STUDIES IN STEROSCOPY

BY

JOHN RANDOLPH DISBROW

S. B., MASSACHUSETTS INSTITUTE OF TECHNOLOGY (1962)

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE at the MASSACHUSETTS INSTITUTE OF TECHNOLOGY JUNE, 1964

Signature of Author Signature redacted
Department of Electrical Engineering, May 22, 1964

Certified by Signature redacted
Thesis Supervisor

Accepted by Signature redacted
Chairman, Departmental Committee on Graduate Students
STUDIES IN STEREOSCOPY

BY

JOHN RANDOLPH DISBROW

Submitted to the Department of Electrical Engineering on May 22, 1964 in partial fulfillment of the requirements for the degree of Master of Science

ABSTRACT

Four experiments are performed with computer-generated stereo images to investigate the data processing model for binocular depth perception. In each of these four experiments Julesz's "Difference Field" data processing model is found to make erroneous predictions. A new data processing model is developed from the experimental results. The data processor of this new model treats a stereo pair as follows: The pictures are fused. The zero order difference field is computed and the area of agreement masked in each field. Higher order difference fields are computed for the unmasked areas, with new masking occurring. From this processing planes become established. Each plane is then locally generalized.

Thesis Supervisor: William F. Schreiber
Title: Associate Professor of Electrical Engineering
ACKNOWLEDGEMENT

This work was done in the Cognitive Information Processing Group of the Research Laboratory of Electronics. In addition the facilities of the MIT Computation Center were used.
# TABLE OF CONTENTS

## I BACKGROUND

1. Depth Perception .................................................. 5
2. Isolating Binocular Parallax ................................. 7
3. Making a Pair of Computer-Generated-Random-Patterns .. 8
4. Viewing Stereo Pairs ............................................. 10

## II THEORY

1. Binocular Parallax, a Small Part of Seeing ................. 13
2. Binocular Data Processing ..................................... 15
3. Difference Matrix .............................................. 16
4. A Stereo Pair and Its Difference Fields ...................... 17
5. Simple Difference Field Processing .......................... 19
6. Sequential Difference Field Processing .................... 20

## III EXPERIMENTAL RESULTS

1. Compare Sequential to Simple Difference Field Processing - First Experiment .......... 22
2. Another Experimental Test of the Sequential Difference Field Model .................. 25
3. Experimental Evidence for Local Generalization ............. 27
4. Investigation of Local Generalization ........................ 31

## IV CONCLUSION ...................................................... 36

BIBLIOGRAPHY ...................................................... 37
FRESNEL LENSES ................................................... 38
APPENDIX ............................................................. 39
I BACKGROUND

The material of Part I provides some of the background necessary for understanding this paper. In section 1 the monocular and binocular cues for depth perception are discussed. The idea of binocular parallax, one of the binocular depth cues, is introduced. In section 2 computer-generated-random-pattern stereo pairs are seen as an advance over the previous tool for studying depth perception from binocular parallax. With the aid of an illustrative figure, section 3 shows how a typical pair of computer-generated-random-patterns is made. An additional figure shows how such a stereo pair might come to be interpreted as a plane floating in front of a background. Section 4 explains the operation of a simple stereo viewer. Section 5 lists a few facts about stereo pairs that are well known to those familiar with the references in the bibliography.

1. Depth Perception

Depth perception is the product of complex mental processes. The input to these processes are divided into two classes: monocular cues and binocular cues.

Monocular cues are qualities or properties of the perceived object that can be seen with only one eye. An example of a monocular cue is the convergence to a point of a pair of rail road tracks in an artist's perspective-line-drawing. Typically such drawings are viewed with both eyes. However
the depth cue, the convergence of the two tracks, can be seen with just one eye. If any cue can be seen with just one eye, it is labeled a monocular cue.

Binocular depth cues are qualities of the perceived object that can only be seen with two eyes. For example consider binocular parallax. This is related to the familiar fact that what one sees out of one eye is slightly different than what one sees out of the other eye. Looking through a door into another room, each eye can see a little of this room that the other eye cannot. Depth in this case is said to result from the difference between the two retinal images. Both eyes are required for the perception of such cues.

The following list of monocular and binocular depth cues is taken from Reference 1.

Binocular Depth Cues

Binocular parallax
Convergence of eyes
Correlative accommodation (focusing)

Monocular Depth Cues

Linear perspective (such as converging railroad tracks)
Apparent size of objects of known size (which decreases with distance of observer)
Monocular parallax (change of appearance with change of observer's position)
Shadow patterns (the light-and-shade relation yeilding relief)
Interposition (the superposing of near objects on far objects)
Changes due to atmospheric conditions (such as hase, blurring of outlines)
Accommodation (focusing on an object with one eye)
Retinal gradient of texture (decreasing size of texture elements with distance)
Retinal gradient of size of similar objects (rate of decrease of houses, fence posts, telegraph poles, etc.)
This paper is concerned with the principal binocular depth cue: Binocular Parallax.

2. Isolating Binocular Parallax

Figure 1: Line drawing stereo pair

Investigators studying depth perception have usually used stimuli containing both monocular and binocular cues. Often they do not want to mix cues, but can think of no way to make stimuli containing only one cue.

For years investigators have used line drawings, such as Figure 1, to study binocular parallax. The depth perceived when viewing Figure 1 through a stereoscope is thought to result from binocular parallax. However in such drawings the shape and location of the object to be perceived in depth is defined monocularly. Thus it is not certain that line drawing studies examine binocular parallax alone.

In 1960 Bella Julesz presented a new tool for studying depth perception. The essence of this tool is a stereo pair
in which each picture is a computer-generated-random-pattern. An example of such a pair is seen in Figure 6. When this figure is viewed stereoscopically, certain correlated point domains are seen in depth. The perception is again of a rectangle above a background. However its shape and location are no longer visible monocularly. Thus it is believed that binocular parallax has been isolated.

3. Making a Pair of Computer-Generated-Random-Patterns

Each of the computer-generated-random-patterns of Figure 6 is made in two steps. First the IBM 7094 computer generates a matrix of random numbers which it writes on tape. Then this tape is fed into a machine which converts a matrix of numbers on tape into a matrix of gray levels on film. The details of this operation are discussed in the Appendix.

The two pictures in a pair are very similar. For example the two pictures in Figure 6 are identical everywhere except in the small region where a floating rectangle is perceived. In both pictures there is a certain array of two columns by sixteen rows which in one picture has been shifted horizontally by two columns. (The resulting two column vacancy has been filled with new random squares.) This shift results in the impression of depth. Figure 2 illustrates the make up of such a stereo pair.
Shifted rectangle is in capital letters for easy identification.

**Figure 2:** Illustration of how $2 \times 3$ shifted array is made.

By carefully examining Figure 3, one can see how such a shifted square comes to be perceived as floating in front of a background. Note how each separate eye sees the plane against a different part of the background.

**Figure 3:** Illustration of difference between left and right eye views.
4. Viewing Stereo Pairs

The simplest stereoscope employs two lenses. The left-hand picture is placed at the focal point of the left lens. The right-hand picture is placed at the focal point of the right lens. On the other side of the lenses the rays from each picture are parallel. A viewer may place himself so that his left eye intercepts rays from the left-hand picture; and his right eye, rays from the right-hand picture. Since both sets of rays are parallel, the viewer will interpret them as coming from the same place. As a result he will perceive a single, fused picture.

Such a pair of lenses is provided with this paper. Unlike the ground glass lenses in one's eyeglasses, these are flat, plastic, Fresnel Lenses. The lenses, and details as to their use, are to be found just before the Appendix.

5. Facts About Computer-Generated-Random-Stereo-Pairs

If Figure 6 is turned upside down and viewed stereoscopically a plane again floats in front of the background. It is easy to show that any of these stereo pairs can be viewed the way they are given or upside down.

If the left-hand and right-hand images of Figure 6 are transposed, then a plane will be found to float behind the background in a rectangular well. Consider the example of Figure 4. The left and center pictures together form a stereo pair. In that pair a pyramid of planes is seen rising above
Figure 4

$f_L$ — Pyramid — $f_R$

$f_L$ — Pyramid — $f_R$

Inverted
the background. The picture on the extreme right is identical to the one on the extreme left. Thus the center picture and the right picture form a transposed pair. In this pair a pyramid is seen below the background.

Many kinds of photographic imperfections do not impair the perception of stereo. If the rows and columns in both pictures of a pair are bent, the stereo does not suffer. If one of the pictures is a little lighter or a little larger than the other, stereo does not suffer. For example depth is perceived in Figure 4 even though these pictures are bent and blackened.
II THEORY

Part II provides all the theory that must be understood if the experimental results are to be meaningful. Section 1 provides perspective. A problem about how binocular parallax effects depth perception is seen to be a small part of the overall problem of how a man sees. Section 2 introduces the idea that binocular parallax effects can be explained by postulating the existence of a simple data processing device between the retinas and the conscious sensation of depth. In order to understand how this data processor works, section 3 introduces the idea of the difference matrix. Section 4 explains how a stereo pair is made up. With Figure 6 as the example it is shown how the difference matrix idea can be used to find the difference fields. Section 5 outlines Julesz's data processing model - the heart of which is the difference field idea. Section 6 outlines the first part of the data processing model developed in this paper.

1. Binocular Parallax, a Small Part of Seeing

The present day conception of seeing may be divided into three steps. Light rays from an object pass through a lens and form an image on the retina. This causes electrical impulses to travel down the optic nerves. These impulses somehow give rise to the conscious sensation of sight.
Understanding of this sequence of steps decreases as one proceeds from object to sensation. The concept of the image being formed on the retina was discovered early in the seventeenth century; this part of seeing is now well understood. The idea of electrical impulses in nerve fibers is now being extensively studied and can be termed somewhat understood. The idea that these electrical impulses give rise to the conscious sensation of sight is very poorly understood.

If one treats image formation on the retina as fully understood, there remains a very large variety of questions in which an investigator of vision might become interested. One of the many questions is: How does the presence of slightly different images on each of a man's two retinas become translated into the perception of depth? This is one of the classic vision problems. It might be labeled the question of the role of binocular parallax in depth perception.

In his 1960 paper "Binocular Depth Perception of Computer-Generated Patterns", Bella Julesz deals with the question of the role of binocular parallax in depth perception. He treats the phenomena of image formation on the retina as understood. The other two steps in vision are lumped together. Nerve impulses are not discussed. Rather the processing between retina and conscious sensation is studied. This processing is considered to be data processing and a model for that data processing is developed.
The subject of this paper is the data processing model by which the presence of slightly different images on each of a man's retinas is translated into the perception of depth. The Julesz paper is the starting point of this work.

2 Binocular Data Processing

In his 1960 paper Bella Julesz forward various possible data processing models to explain the perception of depth from binocular parallax. With respect to pair of his random patterns, he concludes that the two monocular fields are fused into a binocular field before any processing occurs. He develops a model for the subsequent binocular data processing. He calls his model the "difference field" model.

The work reported here is concerned with this binocular field processing. Since the results of some experiments in this work are not explained by the difference field model, a new model is postulated.

In order to understand the results of this work it is necessary that the idea of the difference field be understood and that the distinctions between the difference field model, and sequential difference field model be seen. So as to make it easier to determine which difference field model is being referred to, Julesz's difference field model will be referred to as the simple difference field model. In order to facilitate the initial understanding of, and subsequent talking about difference fields, the idea of the difference matrix is developed in the next section.
3. Difference Matrix

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

lower matrix

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

difference matrix for zero shift, $D_0$.

For shift of 1, $D_1$

Figure 5

Consider two identical matrices each of whose elements is randomly selected to be a 0 or 1. Place one matrix exactly over the other. Now above each element in the lower matrix is an element of the upper matrix. Make up a new matrix, the difference matrix, in the following manner. Look at the element in row 1 column 1 of the lower matrix. Subtract that element (0 or 1) from the element just over it in the upper matrix. Enter the result in row 1 column 1 of the difference matrix. Continue to subtract each element from the one over it until the difference matrix, $D_0$, is completed. See Figure 5 for an example.

Instead of placing the two matrices on top of each other, they could be placed over each other but displaced by one column. A difference matrix taken now would be denoted $D_1$. 
where the 1 stands for the one column shift. Such a \( D_1 \), for which the upper matrix has been shifted to the right by one column, is also given in Figure 5.

Compare the two difference matrices \( D_0 \) and \( D_1 \). When the upper and lower matrices completely correlate there results a uniform difference matrix \( D_0 \). When the two matrices do not correlate, the difference matrix has a random appearance. Half the elements of \( D_1 \) can be expected to be 0, and the other half -1.

4. A Stereo Pair and Its Difference Fields

Now consider picture pairs. Each picture is made of black, dark grey, light grey, or white squares (elements). The left hand picture, field left (\( f_L \)), is a 64 x 64 array of squares of random shades. The right hand picture, field right (\( f_R \)), is made up in three steps. A central 2 x 16 area of \( f_L \) is placed in \( f_R \), but shifted to the left by the width of two columns (Displacement=\( \Delta = 2 \)). In \( f_R \) the two columns to the right of the 2 x 16 array are filled with new squares of random shades. The remaining vacancies in \( f_R \) are filled with the same pattern of squares as is found in the corresponding locations in \( f_L \). The result is a picture pair similar to those first used by Julesz. If viewed stereoscopically there is a 2 x 16 rectangular array floating in front of the background. See Figure 6 for such a stereo pair and for its composition.
LEGEND

For Figures: 6, 7, 8, 9, 10

Composition

IDENT = Areas in which the random squares of \( f_L \) and \( f_R \) are identical.

SHIFT = Shifted region. Each region is made up of the identical random squares. However the regions are located in different places in the two fields.

DIFF = Areas of random squares appearing in only one field. The "DIFF" region in \( f_L \) has different random squares then the "DIFF" region in \( f_R \).

Simple Difference Fields

MATCH = Uniform area corresponding to an array of 0's in the difference matrix (area where the two fields match).

RAND = Area corresponding to a random array of \( \pm 3, \pm 2, \pm 1, 0 \) in the difference matrix.

Sequential Difference Fields

MATCH = As above

RAND = As above

IGNORED = Areas ignored by the processor.
Composition of Stereo Pair

Simple Difference Fields

Sequential Difference Fields

Figure 6 (see page 18 for legend)
If 0 denoted white, 1 denoted light grey, 2 denoted dark grey, and 3 denoted black, these pictures could be transformed into matrices. The difference matrices \((D_0, D_1, D_2, \text{etc.})\) could be found. Then each difference matrix could be converted back into a picture. Such a picture will be called a difference field and be denoted by the same symbol as the difference matrix (e.g. \(D_0, D_1, D_2, \text{etc.}\)). A diagram representing some difference fields for the 2 x 16 rectangle case are also given in Figure 6.

5. Simple Difference Field Processing

In his 1960 article Julesz postulated a model for the data processing that occurs between retinal images and consciousness of depth. This model will be called the simple difference field model in order to distinguish it from the sequential difference field model which will be presented later.

In the simple difference field model the following set of operations takes place. The two pictures, \(f_L\) and \(f_R\), are fused. The processor computes \(D_0, D_1, \text{etc.}\), perhaps up to \(D_{10}\). Each uniform area in \(D_x\), some amount larger than might occur at random, is interpreted as resulting from a plane, of that size and shape, above the background. The distance above the background is proportional to the shift \(\Delta = x\). (\(\Delta\) is the running variable, \(x\) is any specific value of \(\Delta\).)
For example consider Figure 6. The processor determines $D_0$, $D_2$ and a number of $D_x$ which, being random, are rejected. It treats the uniform area on $D_0$ as a picture-fram-shaped plane making up the background. The processor interprets the uniform area in $D_2$ to be a $2 \times 16$ rectangle above the background by a distance proportional to $\Delta = 2$.

6. Sequential Difference Field Processing

In the sequential difference field model the following operations take place. The two pictures, $f_L$ and $f_R$, are fused. The processor computes $D_0$ and, finding a uniform area, establishes the background. Then, ignoring all the points it has placed in the background, computes $D_1$. If $D_1$ is random, it goes on to compute $D_2$. If $D_2$ has uniform area, a plane is established at $\Delta = 2$. If any unexplained points remain, the processor proceeds to compute higher order difference fields but only for the explained regions of the fused pictures.

For example consider the case of the $2 \times 16$ rectangle. The difference fields computed by the sequential difference field processor would appear as shown in Figure 6. Compare these difference fields with the simple difference fields above them.

Note how the area in which the processor is searching decreases as $\Delta$ increases. This is because there is only a $64 \times 64$ matrix to fill. If there is a plane at $\Delta = x$, there
must be \( x \) columns of new random black and white squares. This leaves only \( 4-x \) columns to make up the plane.

This last point can perhaps be made clearer by this mechanical analogy. The processor computes \( D_0 \) and finds the uniform area. Thereupon it makes up two masks just the size and shape of the uniform area. One mask is fitted over \( f_L \), one over \( f_R \). In sliding \( f_L \) and \( f_R \) past each other to compute \( D_1 \), the masks have become shifted relative to each other by one column. The processor will only compute a difference field between points in \( f_L \) and points in \( f_R \) — not between a mask and a point in a field. Thus for \( D_1 \) the difference field is \( 4-1=3 \) columns wide.
III EXPERIMENTAL RESULTS

In part III experimental evidence is presented which favors the data processing model proposed in this paper. Section 1 concerns the perception of a single square above a plane. Subjects' ability to see such a square is strong evidence for the sequential difference field model proposed in this paper. Section 2 presents another experiment in which the simple and sequential models differ in their predictions. Again subjects' perceptions are as predicted by the new model proposed in this paper. The experiment in section 3 shows two things. First, the sequential difference field model is seen to correctly predict a perception that is incorrectly predicted by the simple difference field model. Secondly, evidence is seen of another data processing phenomena, local generalization. This processing takes place after sequential difference field processing. The stereo pairs in section 4 illustrate some of the properties of local generalization.

1 Compare Sequential to Simple Difference Field Processing - First Experiment

It is clear that in the case of a 2 x 16 rectangle shifted against a background either data processing model yields the same conscious perception.

However consider what happens as that rectangle is shrunk in size. In particular, consider a 1 x 1 rectangle shifted \( \Delta = 1 \). In figure 7 a single white square has been
Figure 7 (see page 18 for legend)
shifted and the vacancy filled with a gray square.

Simple difference field processing could never resolve a single shifted square. The $D_1$ matrix indeed contains a 0 at the appropriate location, but it also contains 0's at random places all over the whole field. In the simple difference field model the difference matrix must contain a cluster of 0's, above some threshold size, before this cluster of 0's is interpreted as being non-random.

It is clear from the diagram at the bottom of Figure 7 that sequential difference processing can resolve a single shifted square. Viewing the stereo pair, subjects perceive a simple square above the background. This is the most substantial piece of evidence supporting the sequential difference field model.

Additional Details on Experiment One

It is of interest to note the sequential difference field processor also predicts perception of single squares which have been shifted by larger amounts. Stereo pairs with shifts of $\Delta = 2$ and $\Delta = 3$ have been made up and are easily perceived.

Of course all cases of a single square shift will not be perceptible. For example consider the following: an element of value 3 is shifted $\Delta$ where it replaces an element of value 3 in the second matrix. The new element put in the vacancy is also of value 3. Thus both the left and right hand matrices will be identical. Clearly no depth will be perceived.
If the square to be shifted is of value $x$, then from the sequential difference model the requirement for stereo perception is as follows: The element displaced by the shifted element of value $x$ must be some value other than $x$. The element filled into the vacancy must also be some value other than $x$. Note that in Figure 7 $x$ is white. Both the displaced and filled in elements chance to be of the same value, light grey.

2 Another Experimental Test of the Sequential Difference Field Model

This second experiment concerns a $2 \times 16$ rectangle shifted $\Delta = 2$. The stereo pair of Figure 8 is so arranged that the simple difference field model predicts a $2 \times 16$ rectangle above the plane. The sequential difference field model predicts no perception of depth.

The difference in predictions results from the unusual composition of the stereo pairs. A rectangle 2 columns by 16 rows has been shifted by $\Delta = 2$. However the vacancy has not been filled with random grey shades. Rather the vacancy is itself filled with the same $2 \times 16$ rectangle.

The result of this is that the $D_0$ of Figure 8 is considerably different from the $D_0$ of Figure 6. The smaller hole in the $D_0$ mask causes the sequential processor on Figure 8 to be blind to the $\Delta = 2$ match. Thus the sequential difference field model predicts no perception of depth.
Composition of Stereo Pair

Simple Difference Fields

Sequential Difference Fields

Figure 8 (see page 18 for legend)
For the simple difference field processor to see depth, the difference matrix must have a cluster of 0's above some threshold size. Since depth can be seen in Figure 6, a 2 x 16 cluster must be above such a threshold. Thus the simple difference field processor predicts perception of depth.

Subjects, viewing the stereo pair of Figure 8 through a stereo viewer, report no stereopsis. A 2 x 16 rectangle of confusion is perceived against a background. This is another piece of evidence supporting the sequential difference field model.

3. Experimental Evidence for Local Generalization

The third experiment also concerns a 2 x 16 rectangle. The stereo pair of Figure 9 is so arranged that the simple difference field model predicts a 2 x 16 rectangle at $\Delta = 2$ and a 2 x 16 rectangle at $\Delta = 4$. The sequential difference model predicts a 2 x 16 rectangle at $\Delta = 2$ and a 2 x 16 area of confusion.

The difference in predictions results from the special composition of the stereo pair. The left field of the stereo pair in Figure 9 contains not just one shifted 2 x 16 rectangle but two copies of the same rectangle. One copy is at $\Delta = 2$; the other copy, at $\Delta = 4$.

The simple difference field model predicts a plane at $\Delta = 2$ and one at $\Delta = 4$. This can be seen by examining the $D_2$ and $D_4$ for simple difference fields.
Composition of Stereo Pair

Simple Difference Fields

Sequential Difference Fields

Figure 9 (see page 18, for legend)
For sequential difference field processing, \( D_0 \), of course, is the same as in the simple difference field case. However once \( D_2 \) is found, the \( 2 \times 16 \) rectangle in \( f_R \) is removed from further attempts at matching. Thus a match can not be made at \( \Delta = 4 \). The predicted perception is then just a single \( 2 \times 16 \) plane above the background. Next to this plane is a \( 2 \times 16 \) area of confusion.

In viewing the stereo pair of Figure 9 one perceives a \( 2 \times 16 \) plane at \( \Delta = 2 \), an area of confusion in which a number of points are also at \( \Delta = 2 \), and some of the background at \( \Delta = 2 \). This last is particularly prominent at the lower right hand corner of the area of confusion where a domain of some six squares occasionally appears to jump into the upper plane at \( \Delta = 2 \). (The perception is essentially the same with or without the vertical lines.)

The single plane above the background at \( \Delta = 2 \) is the result predicted by the sequential difference model.

The other results can be explained in this way. First planes are located by the sequential difference process. Then the processor takes each valid difference field and recomputes it as follows: It enlarges each of the two masks by, say, two squares all around. Then it recomputes the difference matrix for this enlarged area. Any new zeros appearing in this difference matrix are considered valid members of the shifted plane. Recomputation may again occur beginning with this new valid plane. This process, local generalization, allows
established planes to become enlarged. The result, that there is a plane at $\Delta = x$, is generalized by the processor. However this generalization occurs only in the local neighborhood of the established plane.

Note that there are two kinds of local generalizations. In the first kind, squares at $\Delta = x$ are found in a region of confusion. Thus in Figure 9 many of the elements in the region of confusion are seen to be in the plane at $\Delta = 2$ or in the background ($\Delta = 0$). In the second kind, squares at $\Delta = x$ are found in the background. These are squares that could equally well be perceived as being in the background or in the plane at $\Delta = x$. It is not clear how the processor decides where to put squares that could go up or down. Often these ambiguous points are unstable enough to appear to move up and down from time to time. Such a group of points is found in the background at the lower right hand edge of the region of confusion in the stereo pair of Figure 9.

Thus two things are shown by the stereo pair of experiment three. First, the two data processing models make different predictions as to perception - with the sequential difference model making the correct prediction. Second, that the data processor performs a function called "local generalization". Local generalization is the process of enlargement which occurs after a plane has been formed. The processor having found a stable plane by sequential difference field, it tries to enlarge ("generalize") that plane by picking up adjoining or near neighbor points ("local" points).
4. Investigation of Local Generalization

The stereo pair in Figure 10 is arranged so that the simple difference field model predicts a 5 x 16 rectangle at \( \Delta = 1 \). The sequential difference field model predicts a 4 x 16 rectangle at \( \Delta \neq 1 \). However this experiment is arranged so that local generalization can occur. If local generalization favorable to the \( \Delta = 1 \) plane occurs, a 5 x 10 rectangle will be perceived with the sequential difference model.

These possibilities result from the composition of the stereo pairs. This pair contains a 5 x 16 rectangle shifted by \( \Delta = 1 \). The vacancies remain at their old values. Thus in \( f_R \) there are two identical columns. One is the 1 x 16 "SHIFT" column. The other is the column on its immediate left. It should be noted that the column furthest to the right in the 5 x 16 "SHIFT" area of \( f_L \) is identical to the 1 x 16 "SHIFT" column in \( f_R \). These two columns are identically located in their respective matrices. As a result they will appear to be part of the background when \( D_0 \) is computed.

Simple difference processing yields two non-random difference fields. The predicted perception is of a 5 x 16 rectangle at \( \Delta = 1 \).

Sequential difference field processing yields two difference fields. Note that the rectangle in \( D_1 \) is only 4 x 16, not 5 x 16. The predicted perception is of a 4 x 16 rectangle at \( \Delta = 1 \).

Viewed stereoscopically Figure 10 contains a 4 x 16 rectangle at \( \Delta = 1 \). This evidently confirms the sequential processing theory.
Composition of Stereo Pair

Figure 10 (see page 18 for legend)

Simple Difference Fields

Sequential Difference Fields
However it is possible to ask whether local generalization shouldn't have brought that first column to the right of the rectangle in $f_R$ up alongside this rectangle. Perhaps those two identical columns, which look like one column of wide horizontal bars, are hard to break apart because of the horizontal continuity of the bars. Perhaps if each column were separated from adjoining columns local generalization would occur at $\Delta = 1$.

Separation

In figure 11 such separation has been introduced. The result is still a stable perception of a $4 \times 16$ rectangle. However two things have happened. First, the left hand edge of the rectangle is now straight. In Figure 10 some of the left hand edge has been locally generalized into the background. This does not occur in Figure 11. Second, that first column to the right of the rectangle is not now so firmly established in the background. That column can be forced to join the others in the plane to form a $5 \times 16$ rectangle. An easy way to do this is as follows. Note that the fifth from the bottom square on the right hand edge of the rectangle is white. Now note that to the right of this white square is a vertical white bar three squares long. Thinking of these as forming a letter T lying on its side $\rightarrow$, the viewer can force that extra column up.

Another way to introduce separation is to make up a stereo pair like Figure 10 but for $\Delta = 2$. Such a pair is seen in Figure 12.
Viewed stereoscopically Figure 12 at first looks confused. For one subject a $3 \times 16$ rectangle appeared. A few moments later this grew to $5 \times 16$. For another experienced subject there was a period when he could either make himself see $3 \times 16$ or $5 \times 16$. However after a minute or two $5 \times 16$ became stable. Other subjects see $5 \times 16$ right away. Thus in this case local generalization brings the ambiguous area up into the floating plane.

The three stereo pairs in this experiment confirm the presence of some processing like local generalization. However they do not seem to generate a clear picture of how the processor assigns ambiguous areas - areas that could be locally generalized into either of two planes.
IV CONCLUSION

The operations which occur between retinal image formation and conscious perception may be conceived of as data processing. Bella Julesz postulated a simple data processing model to explain how retinal images of random stereo pairs are converted into perception of a plane above a background. A refined version of Julesz's model is presented in this paper.

A series of experiments was so designed that the two models would make different predictions. The results of these experiments are in agreement with the new, refined model.

The data processor of the new model handles a stereo pair as follows: 1) The pictures are fused, 2) A $D_0$ is computed and the area of agreement masked in each field, 3) Higher and higher order $D_x$'s are computed, with new masking, 4) Planes are established, 5) Each plane is locally generalized.

It is not clear how the processor decides where to put some squares when local generalization would put them in either of two planes. This fact indicates that this model too will require refinement.
BIBLIOGRAPHY


FRESNEL LENSES

In one hand the Fresnel lenses are held 10 to 14 inches from the stereo pair. Looking through the lenses as if they were glasses, the viewer places a piece of cardboard (10 to 14 inches long) between his nose and the stereo pair. This cardboard is between the two pictures and perpendicular to the page. The viewer fuses the two pictures and then waits, perhaps minutes, for the depth to appear. Viewers who wear glasses should not remove them.
This appendix contains the programs used to generate stereo picture pairs such as those found in this thesis. Enough information is given so that one could punch out one of these programs on cards and have sample stereo pairs in his hands in a day or two.

1. Picture Making: Two Steps

There are two steps in the making of these random pictures. The first is the generation of appropriate matrices written on magnetic tape. The second is the conversion of these matrices on tape into pictures on film.

The conversion from tape to pictures is performed by the machine in Figure 13. It was developed by Prof. Schreiber and is presently in the Cognitive Information Processing Group of the Research Laboratory of Electronics. Using this machine, the conversion from magnetic tape to pictures is just a matter of a little button pushing. These details will not be covered here.

The magnetic tapes are generated on the IBM 7094 computer in the MIT computation Center. The Machine is programmed to develop a matrix of random numbers, to manipulate that matrix, and to both print out that matrix on paper and write that matrix on magnetic tape. This appendix is concerned with these matters.
2. The Four Components of the First Step in Picture Making

In order to write a tape an experimenter takes a pile of punched cards and a roll of magnetic tape over to the Computation Center. He turns in the cards at the proper bin in the basement. Upstairs he turns in the reel of tape. The next day the experimenter gets back some computer print out, his original deck of punched cards, and his magnetic tape. The tape now contains matrices for perhaps 12 pictures. The first picture is usually one field of a stereo pair. The other 11 pictures each can be put with the first picture to form a stereo pair - therefore 11 stereo pairs in all.

The pile of punched cards, the program, is made up of four stacks of punched cards piled on top of each other. From the top, they are:

A: First there is the main program. It causes the computer to generate a matrix of random numbers. Then it manipulates this matrix. Different main programs manipulate the random matrix differently. Then the program "64, 2 Level" (A-3) generates a 64 x 64 element matrix in which each element is either a 0 or a 3. It takes a rectangular portion of that matrix and shifts it some amount. This main program "64, 2 Level Sequential" (A-5) generates a 64 x 64 element matrix in which each element is either a 0 or a 3. It shifts a rectangular portion of that matrix. Then it goes on to shift
another and another until it is told to stop. This main program was used to generate Figure 4. Each main program generates a matrix, manipulates it, and leaves the resulting matrix stored in computer memory.

B: Second comes the print out-routine. The main program, once it has finished manipulating the matrix, calls for a subroutine to print that matrix out on paper. The subroutine causes the computer's printer to print 64 lines of numbers, each line containing 64 digits. For example one may use "64, 2 Level" (A-3) as a main program and "SIXFOR" (B-2) as the print output-routine. The machine will print out 64 x 64 digit array where each digit is a 0 or a 3 (corresponding to a black or white square in a picture). Once this subroutine has written the matrix on paper, it calls for another subroutine to write the matrix on tape.

C: Third comes the tape-writing routine. It is a big stack of binary cards which one obtains from whoever is in charge of the machine that converts tapes to pictures. All that need be known about this subroutine is that it will write a matrix on tape. This appendix will not discuss or give examples of tape-writing routines.

D: Last come the data cards. This stack of cards does a number of things. It tells the computer how many matrices will be made up (e.g. LPTOTL = 12, i.e. the loop total is 12). It tells the computer how big a rectangle to shift. (e.g.
JBOTSM = 35, JTOPLG = 50, IBOTSM = 25, ITOPLG = 29 means shift the elements in the matrix in rows 35 through 50 and column 25 through 29 - a 16 x 5 array.) And it tells the computer how far to shift the rectangle (e.g. ISHIFT = 5 means shift the rectangle over 5 elements).

3. A Tabulation of the Varieties of Each of the Four Components

Eight main programs are given in this appendix. They are designated A-x.

\[
\begin{align*}
A-1 & : 128, 4 \text{ LEVEL} \\
A-2 & : 128, 2 \text{ LEVEL} \\
A-3 & : 64, 4 \text{ LEVEL} \\
A-4 & : 64, 2 \text{ LEVEL} \\
A-5 & : 64, 4 \text{ LEVEL SEQUENTIAL} \\
A-6 & : 64, 4 \text{ LEVEL SEQ. NO NOISE BUT FINAL} \\
A-7 & : 64, 2 \text{ LEVEL NO NOISE EVER} \\
A-8 & : 64, 2 \text{ LEVEL SUBSTITUTE}
\end{align*}
\]

Three print-output-subroutines are given in this appendix. They are designated B-x.

\[
\begin{align*}
B-1 & : \text{OUTPUT} \\
B-2 & : \text{SIXFOR} \\
B-3 & : \text{SIXFOR (WITH VERTICLE LINES)}
\end{align*}
\]

The tape-writing subroutine is available in binary form. This stack of cards will be designated C.

Two examples of data are given. The second set of data can be used only with A-8. Data card decks will be designated D-x.

\[
\begin{align*}
D-1 & : \text{Typical data for most programs} \\
D-2 & : \text{Data for "SUBSTITUTE" (A-8)}
\end{align*}
\]
<table>
<thead>
<tr>
<th>MAIN PROGRAM</th>
<th>PRINT OUT</th>
<th>WRITE OUT</th>
<th>DATA</th>
<th>RESULTANT PICTURES</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-1</td>
<td>B-1</td>
<td>C</td>
<td>D-1</td>
<td>Each of the pictures is a shifted rectangle whose vacancy is filled with random digits</td>
</tr>
<tr>
<td>A-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A-3</td>
<td>B-2</td>
<td>C</td>
<td>D-1</td>
<td>Shifted rectangle, vacancy filled with random digits</td>
</tr>
<tr>
<td>A-4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A-5</td>
<td>B-2</td>
<td>C</td>
<td>D-1</td>
<td>This data is not suitable for sequential programs. This program shifts a square, then fills vacancy with random digits, then shifts another square, then fills etc.</td>
</tr>
<tr>
<td>A-6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A-7</td>
<td>B-2</td>
<td>C</td>
<td>D-1</td>
<td>Like A-5, only after each shift the vacancies are left at there old values. Only after the final shift are the final vacancies filled with random digits. (used to make Fig. 4)</td>
</tr>
<tr>
<td>A-8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5. Copies of the Programs

Copies of these main programs, printing subroutines, and sample data are given bellow in alphabetical and numerical order.
4. Combinations of Components with Results

Note:

"64" means the final picture will be 64 squares wide by 64 squares high.

"128" means the final picture will be 128 squares wide by 128 squares high (smaller squares than above).

"2 LEVEL" means the squares in the final picture will be in two gray levels, black and white.

"4 LEVEL" means the squares in the final picture will be in four shades of gray.

B-3 can be used anywhere B-2 is shown. The result will be the same picture except that the picture will have vertical lines in it (such as in Fig.12).

A data card reading 0 0 0 0 0 causes the machine to generate a random matrix, skip over all manipulations, and go directly to the print and tape-writing routine. Thus this card generates a master picture used with each of the others in making a stereo pair.
XEQ

SET DENSITY LOW(10)

LABEL

128X128, FOUR GREY LEVELS

DIMENSION IA(128,128), STORE(805), ISTORE(805), IAPERM(46,46)

FORMAT(27H 128X128, FOUR GREY LEVELS//)

FORMAT(1H/1,128A1)

FORMAT(I5)

FORMAT(10H LOOP,10H LPTOTL//)

FORMAT(2I10//)

FORMAT(5I7)

FORMAT(7H ISHIFT,7H JTOPSM,7H JBOTLG,7H ILFTSM,7H IRGTLG//)

DO 20 K=1,50

20 FLUSH = RANNOF(X)

DO 30 L=1,800

30 STORE(L) = RANNOF(X)

DO 39 L=1,800

IF(STORE(L)-.75) 32,32 ,38

IF(STORE(L)-.50)33,33,37

IF(STORE(L)-.25)34,34,36

INTS =1023

GO TO 39

36 INTS = 682

GO TO 39

37 INTS = 341

GO TO 39

38 INTS = 0

39 ISTORE(L) = INTS

DO 60 J=1,128

DO 60 I=1,128

RND = RANNOF(X)

C

FOUR GREY LEVELS

IF(RND = .75)42,42,48

IF(RND = .50)43,43,47

43 IF(RND = .25)44,44,46

44 INT = 1023

GO TO 49

46 INT = 682

GO TO 49

47 INT = 341

GO TO 49

48 INT = 0

49 IA(I,J) = INT

60 CONTINUE

READ 6, LPTOTL

DO 99 LOOP = 1*LPTOTL

PRINT 4

PRINT 4

PRINT 7

PRINT 8*LOOP,LPTOTL

READ 9,ISHIFT, JTOPSM, JBOTLG, ILFTSM, IRGTLG

PRINT 10

PRINT 7,ISHIFT, JTOPSM, JBOTLG, ILFTSM, IRGTLG

DO 605 J=35,50

DO 605 I=35,80

605 IAPERM(1-34,J-34) = IA(I,J)

IF(ISHIFT)85,85,61
CONTINUE
DO 62 J = JTOPSM, JBOTLG
DO 62 I = ILFTSM, IRTGTLG
INEW = I - ISHIFT
IA(INEW, J) = IA(I, J)
IWIDTH = IRTGTLG - ILFTSM + 1
IF(IWIDTH - ISHIFT) .GT. 63, 63
63
L = 1
INEWRT = IRTGTLG - ISHIFT + 1
DO 64 J = JTOPSM, JBOTLG
DO 64 I = INEWRT, IRTGTLG
IA(I, J) = ISTORE(L)
64
L = L + 1
GO TO 85
70
L = 1
DO 71 J = JTOPSM, JBOTLG
DO 71 I = ILFTSM, IRTGTLG
IA(I, J) = ISTORE(L)
71
L = L + 1
GO TO 85
85
CONTINUE
PRINT 4
CALL OUTPUT (IA)
DO 98 J = 35, 80
DO 98 I = 35, 80
98
IA(I, J) = IAPERM(I-34, J-34)
99
CONTINUE
CALL EXIT
END

TOTAL
* M3013-2403, FMS, DEBUG, 5, 5, 2000, 0 J DISBROW TWO, 128

* XEQ

* SET DENSITY LOW(10)

* LIST

* LABEL

DIMENSION IA(128,128), STORE(805), ISTORE(805), IAPERM(46,46)

FORMAT(27H 128X128, TWO GREY LEVELS//)

FORMAT(1H/, 128A1)

FORMAT(10H LOOP, 10H LPTOTL//)

FORMAT(2110///)

FORMAT(517)

FORMAT(7H ISHIFT, 7H JTOPSM, 7H JBOTLG, 7H ILFTSM, 7H IRGTLG//)

DO 20 K = 1, 50

20 FLUSH = RANNOF(X)

DO 30 L = 1, 800

30 STORE(L) = RANNOF(X)

DO 39 L = 1, 800

IF (STORE(L) - .50) < 0, 34, 38

34 INTS = 1023

GO TO 39

38 INTS = 0

39 ISTORE(L) = INTS

DO 60 J = 1, 128

DO 60 I = 1, 128

RND = RANNOF(X)

IF(RND - .50) 44, 44, 48

44 INT = 1023

GO TO 49

48 INT = 0

49 IAI(I, J) = INT

CONTINUE

READ 6, LPTOTL

DO 99 LOOP = 1, LPTOTL

PRINT 4

PRINT 4

PRINT 3

PRINT 7

PRINT 8, LOOP, LPTOTL

READ 9, ISHIFT, JTOPSM, JBOTLG, ILFTSM, IRGTLG

PRINT 10

PRINT 9, ISHIFT, JTOPSM, JBOTLG, ILFTSM, IRGTLG

DO 605 J = 35, 80

DO 605 I = 35, 80

IAPERM(I - 34, J - 34) = IAI(I, J)

IF(IISHIFT) 85, 85, 61

61 CONTINUE

DO 62 J = JTOPSM, JBOTLG

DO 62 I = ILFTSM, IRGTLG

INew = 1 - IISHIFT

62 IAI(INew, J) = IAI(I, J)

IWIDTH = IRGTLG - ILFTSM + 1

IF(IWIDTH = IISHIFT) 70, 63, 63

L = 1

INewRT = IRGTLG - IISHIFT + 1

DO 64 J = JTOPSM, JBOTLG

DO 64 I = INewRT, IRGTLG

IAI(I, J) = ISTORE(L)
64    \text{L=L+1}
    \text{GO TO 85}

70    \text{L=1}
    \text{DO 71 J=JTOPSM,JBOTLG}
    \text{DO 71 I=ILFTSM,IRGTLG}
    \text{IA(I,J) = ISTORE(L)}

71    \text{L=L+1}

85    \text{CONTINUE}
    \text{PRINT 4}
    \text{CALL OUTPUT (IA)}
    \text{DO 98 J=35,80}
    \text{DO 98 I=35,80}
    \text{IA(I,J) = IAPERM(I-34,J-34)}

99    \text{CONTINUE}
    \text{CALL EXIT}

\text{END}

\text{A - 3}

*63013-2403, FMS, DEBUG, 5, 5, 2000, 0 J DISBROW FOUR, 64
*XEO
*SET DENSITY LOW(10)
*LIST
*LABEL
\text{64-64, FOUR GREY LEVELS}
\text{DIMENSION IA(64,64),STORE(805),ISTORE(805),IAPERM(64,64)}
\text{FORMAT(2TH \hspace{1em} 64X64 \hspace{1em} FOUR GREY LEVELS//)}
\text{FORMAT(1H1)}
\text{FORMAT(1H/\hspaces{1em}64A1)}
\text{FORMAT(15)}
\text{FORMAT(10H \hspace{1em} LOOP,10H LPTOTL//)}
\text{FORMAT(2I10/\hspaces{1em}+++)}
\text{FORMAT(517)}
\text{FORMAT(7H ISHIFT,7H JTOPSM,7H JBOTLG,7H ILFTSM,7H IRGTLG/)}
\text{DO 20 K=1,50}
\text{FLUSH = RANNOF(X)}
\text{DO 30 L=1,800}
\text{STORE(L) = RANNOF(X)}
\text{DO 39 L=1,800}
\text{IF(STORE(L)-.75)32,32,38}
\text{IF(STORE(L)-.50)33,33,37}
\text{IF(STORE(L)-.25)34,34,36}
\text{INTS =1023}
\text{GO TO 39}
\text{INTS = 682}
\text{GO TO 39}
\text{INTS = 341}
\text{GO TO 39}
\text{INTS = 0}
\text{ISTORE(L) = INTS}
\text{DO 60 J=1,64}
\text{DO 60 I=1,64}
\text{RND = RANNOF(X)}
\text{C FOUR GREY LEVELS}
\text{IF(RND - .75)42,42,48}
\text{IF(RND - .50)43,43,47}
\text{IF(RND - .25)44,44,46}
**44**  INT = 1023
GO TO 49

**46**  INT = 682
GO TO 49

**47**  INT = 341
GO TO 49

**48**  INT = 0

**49**  IA(I,J) = INT
CONTINUE

READ 6, LPTOTL
DO 99 LOOP = 1, LPTOTL
PRINT 4
PRINT 4
PRINT 3
PRINT 7
PRINT 8, LOOP, LPTOTL
READ 9, ISHIFT, JTOPSM, JBOTLG, ILFTSM, IRGTLG
PRINT 10
PRINT 9, ISHIFT, JTOPSM, JBOTLG, ILFTSM, IRGTLG

**49**  IA(I,J) = INT
CONTINUE

READ 6, LPTOTL
DO 99 LOOP = 1, LPTOTL
PRINT 4
PRINT 4
PRINT 3
PRINT 7
PRINT 8, LOOP, LPTOTL
READ 9, ISHIFT, JTOPSM, JBOTLG, ILFTSM, IRGTLG
PRINT 10
PRINT 9, ISHIFT, JTOPSM, JBOTLG, ILFTSM, IRGTLG

**60**  IA(I,J) = IA(I,J)
IF(ISHIFT) 85, 85, 61

**61**  CONTINUE

DO 62 J = JTOPSM, JBOTLG
DO 62 I = ILFTSM, IRGTLG
INEW = I - ISHIFT

**62**  IA(INEW, J) = IA(I, J)
IWIDTH = IRGTLG - ILFTSM + 1
IF(IWIDTH - ISHIFT) 70, 63, 63

**63**  L = 1
INEWRT = IRGTLG - ISHIFT + 1
DO 64 J = JTOPSM, JBOTLG
DO 64 I = INEWRT, IRGTLG
IA(I, J) = ISTORE(L)

**64**  L = L + 1
GO TO 85

**70**  L = 1
DO 71 J = JTOPSM, JBOTLG
DO 71 I = ILFTSM, IRGTLG
IA(I, J) = ISTORE(L)

**71**  L = L + 1

**85**  CONTINUE

PRINT 4
CALL SIXFOR(IA)
DO 98 J = 1, 64
DO 98 I = 1, 64

**98**  IA(I, J) = IAPERM(I, J)
CONTINUE

**99**  CALL EXIT
END

*A - 4*

*\[
*\text{M3013-2403, FMS, DEBUG, 5.5, 2000, 0} \quad \text{J DISBROW, TWO, 64}*
*\text{XEQ}*
*\text{SET DENSITY LOW(10)}*
* LIST
* LABEL

64-64, TWO GREY LEVELS

DIMENSION IA(64,64),STORE(805),ISTORE(805),IAPERM(64,64)

FORMAT(27H 64X64, TWO GREY LEVELS/)

FORMAT(1H1)

FORMAT(1H/,64A1)

FORMAT(10H LOOP,10H LPTOTL/)

FORMAT(2I10//)

FORMAT(5I7)

FORMAT(7H ISHIFT,7H JTOPSM,7H JBOTLG,7H ILFTSM,7H IRGTLG//)

DO 20 K=1,50

FLUSH = RANNOF(X)

DO 30 L=1,800

STORE(L) = RANNOF(X)

DO 39 L=1,800

IF(STORE(L)-.50)33,33,37

INTS = 1023

GO TO 39

INTS = 0

ISTORE(L) = INTS

DO 60 J=1,64

DO 60 I=1,64

RND = RANNOF(X)

IF(RND -.50)43,43,47

INT = 1023

GO TO 49

INT = 0

IA(I,J) = INT

CONTINUE

READ 6, LPTOTL

DO 99 LOOP = 1,LPTOTL

PRINT 4

PRINT 4

PRINT 7

PRINT 8, LOOP, LPTOTL

READ 9, ISHIFT, JTOPSM, JBOTLG, ILFTSM, IRGTLG

PRINT 10

PRINT 9, ISHIFT, JTOPSM, JBOTLG, ILFTSM, IRGTLG

DO 605 J=1,64

DO 605 I=1,64

605 IAPERM(I,J) = IA(I,J)

IF(ISHIFT)>5,85,61

CONTINUE

DO 62 J = JTOPSM, JBOTLG

DO 62 I=ILFTSM, IRGTLG

INEW = I-ISHIFT

62 IA(INEW,J) = IA(I,J)

IWIDTH = IRGTLG - ILFTSM +1

IF(IWIDTH-ISHIFT)>70,63,63

L=1

INEWRT = IRGTLG-ISHIFT+1

DO 64 J=JTOPSM, JBOTLG

DO 64 I=INEWRT, IRGTLG

IA(I,J) = ISTORE(L)

64 L=L+1

GO TO 85
L=1
DO 71 J=JTOPSM,JBOTLG
DO 71 I=ILFTSM,IRGTLG
IA(I,J) = ISTORE(L)
71 L=L+1
CONTINUE
PRINT 4
CALL SIXFOR(IA)
DO 98 J=1,64
DO 98 I=1,64
IA(I,J) = IAPERM(I,J)
98 CONTINUE
CALL EXIT
END

*M3013-24 3,FMS,DEBUG,5,5,2000,0
* XEQ
* SET DENSITY LOW(10)
* LIST
* LABEL
SEQUENTIALLY FORMED 64X64, FOUR GREY LEVELS
DIMENSION IA(64,64),STORE(805),ISTORE(805),IAPERM(64,64)
FORMAT(2ITH 64X64, FOUR GREY LEVELS)
FORMAT(1H/1)
FORMAT(1H/64A1)
FORMAT(15)
FORMAT(10H LOOP,10H LPTOTL)
FORMAT(2I10/)
FORMAT(5I17)
FORMAT(7H ISHIFT,7H JTOPSM,7H JBOTLG,7H ILFTSM,7H IRGTLG)
FORMAT(40H PICTURE FORMED BY SEQUENTIAL OPERATIONS)
DO 20 K=1,50
20 FLUSH = RANNOF(X)
DO 30 L=1,800
STORE(L) = RANNOF(X)
DO 39 L=1,800
IF(STORE(L)-.75)32,32,38
IF(STORE(L)-.50)33,33,37
32 IF(STORE(L)-.7)34,34
33 INTS =1023
34 INTS =682
36 INTS =341
37 INTS =0
GO TO 39
38 GO TO 39
39 ISTORE(L) = INTS
GO TO 39
DO 60 J=1,64
DO 60 I=1,64
RND = RANNOF(X)
IF(RND -.75)42,42,48
42 IF(RND -.50)43,43,47
43 IF(RND -.25)44,44,46

FOUR GREY LEVELS
INT = 1023
GO TO 49

INT = 682
GO TO 49

INT = 341
GO TO 49

INT = 0

IA(I,J) = INT
CONTINUE
READ 6,LPTOTL
DO 99 LOOP = 1,LPTOTL
PRINT 4
PRINT 4
PRINT 3
PRINT 7
PRINT 8,LOOP,LPTOTL
READ 9,ISHIFT,JTOPSM,JBOTLG,ILFTSM,IRGTGL
PRINT 10
PRINT 9,ISHIFT,JTOPSM,JBOTLG,ILFTSM,IRGTGL
DO 605 J=1,64
DO 605 I=1,64
IAPERM(I,J) = IA(I,J)
IF(ISHIFT)85,85,61
CONTINUE
DO 62 J = JTOPSM,JBOTLG
DO 62 I=ILFTSM,IRGTGL
INEW = I-ISHIFT
IA(INEW,J) = IA(I,J)
IWIDTH = IRGTGL - ILFTSM +1
IF(IWIDTH-ISHIFT)70,63,63
L=1
INEWRT = IRGTGL-ISHIFT+1
DO 64 J=JTOPSM,JBOTLG
DO 64 I=INEWRT,IRGTGL
IA(I,J) = ISTORE(L)
L=L+1
GO TO 85
L=1
DO 71 J=JTOPSM,JBOTLG
DO 71 I=ILFTSM,IRGTGL
IA(I,J) = ISTORE(L)
L=L+1
CONTINUE
PRINT 4
PRINT 11
PRINT 11
PRINT 4
CALL SIXFOR(IA)
CONTINUE
CALL EXIT
END

A - 6
LIST

LABEL

SEQUENTIALLY FORMED 64X64, FOUR GREY LEVELS

DIMENSION IA(64,64), STORE(805), ISTORE(805), IAPERM(64,64)

FORMAT(27H 64X64, FOUR GREY LEVELS/)

FORMAT(1H1)

FORMAT(1H/ 64A1)

FORMAT(15)

FORMAT(10H LOOP, 10H LPTOTL/)

FORMAT(2I10/)

FORMAT(5I7)

FORMAT(7H ISHIFT, 7H JTOPSM, 7H JBOTLG, 7H ILFTSM, 7H IRGTLG/)

FORMAT(40H PICTURE FORMED BY SEQUENTIAL OPERATIONS/)

DO 20 K = 1, 50

FLUSH = RANNOF(X)

DO 30 L = 1, 800

STORE(L) = RANNOF(X)

DO 39 L = 1, 800

IF(STORE(L) -.75) 32, 32, 38

IF(STORE(L) -.50) 33, 33, 37

IF(STORE(L) -.25) 34, 34, 36

INTS = 1023

GO TO 39

INTS = 682

GO TO 39

INTS = 341

GO TO 39

INTS = 0

ISTORE(L) = INTS

DO 60 J = 1, 64

DO 60 I = 1, 64

RND = RANNOF(X)

CONTINUE

READ 6, LPTOTL

DO 99 LOOP = 1, LPTOTL

PRINT 4

PRINT 3

PRINT 7

PRINT 8, LOOP, LPTOTL

READ 9, ISHIFT, JTOPSM, JBOTLG, ILFTSM, IRGTLG

PRINT 10

PRINT 9, ISHIFT, JTOPSM, JBOTLG, ILFTSM, IRGTLG

DO 605 J = 1, 64

DO 605 I = 1, 64

IAPERM(I, J) = IA(I, J)
IF(ISHIFT)85,85,61

61 CONTINUE
DO 62 J = JTOPSM,JBOTLG
   DO 62 I=ILFTSM,IRGTLG
      INEW = I-ISHIFT
55
62 IA(INEW,J) = IA(I,J)
      IWIDTH = IRGTLG - ILFTSM +1
      IF(LPTOTL-LOOP)625,625,85
625 CONTINUE
IF(IWIDTH-ISHIFT)70,63,63
63 L=1
INEWRT = IRGTLG-ISHIFT+1
   DO 64 J=JTOPSM,JBOTLG
      DO 64 I=INEWRT,IRGTLG
         IA(I,J) = ISTORE(L)
   L=L+1
   GO TO 85
70 L=1
   DO 71 J=JTOPSM,JBOTLG
      DO 71 I=ILFTSM,IRGTLG
         IA(I,J) = ISTORE(L)
   L=L+1
85 CONTINUE
PRINT 4
PRINT 11
PRINT 11
PRINT 4
CALL SIXFOR(IA)
99 CONTINUE
CALL EXIT
END

*M3013-2403,FMS,DEBUG,5,5,2000,0 J DISBROW SUBSTITUTE, TWO(64
*
* SET DENSITY LOW(10)
* LIST
* LABEL
  SUBSTITUTE 64-64, TWO GREY LEVEL
  MMAX(17),KSUB(17),LPTOTL(15),JSTRT---(4I7),ISHIFT----(5I7)
  DIMENSION IA(64,64),STORE(805),ISTORE(805),IAPERM(64,64)
  DIMENSION KSUB(2000)
  3 FORMAT(27H 64X64 , TWO GREY LEVELS/)
  4 FORMAT(1H1)
  5 FORMAT(1H/ ,64A1)
  6 FORMAT(15)
  7 FORMAT(10H LOOPL,10H LPTOTL/)
  8 FORMAT(210//)
  9 FORMAT(517)
  10 FORMAT(7H ISHIFT,7H JTOPSM,7H JBOTLG,7H ILFTSM,7H IRGTLG/)
  11 FORMAT(17)
  12 FORMAT(417)
  13 FORMAT(7H JSTRT,7H JSTOP,7H ISTRT,7H ISTOP/)
  14 FORMAT(417/)
  15 FORMAT(1017)
   DO 20 K=1,50
20 FLUSH = RANNOF(X)
30 DO 30 L=1,800
30 STORE(L) = RANNOF(X)
DO 39 L=1,800
32 IF(STORE(L)>.50)33,33,37
33 INTS = 1023
GO TO 39
37 INTS = 0
39 ISTORE(L) = INTS
DO 60 J=1,64
DO 60 I=1,64
RND = RANNOF(X)
42 IF(RND-.50)43,43,47
43 INT = 1023
GO TO 49
47 INT = 0
49 IA(I,J) = INT
60 CONTINUE
READ 11, MMAX
READ 15, (KSUB(MAN),MAN=1,MMAX)
MAN=1
READ 6, LPTOTL
DO 99 LOOP = 1, LPTOTL
PRINT 4
PRINT 4
PRINT 3
PRINT 7
PRINT 8, LOOP, LPTOTL
READ 12, JSTART, JSTOP, ISTART, ISTOP
PRINT 13
PRINT 14, JSTART, JSTOP, ISTART, ISTOP
READ 9, ISHIFT, JTOPSM, JBOTLG, ILFTSM, IRGTG
PRINT 10
PRINT 9, ISHIFT, JTOPSM, JBOTLG, ILFTSM, IRGTG
DO 605 J=1,64
DO 605 I=1,64
605 IA(I,J) = IA(I,J)
IF(ISHIFT)85,85,61
61 CONTINUE
DO 62 J = JTOPSM, JBOTLG
DO 62 I = ILFTSM, IRGTG
INEW = I-ISHIFT
62 IA(INEW,J) = IA(I,J)
IWIDTH = IRGTG - ILFTG +1
IF(IWIDTH - ISHIFT)70,63,63
63 L=1
INEWRT = IRGTG-ISHIFT+1
DO 64 J=JTOPM, JBOTLG
DO 64 I=INEWRT, IRGTG
IA(I,J) = ISTORE(L)
64 L=L+1
GO TO 85
70 L=1
DO 71 J=JTOPSM, JBOTLG
DO 71 I = ILFTSM, IRGTG
IA(I,J) = ISTORE(L)
71 L=L+1
85 CONTINUE
PRINT 4
DO 89 J=JSTRT,JSTOP
DO 89 I=ISTRT,ISTOP
IA(I,J) = KSUB(MAN)
MAN = MAN+1
IF(IA(I,J)-3)89,88,89
IA(I,J) = 1023
CONTINUE
CALL SIXFOR(IA)
DO 98 J=1,64
DO 98 I=1,64
IA(I,J) = IAPERM(I,J)
CONTINUE
CALL EXIT
END

SUBROUTINE OUTPUT(IA)
DIMENSION IA(128,128),ITAPE(512),Z(128)
FORMAT(1H/ ,128A1)
DO 90 J=1,128
DO 80 I=1,128
IF(IA(I,J)=550)77,77,79
B77 Z(I) = 600000000000
GO TO 80
B79 Z(I) = 440000000000
CONTINUE
PRINT 5,Z(I),I=1,128
CONTINUE
CALL SPACE (15)
CALL OUT 1(5,512)
DO 110 J=1,128
IT = 0
DO 100 I=1,128
IT = IT + 1
100 ITAPE(IT) = IA(I,J)
DO 110 KT=1,4
CALL OUT 2(ITAPE(1),512)
CONTINUE
RETURN
END

SUBROUTINE SIXFOR(IA)
* LABEL
DIMENSION IA(64,64),ITAPE(256),Z(64)
FORMAT(1H1)
FORMAT(1H/ ,64A1)
FORMAT(53H TAPE DEFECTIVE, WRITING OF PICTURE ON TAPE HALTED)
FORMAT(42H AFTER A SPACE, PICTURE REWRITTEN ON TAPE )
DO 90 J=1,64
DO 80 I=1,64
IF(IA(I,J)-1023)71,76,71
71 IF(IA(I,J))72,77,72
72 IF(IA(I,J)-682)73,78,73
73 IF(IA(I,J)-341)74,7
74 Z(I) = 250000000000
GO TO 80
B76 Z(I) = 030000000000
GO TO 80
B77 Z(I) = 000000000000
GO TO 80
B78 Z(I) = 020000000000
GO TO 80
B79 Z(I) = 010000000000
GO TO 80
80 CONTINUE
PRINT 5,(Z(I),I=1,64)
90 CONTINUE
91 CONTINUE
CALL SPACE (15)
CALL OUT 1(5,256)
DO 110 J=1,64
IT = 0
DO 100 I=1,64
DO 100 LT = 1,4
IT = IT + 1
ITAPE(IT) = IA(I,J)
DO 110 KT=1,4
CALL OUT 2(ITAPE(1),256)
IF (SENSE LIGHT 4)120,110
100 CONTINUE
GO TO 130
120 PRINT 6
PRINT 7
PRINT 4
PRINT 6
PRINT 7
PRINT 6
PRINT 7
PRINT 6
PRINT 7
PRINT 4
GO TO 91
130 CONTINUE
RETURN
END

B-3

SUBROUTINE SIXFOR(I,A)
* LABEL
DIMENSION IA(64,64),ITAPE(256),Z(64)
C VERTICAL LINES VERTICAL LINES VERTICAL LINES
4 FORMAT(1H1)
5 FORMAT(1H/64A1)
6 FORMAT(53H TAPE DEFECTIVE. WRITING OF PICTURE ON TAPE HALTED.)
7 FORMAT(42H AFTER A SPACE, PICTURE REWRITTEN ON TAPE///)/
DO 90 J=1,64
DO 80 I=1,64
IF (IA(I, J) = 550), I = 77, J = 77, 79

B77 Z(I) = 600000000000
GO TO 80

B79 Z(I) = 440000000000

80 CONTINUE

PRINT 5, (Z(I), I = 1, 64)

90 CONTINUE

91 CONTINUE

CALL SPACE (15)

CALL OUT 1(5, 256)

DO 110 J = 1, 64
IT = 0
DO 100 I = 1, 64
IT = IT + 1
100 ITAPE(IT) = IA(I, J)
DO 105 IT = 4, 256
105 ITAPE(IT) = 512
DO 110 KT = 1, 4
CALL OUT 2(ITAPE(1, 256))
IF (SENSE LIGHT 4) 120, 110

110 CONTINUE
GO TO 130

120 PRINT 6
PRINT 7
PRINT 4
PRINT 6
PRINT 7
PRINT 6
PRINT 7
PRINT 6
PRINT 7
PRINT 4
GO TO 91

130 CONTINUE
RETURN
END

* DATA

12

0 0 0 0 0
5 35 50 25 29
1 25 40 35 39
2 25 40 35 39
3 25 40 35 39
1 25 40 25 29
2 25 40 25 29
7 30 39 30 39
5 30 39 30 39
3 30 39 30 39
2 30 39 30 39
1 30 39 30 39

D -1
<table>
<thead>
<tr>
<th></th>
<th>DATA</th>
<th>D-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>310</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>41</td>
<td>46</td>
<td>4</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>41</td>
<td>46</td>
<td>4</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>41</td>
<td>46</td>
<td>4</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>41</td>
<td>46</td>
<td>4</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>41</td>
<td>46</td>
<td>4</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>37</td>
<td>42</td>
<td>16</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>37</td>
<td>42</td>
<td>16</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>37</td>
<td>42</td>
<td>16</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>37</td>
<td>42</td>
<td>16</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>37</td>
<td>42</td>
<td>16</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>