

AD-A150 868

FEASIBILITY OF OPTICAL INSTRUMENTS BASED ON  
MULTIAPERTURE OPTICS(U) FLORIDA UNIV GAINESVILLE DEPT  
OF NUCLEAR ENGINEERING SCIENCES J D COX ET AL.

1/1

UNCLASSIFIED

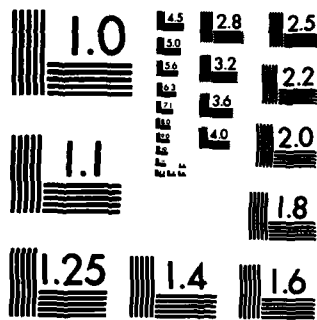
16 OCT 84 AFOSR-TR-85-0111 AFOSR-83-0240

F/G 20/6

NL



					END
					FILMED
					DTIC



MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

Lt Col Carter

2

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

AD-A150 868

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER <b>AFOSR-TR- 85-0111</b>	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Feasibility of Optical Instruments Based on Multiaperature Optics		5. TYPE OF REPORT & PERIOD COVERED Final Report 6/15/83 to 9/30/84
7. AUTHOR(s) John D. Cox Richard T. Schneider Jamil H. Jamed		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of Florida 202 Nuclear Sciences Department Gainesville, Florida 32601		8. CONTRACT OR GRANT NUMBER(s) AFOSR83- 0240
11. CONTROLLING OFFICE NAME AND ADDRESS AFOSR/NE Bolling AFB, D.C. 20332		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102E, 2305, B1
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE 16 October 1984
		13. NUMBER OF PAGES 14
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Allowed for public release. Distribution Unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Low Pixel Numbers Pattern recognition Multiaperature Optics Recognition coefficient. ←		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This paper deals with a recognition system specialized for low pixel numbers. It is assumed that the image consists of 15 x 15 pixels. Using such a low pixel number, the object is distorted beyond recognition if its straight edges are not lined up with the grid. This is especially important if the previously described method based on interpretation of each pixel row as a binary number is used. Therefore, it is shown that a given image of a randomly oriented object can be rotated by a subroutine requiring only little processing time. The resultant image deviates only to a minute degree from the image which would have been over		

DTIC  
ELECTRONIC  
MAR 5 1985  
S  
A

DTIC FILE COPY

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

RE 02 20 072

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

20.

observed with its straight edges lined up with the pixel grid.

*Original: - Supplied key words included: -> front p.*

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
Distribution/	
Availability Codes	
Avail and/or	
Special	
A-1	



Unclassified

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

**AFOSR-TR- 85 - 0 1 1 1**

**Final Report  
for Contract #AFOSR83- 0240**

**on**

**FEASIBILITY OF OPTICAL INSTRUMENTS  
BASED ON MULTIAPERTURE OPTICS**

**by**

**John D. Cox  
Richard T. Schneider**

**and**

**Jamil H. Ahmad**

**October 1, 1984**

**Approved for public release;  
distribution unlimited.**

**Department of Nuclear Engineering Sciences  
University of Florida**

## TABLE OF CONTENTS

	PAGE NO
I. INTRODUCTION	1
II. HARD-WIRED ROTATION	7
A. General Remarks	7
B. Rotation by 45°	7
III. CONCLUSIONS	14
REFERENCES	14

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH (AFSC)  
NOTICE OF TRANSMITTAL TO DTIC  
This technical report has been reviewed and is  
approved for public release IAW AFR 190-12.  
Distribution is unlimited.  
MATTHEW J. KEEPER  
Chief, Technical Information Division

## I. INTRODUCTION

The goal of this research effort has been to address one of the many questions that must be answered to determine if optical instruments are feasible based on the multiaperture optics principle.

Although the scope of this effort is modest, a very important concept is explored. This concept involves the recognition of extended objects in a coarse grid of pixels where the total number of pixels is small (<100). With the constraints imposed by this system, any object has to be described as consisting of squares and/or rectangles defined by the pixels located on an evenly spaced grid. Objects whose edges do not line up parallel or normal to the pixel grid will appear distorted and unrecognizable. In order to mitigate this problem, either the sensing element must be rotated until alignment of the pixel grid and the object's edges is achieved or a rotation has to be achieved through software manipulation of the data. While movement of the sensing element is simple and straightforward, it is not always practical or possible. Therefore, we will explore the viability of software rotation of the object to achieve alignment and recognition. Recognition of objects can be achieved by assignment of binary numbers to the squares and rectangles which make up the object and by comparison with known numbers. Further discussion of this concept is carried out in the next section.

It is the goal of this research effort to examine just one of the many problems and constraints imposed by a multiaperture optical system employing microprocessor based detection and recognition hardware and software, namely a low pixel number pattern recognition scheme with software generated rotation ability.

Multiaperture Optics, the electro-mechanical analogon of the insect eye, offer a large field of view albeit under reduced resolving power<sup>1</sup>.

The biggest advantage of vision systems based on multiaperture optics is their small physical size and the possibility to perform the computation necessary for image evaluation and pattern recognition in the eye itself.

The consequence of this is that the object to be recognized will be described by only a few pixels. Specialization of a system to the case of low pixel number pattern recognition allows reducing of the necessary computations to a manageable level so that, indeed, using VLSI technology, image processing in the eye itself becomes possible.

A pattern recognition scheme using only rectangles and squares as elements of the picture language was described in an earlier paper<sup>2</sup>. The scheme assumes that all shapes can be broken down into rectangles or squares. If the rectangle extends in the x-direction it is considered an x-feature and given the code 01, a y-feature is given 10, and a square would be an x-y feature with the code 11; all this, regardless of the size of the rectangles. The sequence of rectangles which makes up the image, therefore, is represented by a recognition number which is a sequence containing the appropriate codes as defined above. This recognition number is stored in memory. If an unknown object is encountered, its recognition number is computed with a simple program. Then the obtained number is looked up in the memory and the shape so recognized. Therefore, this system will only find things it is supposed to find and will have no problem ignoring anything else. The amount of data collected is optically limited instead of electronically sorted out. A special problem, which is posed by using low pixel numbers, is created by the straight lines of the object. Since the picture language consists only of rectangles and squares, the straight lines are very important for formation of the language elements. Of course, if the straight lines of an object are misaligned with the detector grid, distortions occur. If low pixel numbers



are used for the description of the object, these distortions become so severe that recognition of the object becomes impossible.

For this reason, it is necessary to rotate any image obtained, through various sets of degrees (e.g. 15°, 30° and 45°) of a quadrant until a recognizable image of the unknown object is obtained.

This could be accomplished by rotating the eye mechanically. As a matter of fact, indications are that insects, which also use low pixel numbers and therefore have necessarily the same problems, actually do this by aligning the horizontal band on their eyes with the horizon or their vertical bands with vertical edges, e.g., trees. For a mechanical device physical rotation of the eye is not very practical, therefore, the rotation of the image inside the eye is preferable.

This means for each image obtained, one would ask the question, how this image of an unknown object would look if it had been obtained with the object having a different degree of orientation in respect to the detector grid? There will be one position where the object becomes recognizable and this is when at least some of the straight lines of the object are lined up with the detector grid.

The rotation of the image can be accomplished either by hard wiring or by software.

To explain how rotation by hard wiring could be accomplished, Figure 1 shows how a multiaperture optics system may be configured. The optical elements may be nonimaging concentrators (light horns), the field of view (FOV) of which may be controlled by a field lens. The FOV of each facet is observed by one fiber shaped detector, so that the contents of the FOV of an individual facet is represented by one pixel only. The FOV's of neighboring facets are adjacent and do not overlap significantly. An object, if large

Angle Between Optical Axes  
to be chosen to achieve desired  
overall FOV

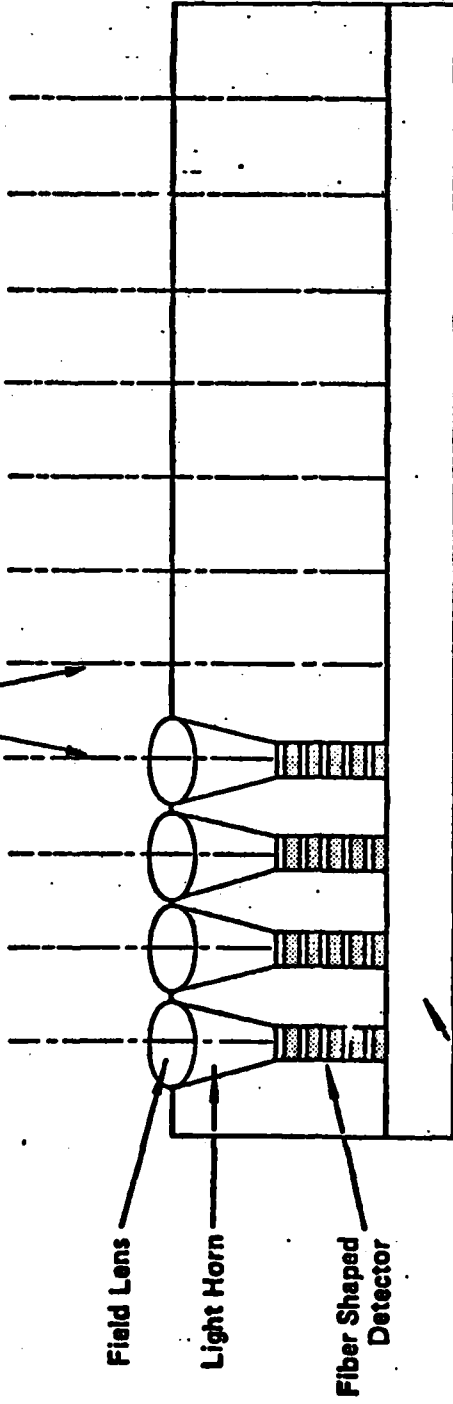


Figure 1 Multiaperture optics configuration

enough, is seen by  $n$  facets, and is therefore represented by  $n$  pixels. Below the layer containing the detectors, a layer of memory cells is located into which the information the detectors received is loaded, maybe after some thresholding and clipping was performed. The sum of the memory cells make up a computer core. Therefore the image resides now as a binary image in a computer core having an organization identical to the detector grid. A rotation can now be accomplished by either software or hard wiring. Reference 2 treats a  $45^\circ$  rotation using a matrix. As it turns out there is no straightforward correspondence between an origin pixel and a destination pixel. It is necessary to divide origin as well as destination pixels into several subpixels and have rules as to how many subpixels in a destination pixel have to be occupied in order to consider the destination pixel occupied. The rules are involved and therefore require additional computing capacity. Such a rotation is best done by software.

In contrast to this (in this paper) the possibility of performing rotations by hard wiring is explored. It is, in principle, possible to locate three more computer cores underneath the one discussed above which could be physically rotated in respect to each other. Or if not physically rotated, still hard wired connections, connecting an address in one core to a corresponding address in the next core, could exist. The corresponding addresses may be, e.g., for  $15^\circ$ ,  $30^\circ$  and  $45^\circ$  rotation. It is still true that there is no straightforward, one-to-one correspondence between locations. However, as will be shown in this paper, only a very simple software procedure will be necessary to obtain a satisfactory image after hard wired rotation has been performed.

1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9	1,10	1,11	1,12	1,13	1,14	1,15
2,1	2,2	2,3	2,4	2,5	2,6	2,7	2,8	2,9	2,10	2,11	2,12	2,13	2,14	2,15
3,1	3,2	3,3	3,4	3,5	3,6	3,7	3,8	3,9	3,10	3,11	3,12	3,13	3,14	3,15
4,1	4,2	4,3	4,4	4,5	4,6	4,7	4,8	4,9	4,10	4,11	4,12	4,13	4,14	4,15
5,1	5,2	5,3	5,4	5,5	5,6	5,7	5,8	5,9	5,10	5,11	5,12	5,13	5,14	5,15
6,1	6,2	6,3	6,4	6,5	6,6	6,7	6,8	6,9	6,10	6,11	6,12	6,13	6,14	6,15
7,1	7,2	7,3	7,4	7,5	7,6	7,7	7,8	7,9	7,10	7,11	7,12	7,13	7,14	7,15
8,1	8,2	8,3	8,4	8,5	8,6	8,7	8,8	8,9	8,10	8,11	8,12	8,13	8,14	8,15
9,1	9,2	9,3	9,4	9,5	9,6	9,7	9,8	9,9	9,10	9,11	9,12	9,13	9,14	9,15
10,1	10,2	10,3	10,4	10,5	10,6	10,7	10,8	10,9	10,10	10,11	10,12	10,13	10,14	10,15
11,1	11,2	11,3	11,4	11,5	11,6	11,7	11,8	11,9	11,10	11,11	11,12	11,13	11,14	11,15
12,1	12,2	12,3	12,4	12,5	12,6	12,7	12,8	12,9	12,10	12,11	12,12	12,13	12,14	12,15
13,1	13,2	13,3	13,4	13,5	13,6	13,7	13,8	13,9	13,10	13,11	13,12	13,13	13,14	13,15
14,1	14,2	14,3	14,4	14,5	14,6	14,7	14,8	14,9	14,10	14,11	14,12	14,13	14,14	14,15
15,1	15,2	15,3	15,4	15,5	15,6	15,7	15,8	15,9	15,10	15,11	15,12	15,13	15,14	15,15

Fig. 2

Pixel Numbering Scheme

## II. HARD-WIRED ROTATION

### A. General Remarks

In the following a grid of 15 x 15 pixels is assumed to be only a part of the overall field of view, yet large enough to contain one object, which is supposed to be recognized. It was felt that 15 x 15 is the smallest size to be useful. Larger fields, of course, experience less distortion, therefore, it is necessary to show that the smallest practical field size can be handled by the proposed rotation scheme. It was also assumed that a-priori knowledge exists as to what objects can be expected to be encountered. Only then is the size 15 x 15 of use.

### B. Rotation by 45°

Figure 2 shows a 15 x 15 grid and the numbering system used. The number before the comma refers to the row while the number after the comma refers to the column. The 45° rotation is accomplished by connecting certain pixels (r,c) (r:row, c:column) in one memory, the origin memory "A" to certain pixels to another memory, the destination memory "B." The pixel (8,8) is the center pixel of the rotation, therefore A(8.8) is connected to B(8.8).

To find the necessary other connections it has to be realized that the image is distorted to begin with. If the object was indeed observed in a 45° position, then the direction which would be the x-direction on the object, if it were lined up with the grid, is observed too long by a factor of  $\sqrt{2}$ . For this reason, the obtained (rotated) dimension needs to be shortened by a factor of  $\sqrt{2}$ . Since one deals with a binary image one can do this only by occasionally removing or moving certain pixels, at a strategic location.

For ease of illustrations the connections between the memories "A" and "B" are made in steps. Step No. 1: The main diagonal line A(1,1) through

A(15,15) is connected to center column of B, which is column 8, as indicated in Figure 2. Next the neighboring diagonals A(1,2) through A(14,15) are connected to the neighboring rows, in a way that A(1,2) connects to B(2,9) and A(1,3) to B(3,10), etc.

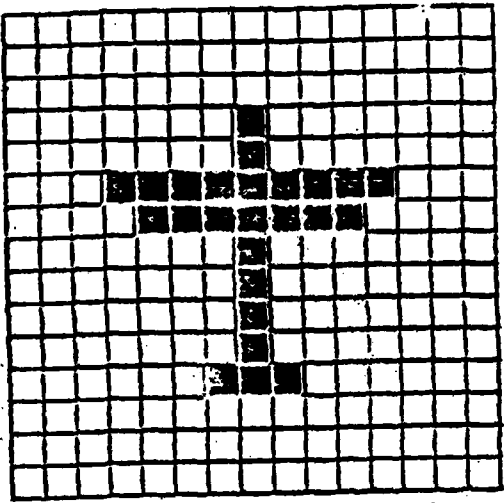
To illustrate the consequences of the different steps of such connections an example is given in Figure 3. The object is depicted in Figure 3a, while Figure 3b shows how it is observed by a 15 x 15 multiaperture optics system if the straight edges of the object are inclined by 45° to the detector grid. The result of the step 1 connections can be seen in Figure 3c. As can be seen, the image is still highly distorted, and some modifications of the connections have to be made. The next step therefore should be to correct for the  $\sqrt{2}$  distortion introduced by the diagonals of the square pixels.

Step No. 2: Shift Columns B(11) through B(15) and B(1) through B(5) up by one unit and remove all pixels from columns B(>13) and B(<3).

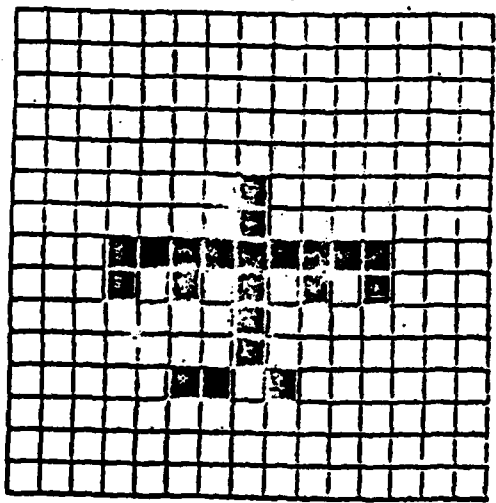
The result of step 2 is depicted in Figure 3d. As can be seen by comparing Figure 3d to Figure 3a, the image is too short in the y-direction. This problem is solved in the next step.

Step No. 3: Insert an empty row between row 10 and row 11, in a way that the old row 11 becomes row 12. Also move the contents of the pixel blocks B(12 through 15, 5 and 6) as well as B(12 through 15, 10 and 11) one unit toward the center. If two occupied pixels should fall on top of each other, the affected pixel is considered just as one occupied pixel. Figure 3a shows the result of step no. 3.

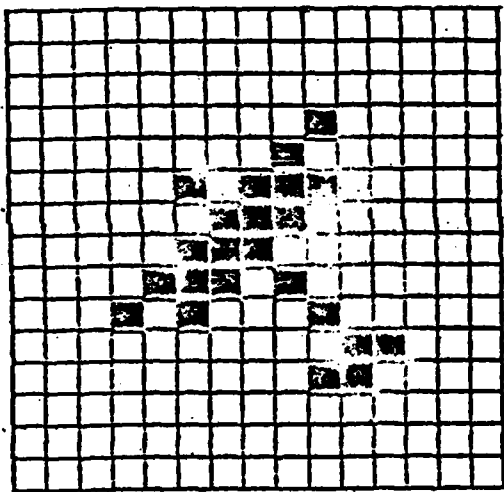
This concludes the hard wired rotation. However the obtained result still needs some improvement. This has to be done by software since the actions to be taken depend on the shape of the observed object. The required procedure however is very simple. The program goes through all rows and



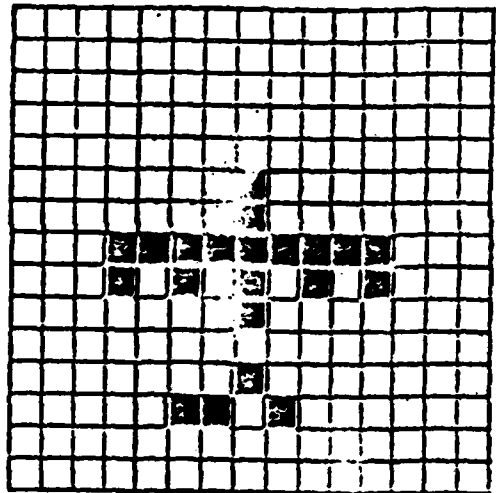
(a)



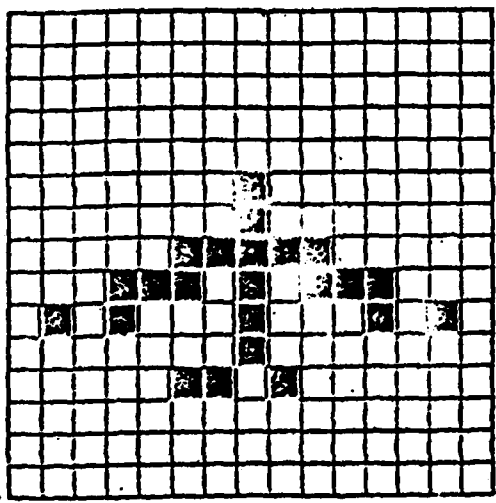
(d)



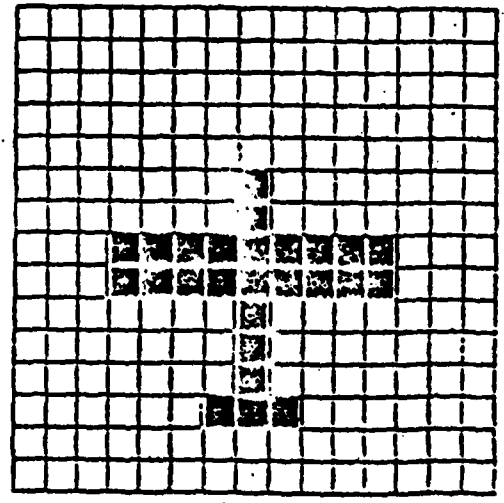
(b)



(e)



(c)



(f)

Fig. 3 45° Rotation

inserts a binary "1" into all pixels containing a "0" and having a neighbor containing a "1" on each side. The same is done for columns. The result of this procedure is shown in Figure 3f.

As can be seen, the obtained image corresponds well to the original, so well that the recognition routine of Reference 1 will recognize it.

Figure 4 shows the 45° rotation of two more shapes. Parts (a) and (aa) of Figure 4 shows the original shapes while parts (b) and (bb) show them as seen by the system if some of the straight edges are inclined by 45° in respect to the grid. The part (c) and (cc) finally show the result of the rotation. Again the results are very close to the original.

In general it can be said that if the object is lying in a 45° position, most of the object pixels are, in fact, a part of the 45° axis or at least of some line parallel to it. This fact favors the hard wire rotation. In the other cases, where the object is lying in a position different from 45°, the object pixels no longer form a part of a relative axis. Here a software rotation<sup>3,4,5</sup> or a software controlled hard wire rotation is required. Again we approach the destination grid in a number of sequential steps.

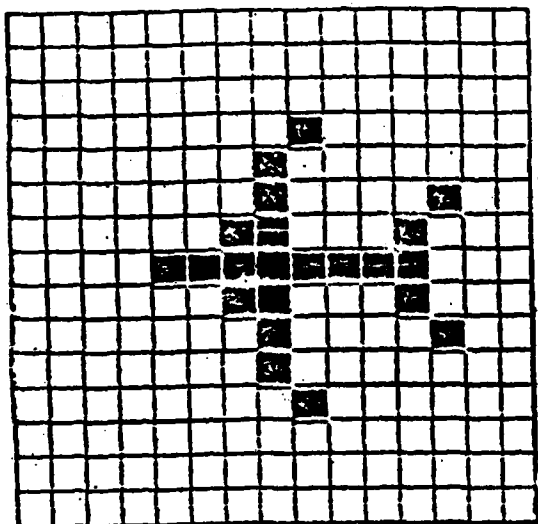
#### Rotation by 30°:

Figure 5(a) is the object lying at 0° position. Figure 5(b) is the same object viewed as if lying at a 30° position. We notice the object shape is less distorted as compared to the previous case (45°), so the x-direction reduction factor has to be smaller. The object size is actually shortened by a factor of  $2/\sqrt{3}$  in the x-direction. Again, we assume the pixel (8,8) as the center pixel or rotation.

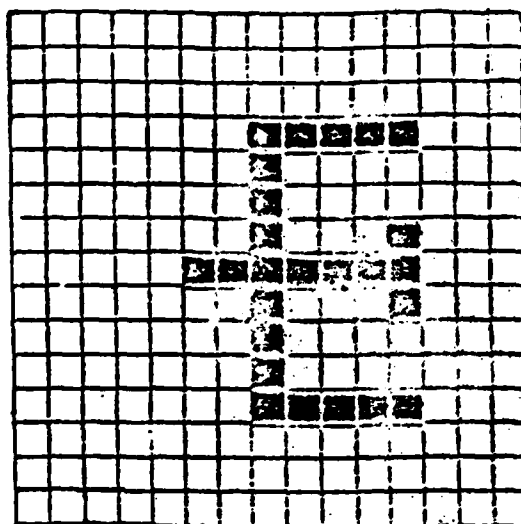
STEP 1: RIGHT SHIFT the pixels of rows 10 through 15 by two units and row 9 by one unit.

: UP SHIFT the contents of columns 10 through 15 by two units and columns 9 by one unit.

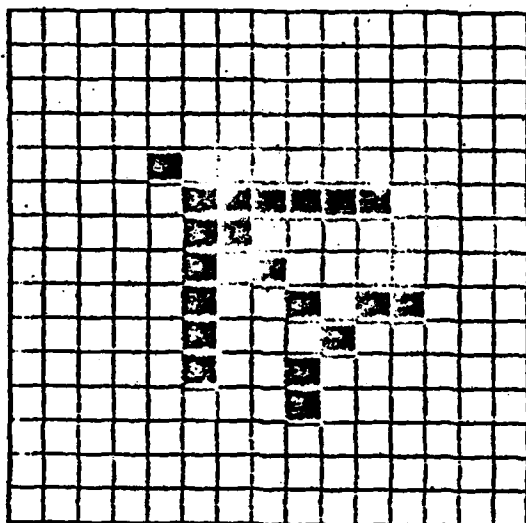




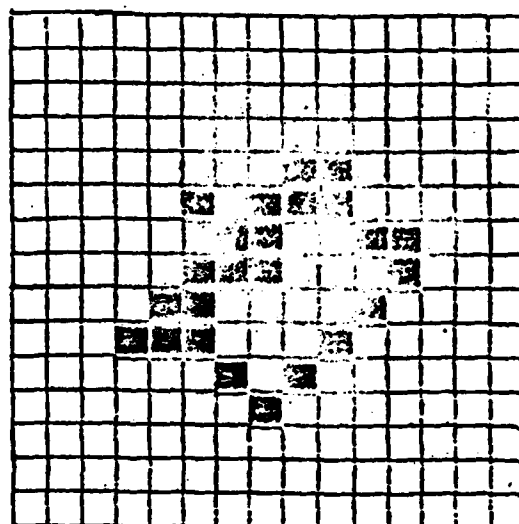
after 45° rotation  
(c)



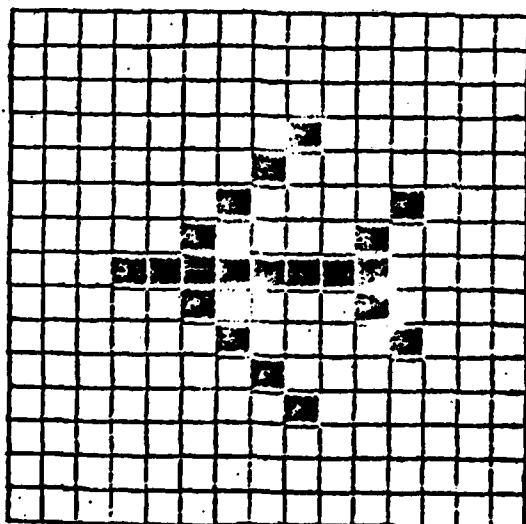
after 45° rotation  
(cc)



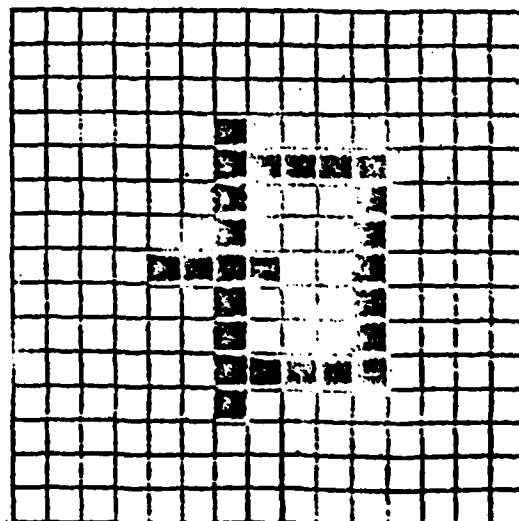
viewed under 45°  
(b)



viewed under 45°  
(bb)

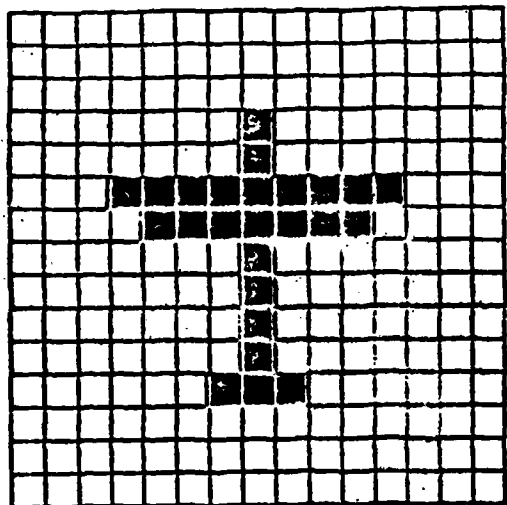


object at 0°  
(a)

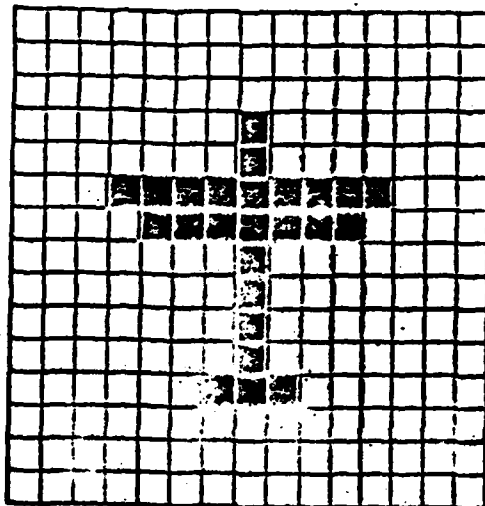


object at 0°  
(aa)

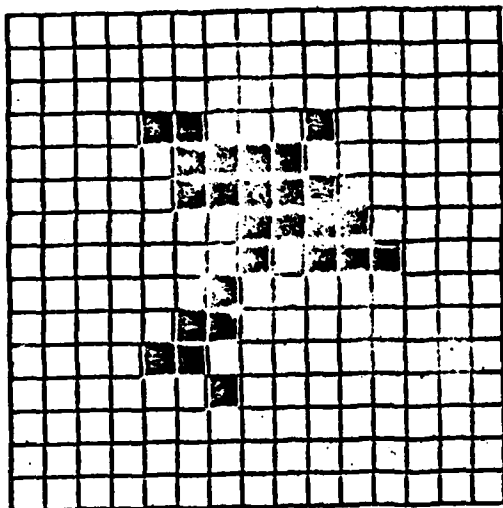
Fig. 4



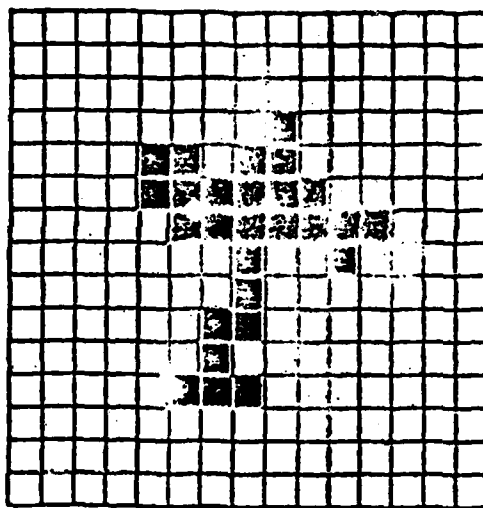
(a)



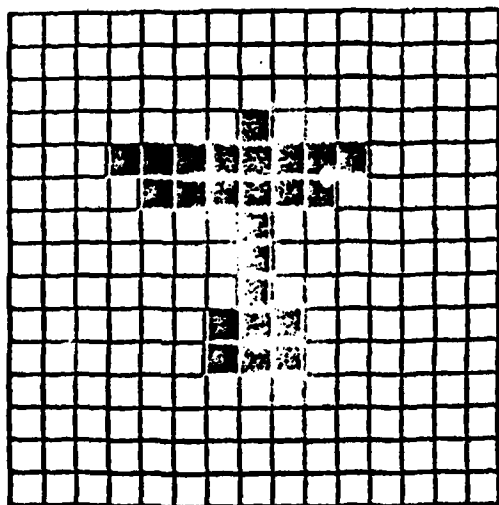
(a)



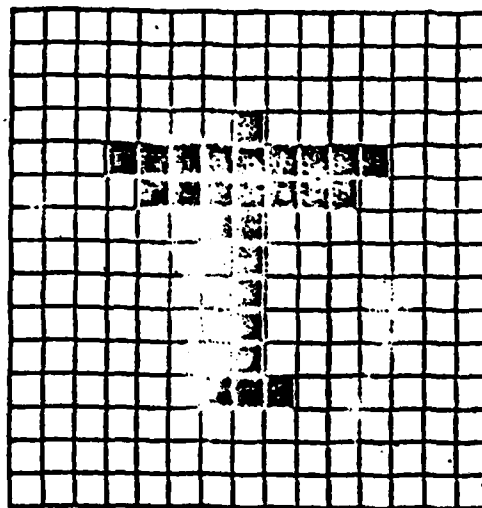
(b)



(b)



(c)



(c)

- : DOWN SHIFT the contents 1 through 6 by one unit.
- : LEFT SHIFT the contents of rows 1 through 4 by two units and row 5 by one unit. DOWN SHIFT the contents of rows 1 through 3 by one unit.

The result of this sequential shifting aligns the object vertical lines to the grid lines but gives a distorted picture due to undesirable pixels at certain locations.

STEP 2: Remove pixels in block size 2 x 1 on both sides of center pixel. The result is a less distorted picture.

STEP 3: This is the same kind of improvement as performed in the 45° rotation case. The program goes through all rows and inserts a binary "1" into all object pixels containing a '0' and having a neighbor containing a '1' on each side. The same is done for columns. The result of this procedure is Figure 5(c).

#### Rotation by 15°:

Again, we assume the pixel (8,8) is the center pixel of rotation. The reduction factor in this case is  $\sqrt{2}(\sqrt{3}-1)$  so we have to shorten the object in x-direction by this factor. Figure 6(b) is the object viewed as if it were lying at 15° position.

#### STEP 1: SHIFT:

LEFT SHIFT the contents of rows 1 through 5 by one unit.

UP SHIFT the contents of columns 9 through 15 by one unit.

RIGHT SHIFT the contents of rows 10 through 15 by one unit.

UP SHIFT the contents of blocks (rows 9 through 15 and columns 8 through 15) and block (row 1 through 9 and column 11 through 15) by one unit.

STEP 2: Remove pixels in block size 2 x 2 on each vertical side of the center pixel. By doing this, most of the distortion is reduced.

STEP 3: Again, for improvement the program goes through all rows and inserts a binary "1" into all object pixels containing a '0' and having a neighbor containing a '1' on each side. The same is done for the columns.

The result of this whole procedure is Figure 6(c).

### III. CONCLUSIONS

The described multiaperture optics system deals with low pixel numbers. This allows handling of the collected information with a minimum of computation. The use of low pixel numbers results in distortion of straight lines of the object. Since these lines would no longer be parallel with the detector grid, they appear distorted and recognition of the object becomes difficult or impossible. It was shown in this paper that this problem can be solved by rotation of the observed image before recognition is attempted.

### REFERENCES

1. "Mechanical Model of the Insect Eye," Richard T. Schneider and James F. Long. SPIE Proceedings Volume 359, 1-10, 1982.
2. "Pattern Recognition with Multiaperture Optics System," Richard T. Schneider and James F. Long, SPIE, San Diego 1983.
3. "Digital Image Processing," Rafael C. Gonzales, Paul Wintz. Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1977.
4. "Fundamentals of Interactive Computer Graphics," J.D. Foley and A. Van Dorn. Addison-Wesley Publishing Company, Reading, Massachusetts, 1982.
5. "Mathematical Elements for Computer Graphics," Roger D. F., Adams J.A., McGraw Hill, New York, 1976.

**END**

**FILMED**

**4-85**

**DTIC**