Title

AN AUTOMATED VISION SYSTEM USING A FAST 2-DIMENSIONAL MOMENT INVARIANTS ALGORITHM

Marwan F. Zakaria

July, 1987

McGill Research Centre for Intelligent Machines
Robotic Mechanical System Laboratory
Department of Mechanical Engineering
McGill University, Montreal

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Engineering.

Abstract

Moment invariants can be used to describe features of an object, so as to reduce ambiguity and difficulty of recognition with a computer vision system. With contiguous images, certain higher order pixel, or picture element, moments have been shown to be invariant with respect to translation, rotation, and scaling of the image. However, due to the iterative nature of the required calculations and to computational speed limitations, these moments cannot be computed in real-time, *e.g.*, fast enough to serve the purpose of many industrial processes. To overcome this limitation a novel algorithm, the *Delta Method*, has been devised and applied to a typical process. This simple, fast algorithm has been implemented in a video/personal computer subsystem and verified with experimental results using textile garment components, intended for automated assembly. In this, and for many other applications the *Delta Method* promises to greatly reduce the time complexity of the processing required to identify an object.

Résumé

Les invariants de moment peuvent être utilisés pour décrire les caractéristiques d'un objet, de façon à réduire les ambiguités et les difficultés de reconnaissance d'un système de vision par ordinateur. Avec des images contigues, certains moments de pixel, ou élément d'une image, d'un ordre supérieur sont invariants par rapport à la translation, la rotation et les dimensions de l'image. Cependant, en raison des restrictions sur la vitesse des calculs, ces moments ne peuvent être calculés en temps réel, comme par exemple, d'une façon assez rapide pour satisfaire les besoins de plusieurs procédés industriels. Afin de surmonter cette restriction, un nouvel algorithme, la Méthode Delta, a été conçu et appliqué à un procédé typique. Cet algorithme, simple et rapide, a été introduit dans un sous-système vidéo/ordinateur personnel et vérifié expérimentalement en utilisant des pièces de tissus de vêtements devant être assemblées par automatisation. Dans cette application, comme dans plusieurs autres, la Méthode Delta promet de réduire considérablement la complexité du problème associé à la durée des calculs requis pour identifier un objet.

Acknowledgements

I am very indebted to both Prof. Louis J. Vroomen and Prof. Paul J. Zsombor-Murray, my supervisors. They were not only the inspiration behind this work, but were of tremendous assistance in guiding and helping me throughout this thesis research. Their encouragement and constructive criticisms helped me enjoy this work and accomplish it with pride. I shall be eternally grateful to them for their help.

Hania A. Zakaria, my wife, for her tremendous love and moral support throughout my scholastic years at McGill.

All the members of our Robotic Mechanical System Laboratory, and especially Louis C. Vroomen for whom it seems there is no unsolvabale software problem.

Table of Contents

Table of Figures

Table of Symbols

a	: vertical size of an object in mm.
b	: horizontal size of an object in mm.
D_h	: the percentage of the smallest horizontal distance to be detected.
D_v	: the percentage of the smallest vertical distance to be detected.
$f(x,y)$	: density (or intensity) distribution function.
$f(i,j)$	: discrete intensity distribution function.
$g(r,\theta)$	: intensity distribution function in polar coordinates.
J	: The Jacobian of a transformation.
m_{pq}	: 2-D moments of order (p+q).
$m_{pq,i}$	: the contribution of row i to the nominals m_{pq}.
m_{pqr}	: 3-D moments of order (p+q+r).
M	: angular and radial moment invariant.
M^*	: angular and radial moment invariant, invariant with respect to size.
$M(u,v)$	: 2-D moment generation function of $f(x,y)$.
R_h	: the horizontal resolution of the object.
R_v	: the vertical resolution of the object.
R	: the overall resolution (accuracy).
$S_R(x,y)$	: rectangular sampling function.
$S_H(x,y)$	: hexagonal sampling function.
X_i	: the X-coordinate of the first pixel in row i.
Y_i	: the Y-coordinate of the first pixel in row i.

Table of Symbols

α	:	weight of the algebraic invariant.
β	:	constant of the translation matrix.
Γ	:	constant of the translation matrix.
δ	:	the number of chained pixels in row i.
Δ	:	determinant of a linear transformation matrix.
Δx	:	sampling intervals.
Δy	:	sampling intervals.
η_{pq}	:	normalized central moments of order (p+q).
$\Phi_r(k,g)$	:	radial and angular moment.
μ_{pq}	:	central moments of order (p+q).
σ	:	constant of the translation matrix.
τ	:	constant of the translation matrix.
ϕ	:	moment invariant.

Chapter 1 INTRODUCTION AND THEORY

1.1 Introduction

In the past two decades, there has been a considerable amount of work devoted to develop a computer-controlled robot with hands, eyes, and ears, or simply an intelligent robot. The demand for such a general purpose manipulator has originated primarily from the need to automate industrial processes, to explore and exploit environments hazardous to life, to handle radioactive and other dangerous materials, and to aid the handicapped and their therapists.

An intelligent robot might consist of a manipulator that is integrated with a vision system which represents the eyes (the camera) and the brains or the intelligence (the microprocessor and its application programs). Our concern in this research was to enhance the intelligence of a manipulator by developing a task-oriented control algorithm, using some of the known algorithms for edge detection and image processing, so the manipulator could "see" and then could compare images of objects intelligently. After considerable investigation, the method of Moment Invariants was chosen. This method has the ability to identify an object (image) independent of its position, size, and orientation using seven invariant parameters that describe a particular image. This means that for the purpose of comparison, only seven numerical descriptors, rather than the entire digitized image, need be stored and compared .

This thesis deals with the design and testing of an automated vision system, using a fast algorithm for moment invariant generation. The algorithm was proven in a system designed for the identification and verification of textile components. The design of such a system involves two steps: Creating the appropriate algorithm to identify the object, then verifying that it is free of any defects. This algorithm would ultimately be implemented in a combined robotic-vision system that will analyze the data provided so as to recognize and reject defective objects while computing the registration correction for objects which are slightly misaligned. The system, using the *delta method*, is able to analyze the two-dimensional features of a textile component and to generate it's moment invariants with a considerable reduction of time over conventional methods. It will be shown that the *delta moment method* of calculating the moments serves both in the identification and the verification process (see Chapter 3 for more details).

1.2 Theoretical Consideration of Moment Invariants

Moment invariants have been used as features in object recognition, image classification and scene matching [1-21]. These invariant features extracted from two-dimensional images are invariant under image translation, scaling and rotation. The use of moment invariants was first proposed by Hu [1-2] in 1962, for two-dimensional character recognition. The application of moment invariants to more complex two-dimensional scenes was extended by Sadjadi and Hall [11,14].

The concept of moment invariants is based on invariant algebra which deals with the properties of certain classes of algebraic expressions which remain invariant under general linear transformations.

1.2.1 A Uniqueness Theorem Concerning Moments

Any geometric pattern can always be represented by a density (or intensity) distribution function f(x,y), with respect to a pair of orthogonal axes fixed to the visual field.

The two-dimensional moments of order (p+q) of an image, computed from the continuous image intensity function f(x,y), are defined in terms of Riemann integrals as :

$$m_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p . y^q . f(x,y) . dx . dy \qquad (1)$$

where $p,q \in \{ 0,1,2,... \}$

If it is assumed that f(x,y) is a piecewise continuous therefore bounded function, and that it can have nonzero values only in the finite part of the *xy* plane; then moments of all orders exist and the following uniqueness theorem can be proven.

Uniqueness Theorem : The double moment sequence $\{m_{pq}\}$ is uniquely determined by f(x,y); and conversely, f(x,y) is uniquely determined by $\{m_{pq}\}$. Hence one may use $\{m_{pq}\}$ as a means of representing any two-dimensional pattern.

It should be noted that the finiteness assumption is important; otherwise, the above uniqueness theorem will not hold.

The computations of m_{pq} consist of multiplying the function $f(x,y)$ by a nominal $x^p y^q$ and integrating the result. The nominals of order 3 or less are x^0y^0, x^0y^1, x^0y^2, x^0y^3, x^1y^0, x^1y^1, x^1y^2, x^2y^0, x^2y^1, and x^3y^0. These are sufficient to describe any two-dimensional object. Any higher order moment can be disregarded.

The moments of order p+q may also be interpreted as the response of an imaging system with the transfer function of, $x^p y^q$, and the input, $f(x,y)$. Low order moments have intuitive relations to objects. For example, m_{00} is related to mass, m_{10} and m_{01} to centre of mass and m_{11}, m_{20}, and m_{02} to the principal axes.

1.2.2 The Moment Generating Function

The moment generating function of $f(x,y)$ may be defined as

$$M(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \exp[ux+vy].f(x,y).dx.dy$$

If *u* and *v* are considered as complex variables, this expression is a two sided Laplace transform. For the invariant development both *u* and *v* are assumed to be real. This function can also be written in the form of :

$$M(u,v) = \sum_{p=0}^{\infty} \sum_{q=0}^{\infty} m_{pq} \cdot \frac{(u)^p}{p!} \cdot \frac{(v)^q}{q!}$$

where the exponential has been expanded by its Taylor series equivalent, assuming that moments of all orders exist. This equation shows that the moments may be determined from the derivatives of the moment generation functions evaluated at the origin.

1.2.3 Central Moments

The central moments of f(x,y) are defined as :

$$\mu_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x - \bar{x})^p.(y - \bar{y})^q.f(x,y).dx.dy \qquad (2)$$

where $\bar{x} = m_{10}/m_{00}$, $\bar{y} = m_{01}/m_{00}$. The central moments μ_{pq} defined in (2) may easily be shown invariant under translation and can also be expressed in terms of the moments m_{pq} defined in (1).

For a digital image, the double integrals in m_{pq} and μ_{pq} can be approximated by double summations as follows :

$$m_{pq} = \sum_{i=0}^{M} \sum_{j=0}^{N} i^p.j^q.f(i,j) \qquad (1.1)$$

$$\mu_{pq} = \sum_{i=0}^{M} \sum_{j=0}^{N} (i - \bar{i})^p.(j - \bar{j})^q.f(i,j) \qquad (2.1)$$

where $\bar{i} = m_{10}/m_{00}$, $\bar{j} = m_{01}/m_{00}$. The summation limits M and N are the dimensions of the intensity matrix f(i,j) in which i and j are the discrete locations of the image pixels. For many industrial applications images can be represented in black and white. Only size, contour, resemblance and contiguity are important, not the color or the shade of the object. In this case the intensity function f(i,j) would only have the values 0 or 1.

The computations of m_{pq} consist of multiplying the function f(i,j) by a corresponding $i^p j^q$ and integrating the results. In the case of a contiguous image, the function f(i,j) is always 1 inside the boundary of the object in the image, 0 for the background. The double-summations of the nominals of order 3 or less (i^0j^0, i^0j^1, ... , i^3j^0) are:

$$m_{00} = \sum_{i=0}^{M} \sum_{j=0}^{N} i^0.j^0 \tag{1.a}$$

$$m_{01} = \sum_{i=0}^{M} \sum_{j=0}^{N} i^0.j^1 \tag{1.b}$$

$$m_{02} = \sum_{i=0}^{M} \sum_{j=0}^{N} i^0.j^2 \tag{1.c}$$

$$m_{03} = \sum_{i=0}^{M} \sum_{j=0}^{N} i^0.j^3 \tag{1.d}$$

$$m_{10} = \sum_{i=0}^{M} \sum_{j=0}^{N} i^1.j^0 \tag{1.e}$$

$$m_{11} = \sum_{i=0}^{M} \sum_{j=0}^{N} i^1.j^1 \tag{1.f}$$

$$m_{12} = \sum_{i=0}^{M} \sum_{j=0}^{N} i^1.j^2 \tag{1.g}$$

$$m_{20} = \sum_{i=0}^{M} \sum_{j=0}^{N} i^2.j^0 \tag{1.h}$$

$$m_{21} = \sum_{i=0}^{M} \sum_{j=0}^{N} i^2.j^1 \tag{1.i}$$

$$m_{30} = \sum_{i=0}^{M} \sum_{j=0}^{N} i^3.j^0 \tag{1.j}$$

Computing these double-summations is lengthy due to the recursive nature of the calculations.

The detailed theory behind these derivations can be found in Elliot [23]. In this paragraph we proceed to use the above in calculating the central moments.

From (2.1), the central moments of order 3 are as follows:

$$\mu_{00} = \sum_{i=0}^{M} \sum_{j=0}^{N} (i - \bar{i})^0 . (j - \bar{j})^0 . f(i,j) \qquad (2.a)$$

$$\mu_{01} = \sum_{i=0}^{M} \sum_{j=0}^{N} (i - \bar{i})^0 . (j - \bar{j})^1 . f(i,j) \qquad (2.b)$$

$$\mu_{02} = \sum_{i=0}^{M} \sum_{j=0}^{N} (i - \bar{i})^0 . (j - \bar{j})^2 . f(i,j) \qquad (2.c)$$

$$\mu_{03} = \sum_{i=0}^{M} \sum_{j=0}^{N} (i - \bar{i})^0 . (j - \bar{j})^3 . f(i,j) \qquad (2.d)$$

$$\mu_{10} = \sum_{i=0}^{M} \sum_{j=0}^{N} (i - \bar{i})^1 . (j - \bar{j})^0 . f(i,j) \qquad (2.e)$$

$$\mu_{11} = \sum_{i=0}^{M} \sum_{j=0}^{N} (i - \bar{i})^1 . (j - \bar{j})^1 . f(i,j) \qquad (2.f)$$

$$\mu_{12} = \sum_{i=0}^{M} \sum_{j=0}^{N} (i - \bar{i})^1 . (j - \bar{j})^2 . f(i,j) \qquad (2.g)$$

$$\mu_{20} = \sum_{i=0}^{M} \sum_{j=0}^{N} (i - \bar{i})^2 . (j - \bar{j})^0 . f(i,j) \qquad (2.h)$$

$$\mu_{21} = \sum_{i=0}^{M} \sum_{j=0}^{N} (i - \bar{i})^2 . (j - \bar{j})^1 . f(i,j) \qquad (2.i)$$

$$\mu_{30} = \sum_{i=0}^{M} \sum_{j=0}^{N} (i - \bar{i})^3 . (j - \bar{j})^0 . f(i,j) \qquad (2.j)$$

From (2.1) page 5, it is quite simple to express the central moments in terms of the ordinary moments :

From (2.a), $\mu_{00} = m_{00}$

From (2.b), $\mu_{01} = m_{01}-(m_{01}/m_{00}).m_{00} = 0$

From (2.c), $\mu_{02} = m_{02}-m_{01}^2/m_{00} = m_{02}-\bar{y}.m_{01}$

From (2.d), $\mu_{03} = m_{03}-3.\bar{y}.m_{02}.+2.\bar{y}^2.m_{01}$

From (2.e), $\mu_{10} = m_{10}-(m_{10}/m_{00}).m_{00} = 0$

From (2.f), $\mu_{11} = m_{11}-(m_{01}/m_{00}).m_{10}$

From (2.g), $\mu_{12} = m_{12}-2.\bar{y}.m_{11}-\bar{x}.m_{02}+2.\bar{y}^2.m_{10}$

From (2.h), $\mu_{20} = m_{20}-m_{10}^2/m_{00} = m_{02}-\bar{x}.m_{01}$

From (2.i), $\mu_{21} = m_{21}-2.\bar{x}.m_{11}-\bar{y}.m_{20}+2.\bar{x}^2.m_{01}$

From (2.j), $\mu_{30} = m_{30}-3.\bar{x}.m_{20}.+2.\bar{x}^2.m_{10}$

1.2.4 Fundamental Theorem of Moment Invariant

To relate the moments to the theory of invariant algebra, one may first expand the exponential term in the moment generation function to obtain:

$$M(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \sum_{p=0}^{\infty} \frac{1}{p!} \cdot (ux+vy)^{p}.f(x,y).dx.dy$$

Now after using the binomial expansion and carrying out integration:

$$M(u,v) = \sum_{p=0}^{\infty} \frac{1}{p!} \cdot (\mu_{p0}, \ldots, \mu_{0p},)(u,v)^{p}.$$

Definitions :

Invariants - An invariant of a single quantic is such a function of the coefficients in that quantic, that it needs at most to be multiplied by a factor which is a function only of the coefficients in any scheme of linear transformation to be made equal to the same function. Similarly for an invariant of two or more quantics is such a function of the two or more sets of coefficients in those quantics, that it needs at most to be multiplied by a factor which is a function only of the coefficients in any scheme of linear transformation, to be made equal to the same function.

Quantics or Forms - A function of any number of variables x, y, z, ... which is rational, integral and homogeneous in those variables is called a quantic in x, y, z, ... If there are two variables, x, y, the quantic is called a binary quantic. If three, then it is called a ternary, if q, a q-ary form. The degree of a quantic in the variables x, y, z, ... is generally spoken of as its order. Quantics of the first, second, third, fourth, ... are called linear, quadratics, cubics, quatrics, ...

The following homogeneous polynomial of two variables u and v,

$$f = a_{p0} \cdot u^p + \binom{p}{1} \cdot a_{p-1,1} \cdot u^{p-1} \cdot v + \binom{p}{2} \cdot a_{p-2,2} \cdot u^{p-2} \cdot v^2 + \ldots + \binom{p}{p-1} \cdot a_{1,p-1} \cdot u \cdot v^{p-1} + a_{0p} \cdot v^p$$

is called a binary algebraic form, or simply a binary form, of order p. Using a notation introduced by Cayley, the above form may be written as:

$$f = (a_{po};\ a_{p-1,1};\ \ldots\ ;\ a_{1,p-1};\ a_{0p})(u,v)^p$$

a homogeneous polynomial f(a) of the coefficients a_{p0}, ... , a_{0p} is an algebraic invariant of weight α, if:

$$f(a'_{p0},\ldots\ldots,a'_{0p}) = \Delta^{\alpha}\, f(a_{p0},\ldots\ldots,a_{0p})$$

where $a'_{p0},\ldots\ldots,a'_{0p}$ are the new coefficients obtained from substituting the following general linear transformation into the original form.

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \beta & \Gamma \\ \sigma & \tau \end{bmatrix} \begin{bmatrix} u' \\ v' \end{bmatrix},$$

where

$$\Delta = \begin{vmatrix} \beta & \Gamma \\ \sigma & \tau \end{vmatrix} \neq 0.$$

if $\alpha=0$ the invariant is called an absolute invariant; if $\alpha\neq 0$ it is called a relative invariant. The invariant defined above may depend upon the coefficients of more than one form.

Theorem : If the algebraic form of order p has an algebraic invariant,

$$f(a'_{p0},\ldots\ldots,a'_{0p}) = \Delta^{\alpha} f(a_{p0},\ldots\ldots,a_{0p})$$

then the moment of order p has an algebraic invariant

$$f(\mu'_{p0},\ldots\ldots,\mu'_{0p}) = |J|\ \Delta^{\alpha} f(\mu_{p0},\ldots\ldots,\mu_{0p})$$

where J is the Jacobian of the transformation.

The importance of this theorem is that an invariant function of moments can be found once a corresponding algebraic function exists.

A point which should be emphasized is the generality involved in linear transformations. The only restriction was $\Delta\neq 0$.

1.2.5 The Normalized Central Moments

Under the similitude transformation, *i.e.*, the equal change of size in both the x and the y,

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \beta & 0 \\ 0 & \beta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \qquad \beta \text{ - constant,}$$

each coefficient of any algebraic form is an invariant

$$a'_{pq} = \beta^{p+q} . a_{pq},$$

For moment invariant we have:

$$\mu'_{pq} = \beta^{p+q+2} . \mu_{pq},$$

by eliminating β between the zeroth order relation,

$$\mu' = \beta^2 . \mu$$

and the remaining ones, we have the following absolute similitude moment invariants:

$$\frac{\mu'_{pq}}{\mu'^{(p+q)/2 + 1}} = \frac{\mu_{pq}}{\mu^{(p+q)/2 + 1}}, \qquad p+q = 2,3, \ldots$$

and $\mu'_{10} = \mu'_{01} = 0$.

As shown previously, the central moments μ_{pq} are simple combinations of the moments m_{pq}.
The normalized central moments, denoted by η_{pq} can now be defined as:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{(p+q)/2 + 1}}, \qquad p+q = 2,3, \ldots \tag{3}$$

These are invariant to size change as well as translation.

From (3) it is quite simple to express the normalized central moments in terms of the central moments :

$$\eta_{00} = \mu_{00} = m_{00} \tag{3.a}$$

$$\eta_{01} = \frac{\mu_{01}}{\mu_{00}^{1/2} + 1} = 0 \tag{3.b}$$

$$\eta_{02} = \frac{\mu_{02}}{\mu_{00} + 1} \tag{3.c}$$

$$\eta_{03} = \frac{\mu_{03}}{\mu_{00}^{3/2} + 1} \tag{3.d}$$

$$\eta_{10} = \frac{\mu_{10}}{\mu_{00}^{1/2} + 1} = 0 \tag{3.e}$$

$$\eta_{11} = \frac{\mu_{11}}{\mu_{00} + 1} \tag{3.f}$$

$$\eta_{12} = \frac{\mu_{12}}{\mu_{00}^{3/2} + 1} \tag{3.g}$$

$$\eta_{20} = \frac{\mu_{20}}{\mu_{00} + 1} \tag{3.h}$$

$$\eta_{21} = \frac{\mu_{21}}{\mu_{00}^{3/2} + 1} \tag{3.i}$$

$$\eta_{30} = \frac{\mu_{30}}{\mu_{00}^{3/2} + 1} \tag{3.j}$$

1.2.6 The Seven Moment Invariants

A set of seven invariant moments (ϕ), invariant to translation, scale change and rotation, has been derived from the normalized central moments. Detailed description and derivation can be found in Hu [1,2]. They are :

$$\phi_1 = \eta_{20}+\eta_{02} \tag{4.a}$$

$$\phi_2 = (\eta_{20}-\eta_{02})^2+4.\eta^2_{11} \tag{4.b}$$

$$\phi_3 = (\eta_{30}-3.\eta_{12})^2+(3.\eta_{21}-\eta_{03})^2 \tag{4.c}$$

$$\phi_4 = (\eta_{30}+\eta_{12})^2+(\eta_{21}+\eta_{03})^2 \tag{4.d}$$

$$\begin{aligned}\phi_5 = {} & (\eta_{30}-3.\eta_{12}).(\eta_{30}+\eta_{12}).\ [(\eta_{30}+\eta_{12})^2-3.(\eta_{21}+\eta_{03})^2] \\ & +(3.\eta_{21}-\eta_{03}).(\eta_{21}+\eta_{03}).\ [3.(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2]\end{aligned} \tag{4.e}$$

$$\begin{aligned}\phi_6 = {} & (\eta_{20}-\eta_{02}).\ [(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2] \\ & +4.\eta_{11}.(\eta_{30}+\eta_{12}).(\eta_{21}+\eta_{03})\end{aligned} \tag{4.f}$$

$$\begin{aligned}\phi_7 = {} & (3.\eta_{12}-\eta_{30}).(\eta_{30}+\eta_{12}).\ [(\eta_{30}+\eta_{12})^2-3.(\eta_{21}+\eta_{03})^2] \\ & +(3.\eta_{21}-\eta_{03}).(\eta_{21}+\eta_{03}).\ [3.(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2]\end{aligned} \tag{4.g}$$

The skew orthogonal invariant (ϕ_7) is used for distinguishing mirror images because it varies considerably under mirroring.

The method described in this section, can be generalized to accomplish pattern identification not only independent of position, size and orientation but also independent of parallel projection, see Hu [1,2] for derivations.

Chapter 2. RELEVANT LITERATURE SURVEY

Recognition of visual patterns and characters independent of position, size, and orientation in the visual field has been a research subject since 1962. In the following paragraphs a summary of this research is presented in chronological order :

Hu [1,2] : Reported in his paper, in 1962, the mathematical foundation of two-dimensional moment invariants and their applications to visual information processing. His results show that recognition schemes based on these invariants could be truly position, size and orientation independent, and also flexible enough to learn almost any set of patterns. Hu adapted the moment invariant method to visual pattern recognition. Other authors succeeding Hu extended his work only slightly, through specific applications. The moment invariants method was never considered for any industrial applications, because it is computationally very costly. Its use in defense applications are extensive, however. This thesis extends Hu's theory, through an efficient simplification, so as to make its industrial applications tractable.

Alt [3] : Applied Hu's results, in 1963, to the recognition of the letters and numerals of a particular printed font and similar patterns.

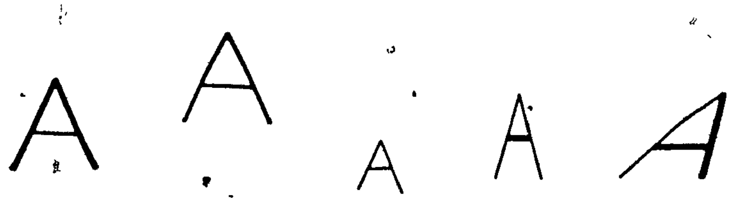


Figure 2.1 Examples of "similar" patterns.

In comparing his system with others which have been proposed, we find both advantages and drawbacks. These other methods use either coincidence - the pattern to be read is matched with a standard pattern, and the requirement for agreement, within a specified tolerance, is imposed - or they concentrate upon certain local or topological properties of the character to be recognized, such as corners, branch points, and closed loops. An example of an instance where moment invariants fail is furnished by the modern Hebrew alphabet, in which, *e.g.*, the characters corresponding to d and r differ only in that the former has a sharp corner where the latter is rounded. This difference would have no more effect on moments than some slight noise or change in type font. In fact, it is the kind of distinction which we wish to disregard; for in the Latin alphabet it is frequently meaningless (see Fig. 2.2).

Figure 2.2 Two Hebrew characters, d and r, differ only in the sharpness of a corner.

Lambert [4] : Performed experiments, in 1969, on the classifications of printed characters using moment invariant features with a reported accuracy of 95%, *i.e.*, 95% of characters read are correctly identified.

Casy [5] : Used moments, in 1970, as a preprocessing tool to normalize patterns of handprinted characters.

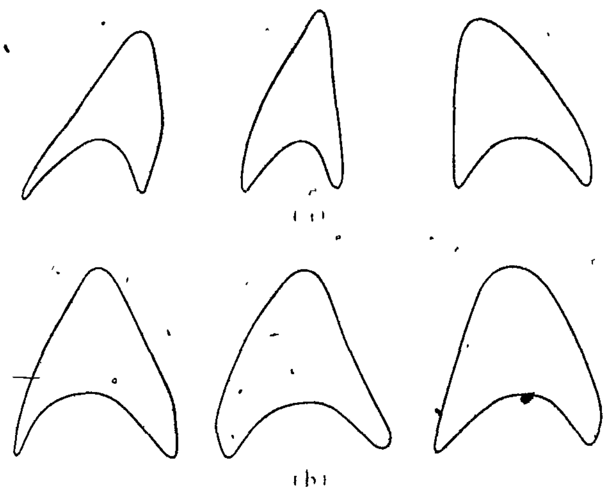


Figure 2.3 Sample contours: (a) input (derived from handprinted A's): (b) normalized.

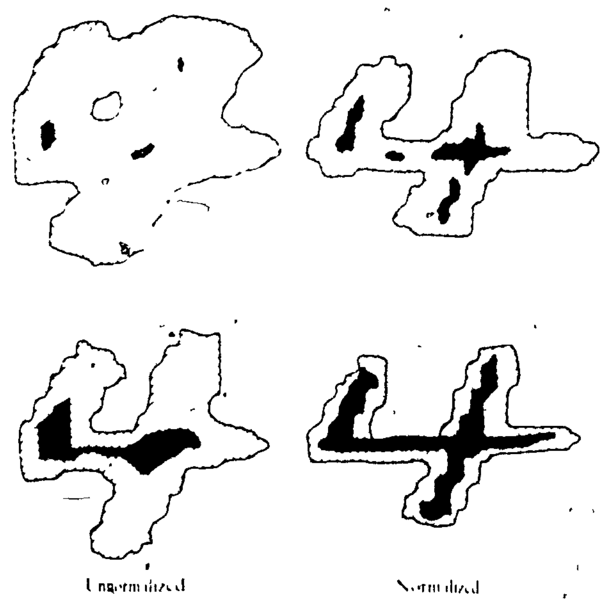


Figure 2.4 Superimposed patterns.

Handprinted characters can be made to appear more uniform, *i.e.,* more like machine print, if an appropriate linear transformation is performed on each input pattern. The transformation can be implemented electronically by programming a flying-spot raster-scanner to scan at a number of specified angles in addition to scans along the principal axes. Alternatively, curve-follower normalization can be achieved by transforming the coordinate waveforms in a linear combining network. Second order moments of the pattern are convenient properties to use in specifying the transformation. By mapping the original pattern into one having a scalar moment matrix all linear pattern variations can be removed. Comparison experiments with three sets of handprinted numerals showed that error rates were reduced by integral factors if the patterns were normalized before scanning recognition (see Fig. 2.3 and 2.4 for normalized patterns).

Smith *et al.* **[6] :** Reported, in 1971, the results of a study undertaken to determine the feasibility of automatic interpretation of ship photographs using the spatial moments of the images as characterizing features. The photo interpretation consisted of estimating the location, orientation, dimensions, and heading of the ship. The study used simulated images in which the outline of the ship was randomly filled with black and white cells to give a low-resolution high-contrast image of the ship such as might be obtained by a high-resolution radar.

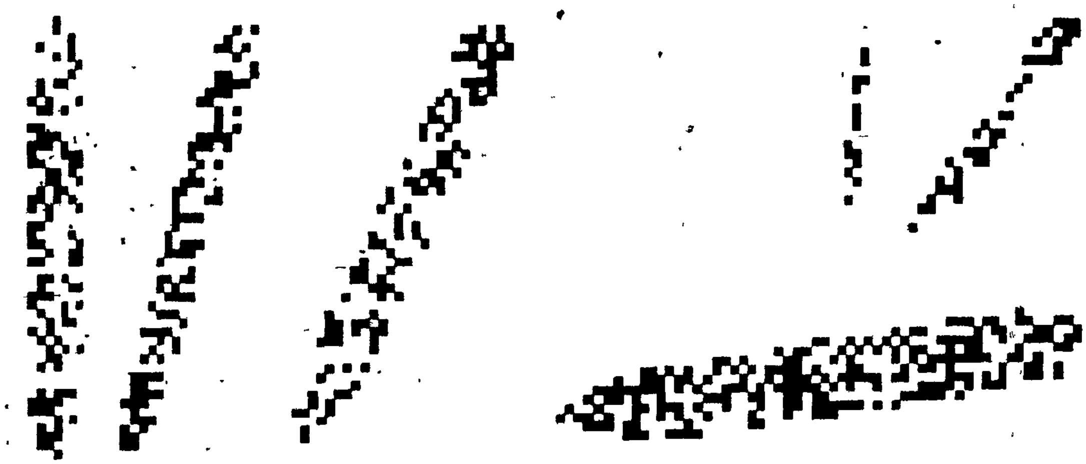


Figure 2.5 Six examples of simulated ship images.

Hall *et al.* **[7-9]** : Used spatial moments, in 1976, as one of the selected features in categorization of profusion of opacities in medical X-rays.

Dudani *et al.* **[10]** : Addressed in his paper, in 1977, the problem of the automatic interpretation of optical images of three-dimensional scenes. He was specifically concerned with the automatic recognition of aircraft types from optical images. An experimental system was described in which certain features called *moment invariants* are extracted from binary television images and are then used for automatic classification. This experimental system has exhibited a significantly lower error rate than human observers in a limited laboratory test involving 132 images on six aircraft types. Preliminary indications were that this performance could have been extended to a wider class of objects.

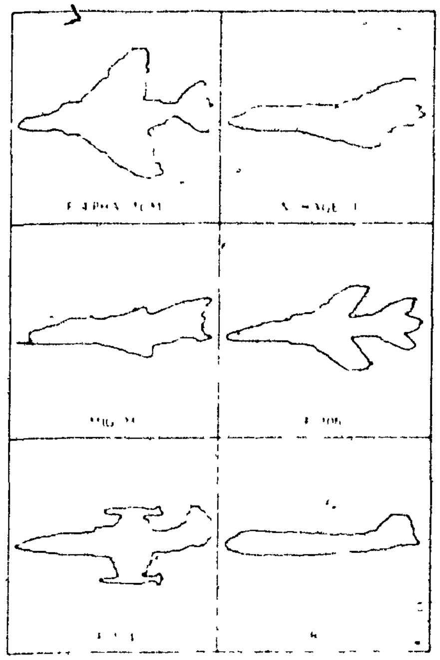


Figure 2.6 Typical images obtained with the experimental image acquisition system.

In his investigation, a recognition class consisting of only six aircraft types were used. It was difficult to arrive at any meaningful results regarding the relationship of recognition accuracy to the number of aircraft in the given class because of the fact that similarity or dissimilarity in shapes of aircraft under consideration greatly affects the recognition accuracy. However, for the aircraft used in the recognition class, the accuracy of correct classification did not increase significantly when lowering the number of aircraft in the recognition class to three.

Sadjadi *et al.* **[11]** : Extended, in 1978, the applications of the method of moment invariants to more complex two-dimensional images without changing the theory. His work was applied in space, spy satellites, and in the guidance systems of long range missiles. Although Sadjadi *et al.* tried to recognize complex images, their work and applications were of a great value to this research thesis.

Sadjadi *et al.* **[14] :** Proposed, in 1980, the use of three-dimensional moment invariants as a tool for the recognition of three-dimensional objects independent of size, position and orientation.

The generalization of the result of 2-D moment invariants which had linked the 2-D moment invariants to binary quantics is done by linking 3-D moment invariants to ternary quantics. The existence and number of n^{th} order moments in two and three dimensions is explored.

The three-dimensional moments of order p+q+r of a density (or intensity) function f(x,y,z) are defined in terms of the Riemann integrals as:

$$m_{pqr} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p.y^q.z^r.f(x,y,z).dx.dy.dz$$

It is assumed that the function f(x,y,z) is piecewise continuous and therefore bounded and it is zero in R^3 space except in a finite part. Based on this assumption it can be proven that the sequence $\{m_{pqr}\}$ determines uniquely f(x,y,z).

The moment generation function for three dimensional moments may be defined as:

$$M(u,v,w) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \sum_{p=0}^{\infty} \exp\,(ux+vy+wz).f(x,y,z).dx.dy$$

which can be expanded into a power series:

$$M(u,v,w) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \sum_{p=0}^{\infty} \frac{1}{p!}.(ux+vy+wz)^{p}.f(x,y,z).dx.dy.dz$$

The central moments μ_{pqr} are defined as:

$$\mu_{pqr} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x-\bar{x})^{p}.(y-\bar{y})^{q}.(z-\bar{z})^{r}.f(x,y,z).dx.dy.dz$$

where $\bar{x} = m_{100}/m_{000}$, $\bar{y} = m_{010}/m_{000}$, $\bar{z} = m_{001}/m_{000}$

.

.

.

The normalized central moments are defined similarly, and the 3-D moment invariants are then derived (see Sadjadi [14] for detailed derivations). This generalization is not trivial because of the difficulties which are present in the derivation of general ternary quantic invariant forms upon which three dimensional moment invariant rely.

As a special but important subset of general ternary quantics, the class of ternary quadratic forms was explored and several geometrical interpretations of invariants were given. It was stated that every geometric property of a quadratic surface which remains invariant under rotation and translation can be presented in terms of its absolute forms.

The 3-Dimensional moment invariants method proposed here could be of a great value to the continuation of this research thesis since the *Delta Method*, proposed in chapter 3, could be ultimately expanded to handle the recognition of 3-D objects.

Reddi [16] : Presented, in 1981, radial and angular moments of images and showed the methods for deriving moment functions that are invariant with respect to rotation, translation, reflection, and size change without the aid of the theory of algebraic invariants. Hu's invariants were expressed in terms of these radial and angular moments, and Reddi claimed that this facilitates visual inspection of invariance properties.

Let $g(r,\theta)$ be the intensity function in polar coordinates [*i.e.*, $f(x,y) \equiv g(r,\theta)$] and define the following radial and angular moments (as defined in (1)):

$$\Phi_r(k,g) = \int_0^{\infty} r^k.g(r,\theta).dr$$

$$\Phi_\theta(p,q,g) = \int_{-\pi}^{\pi} \cos^p \theta.\sin^q \theta.g(r,\theta).d\theta$$

$$\Phi_\theta(g) = \Phi_\theta(0,0,g)$$

$$\Phi(k,p,q,g) = \int_0^{\infty}\int_{-\pi}^{\pi} r^k.g(r,\theta).\cos^p \theta.\sin^q \theta.d\theta.dr. \qquad (1')$$

Expressing μ_{pq} from (2) in polar coordinates we have

$$\mu_{pq} = \int_0^{\infty}\int_{-\pi}^{\pi} r^{p+q+1}.\cos^p \theta.\sin^q \theta.g(r,\theta).dr.d\theta$$

$$= \Phi(p+q+1,p,q,g)$$

since x=r.cos θ, y=r.sin θ and dx.dy=r.dr.dθ.

The seven moment invariants ((4.1)-(4.7)) derived by Hu [1,2] could be then expressed in terms of angular and radial moment as follows :

$$M_1 = \Phi_r(3,\Phi_\theta(g))$$
$$M_2 = |\Phi_r(3,\Phi_\theta(g.e^{j2\theta}))|^2$$
$$M_3 = |\Phi_r(4,\Phi_\theta(g.e^{j3\theta}))|^2$$
$$M_4 = |\Phi_r(4,\Phi_\theta(g.e^{j\theta}))|^2$$
$$M_5 = RP\ \{\Phi_r(4,\Phi_\theta(g.e^{j3\theta})).(\Phi_r^3(4,\Phi_\theta(g.e^{-j\theta}))\}$$
$$M_6 = RP\ \{\Phi_r(3,\Phi_\theta(g.e^{j2\theta})).(\Phi_r^2(4,\Phi_\theta(g.e^{-j\theta}))\}$$
$$M_7 = IP\ \{\Phi_r(4,\Phi_\theta(g.e^{j3\theta})).(\Phi_r^3(4,\Phi_\theta(g.e^{-j\theta}))\}.$$

Here RP and IP stand for real and imaginary parts, respectively. The functions M_1 through M_6 are invariant with respect to rotation and reflection, whereas M_7 changes sign under reflection.

The advantage of using radial and angular moments is that it is simple to write the invariants directly (without going through the theory of algebraic invariants as Hu does). Thus we may write:

$$|\Phi_r(k,\Phi_\theta(e^{jl\theta}))|^2$$

as an invariant under rotation and reflection for any k and l since changing (θ) to ($\theta+\alpha$) or ($-\theta$) leaves the expression unchanged. Also instead of having a weighting function such as r^k, one can have exponentials and similar functions in r. For instance:

$$|\Phi_r(0,\Phi_\theta(e^{-\alpha r}.e^{jl\theta}))|^2$$

can be used as an invariant.

Radial and angular moments can be made invariant with respect to size in a simple manner. Let $g_\alpha=g(\alpha.r,\theta)$ denote the image contracted/expanded by α and M'_i denote the new ith moment function of g_α. Since

$$\Phi_r(k,g_\alpha) = \alpha^{-(k+1)}.\Phi_r(k,g),$$

we have:

$$M_1' = \alpha^{-4}.M_1$$
$$M_2' = \alpha^{-8}.M_2$$
$$M_3' = \alpha^{-10}.M_3$$
$$M_4' = \alpha^{-10}.M_4$$
$$M_5' = \alpha^{-20}.M_5$$
$$M_6' = \alpha^{-14}.M_6$$
$$M_7' = \alpha^{-20}.M_7$$

and hence M_2 through M_7 can be made size invariant as follows:

$$M_2^* = M_2/M_1^2$$
$$M_3^* = M_3/M_1^{2.5}$$
$$M_4^* = M_4/M_1^{2.5}$$

$$M_5^* = M_5/M_1^5$$
$$M_6^* = M_6/M_1^{3.5}$$
$$M_7^* = M_7/M_1^5$$

It may be noted that M_5^* is inversely proportional to the fifth power of M_1 thus making it very sensitive to variations in M_1.

Although it is easier to derive the angular and radial moment invariants, it is, however, more time consuming to calculate them. Comparing the cartesian moments in (1) to the angular and radial moments in (1'), the amount of calculations to be performed for each pixel in the intensity matrix $g(r,\theta)$ is much greater than that in the $f(x,y)$ matrix (compare calculating $x^p.y^q.f(x,y)$ in (1) to calculating $r^k.\cos^p\theta.\sin^q\theta.g(r,\theta)$ in (1')).

It is worth noting that the *delta method* derived in the next chapter could be expanded to apply to the radial and angular moment invariants.

Teh *et al.* **[17]** : Presented, in 1985, a better formulation of the moment invariants using the numerical integration approaches. The undersampling and digitizing effects of a digital image as well as the quantization effect of the intensity levels on moment invariants were also presented.

The transformation of $f(x,y)$ into its discrete version $f(i,j)$ consists of sampling the continuous image function with an M X N array of points (pixels) and quantizing the continuous intensity function into K discrete levels. The sampling process can be viewed as multiplying $f(x,y)$ by a sampling function $s(x,y)$ to obtain $f(i,j)$.

Two different sampling functions were considered, the traditional rectangular sampling function and the hexagonal function, defined by :

$$s_R(x,y) = \sum_{M=-\infty}^{\infty} \sum_{N=-\infty}^{\infty} \epsilon.(x-M.\_x,y-N.\_y)$$

and

$$s_H(x,y) = \sum_{M=-\infty}^{\infty} \sum_{N=-\infty}^{\infty} \epsilon.(x - \frac{2M-N}{2} .\_x,y-N.\_y)$$

respectively. Two sampled versions of the test image, $f_R(i,j)$ and $f_H(i,j)$, were then computed by

$$f_R(i,j) = f(x,y) \times s_R(x,y)$$

and

$$f_H(i,j) = f(x,y) \times s_H(x,y)$$

respectively.

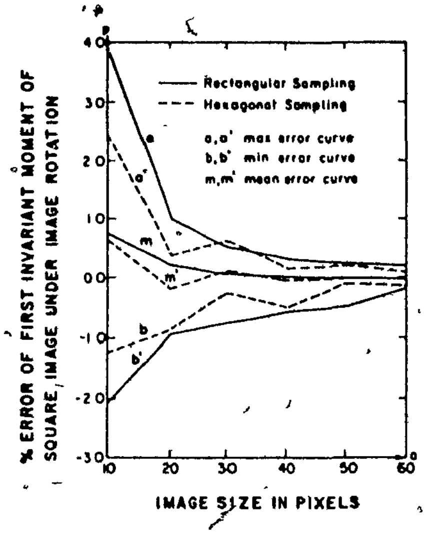


Figure 2.7 Rotation invariant error of a square image due to digitization. Teh et al. [17].

The set of seven invariant moments given by Hu are invariant for the case of continuous image intensity function. For digital processing the image intensity needs to be quantized and the formulation approximated by summations, therefore, the moments are expected not to be invariant due to the error incroduced by the approximations.

Possible better approximation methods to calculate the moment invariants by numerical integration approaches were discussed and the plotted results are shown in Figure 2.8.

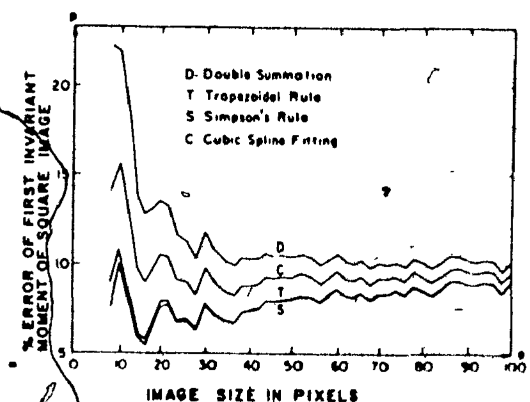


Figure 2.8 Error due to numerical approximation. Teh et al. [17].

The relevance of this work is diminished with the use of a new generation of digital cameras capable of handling considerably larger intensity matrices that would minimize the undersampling, quantization and digitization errors.

Hatamian [18] : Presented, in 1986, a fast algorithm and its single chip VLSI implementation for generating moments of two-dimensional digital images in real-time image processing applications. The basic building block of the algorithm is a single-pole digital filter implemented with a single accumulator. These filters are cascaded together in both horizontal and vertical directions in a highly regular structure which makes it very suitable for VLSI implementation. The chip has been implemented in 2.5 µ CMOS technology, it occupies 6100 µm X 6100 µm of silicon area. The chip can also be used as a general cell in a systolic architecture for implementing 2-D transforms having polynomial basis structure.

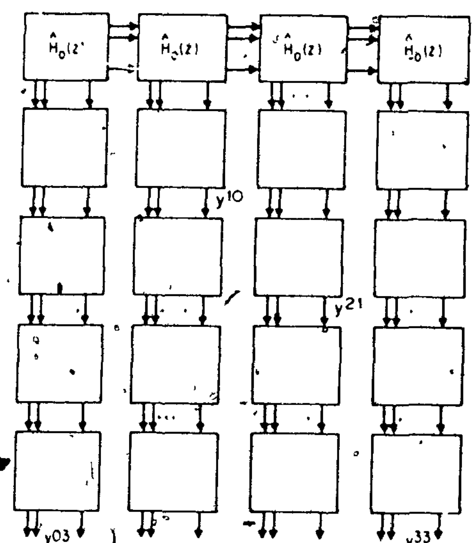


Figure 2.9 A 2-D digital filter structure for generating linear combination of moments of an image Hatamian [17].

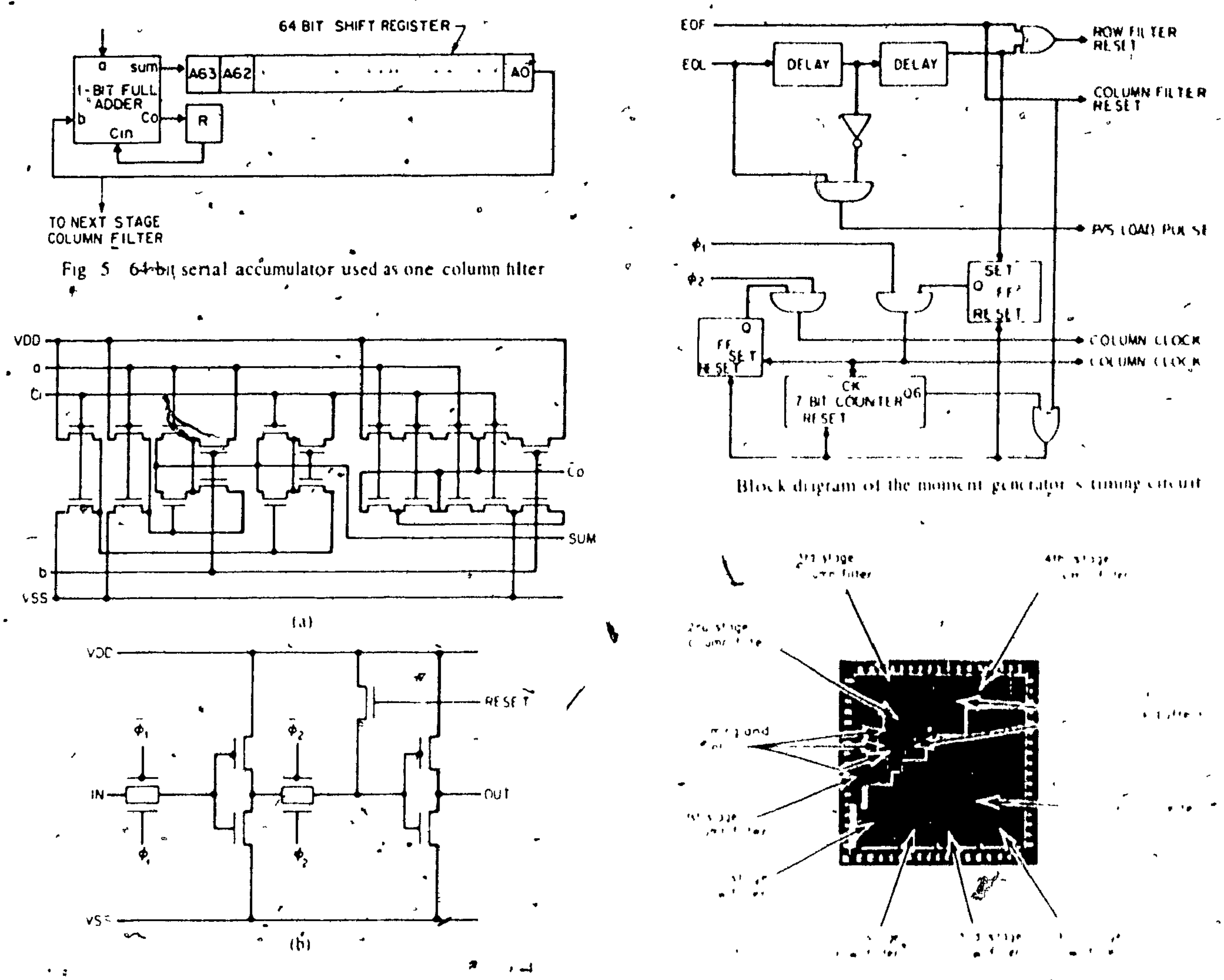


Figure 2.10 (a) Circuit diagram of one-bit CMOS transmission gate full adder used in column filters.
(b) Circuit diagram of the dinamic two phase-clock register.
(c) Block diagram of the moment generator's timing circuit.
(d) Photomicrograph of the moment generator chip.
Hatamian [17].

Chapter 3 AN ALTERNATIVE APPROACH: THE DELTA METHOD

3.1 Introduction

The computations of m_{pq} consist of multiplying the function f(i,j) by a corresponding $i^p j^q$ and integrating the results (see (1)). In the case of a contiguous image, the function f(i,j) is always 1 inside the boundary of the object in the image, 0 for the background.

The identification and verification method should take into consideration important factors such as size, contour, resemblance and contiguity and not the color or the shade of the ply fabric. To eliminate differences resulting from the presence of stripes, colors, shades or patterns, a black and white image (0 or 1) has been deliberately chosen to represent the object. This supplies necessary sufficient information about the object, while rejecting superflous and confusing information.

For a contiguous image all bits are "on" (equal to 1) and therefore all bytes inside the boundaries contain the unsigned binary integer, 255 (except those on the left and right-hand boundaries). The idea of the delta algorithm is quite simple, instead of performing the lengthy computations of (1.a)-(1.j) for each pixel, a line of pixels is chained and the computations are performed only once per line. This new algorithm simplifies the representation of the intensity matrix f(i,j), where f(i,j) could be represented now in bytes and programmed in a higher level language instead of bits which require low level language programming. It also reduced the scanning time by a factor of 8 since now bytes are scanned, instead of bits (see the straightforward approach [1-18]). For the first and the last byte of a given line of an image, up to 8 tests may have to be performed to determine the boundary (see subroutines BYTE-RIGHT and BYTE-LEFT and their description, in Appendix III). Once the boundaries have been established, an entire line of pixels is then considered as one entity, and the recursive (lengthy) calculations of the 2-D moments would be performed once per line rather than once per pixel (as in the straightforward approach), and if a hole is present the rest of the pixels in the line will be ignored to magnify the flaw.

In addition to these simplifications, the delta algorithm introduces great reduction in the time complexity of the computations resulting from its short-cut equations.

3.2 Detailed Derivation of the Algorithm

This algorithm utilizes new variables and subsequently new equations to represent the 2-D moments.

The variables are defined as follows:

δ : the number of chained pixels in row i. (see Figure 3.1)

X_i : the x-coordinate of the first pixel in row i.

Y_i : the y-coordinate of the first pixel in row i.

$m_{pq,i}$: the contribution of row i to the nominals m_{pq}.

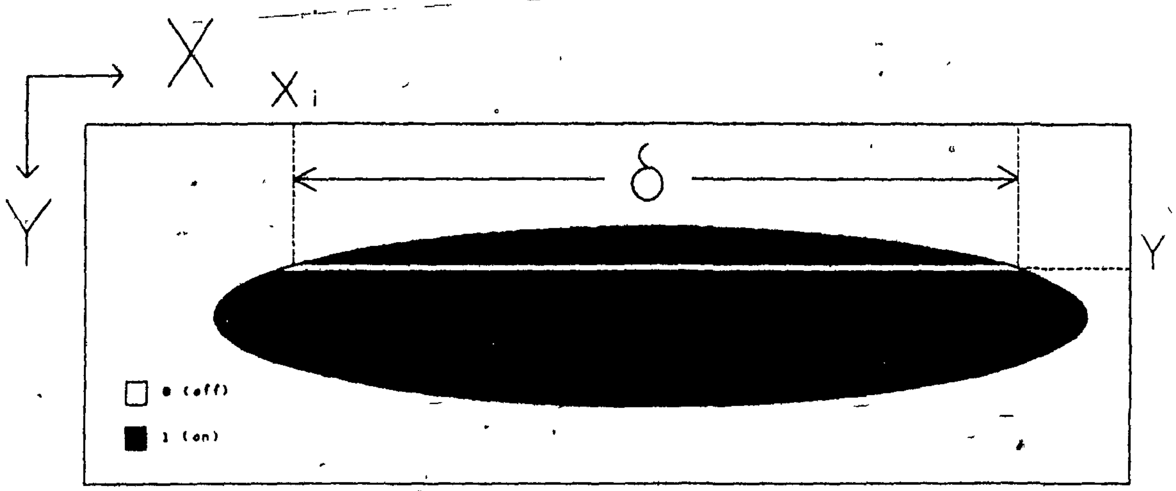


Figure 3.1 The Delta method.

Each $m_{pq,i}$ can be expressed in terms of X_i, Y_i, and δ.

From (1.a), $m_{00} = 1+1+1+1+ \ldots\ldots +1 = \delta$

From (1.b), $m_{01} = Y_i+Y_i+Y_i+ \ldots\ldots +Y_i = \delta.Y_i$

From (1.c), $m_{02} = Y_i^2+Y_i^2+Y_i^2+ \ldots\ldots +Y_i^2 = \delta.Y_i^2$

From (1.d), $m_{03} = Y_i^3+Y_i^3+Y_i^3+ \ldots\ldots +Y_i^3 = \delta.Y_i^3$

From (1.e), $m_{10} = X_i+(X_i+1)+(X_i+2)+(X_i+3)+ \ldots\ldots +(X_i+\delta-1)$

$= \delta.X_i+(0+1+2+3+4+5+ \ldots\ldots +\delta-1)$

$= \delta.X_i+(0+\delta-1)/2.\delta$

$= \delta.X_i+(\delta^2-\delta)/2$

From (1.f), $m_{11} = X_i.Y_i+(X_i+1).Y_i+(X_i+2).Y_i+ \ldots\ldots$

$+ (X_i+\delta-1).Y_i$

$= Y_i.[X_i+(X_i+1)+(X_i+2)+ \ldots\ldots +(X_i+\delta-1)]$

The term inside the brackets equals m_{10} therefore :

$m_{11} = Y_i.[\delta X_i+(\delta^2-\delta)/2]$

From (1.g), $m_{12} = X_i.Y_i^2+(X_i+1).Y_i^2+(X_i+2).Y_i^2+ \ldots\ldots$

$+ (X_i+\delta-1).Y_i^2$

$$= Y_i^2.[X_i+(X_i+1)+(X_i+2)+ \;\ldots\ldots\; +(X_i+\delta-1)]$$
$$= Y_i^2.[\delta X_i+(\delta^2-\delta)/2]$$

From (1.h), $m_{20} = X_i^2+(X_i+1)^2+(X_i+2)^2+ \;\ldots\ldots\; +(X_i+\delta-1)^2$

$$= X_i^2+X_i^2+2.X_i+1+ \;\ldots\ldots\; +X_i^2+2(\delta-1).X_i+(\delta-1)^2$$

By grouping terms and factoring out X_i, this becomes :

$$m_{20} = \delta.X_i^2+2.(0+1+2+3+4+ \;\ldots\ldots\; +\delta-1).X_i+(0+1+4+9+ + \;\ldots\ldots\; +(\delta-1)^2)$$

Using the polynomial theorem :

$$m_{20} = \delta.X_i^2+(\delta^2-\delta).X_i+ \sum_{n=0}^{\delta-1} n^2$$

where the last term:

$$\sum_{n=0}^{\delta-1} n^2 = \delta^3/3-\delta^2/2+\delta/6$$

So the total contribution from (1.h) is:

$$= \delta.X_i^2+(\delta^2-\delta).X_i+1/3\ \delta^3-1/2\ \delta^2+1/6\delta$$

From (1.i), $m_{21} = X_i^2.Y_i+(X_i+1)^2.Y_i+(X_i+2)^2.Y_i+ \;\ldots\ldots\; +(X_i+\delta-1)^2.Y_i$

$$= Y_i.[X_i^2+(X_i+1)^2+(X_i+2)^2+ \;\ldots\ldots\; +(X_i+\delta-1)^2]$$
$$= Y_i.[\text{contribution from one row of } m_{20}]$$
$$= Y_i.[\ \delta.X_i^2+(\delta^2-\delta).X_i+1/3\ \delta^3-1/2\ \delta^2+1/6\delta$$

From (1.j), $m_{30} = X_i^3+(X_i+1)^3+(X_i+2)^3+ \;\ldots\ldots\; +(X_i+\delta-1)^3$

$$= X_i^3+X_i^3+3.X_i^2+3X_i+1+X_i^3+3.2.X_i^2+3.2^2.X_i+1.2^3+ + \;\ldots\ldots\; +X_i^3+3(\delta-1).X_i^2+3.(\delta-1)^2X_i+(\delta-1)^3$$

By grouping terms similar to m_{20} this becomes :

$$m_{30} = \delta.X_i^3+3.(0+1+2+3+4+ \;\ldots\ldots\; +\delta-1).X_i^2+3.(0+1+4+ +9+ \;\ldots\ldots\; +(\delta-1)^2).Xi+(0+1+8+27+ \;\ldots\ldots\; +(\delta-1)^3)$$

Using the polynomial theorem :

$$m_{30} = \delta.X_i^3+3.(\delta^2-\delta)/2.X_i^2+3.[\delta^3/3-\delta^2/2-\delta/6].X_i + \sum_{n=0}^{\delta-1} n^3$$

where the last term:

$$\sum_{n=0}^{\delta-1} n^3 = \delta^4/4-\delta^3/2+\delta^2/4$$

So the total contribution from (1.j) is:

$$m_{30} = \delta.X_i^3+3.(\delta^2-\delta)/2.X_i^2+3.[\delta^3/3-\delta^2/2-\delta/6].X_i + \delta^4/4-\delta^3/2+\delta^2/4$$

The above can be summarized as follows :

From (1.a), $m_{00,i} = \delta$

From (1.b), $m_{01,i} = \delta.Y_i$

From (1.c), $m_{02,i} = \delta.Y_i^2$

From (1.d), $m_{03,i} = \delta.Y_i^3$

From (1.e), $m_{10,i} = \delta X_i+(\delta^2-\delta)/2$

From (1.f), $m_{11,i} = Y_i.[\delta X_i+(\delta^2-\delta)/2]$

From (1.g), $m_{12,i} = Y_i^2.[\delta X_i+(\delta^2-\delta)/2]$

From (1.h), $m_{20,i} = \delta.X_i^2+(\delta^2-\delta).X_i+\delta^3/3-\delta^2/2+\delta/6$

From (1.i), $m_{21,i} = Y_i.[\delta.X_i^2+(\delta^2-\delta).X_i+\delta^3/3-\delta^2/2+\delta/6]$

From (1.j), $m_{30,i} = \delta.X_i^3+3.(\delta^2-\delta)/2.X_i^2+3.[\delta^3/3-\delta^2/2+\delta/6].X_i + \delta^4/4-\delta^3/2+\delta^2/4$

We define the following abbreviations for the sums :

$$S1 = \sum_{n=0}^{\delta-1} n = (\delta^2-\delta)/2$$

$$S2 = \sum_{n=0}^{\delta-1} n^2 = (\delta^3/3-\delta^2/2+\delta/6)$$

$$S3 = \sum_{n=0}^{\delta-1} n^3 = (\delta^4/4-\delta^3/2+\delta^2/4)$$

The $m_{pq,i}$ represent the contributions to m_{pq} from each line of pixels, in another words:

$$m_{pq} = \sum_{i=0}^{N} m_{pq,i}$$

Using the S1, S2, and S3 simplifications, the $m_{pq,i}$ calculations are reduced to :

$$m_{00,i} = \delta \quad (1.a')$$

$$m_{01,i} = \delta.Y_i \quad (1.b')$$

$$m_{02,i} = \delta.Y_i^2 \quad (1.c')$$

$$m_{03,i} = \delta.Y_i^3 \quad (1.d')$$

$$m_{10,i} = \delta.X_i+S1 \quad (1.e')$$

$$m_{11,i} = Y_i.\ [\ \delta.X_i+S1\ ] = Y_i.m_{10,i} \quad (1.f')$$

$$m_{12,i} = Y_i^2.\ [\ \delta.X_i+S1\ ] = Y_i^2.m_{10,i} \quad (1.g')$$

$$m_{20,i} = \delta.X_i^2+2.S1.X_i+S2 \quad (1.h')$$

$$m_{21,i} = Y_i.m_{20,i} \quad (1.i')$$

$$m_{30,i} = \delta.X_i^3+3.S1.X_i^2+3.S2.X_i+S3 \quad (1.j')$$

Calculating the central moment (μ_{pq}), the normalized central moments (η_{pq}), and the moment invariants (ϕ's) is a simple task, which can be done quickly since these moments can all be represented as a linear combination of the 2-D moment (m_{pq}), which has to be calculated only once.

Chapter 4 DESCRIPTION OF THE IMAGING SYSTEM

4.1 Introduction

This thesis deals with the design and implementation of an automated vision system. The software and the hardware have been implemented and this chapter provides a detailed description of the integrated system.

4.1.1 Experimental Set-Up

The integrated system shown in the picture of Figure 4.1 is the result of research which achieved an industrially feasible, cost effective industrial vision system which combines a commercially available camera sub-system with a PC-AT. The system consists of of the following components:

1- Hewlett Packard PC Vectra (IBM AT compatible) equipped with an 80287 co-processor.

2- An IDETIX vision sub-system by MICRON TECHNOLOGIES INC. equipped with an IS256 OpticRAM, a 63701 microcomputer and a MOS digital camera.

3- 150 watt light source.

4- An optical bench 1.5m in length and a 1m by 1m blackboard (for the background).

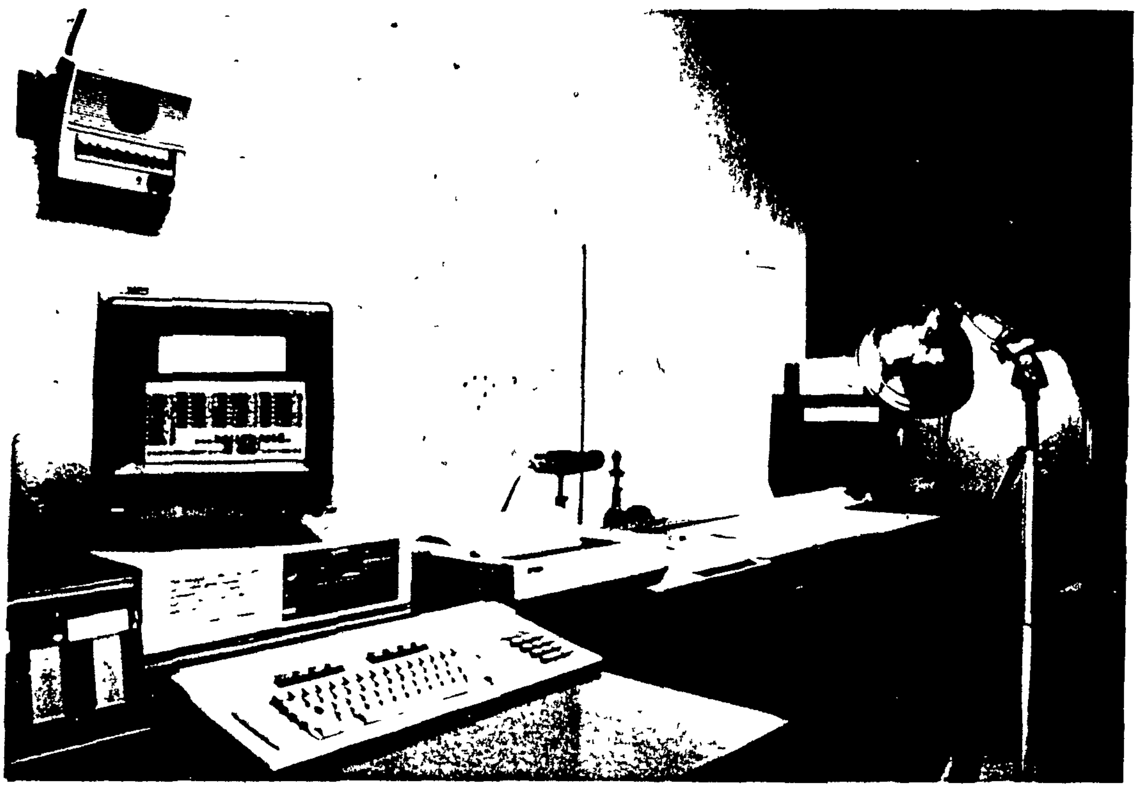


Figure 4.1 The Integrated Vision System.

4.2 The IDETIX System

The IDETIX is a simple, inexpensive solution to numerous applications requiring a low cost, all digital imaging sub-system. Its electro-optical system is suitable for use with any IBM PC/XT/AT compatible computer. The IDETIX has been designed to interface easily with customer-generated software.

The low cost of IDETIX is directly attributable to the technological advance represented by Micron's OpticRAM. In terms of cost per pixel, the OpticRAM represents a 1000X reduction in price over earlier generation image-sensing chips such as the CCD (Charge Coupled Device).

4.3 System Hardware

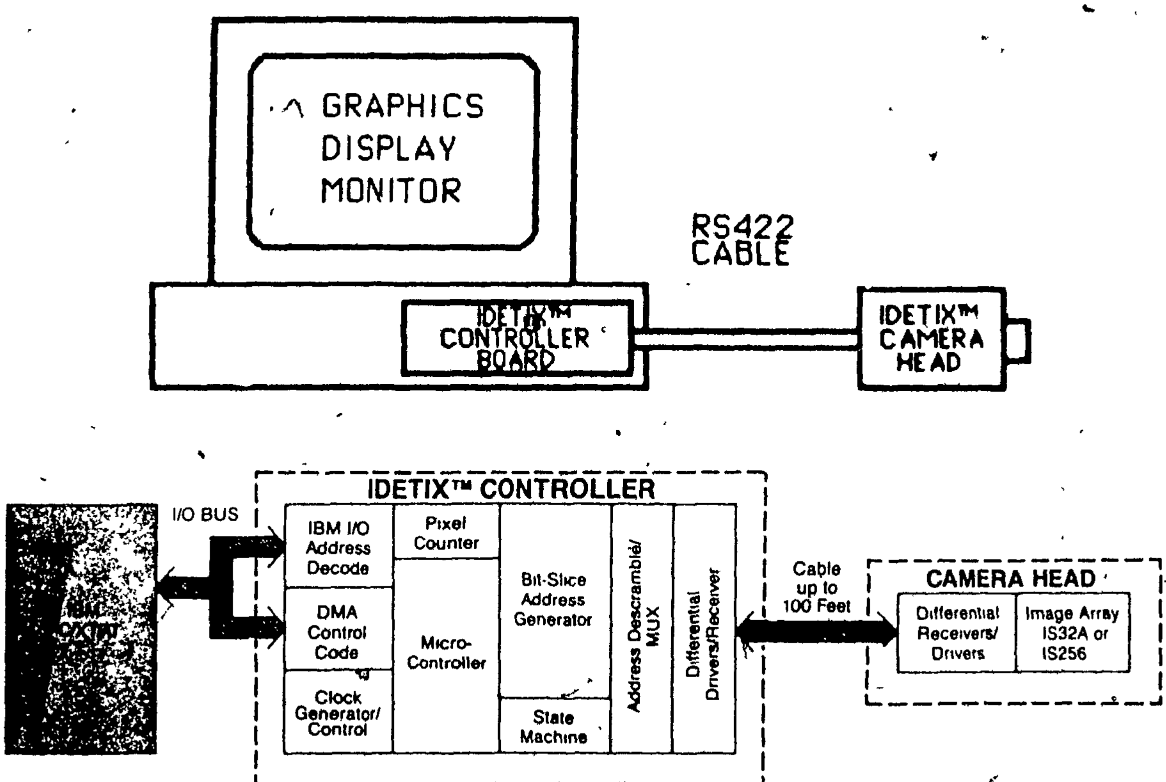


Figure 4.2 Hardware Block Diagram.
Micron Technologies Inc. [24].

4.3.1 The IDETIX Camera

The OpticRAM, the heart of the system is, located in the camera head (see Figure 4.3 for the camera head drawing). The camera head and host computer are connected via an RS422 cable up to 100 feet in length.

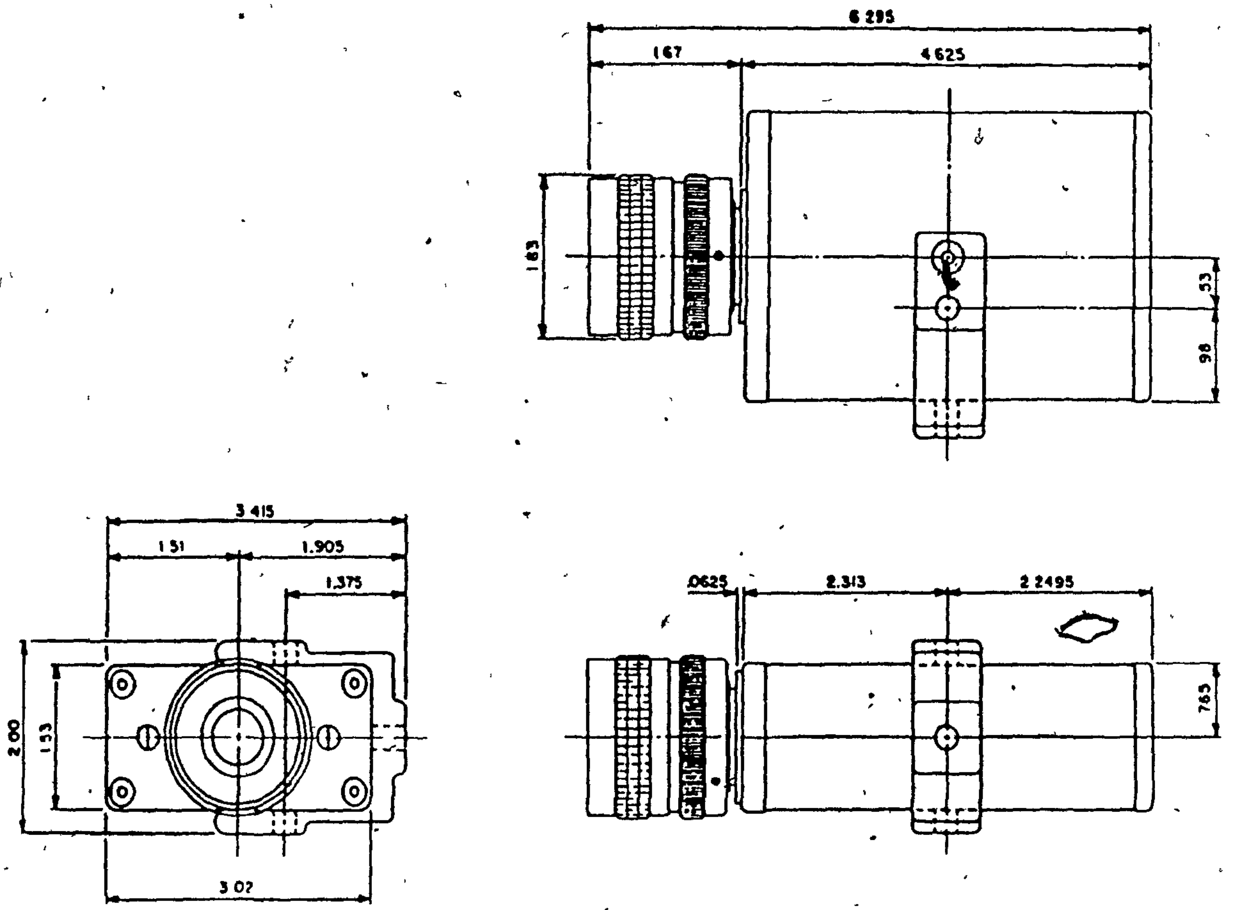
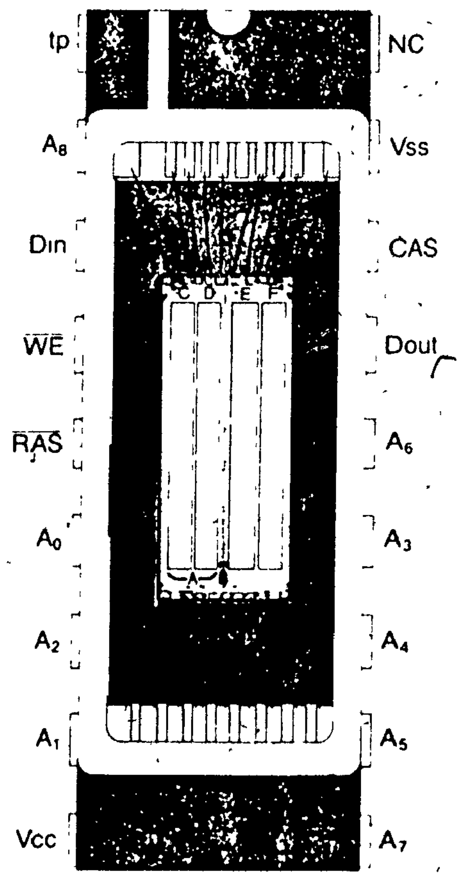


Figure 4.3 Camera Head Drawing.
Micron Technologies Inc. [24].

4.3.2 The OpticRAM

The IS256 OpticRAM image sensor is a solid-state device capable of sensing an image and translating it to digital computer-compatible signals. Each of the four arrays on the chip contains 65,536 sensors arranged as 128 rows by 512 columns of sensors (Figure 4.4 for whole chip diagram and Figure 4.5 for array pair topological information). In our application we utilized only one of the arrays since the arrays are separated by an optical "dead zone" 87 microns wide. However, all arrays can be used.



A One array pair as illustrated in topological information
B Column decoder spacing between array pairs 357.9μ
C Rows 0 127
D Rows 128 255
E Rows 255-383
F Rows 384 511

Figure 4.4 IS256 OpticRAM whole chip.
Micron Technologies Inc. [24].

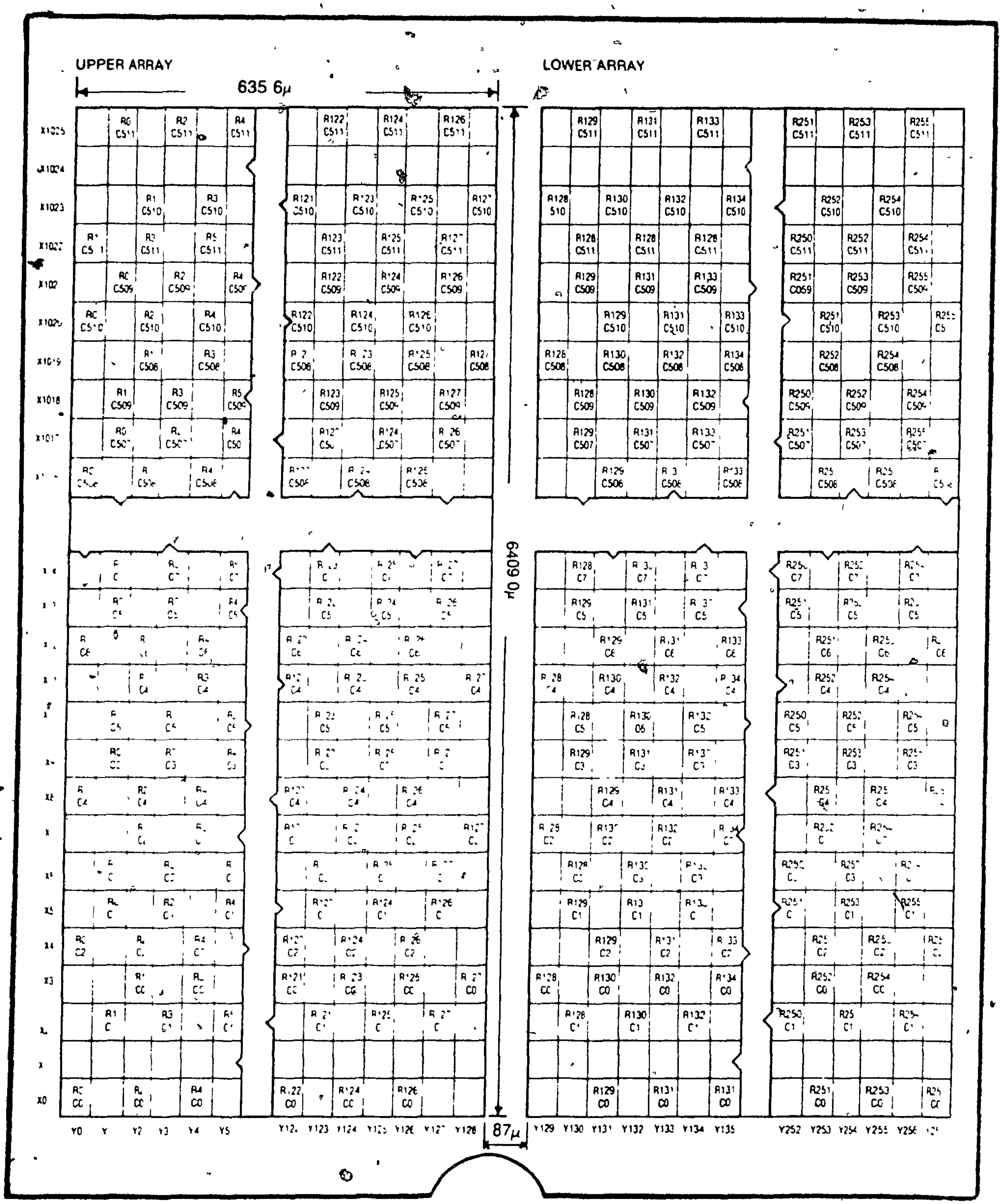


Figure 4.5 IS256 OpticRAM Topological Information. Micron Technologies Inc. [24].

4.3.3 The 63701 Microcomputer

The 63701 microcomputer, a second source CMOS/EPROM, peripheral interface onboard version of the 6801 extended Motorola 6800, is only a part of the interface. It has one 8 bit port for data transfer between the PC host and the 63701 internal memory. This bus is also used to transfer data to the 4-bit microcomputer slices' internal dual port ram (see [24] for further details on the 63701 micro and its interface).

Communication between the PC/AT and the microcomputer is via an 18 byte command and data register set and a status register.

4.4 System Software

IDETIX is an intelligent machine vision subsystem. The MOS (Metal Oxide Semi-conductor) sensor based camera head is connected via RS422 interface to the controller board. IDETIX drivers are supplied as a subroutine library coded in assembly language for efficiency.

4.5 Testing Environment

Several hundred images were digitized and consequently their moment invariants computed in order to establish:

1) The best method of lighting,

2) The amount of light required to produce the best image, since three factors were involved:

a) changing the exposure time
b) changing the f-stop on the lens
c) changing the intensity of the light source

3) The best lens, given a maximum distance between lens and object of 1.5 m.

The combination of factors that produced the best image was then established and the selection is as follows:

Lens : F1.6, 8.5mm (wide-angle)

Light : a combination of front lighting and elimination of spectral reflections using 150 Watt incandescent light-bulb (see the section on lighting considerations)

f-Stop : 8

Exposure time : 500 msec.

Object : a 50mm X 250mm white object on a black background, or vice-versa. This size was chosen solely so that the image could be displayed on a CRT terminal with 200X640 resolution, so that the camera digitization is displayed on the CRT monitor. In a fully automated set-up, the CRT image is not required and larger objects can be digitized and recognized.

OpticRAM Physical Area Used : only rows 300 - 379 and columns 160 - 319 of array E (see Figure 4.4 of the OpticRAM chip) were used to minimize lens distortion, since wide-angle lenses suffer images with considerable edge distortion .

S'-Distance : a distance of 1.25m (less than the maximum of 1.5m) was used between the lens and the object. This distance allowed the object to fill 65% of the 80 X 640 matrix on the CRT monitor, leaving space for the object to be translated and rotated.

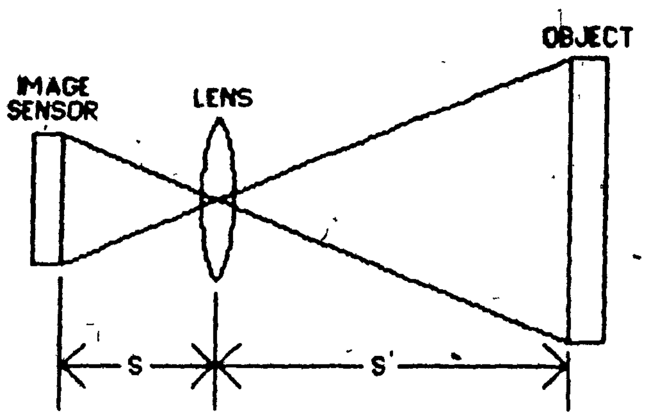
4.6 Lens Selection and Sample Calculations

4.6.1 Lens Selection

The lens supplied with the IDETIX system is an F1.6 16mm lens with adjustable f-stop and focus control. The f-stop controls the amount of light admitted through the lens while the focus control focuses the image on the surface of the OpticRAM. In our particular application a wide angle lens is required for close-up viewing.

The selection of a lens requires the consideration of many parameters such as lighting, edge sharpness of the scene, and distance from the camera to the scene. The lens provides a projection of the scene into the OpticRAM. This means if the lens is not selected properly or is misadjusted, the information that the OpticRAM sees will not adequately represent the scene.

The fcllowing equations represents basic lens optics:



M(magnification) = Image Field of View/opticRam size
= S'/S
= F/(S-F)
= (S'-F)/F

F(Focal length) = S'/(M+1)
= (S*M)/(M+1)
= (S+S')/(M+2+(1/M))
= (M*(S+S'))/(M+1)

S'(Lens to Object Distance) = S*M
= F*(M+1)
=(S*F)/(S-F)

Figure 4.6 Simple Lens Equations.

4.6.2 Sample Calculations

Given the average distance S'=1250mm, the magnification (M) required to project the object on the OpticRAM

$$M = (1250mm-8.5mm)/8.5mm = 146.$$

Accuracy is the degree to which the measurement represents the true value of the quantity being measured. Under ideal conditions, error in accuracy will not exceed the resolution of the measurement system. When measuring the distance between two edges of an image, the accuracy is equivalent to one element per edge when the optical image of the object's edge is sharp.

The resolution is equivalent to the least resolvable element or increment, *i.e.*, one pixel, in this case. The scene resolution, on the other hand, is the pixel size multiplied by the lens magnification (element size is one pixel of 4.64µ X 4.64µ).

R_h = The horizontal resolution of the object
= M(magnification) X Horizontal Size
= $146 \times 4.64 \times 10^{-3}$ = 0.677mm

D_h = *The percentage of the smallest horizontal distance detected*
= *(0.677mm / 250mm) X 100 = 0.271%*

R_v = The vertical resolution of the object
= M(magnification) X Vertical Size
= $146 \times 4.64 \times 10^{-3}$ = 0.677mm

D_v = *The percentage of the smallest vertical distance detected*
= *(0.744mm / 50mm) X 100 = 1.35%*

Therefore the smallest area detected (in percentage) given a magnification (M) of 146 times and an object of 250mm by 50mm is equivalent to the resolution (or accuracy) R.

$$R = 0.271\% \times 1.345\%$$

$$= \mathbf{0.360\%\ resolution\ (or\ accuracy).}$$

In general, Given the magnification (M), and the size of the object (a and b), the resolution (R) could be calculated as follows:

$$R_h = M \times 4.64 \times 10^{-3}$$

$$R_v = M \times 4.64 \times 10^{-3}$$

$$D_h = \frac{R_h}{b} \times 100$$

$$D_v = \frac{R_v}{a} \times 100$$

$$R = D_h \times D_v$$

$$R = \frac{0.464^2 \times M^2}{a \times b}$$

Chapter 5 APPLICATION SOFTWARE

5.1 The Straight-Forward Approach Program

To verify the feasibility of using the moment invariants method and the running time required, implementation programs were written in both Turbo Pascal and Fortran-77, using a Hewlett Packard Vectra.

The FORTRAN-77 program starts by generating (simulating) a digitized image of size 256 X 256. It then calculates the 2-D moments in a recursive loop and ultimately the moment invariants. The program required *much* more than the pre-set limit of 1 second because of the recursive nature of the calculation, in fact for that size of matrix the running time reached one minute (see complete listings in APPENDIX I and time complexity analysis for the time spent).

5.2 The Delta Method Program

At first, the delta method was implemented in Turbo-Pascal to verify the feasibility and the running time. The input was a simulated image of 128 X 512 bits of information. This program took considerably less than one second (see APPENDIX II for a complete listing of the Turbo-Pascal program).

Once satisfactory results had been achieved, an application program, that interacts with the IDETIX system and uses the delta method, was written. This program is listed in APPENDIX III and the following is a detailed description of the program :

Lines 001-690 : setting the IDETIX hardware parameters

Lines 700-720 : subroutine to reset the IDETIX parameters and/or stop the camera

Lines 1010-3240 : calling the camera driver system routines for image digitizing, image enhancement and display of image on CRT

Lines 4000-4100 : service the keyboard for interactive programming "c" will calculate moments and print results "C" will calculate moments and display the results. "L" or "l" will look at memory location APTR(5)+C, where the enhanced image is stored. "D" or "d" will calculate first the dimension of the object in pixels for further enhancing of the edges. Then it will store the seven invariants for later comparisons. This should be done before "c" or "C". "S" or "s" will stop the IDETIX camera. "R" or "r" only these keys will resume camera operation. "P" or "p" will print the image on an Epson printer.
"Q" or "q" will quit the program.

Lines 5300-5313 : subroutine to set the parameters for enhancing.

Lines 6000-6200 : subroutine to calculate the number of white and black pixels.

Lines 40000-40011 : subroutine to print the image on an Epson printer.

Lines 45000-45200 : subroutine CALCULATE 1, this subroutine will calculate the delta δ, X_{st}, Y_{st} (X_i and Y_i in Figure 3.1) and the $\dot{m}_{pq}$'s.

Lines 45210-45230 : subroutine BYTE-LEFT, this will find the left edge of the image.

Lines 45240-45280 : subroutine BYTE-RIGHT, this will find the right edge of the image.

Lines 45290-50460 : subroutine CALCULATE 2, this will calculate the seven invariants that represent the image.

Lines 55000-55070 : subroutine LOOK, this will print or display the image in HEX numbers.

Lines 56000-57000 : the two subroutines that will calculate the size of the object and then store the first set of moment invariants for later comparisons.

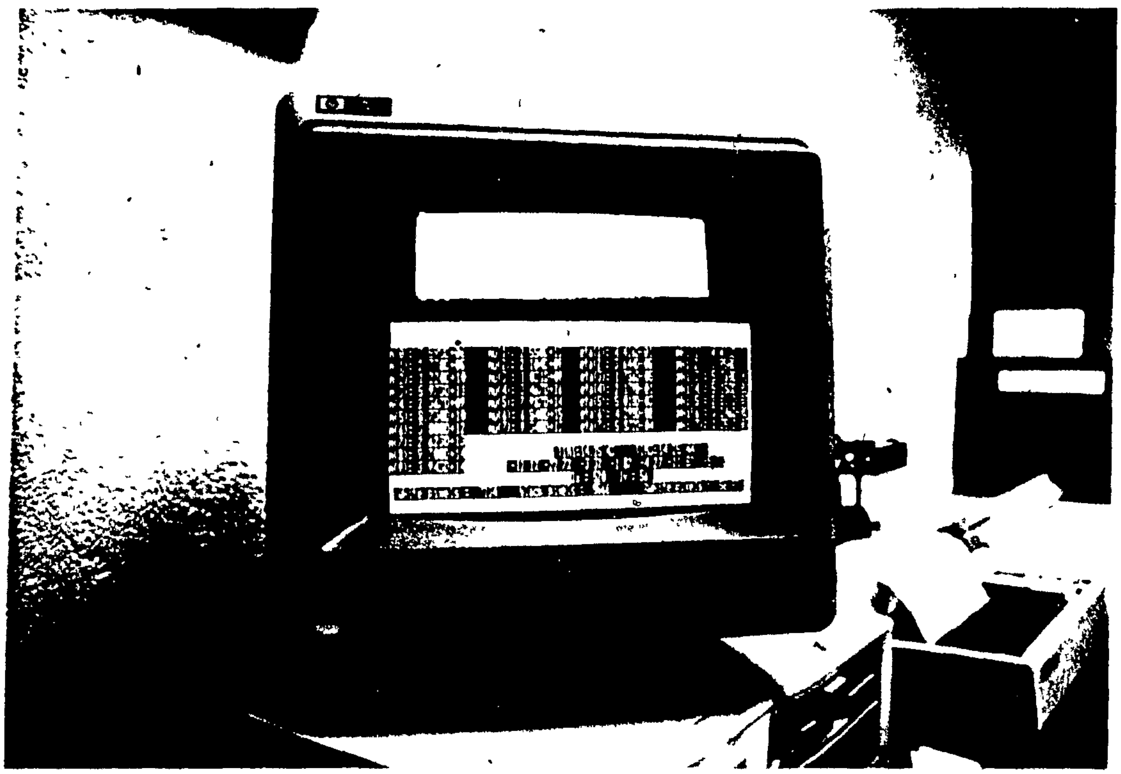


Figure 5.1 The Information Displayed By The Delta Method Interactive Program.

The delta method reduces the amount of recursive calculations needed to compute the moment invariants. Using an optimizing compiler the average running time for a program with a matrix of size 80 X 640 was 0.64 seconds. (see Figure 6.1 to 6.28 for running time).

5.3 Complexity Analysis

5.3.1 Space Complexity

The advantages of the *delta (δ) method* over the straightforward (S) method, can be shown by a comparison of the time complexity (running time). The space complexity (space occupied by the data required for processing) of the two methods is similar.

5.3.2 Time Complexity

To compare *worst* case running time for both the δ *method* and the S method, assume that the image occupies the entire intensity matrix f(i,j).

In the S method a maximum of 10 additions and 20 multiplications is required for each pixel, over the entire image matrix M X N, as each 'on' pixel contributes to the m_{pq}'s (see (1.a)-(1.j)).

In the δ *method* a maximum of N+6 additions and 25 multiplications is required for each *line* of pixels, over the entire image matrix M X N, in order to calculate $m_{pq,i}$ for the corresponding line of pixels i. (see (1.a')-(1.j')).

To calculate the order of time complexity for the intensity matrix f(i,j) of size M X N :

Straightforward method :

# of additions = 10 X M X N (5.1)

$$\text{\# of multiplications} = 20 \times M \times N \tag{5.2}$$

Delta method :

$$\text{\# of additions} = (N + 6) \times M \tag{6.1}$$

$$\text{\# of multiplications} = 25 \times M \tag{6.2}$$

For an 80287-8 co-processor the average number of clock cycles, for a single multiplication, (64 bit real), is 140 cycles, and for a single addition the average is 110 (see [22]).

Combining (5.1) and (5.2), (6.1) and (6.2) gives:

Straightforward method :

$$\text{Average \# of clock cycles} : 3900 \times M \times N \tag{5}$$

Delta method:

$$\text{Average \# of clock cycles} : M \times (110\,N + 4160) \tag{6}$$

The ratio of (5) over (6) for a reasonable large N is given as follows (see Figure 5.2):

$$\frac{(3900 \times N)}{(110 \times N + 4160)} \approx 35$$

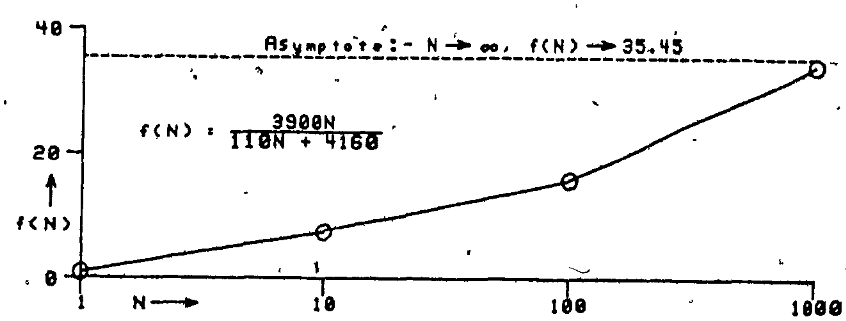


Figure 5.2 The Time Complexity Ratio.

Chapter 6 DISCUSSION AND RESULTS

6.1 Testing Results

In order to verify the resolution (accuracy) of the camera and the sensitivity of the moment invariants, an object was rotated, translated, punctured, truncated and distorted slightly by flawing and adding appendages to the edges.

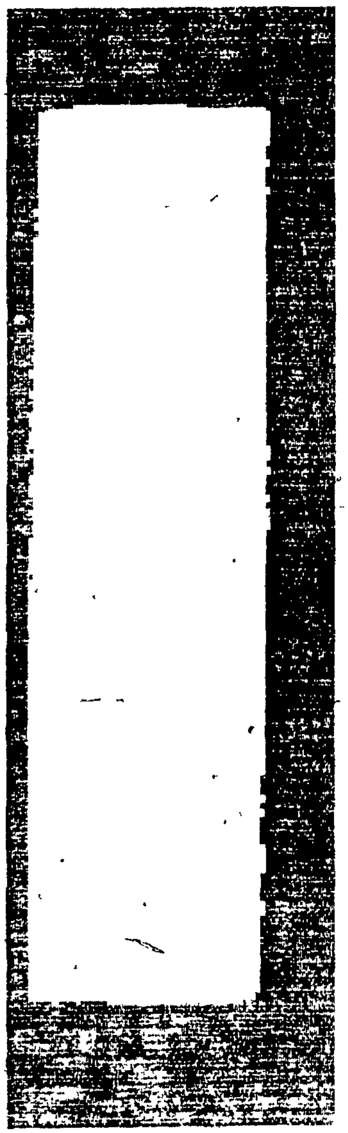
The object shown in Figure 6.1 was digitized and its moment invariants calculated using the delta method. These moments were stored to be compared later by the next set of moments. The image was then rotated and in the same time translated in the X-direction and in the Y-direction (see Figure 6.2-6.10). In the next set of data the object was punctured using an ordinary paper punch with a hole diameter of 7mm (less than 0.40 % of the total area) (Figure 6.11-6.19), and different combinations of rotations and translations were applied.

The object was next distorted slightly by the addition of material, 1.2 % of the total area, in Figure 6.20-6.22 and 0.67 % in Figure 6.23-6.25.
Finally the object was cut diagonally in three different ways to simulate a situation when the robotic picker fails to pick up the piece of fabric properly (see Figure 6.26-6.28). Figure

6.29 summarizes the experimentation results and lists the ϕ's and their percentage deviation from the original image.

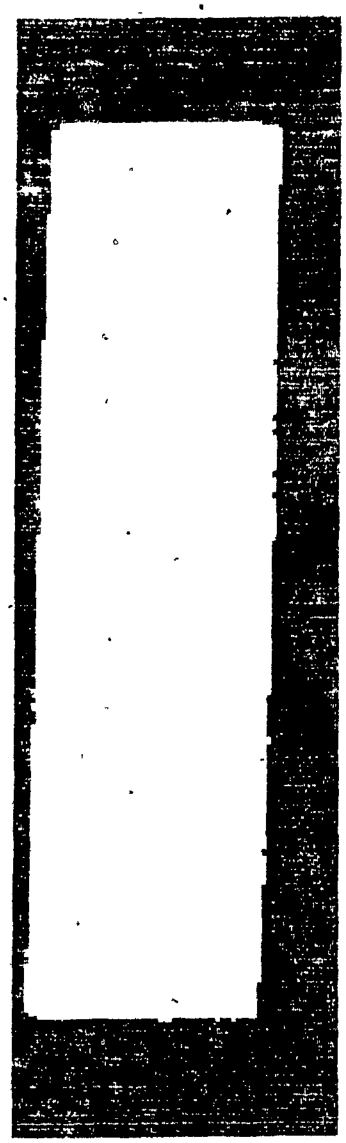
6.1 Testing Results

Figure 6.1: Image # 1.



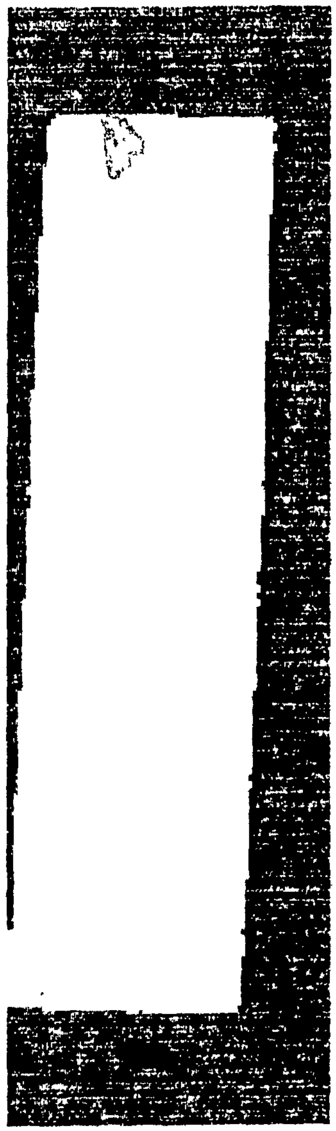
```
M(0)   =  0.833646D+07
M(1)   =  0.277061D+09
M(2)   =  0.114987D+11
M(3)   =  0.534412D+12
M(4)   =  0.297239D+10
M(5)   =  0.991478D+11
M(6)   =  0.413192D+13
M(7)   =  0.448702D+14
M(8)   =  0.151571D+16
M(9)   =  0.617453D+15

MU(1)  =  0.438104D+14
MU(2)  =  0.229063D+10
MU(3)  =  0.360986D+09
MU(4)  =  0.493590D+17
MU(5)  =  0.833931D+15
MU(6)  =  0.897078D+16
MU(7)  =  0.777310D+14

ETA(1) =  0.630397D+00
ETA(2) =  0.329603D-04
ETA(3) =  0.519430D-05
ETA(4) =  0.245987D+00
ETA(5) =  0.415600D-02
ETA(6) =  0.447070D-01
ETA(7) =  0.387383D-03

PHI(1) =  0.630430D+00
PHI(2) =  0.397360D+00
PHI(3) =  0.724158D-01
PHI(4) =  0.646050D-01
PHI(5) =  0.441842D-02
PHI(6) =  0.420008D-01
PHI(7) =  0.199025D-02

TIM1  13:06:45 TIM2            TIM3  13:06:46

% CHANGE IN PHI(1) =    0.0%
% CHANGE IN PHI(2) =    0.0%
% CHANGE IN PHI(3) =    0.0%
% CHANGE IN PHI(4) =    0.0%
% CHANGE IN PHI(5) =    0.0%
% CHANGE IN PHI(6) =    0.0%
% CHANGE IN PHI(7) =    0.0%
```

Figure 6.2: Image # 2.



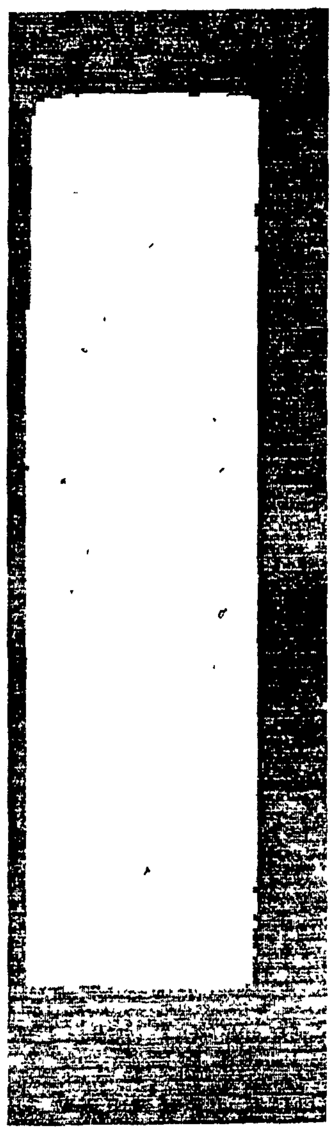
```
M(0)    =  0.864297D+07
M(1)    =  0.277878D+09
M(2)    =  0.114682D+11
M(3)    =  0.531874D+12
M(4)    =  0.308586D+10
M(5)    =  0.100090D+12
M(6)    =  0.416881D+13
M(7)    =  0.460096D+14
M(8)    =  0.153647D+16
M(9)    =  0.643713D+15

MU(1)   =  0.449079D+14
MU(2)   =  0.253424D+10
MU(3)   =  0.876944D+09
MU(4)   =  0.520651D+17
MU(5)   =  0.810218D+15
MU(6)   =  0.905393D+16
MU(7)   =  0.729578D+14

ETA(1)  =  0.601169D+00
ETA(2)  =  0.339252D-04
ETA(3)  =  0.117394D-04
ETA(4)  =  0.237077D+00
ETA(5)  =  0.368930D-02
ETA(6)  =  0.412268D-01
ETA(7)  =  0.332211D-03

PHI(1)  =  0.601203D+00
PHI(2)  =  0.361372D+00
PHI(3)  =  0.662947D-01
PHI(4)  =  0.596954D-01
PHI(5)  =  0.375503D-02
PHI(6)  =  0.415066D-01
PHI(7)  =  0.164448D-02

TIM1 13:08:24 TIM2          TIM3 13:08:25

% CHANGE IN PHI(1) =   -4.6%
% CHANGE IN PHI(2) =   -9.1%
% CHANGE IN PHI(3) =   -8.5%
% CHANGE IN PHI(4) =   -7.6%
% CHANGE IN PHI(5) =  -15.0%
% CHANGE IN PHI(6) =   -1.2%
% CHANGE IN PHI(7) =  -17.4%
```

Figure 6.3: Image # 3.



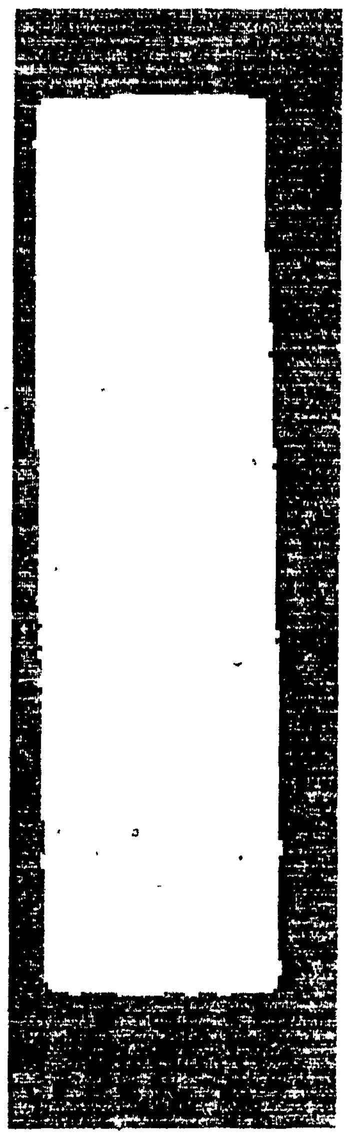
```
M(0)    =  0.909483D+07
M(1)    =  0.278564D+09
M(2)    =  0.114820D+11
M(3)    =  0.533278D+12
M(4)    =  0.326879D+10
M(5)    =  0.102049D+12
M(6)    =  0.428687D+13
M(7)    =  0.487398D+14
M(8)    =  0.160095D+16
M(9)    =  0.688592D+15

MU(1)   =  0.475649D+14
MU(2)   =  0.294991D+10
MU(3)   =  0.192984D+10
MU(4)   =  0.562323D+17
MU(5)   =  0.778942D+15
MU(6)   =  0.924667D+16
MU(7)   =  0.663782D+14

ETA(1)  =  0.575040D+00
ETA(2)  =  0.356632D-04
ETA(3)  =  0.233310D-04
ETA(4)  =  0.225423D+00
ETA(5)  =  0.312262D-02
ETA(6)  =  0.370680D-01
ETA(7)  =  0.266096D-03

PHI(1)  =  0.575075D+00
PHI(2)  =  0.330665D+00
PHI(3)  =  0.589872D-01
PHI(4)  =  0.536272D-01
PHI(5)  =  0.301598D-02
PHI(6)  =  0.422793D-01
PHI(7)  =  0.127110D-02

TIM1 13:10:12 TIM2        TIM3 13:10:13

% CHANGE IN PHI(1) =  -8.8%
% CHANGE IN PHI(2) = -16.8%
% CHANGE IN PHI(3) = -18.5%
% CHANGE IN PHI(4) = -17.0%
% CHANGE IN PHI(5) = -31.7%
% CHANGE IN PHI(6) =   0.7%
% CHANGE IN PHI(7) = -36.1%
```

Figure 6.4: Image # 4.



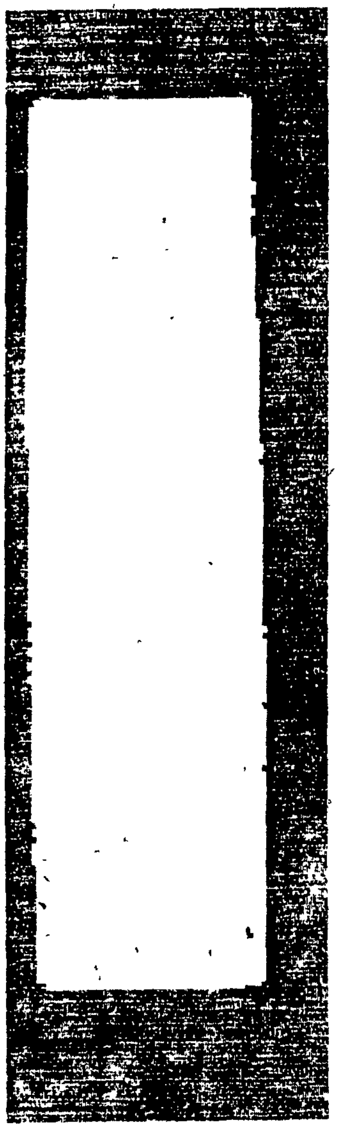
```
M(0)    =  0.815388D+07
M(1)    =  0.269088D+09
M(2)    =  0.110835D+11
M(3)    =  0.511140D+12
M(4)    =  0.292727D+10
M(5)    =  0.965997D+11
M(6)    =  0.397979D+13
M(7)    =  0.470744D+14
M(8)    =  0.155291D+16
M(9)    =  0.607599D+15

MU(1)   =  0.460235D+14
MU(2)   =  0.220331D+10
MU(3)   =  -.374624D+07
MU(4)   =  0.464908D+17
MU(5)   =  0.809760D+15
MU(6)   =  0.880836D+16
MU(7)   =  0.744366D+14

ETA(1)  =  0.692231D+00
ETA(2)  =  0.331395D-04
ETA(3)  =  -.563465D-07
ETA(4)  =  0.244882D+00
ETA(5)  =  0.426526D-02
ETA(6)  =  0.463964D-01
ETA(7)  =  0.392081D-03

PHI(1)  =  0.692264D+00
PHI(2)  =  0.479138D+00
PHI(3)  =  0.731285D-01
PHI(4)  =  0.642633D-01
PHI(5)  =  0.440472D-02
PHI(6)  =  0.414092D-01
PHI(7)  =  0.202627D-02

TIM1 13:16:13 TIM2          TIM3 13:16:14

% CHANGE IN PHI(1) =   9.8%
% CHANGE IN PHI(2) =  20.6%
% CHANGE IN PHI(3) =   1.0%
% CHANGE IN PHI(4) =  -0.5%
% CHANGE IN PHI(5) =  -0.3%
% CHANGE IN PHI(6) =  -1.4%
% CHANGE IN PHI(7) =   1.8%
```

Figure 6.5: Image # 5.



```
M(0)    =  0.848283D+07
M(1)    =  0.296075D+09
M(2)    =  0.128092D+11
M(3)    =  0.619683D+12
M(4)    =  0.306751D+10
M(5)    =  0.106258D+12
M(6)    =  0.459289D+13
M(7)    =  0.498655D+14
M(8)    =  0.169602D+16
M(9)    =  0.642288D+15

MU(1)   =  0.487562D+14
MU(2)   =  0.247533D+10
MU(3)   =  -.807415D+09
MU(4)   =  0.492332D+17
MU(5)   =  0.949184D+15
MU(6)   =  0.979006D+16
MU(7)   =  0.916132D+14

ETA(1)  =  0.677561D+00
ETA(2)  =  0.343994D-04
ETA(3)  =  -.112206D-04
ETA(4)  =  0.234913D+00
ETA(5)  =  0.452896D-02
ETA(6)  =  0.467125D-01
ETA(7)  =  0.437126D-03

PHI(1)  =  0.677596D+00
PHI(2)  =  0.459051D+00
PHI(3)  =  0.685013D-01
PHI(4)  =  0.595554D-01
PHI(5)  =  0.380313D-02
PHI(6)  =  0.290361D-01
PHI(7)  =  0.179994D-02

TIM1  13:12:40 TIM2          TIM3  13:12:41

% CHANGE IN PHI(1) =    7.5%
% CHANGE IN PHI(2) =   15.5%
% CHANGE IN PHI(3) =   -5.4%
% CHANGE IN PHI(4) =   -7.8%
% CHANGE IN PHI(5) =  -13.9%
% CHANGE IN PHI(6) =  -30.9%
% CHANGE IN PHI(7) =   -9.6%
```

Figure 6.6: Image # 6.

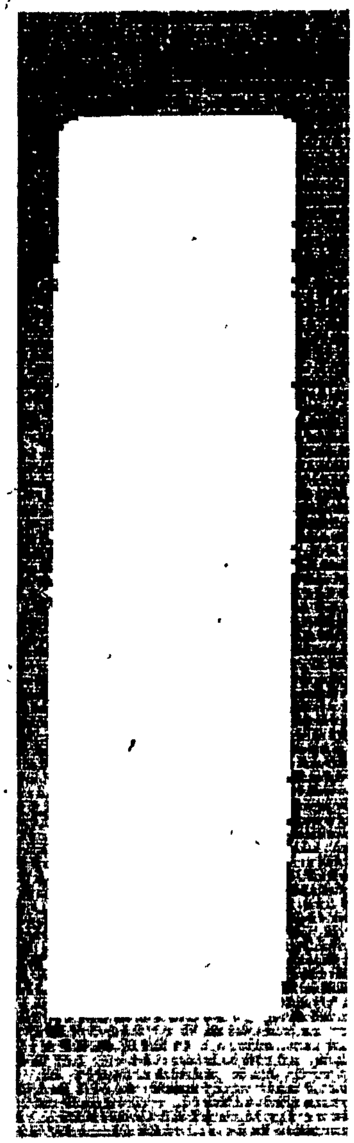


M(0)	=	0.833748D+07
M(1)	=	0.287502D+09
M(2)	=	0.122537D+11
M(3)	=	0.583772D+12
M(4)	=	0.300533D+10
M(5)	=	0.102974D+12
M(6)	=	0.438475D+13
M(7)	=	0.483946D+14
M(8)	=	0.162356D+16
M(9)	=	0.627222D+15
MU(1)	=	0.473113D+14
MU(2)	=	0.233971D+10
MU(3)	=	-.659462D+09
MU(4)	=	0.482587D+17
MU(5)	=	0.907705D+15
MU(6)	=	0.944355D+16
MU(7)	=	0.868335D+14
ETA(1)	=	0.680605D+00
ETA(2)	=	0.336584D-04
ETA(3)	=	-.948681D-05
ETA(4)	=	0.240430D+00
ETA(5)	=	0.452228D-02
ETA(6)	=	0.470487D-01
ETA(7)	=	0.432614D-03
PHI(1)	=	0.680639D+00
PHI(2)	=	0.463184D+00
PHI(3)	=	0.712672D-01
PHI(4)	=	0.622561D-01
PHI(5)	=	0.414608D-02
PHI(6)	=	0.320700D-01
PHI(7)	=	0.194587D-02

TIM1 13:14:16 TIM2 TIM3 13:14:17

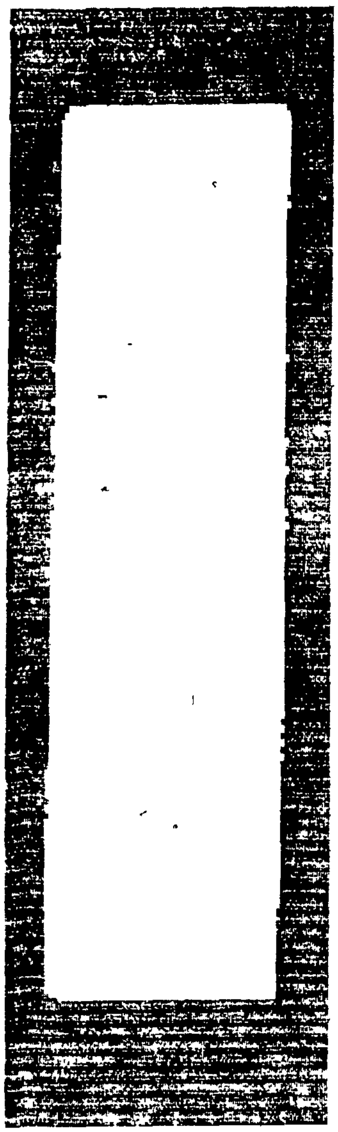
% CHANGE IN PHI(1)	=	8.0%
% CHANGE IN PHI(2)	=	16.6%
% CHANGE IN PHI(3)	=	-1.6%
% CHANGE IN PHI(4)	=	-3.6%
% CHANGE IN PHI(5)	=	-6.2%
% CHANGE IN PHI(6)	=	-23.6%
% CHANGE IN PHI(7)	=	-2.2%

Figure 6.7: Image # 7.



```
M(0)    =  0.851445D+07
M(1)    =  0.304086D+09
M(2)    =  0.132779D+11
M(3)    =  0.646946D+12
M(4)    =  0.302679D+10
M(5)    =  0.108537D+12
M(6)    =  0.476132D+13
M(7)    =  0.446650D+14
M(8)    =  0.162863D+16
M(9)    =  0.628283D+15

MU(1)   =  0.435890D+14
MU(2)   =  0.241771D+10
MU(3)   =  0.437925D+09
MU(4)   =  0.509155D+17
MU(5)   =  0.980623D+15
MU(6)   =  0.979390D+16
MU(7)   =  0.985172D+14

ETA(1)  =  0.601262D+00
ETA(2)  =  0.333497D-04
ETA(3)  =  0.604069D-05
ETA(4)  =  0.240690D+00
ETA(5)  =  0.463565D-02
ETA(6)  =  0.462982D-01
ETA(7)  =  0.465715D-03

PHI(1)  =  0.601295D+00
PHI(2)  =  0.361478D+00
PHI(3)  =  0.705932D-01
PHI(4)  =  0.623717D-01
PHI(5)  =  0.413808D-02
PHI(6)  =  0.394117D-01
PHI(7)  =  0.193320D-02

TIM1 13:30:50 TIM2           TIM3 13:30:50

        % CHANGE IN PHI(1) =   -4.6%
        % CHANGE IN PHI(2) =   -9.0%
        % CHANGE IN PHI(3) =   -2.5%
        % CHANGE IN PHI(4) =   -3.5%
        % CHANGE IN PHI(5) =   -6.3%
        % CHANGE IN PHI(6) =   -6.2%
        % CHANGE IN PHI(7) =   -2.9%
```

Figure 6.8: Image # 8.

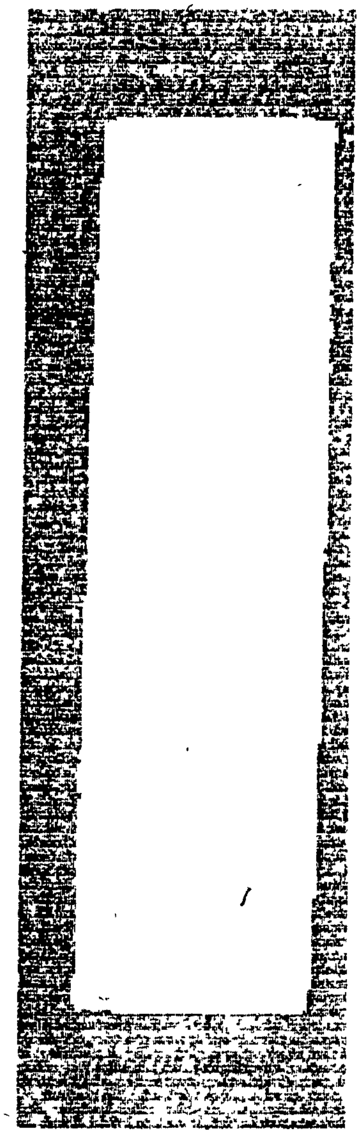


M(0)	=	0.842724D+07
M(1)	=	0.325409D+09
M(2)	=	0.149463D+11
M(3)	=	0.761131D+12
M(4)	=	0.301762D+10
M(5)	=	0.117114D+12
M(6)	=	0.540942D+13
M(7)	=	0.467289D+14
M(8)	=	0.185007D+16
M(9)	=	0.627989D+15
MU(1)	=	0.456484D+14
MU(2)	=	0.238100D+10
MU(3)	=	0.591505D+09
MU(4)	=	0.494822D+17
MU(5)	=	0.114286D+16
MU(6)	=	0.106432D+17
MU(7)	=	0.123240D+15
ETA(1)	=	0.642769D+00
ETA(2)	=	0.335265D-04
ETA(3)	=	0.832891D-05
ETA(4)	=	0.240013D+00
ETA(5)	=	0.554343D-02
ETA(6)	=	0.516250D-01
ETA(7)	=	0.597777D-03
PHI(1)	=	0.642802D+00
PHI(2)	=	0.413113D+00
PHI(3)	=	0.737014D-01
PHI(4)	=	0.630253D-01
PHI(5)	=	0.429421D-02
PHI(6)	=	0.440025D-01
PHI(7)	=	0.212354D-02

TIM1 13:18:05 TIM2 TIM3 13:18:06

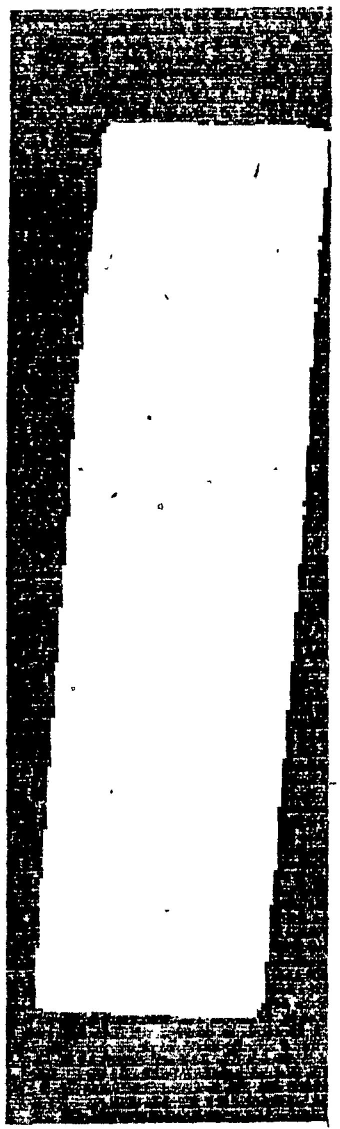
% CHANGE IN PHI(1)	=	2.0%
% CHANGE IN PHI(2)	=	4.0%
% CHANGE IN PHI(3)	=	1.8%
% CHANGE IN PHI(4)	=	-2.4%
% CHANGE IN PHI(5)	=	-2.8%
% CHANGE IN PHI(6)	=	4.8%
% CHANGE IN PHI(7)	=	6.7%

Figure 6.9: Image # 9.



```
M(0)    =  0.873273D+07
M(1)    =  0.370207D+09
M(2)    =  0.182973D+11
M(3)    =  0.996703D+12
M(4)    =  0.311923D+10
M(5)    =  0.133349D+12
M(6)    =  0.665350D+13
M(7)    =  0.460231D+14
M(8)    =  0.201860D+16
M(9)    =  0.652188D+15

MU(1)   =  0.449089D+14
MU(2)   =  0.260305D+10
MU(3)   =  0.111525D+10
MU(4)   =  0.532139D+17
MU(5)   =  0.142390D+16
MU(6)   =  0.120638D+17
MU(7)   =  0.168994D+15

ETA(1)  =  0.588888D+00
ETA(2)  =  0.341337D-04
ETA(3)  =  0.146241D-04
ETA(4)  =  0.236129D+00
ETA(5)  =  0.631835D-02
ETA(6)  =  0.535313D-01
ETA(7)  =  0.749885D-03

PHI(1)  =  0.588922D+00
PHI(2)  =  0.346762D+00
PHI(3)  =  0.727148D-01
PHI(4)  =  0.617273D-01
PHI(5)  =  0.413407D-02
PHI(6)  =  0.454913D-01
PHI(7)  =  0.210677D-02

TIM1 13:23:17 TIM2          TIM3 13:23:18

% CHANGE IN PHI(1) =  -6.6%
% CHANGE IN PHI(2) = -12.7%
% CHANGE IN PHI(3) =   0.4%
% CHANGE IN PHI(4) =  -4.5%
% CHANGE IN PHI(5) =  -6.4%
% CHANGE IN PHI(6) =   8.3%
% CHANGE IN PHI(7) =   5.9%
```

Figure 6.10: Image # 10.

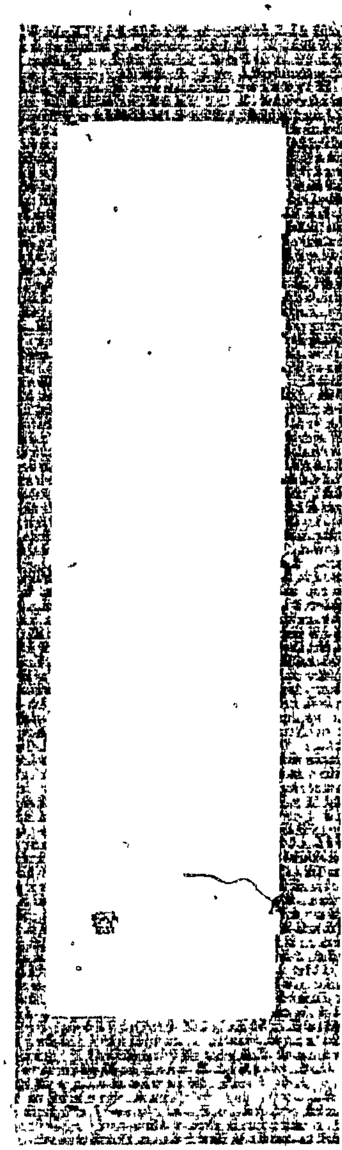


M(0)	=	0.956403D+07
M(1)	=	0.374778D+09
M(2)	=	0.181607D+11
M(3)	=	0.987661D+12
M(4)	=	0.347753D+10
M(5)	=	0.140153D+12
M(6)	=	0.699222D+13
M(7)	=	0.527671D+14
M(8)	=	0.228861D+16
M(9)	=	0.742601D+15
MU(1)	=	0.515027D+14
MU(2)	=	0.347454D+10
MU(3)	=	0.388194D+10
MU(4)	=	0.608816D+17
MU(5)	=	0.135643D+16
MU(6)	=	0.128034D+17
MU(7)	=	0.146179D+15
ETA(1)	=	0.563051D+00
ETA(2)	=	0.379853D-04
ETA(3)	=	0.424392D-04
ETA(4)	=	0.215221D+00
ETA(5)	=	0.479508D-02
ETA(6)	=	0.452611D-01
ETA(7)	=	0.516753D-03
PHI(1)	=	0.563089D+00
PHI(2)	=	0.317102D+00
PHI(3)	=	0.586319D-01
PHI(4)	=	0.505025D-01
PHI(5)	=	0.274742D-02
PHI(6)	=	0.540866D-01
PHI(7)	=	0.134439D-02

TIM1 13:25:49 TIM2 TIM3 13:25:50

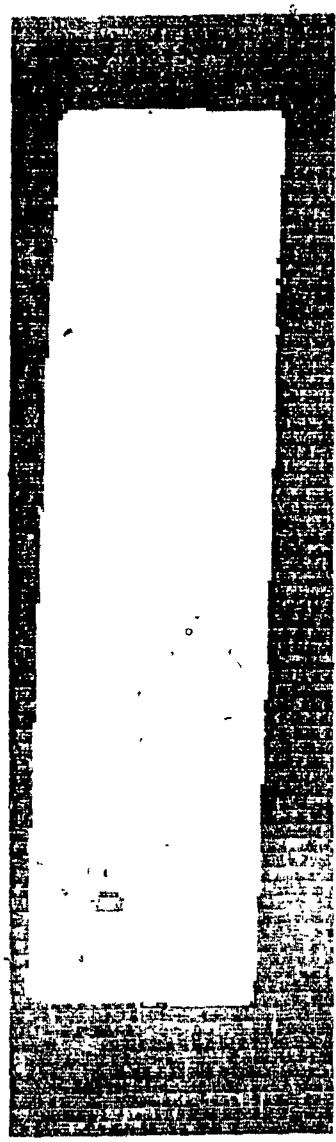
% CHANGE IN PHI(1) = -10.7%
% CHANGE IN PHI(2) = -20.2%
% CHANGE IN PHI(3) = -19.0%
% CHANGE IN PHI(4) = -21.8%
% CHANGE IN PHI(5) = -37.8%
% CHANGE IN PHI(6) = 28.8%
% CHANGE IN PHI(7) = -32.5%

Figure 6.11: Image # 11.



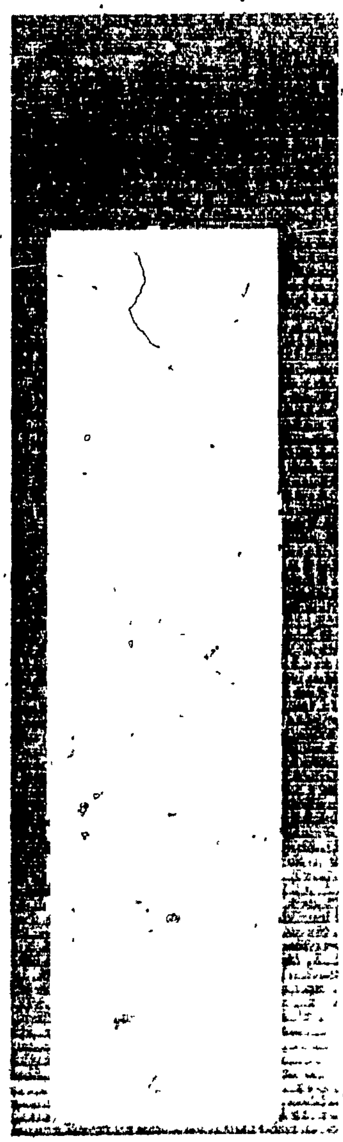
```
M(0)    =  0.649893D+07
M(1)    =  0.249755D+09
M(2)    =  0.112905D+11
M(3)    =  0.556833D+12
M(4)    =  0.213193D+10
M(5)    =  0.834418D+11
M(6)    =  0.382262D+13
M(7)    =  0.321890D+14
M(8)    =  0.128344D+16
M(9)    =  0.376094D+15

MU(1)   =  0.314897D+14
MU(2)   =  0.169238D+10
MU(3)   =  0.151132D+10
MU(4)   =  0.274302D+17
MU(5)   =  0.799747D+15
MU(6)   =  0.687213D+16
MU(7)   =  0.936823D+14

ETA(1) =  0.745563D+00
ETA(2) =  0.400695D-04
ETA(3) =  0.357827D-04
ETA(4) =  0.254756D+00
ETA(5) =  0.742759D-02
ETA(6) =  0.638244D-01
ETA(7) =  0.870068D-03

PHI(1) =  0.745603D+00
PHI(2) =  0.555888D+00
PHI(3) =  0.903735D-01
PHI(4) =  0.729258D-01
PHI(5) =  0.591578D-02
PHI(6) =  0.879038D-01
PHI(7) =  0.308758D-02

TIM1 13:50:39 TIM2           TIM3 13:50:40

      % CHANGE IN PHI(1) =  12.5%
      % CHANGE IN PHI(2) =  26.5%
      % CHANGE IN PHI(3) =  40.3%
      % CHANGE IN PHI(4) =  36.9%
      % CHANGE IN PHI(5) =  89.7%
      % CHANGE IN PHI(6) =  74.6%
      % CHANGE IN PHI(7) =  94.1%
```

Figure 6.12: Image # 12.



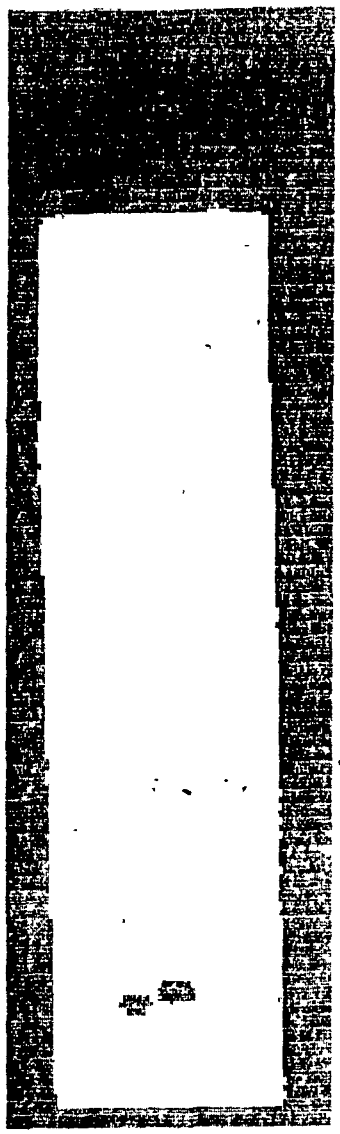
```
M(0)    =  0.633624D+07
M(1)    =  0.239775D+09
M(2)    =  0.108865D+11
M(3)    =  0.539765D+12
M(4)    =  0.210671D+10
M(5)    =  0.835420D+11
M(6)    =  0.390951D+13
M(7)    =  0.340991D+14
M(8)    =  0.141575D+16
M(9)    =  0.383525D+15

MU(1)   =  0.333986D+14
MU(2)   =  0.181297D+10
MU(3)   =  0.382017D+10
MU(4)   =  0.259910D+17
MU(5)   =  0.766273D+15
MU(6)   =  0.685546D+16
MU(7)   =  0.872037D+14

ETA(1)  =  0.831889D+00
ETA(2)  =  0.451574D-04
ETA(3)  =  0.951524D-04
ETA(4)  =  0.257184D+00
ETA(5)  =  0.758236D-02
ETA(6)  =  0.678355D-01
ETA(7)  =  0.862891D-03

PHI(1)  =  0.831934D+00
PHI(2)  =  0.692558D+00
PHI(3)  =  0.960252D-01
PHI(4)  =  0.748207D-01
PHI(5)  =  0.633443D-02
PHI(6)  =  0.167813D+00
PHI(7)  =  0.331059D-02

TIM1 13:54:14 TIM2                TIM3 13:54:14

% CHANGE IN PHI(1) =  25.5%
% CHANGE IN PHI(2) =  57.6%
% CHANGE IN PHI(3) =  49.1%
% CHANGE IN PHI(4) =  40.4%
% CHANGE IN PHI(5) = 103.1%
% CHANGE IN PHI(6) = 233.2%
% CHANGE IN PHI(7) = 108.2%
```

Figure 6.13: Image # 13.



```
M(0)    =  0.637755D+07
M(1)    =  0.247801D+09
M(2)    =  0.114341D+11
M(3)    =  0.577665D+12
M(4)    =  0.168787D+10
M(5)    =  0.670040D+11
M(6)    =  0.315369D+13
M(7)    =  0.331283D+13
M(8)    =  0.131809D+15
M(9)    =  0.242606D+15

MU(1)   =  0.286612D+13
MU(2)   =  0.180576D+10
MU(3)   =  0.142132D+10
MU(4)   =  0.278779D+17
MU(5)   =  0.647268D+15
MU(6)   =  0.441101D+16
MU(7)   =  0.950179D+14

ETA(1)  =  0.704672D-01
ETA(2)  =  0.443969D-04
ETA(3)  =  0.349450D-04
ETA(4)  =  0.271410D+00
ETA(5)  =  0.630158D-02
ETA(6)  =  0.429441D-01
ETA(7)  =  0.925061D-03

PHI(1)  =  0.705116D-01
PHI(2)  =  0.503940D-02
PHI(3)  =  0.801192D-01
PHI(4)  =  0.790483D-01
PHI(5)  =  0.629081D-02
PHI(6)  =  0.331966D-01
PHI(7)  =  0.271543D-02

TIM1 13:55:51 TIM2          TIM3 13:55:51

% CHANGE IN PHI(1) = -89.4%
% CHANGE IN PHI(2) = -98.9%
% CHANGE IN PHI(3) =  24.4%
% CHANGE IN PHI(4) =  48.4%
% CHANGE IN PHI(5) = 101.7%
% CHANGE IN PHI(6) = -34.1%
% CHANGE IN PHI(7) =  70.7%
```

Figure 6.14: image # 14.



M(0) = 0.563091D+07
M(1) = 0.231572D+09
M(2) = 0.116038D+11
M(3) = 0.638646D+12
M(4) = 0.154140D+10
M(5) = 0.651188D+11
M(6) = 0.335525D+13
M(7) = 0.605669D+13
M(8) = 0.227879D+15
M(9) = 0.241850D+15

MU(1) = 0.563475D+13
MU(2) = 0.208032D+10
MU(3) = 0.172841D+10
MU(4) = 0.248366D+17
MU(5) = 0.662200D+15
MU(6) = 0.438537D+16
MU(7) = 0.994702D+14

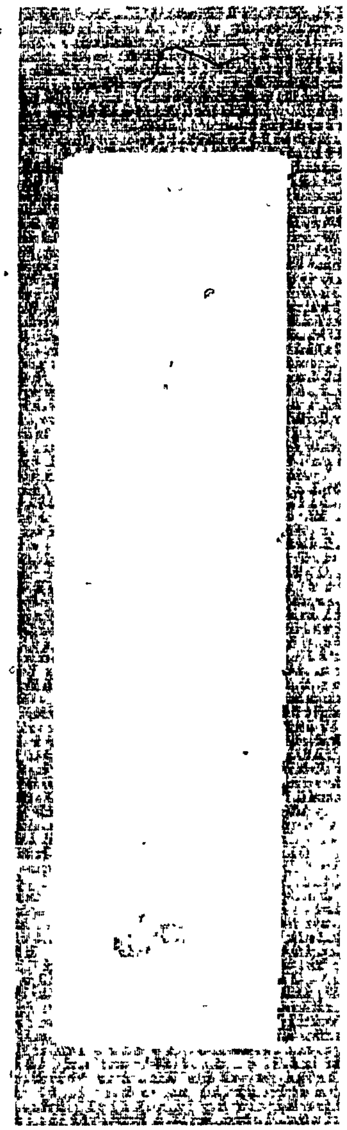
ETA(1) = 0.177712D+00
ETA(2) = 0.656104D-04
ETA(3) = 0.545116D-04
ETA(4) = 0.330100D+00
ETA(5) = 0.880121D-02
ETA(6) = 0.582854D-01
ETA(7) = 0.132205D-02

PHI(1) = 0.177778D+00
PHI(2) = 0.317531D-01
PHI(3) = 0.122345D+00
PHI(4) = 0.118407D+00
PHI(5) = 0.142514D-01
PHI(6) = 0.919397D-01
PHI(7) = 0.663760D-02

TIM1 13:57:25 TIM2 TIM3 13:57:26

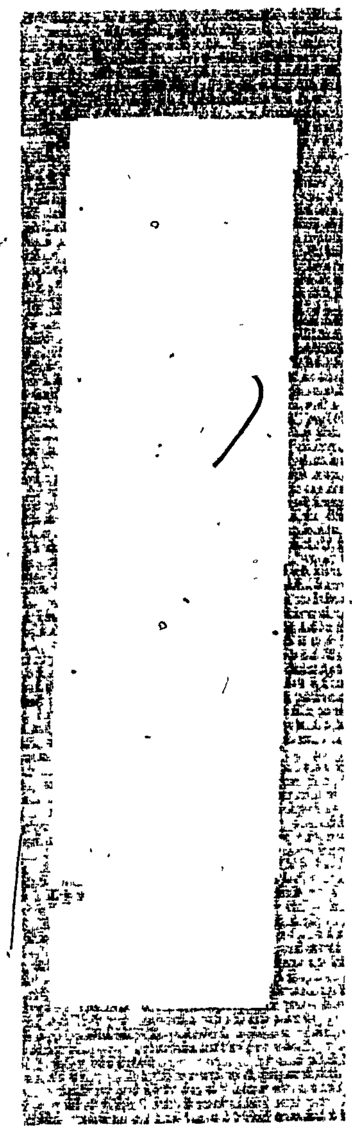
% CHANGE IN PHI(1) = -73.2%
% CHANGE IN PHI(2) = -92.8%
% CHANGE IN PHI(3) = 89.9%
% CHANGE IN PHI(4) = 122.3%
% CHANGE IN PHI(5) = 356.9%
% CHANGE IN PHI(6) = 82.6%
% CHANGE IN PHI(7) = 317.3%

Figure 6.15: Image # 15.



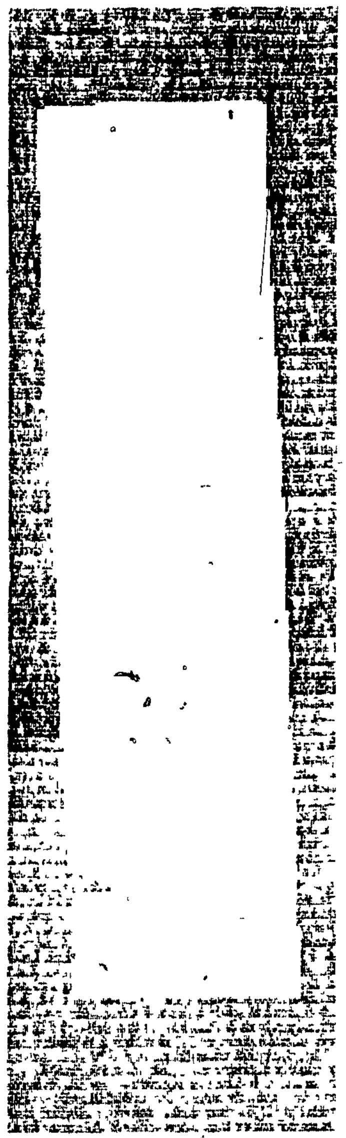
```
M(0)    =  0.517344D+07
M(1)    =  0.200774D+09
M(2)    =  0.962151D+10
M(3)    =  0.506182D+12
M(4)    =  0.153516D+10
M(5)    =  0.613545D+11
M(6)    =  0.301835D+13
M(7)    =  0.163005D+14
M(8)    =  0.671501D+15
M(9)    =  0.252739D+15

MU(1)   =  0.158450D+14
MU(2)   =  0.182977D+10
MU(3)   =  0.177694D+10
MU(4)   =  0.203473D+17
MU(5)   =  0.587303D+15
MU(6)   =  0.452831D+16
MU(7)   =  0.767968D+14

ETA(1)  =  0.592014D+00
ETA(2)  =  0.683656D-04
ETA(3)  =  0.663918D-04
ETA(4)  =  0.334239D+00
ETA(5)  =  0.964748D-02
ETA(6)  =  0.743855D-01
ETA(7)  =  0.126152D-02

PHI(1)  =  0.592083D+00
PHI(2)  =  0.350689D+00
PHI(3)  =  0.142444D+00
PHI(4)  =  0.123981D+00
PHI(5)  =  0.164724D-01
PHI(6)  =  0.179803D+00
PHI(7)  =  0.844481D-02

TIM1 13:59:29 TIM2             TIM3 13:59:29

     % CHANGE  IN PHI(1)  = -10.7%
     % CHANGE  IN PHI(2)  = -20.2%
     % CHANGE  IN PHI(3)  = 121.1%
     % CHANGE  IN PHI(4)  = 132.7%
     % CHANGE  IN PHI(5)  = 428.1%
     % CHANGE  IN PHI(6)  = 257.1%
     % CHANGE  IN PHI(7)  = 431.0%
```

Figure 6.16: Image # 16.



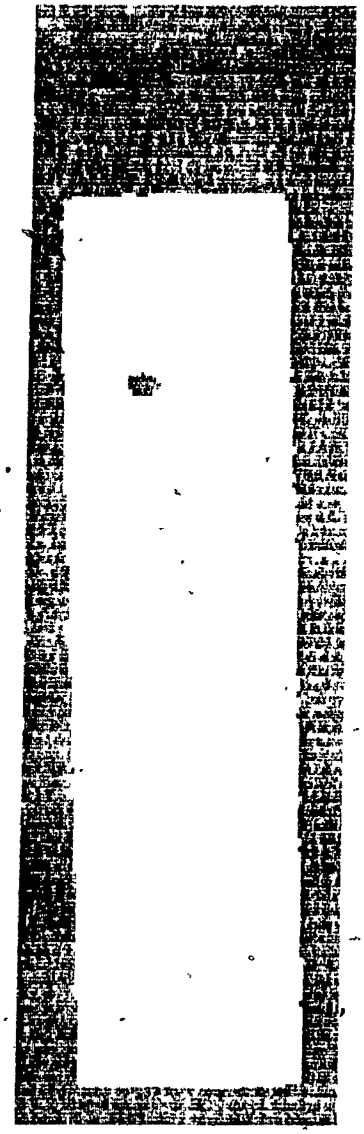
```
M(0)    =  0.646425D+07
M(1)    =  0.259026D+09
M(2)    =  0.117329D+11
M(3)    =  0.577463D+12
M(4)    =  0.211504D+10
M(5)    =  0.872177D+11
M(6)    =  0.403826D+13
M(7)    =  0.311473D+14
M(8)    =  0.132942D+16
M(9)    =  0.376648D+15

MU(1)   =  0.304553D+14
MU(2)   =  0.135363D+10
MU(3)   =  0.246679D+10
MU(4)   =  0.277677D+17
MU(5)   =  0.862589D+15
MU(6)   =  0.712306D+16
MU(7)   =  0.105639D+15

ETA(1)  =  0.728830D+00
ETA(2)  =  0.323940D-04
ETA(3)  =  0.590332D-04
ETA(4)  =  0.261363D+00
ETA(5)  =  0.811911D-02
ETA(6)  =  0.670458D-01
ETA(7)  =  0.994322D-03

PHI(1)  =  0.728863D+00
PHI(2)  =  0.531375D+00
PHI(3)  =  0.962291D-01
PHI(4)  =  0.772503D-01
PHI(5)  =  0.665494D-02
PHI(6)  =  0.120489D+00
PHI(7)  =  0.350670D-02

TIM1 14:03:34 TIM2           TIM3 14:03:35

% CHANGE IN PHI(1) =    9.9%
% CHANGE IN PHI(2) =   20.9%
% CHANGE IN PHI(3) =   49.4%
% CHANGE IN PHI(4) =   45.0%
% CHANGE IN PHI(5) =  113.4%
% CHANGE IN PHI(6) =  139.3%
% CHANGE IN PHI(7) =  120.5%
```

Figure 6.17: Image # 17.



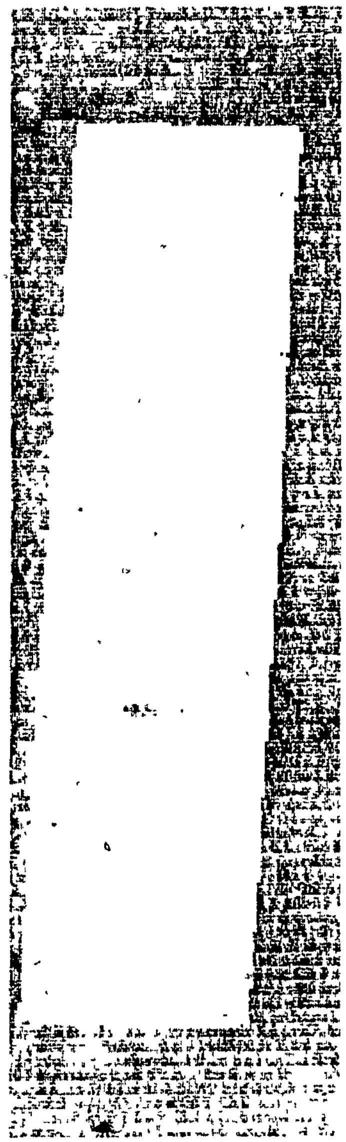
```
M(0)    =  0.660909D+07
M(1)    =  0.301859D+09
M(2)    =  0.153848D+11
M(3)    =  0.840464D+12
M(4)    =  0.223815D+10
M(5)    =  0.102723D+12
M(6)    =  0.528305D+13
M(7)    =  0.350386D+14
M(8)    =  0.158892D+16
M(9)    =  0.416611D+15

MU(1)   =  0.342806D+14
MU(2)   =  0.159792D+10
MU(3)   =  0.499074D+09
MU(4)   =  0.305285D+17
MU(5)   =  0.118593D+16
MU(6)   =  0.878118D+16
MU(7)   =  0.159934D+15

ETA(1)  =  0.784811D+00
ETA(2)  =  0.365823D-04
ETA(3)  =  0.114257D-04
ETA(4)  =  0.271864D+00
ETA(5)  =  0.105610D-01
ETA(6)  =  0.781986D-01
ETA(7)  =  0.142425D-02

PHI(1)  =  0.784848D+00
PHI(2)  =  0.615880D+00
PHI(3)  =  0.112056D+00
PHI(4)  =  0.861037D-01
PHI(5)  =  0.844542D-02
PHI(6)  =  0.744598D-01
PHI(7)  =  0.456148D-02

TIM1 14:05:16 TIM2             TIM3 14:05:17

% CHANGE IN PHI(1) =  18.4%
% CHANGE IN PHI(2) =  40.1%
% CHANGE IN PHI(3) =  73.9%
% CHANGE IN PHI(4) =  61.6%
% CHANGE IN PHI(5) = 170.8%
% CHANGE IN PHI(6) =  47.9%
% CHANGE IN PHI(7) = 186.8%
```

Figure 6.18: Image # 18:



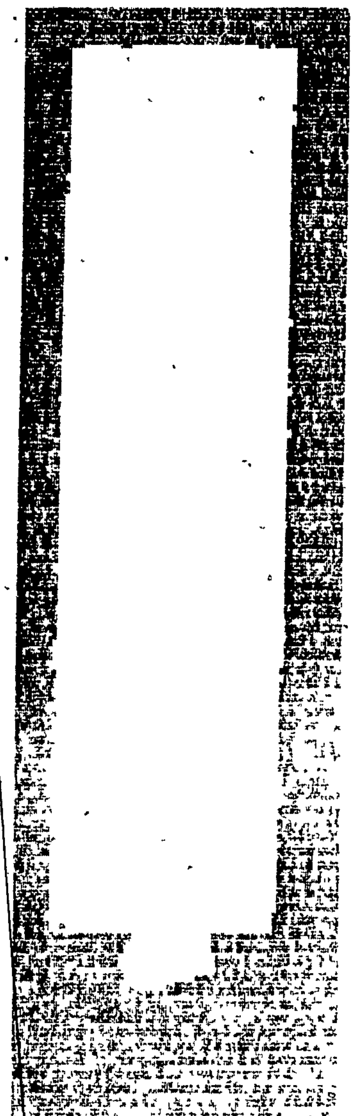
```
M(0)    =  0.747354D+07
M(1)    =  0.308558D+09
M(2)    =  0.148755D+11
M(3)    =  0.785790D+12
M(4)    =  0.214614D+10
M(5)    =  0.903168D+11
M(6)    =  0.445633D+13
M(7)    =  0.138273D+14
M(8)    =  0.528952D+15
M(9)    =  0.333644D+15

MU(1)   =  0.132110D+14
MU(2)   =  0.213612D+10
MU(3)   =  0.170981D+10
MU(4)   =  0.337279D+17
MU(5)   =  0.929250D+15
MU(6)   =  0.642007D+16
MU(7)   =  0.133591D+15

ETA(1)  =  0.236527D+00
ETA(2)  =  0.382448D-04
ETA(3)  =  0.306122D-04
ETA(4)  =  0.220889D+00
ETA(5)  =  0.608579D-02
ETA(6)  =  0.420459D-01
ETA(7)  =  0.874905D-03

PHI(1)  =  0.236566D+00
PHI(2)  =  0.559886D-01
PHI(3)  =  0.567503D-01
PHI(4)  =  0.533597D-01
PHI(5)  =  0.293625D-02
PHI(6)  =  0.312919D-01
PHI(7)  =  0.142152D-02

TIM1 14:07:15 TIM2             TIM3 14:07:15

% CHANGE  IN PHI(1)  = -64.3%
% CHANGE  IN PHI(2)  = -87.3%
% CHANGE  IN PHI(3)  = -11.9%
% CHANGE  IN PHI(4)  =   0.2%
% CHANGE  IN PHI(5)  =  -5.9%
% CHANGE  IN PHI(6)  = -37.9%
% CHANGE  IN PHI(7)  = -10.6%
```

Figure 6.19: Image # 19.



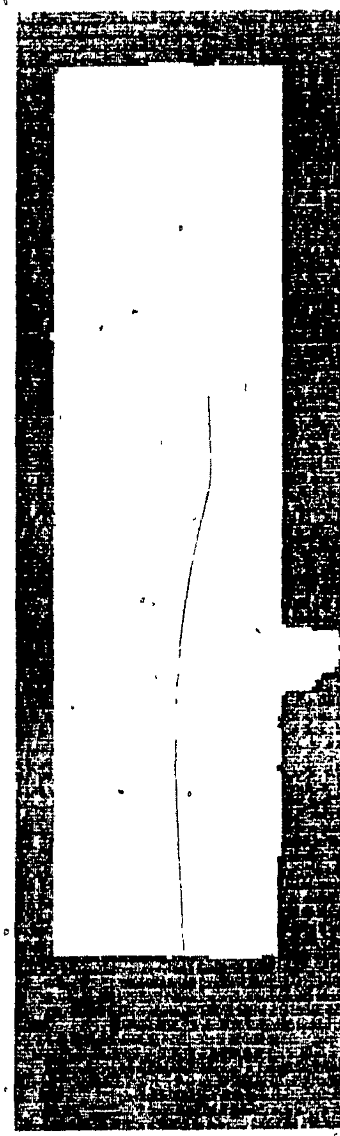
```
M(0)    =  0.681819D+07
M(1)    =  0.252308D+09
M(2)    =  0.114391D+11
M(3)    =  0.574399D+12
M(4)    =  0.218894D+10
M(5)    =  0.871767D+11
M(6)    =  0.415812D+13
M(7)    =  0.333622D+14
M(8)    =  0.144143D+16
M(9)    =  0.390591D+15

MU(1)   =  0.326595D+14
MU(2)   =  0.210234D+10
MU(3)   =  0.617461D+10
MU(4)   =  0.260152D+17
MU(5)   =  0.761393D+15
MU(6)   =  0.680823D+16
MU(7)   =  0.877541D+14

ETA(1) =  0.702540D+00
ETA(2) =  0.452237D-04
ETA(3) =  0.132822D-03
ETA(4) =  0.214316D+00
ETA(5) =  0.627243D-02
ETA(6) =  0.560869D-01
ETA(7) =  0.722927D-03

PHI(1) =  0.702585D+00
PHI(2) =  0.494655D+00
PHI(3) =  0.662887D-01
PHI(4) =  0.518867D-01
PHI(5) =  0.303956D-02
PHI(6) =  0.140999D+00
PHI(7) =  0.158722D-02

TIM1 14:01:36 TIM2            TIM3 14:01:37

% CHANGE IN PHI(1) =    6.0%
% CHANGE IN PHI(2) =   12.6%
% CHANGE IN PHI(3) =    2.9%
% CHANGE IN PHI(4) =   -2.6%
% CHANGE IN PHI(5) =   -2.6%
% CHANGE IN PHI(6) =  180.0%
% CHANGE IN PHI(7) =   -0.2%
```

Figure 6.20: Image # 20.



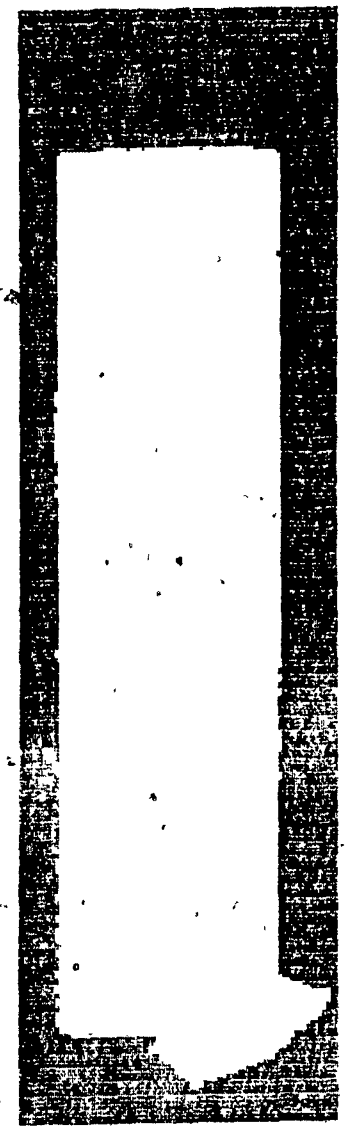
```
M(0)    =  0.696762D+07
M(1)    =  0.261039D+09
M(2)    =  0.115898D+11
M(3)    =  0.568964D+12
M(4)    =  0.249879D+10
M(5)    =  0.944188D+11
M(6)    =  0.423118D+13
M(7)    =  0.462604D+14
M(8)    =  0.176629D+16
M(9)    =  0.492504D+15

MU(1)   =  0.453643D+14
MU(2)   =  0.181010D+10
MU(3)   =  0.802520D+09
MU(4)   =  0.329950D+17
MU(5)   =  0.890869D+15
MU(6)   =  0.856025D+16
MU(7)   =  0.930634D+14

ETA(1)  =  0.934427D+00
ETA(2)  =  0.372849D-04
ETA(3)  =  0.165305D-04
ETA(4)  =  0.257476D+00
ETA(5)  =  0.695189D-02
ETA(6)  =  0.667999D-01
ETA(7)  =  0.726219D-03

PHI(1)  =  0.934464D+00
PHI(2)  =  0.873101D+00
PHI(3)  =  0.958589D-01
PHI(4)  =  0.744821D-01
PHI(5)  =  0.628593D-02
PHI(6)  =  0.804181D-01
PHI(7)  =  0.324852D-02

TIM1 14:12:15 TIM2          TIM3 14:12:15

% CHANGE IN PHI(1) =   41.0%
% CHANGE IN PHI(2) =   98.7%
% CHANGE IN PHI(3) =   48.8%
% CHANGE IN PHI(4) =   39.8%
% CHANGE IN PHI(5) =  101.5%
% CHANGE IN PHI(6) =   59.7%
% CHANGE IN PHI(7) =  104.3%
```

Figure 6.21: Image # 21.



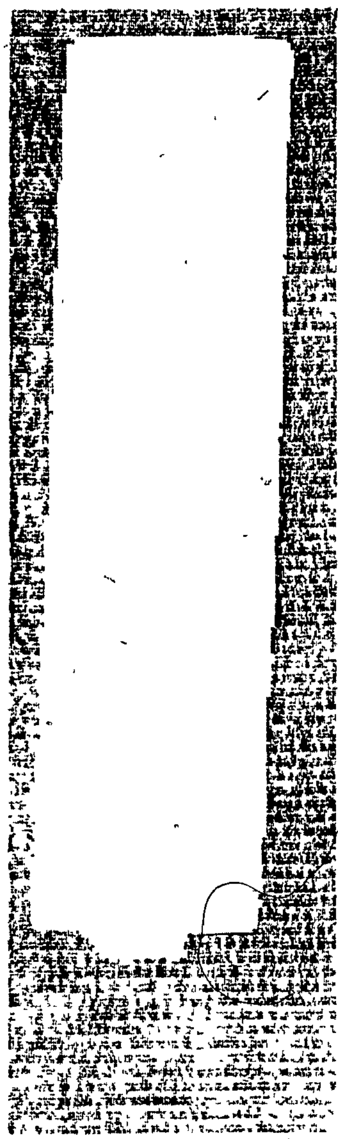
```
M(0)    =  0.853944D+07
M(1)    =  0.364058D+09
M(2)    =  0.187155D+11
M(3)    =  0.107364D+13
M(4)    =  0.316091D+10
M(5)    =  0.138735D+12
M(6)    =  0.732844D+13
M(7)    =  0.650186D+14
M(8)    =  0.302341D+16
M(9)    =  0.633855D+15

MU(1)   =  0.638486D+14
MU(2)   =  0.319476D+10
MU(3)   =  0.397747D+10
MU(4)   =  0.393042D+17
MU(5)   =  0.145931D+16
MU(6)   =  0.129184D+17
MU(7)   =  0.168073D+15

ETA(1)  =  0.875573D+00
ETA(2)  =  0.438106D-04
ETA(3)  =  0.545441D-04
ETA(4)  =  0.184445D+00
ETA(5)  =  0.684817D-02
ETA(6)  =  0.606226D-01
ETA(7)  =  0.788721D-03

PHI(1)  =  0.875617D+00
PHI(2)  =  0.766747D+00
PHI(3)  =  0.596529D-01
PHI(4)  =  0.403643D-01
PHI(5)  =  0.197140D-02
PHI(6)  =  0.707291D-01
PHI(7)  =  0.100425D-02

TIM1 14:10:34 TIM2          TIM3 14:10:35

% CHANGE IN PHI(1) =  32.1%
% CHANGE IN PHI(2) =  74.5%
% CHANGE IN PHI(3) =  -7.4%
% CHANGE IN PHI(4) = -24.2%
% CHANGE IN PHI(5) = -36.8%
% CHANGE IN PHI(6) =  40.5%
% CHANGE IN PHI(7) = -36.9%
```

Figure 6.22: Image # 22.



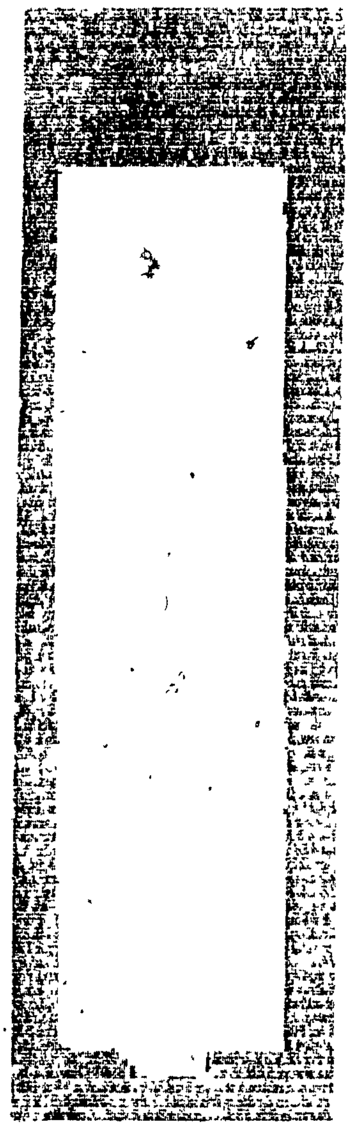
```
M(0)    =  0.894489D+07
M(1)    =  0.396672D+09
M(2)    =  0.210191D+11
M(3)    =  0.122856D+13
M(4)    =  0.278280D+10
M(5)    =  0.125865D+12
M(6)    =  0.679252D+13
M(7)    =  0.263778D+14
M(8)    =  0.119955D+16
M(9)    =  0.479591D+15

MU(1)   =  0.255121D+14
MU(2)   =  0.342822D+10
MU(3)   =  0.245875D+10
MU(4)   =  0.448109D+17
MU(5)   =  0.139008D+16
MU(6)   =  0.977992D+16
MU(7)   =  0.198135D+15

ETA(1)  =  0.318857D+00
ETA(2)  =  0.428468D-04
ETA(3)  =  0.307301D-04
ETA(4)  =  0.187260D+00
ETA(5)  =  0.580900D-02
ETA(6)  =  0.408694D-01
ETA(7)  =  0.827987D-03

PHI(1)  =  0.318900D+00
PHI(2)  =  0.101704D+00
PHI(3)  =  0.436738D-01
PHI(4)  =  0.390145D-01
PHI(5)  =  0.161025D-02
PHI(6)  =  0.275428D-01
PHI(7)  =  0.829986D-03

TIM1 14:09:00 TIM2              TIM3 14:09:01

% CHANGE IN PHI(1) = -51.9%
% CHANGE IN PHI(2) = -76.9%
% CHANGE IN PHI(3) = -32.2%
% CHANGE IN PHI(4) = -26.8%
% CHANGE IN PHI(5) = -48.4%
% CHANGE IN PHI(6) = -45.3%
% CHANGE IN PHI(7) = -47.8%
```

Figure 6.23: Image # 23.



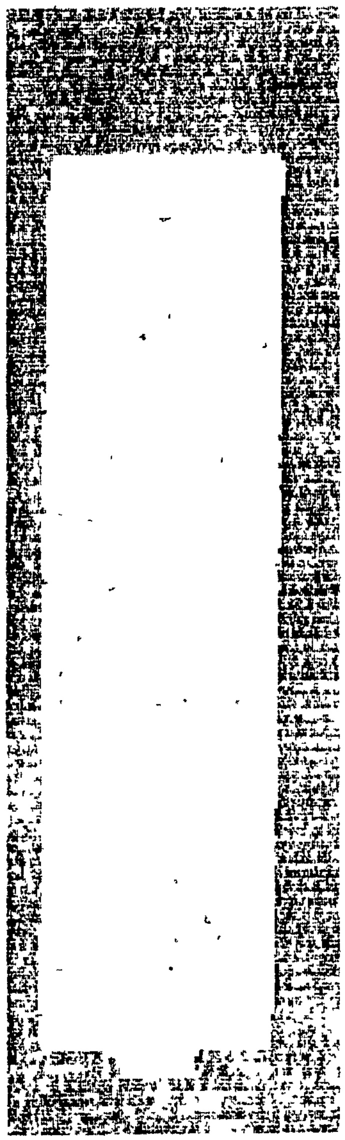
```
M(0)    =  0.712419D+07
M(1)    =  0.255110D+09
M(2)    =  0.109956D+11
M(3)    =  0.527662D+12
M(4)    =  0.262005D+10
M(5)    =  0.967050D+11
M(6)    =  0.426030D+13
M(7)    =  0.525804D+14
M(8)    =  0.198802D+16
M(9)    =  0.530332D+15

MU(1)   =  0.516168D+14
MU(2)   =  0.186041D+10
MU(3)   =  0.288381D+10
MU(4)   =  0.332371D+17
MU(5)   =  0.853358D+15
MU(6)   =  0.886718D+16
MU(7)   =  0.830897D+14

ETA(1)  =  0.101700D+01
ETA(2)  =  0.366553D-04
ETA(3)  =  0.568192D-04
ETA(4)  =  0.245349D+00
ETA(5)  =  0.629930D-02
ETA(6)  =  0.654556D-01
ETA(7)  =  0.613350D-03

PHI(1)  =  0.101703D+01
PHI(2)  =  0.103442D+01
PHI(3)  =  0.895994D-01
PHI(4)  =  0.676919D-01
PHI(5)  =  0.526309D-02
PHI(6)  =  0.121872D+00
PHI(7)  =  0.269868D-02

TIM1  14:17:23 TIM2            TIM3  14:17:23

   % CHANGE IN PHI(1) =   53.4%
   % CHANGE IN PHI(2) =  135.4%
   % CHANGE IN PHI(3) =   39.1%
   % CHANGE IN PHI(4) =   27.1%
   % CHANGE IN PHI(5) =   68.7%
   % CHANGE IN PHI(6) =  142.0%
   % CHANGE IN PHI(7) =   69.7%
```

Figure 6.24: Image # 24.



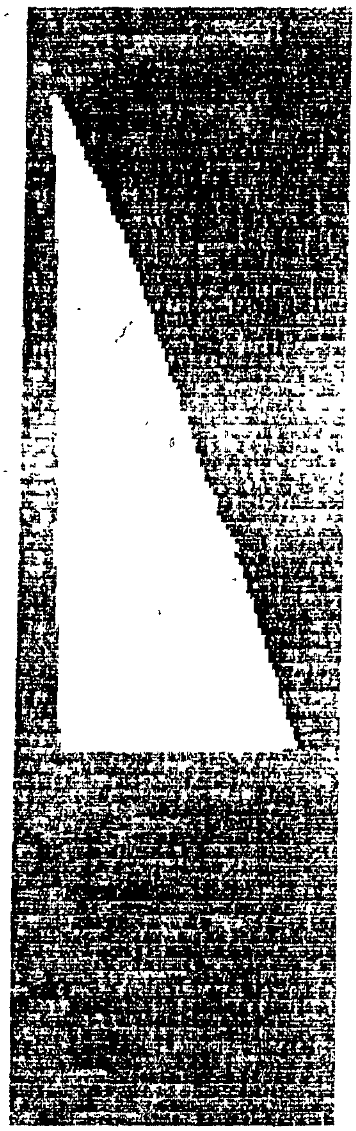
```
M(0)   =  0.712878D+07
M(1)   =  0.275165D+09
M(2)   =  0.124389D+11
M(3)   =  0.620509D+12
M(4)   =  0.212304D+10
M(5)   =  0.827645D+11
M(6)   =  0.378961D+13
M(7)   =  0.176876D+14
M(8)   =  0.687332D+15
M(9)   =  0.337729D+15

MU(1)  =  0.170553D+14
MU(2)  =  0.181779D+10
MU(3)  =  0.817086D+09
MU(4)  =  0.327389D+17
MU(5)  =  0.803450D+15
MU(6)  =  0.620298D+16
MU(7)  =  0.104132D+15

ETA(1) =  0.335606D+00
ETA(2) =  0.357694D-04
ETA(3) =  0.160782D-04
ETA(4) =  0.241283D+00
ETA(5) =  0.592135D-02
ETA(6) =  0.457154D-01
ETA(7) =  0.767440D-03

PHI(1) =  0.335642D+00
PHI(2) =  0.112625D+00
PHI(3) =  0.685597D-01
PHI(4) =  0.632705D-01
PHI(5) =  0.416693D-02
PHI(6) =  0.318894D-01
PHI(7) =  0.198482D-02

TIM1 14:15:52 TIM2            TIM3 14:15:53

% CHANGE IN PHI(1) = -49.4%
% CHANGE IN PHI(2) = -74.4%
% CHANGE IN PHI(3) =   6.4%
% CHANGE IN PHI(4) =  18.8%
% CHANGE IN PHI(5) =  33.6%
% CHANGE IN PHI(6) = -36.7%
% CHANGE IN PHI(7) =  24.8%
```

Figure 6.25: Image # 25.



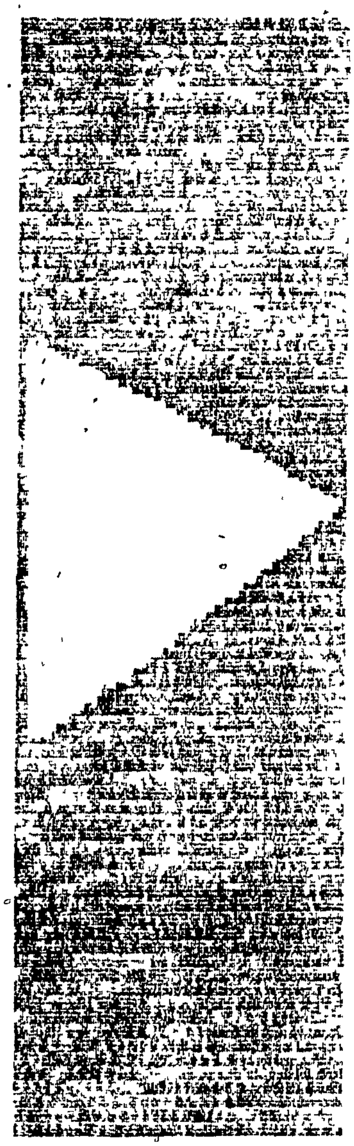
```
M(0)    =  0.732360D+07
M(1)    =  0.270301D+09
M(2)    =  0.118540D+11
M(3)    =  0.575881D+12
M(4)    =  0.223776D+10
M(5)    =  0.844896D+11
M(6)    =  0.378537D+13
M(7)    =  0.231225D+14
M(8)    =  0.923751D+15
M(9)    =  0.365823D+15

MU(1)   =  0.224388D+14
MU(2)   =  0.187763D+10
MU(3)   =  0.189786D+10
MU(4)   =  0.326551D+17
MU(5)   =  0.774295D+15
MU(6)   =  0.647919D+16
MU(7)   =  0.935247D+14

ETA(1)  =  0.418360D+00
ETA(2)  =  0.350074D-04
ETA(3)  =  0.353846D-04
ETA(4)  =  0.224978D+00
ETA(5)  =  0.533451D-02
ETA(6)  =  0.446385D-01
ETA(7)  =  0.644340D-03

PHI(1)  =  0.418395D+00
PHI(2)  =  0.175078D+00
PHI(3)  =  0.614314D-01
PHI(4)  =  0.550943D-01
PHI(5)  =  0.320485D-02
PHI(6)  =  0.455167D-01
PHI(7)  =  0.154856D-02

TIM1 14:14:21 TIM2          TIM3 14:14:21

% CHANGE IN PHI(1) = -36.9%
% CHANGE IN PHI(2) = -60.2%
% CHANGE IN PHI(3) =  -4.6%
% CHANGE IN PHI(4) =   3.4%
% CHANGE IN PHI(5) =   2.7%
% CHANGE IN PHI(6) =  -9.6%
% CHANGE IN PHI(7) =  -2.6%
```

Figure 6.26: Image # 26.



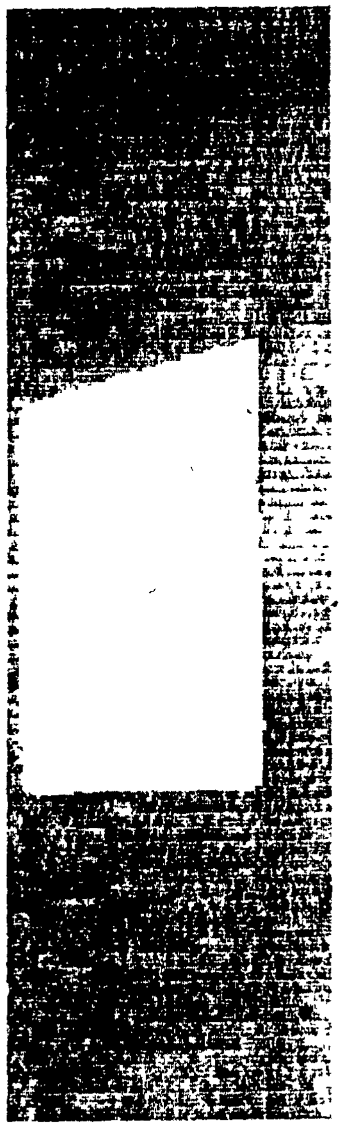
```
M(0)    =  0.400554D+07
M(1)    =  0.158422D+09
M(2)    =  0.771635D+10
M(3)    =  0.421172D+12
M(4)    =  0.149090D+10
M(5)    =  0.599574D+11
M(6)    =  0.299563D+13
M(7)    =  0.344627D+14
M(8)    =  0.143634D+16
M(9)    =  0.261455D+15

MU(1)   =  0.339078D+14
MU(2)   =  0.145066D+10
MU(3)   =  0.991368D+09
MU(4)   =  0.146558D+17
MU(5)   =  0.592410D+15
MU(6)   =  0.564729D+16
MU(7)   =  0.629455D+14

ETA(1)  =  0.211338D+01
ETA(2)  =  0.904158D-04
ETA(3)  =  0.617892D-04
ETA(4)  =  0.456412D+00
ETA(5)  =  0.184489D-01
ETA(6)  =  0.175868D+00
ETA(7)  =  0.196025D-02

PHI(1)  =  0.211347D+01
PHI(2)  =  0.446624D+01
PHI(3)  =  0.437156D+00
PHI(4)  =  0.257116D+00
PHI(5)  =  0.851551D-01
PHI(6)  =  0.751652D+00
PHI(7)  =  0.387269D-01

TIM1 14:21:14 TIM2            TIM3 14:21:14

% CHANGE IN PHI(1) = 218.8%
% CHANGE IN PHI(2) = 916.3%
% CHANGE IN PHI(3) = 578.6%
% CHANGE IN PHI(4) = 382.6%
% CHANGE IN PHI(5) =  2630.0%
% CHANGE IN PHI(6) =  1392.7%
% CHANGE IN PHI(7) =  2335.0%
```

Figure 6.27: Image # 27.



```
M(0)    =  0.398412D+07
M(1)    =  0.173910D+09
M(2)    =  0.976424D+10
M(3)    =  0.602113D+12
M(4)    =  0.164792D+10
M(5)    =  0.779715D+11
M(6)    =  0.455190D+13
M(7)    =  0.369960D+14
M(8)    =  0.189923D+16
M(9)    =  0.334869D+15

MU(1)   =  0.363144D+14
MU(2)   =  0.217293D+10
MU(3)   =  0.603865D+10
MU(4)   =  0.266019D+17
MU(5)   =  0.797528D+15
MU(6)   =  0.783658D+16
MU(7)   =  0.841533D+14

ETA(1)  =  0.228778D+01
ETA(2)  =  0.136893D-03
ETA(3)  =  0.380430D-03
ETA(4)  =  0.839618D+00
ETA(5)  =  0.251718D-01
ETA(6)  =  0.247341D+00
ETA(7)  =  0.265607D-02

PHI(1)  =  0.228792D+01
PHI(2)  =  0.524279D+01
PHI(3)  =  0.113051D+01
PHI(4)  =  0.810360D+00
PHI(5)  =  0.773433D+00
PHI(6)  =  0.695800D+01
PHI(7)  =  0.398024D+00

TIM1 14:19:24 TIM2              TIM3 14:19:24

% CHANGE IN PHI(1) = 245.1%
% CHANGE IN PHI(2) =  1093.0%
% CHANGE IN PHI(3) =  1654.9%
% CHANGE IN PHI(4) =  1421.1%
% CHANGE IN PHI(5) =  24695.4%
% CHANGE IN PHI(6) =  13717.5%
% CHANGE IN PHI(7) =  24926.3%
```

Figure 6.28: Image # 28.



```
M(0)    =  0.405552D+07
M(1)    =  0.143490D+09
M(2)    =  0.635014D+10
M(3)    =  0.310162D+12
M(4)    =  0.136165D+10
M(5)    =  0.507825D+11
M(6)    =  0.233118D+13
M(7)    =  0.290388D+14
M(8)    =  0.115620D+16
M(9)    =  0.197569D+15

MU(1)   =  0.285816D+14
MU(2)   =  0.127326D+10
MU(3)   =  0.260541D+10
MU(4)   =  0.102436D+17
MU(5)   =  0.432977D+15
MU(6)   =  0.423561D+16
MU(7)   =  0.456207D+14

ETA(1)  =  0.173778D+01
ETA(2)  =  0.774146D-04
ETA(3)  =  0.158410D-03
ETA(4)  =  0.309268D+00
ETA(5)  =  0.130722D-01
ETA(6)  =  0.127879D+00
ETA(7)  =  0.137735D-02

PHI(1)  =  0.173785D+01
PHI(2)  =  0.302125D+01
PHI(3)  =  0.219050D+00
PHI(4)  =  0.120610D+00
PHI(5)  =  0.192575D-01
PHI(6)  =  0.584063D+00
PHI(7)  =  0.806967D-02

TIM1 14:24:10 TIM2          TIM3 14:24:10

% CHANGE  IN PHI(1)  = 162.1%
% CHANGE  IN PHI(2)  = 587.5%
% CHANGE  IN PHI(3)  = 240.0%
% CHANGE  IN PHI(4)  = 126.4%
% CHANGE  IN PHI(5)  = 517.4%
% CHANGE  IN PHI(6)  =  1059.9%
% CHANGE  IN PHI(7)  = 407.4%
```

6.1 Testing Results

% OF CHANGE						
	IMAGE # 1	IMAGE # 2	IMAGE # 3	IMAGE # 4	IMAGE # 5	IMAGE # 6
ϕ_1 :	00.0 %	-4.6 %	-8.8 %	9.8 %	7.5 %	8.0 %
ϕ_2 :	00.0 %	-9.1 %	-16.8 %	20.6 %	15.5 %	16.6 %
ϕ_3 :	00.0 %	-8.5 %	-18.5 %	1.0 %	-5.4 %	-1.6 %
ϕ_4 :	00.0 %	-7.6 %	-17.0 %	-0.5 %	-7.8 %	-3.6 %
ϕ_5 :	00.0 %	-15.0 %	-31.7 %	-0.3 %	-13.9 %	-6.2 %
ϕ_6 :	00.0 %	-1.2 %	0.7 %	-1.4 %	-30.9 %	-23.6 %
ϕ_7 :	00.0 %	-17.4 %	-36.1 %	1.8 %	9.6 %	-2.2 %
AVG.=	00.0 %	9.0 %	18.5 %	5.0 %	12.9 %	8.2 %

% OF CHANGE						
	IMAGE # 7	IMAGE # 8	IMAGE # 9	IMAGE #10	IMAGE #11	IMAGE #12
ϕ_1 :	-4.6 %	2.0 %	-6.6 %	-10.7 %	12.5 %	25.5 %
ϕ_2 :	-9.0 %	4.0 %	-12.7 %	-20.2 %	26.5 %	57.6 %
ϕ_3 :	-2.5 %	1.8 %	0.4 %	-19.0 %	40.3 %	49.1 %
ϕ_4 :	-3.5 %	-2.4 %	4.5 %	-21.8 %	36.9 %	40.4 %
ϕ_5 :	-6.3 %	-2.8 %	-6.4 %	-37.8 %	89.7 %	103.1 %
ϕ_6 :	-6.2 %	4.8 %	8.3 %	28.8 %	74.6 %	233.2 %
ϕ_7 :	-2.9 %	6.7 %	5.9 %	-32.5 %	94.1 %	108.2 %
AVG.=	5.0 %	3.5 %	6.4 %	24.4 %	53.5 %	88.2 %

% OF CHANGE						
	IMAGE #13	IMAGE #14	IMAGE #15	IMAGE #16	IMAGE #17	IMAGE #18
ϕ_1 :	-89.4 %	-73.2 %	-10.7 %	9.9 %	18.4 %	-64.3 %
ϕ_2 :	-98.9 %	-92.8 %	-20.2 %	20.9 %	40.1 %	-87.3 %
ϕ_3 :	24.4 %	89.9 %	121.1 %	49.4 %	73.9 %	-11.9 %
ϕ_4 :	48.4 %	122.3 %	132.7 %	45.0 %	61.6 %	0.2 %
ϕ_5 :	101.7 %	356.9 %	428.1 %	113.4 %	170.8 %	-5.9 %
ϕ_6 :	-34.1 %	82.6 %	257.1 %	139.3 %	47.9 %	-37.9 %
ϕ_7 :	70.7 %	317.3 %	431.0 %	120.5 %	186.8 %	-10.6 %
AVG.=	66.8 %	162.1 %	200.1 %	72.2 %	85.6 %	31.2 %

% OF CHANGE						
	IMAGE #19	IMAGE #20	IMAGE #21	IMAGE #22	IMAGE #23	IMAGE #24
ϕ_1 :	6.[illegible] %	41.0 %	-32.1 %	-51.9 %	53.4 %	-49.4 %
ϕ_2 :	12.6 %	98.7 %	74.5 %	-76.9 %	135.4 %	-74.4 %
ϕ_3 :	2.9 %	48.4 %	-7.4 %	-32.2 %	39.1 %	6.4 %
ϕ_4 :	-2.6 %	39.8 %	-24.2 %	-26.8 %	27.1 %	18.8 %
ϕ_5 :	-2.6 %	101.5 %	-36.8 %	-48.4 %	68.7 %	33.6 %
ϕ_6 :	180.0 %	59.7 %	40.5 %	-45.3 %	142.0 %	-36.7 %
ϕ_7 :	-0.2 %	104.3 %	-36.9 %	-47.8 %	69.7 %	24.8 %
AVG.=	19.3 %	70.6 %	36.1 %	47.1 %	76.5 %	34.9 %

Figure 6.29 Summary of Results and % of deviation.

6.2 Discussion of Results

6.2.1 Errors Due to Hardware and the Edge Enhancement Solution

Digital computers can not directly process continuous data. Integrals over infinite limits must be replaced by summations over finite range. (see eq.- 1, 2, 1.1, and 2.1). This approximation introduces slight errors moment invariants. The seven moment invariants of the rotated image in Figure 10.2-10.10 show a relatively small percentage of change. These variations or deviation from the theory are due to three reasons other than digitization (approximation) error:

1- The raw data of a contiguous image is not contiguous itself, but looks like a checker-board. This is due to the fact that the IS256 OpticRAM has dead or inactive pixels covering more than 50% of its total area (see Figure 4.5). In order to apply the delta method on the digitized image, the raw image had to be enhanced. The first routine had to fill in the blank pixels and the second enhancing routine had to correct the aspect ratio in the X and Y coordinates (see [24] for a complete listings of the IDETIX subroutines). The resulting image is an enhanced image of 80 X 640. Its contour, however, has ragged edges (see Figure 6.1-6.28).

2- The distortion generated by the wide-angle lens is significant. A square object, for example, will not necessarily give a square image after translation and/or rotation. To minimize this distortion, the OpticRAM physical area used was rows 300 - 379 which is in the centre of the OpticRAM chip (see Figure 4.5).

3- The delta method applied to an image such as shown in Figure 6.30 would eliminate six entire rows of pixels, as a black pixel circled means the end of a line.

This was verified by tracing the computations of the moments and the *delta method* eliminated six entire lines, as shown in Figure 6.31.

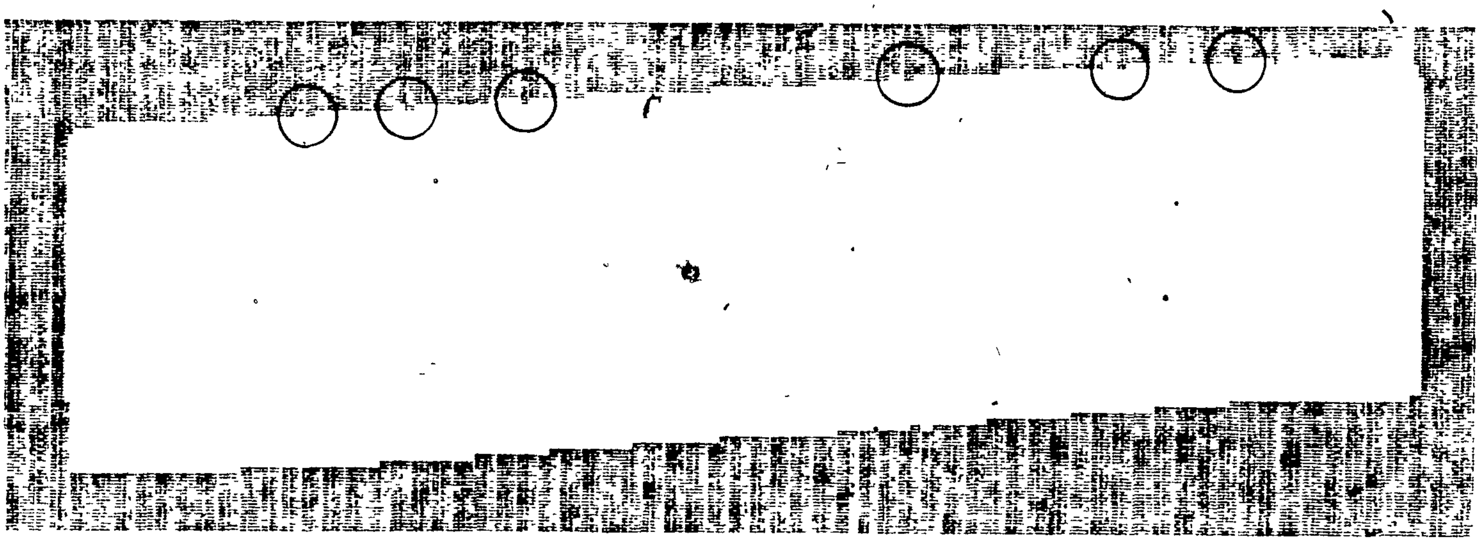


Figure 6.30 Rotated Image.

This problem was corrected by further enhancing the image, by adding a special subroutine to the delta method program (lines 56000-57000). This subroutine will scan vertically, and calculate approximately the length of the object in the Y-direction (see Figure 3.1 for axis and APPENDIX III for the program) and store it in memory, under the variable LY (Length in Y), along with the first set of moment invariants.

6.2 Discussion of Results

During the regular scanning procedure, if a byte, on the edge of the image, has one bit "off", and the next byte is non-zero, this bit will be turned "on" and scanning will continue. This feature will still allow the program to detect holes in the fabric as the edge flag is not set and a single "off" bit in a line will cancel the remainder.

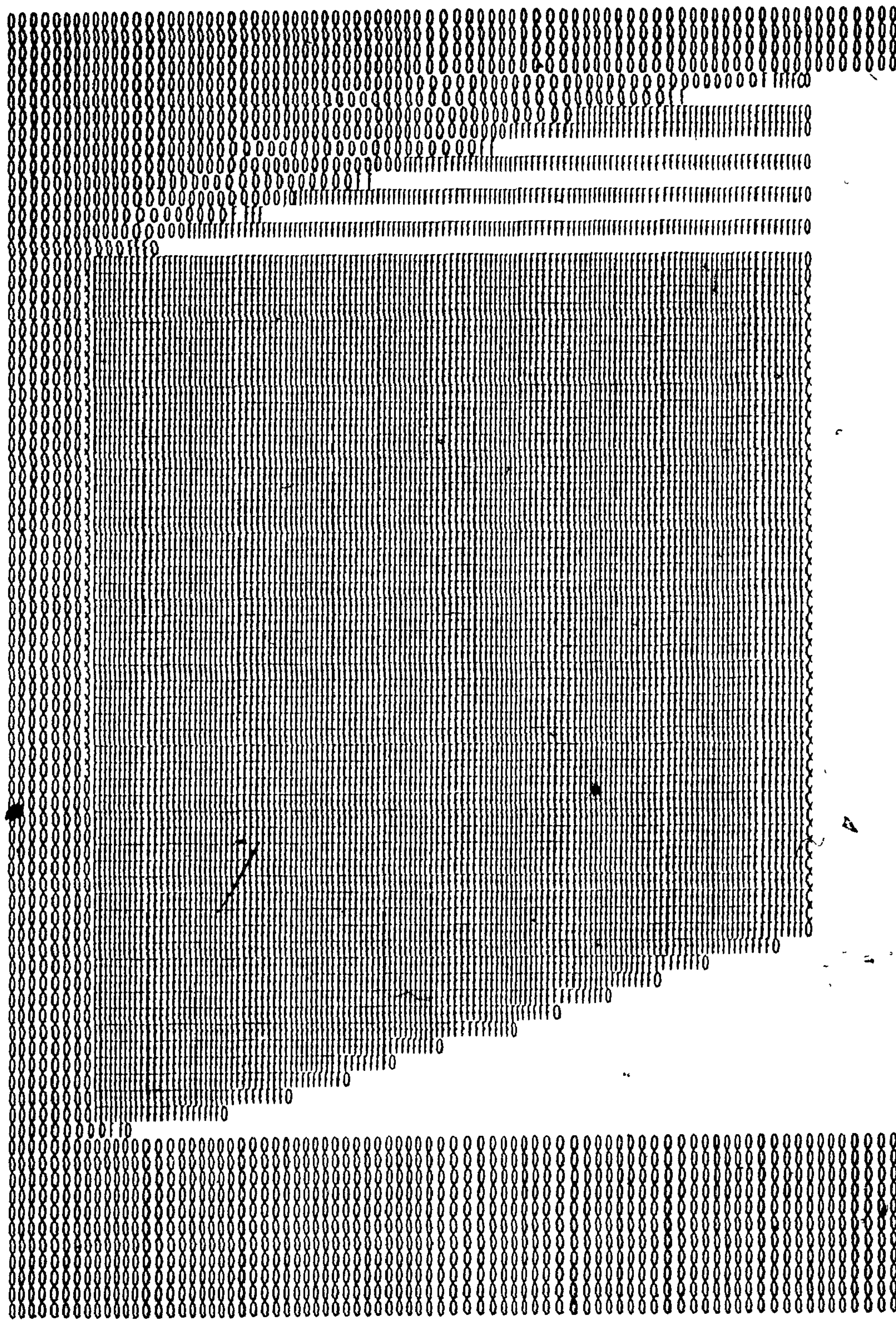


Figure 6.31 The Rotated Image as the Delta Method sees it before Edge Enhancement.

6.2.2 Description of Results and their Sensitivity

Results and their percentage deviation are summarized in the tables of Figure 6.29. The moment invariants of Figure 6.1-6.10 of the image after rotation and translation, show variations in the ϕ's of up to 40% for a 'good' images (no holes or unwanted flaws are present) and a variation of up to 25% over the mean value of all the seven ϕ's. The reason of these variations is contributed to the digitization and undersampling errors discussed earlier.

Creating a hole of 0.4% of the total area of the object changes the percentage deviation of the moment invariant enormously (see results of IMAGE #11 through IMAGE #20). The same magnitude of change took place when an extra amount of material of 0.67% or more was added to the original image.

Therefore the method is sensitive up to 99.6% for holes and discontinuities and up to 99.33% for other flaws. The sensitivity of the hardware (the vision-system) was calculated to be 99.64% (see section on sample calculations). *This gives an overall sensitivity (accuracy) for the combined system (hardware/software) of more than 99.33%.*

6.3 Criteria for Accepting a 'GOOD' Object

To compensate and rectify the error due to the hardware design discussed earlier, we will allow a 40% tolerance change in any ϕ and an overall change of 25% from the mean value of the seven ϕ's before accepting a 'good' object. Therefore the criteria are as follows:

If any of the two following conditions are satisfied, the object is labeled 'BAD' or 'DEFECTIVE' : percentage deviation *of any given ϕ is above 40% AND/OR percentage deviation from the mean value of all seven ϕ's is above 25%.*

Chapter 7 LIGHTING CONSIDERATIONS

7.1 Introduction

The IDETIX camera needs a high contrast scene in order to image the object into the OpticRAM. Unlike a TV camera which can respond to shades of gray, the OpticRAMs are digital chips where each picture element makes a black/white judgment based on an arbitrary light level used as a threshold (trip light level). Portions of the scene that are lighter than the threshold level will be judged as black.

Doubling the exposure time is the same as opening the f-stop by one (changing the f-stop to the next smaller number) or, in other words, doubling the amount of light. Contrast can now be defined as a minimum difference between adjacent threshold changes or slices.

7.2 Front Lighting

A front lit scene, where the camera is on the same side of the scene as the source light or ambient light, is usually low in contrast. In this case extreme care in setting up uniform lighting on the scene is necessary and the optimum trip light level needs to be used. Front lighting requires multiple diffused light sources such that the contrast in the scene is increased.

7.3 Back Lighting

For a backlit scene, the light comes from behind the scene so that the object being viewed is shadowed into the camera. Backlighting the object for maximum contrast will give the best repeatable results. Backlighting is recommended if the camera is used to measure the object. Other techniques of illuminating the object such as shadow imaging, spectral illumination, spectral elimination and collimated lighting are illustrated in Figure 7.1.

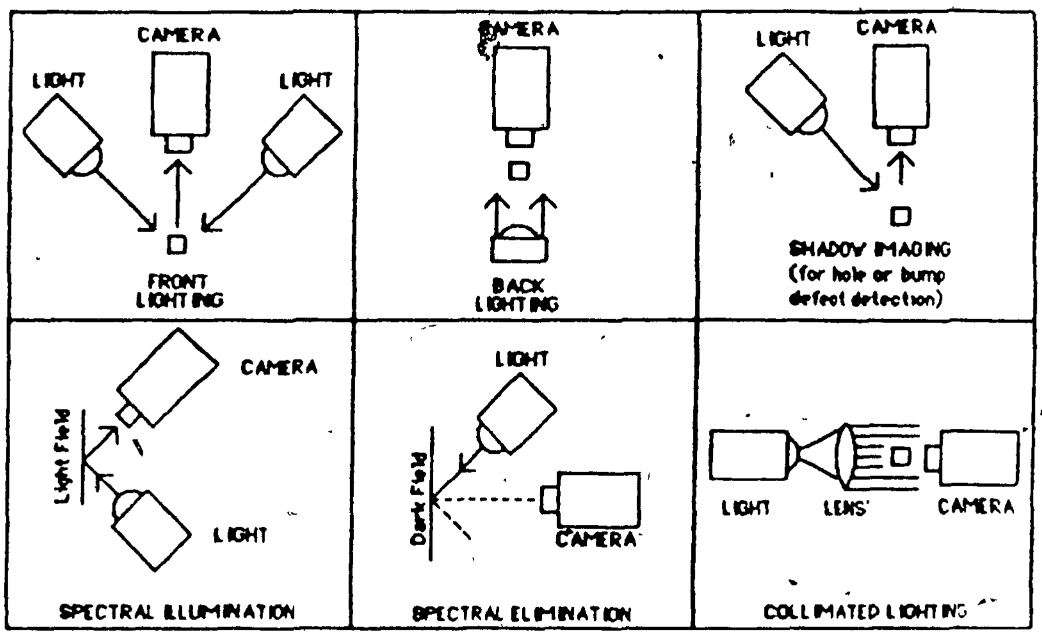


Figure 7.1 Illumination Techniques.

Chapter 8 CONCLUSION

The solution for the vision system, presented in this thesis, consists of the following parts:

The Hardware: An IDETIX vision system by MICRON TECHNOLOGIES INC. equipped with an IS256 OpticRAM, any IBM AT compatible Personal Computer equipped with an 80287 co-processor, running at 8 MHz.

The software: An application program that accepts a digitized image of size 80 X 640 pixels as an input, and calculates the moment invariants using the *delta method*, in real-time, spending less than 0.64 second.

The sensitivity of the hardware (the vision system) is 99.64% and the experimental sensitivity of the combined hardware/software system is above 99.33%.

The economy in hardware and software, the execution speed and the achieved accuracy/sensitivity make this work and the resulting system implementation particularly suited to industrial application. Extension of this system; and the method it embodies, to 3-dimensional industrial recognition problems appears to be quite promising.

REFERENCES

1. M. K. Hu, Pattern Recognition by Moment Invariants, *Proc. IRE*, 49, September 1961, pp. 1428

2. M. K. Hu, Visual Pattern Recognition by Moment Invariants, *IRE Trans. Inform. Theory*. Vol. IT-8, February 1962, pp. 179-187.

3. F. Alt, *Digital Pattern Recognition By Moments, Optical Character Recognition*. (G. L. Fischer *et al.* Eds.) pp.153-179, Spartan. Washington, D.C., 1962; also published in *J. ACM*. 9, April 1962, pp.240-258.

4. P.F. Lambert, Designing Pattern Categorizers with External Paradigm Information, Methodologies of Pattern Recognition (S. Watanabe, Ed.), Academic Press, New York, 1969.

5. R.G. Casey, Moment Normalization of Handprinted Characters *IBM J. Res. Develop*. 14, September 1970, pp. 548-557.

6. F. W. Smith and M. H. Wright, Automatic Ship Photo Interpretation by the Method of Moments, *IEEE Trans. Computers* C-20, September 1971, pp. 39-46.

7. E.L. Hall, W.O. Crawford, and F.E. Roberts, Computer Classification of Pneumoconiosis from Radiographs of Coal Workers, *IEEE Trans. Biomedical Engineering* BME-22, Nov. 1975, pp. 518-527.

8. E. L. Hall, R. P. Kruger, S. J. Dwyer, R. W. McLaren and G. S. Lodwick, A Survey of Preprocessing and Feature Extraction Techniques of Radio-graphic Images. *IEEE Tran. Comput*. 20, No. 9, pp. 1032-1044.

9. E.L. Hall and W. Frei. Invariant Features for Quantitative Scene Analysis. Final Rep., Contract F 08606-72-C-0008, *Image Process. Inst.* Univ. of Southern California, Los Angeles, 1976.

10. S. A. Dudani, K. J. Kenneth, and R. B. McGhee, Aircraft Identification by Moment Invariants, *IEEE Trans.Computers* C-26, January 1977, pp. 39-46.

11. F. A. Sadjadi and E. L. Hall, Numerical Computation of Moment Invariants for Scene Analysis, *Proceedings of 1978 IEEE Conference on Pattern Recognition and Image Processing, Chicago. IL. 1978,* pp. 181-187

References

12. S. Maitra, Moment Invariants, *Proc. IEEE.* 67, No. 4, April 1979, pp. 697-699.

13. E. L. Hall, *Computer Image Processing and Recognition.* Academic Press, 1979.

14. F. A. Sadjadi and E. L. Hall, Three-Dimensional Moment Invariants, *IEEE Trans. Pattern Analysis and Machine Intelligence 2,* No. 2, March 1980, pp. 127-136

15. T.C. Hsia, A Note on Invariant Moments in Image Processing *IEEE Trans. System, Man, and Cyber.* SMC-11, No. 12, Dec. 1981, pp. 831-834.

16. S.S. Reddi, Radial and Angular Moment Invariants for Image Identification, *IEEE Trans. Pattern Analysis and Machine Intelligence* 3, No. 2, March 1981, pp. 240-242.

17. Cho-Huak Teh and Roland T. Chin, On Digital Approximation of Moment Invariants, *Proceedings of 1985 IEEE Conference on Pattern Recognition and Image Processing, San Francisco CA. 1985,* pp. 640-642

18. M. Hatamian, A Real-Time Two-Dimensional Moment Generating Algorithm and Its Single Chip Implementation, *IEEE Trans. on Acoustics, Speech, and Signal Processing,* V. ASSP-34, No. 3, June 1986, pp. 546-553.

19. M.F. Zakaria, L.J. Vroomen, P.J. Zsombor-Murray and J.M.H. Van Kessel, Fast Algorithm For The Computation of Moment Invariants, Paper accepted for publications in *The Journal Of Pattern Recognition Society*, April, 1987.

20. M.F. Zakaria, L.J. Vroomen, P.J. Zsombor-Murray and J.M.H. Van Kessel, Real-Time Computations of Moment Invariants, *Proceedings of The 5th Scandinavian Conference on Image Analysis,* Stokholm June 2-5, 1987.

21. M.F. Zakaria, L.J. Vroomen, P.J. Zsombor-Murray and J.M.H. Van Kessel, A Fast Algorithm For Moment Invariants Generation, *Proceedings of The IV International Conference on Image Analysis and Processing,* Cefalu' Italy, September 23-25, 1987.

22. intel Corporation, *Microsystem Components Handbook (Microprocessors Volume I)*, 1986, Santa Clara, CA, pp. 4.56-4.81

23. E. Elliot, Algebra of Quantics, Oxford University Press, 1913.

24. MICRON TECHNOLOGIES INC. User's Manual for the IDETIX system.

APPENDIX I - LISTING OF THE FORTRAN-77 PROGRAM

```
$NOFLOATCALLS

C       ---------------------------------------------------------------
C       |                                                             |
C       |              CALCULATION  OF THE  MOMENT INVARIANTS         |
C       |        FOR AN IMAGE USING THE STRAIGHT-FORWARD APPROACH     |
C       |                                                             |
C       |-------------------------------------------------------------|
C       |                                                             |
C       |              Copyright (C) MR. MARWAN ZAKARIA               |
C       |             ROBOTIC MECHANICAL SYSTEMS LABORATORY           |
C       |              MCGILL UNIVERSITY, MONTREAL CANADA             |
C       |                                                             |
C       ---------------------------------------------------------------
C
C
C       interface for access to time and date
C
        INTERFACE TO SUBROUTINE TIME (N,STR)
        CHARACTER*10 STR [NEAR,REFERENCE]
        INTEGER*2 N [VALUE]
        END
C
        INTERFACE TO SUBROUTINE DATE (N,STR)
        CHARACTER*10 STR [NEAR,REFERENCE]
        INTEGER*2 N [VALUE]
        END
C
        CHARACTER*10 TIME1,TIME2,TIME3,TESTDATE
        INTEGER*2 F(256,256),I,J,N,M,NN,MM
        REAL MU(7),ETA(7),PHI(7),XX,YY,MX(10)
        CHARACTER*1 CF(256,256)
C
C       Get Time/Date
C
        CALL DATE (10,TESTDATE)
        CALL TIME (10,TIME1)
            READ(*,550) ((CF(I,J),J=1,256),I=1,256)
        DO 5 I=1,10
          MX(I)=0
    5   CONTINUE
        DO 6 I=1,256
          DO 6 J=1,256
            F(I,J)=ICHAR(CF(I,J))
    6   CONTINUE
        CALL TIME (10,TIME2)
```

```
      DO 7 I=1,128
        IF (F(I,128).GT.10) THEN
          M=I
          GOTO 8
        ENDIF
  7   CONTINUE
  8   MM=256-M
      DO 9 I=1,128
        IF (F(128,I).GT.10) THEN
          N=I
          GOTO 91
        ENDIF
  9   CONTINUE
 91   NN=256-N
      DO 10 I=M,MM
        DO 10 J=N,NN
          IIF=I*F(I,J)
          JJF=J*F(I,J)
          II=I*I
          JJ=J*J
          MX(1)=MX(1)+F(I,J)
          MX(2)=MX(2)+JJF
          MX(3)=MX(3)+JJ*JJF
          MX(4)=MX(4)+IIF
          MX(5)=MX(5)+I*JJF
          MX(6)=MX(6)+I*IIF
          MX(7)=MX(7)+II*JJF
          MX(8)=MX(8)+II*IIF
          MX(9)=MX(9)+JJ*IIF
          MX(10)=MX(10)+J*JJF
 10   CONTINUE
      XX=MX(4)/MX(1)
      YY=MX(2)/MX(1)
      MU(1)=MX(6)-XX*MX(4)
      MU(2)=MX(10)-YY*MX(2)
      MU(3)=MX(5)-YY*MX(4)
      MU(4)=MX(8)-3*XX*MX(6)+2*MX(4)*XX**2
      MU(5)=MX(9)-2*YY*MX(5)-XX*MX(10)+2*YY**2*MX(4)
      MU(6)=MX(7)-2*XX*MX(5)-YY*MX(6)+2*XX**2*MX(2)
      MU(7)=MX(3)-3*YY*MX(10)+2*YY**2*MX(2)
      ETA(1)=MU(1)/(MX(1)**2)
      ETA(2)=MU(2)/(MX(1)**2)
      ETA(3)=MU(3)/(MX(1)**2)
      ETA(4)=MU(4)/(MX(1)**2.5)
      ETA(5)=MU(5)/(MX(1)**2.5)
      ETA(6)=MU(6)/(MX(1)**2.5)
      ETA(7)=MU(7)/(MX(1)**2.5)
      PHI(1)=ETA(1)+ETA(2)
      PHI(2)=(ETA(1)-ETA(2))**2+4*ETA(3)**2
      PHI(3)=(ETA(4)-3*ETA(5))**2+(3*ETA(6)-ETA(7))**2
      PHI(4)=(ETA(4)+ETA(5))**2+(ETA(6)+ETA(7))**2
```

```
      PHI(5)=(ETA(4)-3*ETA(5))*(ETA(4)+ETA(5))*((ETA(4)+
     + ETA(5))**2-3*(ETA(6)+ETA(7))**2) +(3*ETA(6)-ETA(7)
     + )*(ETA(6)+ETA(7))*(3*(ETA(4)+ETA(5))**2-(ETA(6)+ETA(7))**2)
      PHI(6)=(ETA(1)-ETA(2))*((ETA(4)+ETA(5))**2-(ETA(6)
     + +ETA(7))**2) + 4*ETA(3)*(ETA(4)+ETA(5))*(ETA(6)+
     + ETA(7))
      PHI(7)=(3*ETA(6)-ETA(7))*(ETA(4)+ETA(5))*((ETA(4)+
     + ETA(5))**2-3*(ETA(6)+ETA(7))**2) +
     + (3*ETA(5)-ETA(7))*(ETA(6)+ETA(7))*(3*(ETA(4)+
     + ETA(5))**2-(ETA(6)+ETA(7))**2)
      CALL TIME (10,TIME3)
 500  FORMAT(' ',T15,'****** Starting Time 1: ',A10,'*******',/
     +        ' ',T15,'****** Today''s   Date : ',A10,'*******',/)
 550  FORMAT (100A1/)
 600  FORMAT(' ',T15,'****** Starting Time 2: ',A10,'*******',/)
 700  FORMAT(' ',T15,'****** Ending   Time 3: ',A10,'*******'/)
1000  FORMAT(' ',T23,'PHI(',I1,') =',D25.14)
2000  FORMAT(' ',T23,'ETA(',I1,') =',D25.14)
3000  FORMAT(' ',T23,'MU(',I1,')  =',D25.14)
4000  FORMAT(' ',T23,'MX(',I2,')  =',D25.14)
5000  FORMAT(' ',//)
6000  FORMAT(' ',T15,'M=',I3,' MM=',I3,' N=',I3,' NN=',I3//)
      DO 100 I=1,10
        WRITE(*,4000) I,MX(I)
 100  CONTINUE
      WRITE(*,5000)
      DO 101 I=1,7
        WRITE(*,3000) I,MU(I)
 101  CONTINUE
      WRITE(*,5000)
      DO 102 I=1,7
        WRITE(*,2000) I,ETA(I)
 102  CONTINUE
      WRITE(*,5000)
      DO 103 I=1,7
        WRITE(*,1000) I,PHI(I)
 103  CONTINUE
      WRITE(*,5000)
      WRITE(*,500) TIME1,TESTDATE
      WRITE(*,600) TIME2
      WRITE (*,700)TIME3
      WRITE (*,6000) M,MM,N,NN
      STOP
      END
```

APPENDIX II - LISTING OF THE TURBO PASCAL PROGRAM

```
{-----------------------------------------------------------}
|                                                           |
|        CALCULATION  OF THE  2-DIMENSIONAL MOMENTS         |
|            OF AN IMAGE USING THE DELTA METHOD             |
|                                                           |
|-----------------------------------------------------------|
|                                                           |
|            Copyright (C) MR. MARWAN ZAKARIA               |
|          ROBOTIC MECHANICAL SYSTEMS LABORATORY            |
|           MCGILL UNIVERSITY, MONTREAL CANADA              |
|                                                           |
L-----------------------------------------------------------J

program linesc;

type Mar      =array[0..9] of real;

var XST,YST,L :integer;
    M         :Mar;
    P         :array[0..63,0..127] of byte;
    X,LE      :integer;
    A,B,C,D   :INTEGER;



PROCEDURE Mnm_Update(XST,YST,L:integer;VAR M:Mar);

VAR L2,X2,Y2 :real;
    S1,S2,S3,L3 :real;
    M10,M20     :real;
    s4          :real;


begin
  L2:=L*L;
  L3:=L2*L;
  X2:=XST*XST;
  Y2:=YST*YST;
  S1:=(L2-L)/2;
  S2:=(2*L3-3*L2+L)/6;
  S3:=(L*L3-2*L3+L2)/4;
  s4:=1;
  M[0]:=M[0]+L;
  M[1]:=M[1]+s4*L*YST;
  M[2]:=M[2]+s4*L*Y2;
  M[3]:=M[3]+s4*L*Y2*YST;
```

```
  M10:=s4*L*XST+S1;
  M[4]:=M[4]+M10;
  M[5]:=M[5]+s4*YST*M10;
  M[6]:=M[6]+s4*Y2*M10;
  M20:=s4*L*X2+s4*2*S1*XST+S2;
  M[7]:=M[7]+M20;
  M[8]:=M[8]+s4*YST*M20;
  M[9]:=M[9]+s4*L*X2*XST+s4*3*S1*X2+s4*3*S2*XST+S3;
END;


PROCEDURE BYTE_LEFT(V:BYTE;VAR AANT:INTEGER);

BEGIN
  IF V>15
     THEN IF V>63
                   THEN IF V>127
                              THEN AANT:=8
                              ELSE AANT:=7

                   ELSE IF V>31
                              THEN AANT:=6
                              ELSE AANT:=5

     ELSE IF V>3
                   THEN IF V>7
                              THEN AANT:=4
                              ELSE AANT:=3
                   ELSE IF V>1
                              THEN AANT:=2
                              ELSE AANT:=1
END;


PROCEDURE BYTE_RIGHT(NUM:BYTE;VAR AANT:INTEGER);

VAR TMP:byte;
BEGIN
  TMP:=255-NUM;
  BYTE_LEFT(TMP,AANT);
  AANT:=8-AANT;
END;

PROCEDURE TIMER(VAR HOUR,MIN,SEC,FRAC:INTEGER);
TYPE
 REGPACK = RECORD
            AX,BX,CX,DX,BP,SI,DI,DS,ES,FLAGS:INTEGER;
            END;
            VAR REGS: REGPACK;

BEGIN
```

```
  WITH REGS DO
  BEGIN
   AX:=$2C00;
   MSDOS(REGS);
   HOUR:=HI(CX);
   MIN:=LO(CX);
   SEC:=HI(DX);
   FRAC:=LO(DX);
  END;
   WRITELN(HOUR:3,MIN:3,SEC:3,FRAC:3);
END;


BEGIN
  FOR YST:=0 TO 127 DO
   BEGIN FOR XST:=0 TO 63 DO
          BEGIN P[XST,YST]:=0;
          END;
   END;

  FOR XST:=3 TO 60 DO BEGIN
    FOR YST:=3 TO 125 DO BEGIN
    P[XST,YST]:=255;
    END;
    END;

   for XST:=0 to 9 do begin
   M[XST]:=0; end;



TIMER(A,B,C,D);

 FOR YST:=0 TO 127 DO

    BEGIN X:=0;L:=0;
          WHILE ((P[X,YST]=0) AND (X<64)) X:=X+1;
          BEGIN IF X<64 THEN BYTE_LEFT(P[X,YST],LE);
                             XST:=8*X+8-LE;
                             X:=X+1;
                             L:=L+LE;
                             WHILE ((P[X,YST]=255) AND (X<64)) DO
                             BEGIN L:=L+8;
                                   X:=X+1;
                             END;
                             IF X<64
                               THEN BEGIN BYTE_RIGHT(P[X,YST],LE);
                                          L:=L+LE;
                                    END;
                             Mnm_Update(XST,YST,L,M);
          END;
```

```
    END;
    TIMER(A,B,C,D);
    FOR A:=0 TO 9 DO BEGIN
    WRITELN(M[A]) END;
END.
```

APPENDIX III - LISTING OF THE DELTA METHOD PROGRAM

```
1 REM
2 REM    ┌-------------------------------------------------------------┐
3 REM    |                                                             |
4 REM    |      REAL-TIME CALCULATION  OF THE  MOMENT INVARIANTS       |
5 REM    |      FOR A CONTIGUOUS IMAGE USING THE DELTA  "δ" METHOD      |
6 REM    |                                                             |
7 REM    ├-------------------------------------------------------------┤
8 REM    |                                                             |
9 REM    |               USING IS256 OPTIC RAM (IDETIX)                |
11 REM   |                                                             |
12 REM   ├-------------------------------------------------------------┤
13 REM   |                                                             |
14 REM   |              Copyright (C) MR. MARWAN ZAKARIA               |
15 REM   |            ROBOTIC MECHANICAL SYSTEMS LABORATORY            |
16 REM   |             MCGILL UNIVERSITY, MONTREAL CANADA              |
17 REM   |                                                             |
18 REM   └-------------------------------------------------------------┘
19 REM
29 DEFINT A-Z:Width "LPT1:",132    : REM ALL VARIABLES ARE INTEGERS UNLESS SPECIFIED OTHERWISE
32 CAMPORT=&H260                   : REM  CHANGE THIS IF YOU CHANGE THE  SWITCH
50 KEY OFF
70 DIM APTR(14),CAM(29),PBC(6),FBUF(15),MOVS(5),SAV(8),LOD(8),BITM#(8), MU#(7), ETA#(7), PHI#(7), M#(10),
PRN(3),P(80,80)
72 REM                             : REM APTR = SEGMENT AND OFFSET LIST FOR DRIVER ROUTINE
74 REM                             : REM CAM  = PARAMETER LIST FOR CAMDRIVE - IDETIX H/W DRIVERS
76 REM                             : REM PBC  = PARAMETER LIST FOR DMACALC  - DMA XFER BUFFER SET UP
78 REM                             : REM FBUF = PARAMETER LIST FOR MKBUF  - IMAGE ENHANCE ROUTINES
80 REM                             : REM  MOVS = PARAMETER LIST FOR MOVSCR  - GRAPHICS DISPLAY ROUTINE
305 OPEN "INSTBAS.DAT" AS #1
310 FIELD #1, 128 AS B$
315 GET #1
316 CLOSE
320 K=1: FOR I=1 TO 14
325 APTR(I)=CVI(MID$(B$,K,2))      : REM SET UP SEGMENTS AND OFFSETS
330 K=K+2 : NEXT I
332 DEF SEG=APTR(3)                : REM POINT AT DRIVER CODE SEGMENT
334 PBC(1)=APTR(2)                 : REM DMA SEGMENT = DRIVER DATA SEGMENT
336 PBC(2)=APTR(4)                 : REM DMA OFFSET = BITMAP ADDRESS
338 DMACALC=APTR(6)                : REM DMA XFER BUFFER SETUP ROUTINE
340 CAMDRIVE=APTR(7)               : REM IDETIX DRIVER ROUTINE
342 MKFBUF=APTR(8)                 : REM IMAGE ENHANCE ROUTINE
344 MOVSCR=APTR(9)                 : REM GRAPHICS DISPLAY ROUTINE
346 PBC(3)=16384                   : REM BUFFER SIZE = 128K PIXELS
347 MOVPRN=APTR(14)                : REM MOVE PRINT ROUTINE
348 CALL ABSOLUTE(PBC(1),DMACALC)  : REM DMA SETUP
350 IF PBC(4) THEN SWAP APTR(5),APTR(4) : GOTO 336
400 DEF SEG
```

```
610 CREG =&H1                     : REM COMMAND REGISTER
612 CAM(2)=0                      : REM REFRESH START ADDRESS
614 CAM(3)=255                    : REM REFRESH LAST ADDRESS
616 CAM(4)=300                    : REM ROW START ADDRESS
618 CAM(5)=1                      : REM ROW INCREMENT
620 CAM(6)=379                    : REM ROW LAST ADDRESS
622 CAM(7)=160                    : REM COL START ADDRESS
624 CAM(8)=1                      : REM COL INCREMENT
626 CAM(9)=319                    : REM COL LAST ADDRESS
628 CAM(10)=5                     : REM ST       ROBE COUNT
630 CAM(11)=500                   : REM EXPOSURE TIME ( IN .1 ms FOR LOW RANGE )
631 CAM(12)=&HF                   : REM  NO HEAD MPX
632 CAM(13)=CAMPORT               : REM IDETIX I/O PORT ADDRESS
634 CAM(14)=PBC(6)                : REM  DMA BUFFER POINTER
636 CAM(15)=PBC(5)                : REM  DMA PAGE
638 CAM(16)=0                     : REM CHECK FOR TRANSFER COMPLETE
640 CAM(17)=1                     : REM INITIALIZE THE DMA CONTROLLER
642 CAM(18)=0                     : REM HANG INDICATOR IS ON
644 CAM(19)=1                     : REM PIXEL COUNTER IS ON
650 FBUF(1)=1 : FBUF(2)=1         : REM       PICTYPE 3
651 FBUF(3)=0 : FBUF(4)=0         : REM       PROCESS ENTIRE BUFFER
658 FBUF(10)=APTR(4)              : REM POINTER TO BITMAP - INPUT
660 FBUF(11)=APTR(5)              : REM POINTER TO WORKMAP - OUTPUT
676 MOVS(4)=APTR(5)               : REM POINTER TO WORKMAP = FINAL IMAGE
678 MOVS(5)=0                     : REM SCREEN,START OFFSET
680 PICTYPE=3
682 EXPRNG =0                     : REM  LOW RANGE
684 FMODE  =1                     : REM  SINGLE FRAME
686 PIXC=0                        : REM INITIALIZE PIXEL COUNTER
690 GOSUB 700  : GOTO 1010        : REM RESET THE IDETIX PROCESSORS
700 REM    ------------------------------------------------------
701 REM    |SUBROUTINE TO RESET THE IDETIX PROCESSORS AND/OR STOP THE |
702 REM    |CAMERA (VIA SOFTWARE).                                    |
703 REM    ------------------------------------------------------
710 DEF SEG=0 : I=INP(CAMPORT+5)
711 FOR I=1 TO 1000 : NEXT I      :REM TIME DELAY
713 IF ((INP(CAMPORT) AND 7 ) <> 7 ) THEN GOTO 710
715 DEF SEG = APTR(3)             :REM POINT AT DRIVER CODE SEGMENT
720 RETURN:REM------------------------------------------------------
1010 SCREEN 2                     : REM 640 X 200 GRAPHICS MODE
1015 CLS : STFLG=0
1680 EXPRNG=&H0
1682 CREG=( CREG AND &HFB) OR EXPRNG
1700 FMODE=&H1                    : REM FMODE=&H2 IS CONTINUOUS GRABBING OF IMAGE, =&H1 IS SINGLE IMAGE
GRABBING
1702 CREG   =(CREG   AND &HFC) OR FMODE
3000 GOSUB 700                    : REM-------> TAKE A PICTURE <---------
3001 CLS                          : REM CLEAR THE SCREEN
3010 EF=0
3020 GOSUB 5300                   : REM ENHANCE THE PICTURE
3056 REM LOCATE 25,54: PRINT "    time (ms) =":LOCATE 25,70 : PRINT CAM(11)/10;
```

```
3100 CAM(19)=1 : CAM(1)=0 : CAM(16)=1 :  CALL ABSOLUTE(CAM(1),CAMDRIVE)
3101 GOSUB 6000 : LOCATE 24,2 :PRINT USING "white pixels =######  .  black pixels = ######      %white
pixels = ###.#% ";WPX,BPX,(WPX/(WPX+BPX))*100;
3102 CAM(1)=CREG :CAM(16)=0 :CAM(17)=1 :CALL ABSOLUTE(CAM(1),CAMDRIVE)
3103 FBUF(5)=CAM(23)/2
3104 FBUF(6)=CAM(24)
3110 FBUF(7)=CAM(23)
3120 FBUF(8)=CAM(22)              : REM  PASS INPUT BUFFER DESCRIPTOR TO MKFBUF
3130 FBUF(9)=CAM(24)
3200 CALL ABSOLUTE(FBUF(1),MKFBUF) : REM  ENHANCE THE RAW DATA
3210 MOVS(1)=FBUF(12)
3220 MOVS(2)=FBUF(12)             : IF MOVS(2) > 80 THEN MOVS(2)=80
3230 MOVS(3)=FBUF(13)-1
3240 CALL ABSOLUTE(MOVS(1),MOVSCR) : REM  DISPLAY THE PICTURE
4000 REM                          : REM  SERVICE THE KEYPRESSES
4001 REM                          : REM  IF NO KEY IS PRESSED THEN A CONTINUOUS IMAGE IS GRABBED
4002 REM                          : REM  C = CALCULATE MOMENT INVARIANTS  AND DISPLAY ON SCREEN
4003 REM                          : REM  L = LOOK AT MEMORY CONTENTS BEFORE PROCESSING
4004 REM                          : REM  Q = QUIT THE CURRENT PROGRAM AND PROMPT C>
4005 REM                          : REM  P = PRINT THE IMAGE DISPLAYED ON THE SCREEN
4006 REM                          : REM  S = STOP THE CONTINUOUS LOOP FOR IMAGE GRABBING (PAUSE)
4007 REM                          : REM  R = RESUME, ONLY "R" CANCELS THE "S"COMMAND
4010 KP$=INKEY$ : IF KP$="" THEN GOTO 3100  :REM TAKE A PICTURE CONTINUOUSLY
4015 IF KP$="C" OR KP$="c" THEN :GOSUB 700:GOSUB 45000:REM GOSUB 40000:CLS:SCREEN 2 :GOTO 4100
:REM CALCULATE
4017 IF KP$="L" OR KP$="l" THEN :GOSUB 700:GOSUB 55000:CLS:SCREEN 0,0,0 :GOTO 4100 :REM LOOK AT APTR(5)+C
MEMORY LOCATION
4020 IF KP$="Q" OR KP$="q" THEN :GOSUB 700:CLS :SCREEN 0,0,0:STOP :REM QUIT
4030 IF KP$="P" OR KP$="p" THEN :GOSUB 700:GOSUB 40000          :REM PRINT
4080 IF KP$="S" OR KP$="s" THEN :GOSUB 700 :XX$="":XX$=INKEY$:IF XX$<>"R" AND  XX$<> "r" THEN 4080 :REM
***** STOP THE CAMERA BY PRESSING S, RESUME BY R. *****
4090 GOTO 4010                    : REM GET THE NEXT INPUT
4100 RUN                          : REM START THE PROGRAM OVER AFTER CALCULATING THE PHI'S "φ"
5300 REM     ┌-----------------------------------------------------------┐
5301 REM     |SUBROUTINE TO SET THE PARAMETERS ON FOR THE ENHANCE/LAYOUT  |
5302 REM     |PICTURE.  (PICTYPE 3)                                       |
5303 REM     └-----------------------------------------------------------┘
5310 FBUF(1)=1 : FBUF(2)=1 : MOVS(4)=APTR(5)
5312 IF(((CAM(9)-CAM(7)+1)/CAM(8))*((CAM(6)-CAM(4)+1)/CAM(5)))>32768# THEN EEE$="PIC      TYPE 3 OVER-
FlOW - REDUCE WINDOW SIZE" : EF=1
5313 RETURN:REM-----------------------------------------------------------┘
6000 REM     ┌-----------------------------------------------------------┐
6001 REM     |SUBROUTINE TO CALCULATE THE NUMBER OF WHITE PIXELS, BLACK   |
6002 REM     |PIXELS AND THE PERCENTAGE OF WHITE PIXELS; USING COUNTERS   |
6003 REM     |FROM THE HARDWARE.                                          |
6004 REM     └-----------------------------------------------------------┘
6005 IF CAM(28)<0 THEN CAM(28)=0
6010 IF CAM(28)>&HFF00 THEN CAM(27)=CAM(27)-1
6020 IF CAM(27)<0 THEN CAM(27)=0
6030 IF CAM(25)<0 THEN CAM(25)=0
6040 WPX=(CAM(25)*&H100)+(CAM(26))
```

```
6050 BPX=(CAM(27)*&H100)+(CAM(28))
6100 IF BPX=256 THEN BPX=1:WPX=32767
6150 IF WPX=0 THEN WPX=1:BPX=32767
6200 RETURN:REM------------------------------------------------------------┘
40000 REM   ┌-----------------------------------------------------------┐
40001 REM   |SUBROUTINE TO PRINT THE IMAGE SHOWN ON THE SCREEN ON AN     |
40002 REM   |FX80/85 COMPATIBLE DOT MATRIX PRINTER.                      |
40003 REM   └-----------------------------------------------------------┘
40009 PRN(1)=MOVS(1):PRN(2)=MOVS(2):PRN(3)=MOVS(3):PRN(4)=MOVS(4)
40010 CALL ABSOLUTE(PRN(1),MOVPRN)
40011 RETURN:REM-----------------------------------------------------------┘
45000 REM   ┌-----------------------------------------------------------┐
45001 REM   |SUBROUTINE  CALCULATE 1 THIS SUBROUTINE CALCULATES THE      |
45002 REM   |DELTA "δ", XST, YST AND THE m_pq 'S. (FOR 80 X 640 MATRIX)  |
45003 REM   └-----------------------------------------------------------┘
45004 C=0
45005 DEF SEG=APTR(2)                 : REM POINT AT DRIVER DATA SEGMENT
45006 FOR I=0 TO 9:M#(I)=0:NEXT I     : REM INITIALIZE THESE VARIABLES
45007 REM LPRINT CHR$(27)"!"CHR$(4):WIDTH "LPT1:",160 LETTRIX #15,"10,199,85,C
45009 TIM1$=TIME$:LLY=LY              : REM TAKE THIS TIME AS STARTING TIME1
45010 FOR YST=0 TO 79
45020       L=0:C=YST*80: REM IF (X<80) THEN LPRINT " "
45030       FOR X=0 TO 79
45035              P(X,YST)=PEEK(APTR(5) + C) : C=C+1
45037              REM BYTE$=HEX$(P(X,YST)):LPRINT USING "\\ \\";BYTE$;
45040              IF (P(X,YST) > 0 ) THEN GOTO 45060
45050       NEXT X
45060       IF (X=80) THEN GOTO 45160 ELSE GOSUB 45210
45070       XST=8*X+8-LE
45080       X=X+1
45090       L=L+LE
45100       FOR X1=X TO 79
45105              P(X1,YST)=PEEK(APTR(5) + C) : C=C+1
45107              REM BYTE$=HEX$(P(X1,YST)):LPRINT USING "\\ \\";BYTE$;
45110              IF (P(X1,YST) < 255) AND LLY>5 THEN GOTO 45130 ELSE L=L+8
45120       NEXT X1
45130       LLY=LLY-1:X=X1
45140       IF (X=80) THEN GOTO 45150 ELSE GOSUB 45240 : L=L+LE
45150       GOSUB 45290
45160 NEXT YST
45165 REM TIM2$=TIME$
45170 GOSUB 50215
45171 LINE (1,80)-(639,199),,BF
45172 LOCATE 13,1:FOR I=1 TO 7:PRINT SPACE$(62);:PRINT USING  "phi(#)=##.###^^^^";I;PHI#(I):NEXT I
45173 LOCATE 13,1:FOR I=1 TO 7:PRINT SPACE$(40);:PRINT USING  "eta(#)=##.###^^^^";I;ETA#(I):NEXT I
45174 LOCATE 13,1:FOR I=1 TO 7:PRINT SPACE$(20);:PRINT USING  "mu(#)=##.###^^^^";I ;MU#(I):NEXT I
45175 LOCATE 13,1 :FOR I=0 TO 9:PRINT USING "m(#)=##.###^^^^";I;M#(I):NEXT I
45180 LOCATE 21,35:PRINT USING "TIM1=\\      \\  TIM3=\\       \\";TIM1$,TIM3$
45200 RETURN:REM-----------------------------------------------------------┘
45210 REM   ┌-----------------------------------------------------------┐
45211 REM   |SUBROUTINE  BYTE-LEFT, WHEN THE PROGRAM ENCOUNTERS THE      |
```

```
45212 REM   |FIRST NON-ZERO BYTE ON A GIVEN LINE BYTE-LEFT WILL TELL WHERE|
45213 REM   |THE EDGE EXACTLY IS BY MASKING THAT PARTICULAR BYTE.          |
45214 REM   L-------------------------------------------------------------J
45215 V=P(X,YST)
45220 IF V>15 THEN IF V>63 THEN IF V>127 THEN LE=8 ELSE LE=7 ELSE IF V>31 THEN LE=6 ELSE LE=5 ELSE IF V>3
THEN IF V>7 THEN LE=4 ELSE LE=3 ELSE IF V>1 THEN LE=2 ELSE LE=1
45230 RETURN:REM----------------------------------------------------------J
45240 REM   r-------------------------------------------------------------1
45241 REM   |SUBROUTINE  BYTE-RIGHT, WHEN THE PROGRAM ENCOUNTERS THE       |
45242 REM   |FIRST BYTE < 255 AT THE OTHER SIDE OF THE SURFACE IT USES     |
45243 REM   |BYTE-RIGHT TO TELL WHERE THE RIGHT-HAND EDGE EXACTLY IS.      |
45244 REM   L-------------------------------------------------------------J
45245 NUM=P(X ,YST)
45250 TMP=255-NUM
45260 V=TMP:GOSUB 45220
45270 LE=8-LE
45280 RETURN:REM----------------------------------------------------------J
45290 REM   r-------------------------------------------------------------1
45291 REM   |SUBROUTINE  CALCULATE 2, THIS SUBROUTINE CALCULATES  THE      |
45292 REM   |MU'S "μ(p,q)", ETA'S " ", AND PHI'S "φ" FOR THE GIVEN IMAGE.  |
45294 REM   L-------------------------------------------------------------J
45295 SS#=1:L2#=SS#*L*L
45300 L3#=L2#*L
45310 X2#=SS#*XST*XST
45320 Y2#=SS#*YST*YST
45330 S1#=(L2#-L)/2
45340 S2#=(L3#*2-L2#*3+SS#*L)/6
45350 S3#=(L3#*L-L3#*2+L2#)/4
45360 M#(0)=M#(0)+L
45370 M#(1)=M#(1)+SS#*L*YST:
45380 M#(2)=M#(2)+Y2#*L
45390 M#(3)=M#(3)+Y2#*YST*L
45400 M10#=SS#*L*XST+S1#
45410 M#(4)=M#(4)+M10#
45420 M#(5)=M#(5)+M10#*YST
45430 M#(6)=M#(6)+Y2#*M10#
45440 M20#=X2#*L+S1#*2*XST+S2#
45450 M#(7)=M#(7)+M20#
45460 M#(8)=M#(8)+M20#*YST
45470 M#(9)=M#(9)+X2#*L*XST+S1#*3*X2#+S2#*3*XST+S3#
45480 RETURN:REM----------------------------------------------------------J
50215 DEF SEG=APTR(3)
50220  XX#=M#(4)/M#(0)
50230  YY#=M#(1)/M#(0)
50240  MU#(1)=M#(7)-XX#*M#(4)
50250  MU#(2)=M#(2)-YY#*M#(1)
50260  MU#(3)=M#(5)-YY#*M#(4)
50270  MU#(4)=M#(9)-3*XX#*M#(7)+2*M#(4)*XX#^2
50280  MU#(5)=M#(6)-2*YY#*M#(5)-XX#*M#(2)+2*YY#^2*M#(4)
50290  MU#(6)=M#(8)-2*XX#*M#(5)-YY#*M#(7)+2*XX#^2*M#(1)
50300  MU#(7)=M#(3)-3*YY#*M#(2)+2*YY#^2*M#(1)
```

```
50310  ETA#(1)=MU#(1)/(M#(0)^2)
50320  ETA#(2)=MU#(2)/(M#(0)^2)
50330  ETA#(3)=MU#(3)/(M#(0)^2)
50340  ETA#(4)=MU#(4)/(M#(0)^2.5)
50350  ETA#(5)=MU#(5)/(M#(0)^2.5)
50360  ETA#(6)=MU#(6)/(M#(0)^2.5)
50370  ETA#(7)=MU#(7)/(M#(0)^2.5)
50380  PHI#(1)=ETA#(1)+ETA#(2)
50390  PHI#(2)=(ETA#(1)-ETA#(2))^2+4*ETA#(3)^2
50400  PHI#(3)=(ETA#(4)-3*ETA#(5))^2+(3*ETA#(6)-ETA#(7))^2
50410  PHI#(4)=(ETA#(4)+ETA#(5))^2+(ETA#(6)+ETA#(7))^2
50420  PHI#(5)=(ETA#(4)-3*ETA#(5))*(ETA#(4)+ETA#(5))*((ETA#(4)+ETA#(5))^2-3*(ETA#(6)+ETA#(7))^2)
+(3*ETA#(6)-ETA#(7) )*(ETA#(6)+ETA#(7))*(3*(ETA#(4)+ETA#(5))^2-(ETA#(6)+ETA#(7))^2)
50430  PHI#(6)=(ETA#(1)-ETA#(2))*((ETA#(4)+ETA#(5))^2-(ETA#(6)+ETA#(7))^2) +
4*ETA#(3)*(ETA#(4)+ETA#(5))*(ETA#(6)+ETA#(7))
50440  PHI#(7)=(3*ETA#(6)-ETA#(7))*(ETA#(4)+ETA#(5))*((ETA#(4)+ETA#(5))^2-3*(ETA#(6)+ETA#(7))^2) +
(3*ETA#(5)-ETA#(7))*(ETA#(6)+ETA#(7))*(3*(ETA#(4)+ ETA#(5))^2-(ETA#(6)+ETA#(7))^2)
50450 TIM3$=TIME$
50460 RETURN:REM-----------------------------------------------------------
55000 REM    ┌----------------------------------------------------------┐
55001 REM    |SUBROUTINE LOOK , THIS SUBROUTINE WILL LOOK AT MEMORY     |
55002 REM    |CONTENT STARTING AT LOCATION APTR(5) + 0 AND PRINT IT.    |
55003 REM    └----------------------------------------------------------┘
55004 DEF SEG=APTR(2):C=0
55005 FOR YST=0 TO 79
55010   FOR X=0 TO 79
55020       P(X,YST)=PEEK(APTR(5) + C) : C=C+1
55030       BYTE$=HEX$(P(X,YST)):LPRINT USING "&";BYTE$;
55040   NEXT X
55050 NEXT YST
55060 DEF SEG=APTR(3)
55070 RETURN:REM-----------------------------------------------------------
56000 DEF SEG=APTR(2):LY=0:LX=0
56005 FOR II=40 TO 6320 STEP 80
56010   BYT=PEEK(APTR(5)+II)
56020   IF BYT>0 THEN LY=LY+1
56030 NEXT II
56040 FOR JJ=3200 TO 3280
56050   BYT=PEEK(APTR(5)+JJ)
56053   IF BYT>0 THEN LX=LX+8
56055 NEXT JJ
56060 DEF SEG=APTR(3)
56070 RETURN
57000 FOR I=1 TO 7:FI#(I)=PHI#(I):NEXT I:RETURN
```