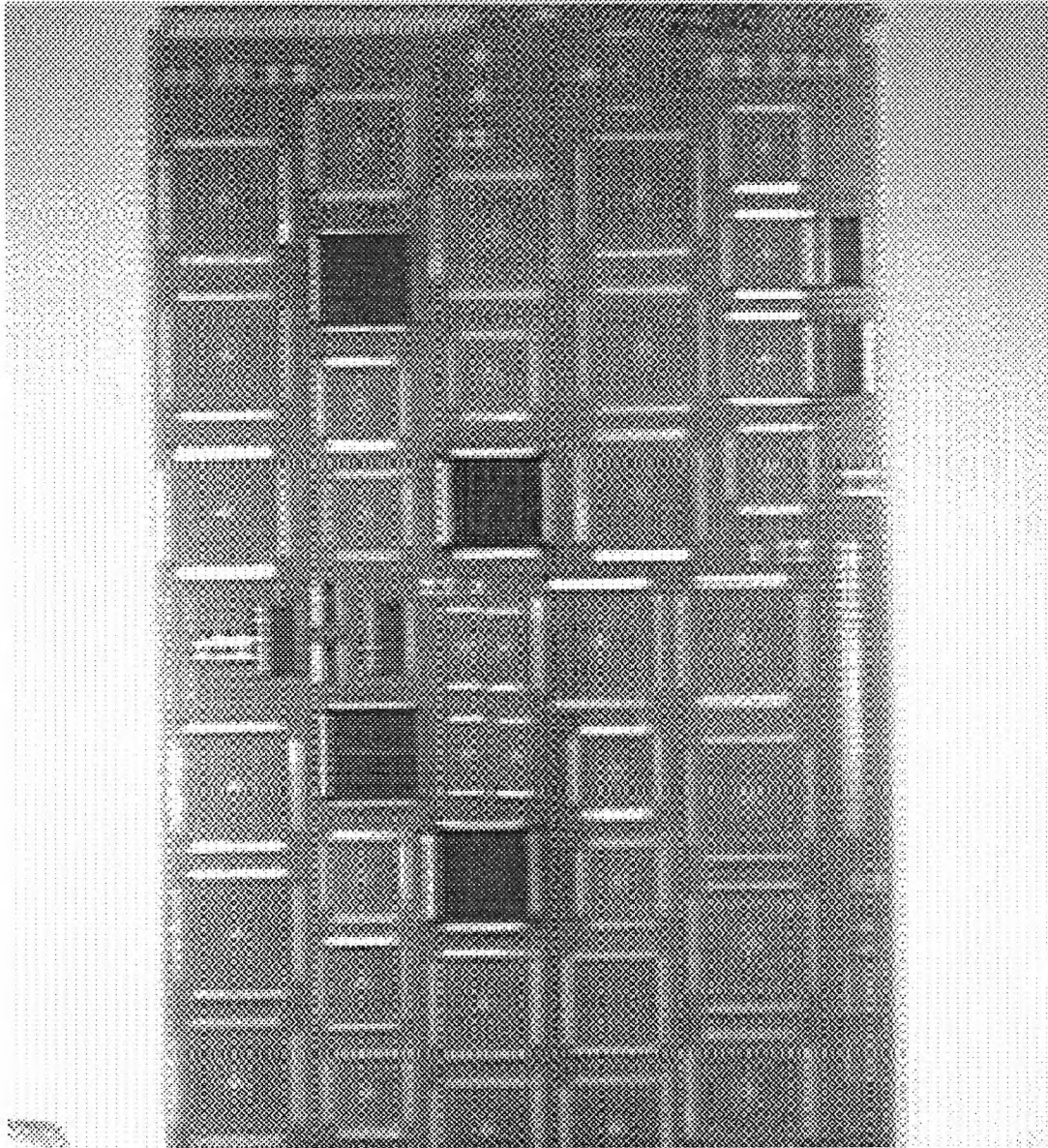
# **EXPERIMENT AND RESULTS**

This chapter introduces the experimental procedure and presents the experiment results. During the experiment, infrared images of solder joints are taken by the Datacube image processing system and stored into the host computer. The histogram of the raw infrared images are modified by a professional image processing software package named Khoros to achieve higher quality. After processing, the images are input into the Solder Joint Inspection Sunview Window Tool to do the classification.

### **5.1 The Sample Surface Mount Circuit Board**

The sample surface mount experimental circuit board used in our research was designed and built by AT&T Bell Lab. The experimental circuit board is designed for surface mount solder joint inspection purpose. There are ten chips of various types and sizes mounted on the board. All the leads are electrically isolated and all the chips are dump packages, as shown in Figure 5.1.

There are two 50-mil chips with 16 Gull-wing leads (IC16.0, IC16.1), four 50-mil chips with 28 Gull-wing lead (IC28.0, IC28.1, IC28.2, IC 28.3), two 50-mil chips with 68 J-lead leads (IC68.0, IC 68.1) and two 25-mil chips with 132 Gull-wing leads (IC132.3, IC132.4) on the experimental circuit board. This circuit board is made as follows: during the pasting procedure, a special stencil with large and small hole alternatively on it was applied, so that paste volume is made to be one in large unit and one in small unit,



**Figure 5.1** The Experimental Circuit Board

alternatively. Then, some joints are scraped to have no paste after the pasting procedure, some are scraped to half of the normal paste volume, while the rest are left untouched. Therefore, for this experimental circuit board, we have various types of joints with different paste volume on the circuit board.

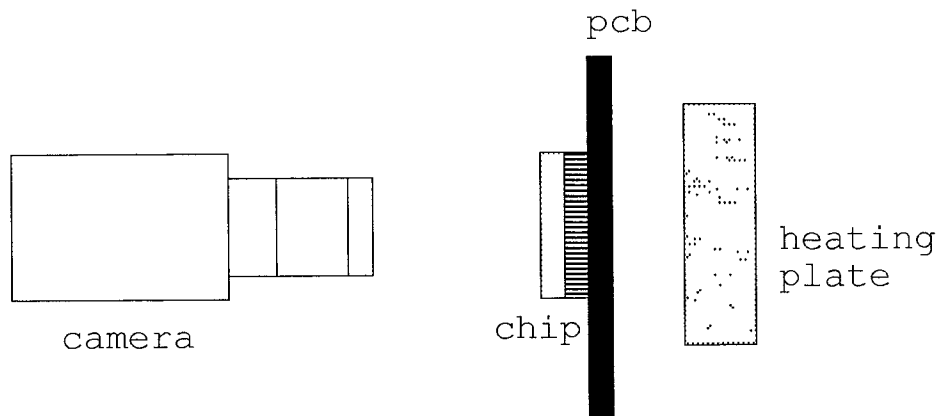
After soldering, the joints were checked to see if they are electrically connected well. It was found that some joints with no paste were electrically disconnected. The types of defects of the joints on the board were recorded by AT&T researchers. Appendix D contains the diagram of this circuit board, and a detail description of each joint's condition. All the materials in Appendix D was provided by Dr. Frank Higgins, AT&T Bell Laboratory, Princeton.

## **5.2 The Experiment Procedure**

### **5.2.1 Heating Procedure**

The testing circuit board prepared by AT&T Bell Laboratory was heated and then cooled during the experiment. A CCD infrared camera with a 320x244 element array of PtSi Schottky-barrier detectors was used to take infrared pictures of the joints. The infrared images of the solder joints were recorded with the VCR and the Datacube real time image processing system, during both the heating and cooling processes.

Two heating procedures were used in our infrared inspection. The first is heating from the back of the circuit board with a temperature controllable heating plate, as illustrated by Figure 5.2. In this test, the temperature of the circuit board is controlled to raise from the room temperature to about 90°C and then the heating plate is removed. The circuit board cools down naturally in the air to the room temperature. The second heating procedure is heating the circuit board with an oven. The temperature of the oven is controlled to



**Figure 5.2** Plate Heating Procedure

reach and stay at about  $100^{\circ}\text{C}$ . The circuit board is then taken out and the cooling process is observed.

### 5.2.2 Infrared Imaging

The IR-CCD infrared camera can output both corrected and uncorrected signals. In our experiment, the corrected signal is connected to the input of the Datacube image processing system. The Datacube image processing system was driven by the host computer to take infrared images from the infrared camera. In order to grasp the best moment to obtain a high quality infrared image, a series of images during an appropriate period were recorded by the system while the circuit board was in heating or cooling process. A special Unix C shell script was written to coordinate the operation of the Datacube image processing system. A VCR was used to record the circuit board infrared images of the whole experimental procedure for future reference.

Because of the small size of solder joints, each experiment could only cover a small section of the whole circuit board. Even though a large number of infrared images were taken at each section of the circuit board, only one was chosen as the infrared image of that section. Thirty-four experiments are needed to cover all the solder joints on the testing circuit board.

### 5.2.3 Image Processing

The raw infrared images were processed before being used in the pattern recognition procedure. An advanced professional image processing software named Khoros was used to perform the image process task in our project.

Khoros is a software environment for research in the use of visual programming as a tool for software development for scientific visualization. The Khoros infrastructure consists of several layers of interacting subsystems. A user interface development system (UIDS) combines a high-level user interface specification with methods of software development and maintenance that are embedded in a code generation tool set. An interoperable data exchange format (VIFF) and algorithm library contain the application specific layer. The UIDS can be used to create three user interface styles:

- (1) a command line user interface,
- (2) a graphical, form/menu based user interface, and
- (3) the basic operations of a visual programming language.

The result is an extensible visual programming environment. The current library of over 260 routines has been developed to facilitate research in image processing, signal processing, pattern recognition, remote sensing, machine vision, and geographic information systems.

The image analysis functions of Khoros can even help users to analyze the some complicated features of a image which are not easy to observe by human eyes. Under the assistance of Khoros, a set of new feature which is more suitable for surface mount joints has been developed.

As a matter of fact, the histogram distribution of raw infrared images is concentrated in a narrow range of gray level. The raw infrared images are not



easy to recognized by the human eyes, even though they may contain rich information of the solder joints. Histogram modification must be applied to help people analyze the pattern features. Furthermore, raw infrared images of different experiment may have different background gray level, because it is impossible to take pictures at exactly the same temperature during different experiment. Therefore, the histogram of each infrared image must be modified until all images have the same background gray level. Besides, images from different experiment may have different tilt angle, which may influence the pattern recognition result. Other defects on the raw images such as impuls noise, need to be compensated to achieve better results.

The image processing procedures are listed below.

### 1. Background Gray Level Modification

Raw infrared images are input into the Khoros software to calculate the mean back gray level. Once the mean background gray level has been decided, a histogram transferring illustrated in Figure 5.3. All the raw images will have the same background gray level.

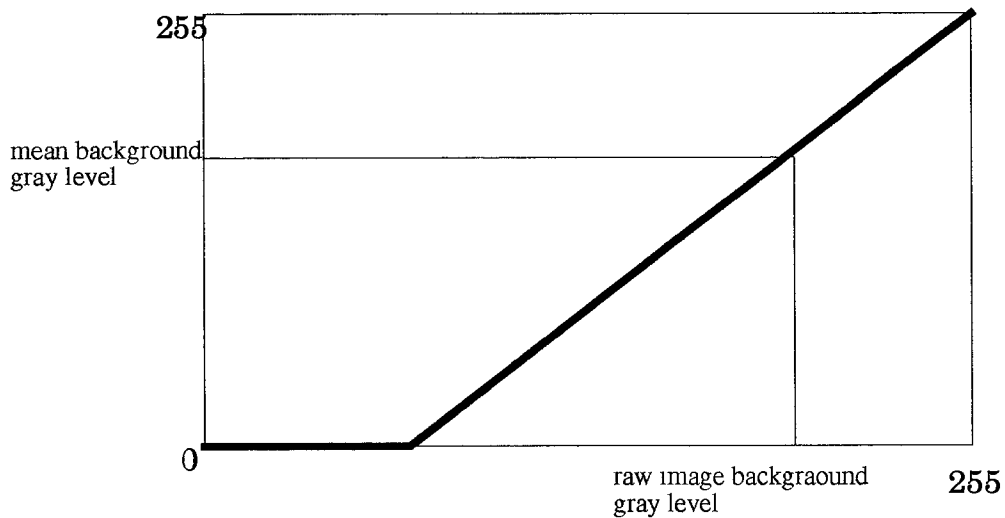
### 2. Image Rotation

Because the pattern features used in our experiment do not take the rotation angle into account for simplicity, tilt angles of the solder joint subimages will influent the recognition results. Every raw image is rotated until the solder joint line is exactly vertical using the rotation function of the Khoros software.

### 3. Noise Compression

Some "bad" pixels which are corrupted by impuls noise are detected and replaced by a mean gray level of the neighbor pixels.

After these processing, a raw image in ready for patter recognition.



**Figure 5.3** Histogram Modification

### 5.2.4 Pattern Recognition

The processed images are sent to the Sunview Window Solder Joint Inspection Tool to perform the pattern recognition procedure. The source code of this software are shown in Appendix A. Position files are created and group of joints are send as samples for training. \*.fea and \*.lmm files are recorded. The *sorted* results are recorded in both .out files and .sot files. For more detail about how to use the Sunview Window Solder Joint Inspection Tool, please consult section 2 of chapter 4.

A new set of features is developed for the surface mount solder joints. Below is a brief description of these features:

1. left half area mean gray level.

$$m_l = \frac{1}{\frac{x}{2} \times \frac{y}{2}} \sum_{i=0}^{\frac{x}{2}} \sum_{j=0}^{\frac{y}{2}} G(i, j)$$

where  $m_l$  is the mean left half area gray level,  $x$  is the width of the subimages window,  $y$  is the height of the subimage window,  $G(i, j)$  is the gray value of the image.

### 2. right half area mean gray level

$$m_r = \frac{1}{\frac{x}{2} \times \frac{y}{2}} \sum_{i=\frac{x}{2}+1}^x \sum_{j=\frac{y}{2}+1}^y G(i, j)$$

where  $m_r$  is the mean right half area gray level,  $x$  is the width of the subimages window,  $y$  is the height of the subimage window,  $G(i, j)$  is the gray value of the image.

### 3. left half area variance

$$\sigma_l = \sqrt{\sum_{i=0}^{\frac{x}{2}} \sum_{j=0}^{\frac{y}{2}} (G(i, j) - m_l)^2}$$

where  $\sigma_l$  is the left half area variance,  $m_l$  is the mean left half area gray level,  $x$  is the width of the subimages window,  $y$  is the height of the subimage window,  $G(i, j)$  is the gray value of the image.

### 4. right half area variance

$$\sigma_r = \sqrt{\sum_{i=\frac{x}{2}+1}^x \sum_{j=\frac{y}{2}+1}^y (G(i,j) - m_r)^2}$$

where  $\sigma_r$  is the right half area variance,  $m_r$  is the mean right half area gray level,  $x$  is the width of the subimages window,  $y$  is the height of the subimage window,  $G(i, j)$  is the gray value of the image.

Even though these four features appear quiet simple, they worked very well in our experiment. Actually, features are dependent on the direction of the solder joint subimages. However, after the rotation process of the Khoros image processing software, we can make all the solder joint subimages having the same direction: left is the chip side, right is the outer side. Therefore, these four features will not loose the generality of direction.

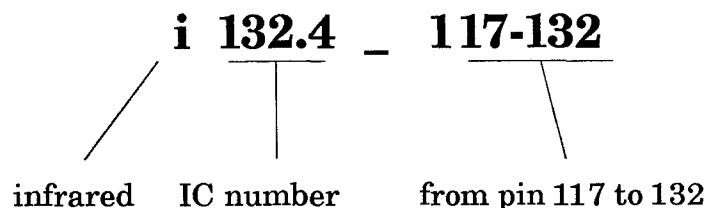
## 5.3 Experiment Results

### 5.3.1 Solder Volume Inspection

As described in the first section of this chapter, three different solder volumes are made on the experimental circuit board for each chip. The three different volumes are scraped to no paste, scraped to half paste volume and normal paste volume. The purpose of our volume inspection experiment is to recognize the solder volume of each solder joint, classify each joint into one of the three group mentioned above.

Twenty-three infrared images are taken to cover 23 selected regions on the circuit board which contain solder joints of various type, size, and solder volume. For each of the images, experimental procedures were followed in the order as described in section 2. Each selected region has a name like

i132.4\_117-132. The meaning of each figure are illustrated in Figure 5.4



**Figure 5.4** Name of the Selected Experiment Region.

These selected regions are briefly described below:

1. i16.1:

all of the 16 sites of IC16.1 (16 pin SOIC, 50-mil pitch, Gull-wing lead)

2. i28.1\_1-7, i28.1\_8-14, i28.1\_15-21, i28.1\_22-28:

all of the 28 sites of IC 28.1 (28 pin SOIC, 50-mil pitch, Gull-wing lead)

3. i68.1\_1-6, i68.1\_7-12, i68.1\_13-17, i68.1\_18-25, i68.1\_27-34, i68.1\_35-40, i68.1\_41-46, i68.1\_47-51, i68.1\_52-59, i68.1\_61-68:

all of the 68 sites of IC68 (68 pin quad flat pack, 50 mil pitch, J-lead)

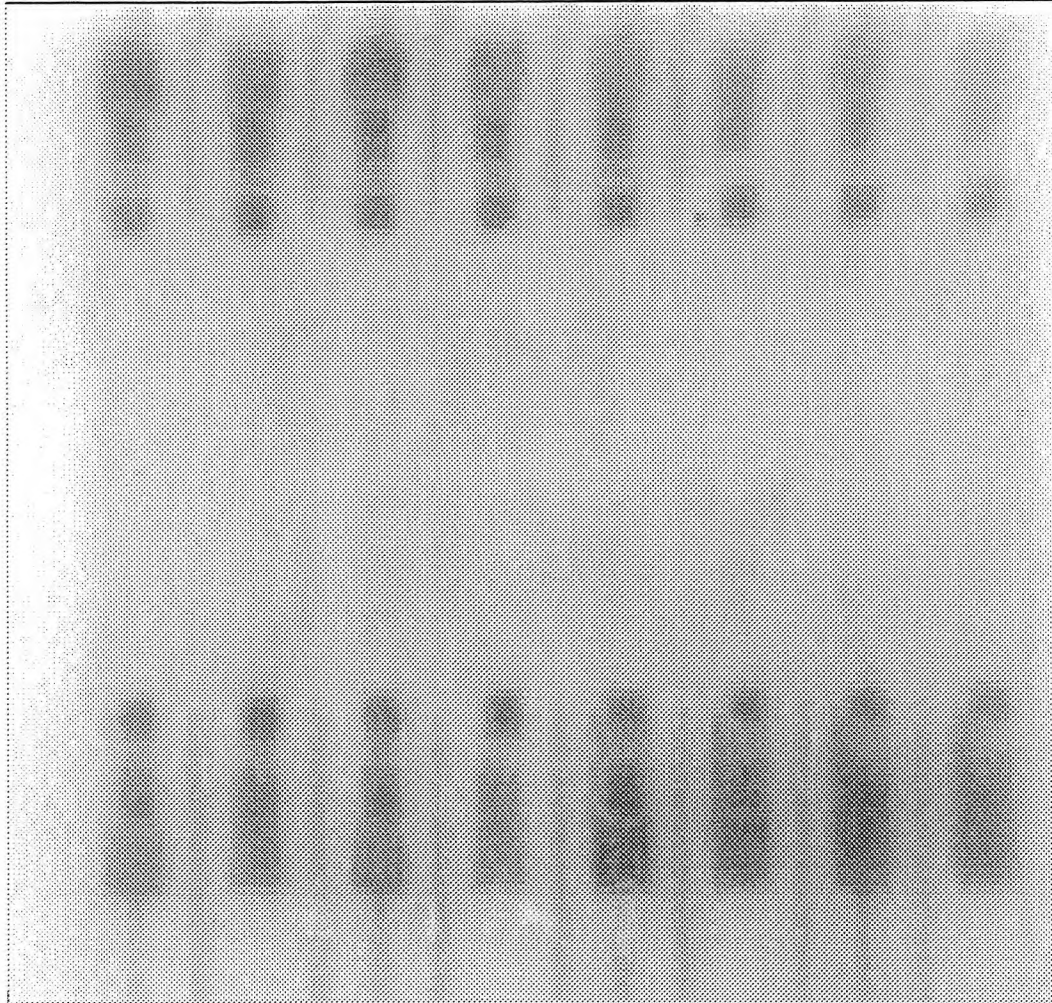
4. i132.4\_1-17, i132.4\_18-33, i132.4\_34-51, i132.4\_52-66, i132.4\_67-73, i132.4\_74-99, i132.4\_100-116, i132.4\_117-132:

all the 132 sites of IC 132 ( 132 pin quad flat pack, 25-mil pitch, Gull wing lead)(This IC only have two different solder volume: scraped to no pasted and normal)

One of the infrared images, i16.1, is shown in Figure 5.5.

In this image, the lower right four joints are scraped to no paste, the upper left four joints are scraped to half paste, and the rest are normal paste. We can recognize some of the patterns even by our eyes.

All of the 23 infrared images for each selected area of the experimental circuit board are shown in Appendix B. The pattern recognition results for



**Figure 5.5** the Infrared Image of IC16.1: i16.1.

each image are recorded in .fea files and .out files, as described in the Section 4 of Chapter 4. Result data are sorted into the following three tables, which is for 50-mil Gull-wing, 50-mil J-lead and 25-mil Gull-wing, respectively. Inside these tables, the four columns in the joints categories are the total number of joints in each images, the number of joints which is in normal paste, scraped to half paste and no paste at all, respectively.

Table 1: Pattern Recognition Results For 50-mil J-lead ICs

images	number of joints				classify results	
	total	normal paste	half paste	no paste	correct number	correct rate
i68.1_1-6	6	6	0	0	6	100%
i68.1_7-12	6	3	0	3	6	100%
i68.1_13-17	6	0	0	6	6	100%
i68.1_18-25	8	8	0	0	8	100%
i68.1_27-34	8	0	8	0	6	75%
i68.1_35-40	6	6	0	0	5	83%
i68.1_41-46	6	3	3	0	6	100%
i68.1_47-51	6	0	6	0	4	67%
i68.1_52-59	8	8	0	0	7	87.5%
i68.1_61-68	8	0	0	8	8	100%
total	68	34	17	17	62	91.1%

Table 2: Pattern Recognition Results For 25-mil Gull-wing ICs

images	number of joints			classify results	
	total	normal paste	no paste	correct number	correct rate
i132.4_1-17	17	17		17	100%
i132.4_18-33	16	0	16	16	100%
i132.4_34-51	17	17	0	17	100%
i132.4_52-66	16	16	0	16	100%
i132.4_67-83	17	17	0	17	100%
i132.4_84-99	16	0	16	16	100%
total	132	100	32	132	100%

Table 2: Pattern Recognition Results For 25-mil Gull-wing ICs

images	number of joints			classify results	
	total	normal paste	no paste	correct number	correct rate
i132.4_100-116	17	17	0	17	100%
i132.4_117-132	16	16	0	16	100%
total	132	100	32	132	100%

Table 3: Pattern Recognition Results For 50-mil Gull-wing ICs

images	number of joints				classify results	
	total	normal paste	half paste	no paste	correct number	correct rate
i16.1	16	8	4	4	16	100%
i28.1_1-7	7	7	0	0	7	100%
i28.1_8-14	7	0	7	0	6	86%
i28.1_15-21	7	7	0	0	7	100%
i28.1_22-28	7	0	0	7	7	100%
total	44	22	11	11	43	97.7%

It should be noted that some of the "misclassification" in our experiment is due to the ununiformity of the solder volumes in each class, especially in the "scraped to half" class. Because the site areas of IC16.1, IC28.1 and IC68.1 are in a large/small alternating pattern, as described at Appendix D, the solder volume of the joints in small site area may be smaller than that of the joints in large area being scraped to half paste volume. Therefore, there is not surprising that the pattern recognition software classifying them into "wrong" class. Considering of this, the results listed in



these three tables is quiet satisfying.

### **5.3.2 Electrical Connectivity Inspection**

Electrical connectivity inspection has also been investigated in this project. There are four joints on the experiment circuit board known as electrically unconnected. Three of them are 50-mil Gull-wing with no paste; while the other is 50-mil Gull-wing with half paste.

Experiments were designed to reveal the information about electrical connectivity containing in the infrared images. One of them is taking the sequence of infrared images during both the heating and the cooling process. The difference between these time sequent images has been then calculated and analyzed to find the trace of electrical connectivity. Im another experiment, heat sink has been used to accelerate the cooling process has also been investigated. An aluminum heat sink is pressed on the surface of the testing chip during both the heating and cooling processes. This procedure, however, did not get any useful results.

Still in another effort, we have tried is to recognize the electrical connectivity by extracting more complicated features concealing in the infrared images. Again, the Khoros image processing software has been applied to analyze the advanced features such as invariant moment and 2nd moment. This work however, just at the beginning stage and is expected to be continued at NJIT by other investigators.

# Chapter 6

## VISIBLE LIGHT

### INSPECTION EXPERIMENT

#### 6.1 Visible Light Inspection System

The visible light inspection system configuration is very similar to that of the infrared inspection system. The CCD visible light camera is fixed on a tripod. The “Green” channel which contains the synchronizing signal is connected to the input (P12) of a MAX-SCAN module mounted on the Datacube chassis. The video output of the camera is connected to a color monitor. The system diagram is in Figure 6.1.

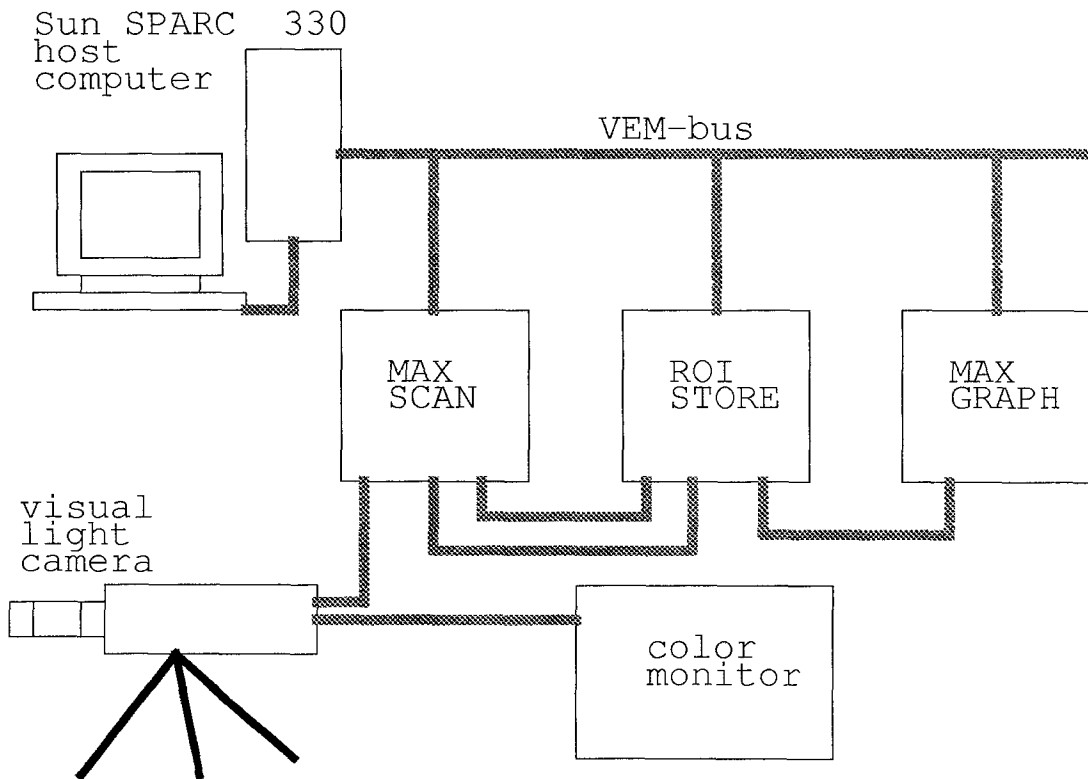
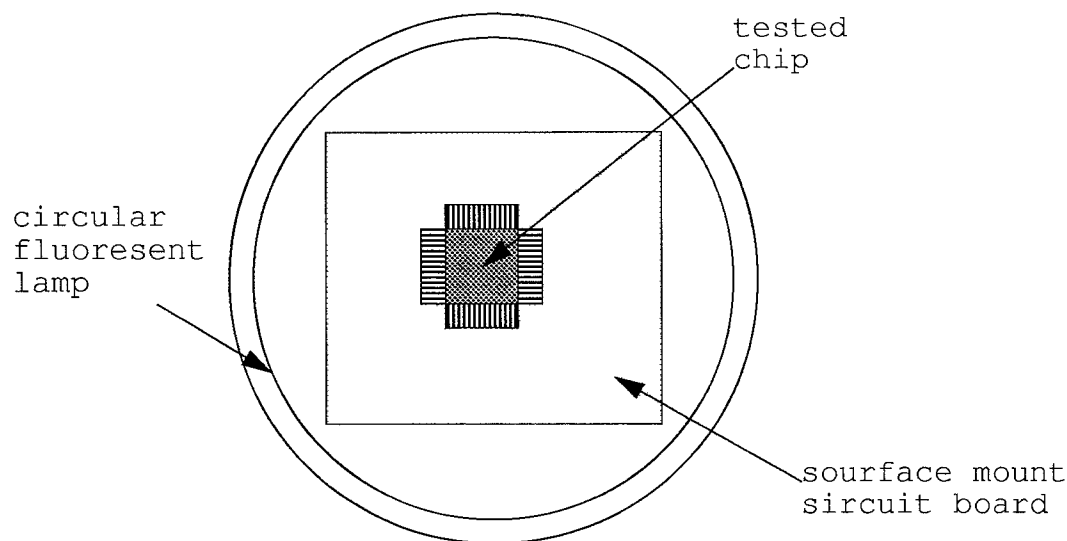


Figure 6.1 Visible Light Inspection System

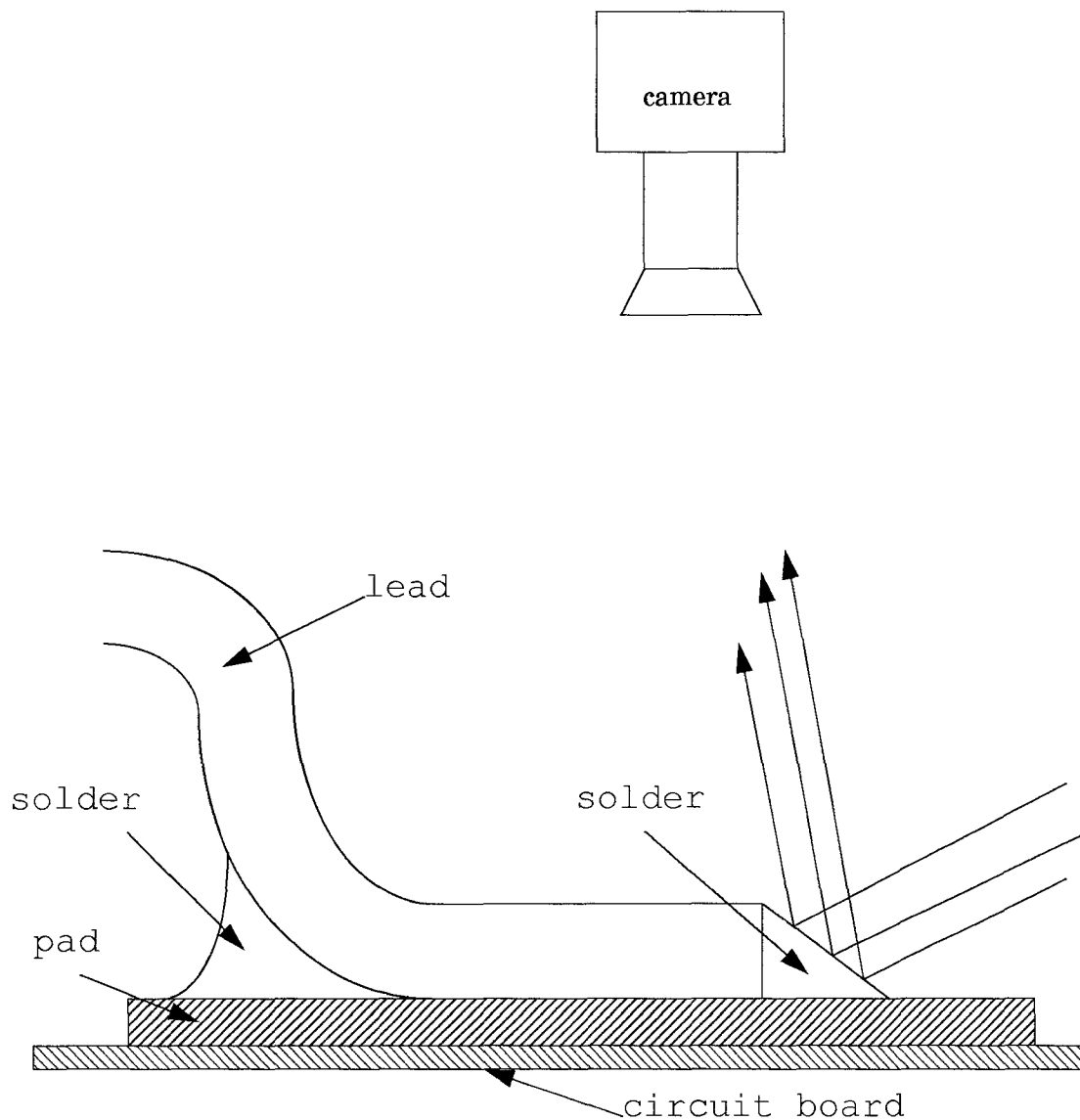
## 6.2 Illumination Method

A circular fluorescent lamp is used as the light source. The lamp is put very low to let the light projected to the surface of joints at a fairly low angle, as shown in Figure 6.2.1. For those joints having enough solder volume, the slant surface of the solder will reflect the incoming light horizontally to the direction of the camera, as shown in Figure 6.2.2 (a). For those joints with insufficient solder volume or without solder at all, the surface angle of the solder will change. The reflected light will turn to other direction rather than pointing to the camera, as shown Figure 6.2.2 (b).



**Figure 6.2.1** Top View of the Illuminating Position

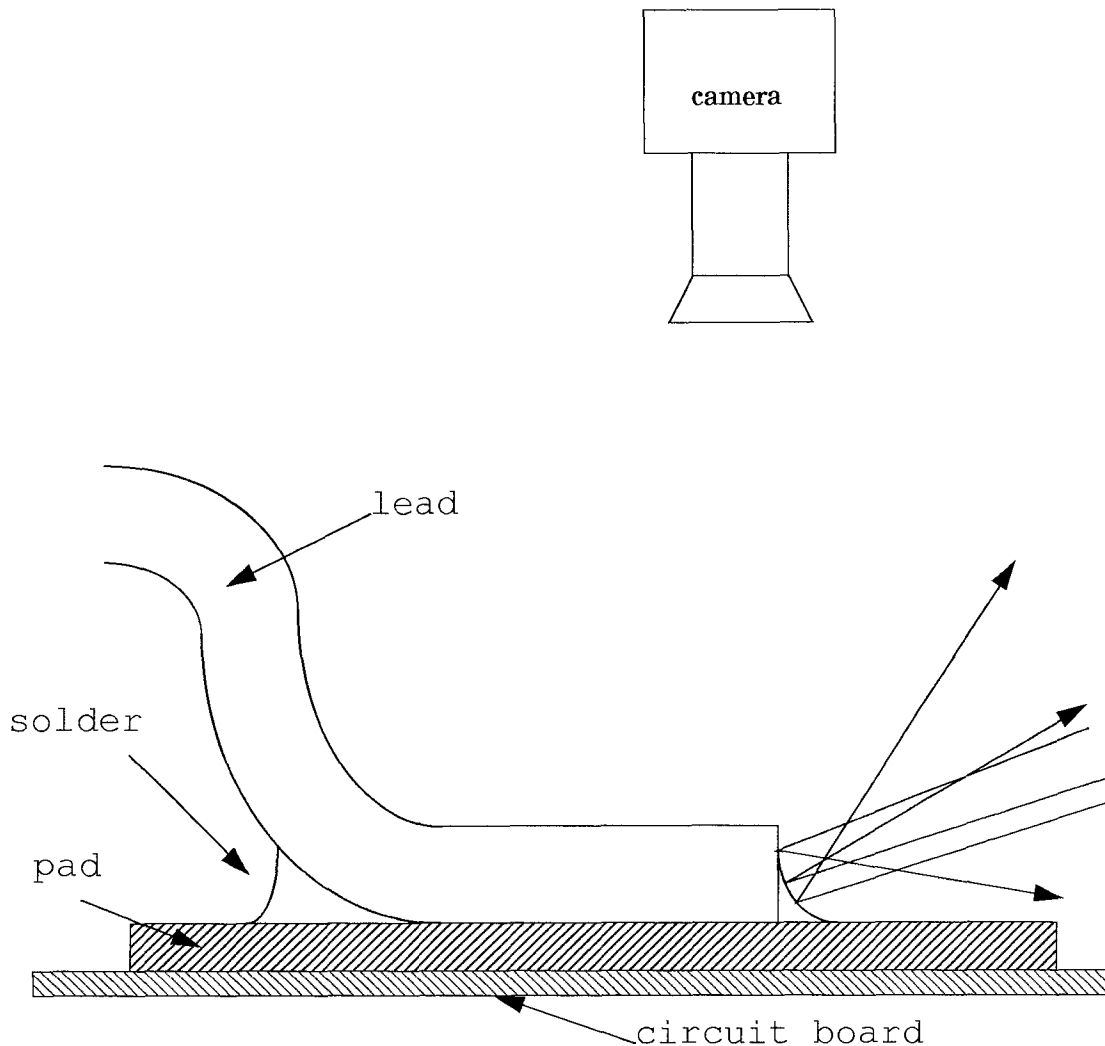
It should be noted that even though this illumination method was found to be useful from the point of view of inspection, it might not be practical in the industry. Sometimes, components may be highly impacted on the circuit board, blocking lights projecting from low angle. Other illuminating method have also been test in our experiment, however, they were found to be less effective than the one we above mentioned.



**Figure 6.2.2a** Reflected Light on Normal SolderJoint Surface

### 6.3 The Experiment Procedure

When the lamp and the focus of the camera is adjusted appropriately, we start the image taking program to drive the Datacube image processing system to take the picture. A program named *take* is used to scan the input data and store it in the ROI-STORE module of the Datacube system. The images are stored into the hard disk of the host computer later and transferred to the



**Figure 6.2.2 b** Reflect light on Insufficient SolderJoint Surface

Msaol Lab in the CADl Department, NJIT through the local network for image processing.

At the Msaol Lab, Department of CAD, we apply the Khoros image processing software to do the image enhancement, as described in Chapter 5. The processed images are transferred back to the host computer of our system.

The pattern recognition Sunview window tool introduced in Chapter 4 is applied to process the images. For more detail about how to create position

files, train features for each class, and compute the feature distance of each tested solder joint subimage, please consult Chapter 4 and Chapter 5.

The same features as those for the infrared inspection are applied for visible light inspection, but with different weighting index, since the importance of each features is different from each other. More weight is added to the variance features and less weighting is applied for the mean gray level features. Different size of the solder joint subimages varying from 16x16 to 46x46 have been tried to compute the feature distance of each solder joint image. Just as expected, each image has its own suitable solder joint subimage size. The final results of each experiment are contained in the .sot files and .out files. (For more detail about .sot, .out., .fea and .lnm files, please see Chapter 4)

## 6.4 Experiment Result

In the visible light inspection experiment, regions are selected different from those of infrared inspection experiment, since the visible light camera has a fairly powerful zooming lens installed and has higher resolution the infrared camera. The new regions are:

1. v16.1: all of the 16 pins of IC16.1;
2. v28.1: all of the 28 pins of IC28.1;
3. v68.1\_1-17, v68.1\_18-34, v68.1\_35-51, v68.1\_32-68:  
four sides of IC68.1, covering all of its 68 pins;
4. the eight regions for IC132.4 as the same as those of infrared inspection experiment. The meaning of each are as the same as that described in last chapter.

The final results are contained in the following three tables, similar to

last chapter.

Table 4: pattern recognition results for 50-mil J-lead ICs

images	number of joints				classify results	
	total	normal paste	half paste	no paste	correct number	correct rate
v68.1_1-17	17	9	0	8	17	100%
v68.1_18-34	17	9	8	0	15	88%
v68.1_35-51	17	9	8	0	12	70.5%
v68.1_52-68	17	9	0	8	17	100%
total	68	36	16	16	61	89.7%

Table 5: pattern recognition results for 50-mil gull-wing ICs

images	number of joints				classify results	
	total	normal paste	half paste	no paste	correct number	correct rate
v16.1	16	8	4	4	16	100%
v28.2	28	14	7	7	23	82%
total	44	22	11	11	39	88.6%

Table 6: pattern recognition results for 25-mil gull-wing ICs

images	number of joints			classify results	
	total	normal paste	no paste	correct number	correct rate
i132.4_1-17	17	17		17	100%
i132.4_18-33	16	0	16	16	100%
i132.4_34-51	17	17	0	17	100%
i132.4_52-66	16	16	0	16	100%
i132.4_67-83	17	17	0	17	100%
i132.4_84-99	16	0	16	16	100%
i132.4_100-116	17	17	0	17	100%
i132.4_117-132	16	16	0	16	100%
total	132	100	32	132	100%



## **Chapter 7**

# **CONCLUSIONS AND RECOMMENDATION**

Various experiments have been conducted in this thesis work on the sample joints of the surface mount printed circuit board provided by AT&T Bell Laboratory. In the heating and cooling process, infrared images of solder joints were taken at certain temperature range. Statistical pattern recognition technique was then applied. It has been demonstrated that infrared imaging technique can identify solder joints of surface mount printed circuit boards according to their solder volume. The correct classification rate was found to be in the range of 89% to 100%. This concludes that infrared imaging technique is promising for the automatic inspection of solder joints on surface mount printed circuit boards.

For further quantitative study of solder volume inspection, an experimental circuit board containing a series of solder joints with various solder volumes is needed. The solder volumes should be purposely made from the smallest to the largest possible quantities with a linear distribution, for example,  $1v$ ,  $2v$ ,  $3v$ , ..., where  $v$  is an appropriate volume unit. Hence, a detail quantitative relation between the solder volume and the intensity pattern of the infrared images can be investigated. For the study of electrical connectivity inspection, more work is still needed.

For more effective infrared imaging for industrial application, the resolution of the infrared camera should be increased. For instance, SR-170 video on the resolution of 512 or 640 elements in horizontal and 484 in vertical would be desirable, but 1040x1040 infrared imager recently introduced by Mitsubishi would still be better. The other choice is a line sensor with 10,000

to 20,000 elements.

Even though a sequence of infrared images has been taken during the experiments, only one of them was used in the classification process of our experiment. For better results, or in more advanced studies, a sequence of frames would be used to study the dynamic response.

# APPENDIX

## A Source Code of The Solder Joint Inspection Tool

```
#include <stdio.h>
#include <math.h>
#include <suntool/sunview.h>
#include <suntool/panel.h>
#include <suntool/canvas.h>
#include <suntool/tty.h>

#define CMS_SIZE256

static void help_proc(), quit_proc(), pos_proc(), fuse_proc();
static void output_proc(), sort_proc(), train_proc(), draw();
static void handle_event(), read_proc(), new_proc(), features();
static void clear_proc(), open_proc(), size_proc(), rpos_proc();

static int value_to_op();
static char value_to_c(), value_to_d();

static char cmdstring[256], *fv;
static short posvect[512]=NULL;
static int pcbpix[64][64], xsize, ysize;
static int bx2, by2, cswx, cswy, ofrx, ofry;
static float feavect[20], fea[120][10], data[512], data1[512], -
mean[6][20];
static float deltax, deltax, delta, deltax2, deltax2, pos, vec;
static float sjpgix[64][64], ll[5][20], mm[5][20]; ;

static Panel_item fname_item;
static Panel_item f1name_item;
static Panel_item f2name_item;
static Panel_item f3name_item;
static Panel_item box_xsize;
static Panel_item box_ysize;
static Panel_item weight_num;
static Panel_item visual_infrared;
static Panel_item fuse_vec;
static Panel_item cursor_x;
static Panel_item cursor_y;
Pixfont *bold;
struct rasterfile rh;

static Frame frame;
static Panel panel;
static Canvas canvas;
static Tty tty;
static Cursor cursor;
static Menu menu;
```

```

main(argc, argv)
int    argc;
char   **argv;
{
    Rect    *r;
    register Pixwin *pw;
    register inti, k ;
    u_char  red[CMS_SIZE], green[CMS_SIZE], blue[CMS_SIZE];
    struct pixrect*orig_pr, *new_pr;

/*    printf(fp1,"%3d %3d %3d \n", xsize+1, cswx, ofrx);*/
    pos=0;
    bold=pf_open("/usr/lib/fonts/fixedwidthfonts/screen.b.12");
    if (bold==NULL) exit(1);

    frame=window_create(0, FRAME, FRAME_LABEL, "PCB Inspection",
                        FRAME_CMDLINE_HELP_PROC, help_proc,
                        FRAME_ARGS, argc, argv,
                        /*    FRAME_ICON, &my_icon,*/
                        FRAME_NO_CONFIRM,TRUE,
                        WIN_ERROR_MSG, "Can't create win-
dow.",
                        WIN_HEIGHT, 484,
                        WIN_WIDTH, 512,
                        0);

    menu=menu_create(MENU_ACTION_ITEM, "Quit",  quit_proc,
                    0);

    canvas=window_create(frame, CANVAS,
                        WIN_CONSUME_KBD_EVENT, WIN_ASCII-
I_EVENTS,
                        WIN_CONSUME_PICK_EVENT, LOC_DRAG,
                        WIN_HEIGHT, 484,
                        WIN_WIDTH, 512,
                        WIN_EVENT_PROC, handle_event,
                        0);

    window_fit(canvas);

/* copy the original cursor image */
    orig_pr = (struct pixrect *) (LINT_CAST(
        cursor_get(window_get(canvas, WIN_CURSOR), CURSOR_IMAGE)));
    new_pr = (struct pixrect *) (LINT_CAST(mem_create(
        orig_pr->pr_width, orig_pr->pr_height, orig_pr->
pr_depth)));
    (void)pr_rop(new_pr, 0, 0, new_pr->pr_width, new_pr->pr_height,
PIX_SRC,
        orig_pr, 0, 0);

    cursor = cursor_create(CURSOR_IMAGE, new_pr,

```





```

    0);

fuse_vec = panel_create_item(panel, PANEL_CYCLE,
    PANEL_LABEL_X, ATTR_COL(16),
    PANEL_LABEL_Y, ATTR_ROW(5)+4,
    PANEL_LABEL_STRING, "Fuse vec:",
    PANEL_VALUE, 4,
    PANEL_CHOICE_STRINGS,
    "0.1", "0.2", "0.3", "0.4", "0.5", "0.6",
"0.7", "0.8", "0.9", 0,
    0);

cursor_x = panel_create_item(panel, PANEL_SLIDER,
    PANEL_LABEL_X, ATTR_COL(0),
    PANEL_LABEL_Y, ATTR_ROW(7)-4,
    PANEL_LABEL_STRING, "X:",
    PANEL_SLIDER_WIDTH, 256,
    PANEL_VALUE, 0,
    PANEL_MIN_VALUE, 0,
    PANEL_MAX_VALUE, 511,
    0);

cursor_y = panel_create_item(panel, PANEL_SLIDER,
    PANEL_LABEL_X, ATTR_COL(0),
    PANEL_LABEL_Y, ATTR_ROW(8)+4,
    PANEL_LABEL_STRING, "Y:",
    PANEL_SLIDER_WIDTH, 242,
    PANEL_VALUE, 0,
    PANEL_MIN_VALUE, 0,
    PANEL_MAX_VALUE, 483,
    0);

panel_create_item(panel, PANEL_BUTTON, PANEL_LABEL_IMAGE,
    panel_button_image(panel, "Pos", 5, 0),
    PANEL_ITEM_X, ATTR_COL(20),
    PANEL_ITEM_Y, ATTR_ROW(2)+10,
    PANEL_NOTIFY_PROC, pos_proc, 0);

panel_create_item(panel, PANEL_BUTTON, PANEL_LABEL_IMAGE,
    panel_button_image(panel, "Fuse", 5, 0),
    PANEL_ITEM_X, ATTR_COL(28),
    PANEL_ITEM_Y, ATTR_ROW(2)+10,
    PANEL_NOTIFY_PROC, fuse_proc, 0);

panel_create_item(panel, PANEL_BUTTON, PANEL_LABEL_IMAGE,
    panel_button_image(panel, "Sort", 5, 0),
    PANEL_ITEM_X, ATTR_COL(36),
    PANEL_ITEM_Y, ATTR_ROW(2)+10,
    PANEL_NOTIFY_PROC, sort_proc, 0);

panel_create_item(panel, PANEL_BUTTON, PANEL_LABEL_IMAGE,
    panel_button_image(panel, "Output", 6,
0),
    PANEL_ITEM_X, ATTR_COL(44),

```

```

        PANEL_ITEM_Y, ATTR_ROW(2)+10,
        PANEL_NOTIFY_PROC, output_proc, 0);

panel_create_item(panel, PANEL_BUTTON, PANEL_LABEL_IMAGE,
        panel_button_image(panel, "Clear", 6, 0),
        PANEL_ITEM_X, ATTR_COL(53),
        PANEL_ITEM_Y, ATTR_ROW(2)+10,
        PANEL_NOTIFY_PROC, clear_proc, 0);

panel_create_item(panel, PANEL_BUTTON, PANEL_LABEL_IMAGE,
        panel_button_image(panel, "Quit", 5, 0),
        PANEL_ITEM_X, ATTR_COL(20),
        PANEL_ITEM_Y, ATTR_ROW(4)-6,
        PANEL_NOTIFY_PROC, quit_proc, 0);

panel_create_item(panel, PANEL_BUTTON, PANEL_LABEL_IMAGE,
        panel_button_image(panel, "Read", 5, 0),
        PANEL_ITEM_X, ATTR_COL(28),
        PANEL_ITEM_Y, ATTR_ROW(4)-6,
        PANEL_NOTIFY_PROC, read_proc, 0);

panel_create_item(panel, PANEL_BUTTON, PANEL_LABEL_IMAGE,
        panel_button_image(panel, "New", 5, 0),
        PANEL_ITEM_X, ATTR_COL(36),
        PANEL_ITEM_Y, ATTR_ROW(4)-6,
        PANEL_NOTIFY_PROC, new_proc, 0);

panel_create_item(panel, PANEL_BUTTON, PANEL_LABEL_IMAGE,
        panel_button_image(panel, "Train", 6, 0),
        PANEL_ITEM_X, ATTR_COL(44),
        PANEL_ITEM_Y, ATTR_ROW(4)-6,
        PANEL_NOTIFY_PROC, train_proc, 0);

panel_create_item(panel, PANEL_BUTTON, PANEL_LABEL_IMAGE,
        panel_button_image(panel, "Open", 6, 0),
        PANEL_ITEM_X, ATTR_COL(53),
        PANEL_ITEM_Y, ATTR_ROW(4)-6,
        PANEL_NOTIFY_PROC, open_proc, 0);

}

static void
train_proc()/* input: file, file.pos. output: file.fea */
{
    Pixrect *image;
    FILE *fpi, *fopen();
    register inti=0, j=0, k=0;
    int x, y;
    short l;
    float fea[128][10];
    Pixwin *pw=canvas_pixwin(canvas);

    new_proc();

```



```

sprintf(cmdstring, "%s.pos", panel_get_value(flname_item));
fv=&cmdstring[0];
fpi = fopen(fv, "r");
while(i<512)
    fscanf(fp1, "%h1", &posvect[1++]);
fclose(fp1);

for(i=0; i<20; i++){
    mean[0][1]=0;
    mean[5][1]=0;
}

i=0;
while((x=posvect[1++])!=0){
    y=posvect[1++];
    pw_batch_on(pw);
    for(j=0; j<xsize+1; j++){
        for(k=0; k<ysize+1; k++){
            pcbpix[j][k]=pw_get(pw, x-bx2+j, y-by2+k);
        }
    }
    pw_batch_off(pw);
    features(1, x, y);
    k=i/2;
    for(j=0; j<7; j++){
        if(ll[0][j]>feavect[j]) ll[0][j]=feavect[j];
        if(mm[0][j]<feavect[j]) mm[0][j]=feavect[j];
        mean[0][j]=mean[0][j]+feavect[j];
        fea[k][j]=feavect[j];
    }
    pw_batch_on(pw);
    for(k=0; k<ysize+1; k++){
        for(j=0; j<xsize+1; j++){
            pw_put(pw, x-bx2+j, y-by2+k, 0);
        }
    }
    pw_batch_off(pw);
}

k=1/2;
for(i=0; i<7; i++){
    mean[0][i]=mean[0][1]/k;
    for(j=1; j<=k; j++){
        mean[5][i]=mean[5][i]+(fea[j][i]-mean[0][1])*(-
fea[j][1]-mean[0][1]);
        mean[5][i]=(float)sqrt((double)mean[5][1]/k);
    }
}
sprintf(cmdstring, "%s.fea", panel_get_value(flname_item));
fv=&cmdstring[0];
fpi = fopen(fv, "w");
for(i=0; i<7; i++){
    fprintf(fp1, "%f ", mean[0][i]);
    fprintf(fp1, "    for %3d samples\n", k);
}

for(i=0; i<7; i++)

```

```

        fprintf(fp1, "%f ", mean[5][1]);
        fprintf(fp1, "\n");

for(i=0; i<7; i++)
    fprintf(fp1, "%.4f% ", mean[5][i]/mean[0][1]*100);
    fprintf(fp1, "\n");

for(i=0; i<7; i++)
    fprintf(fp1, "%.4f ", mean[0][i]/mean[5][i]);
    fprintf(fp1, "\n");
fclose(fp1);

sprintf(cmdstring, "%s.fea", panel_get_value(fname_item));
fv=&cmdstring[0];
fp1 = fopen(fv, "a");
for(i=0; i<7; i++)
    fprintf(fp1, "%f ", mean[0][i]);
    fprintf(fp1, "\n");
fclose(fp1);

sprintf(cmdstring, "%s.lnm", panel_get_value(fname_item));
fv=&cmdstring[0];
fp1 = fopen(fv, "a");
for(i=0; i<7; i++) {
    fprintf(fp1, "%f ", ll[0][i]);
    ll[0][1]=1000;
}
    fprintf(fp1, "\n");
for(i=0; i<7; i++) {
    fprintf(fp1, "%f ", mm[0][i]);
    mm[0][1]=0;
}
    fprintf(fp1, "\n");
fclose(fp1);
}

static void
sort_proc()/* input: f, f.pos, f1.fea-f5.fea. output: f.cls */
{
    Pixrect *image;
    FILE *fp1, *fopen();
    register inti=0, j=0, k=0;
    int x, y, c, port[5];
    short l;
    float dist[5], dd;
    char cc;
    Pixwin *pw=canvas_pixwin(canvas);

    new_proc();

    sprintf(cmdstring, "weight%c.vec", value_to_c(panel_get_value(-
weight_num)));
    fv=&cmdstring[0];

```

```

fp1 = fopen(fv, "r");
for(i=0;i<7;i++){
    fscanf(fp1,"%f", &mean[0][i]);
    ll[0][i]=1000;
    mm[0][i]=0;
}
fclose(fp1);

sprintf(cmdstring, "%s.fea", panel_get_value(fname_item));
fv=&cmdstring[0];
fp1 = fopen(fv, "r");
for(i=1;i<5;i++){
    port[i]=0;
    for(j=0;j<7;j++){
        fscanf(fp1,"%f", &mean[1][j]);
    }
}
fclose(fp1);

sprintf(cmdstring, "%s.lnm", panel_get_value(fname_item));
fv=&cmdstring[0];
fp1 = fopen(fv, "r");
for(j=1;j<5;j++){
    for(i=0;i<7;i++) fscanf(fp1,"%f", &ll[j][i]);
    for(i=0;i<7;i++) fscanf(fp1,"%f", &mm[j][i]);
}
fclose(fp1);

for(j=0;j<7;j++){
    for(i=1;i<5;i++){
        if(ll[0][j]>ll[1][j]) ll[0][j]=ll[i][j];
        if(mm[0][j]<mm[i][j]) mm[0][j]=mm[i][j];
    }
}

for(i=1;i<5;i++)
    for(j=0;j<7;j++)
        mean[i][j]=mean[1][j]/mm[0][j];

i=0;
sprintf(cmdstring, "%s.pos", panel_get_value(flname_item));
fv=&cmdstring[0];
fp1 = fopen(fv, "r");
while(i<512)
    fscanf(fp1,"%hi", &posvect[i++]);
fclose(fp1);

sprintf(cmdstring, "%s.out", panel_get_value(flname_item));
fv=&cmdstring[0];
fp1 = fopen(fv, "w");

i=0;
while((x=posvect[i++])!=0){
    y=posvect[i++];
    pw_batch_on(pw);
}

```

```

for(j=0;j<xsize+1;j++)
    for(k=0;k<ysize+1;k++){
        pcbpix[j][k]=pw_get(pw, x-bx2+j, y-by2+k);
    }
pw_batch_off(pw);

features(1, x, y);

for(j=0;j<7;j++)
    feavect[j]=feavect[j]/mm[0][j];

for(k=1;k<5;k++){
    dist[k]=0;
    for(j=0;j<7;j++){
        dd=(mean[k][j]-feavect[j])*mean[0][j];
        dist[k]=dist[k]+dd*dd;
    }
    dist[k]=sqrt((double)dist[k]);
}
dd=1000;
for(k=1;k<5;k++){
    if(dd>dist[k]){
        dd=dist[k];
        c=k;
    }
}
port[c]++;
if(c==1)cc='1';
if(c==2)cc='2';
if(c==3)cc='3';
if(c==4)cc='4';
fprintf(fp1, "%3d %3d %.4f %.4f %.4f %.4f    %c \n",
        x, y, dist[1], dist[2], dist[3], dist[4], cc);
pw_batch_on(pw);
for(j=0;j<xsize+1;j++)
    for(k=0;k<ysize+1;k++)
        pw_put(pw, x-bx2+j, y-by2+k, 0);
pw_batch_off(pw);
}

fclose(fp1);

sprintf(cmdstring, "%s.sot", panel_get_value(fname_item));
fv=&cmdstring[0];
fp1 = fopen(fv, "a");
k=i/2;
for(i=1;i<5;i++)
    fprintf(fpi, "    %d %3.2f%%", i, (float)port[i]/k*100);
fprintf(fp1, "    for %3d samples\n", k);
fclose(fpi);

}

static void

```

```

size_proc()
{
    xsize=value_to_op(panel_get_value(box_xsize));
    ysize=value_to_op(panel_get_value(box_ysize));
    bx2=xsize/2;
    by2=ysize/2;
    cswx=(int)((float)bx2*.3);
    cswy=(int)((float)by2*.3);
    ofrx=(int)((float)bx2*.7);
    ofry=(int)((float)by2*.7);

    deltax=(float)1/(float)(xsize+1);
    deltax=(float)1/(float)(ysize+1);
    delta=deltax*deltay;
    deltx2=deltax*deltax;
    delty2=deltay*deltay;

}

static int
value_to_op(value)
register intvalue;
{
    switch (value) {
        case 0: return 8;
        case 1: return 12;
        case 2: return 14;
        case 3: return 16;
        case 4: return 18;
        case 5: return 20;
        case 6: return 22;
        case 7: return 24;
        case 8: return 30;
        case 9: return 36;
        case 10: return 40;
    }
}

static char
value_to_c(value)
register intvalue;
{
    switch (value) {
        case 0: return '1';
        case 1: return '2';
        case 2: return '3';
        case 3: return '4';
        case 4: return '5';
        case 5: return '6';
        case 6: return '7';
        case 7: return '8';
        case 8: return '9';
    }
}

```

```

}

static char
value_to_d(value)
register int value;
{
    switch (value) {
        case 0: return 'i';
        case 1: return 'v';
    }
}

static char
value_to_e(value)
register int    value;
{
    switch (value) {
        case 0: return 1;
        case 1: return 2;
        case 2: return 3;
        case 3: return 4;
        case 4: return 5;
        case 5: return 6;
        case 6: return 7;
        case 7: return 8;
        case 8: return 9;
    }
}

+

static void
help_proc(name)
char    *name;
{
    fprintf(stderr, "This program is %s", name);
    frame_cmdline_help(name);
}

static void
quit_proc()
{
    window_set(frame, FRAME_NO_CONFIRM, TRUE, 0);
    window_destroy(frame);
}

static void
ls_proc()
{
    char    cmdstring[256];
    sprintf(cmdstring, "ls -l %s*\n", panel_get_value(fname_item));
    ttysw_input(tty, cmdstring, strlen(cmdstring));
}

```

```
}

```

```
static void
pos_proc()
{
    int    x, y;

    x=(int) (LINT_CAST(panel_get_value(cursor_x)));
    y=(int) (LINT_CAST(panel_get_value(cursor_y)));
    window_set(canvas, WIN_MOUSE_XY, x, y, 0);
}

```

```
static void
fuse_proc(item, event)
Panel_item item;
Event      *event;
{
    FILE    *fp1, *fopen();
    register int i=0, j=0, k=0;
    int     x, y, c, port[5];
    short   l;
    float   dist[5], dd;
    char    cc;
    Pixwin  *pw=canvas_pixwin(canvas);

    vec=value_to_e(panel_get_value(fuse_vec));
    vec=vec/10;
    for(j=0; j<512; j++) {
        data[j]=0;
        data1[j]=0;
    }

    sprintf(cmdstring, "%s.out", panel_get_value(flname_item));
    fv=&cmdstring[0];
    fpi = fopen(fv, "r");
    while(i<512) {
        fscanf(fp1, "%f", &data[i++]);
    }
    fclose(fp1);

    l=0;
    sprintf(cmdstring, "%s.out", panel_get_value(f3name_item));
    fv=&cmdstring[0];
    fpi = fopen(fv, "r");
    while(i<512) {
        fscanf(fpi, "%f", &data1[l++]);
    }
    fclose(fpi);

    sprintf(cmdstring, "%s.fus", panel_get_value(flname_item));
}

```

```

fv=&cmdstring[0];
fpi = fopen(fv, "w");

i=0;
while((dd=data[i])!=0) {
    x=(int) data[i++];
    y=(int) data[i++];
    printf("x=%d ", x);
    for(k=1;k<5;k++) {
        dist[k]=vec*data[i]+(1-vec)*data1[i];
        i++;
    }

    i++;
    dd=1000;
    for(k=1;k<5;k++){
        if(dd>dist[k]) {
            dd=dist[k];
            c=k;
        }
    }

    port[c]++;
    if(c==1) cc='1';
    if(c==2) cc='2';
    if(c==3) cc='3';
    if(c==4) cc='4';

    fprintf(fpi, "%3d %3d %.4f %.4f %.4f %.4f   %c \n",
           x, y, dist[1], dist[2], dist[3], dist[4], cc);
}

fclose(fpi);
}

```

```

static void
draw(canvas_local)
Canvas canvas_local;
{
    struct pixrect *image;
    FILE *fpi, *fopen();
    colormap_t*colormap=NULL;
    Pixwin *pw=canvas_pixwin(canvas_local);

    sprintf(cmdstring, "%s", panel_get_value(fname_item));
    fv=&cmdstring[0];
    rh.ras_magic = RAS_MAGIC;
    rh.ras_width = 512;
    rh.ras_height = 484;
    rh.ras_depth = 8;
    rh.ras_length = 512*484;
    rh.ras_type = RT_STANDARD;
    rh.ras_maptype = RMT_NONE;
    rh.ras_maplength = 512;
    fpi = fopen(fv, "r");
    image = pr_load_std_image(fpi, &rh, colormap);
}

```



```

        pw_write(pw, 0, 0, 512, 484, PIX_SRC^PIX_DST, image, 0, 0);
    }

static void
read_proc()
{
    FILE    *fp1, *fopen();
    register short i=0;
    short   x, y, k;
    Pixwin  *pw=canvas_pixwin(canvas);
    i=pos;
    sprintf(cmdstring, "%s.pos", panel_get_value(fname_item));
    fv=&cmdstring[0];
    fp1 = fopen(fv, "r");
    while(i<512){
        fscanf(fp1,"%hi", &posvect[i++]);
    }
    fclose(fp1);
    i=0;
    while(posvect[i]!=0 || posvect[i+1]!=0) i+=2;
    pos=i;

    i=0;
    while(posvect[i]!=0 || posvect[i+1]!=0){
        x=posvect[i++];
        y=posvect[i++];
        pw_batch_on(pw);
        pw_vector(pw, x-bx2, y-by2, x+bx2, y-by2, PIX_SRC, 255);
        pw_vector(pw, x+bx2, y-by2, x+bx2, y+by2, PIX_SRC, 255);
        pw_vector(pw, x-bx2, y-by2, x-bx2, y+by2, PIX_SRC, 255);
        pw_vector(pw, x-bx2, y+by2, x+bx2, y+by2, PIX_SRC, 255);
        pw_batch_off(pw);
    }
    window_set(canvas, WIN_MOUSE_XY, 256, 242, 0);
}

static void
clear_proc()
{
    FILE    *fpi, *fopen();
    register short i=0;
    short   x, y, k;

    sprintf(cmdstring, "%s.sot", panel_get_value(fname_item));
    fv=&cmdstring[0];
    fp1 = fopen(fv, "w");
    fclose(fp1);
}

static void

```

```

open_proc()
{
    FILE    *fpi, *fopen();
    register shorti=0;
    short   x, y, k;

    sprintf(cmdstring, "%s.fea", panel_get_value(fname_item));
    fv=&cmdstring[0];
    fpi = fopen(fv, "w");
    fclose(fpi);

    sprintf(cmdstring, "%s.lnm", panel_get_value(fname_item));
    fv=&cmdstring[0];
    fpi = fopen(fv, "w");
    fclose(fpi);

    for(i=0;i<20;i++){
        ll[0][i]=1000.0;
        mm[0][i]=0;
    }
}

static void
new_proc()
{
    register inti=0;
    Pixwin *pw=canvas_pixwin(canvas);

    while(i<512)
        posvect[i++]=0;
    pos=0;
    pw_writebackground(pw, 0, 0, 512, 484, PIX_SRC);
    draw(canvas);
}

static void
handle_event(canvas_local, event, arg)
Canvas canvas_local;
Event *event;
caddr_t arg;
{
    Pixwin *pw=canvas_pixwin(canvas);
    register inti=0, j=0, k;
    short   x, y;

    if (event_is_up(event))
        return;

    x=event_x(event);
    y=event_y(event);
    k=event_id(event);
}

```

```

switch(k) {
    case 'w':
        y--;
        j=1;
        break;
    case 'a':
        x--;
        j=1;
        break;
    case 's':
        x++;
        j=1;
        break;
    case 'z':
        y++;
        j=1;
        break;
    case 'e':
        while(posvect[i]!=0 || posvect[i+1]!=0) i+=2;
        posvect[i++]=x;
        posvect[i++]=y;
        pw_batch_on(pw);
        pw_vector(pw, x-bx2, y-by2, x+bx2, y-by2, PIX_SRC, 255);
        pw_vector(pw, x+bx2, y-by2, x+bx2, y+by2, PIX_SRC, 255);
        pw_vector(pw, x-bx2, y-by2, x-bx2, y+by2, PIX_SRC, 255);
        pw_vector(pw, x-bx2, y+by2, x+bx2, y+by2, PIX_SRC, 255);
        pw_batch_off(pw);
    break;

    case 'q':
        while(posvect[i]!=0 || posvect[i+1]!=0) i+=2;
        i--; k=0;
        while(i>0) {
            if(abs(y-posvect[i]) < bx2 && abs(x-posvect[i-1])
< bx2) {
                k=1; i=0;
            }
            else i-=2;
        }
        if(k!=0) {
            while(k<254) {
                posvect[k-1]=posvect[k+1];
                posvect[k]=posvect[k+2];
                k+=2;
            }
            pw_writebackground(pw, 0, 0, 512, 484, PIX_SRC);
            draw(canvas);
            while(posvect[i]!=0 || posvect[i+1]!=0) {
                x=posvect[i++];
                y=posvect[i++];
                pw_batch_on(pw);
                pw_vector(pw, x-bx2, y-by2, x+bx2, y-by2,
PIX_SRC, 255);
                pw_vector(pw, x+bx2, y-by2, x+bx2, y+by2,

```

```

PIX_SRC, 255);
                                pw_vector(pw, x-bx2, y-by2, x-bx2, y+by2,
PIX_SRC, 255);
                                pw_vector(pw, x-bx2, y+by2, x+bx2, y+by2,
PIX_SRC, 255);
                                pw_batch_off(pw);
                                }
                                }
                                break;

                                default:
                                    break;
                                }
if(j==1){
    window_set(canvas, WIN_MOUSE_XY, x, y, 0);
    return;
}

switch (event_action(event)) {
case MS_LEFT:
case MS_MIDDLE:
case MS_RIGHT:
    /* translate the event to window space,
     * then show the menu.
     */
    (void) menu_show(menu, canvas_local,
                    canvas_window_event(canvas_local, event), 0);
    break;

default:
    break;
}

}

static void
output_proc()

{
    FILE    *fpi, *fopen();
    register inti=0, k;
    int     o, p, r;
    short   m, n, l, q;

    while(posvect[i]!=0 || posvect[i+1]!=0) i+=2;
    p=i-1;

    o=1;
    r=1;

while(o<p) {

```

```

l=posvect[0];
for(i=0+2;i<=p;i+=2)
    if(abs(posvect[i]-1)>20) break;
r=1;
for(i=r+2;i<=p;i+=2){
    if(abs(posvect[i]-1)<=20){
        m=posvect[i-1];
        n=posvect[i];
        posvect[i-1]=posvect[r-1];
        posvect[i]=posvect[r];
        posvect[r-1]=m;
        posvect[r]=n;
        r+=2;
    }
}
for(i=0;i<r;i+=2){
    q=posvect[i-1];
    for(k=i+2;k<r;k+=2)
        if(posvect[k-1]<q){
            m=posvect[k-1];
            n=posvect[k];
            posvect[k-1]=posvect[i-1];
            posvect[k]=posvect[i];
            posvect[i-1]=m;
            posvect[i]=n;
            q=m;
        }
}
o=r;
}

i=0;
sprintf(cmdstring, "%s.pos", panel_get_value(filename_item));
fv=&cmdstring[0];
fpi = fopen(fv, "w");
while(posvect[i]!=0 || posvect[i+1]!=0){
    fprintf(fp1,"%d ",posvect[i++]);
    fprintf(fp1,"%d\n",posvect[i++]);
}
fclose(fp1);
}

static void
features(oi, ox, oy)/* output: file.fea */
int    oi, ox, oy;
{
    FILE    *fp1, *fopen();
    register inti=0, j=0, k=0;
    short   l, m;
    float   midd, vtot, sigma2, vleft, vright, dlleft, dlright,
            vcsw, vofr, x, y, x2, y2;

    float   sum;
    float   xcm, ycm, iaa, ibb, icc, iratio;

```

```

float  ia, ib, ic, id, mij, zz;

    sprintf(cmdstring, "%c", value_to_c(panel_get_value(weight_
num)));
    m=cmdstring[0];
/*    printf("weightinput = %c ",m);
*/    sprintf(cmdstring, "%c", value_to_d(panel_get_value(visual_in-
frared)));
    m=cmdstring[0];
/*    printf("input = %c  %d",m ,m);
*/
if(m==105) {

/* normalization */
    l=255;
    m=0;
    for(j=0;j<xsize+1;j++)
        for(i=0;i<ysize+1;i++) {
            if(l>pcbpix[j][i])l=pcbpix[j][i];
            if(m<pcbpix[j][i])m=pcbpix[j][i];
        }
    for(j=0;j<xsize+1;j++)
        for(i=0;i<ysize+1;i++)
            sjpix[j][i]=(float)pcbpix[j][i]/m;

/* Mean grey level, Normalized volume */
    sum=0;
    for(j=0;j<xsize+1;j++)
        for(i=0;i<ysize+1;i++)
            sum=sum+sjpix[j][i];
    vtot=sum*delta;
    iaa=vtot;

/* Central subwindow mean */
    sum=0;
    for(j=bx2-cswx;j<by2+cswy+1;j++)
        for(i=bx2-cswx;i<by2+cswy+1;i++)
            sum=sum+sjpix[j][i];
    vcsw=sum*delta;

/* Outer frame region volume */
    sum=0;
    for(j=bx2-ofrx;j<by2+ofry+1;j++)
        for(i=bx2-ofrx;i<by2+ofry+1;i++)
            sum=sum+sjpix[j][i];
    vofr=sum*delta;
    vofr=vtot-vofr;
    iratio=vofr;

/*middle frame region volume */
    midd=vtot-vcsw-vofr;
    icc=midd;

```

```

/* Variance of normalized grey level */
sum=0;
for(j=0;j<xsize+1;j++)
    for(i=0;i<yssize+1;i++)
        sum=sum+(sjpix[j][i]-midd)*(sjpix[j][i]-midd);
vtot=sum/((xsize+1)*(ysize+1)-1);

/* Central subwindow variance */
sum=0;
for(j=bx2-cswx;j<by2+cswy+1;j++)
    for(i=bx2-cswx;i<by2+cswy+1;i++)
        sum=sum+(sjpix[j][i]-midd)*(sjpix[j][i]-midd);
vcsw=sum*delta;

/* Outer frame region variance */
sum=0;
for(j=bx2-ofrx;j<by2+ofry+1;j++)
    for(i=bx2-ofrx;i<by2+ofry+1;i++)
        sum=sum+(sjpix[j][i]-midd)*(sjpix[j][i]-midd);
vofr=sum*delta;
sigma2=vofr-vcsw;
vofr=vtot-vofr;

/* Inertia Features */

}

else{

/* normalization */
l=255;
m=0;
for(j=0;j<xsize+1;j++)
    for(i=0;i<yssize+1;i++){
        if(l>pcbpix[j][i])l=pcbpix[j][i];
        if(m<pcbpix[j][i])m=pcbpix[j][i];
    }
for(j=0;j<xsize+1;j++)
    for(i=0;i<yssize+1;i++)
        sjpix[j][i]=(float)pcbpix[j][i]/m;

/* Mean Grey level, Normalized volume */
sum=0;
for(j=0;j<xsize+1;j++)
    for(i=0;i<yssize+1;i++)
        sum=sum+sjpix[j][i];
vtot=sum*delta;

/* Variance of normalized grey level */
sum=0;
for(j=0;j<xsize+1;j++)
    for(i=0;i<yssize+1;i++)
        sum=sum+(sjpix[j][i]-vtot)*(sjpix[j][i]-vtot);
sigma2=sum/((xsize+1)*(ysize+1)-1);

```

```

/* left half area mean grey level */
sum=0;
for(j=0;j<bx2+1;j++)
    for(i=0;i<ysize+1;i++)
        sum=sum+sjpix[j][i];
vleft=sum/(ysize+1)/bx2;

/* right half area mean grey level */
sum=0;
for(j=bx2+1;j<xsize+1;j++)
    for(i=0;i<ysize+1;i++)
        sum=sum+sjpix[j][i];
vright=sum/(ysize+1)/bx2;

/* left half area variance */
sum=0;
for(j=0;j<bx2+1;j++)
    for(i=0;i<ysize+1;i++)
        sum=sum+(sjpix[j][i]-vleft)*(sjpix[j][i]-vleft);
dltleft=sum/ysize/bx2;

/* right half area variance */
sum=0;
for(j=bx2+1;j<xsize+1;j++)
    for(i=0;i<ysize+1;i++)
        sum=sum+(sjpix[j][i]-vright)*(sjpix[j][i]-
vright);
dltright=sum/ysize/bx2;

}

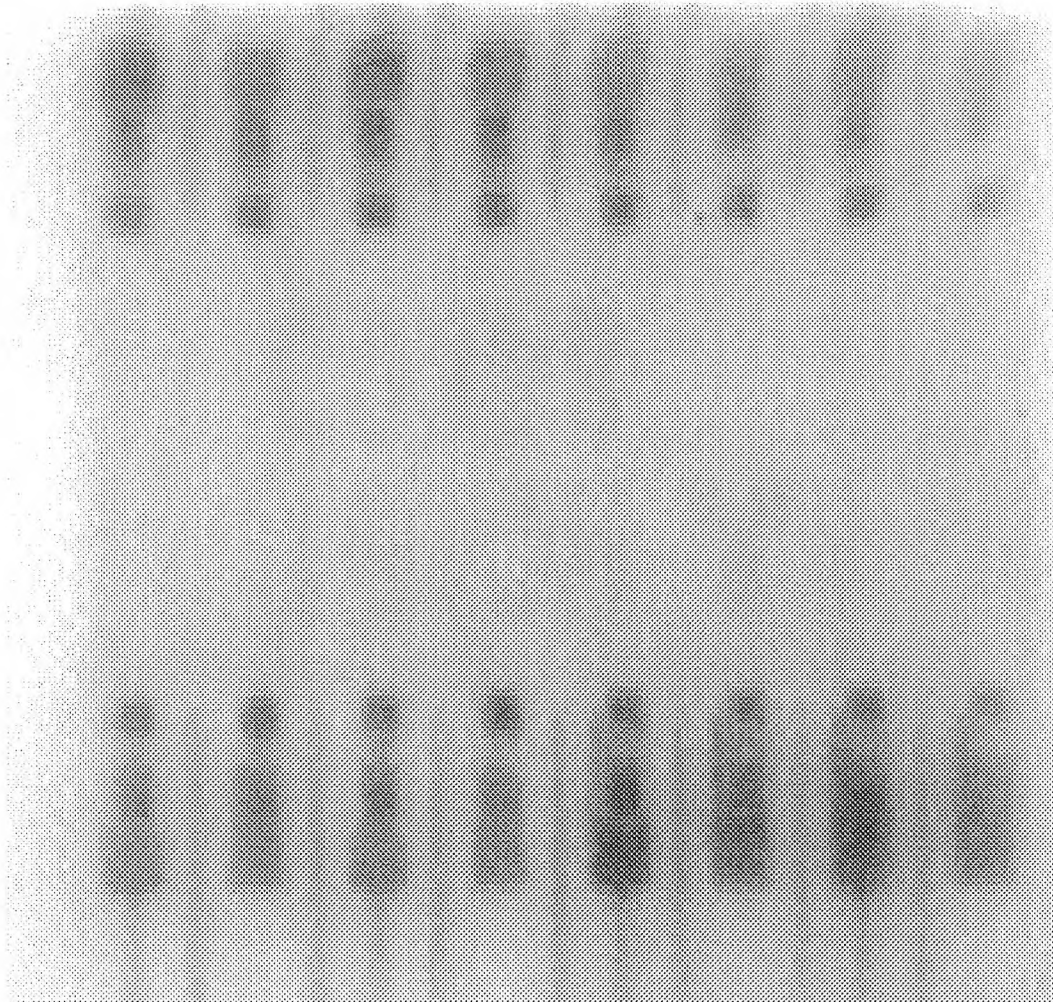
feavect[0]=vleft;
feavect[1]=vright;
feavect[2]=dltleft;
feavect[3]=dltright;
feavect[4]=1.0;
feavect[5]=1.0;
feavect[6]=1.0;

}

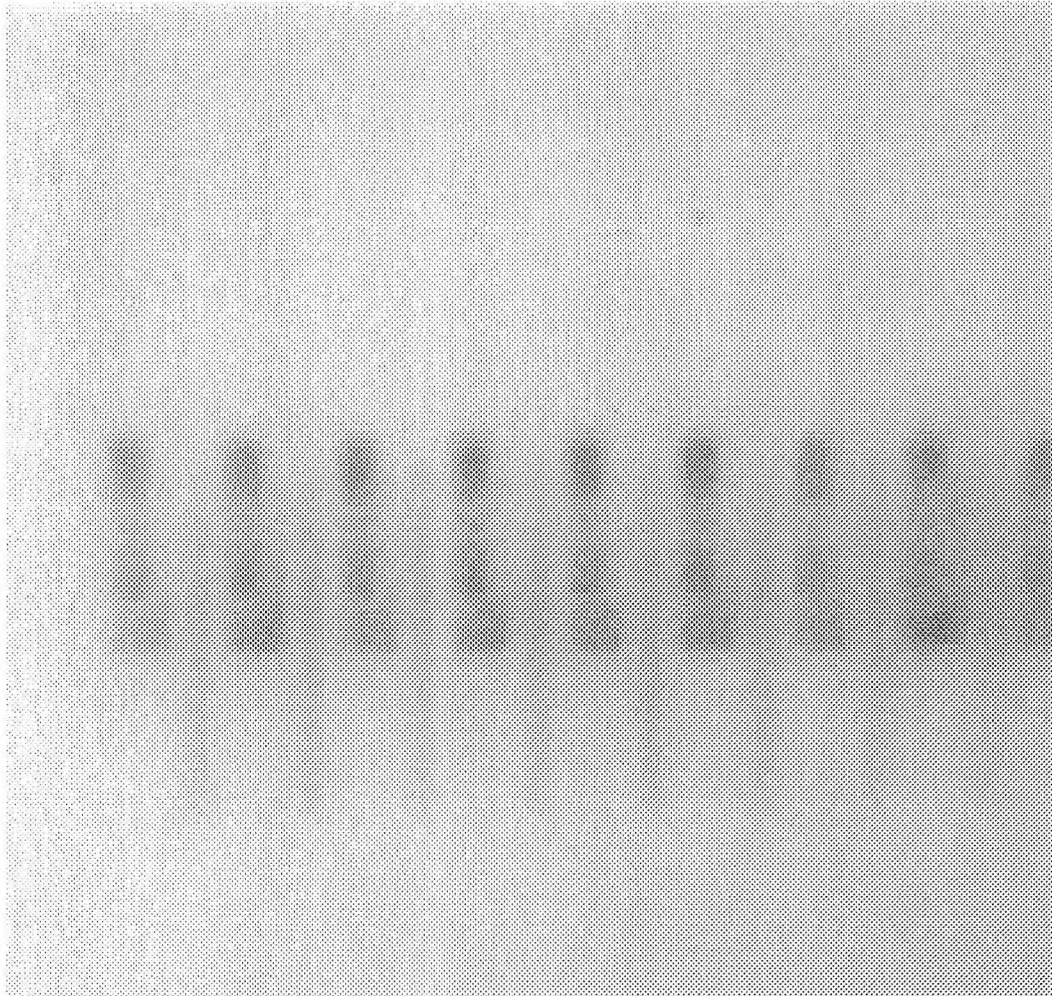
```



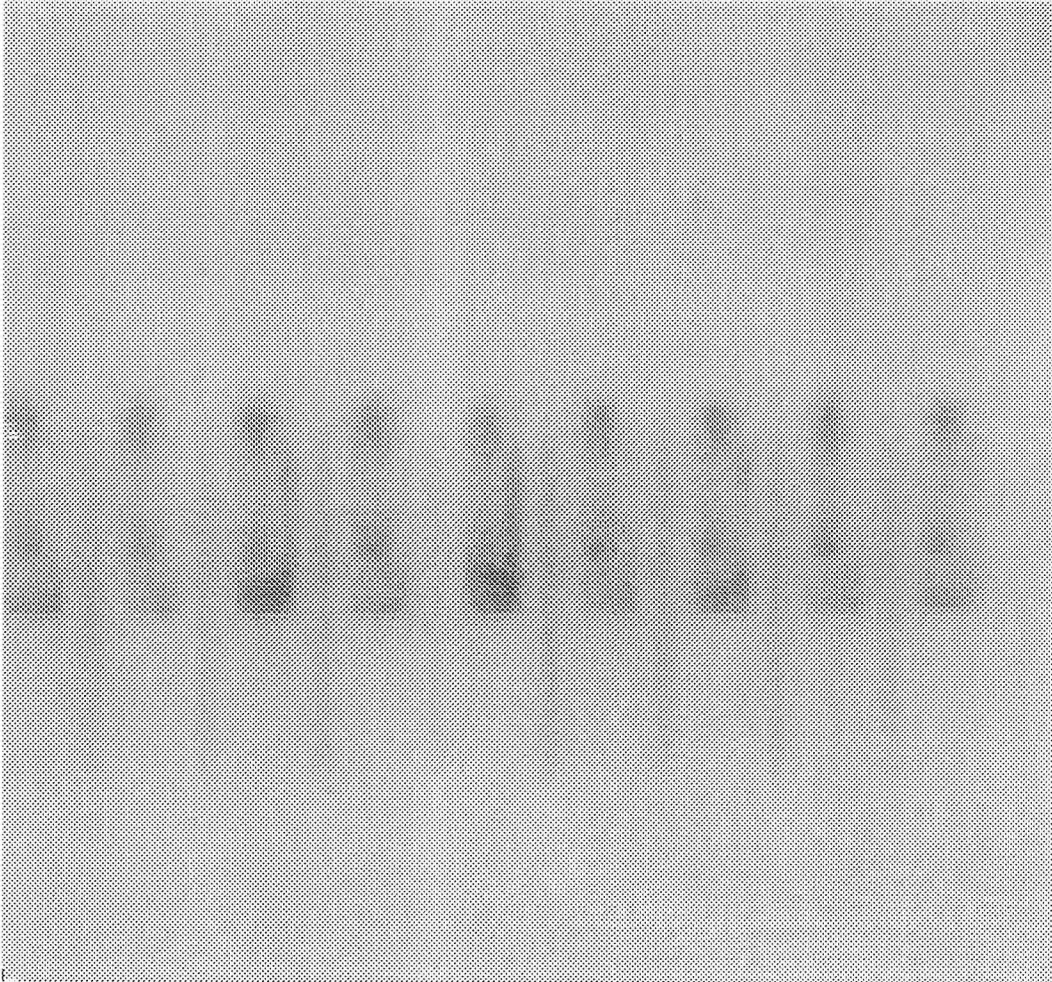
## B Infrared Images



i16.1: The darkest four joints at the lower right corner are scraped to no paste. the upper left corner four joints are scraped to half paste volume, while the rest are with normal paste volume.

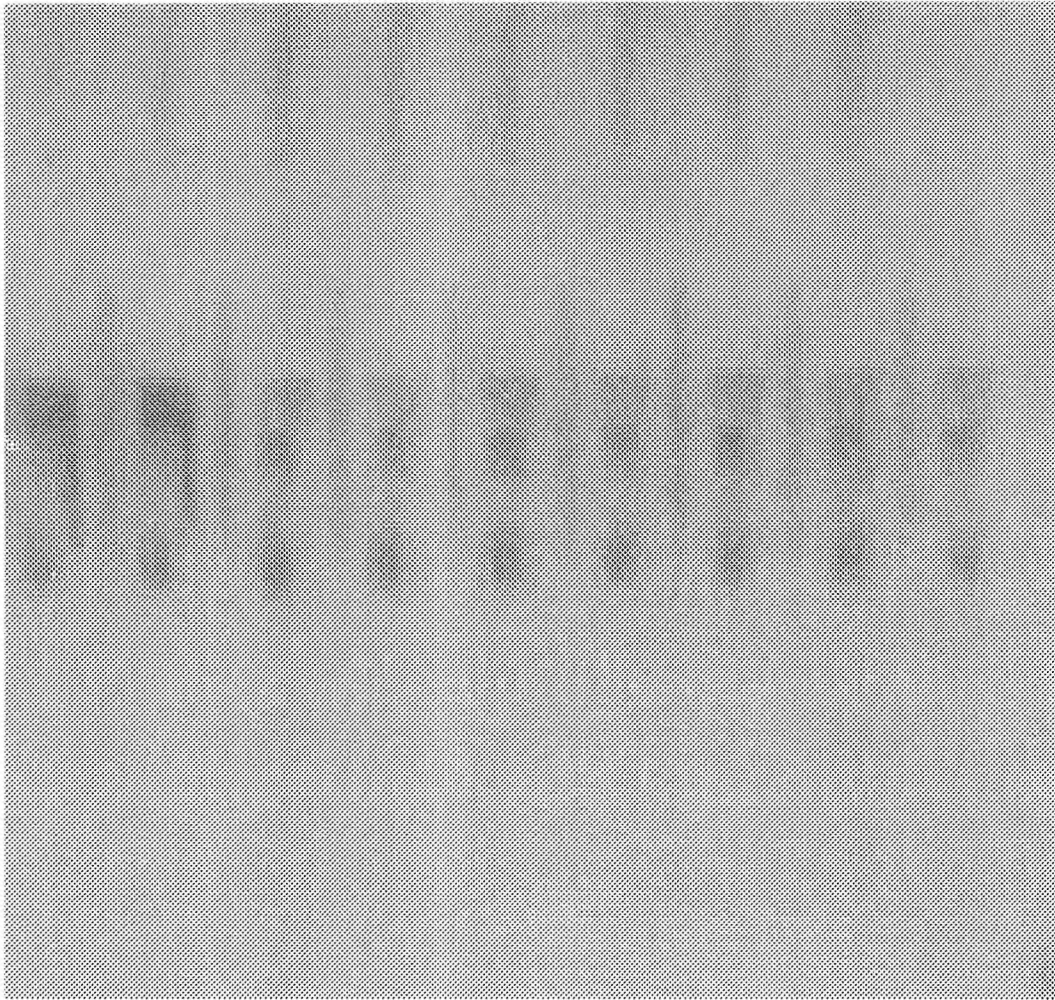


i28.1\_1-7: The left seven joints, number 1 to number 7, are all with normal paste volume.

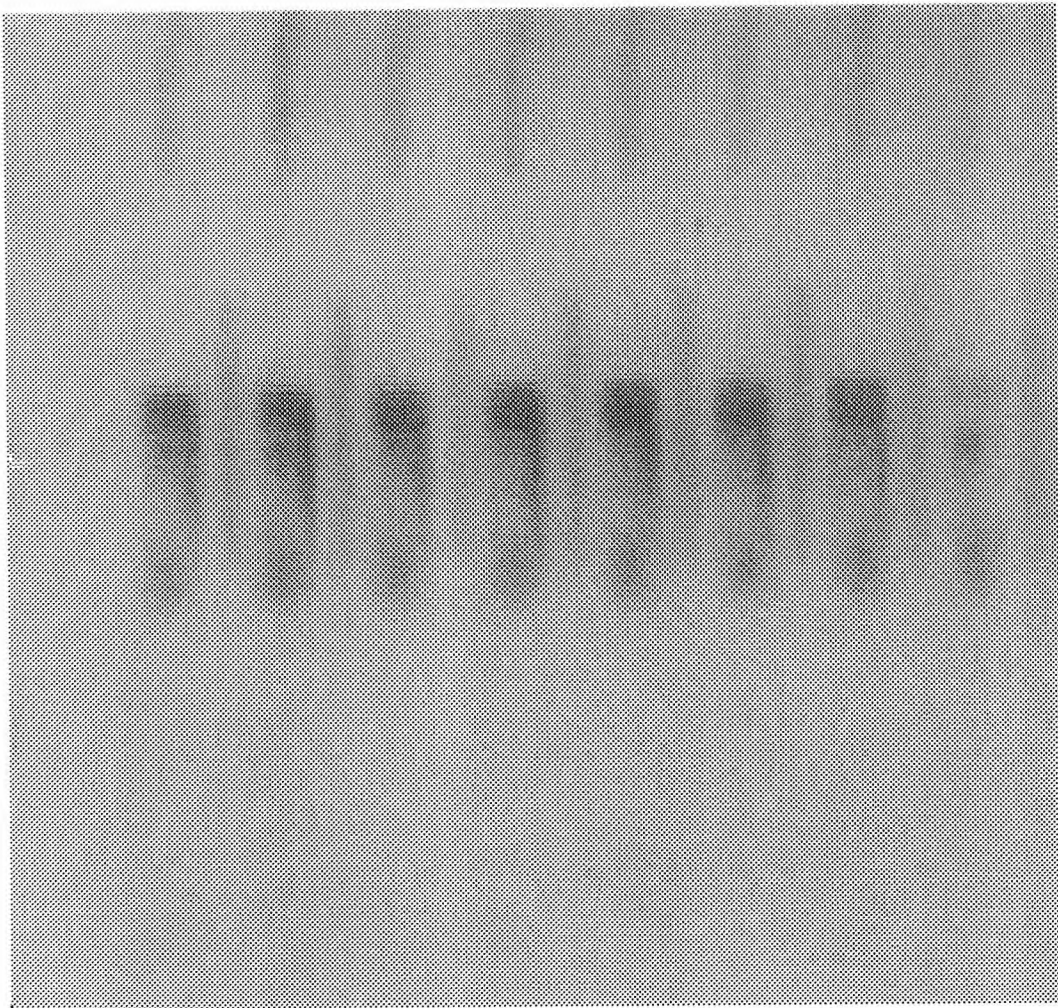


i28.1\_8-14: The right 7 joints, pin 8 to pin 14, are all scraped to half paste volume.

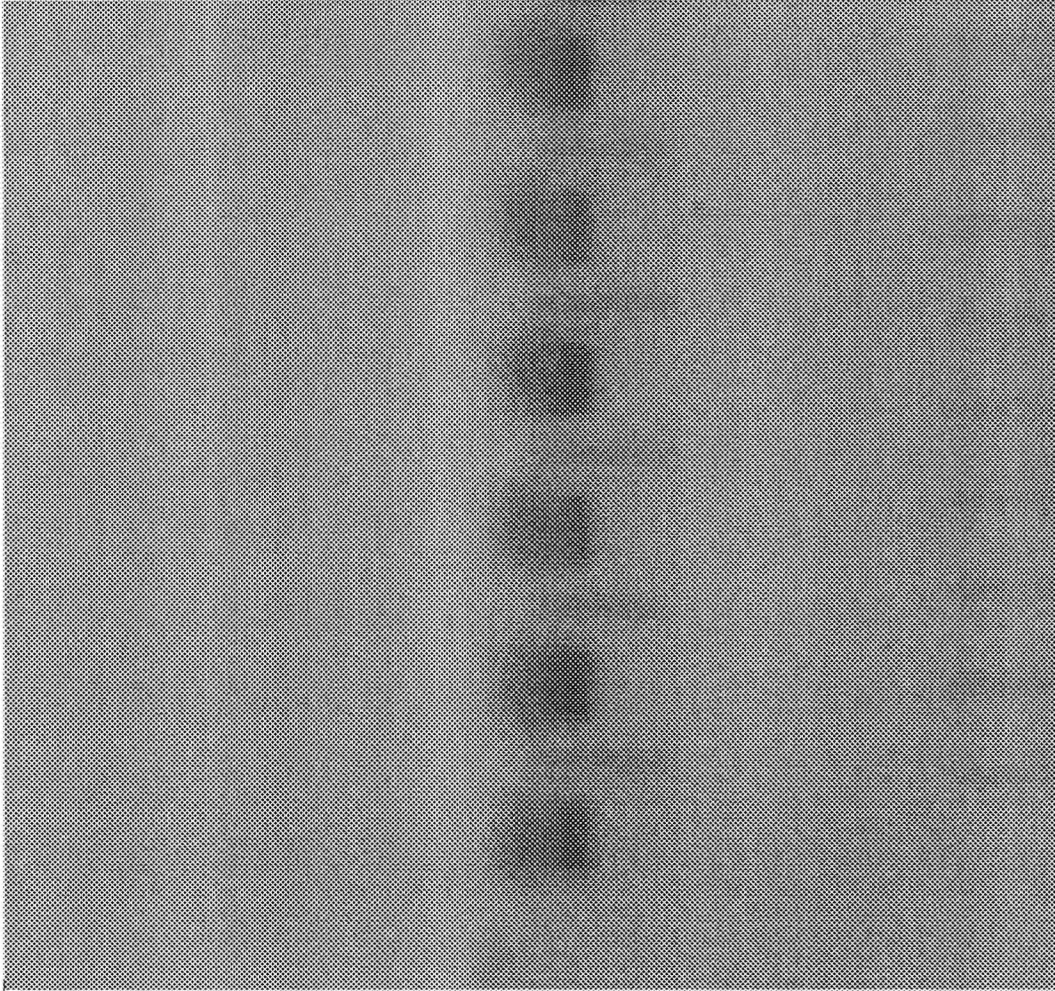




v28.1\_15-21: The right seven joints, pin 15 to pin 21, are with normal paste.

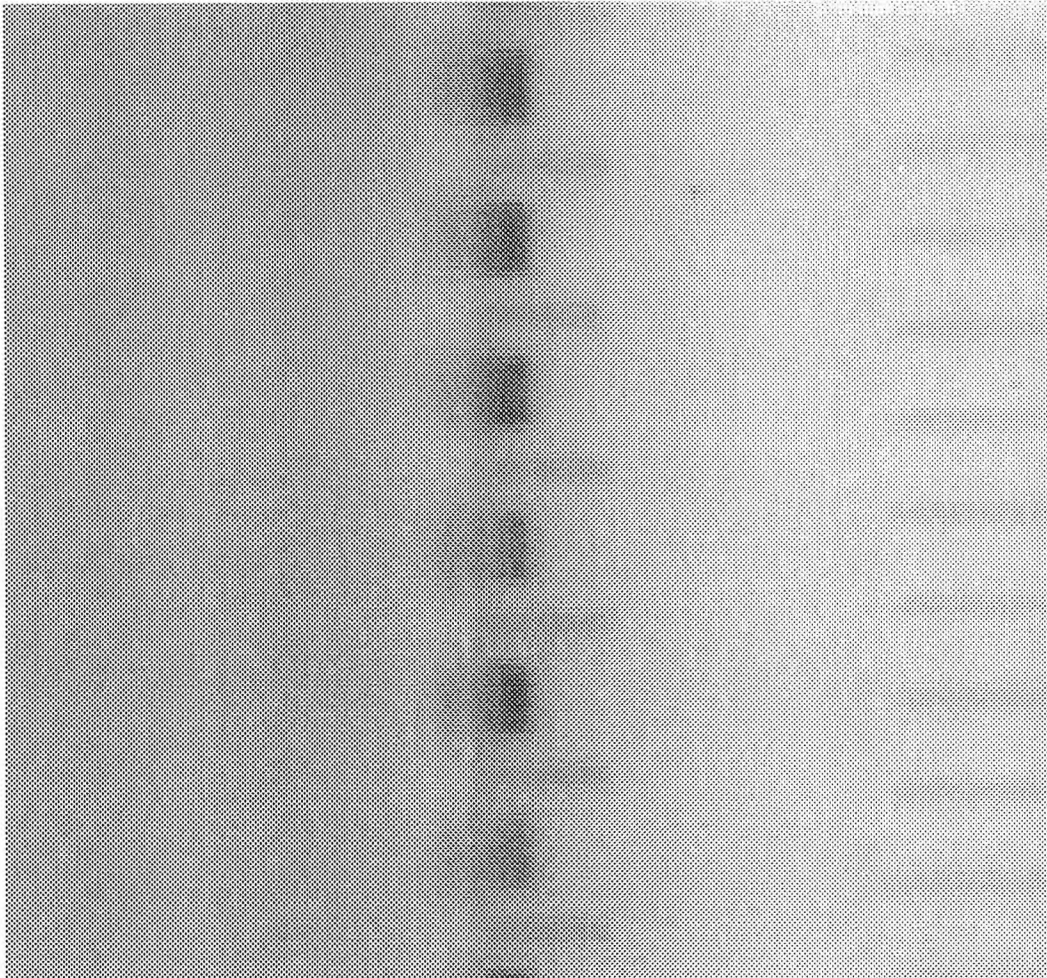


i28.1\_22-28: The left seven joints, pin 22 to pin 28, are scraped to no paste.

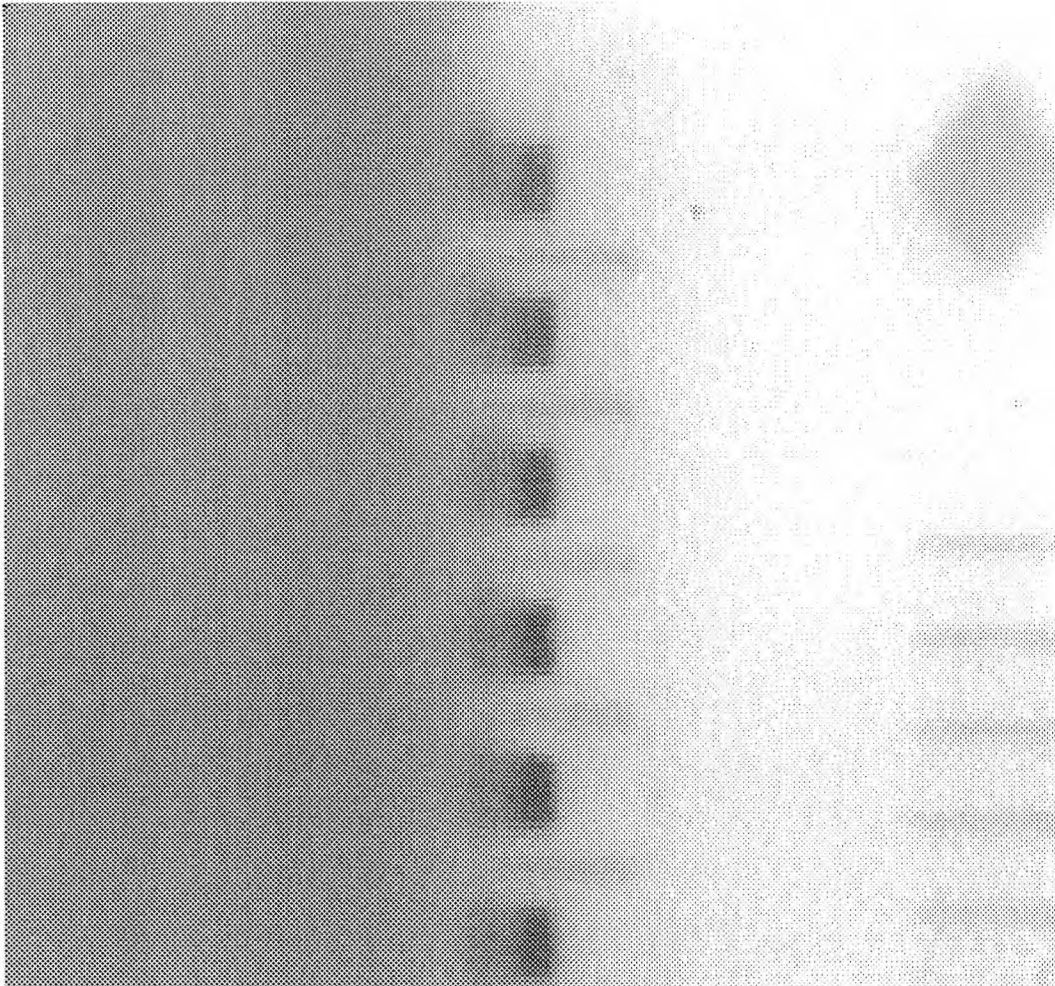


i68.1\_1-6: These six joints, pin 1 to pin 6 form bottom to top, are with normal paste.



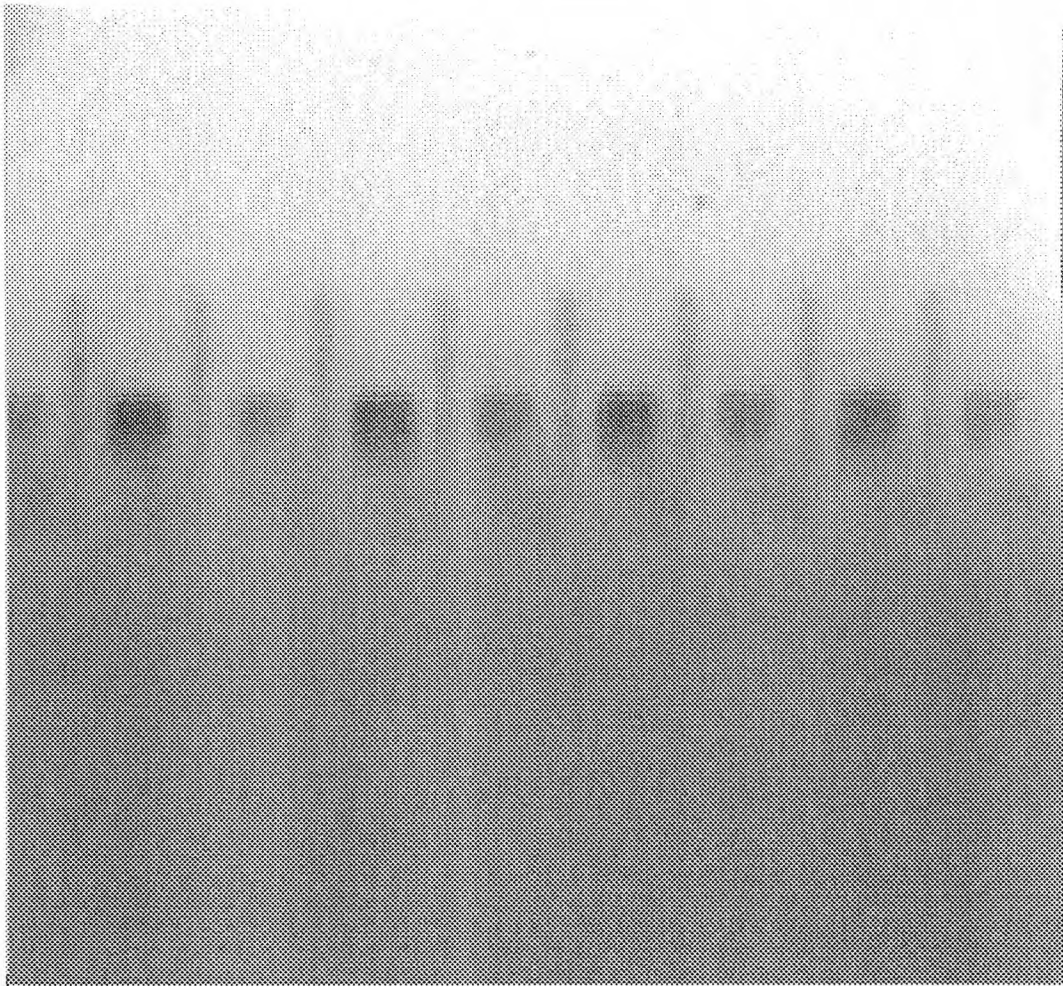


i68.1\_7-12: The lower three joints are with normal paste volume, the upper three joints are scraped to no paste.

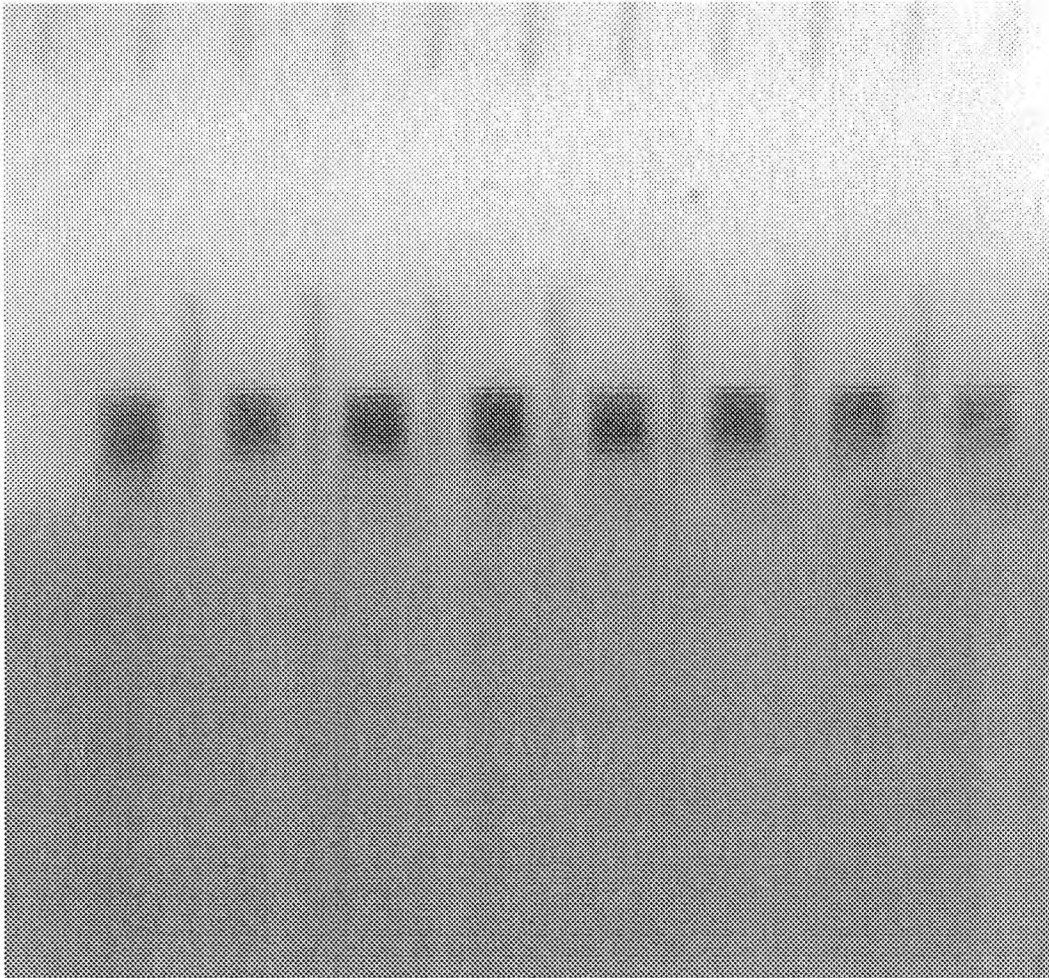


i68.1\_13-17: These six joints are all scraped to no paste.

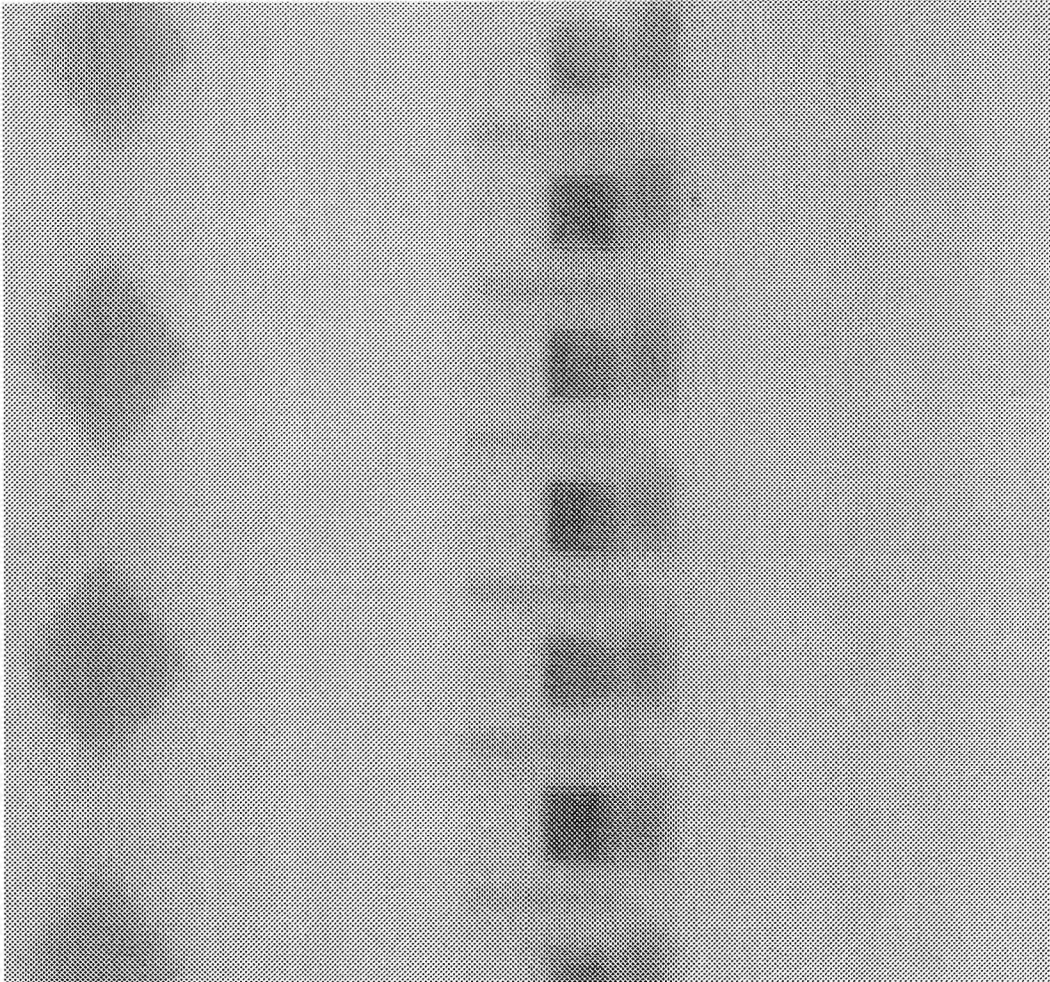




i68.1\_18-25: All of these eight joints, pin 18 to pin 25, are with normal paste.

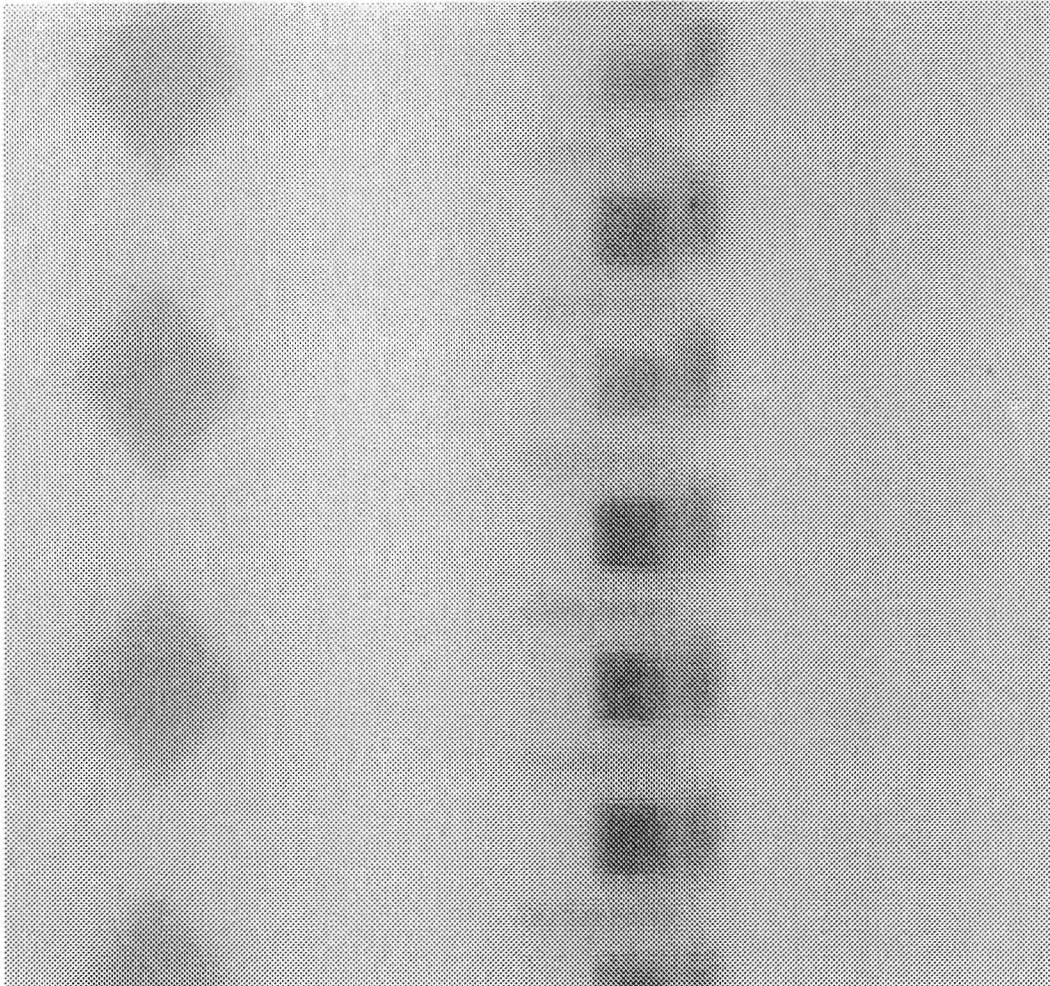


i68.1\_27-34: All of the joints in this image are scraped to half paste volume.

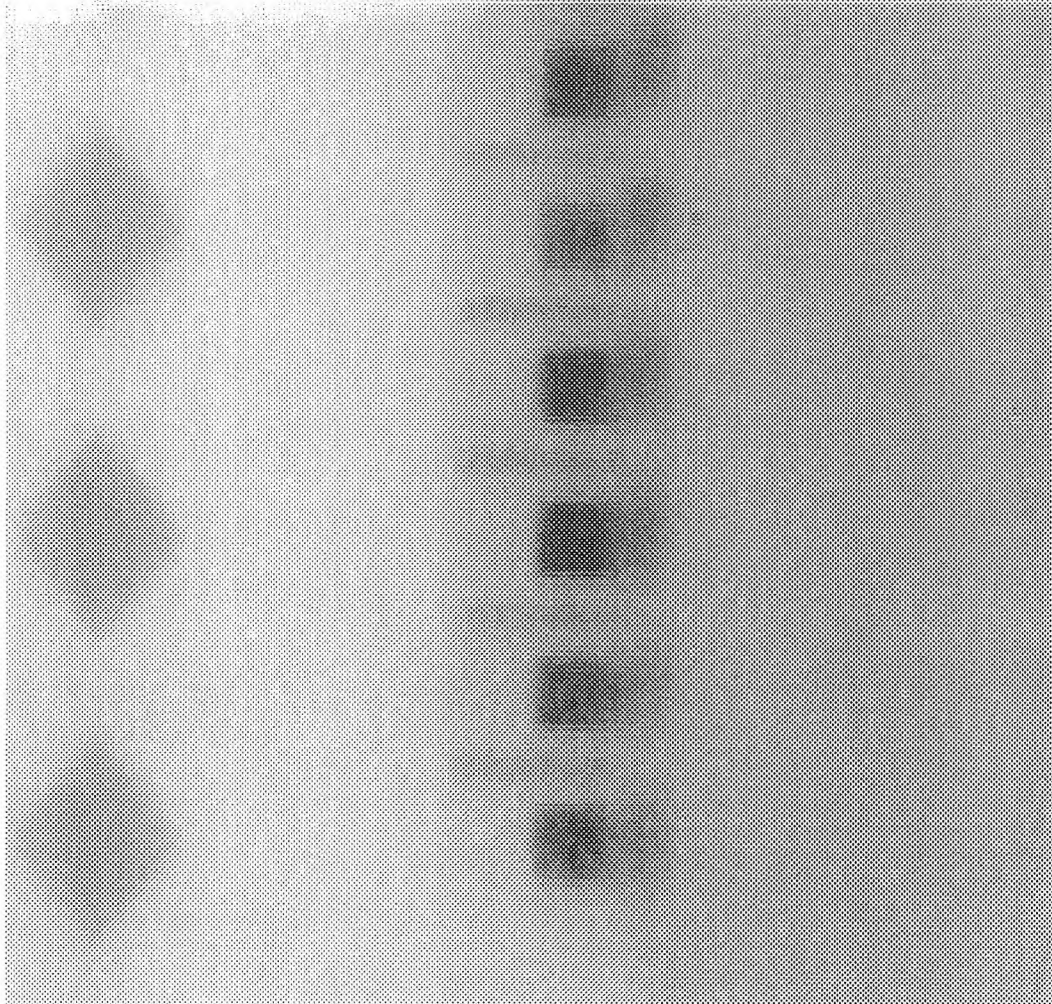


i68.1\_35-40: The upper six joints are with normal paste volume.

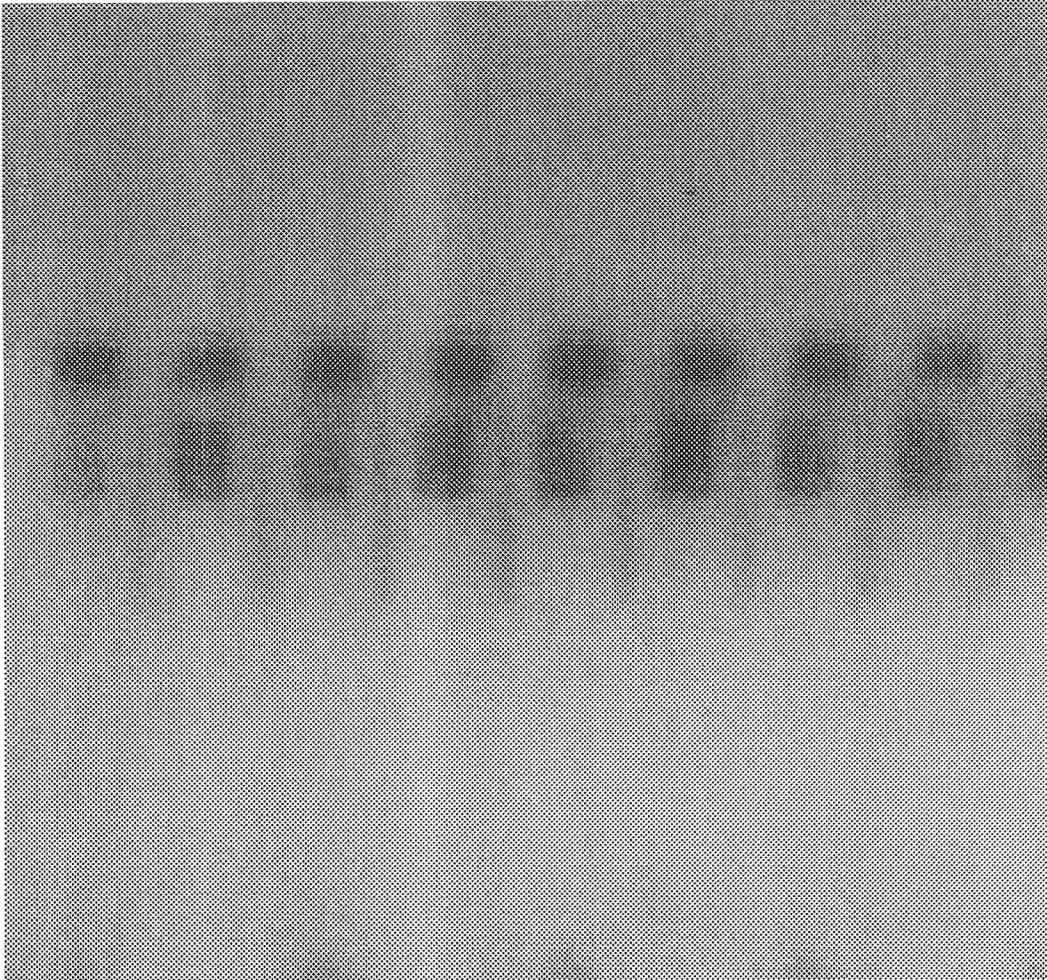




i68.1\_41-46: The upper three joints are with normal paste volume, while the rest are all scraped to half paste volume.

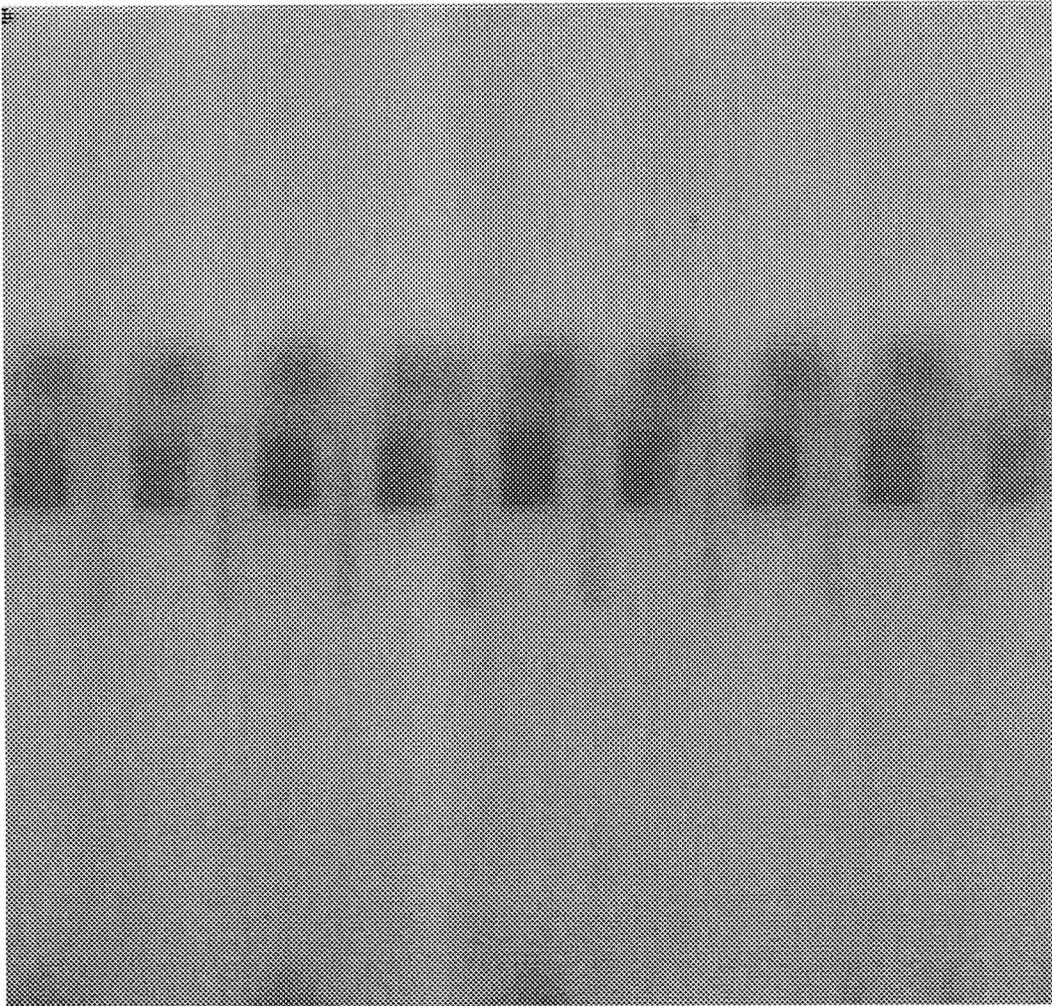


i68.1\_47-51: All the joints are scraped to half paste volume.

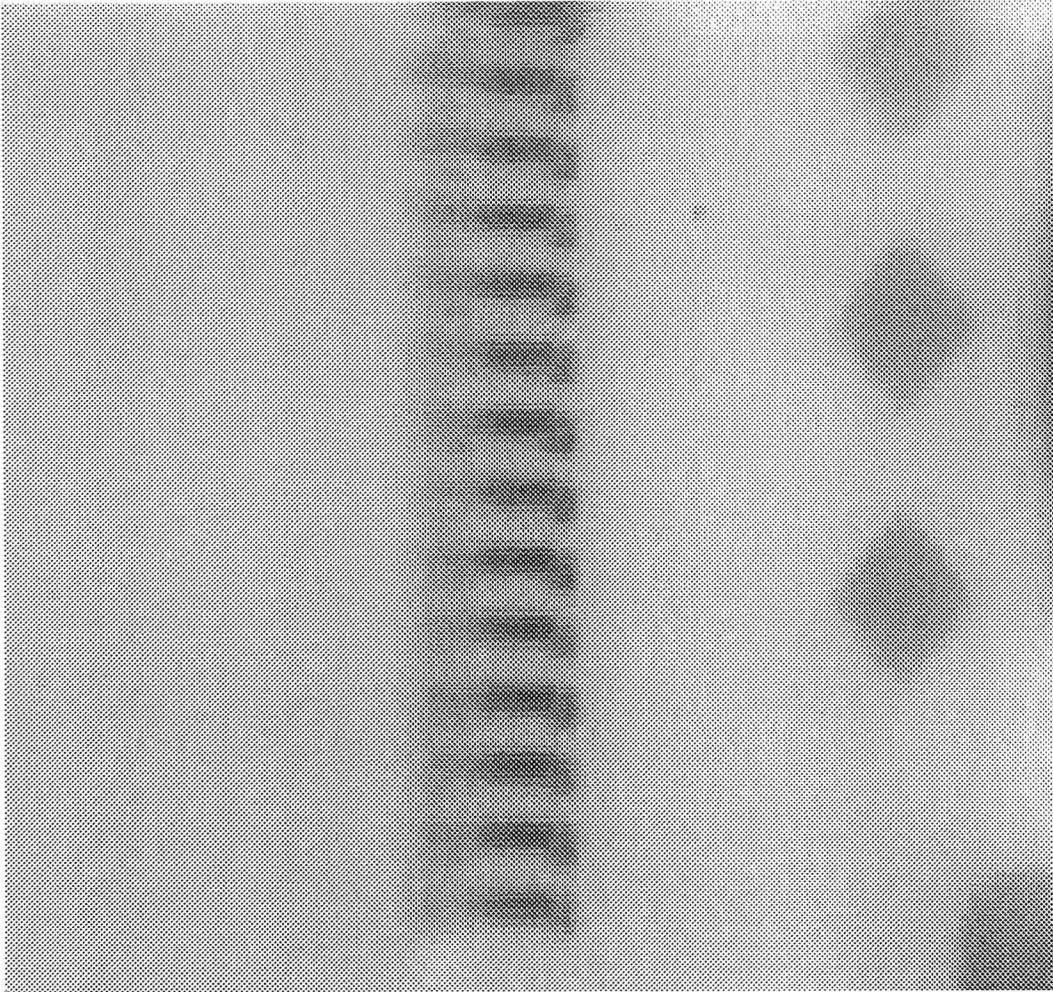


i68.1\_52-60: The left eight joints, pin 52 to pin 60, are all with normal paste volume.



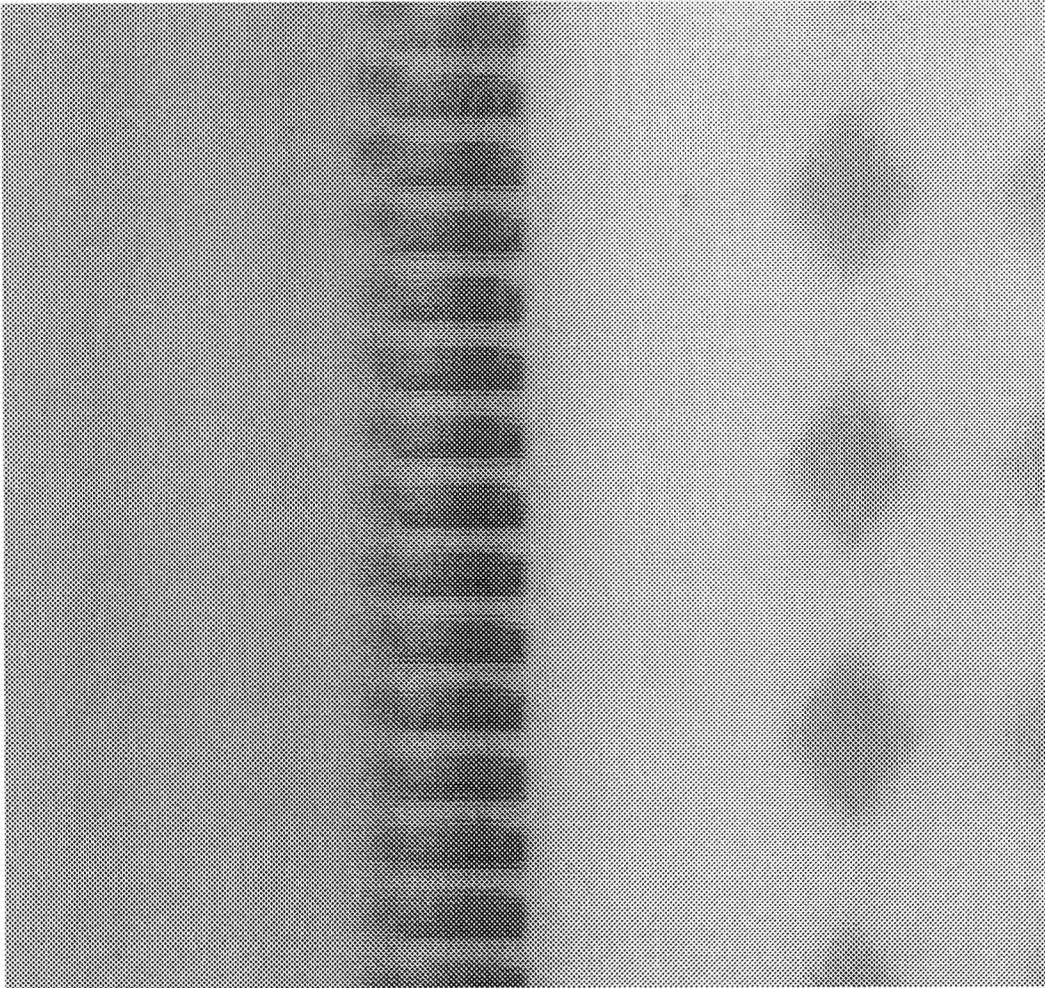


i68.1\_61-68: The right eight joints, pin 61 to pin 68, are scraped to no paste.

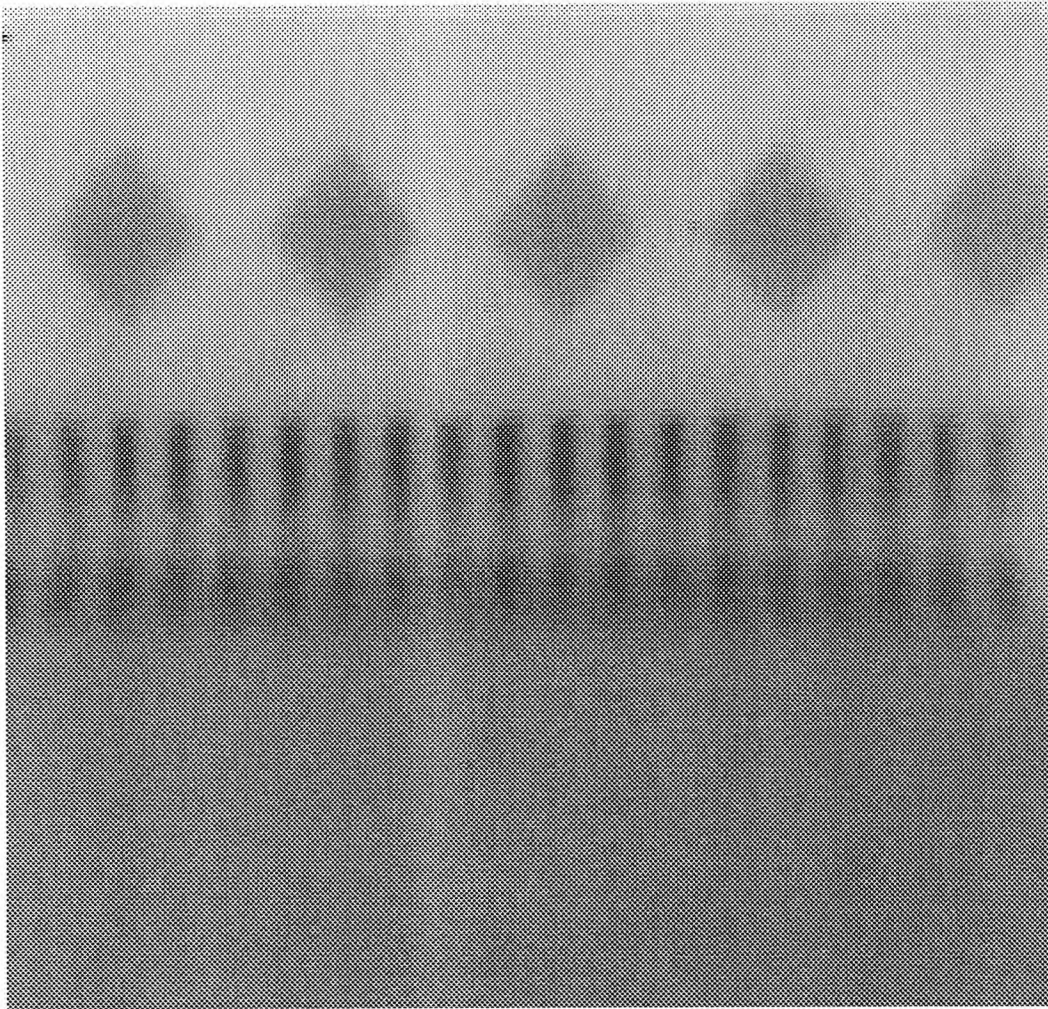


i132.4\_1-17; All the joints are with normal paste volume.

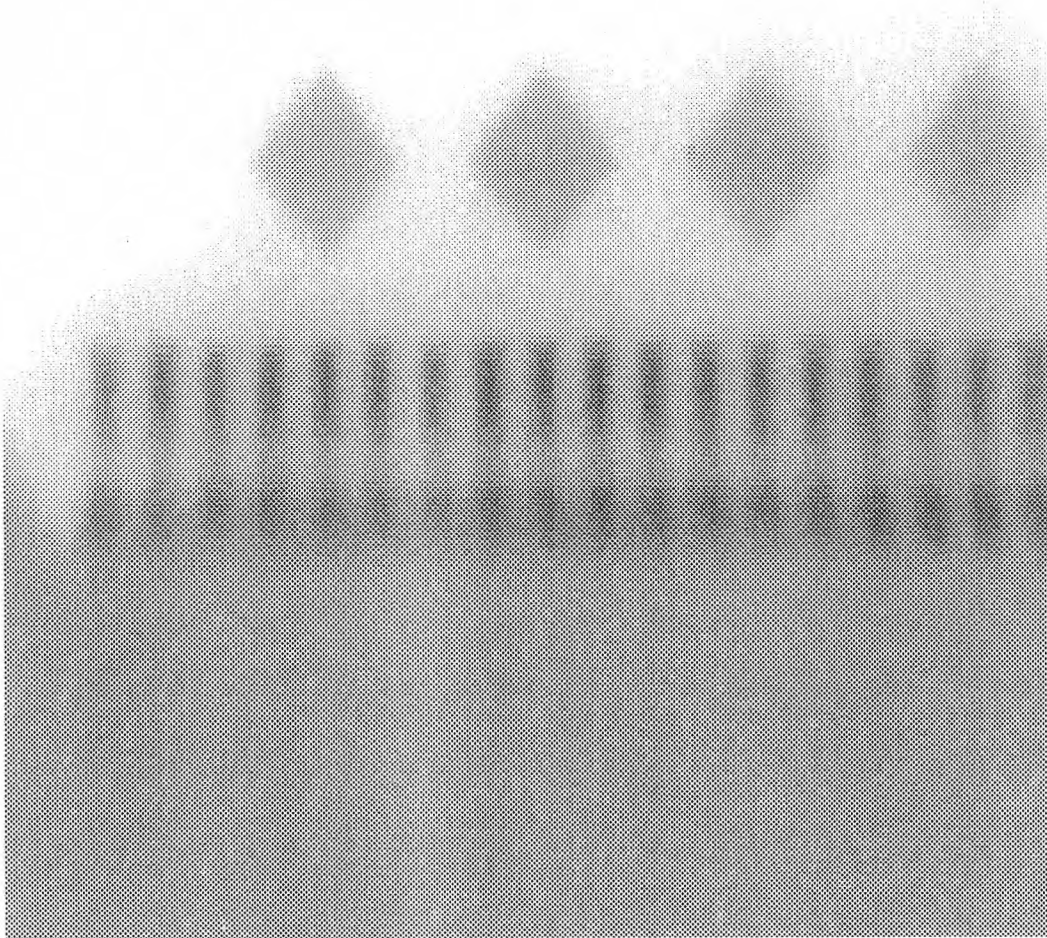




i132.4\_18-33: All the joints are scraped to no paste.

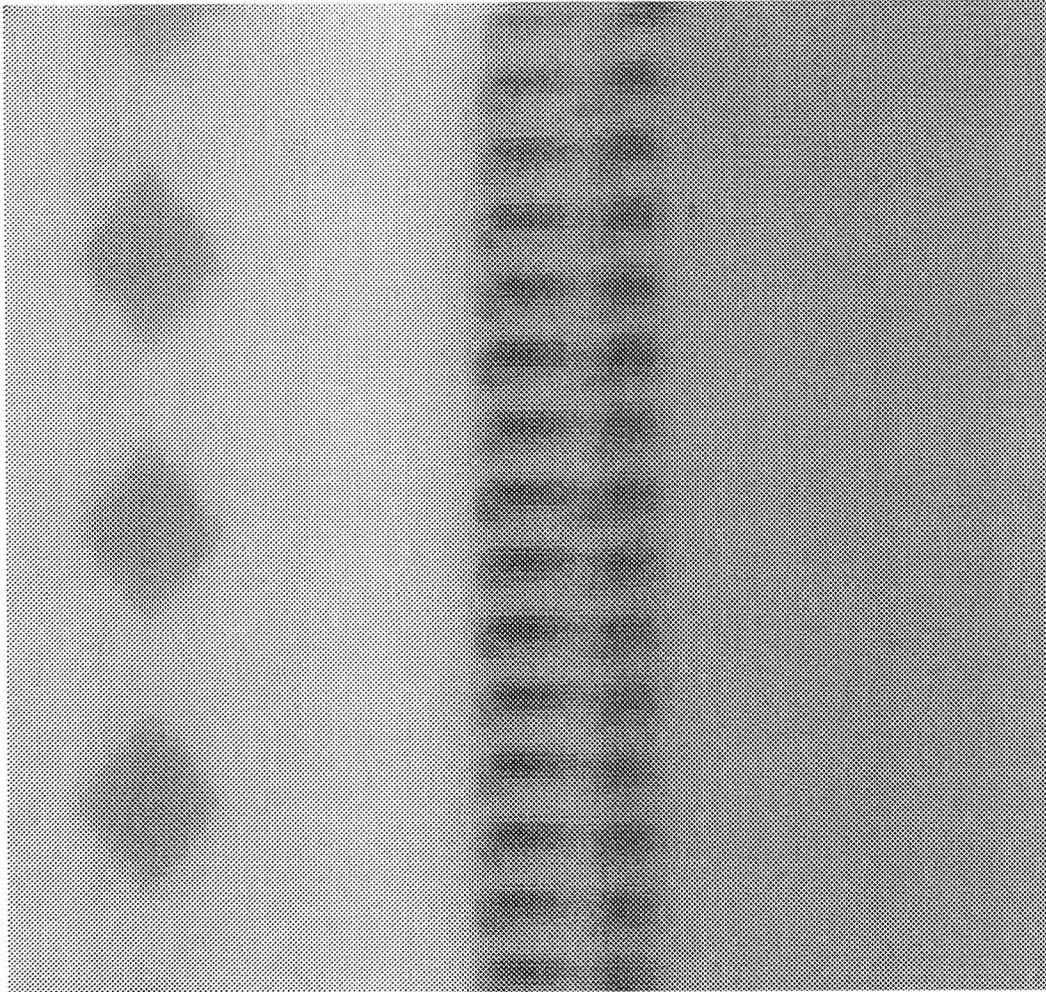


i132.4\_34-51: All the joints are with normal paste volume.

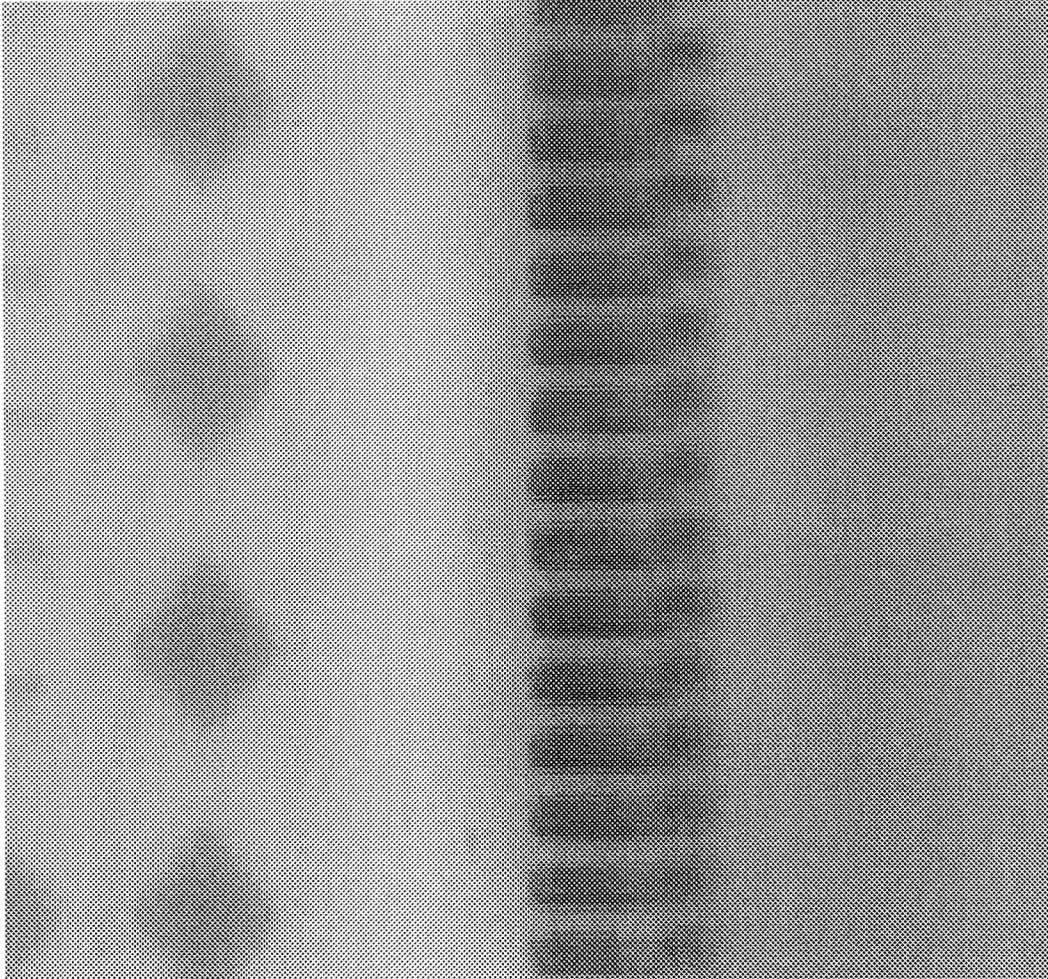


i132.4\_52-66: All the joints are with normal paste volume.

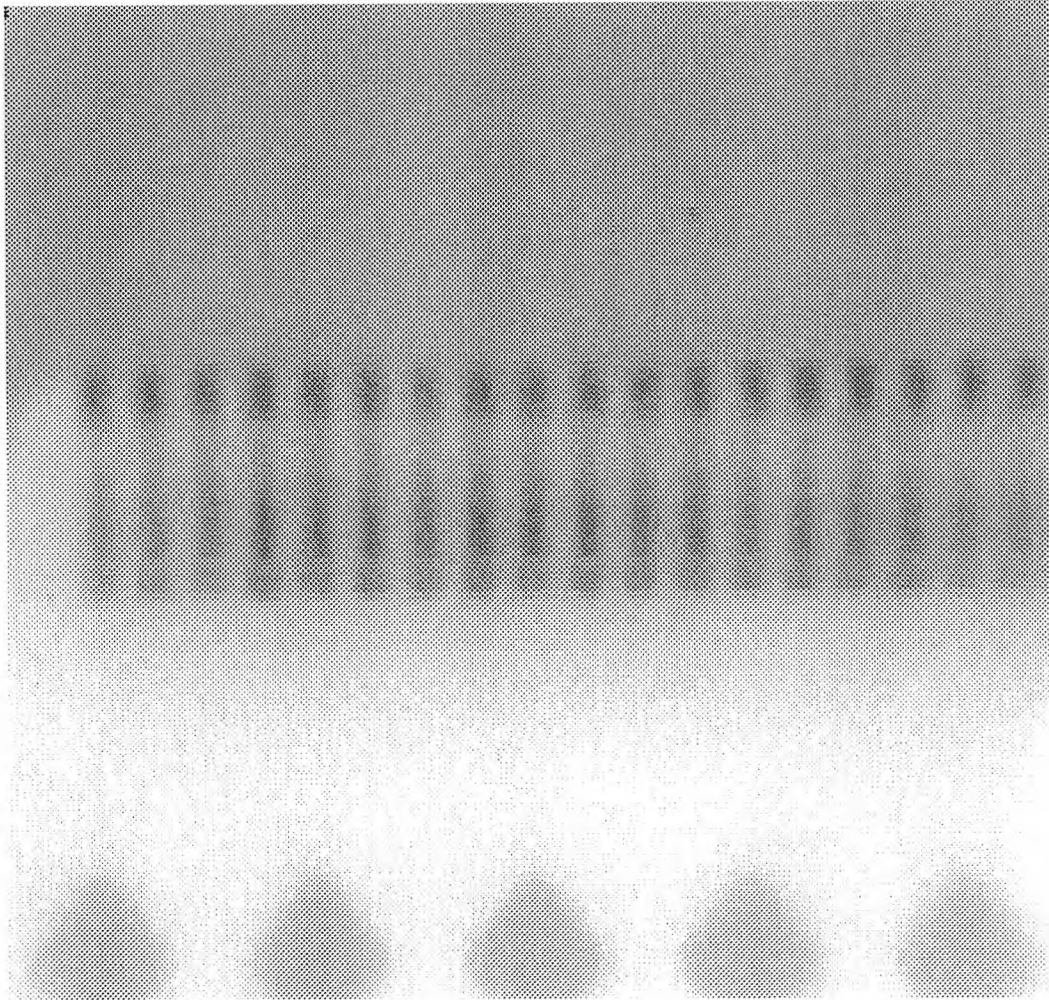




i132.4\_67-73: All the joints are with normal paste volume.

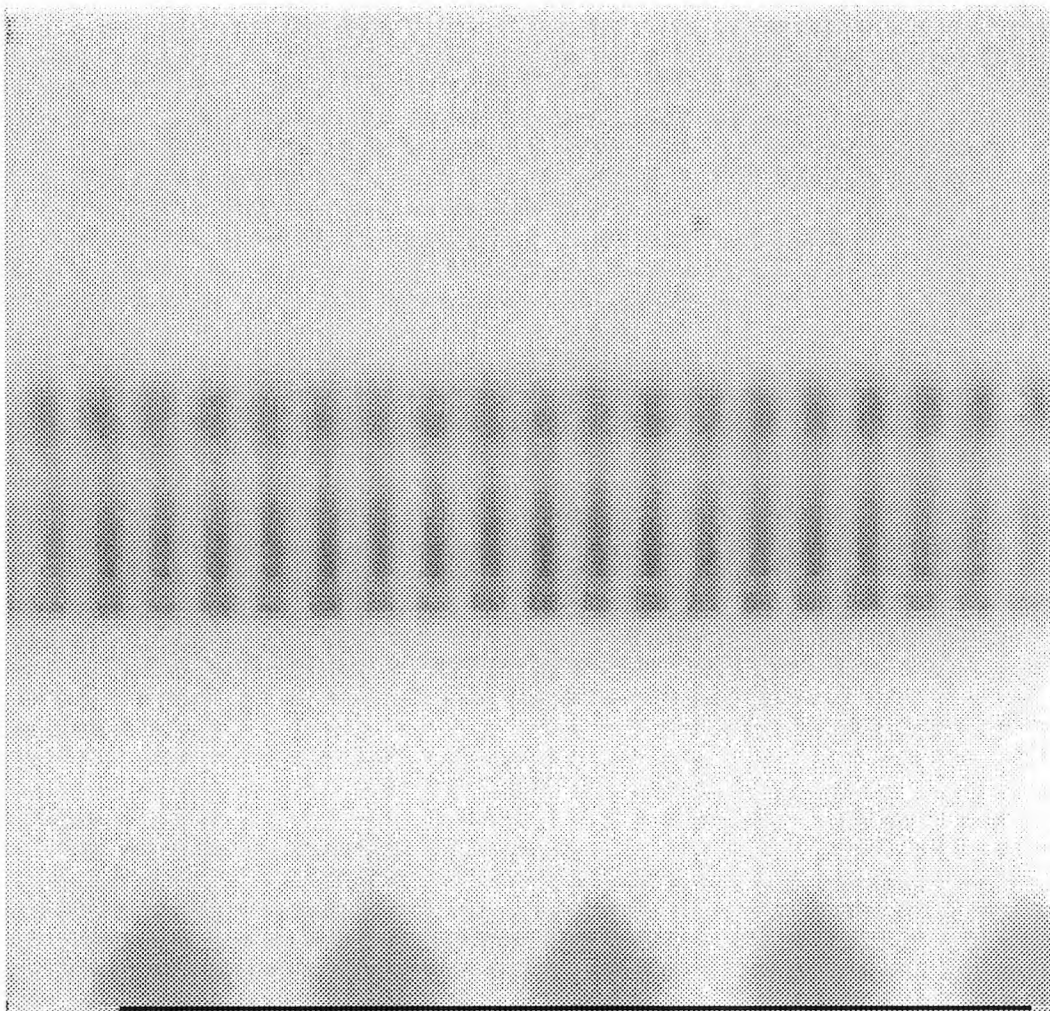


i132.4\_74-99: All the joints are scraped to no paste.



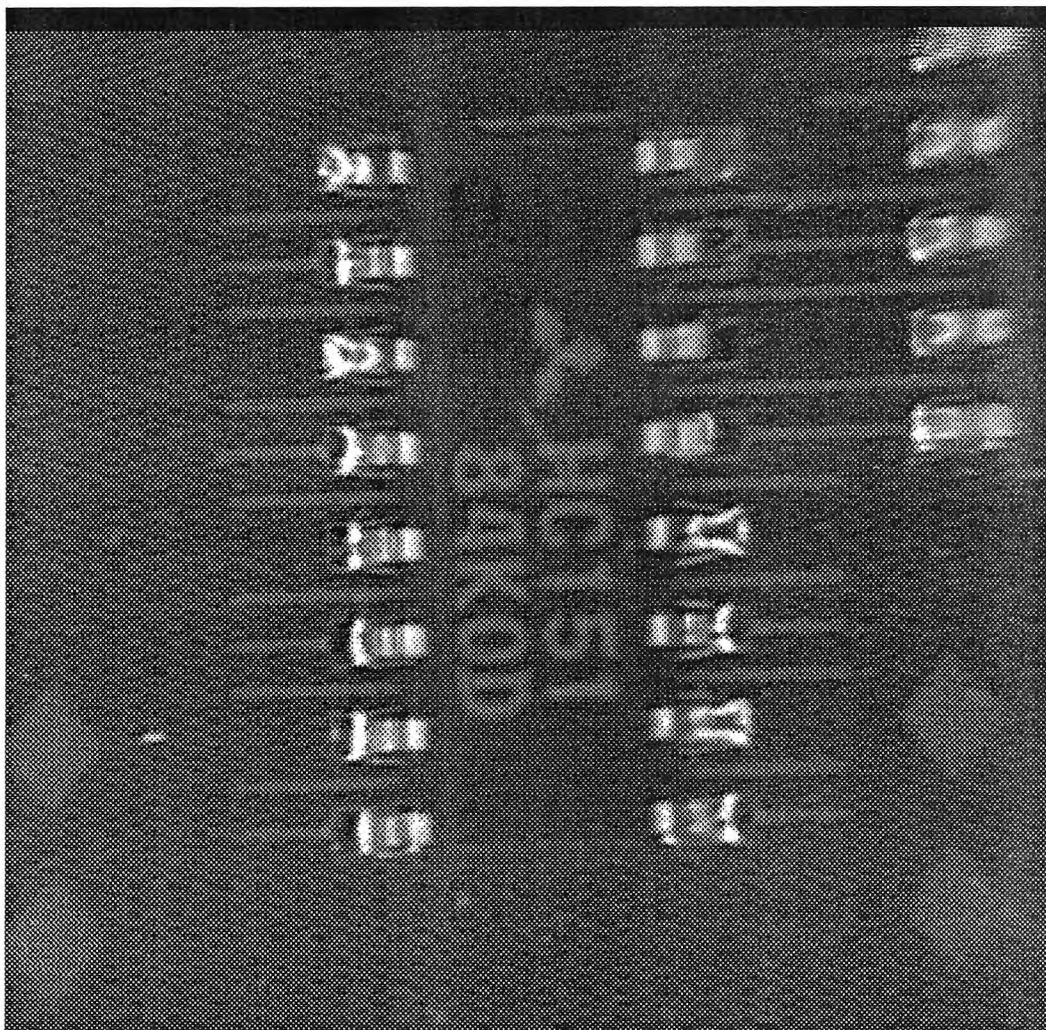
i132.4\_100-116: All the joints are with normal paste volume.





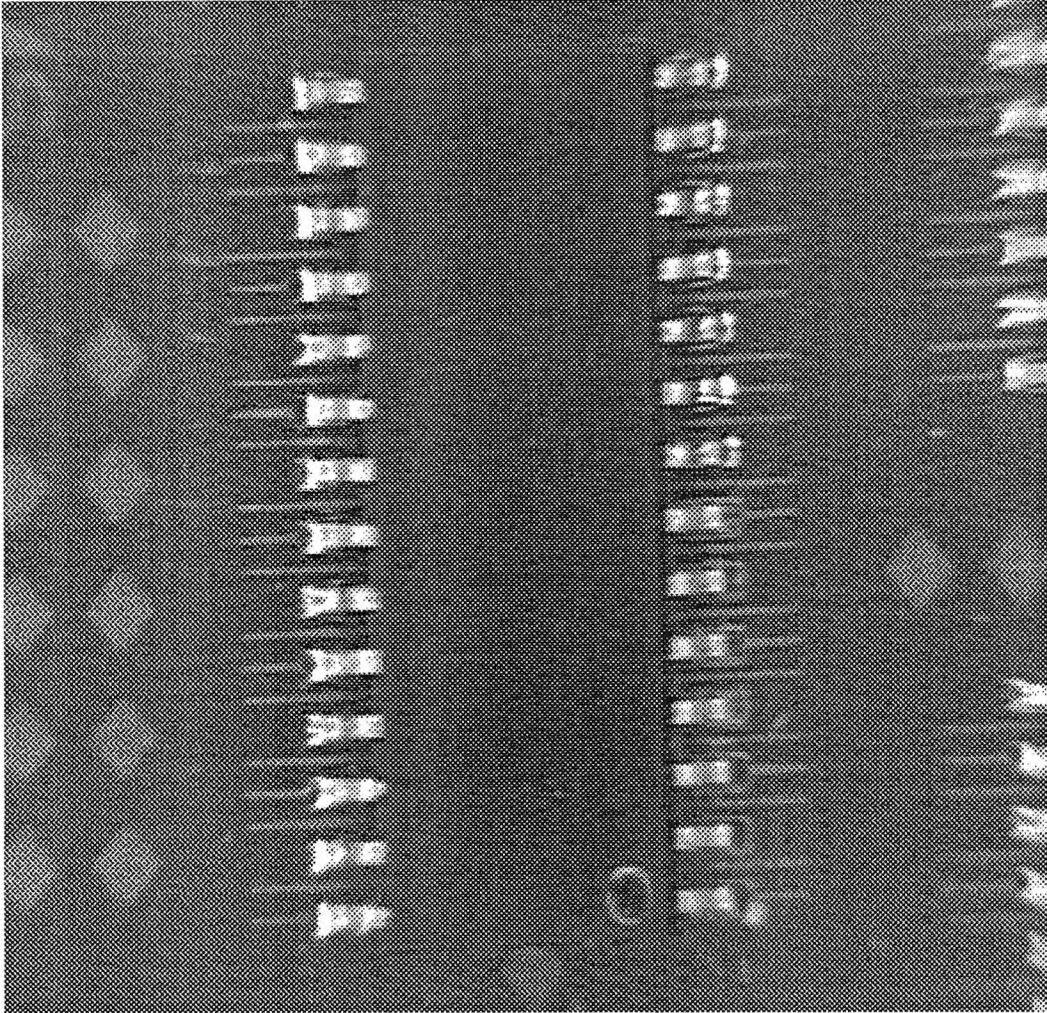
i132.4\_117-132: All the joints are with normal paste volume.

## C Visual Light Images

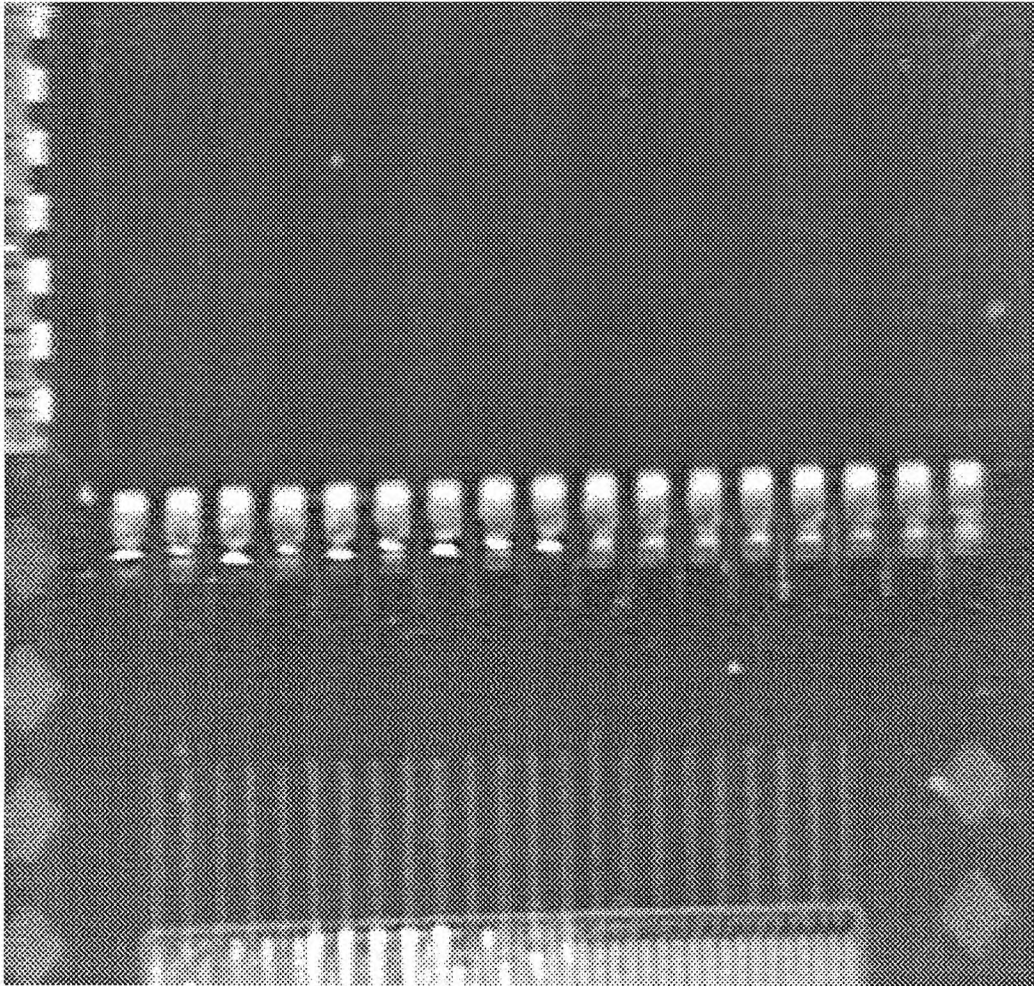


v16.1: The upper right four joints are scraped to no paste, the lower left four joints are scraped to half paste volume, and the rest are with normal paste volume.

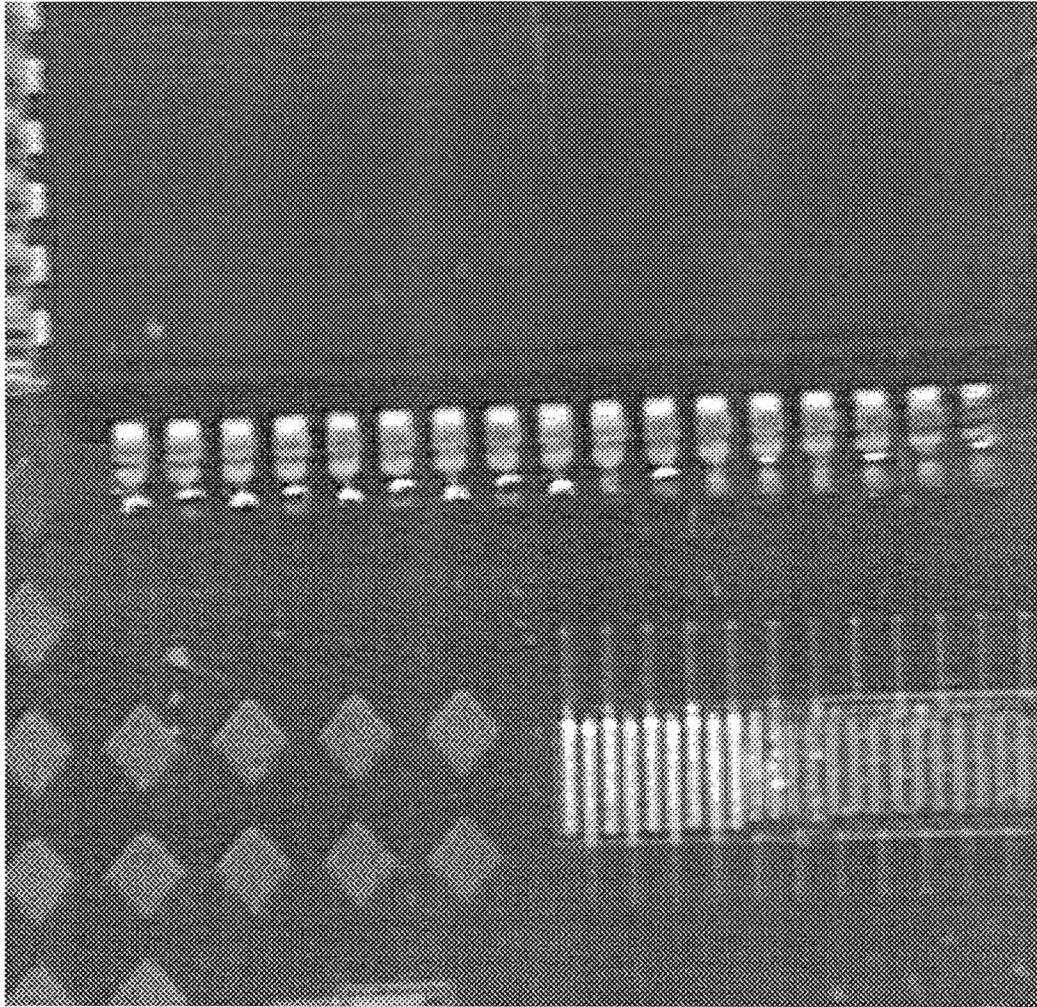




v28.2: The lower right seven joints are scraped to no paste, the upper left seven joints are scraped to half paste volume, and the rest are with normal paste volume.

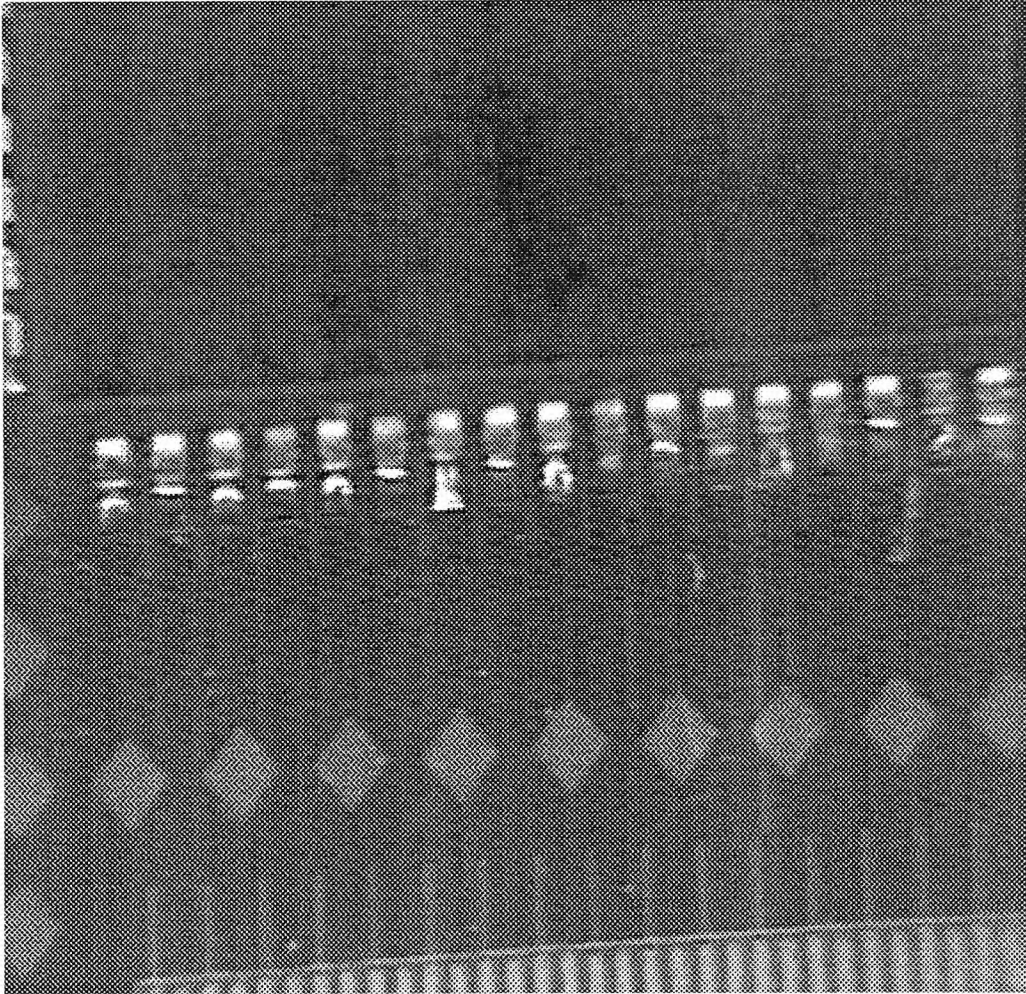


v68.1\_1-17: The left nine joints are with normal paste volume, while the right eight joints are scraped to no paste.

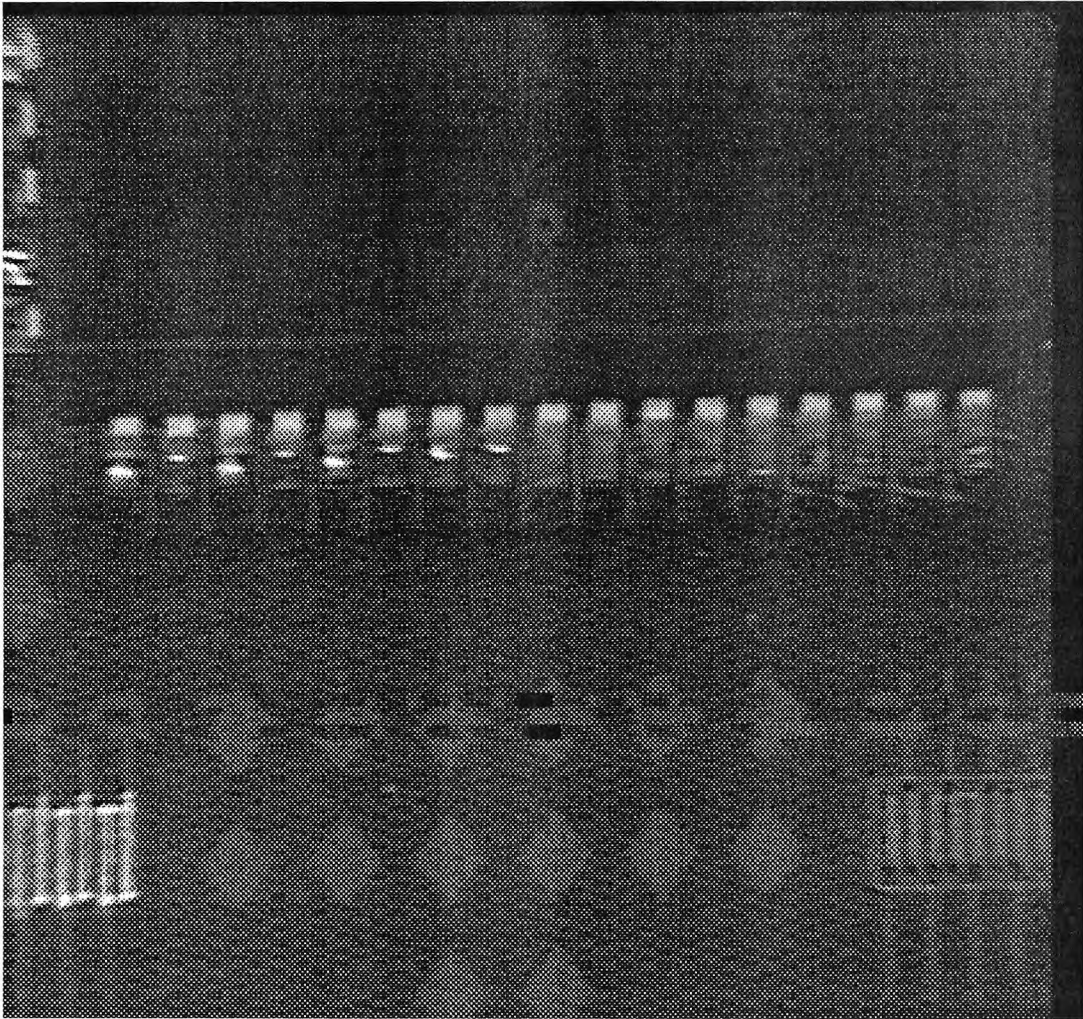


v68.1\_18-34: The left nine joints are with normal paste volume, while the right eight joints are scraped to no paste.

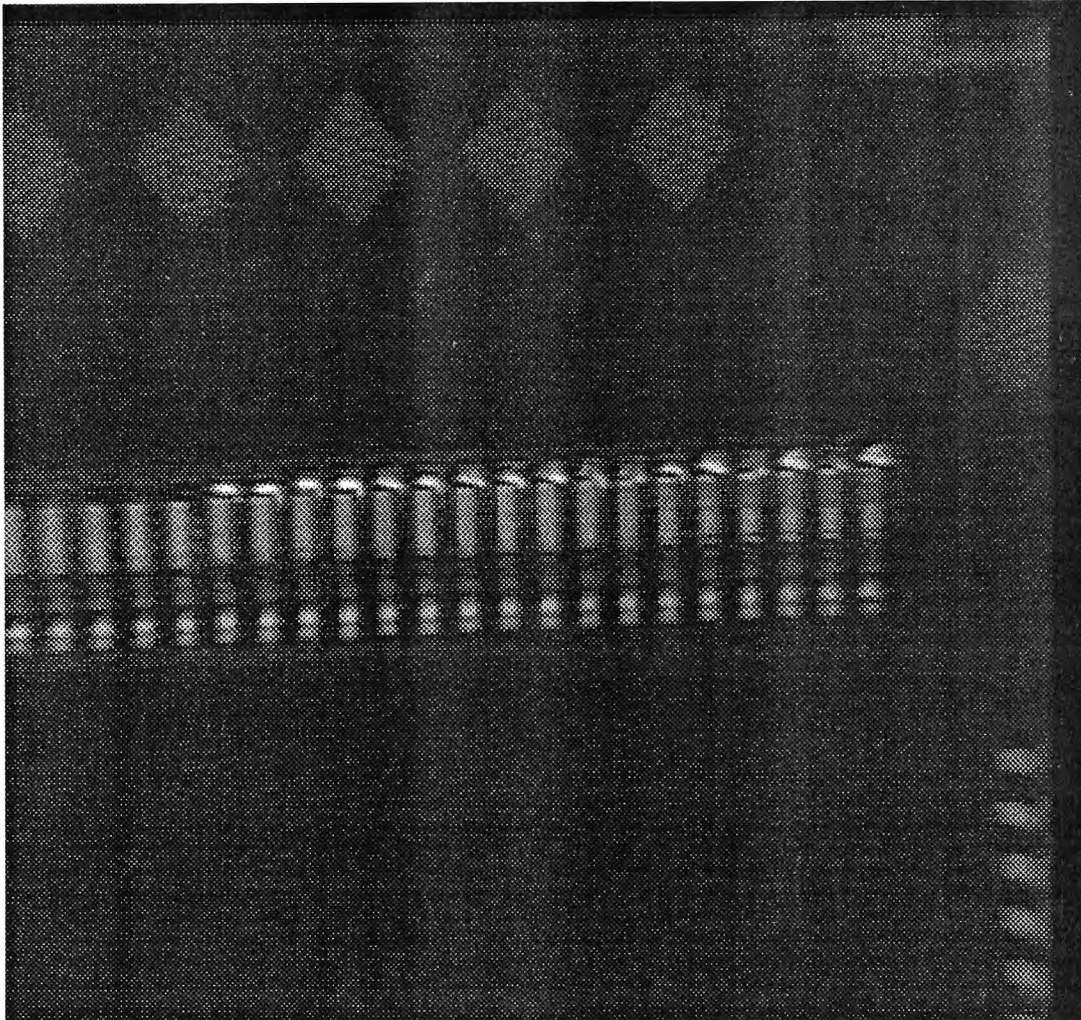




v68.1\_35-51: The left nine joints are with normal paste volume, while the right eight joints are scraped to half paste volume.

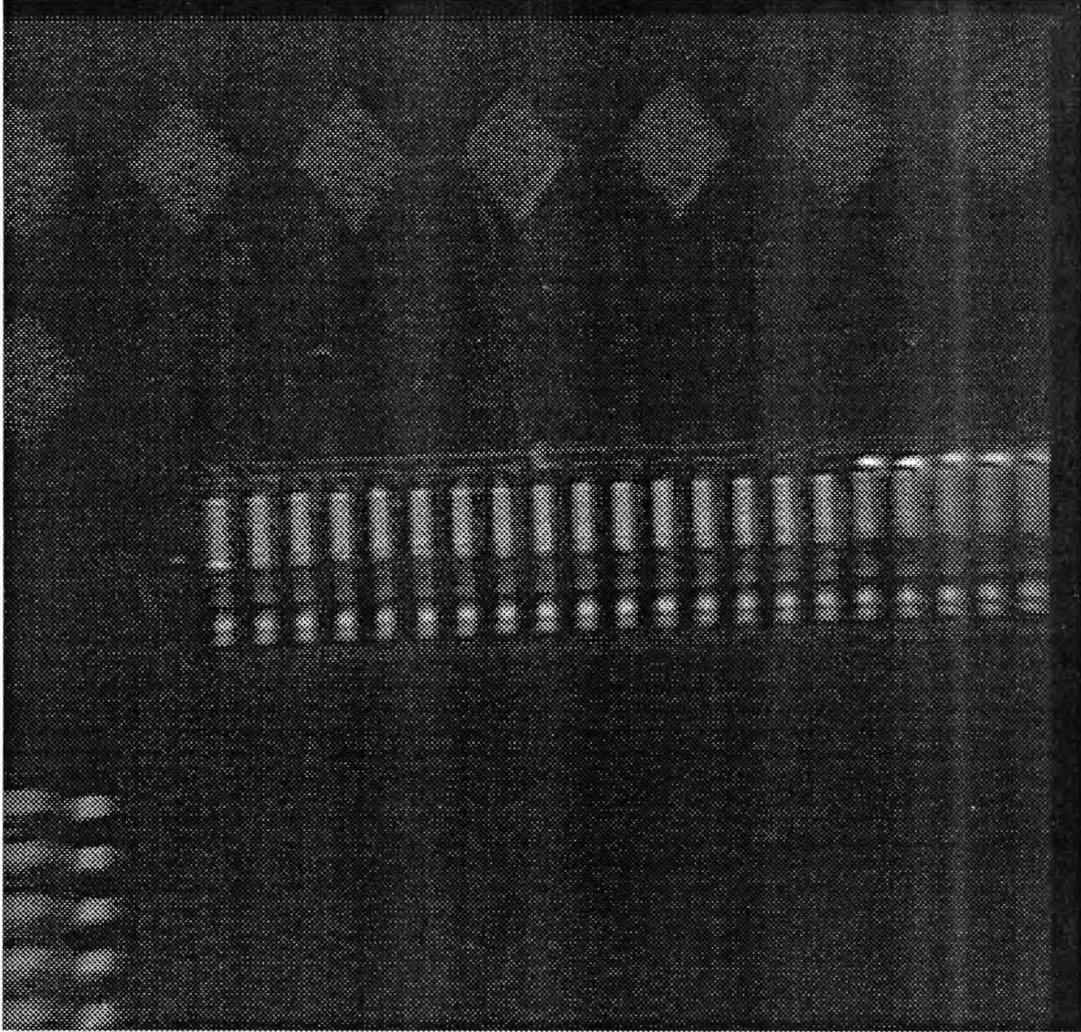


v68.1\_52-68: The left nine joints are with normal paste volume, while the right eight joints are scraped to half paste volume.

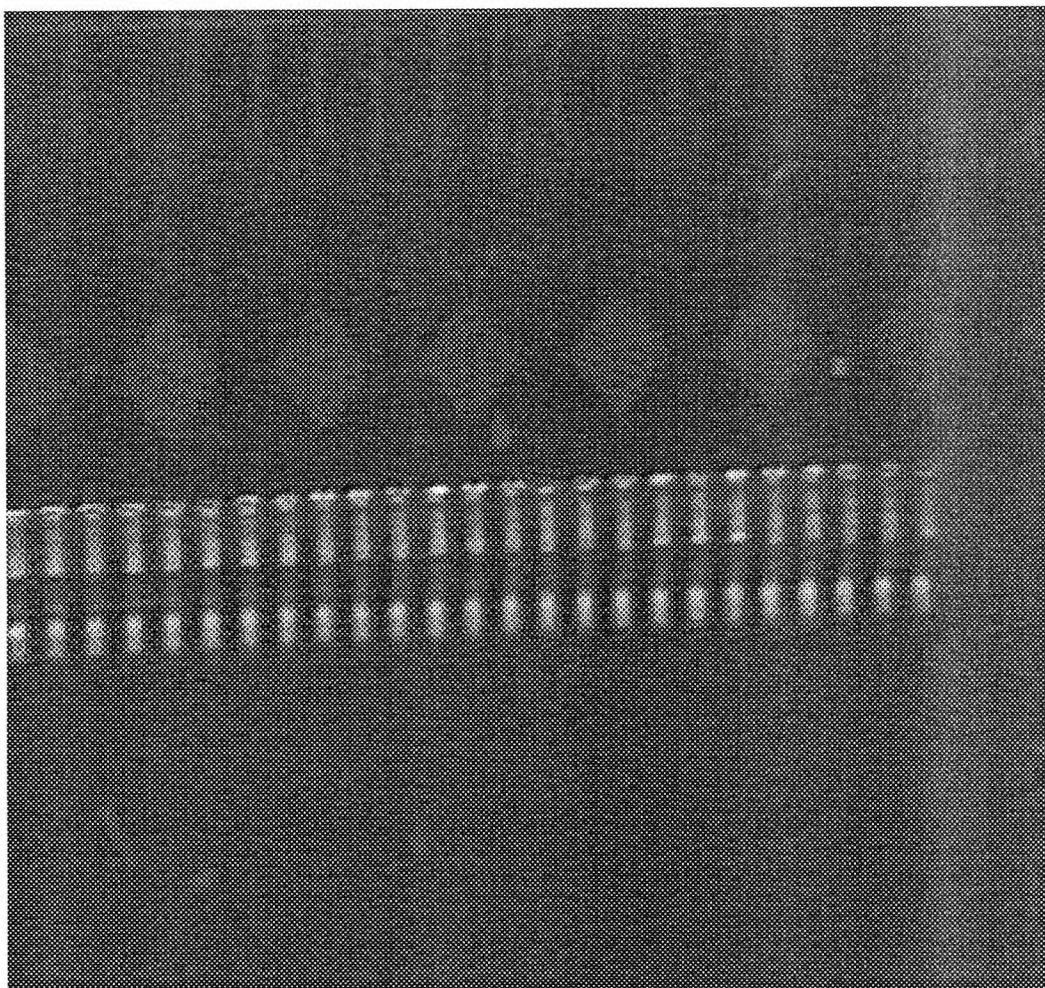


v132.4\_1-17: The right 17 joints are with normal paste volume, while the rest are scraped to no paste.



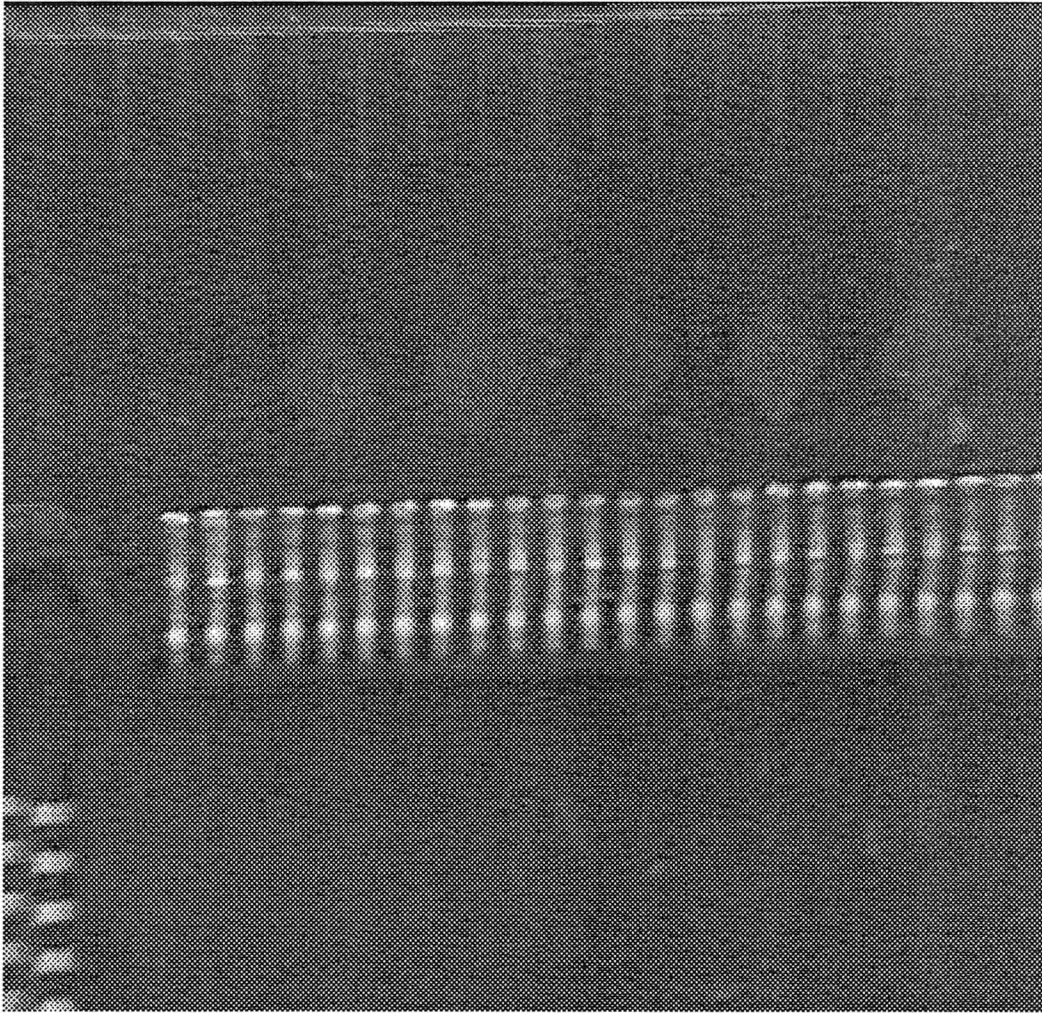


v132.4\_17-33: The left 16 joints are scraped to no paste, while the rest are with normal joints.

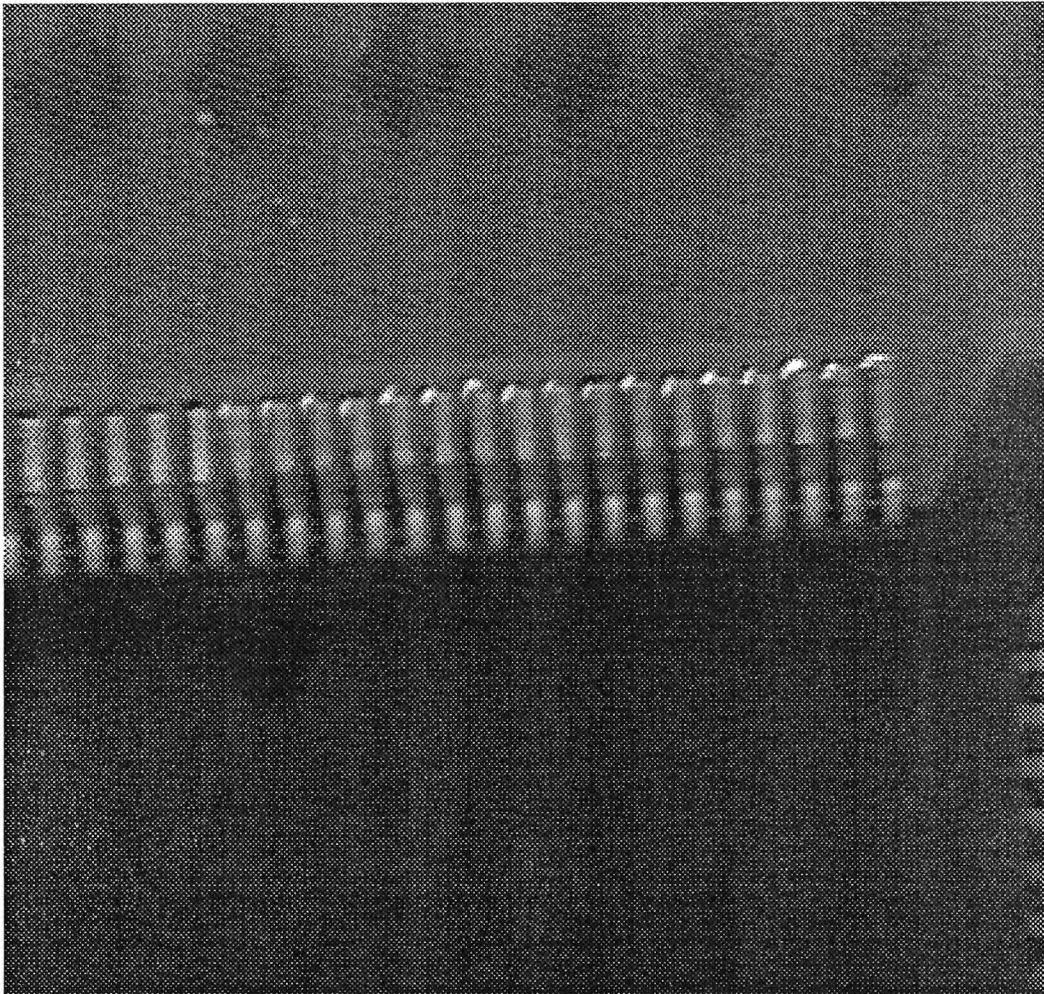


v132.4\_34-52: All the joints are with normal paste volume.

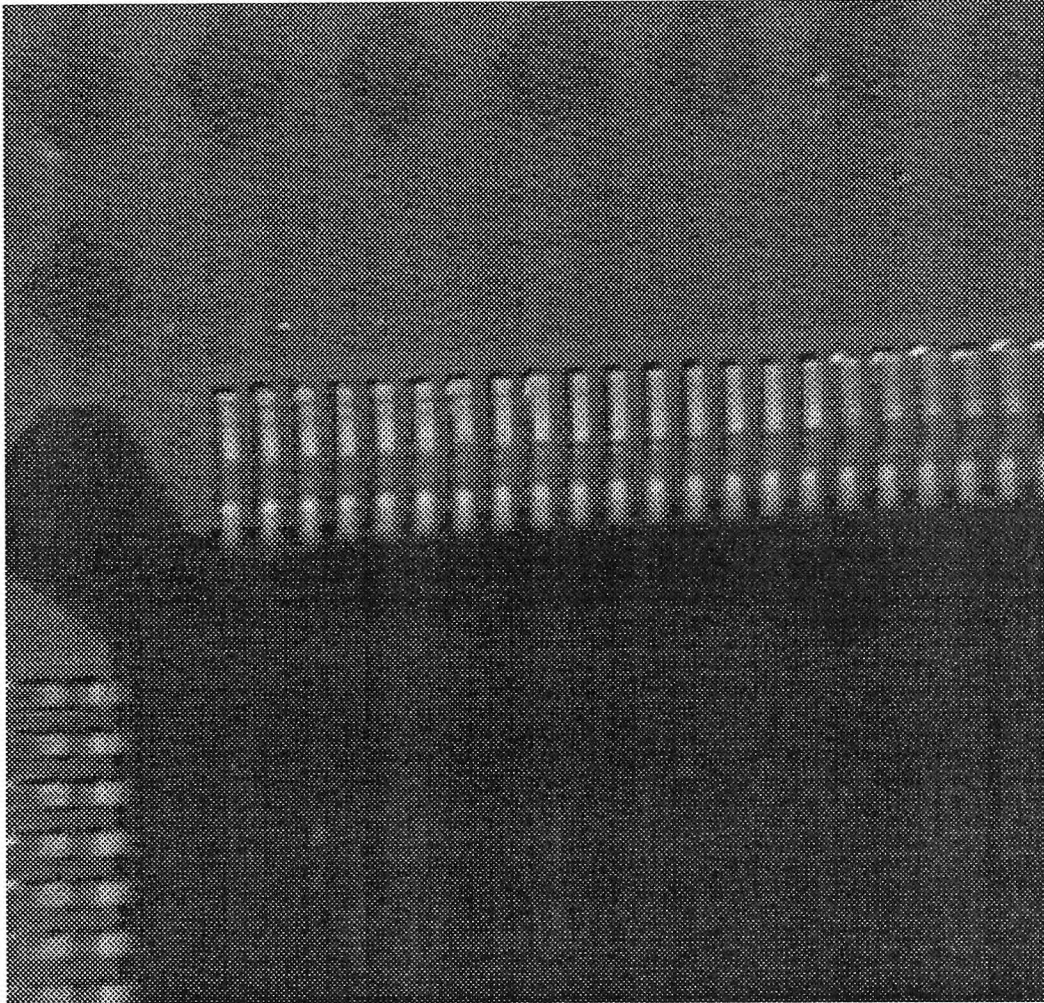




v132.4\_50-66: All the joints are with normal paste volume.

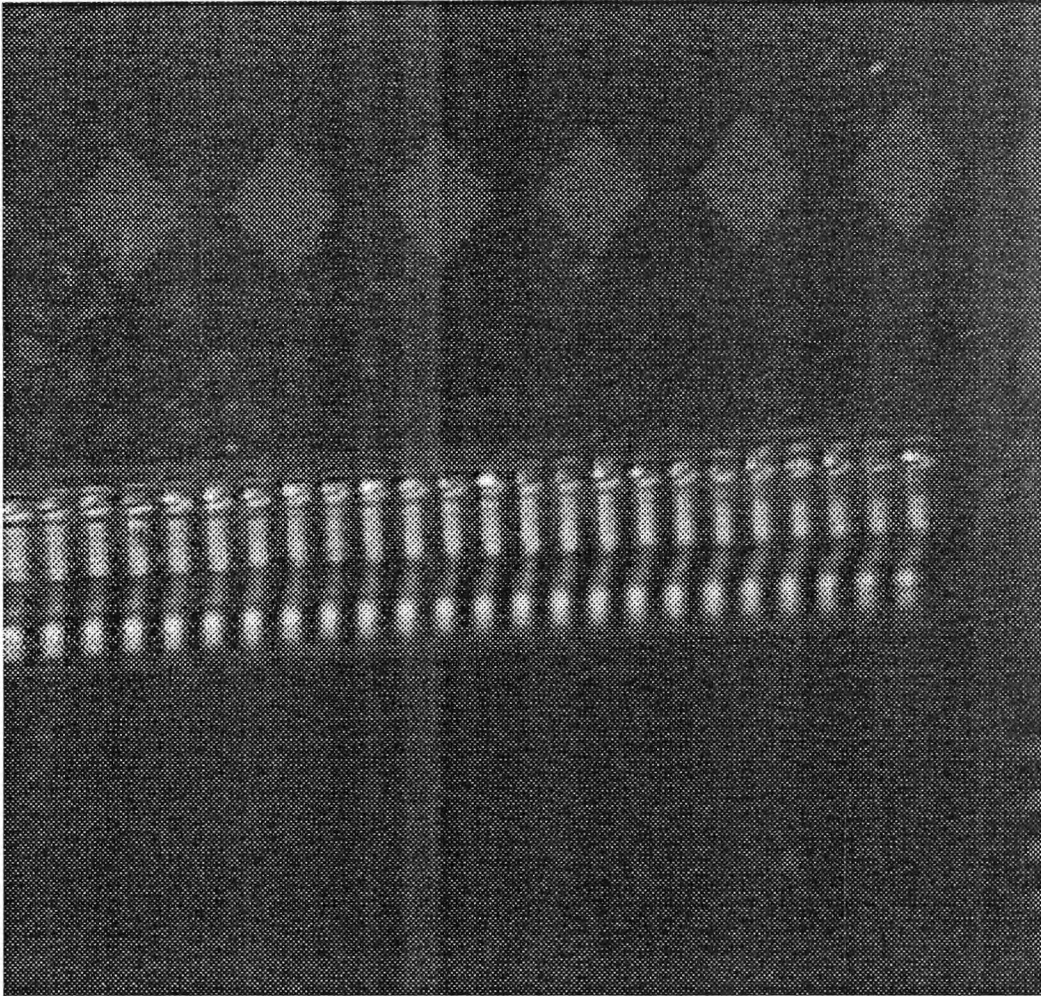


v132.4\_67-83: The right 17 joints are with normal paste volume, while the rest are scraped to no paste.

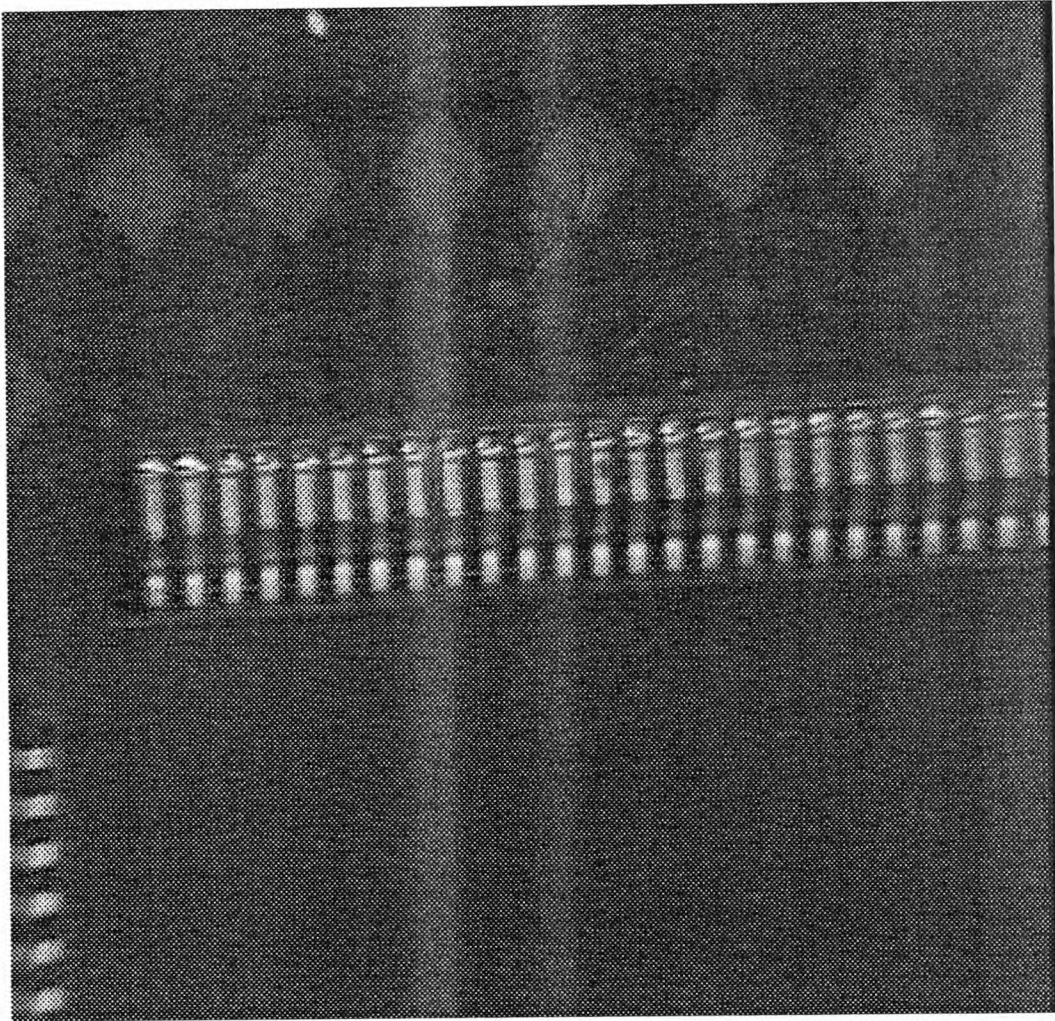


v132.4\_84-99: The left 16 joints are scraped to no paste, while the rest are with normal joints.



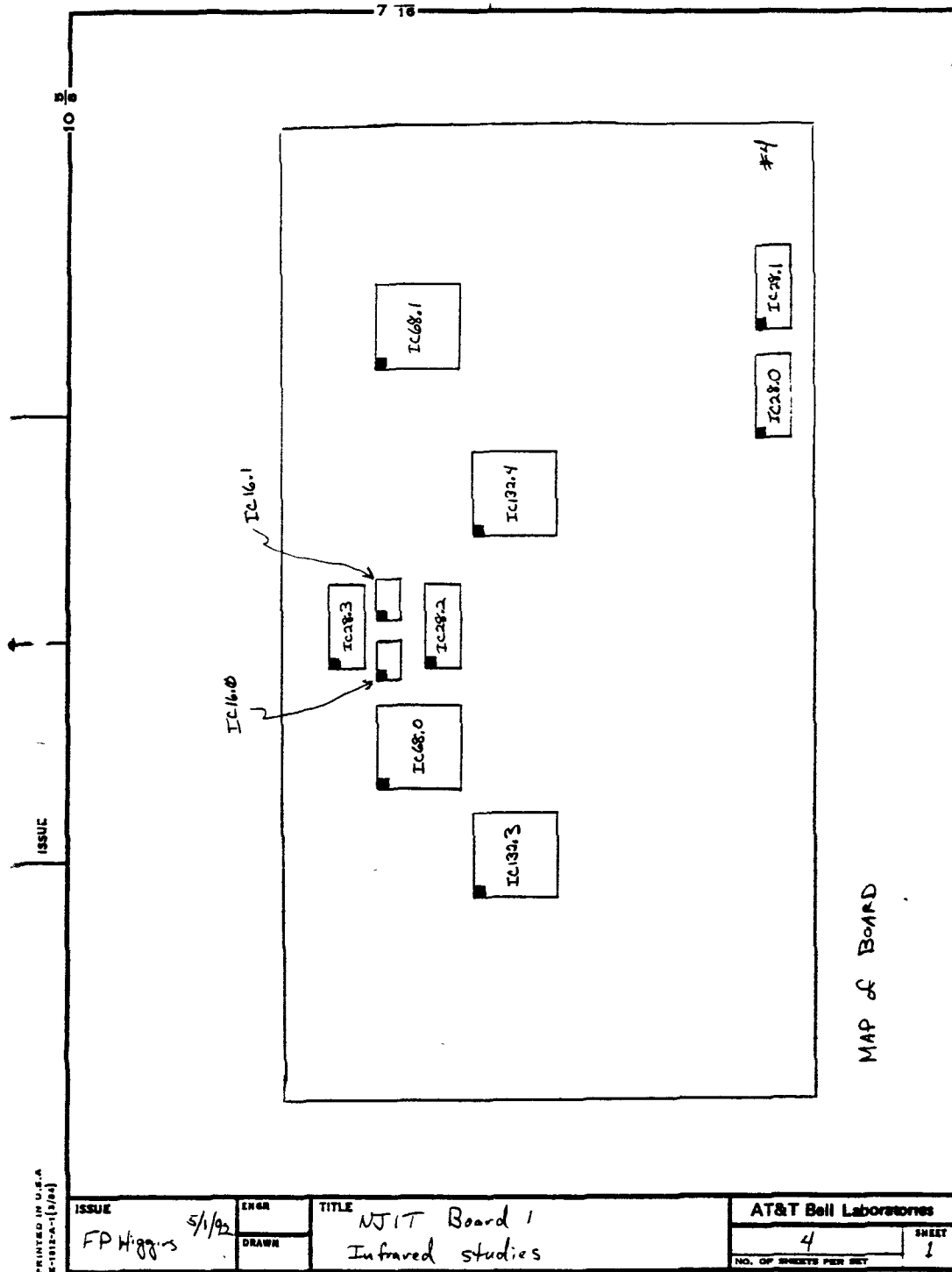


v132.4\_100-114: All the joints are with normal paste volume.



v132.4\_114-132: All the joints are with normal paste volume.

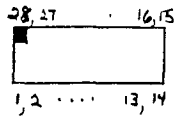
### D The Diagram Of The Experiment Circuit Board



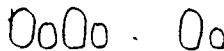
IC 28.x

28 pin SOIC, 50 mil. pitch, gull wing lead

site description

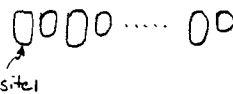


Normal paste deposition



Half paste ○○○○

Alternating stencil

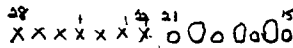


IC 28.0, IC 28.3 Untouched, normal paste deposition (above)

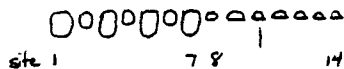
IC 28.1 sites 1-7 } Untouched, normal paste deposition (above)  
15-21 }

8-14 - Scraped to half paste (above)

22-28 - Scraped to no paste



Example



Measured open resistance on sites 10, 22, 23, 25

IC 28.2 sites 1-7 - Scraped to half paste  
8-14 - Normal, untouched  
15-21 - Scraped to no paste  
22-28 - Normal, untouched

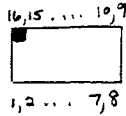
PRINTED IN U.S.A.  
8-1013-A-(1/76)

ISSUE F.P. Higgins 5/1/92	ENGR DRAWN	TITLE NJIT Board 1 Infrared studies	AT&T Bell Laboratories	SHEET 2
			4 NO OF SHEETS PER SET	

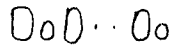
7 16

IC16.x 16 pin SOIC, 50 mil pitch, gull wing lead

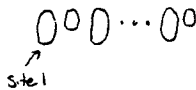
Site description



Normal paste deposition



Alternating stencil

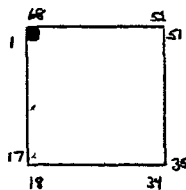


IC16.0 - Untouched, normal paste deposition

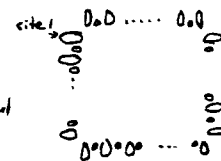
IC16.1 - Sites 1-4, untouched, normal paste deposition  
 sites 5-8, Scraped to half paste  
 sites 9-12, untouched, normal paste deposition  
 sites 13-16, Scraped to no paste

IC68.x 68 pin quad flat pack, 50 mil pitch, J lead

Site description



Normal paste deposition



Alternating stencil

IC68.0 - Untouched, normal paste deposition

IC68.1 - Sites 1-9, untouched, normal paste deposition  
 sites 10-17, Scraped to no paste  
 sites 18-26, untouched, normal paste deposition  
 sites 27-34, Scraped to half paste  
 sites 35-43, untouched, normal paste deposition  
 sites 44-51, Scraped to half paste  
 sites 52-59, untouched, normal paste deposition  
 sites 60-68, Scraped to no paste

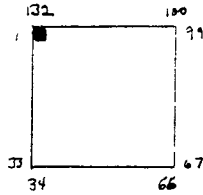
PRINTED IN U.S.A.  
 6-11112-01(1/74)

ISSUE	5/1/72	ENG'R	TITLE	AT&T Bell Laboratories	
	F.P. Higgins	DRAWN	WJIT Board 1 Infrared Studies	4	SHEET 3
				NO. OF SHEETS PER SET	



IC132.x 132 pin quad flat pack, 25 mil pitch, 3/4" wing

site description



Normal paste deposition

Constant stencil

IC132.3 Untouched, normal paste deposition

IC132.4 sites 1-17, untouched, normal paste deposition  
 sites 18-33 Scraped to no paste  
 sites 34-83 untouched, normal paste deposition  
 sites 84-99 Scraped to no paste  
 sites 100-132 untouched, normal paste deposition

PRINTED IN U.S.A.  
 E-1812-A-1 (5/64)

ISSUE F.P. Higgins 5/1/92	ENGR	TITLE NJIT Board 1 Infrared Studies	AT&T Bell Laboratories	
	DRAWN		NO. OF SHEETS PER SET 4	SHEET 4

## REFERENCES

1. Besl, P. J., E.J. Delp and R. Jain, "Automatic Visual Solder Joint Inspection." *IEEE Journal of Robotics and Automation*. Vol. RA-1, (Mar. 1985):42-56.
2. Gonzalez, Rafael C. and Paul Wintz, "Digital Image Processing." (Second Edition). *Addison-Wesley Publishing Company*,(Nov. 1987).
3. Tou, Julius T. and Rafael C. Genzalez, "Pattern Recognition Principles." *Addision-Wesley Publishing Company*, (1974).
4. "Datacube User's Manual." *Datacube Inc.*,( Oct, 1988).
5. Zhou, Ken Kekuan, "Automatic Infrared Inspection of Solder Joints on Printed Circuit Boards." M.S. thesis directed by Dr. Yun Q. Shi, Electrical and Computer Engineering Department, New Jersey Institute of Technology, (May, 1990).
6. Burnay, S. G. , T. L. Williams and C. H. Johnes, "Applications of Thermal Imaging." *IOP Publishing Ltd.*, (1988).
7. "Sunview Programmer's Guide." *SUN Microsystem Inc.*, (1991).
8. Blume, Steven J. and Paul D. Blume, "Infrared TV Camera User's Guide." (Dec. 1989).
9. Bartkett, Sandra L. , Charles L. Cole and Ramesh Jain, "Expert System For Visual Solder Joint Inspection." *Conf. 3rd IEEE Artificial Intelligence Application*, Orlando, Florida, (Feb. 1987).
10. Chen, Roland T. , "Survey Automate Inspection:1981 to 1987." *Computer vision, Graphics, and Image processing*, 41, (1988):346-381.
11. Bartlett, Sandra L., Paul J. Besl, Charles L. Cole, Ramesh Jain, Debashish Mukherjee and Kurt D. Skifstad, "Automatic Solder Joint Inspection." *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 1, Japan. (1988):31-34.
12. Nakagawa, Yasuo, "Automatic Visual Inspection Of Solder Joints On Printed Circuit Boards." *Proc. SPIE, Robot Vision*, Vol. 336, (May 1982):121-127.

13. Huang, Jiang , "Automatic Infrared and Visual Inspection of Solder Joints on PCBs." M. S. thesis directed by Dr. Y. Q. Shi., ECE Department, NJIT, (May 1991).