ON RESOLUTION OF AMBIGUITY OF RASTER IMAGES

Vladimir Wojcik

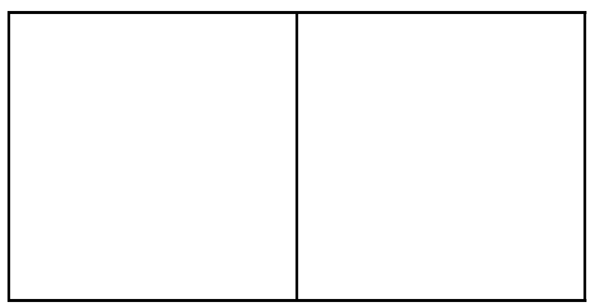
Technical Report # CS-95-01
March 1995

ABSTRACT

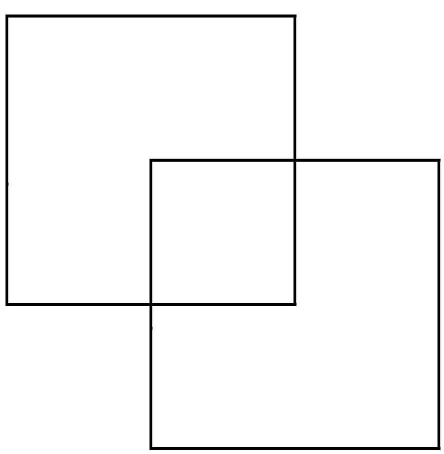
Although raster images are inherently ambiguous, humans exhibit uncanny abilities to come up with their likely interpretations. A methodology to simulate such behaviour on a computer is presented here. A numerical method for rejection of unlikely interpretations is given. A likely interpretation selected is found to depend on a particular raster pattern under study, but also on the architecture of the machine performing the study, as well as on the machine language used to elaborate possible interpretations. A number of suggestions for further research emerge, pertaining to the image description languages, the machine vision computer architectures, graph interpretation systems, as well as comments on the feasibility of automatic, intelligent retouching of partially damaged images.

LOOK AT THE THREE PATTERNS BELOW: WHAT DO YOU SEE?

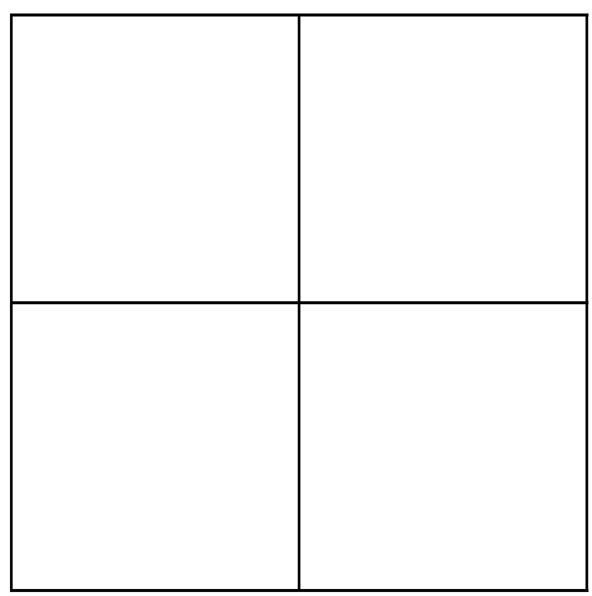
Please take a look at the three patterns below and decide what it is that you seem to see. Please remember this first impression while reading this paper.



Pattern A



Pattern B



Pattern C

RESULTS OF THE HUMAN STUDY

Two studies were performed, consisting of asking human subjects the above question, taken out of context. The feedback of the geometric pattern recognition was sought. Extreme care was exercised in order to minimize the influence on the respondents.

The goal of the preliminary study was to construct as objective an instrument as possible. In this informal study (involving approx. 20 individuals) the author has discovered that the way the patterns were drawn while the respondents were watching influenced the responses. Similarly, each particular sequence of displaying of three pieces of paper with the pre-drawn patterns seemed to exert some influence. It was therefore decided that the final instrument would consist of a single piece of paper, containing the three patterns in question. In the second poll no hint that these patterns were to be interpreted in geometric terms was given, in order to maximize the wealth of potential responses. Luckily, most responses were given in geometric terms anyway. Interestingly, no respondent found the instrument ambiguous in terms where one pattern ends and the other begins, although it was presented as a laser printout, consisting of black dots. This remarkable ability to connect the dots in the "right" way is the subject of study of this paper.

Some responses, however, used references to everyday objects. In this situation the author was forced to interpret the responses in simple geometric terms. To illustrate the types of interpretations conducted, consider the following: A response that pattern A represented a "domino chip" would be translated into "a rectangle divided by a line", while a response referring to a "basement window" would become "two adjacent squares", in reference to the window panes. Ambiguous responses, like "equal" were ignored.

In the final study 60 human subjects were involved. Some respondents chose to give more than one interpretation of a given pattern. For example, pattern A was found by one respondent to be "two squares side by side or a rectangle with intersecting line". Two responses were then registered. However, an interpretation of the pattern C as "4 squares making up 1 large square" was taken to mean " four adjacent squares", the fifth square being the emerging feature. The results of this study (shown in detail in the Appendix) were remarkably convergent.

Pattern A: 64 responses were given, 62 of them unambiguous. Pattern A was found to represent "two adjacent rectangles" 81 percent of the time, or a "rectangle vertically divided by a line" (i.e. a schematic "domino chip"), 14 per cent of the time, or "three rectangles" 2 percent of the time.

Pattern B: 61 responses were given, 42 of them unambiguous. Pattern B was found to represent "two overlapping rectangles" 59 percent of the time, and "three rectangles" 10 percent of the time.

Pattern C: 62 responses were given, 52 of them unambiguous. Pattern C was alternatively interpreted as "four adjacent rectangles" 63 percent of the time, or as "five rectangles" 13 percent of the time. The only other interpretation of "a rectangle divided twice" (i.e. a schematic "window frame" pattern) was offered 8 percent of the time.

In calculation of the above percentages, the total numbers of responses were used (including the ambiguous responses). This approach tends to lower the percentage values. It is remarkable that the answers to a question so ambiguous, and posed out of context, were so consistent. After all, the subjects of the study were shown only a laser printed page containing the three patterns. Any laser printout consists solely of a particular arrangement of black dots. Anything else we see in it is merely a figment of our imagination.

The fundamental problems studied in this paper are:

- Why do humans appear here to be so consistent, when selecting from a multitude of possible interpretations?
- Is it possible to construct a computer program showing similar interpretation preferences?

Indeed, one could argue that there are infinitely many interpretations of each pattern. For example, one could construct infinitely many interpretations even of a single line segment. Such line segment can be seen as two (or infinitely many) line segments, one being an extension of another. Or, a line segment can be viewed as a collection of (possibly infinitely many) partially or completely overlapping line segments. A mixture of both views is possible too. In this light, the ability for humans to come up with so few interpretations strikes the author as remarkable.

OTHER POSSIBLE PATTERN INTERPRETATIONS

In order to offer answers to the above, let us first provide a (grossly incomplete) list of possible interpretations. In the graphs that follow, the line segments are depicted as arrows directed from the starting point to the terminating point. Polygons are identified by an appropriate fill.

Below (Fig.1) are some possible interpretations of pattern A, starting from those most frequently mentioned. The number of each interpretation is beside the bottom right corner of its graph.

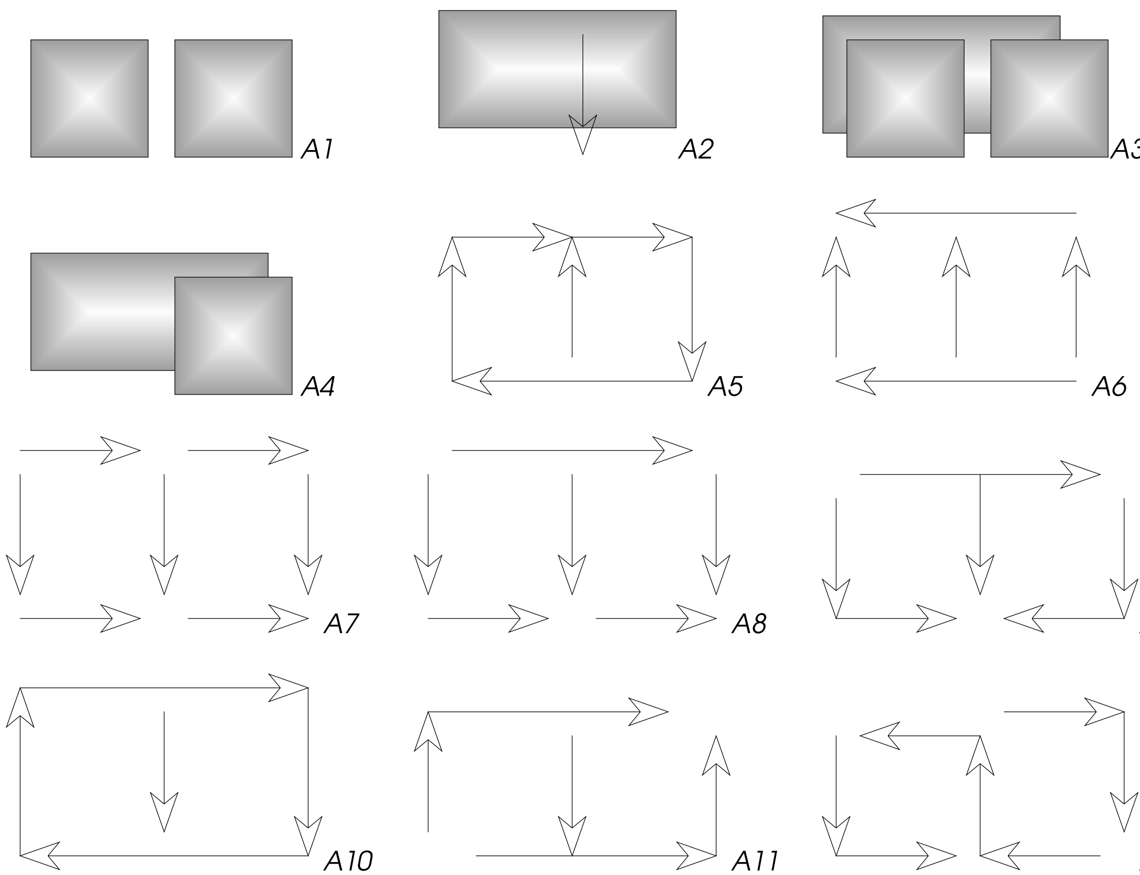


Fig. 1. Some possible interpretations of pattern A.

The survey respondents seemed to prefer seeing polygons (typically rectangles or squares) wherever and whenever possible. Interestingly, no interpretation consisting solely of line segments was ever offered.

Similar is true for patterns B and C, despite of the fact that their increasing complexity facilitates the creation of purely linear interpretations. Consequently, we de-emphasize the interpretations of these patterns that would consists of line segments only.

Patterns B and C could then be interpreted as:

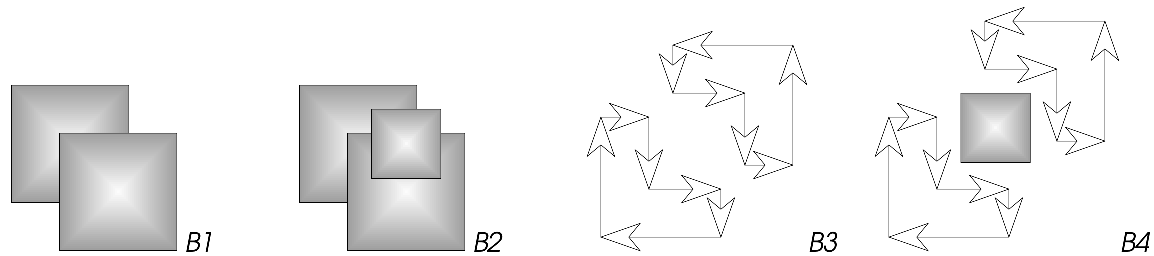


Fig. 2. Some possible interpretations of pattern B.

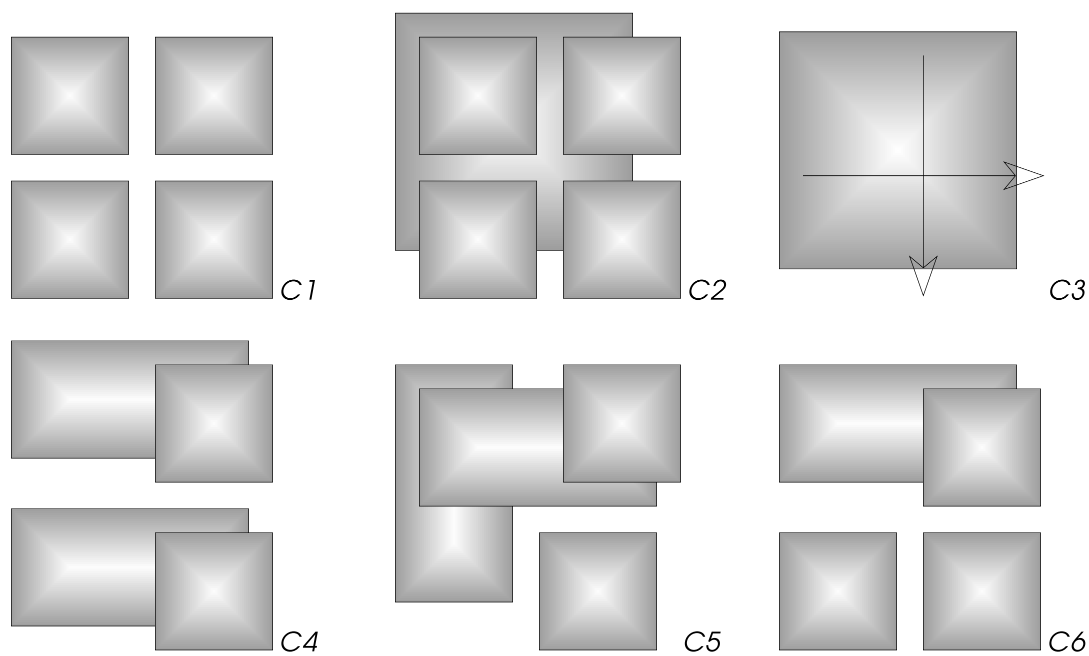


Fig. 3. Some possible interpretations of pattern C.

It appears that humans exhibit a preference in seeing things in a certain way, which, while helpful in solving problems most of the time, is likely to lower the chances of finding certain solutions. The puzzle at the end of this paper is a hopefully informative and entertaining case in point.

LINGO ONE: A LANGUAGE FOR DESCRIPTION OF INTERPRETATIONS

All respondents in the study were giving verbal descriptions of pattern interpretations. Let us simulate this process of translation of a given picture into a brief sequence of "words". To do this we need to design a simple language for description of pictures.

In our language a collection of objects will constitute a picture. Objects will be drawn in sequence, one after another. Each command to draw an object will consist of an information regarding an object kind (point, line segment, polygon, etc.), some numerical information regarding the shape of the object and its placement on the screen. Each command will end in a terminator.

To formalize this language definition, we will use a modified BNF notation. Specifically:

Metalanguage notation: each line specifies one construct of the language
vertical bars | separate construct alternatives
curly braces `{}` indicate one or more repetitions of their contents
`italics` indicate predefined types on the underlying computer
entities requiring further definition are shown like `<this>`
literals are within single quotes, like this `';'`

LINGO ONE **syntax definition:**

```
<picture> ::= <object> | <object> <picture>
<object>  ::= <kind> {<point>} ';'
<kind>    ::= byte
<point>   ::= integer integer
```

To reflect the current state of computing and graphics technology, we decide to use a byte to store the information regarding the kind of an object. This will permit us to draw up to 256 kinds of objects – more than enough, when compared to the current graphics packages.

In LINGO ONE a point is simply a pair of numbers (coordinates). We will assume that the size of an integer is two bytes (16 bits). This will allow us to take into account the resolution of current output devices, while being economical on RAM.

For the purpose of this paper, we will define only three kinds of objects:

Mnemonic	**Meaning**
`P`	point
`L`	line segment
`R`	rectangle

In a language so simply defined, only three commands, to draw points, rectangles, and sequences of line segments (i.e. broken lines), can be issued. To foster the economical use of memory, when drawing connected line segments we refrain from repeating twice the coordinates of the points where one line segment ends and another begins. Similarly, let us assume that rectangles represented by the mnemonic `R` have horizontal and vertical sides. (We could use other mnemonics for definition of rectangles with other, perhaps more general, orientations). Consequently, such rectangles can be fully defined by specifying the position of their diagonal corners.

Representing the integer values by the hash character '#', we therefore have:

LINGO ONE **table of commands:**

```
P # #;            -- draws a point
L # # {# #};      -- draws one or more line segments
R # # # #;        -- draws a horizontal/vertical rectangle
```

The spaces between the command mnemonics and integer values are added for the comfort of a human reader. They are absent in any machine program.

SIZE OF A DESCRIPTION: A PRELIMINARY CRITERION

With a language so defined, storing every command requires a given number of bytes. For example, a command to draw a point requires 6 bytes, while a command to draw a rectangle needs 10 bytes of storage. The command to draw a broken line, made of line segments is of variable length. To store a command drawing n segments we need $6 + 4n$ bytes.

Let us investigate the sizes of programs needed to draw various interpretations of patterns A, B and C. The programs in LINGO ONE are:

Program A1:

```
R # # # #; R # # # #;
```

Program A2:

```
R # # # #; L # # # #;
```

Program A3:

```
R # # # #; R # # # #; R # # # #;
```

Program A4:

```
R # # # #; R # # # #;
```

Program A5:

```
L # # # # # # # # # # # # # #;
```

Program A6:

```
L # # # #; L # # # #; L # # # #;
L # # # #; L # # # #;
```

Program A7:

```
L # # # #; L # # # #; L # # # #;
L # # # #; L # # # #; L # # # #;
L # # # #;
```

Program A8:

```
L # # # #; L # # # #; L # # # #;
L # # # #; L # # # #; L # # # #;
```

Program A9:

```
L # # # #; L # # # #;
L # # # # # #; L # # # # # #;
```

Program A10:

```
L # # # # # # # # # #; L # # # #;
```

Program A11:

```
L # # # # # #; L # # # # # #;
L # # # #;
```

Program A12:

```
L # # # # # #; L # # # # # #;
L # # # # # # # #;
```

Program B1:

```
R # # # #; R # # # #;
```

Program B2:

```
R # # # #; R # # # #; R # # # #;
```

Program B3:

```
L # # # # # # # # # # # # # #;
L # # # # # # # # # # # # # #;
```

Program B4:

```
L # # # # # # # # # # # # # #;
L # # # # # # # # # # # # # #;
R # # # #;
```

Program C1:

```
R # # # #; R # # # #;
R # # # #; R # # # #;
```

Program C2:

```
R # # # #; R # # # #; R # # # #;
R # # # #; R # # # #;
```

Program C3:

```
R # # # #;
L # # # #; L # # # #;
```

Program C4:

```
R # # # #; R # # # #;
R # # # #; R # # # #;
```

Program C5:

```
R # # # #; R # # # #;
R # # # #; R # # # #;
```

Program C6:

```
R # # # #; R # # # #;
R # # # #; R # # # #;
```

Table 1 shows sizes (in bytes) of the corresponding interpretations. Please observe that the interpretations offered by the survey respondents are among the most concise interpretations. Size of an interpretation appears to be a good preliminary criterion for weeding out the majority of "implausible" interpretations. It appears that humans have preference to "see" rectangles, as opposed to interpretations consisting solely of line segments.

	Interpretation Number											
	1	2	3	4	5	6	7	8	9	10	11	12
Pattern A	20	20	30	20	30	50	70	60	48	32	38	46
Pattern B	20	30	60	70								
Pattern C	40	40	30	40	40	40						

Table 1. Interpretation sizes (in bytes), using LINGO ONE.

We will attempt to be more specific, and to modify our elimination criterion in order to increase the winning chances for the interpretations offered by the participants in the survey.

THE EFFICIENCY OF AN INTERPRETATION

Human image processing occurs at the conscious and the unconscious level. Given that the human retinas consist of grids of sensors (rods and cones), all images we perceive are the interpretations of sets of pixels, being the stimulation patterns of our sensors.

We are consciously unaware of the processing done at the pixel level. It appears that at the conscious level we strive to interpret images in terms of the primitive interpretation components like lines, arcs, areas, shadings, textures, etc. The most primitive element of those is a segment of a straight line, which could be defined by specifying its two end points.

It appears that we intuitively strive for most concise and unambiguous interpretations. In fact, one could suggest a measure of efficiency e of an interpretation as

$$e = \frac{1}{s\,a}$$

where s is the size of the interpretation, and a is some measure of its ambiguity.

THE RANKING OF INTERPRETATIONS USING LINEAR AMBIGUITY

At the conscious level, we like to be able to account for all the interpretation components (lines, arcs, etc.) recognized in a given picture, and to do so in an unambiguous way. Ideally every such component should belong to one and only one recognized object.

To illustrate this idea, let us consider pattern A as an example. This pattern consists of three vertical line segments and of some number of horizontal line segments, depending on the interpretation. At the most primitive level, there are four horizontal line segments, but we might notice that a certain line segments appear to be perfect extensions of other segments, and prefer to lump them together. This is the matter of interpretation, though. Similarly, we may choose to consider certain groups of four line segments to be considered rectangles.

We would like to come up with an interpretation which traces each primitive line segment once, and only once. Let us calculate the efficiencies of the interpretations A1 and A2, using this criterion. Interpretation A1 traces 8 line segments (by visiting the middle segment twice), while the interpretation A2 traces 7 line segments, visiting each primitive line segment exactly once. The ambiguity of the interpretation A1 is therefore 8/7 = 1.14, while the ambiguity of A2 is 1. Both interpretations have the same size (see Table 1), therefore interpretation A2 is found to be more efficient than A1, according to this criterion.

In the same vein, patterns B and C consist of 12 primitive line segments each.

Let us call "linear ambiguity" the measure of ambiguity so defined. The following Table 2 offers the comparisons between various interpretations of patterns A, B, C, according to this criterion.

	Pattern A		
Int	size	ambiguity	effic.
A1	20	1.14	**0.04375**
A2	20	1.00	**0.05000**
A3	30	2.00	0.01667
A4	20	1.43	0.03500
A5	30	1.00	0.03333
A6	50	1.00	0.02000
A7	70	1.00	0.01429
A8	60	1.00	0.01667
A9	48	1.00	0.02083
A10	32	1.00	0.03125
A11	38	1.00	0.02632
A12	46	1.00	0.02174

	Patterns B and C		
Int	size	ambiguity	effic.
B1	20	1.00	**0.05000**
B2	30	1.33	**0.02500**
B3	60	1.00	0.01667
B4	70	1.33	0.01074
C1	40	1.33	**0.01875**
C2	40	2.00	0.01250
C3	30	1.00	**0.03333**
C4	40	1.67	0.01500
C5	40	1.67	0.01500
C6	40	1.50	0.01667

Table 2. Comparison of linear ambiguities and efficiencies of various interpretations, at conscious level, using LINGO ONE. Two winning interpretations of each pattern are shown in bold, the winner being in a shaded cell.

Interestingly, interpretation A4, although as concise as A1 or A2, is too ambiguous, and loses in competition. Purely linear interpretations, although unambiguous, suffer from excessive verbosity and similarly lose.

THE RANKING OF INTERPRETATIONS USING PIXEL AMBIGUITY

It may be argued that the linear ambiguity is an imperfect measure, because:

- It imposes equal penalty for each overlap of two line segments, regardless of the length of the overlap;
- The argument that there are infinite number of interpretations of a single line segment can be disputed on the grounds that line segments are already interpretations offered by our unconscious vision systems (seen the discrete structure of our retinas), rather than the reality we directly observe;
- At the unconscious level, we process information available to us in the form of stimulation of the sets of pixels, corresponding to the rods and cones on our retinas. In fact, our image processing happens both at the conscious and unconscious levels.

Let us attempt to analyze visual information processing, done by humans at the unconscious level. To do that, consider a vision system equipped with an artificial "retina" of the resolution (i.e. acuity) of p pixels per unit of measure. To preserve generality of our reasoning, this measure may be linear, like the number of dots per inch, being the printing resolution of a given device, or angular, as in the fields of view of a human eye or a CCD sensor of a TV camera.

The main line of our reasoning will remain unchanged. Now, at the unconscious level, we would like to be able to account for all the pixels of a given picture, and to do so in an unambiguous way. Ideally every pixel should belong to one and only one recognized object.

Let us assume that the heights of patterns A, B, and C are one, three and two units, correspondingly. Assuming that the thickness of the lines in the pattern is one pixel, and the patterns are drawn with resolution p, we can count the number of pixels used to draw each pattern.

Pattern A consists of seven primitive line segments of unitary length, therefore the total number of pixels used to draw it is $7p - 8$. We subtract 8 here to account for pixels visited more than once. The four corner pixels of the large rectangle were drawn twice each (4 extra pixels). The end pixels of the center line, dissecting the rectangle, were drawn three times each (as the ends of three line segments meeting there), so we count 4 extra pixels again. The total number of extra pixels drawn is 8, and by subtracting it from $7p$ we arrive at the total number of pixels needed to draw the pattern A.

In similar vein we conclude that patterns B and C are made of $12p - 14$ pixels and of $12p - 15$ pixels, correspondingly.

We will define pixel ambiguity as the ratio of the total number of pixels traced by a particular interpretation to the total number of pixels that a given pattern is made of. The lowest value of ambiguity measure so defined is 1, when each interpretation pixel is drawn exactly once.

A line segment of the length l is made of lp pixels, a rectangle of the perimeter r is made of rp pixels. The pixel ambiguity measures of various interpretations of patterns A, B, and C, follow.

$$a_{A1} = \frac{8p}{7p-8} \qquad a_{A2} = \frac{7p}{7p-8} \qquad a_{A3} = \frac{14p}{7p-8} \qquad a_{A4} = \frac{10p}{7p-8}$$

$$a_{A5} = a_{A6} = a_{A7} = a_{A8} = a_{A9} = a_{A10} = a_{A11} = a_{A12} = \frac{7p}{7p-8}$$

$$a_{B1} = \frac{16p}{12p-14} \qquad a_{B2} = \frac{20p}{12p-14} \qquad a_{B3} = \frac{16p}{12p-14} \qquad a_{B4} = \frac{20p}{12p-14}$$

$$a_{C1} = \frac{16p}{12p-15} \qquad a_{C2} = \frac{24p}{12p-15} \qquad a_{C3} = \frac{12p}{12p-15} \qquad a_{C4} = \frac{20p}{12p-15}$$

$$a_{C5} = \frac{20p}{12p-15} \qquad a_{C6} = \frac{18p}{12p-15}$$

Intuition suggests that what a vision system may see depends on the resolution (i.e. acuity) of its imaging device. In our approach, the lowest resolution to be able to "see" line segments is $p = 2$. In this case the vision system would be able to see merely two points and speculate that they might be the end points of a line segment.

The following Table 3 shows the ambiguities and efficiencies of the presented interpretations of the three patterns, at the unconscious level, with the crudest feasible resolution of $p = 2$.

	Pattern A		
Int	size	ambiguity	effic.
A1	20	2.67	**0.01875**
A2	20	2.33	**0.02143**
A3	30	4.67	0.00714
A4	20	3.33	0.01500
A5	30	2.33	0.01429
A6	50	2.33	0.00857
A7	70	2.33	0.00612
A8	60	2.33	0.00714
A9	48	2.33	0.00893
A10	32	2.33	0.01339
A11	38	2.33	0.01128
A12	46	2.33	0.00932

	Patterns B and C		
Int	size	ambiguity	effic.
B1	20	3.20	**0.01562**
B2	30	4.00	**0.00833**
B3	60	3.20	0.00521
B4	70	4.00	0.00357
C1	40	3.56	**0.00703**
C2	40	5.33	0.00469
C3	30	2.67	**0.01250**
C4	40	4.44	0.00562
C5	40	4.44	0.00562
C6	40	4.00	0.00625

Table 3. Comparison of linear ambiguities and efficiencies of various interpretations, at unconscious level, with the crudest resolution of $p = 2$, using LINGO ONE. Two winning interpretations of each pattern are shown in bold, the winner being in a shaded cell.

On the other extreme, Table 4 illustrates the limit values of the ambiguities and efficiencies, with the resolution approaching infinity.

	Pattern A		
Int	size	ambiguity	effic.
A1	20	1.14	**0.04375**
A2	20	1.00	**0.05000**
A3	30	2.00	0.01667
A4	20	1.43	0.03500
A5	30	1.00	0.03333
A6	50	1.00	0.02000
A7	70	1.00	0.01429
A8	60	1.00	0.01667
A9	48	1.00	0.02083
A10	32	1.00	0.03125
A11	38	1.00	0.02632
A12	46	1.00	0.02174

	Patterns B and C		
Int	size	ambiguity	effic.
B1	20	1.33	**0.03750**
B2	30	1.67	**0.02000**
B3	60	1.33	0.01250
B4	70	1.67	0.00857
C1	40	1.33	**0.01875**
C2	40	2.00	0.01250
C3	30	1.00	**0.03333**
C4	40	1.67	0.01500
C5	40	1.67	0.01500
C6	40	1.50	0.01667

Table 4. Comparison of linear ambiguities and efficiencies of various interpretations, at unconscious level, with the infinite resolution, using LINGO ONE. Two winning interpretations of each pattern are shown in bold, the winner being in a shaded cell.

Comparison of data from the preceding tables yields several conclusions, some more obvious than others:

- At the unconscious level, as the resolution of the "retina" increases, the ambiguity of the discerned pattern decreases;
- Even at infinite resolution, the interpretation ambiguity does not usually decrease to the ideal value of 1 because certain interpretations are inherently ambiguous, no matter what resolution;
- The winning interpretations seem to be insensitive to the resolution;
- The competition between interpretations, using conscious and unconscious criteria, seems to yield the same winners and runners-up;
- Certain conscious interpretations appear as limits to some unconscious interpretations at infinite resolutions: please observe that the numbers for pattern A in Tables 2 and 4 are identical.

The above conclusions may be interesting, but we have so far failed to construct a system that would exhibit "human" preferences when selecting from a multitude of possible pattern interpretations. Let us remedy this problem.

LINGO TWO: A SMALL MODIFICATION OF LINGO ONE

To achieve that, we will construct a language LINGO TWO, obtained through a small modification to LINGO ONE. We will leave unchanged the syntax and the command mnemonics. We will modify only the format of commands, viz.:

LINGO TWO **table of commands:**

```
P {# #};         -- draws one or more points
L # # {# #};     -- draws one or more line segments
R # # {# #};     -- draws one or more rectangles
```

This modification will permit us to draw sequences of rectangles sharing a common corners in the same way we used to draw broken lines as sequences of line segments sharing start/end points. For elegance we apply these changes to the command to draw points. We will, however, refrain from drawing points whenever possible (certainly in this paper!), because an interpretation involving free-standing points is likely to be too loquacious, thus inefficient.

The corresponding interpretations in LINGO TWO are:

Program A1:

```
R # # # # # #;
```

Program A2:

```
R # # # #; L # # # #;
```

Program A3:

```
R # # # # # # # #;
```

Program A4:

```
R # # # # # #;
```

Programs A5 to A12 remain as in LINGO ONE.

Program B1:

```
R # # # #; R # # # #;
```

Program B2:

```
R # # # # # # # #;
```

Programs B3 and B4 remain as in LINGO ONE.

Program C1:

```
R # # # # # # # # # #;
```

Program C2:

```
R # # # # # # # # # #; R # # # #;
```

Program C3:

```
R # # # #; L # # # #; L # # # #;
```

Program C4:

```
R # # # # # # # # # #;
```

Program C5:

```
R # # # # # # # #; R # # # #;
```

Program C6:

```
R # # # # # # # # # #;
```

THE RANKING OF INTERPRETATIONS USING LINGO TWO

Table 5 corresponds to Table 2 before, and depicts the comparison of sizes, linear ambiguities and interpretation efficiencies at conscious level, using LINGO TWO.

	Pattern A		
Int	size	ambiguity	effic.
A1	14	1.14	**0.06250**
A2	20	1.00	**0.05000**
A3	18	2.00	0.02778
A4	14	1.43	**0.05000**
A5	30	1.00	0.03333
A6	50	1.00	0.02000
A7	70	1.00	0.01429
A8	60	1.00	0.01667
A9	48	1.00	0.02083
A10	32	1.00	0.03125
A11	38	1.00	0.02632
A12	46	1.00	0.02174

	Patterns B and C		
Int	size	ambiguity	effic.
B1	20	1.00	**0.05000**
B2	18	1.33	**0.04167**
B3	60	1.00	0.01667
B4	70	1.33	0.01071
C1	22	1.33	**0.03409**
C2	32	2.00	0.01562
C3	30	1.00	**0.03333**
C4	22	1.67	0.02727
C5	28	1.67	0.02143
C6	22	1.50	0.03030

Table 5. Comparison of sizes, linear ambiguities and efficiencies of various interpretations, at conscious level, using LINGO TWO. The winning interpretations of each pattern are shown in bold, the winner being in a shaded cell.

Tables 6 and 7 correspond to Tables 3 and 4 before, demonstrating the comparison of sizes, linear ambiguities and interpretation efficiencies at unconscious level, now using LINGO TWO.

	Pattern A		
Int	size	ambiguity	effic.
A1	14	2.67	**0.02679**
A2	20	2.33	**0.02143**
A3	18	4.67	0.01190
A4	14	3.33	**0.02143**
A5	30	2.33	0.01429
A6	50	2.33	0.00857
A7	70	2.33	0.00612
A8	60	2.33	0.00714
A9	48	2.33	0.00893
A10	32	2.33	0.01339
A11	38	2.33	0.01128
A12	46	2.33	0.00932

	Patterns B and C		
Int	size	ambiguity	effic.
B1	20	3.20	**0.01562**
B2	18	4.00	**0.01389**
B3	60	3.20	0.00521
B4	70	4.00	0.00357
C1	22	3.56	**0.01278**
C2	32	5.33	0.00586
C3	30	2.67	**0.01250**
C4	22	4.44	0.01023
C5	28	4.44	0.00804
C6	22	4.00	0.01136

Table 6. Comparison of linear ambiguities and efficiencies of various interpretations, at unconscious level, with the crudest resolution of p = 2, using LINGO TWO. The winning interpretations of each pattern are shown in bold, the winner being in a shaded cell.

	Pattern A		
Int	size	ambiguity	effic.
A1	14	1.14	**0.06250**
A2	20	1.00	**0.05000**
A3	18	2.00	0.02778
A4	14	1.43	**0.05000**
A5	30	1.00	0.03333
A6	50	1.00	0.02000
A7	70	1.00	0.01429
A8	60	1.00	0.01667
A9	48	1.00	0.02083
A10	32	1.00	0.03125
A11	38	1.00	0.02632
A12	46	1.00	0.02174

	Patterns B and C		
Int	size	ambiguity	effic.
B1	20	1.33	**0.03750**
B2	18	1.67	**0.03333**
B3	60	1.33	0.01250
B4	70	1.67	0.00857
C1	22	1.33	**0.03409**
C2	32	2.00	0.01562
C3	30	1.00	**0.03333**
C4	22	1.67	0.02727
C5	28	1.67	0.02143
C6	22	1.50	0.03030

Table 7. Comparison of linear ambiguities and efficiencies of various interpretations, at unconscious level, with the infinite resolution, using LINGO TWO. The winning interpretations of each pattern are shown in bold, the winner being in a shaded cell.

This time we were able to identify the winning interpretations according to human preferences. More importantly, armed with the preceding reasoning, we are able now to make a number of observations:

- Two vision systems of identical architecture and "retina" resolution, using the same interpretation language but different interpretation elimination criteria may arrive at different interpretations of the same pattern;
- A command in a typical interpretation language is made of two parts. One part is of fixed size (in bits), for example the command mnemonic and terminator; the other part is of variable bit size, for example a sequence of numbers making the body of the command. Therefore, a command size (in bits) may vary. Moreover, we may choose different number of bits to stand for the mnemonic, terminator, and each number in the sequence. Therefore, if by architecture we mean a particular choice of the number of bits needed to represent different these entities, we have:
  Two vision systems of different architecture, but of the same retina resolution, pattern interpretation language and elimination criterion, may arrive at different interpretations of the same pattern;
- Two vision systems of the same architecture and retina resolution, using the same elimination criterion but equipped with different interpretation languages may arrive at different interpretations of the same pattern.

The author finds the last observation particularly striking. Humans, treated as vision systems, seem to share the same architecture (nervous system and sensors). Through education they seem to disseminate the same elimination criteria. They are, however, determined to use different languages, under the notion of "cultural heritage". No wonder that they usually arrive at different interpretations of reality.

The Canadian experience of a mild friction between Anglophones and Francophones is a case in point. The author (who is neither an Anglophone nor Francophone, but fluent in both languages) finds that the French version of a given text contains typically more characters than an English version. Assuming that both versions are equally ambiguous, the shorter version is more efficient.

This is not to say that English is more efficient than French, but merely that certain texts in English are more efficient than their French counterparts.To illustrate that point, let us briefly compare English to Inuktutut, the language of the Inuit in the Arctic. In Inuktutut, there are more than twenty expressions describing different kind of snow, but not many expressions describing different legal concepts. Humans use the language as a survival tool. Twenty or more terms for snow may be an inefficient verbosity in southern Canada, but may be a survival necessity in the Arctic. The converse is true for legal terms.

In fact, there is ample evidence that languages evolve. Evolution amounts to a sequence of small changes to the language definition. In this paper we have arrived at LINGO TWO via a minor change to LINGO ONE. As illustrated, minor changes in languages may lead to qualitative changes in interpretations of reality. The process of evolution tends to eliminate vision systems that frequently come up with "wrong" interpretations.

In fact, it is arguable that all animals (or certainly vertebrates) possess linguistic capabilities, because they seem to be able to interpret the images. (An interpretation is any exercise of lumping together a number of dots, a language being any mechanism of expressing interpretations). Linguistic capabilities should not be confused with speech. Speech is merely one mechanism of linguistic expression. The sign language of the mute is another. Bees communicate between themselves through a certain "dance".The author has no difficulty communicating with his dog, to give another example.

It is a matter of further discussion to decide, whether the above definition of the linguistic capabilities is a highly specialized one, or the most general.

ANOTHER EFFICIENT INTERPRETATION

The problem how to generate the candidate interpretations is outside the scope of this paper. The author finds it interesting, however, that no respondent offered the following interpretation of pattern C.

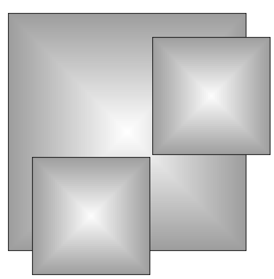


Fig. 4. Another winning interpretation of pattern C.

This interpretation would be a winner in the LINGO TWO competition, had it been offered by the survey respondents. In fact, it took the author some time to arrive at this interpretation.

ON INTELLIGENT INTERPRETATION OF GRAPHS

The presented reasoning, while applicable to the general vision systems, may be of more practical and immediate relevance to a particular class of expert systems, capable of understanding technical drawings.

Imagine a vision system equipped with, say, LINGO TWO, in which command mnemonics, instead of identifying points, lines and rectangles, would refer to wires, resistors, capacitors, coils, transistors, diodes, etc. Such a system, when presented with a drawing of an electronic circuit, could arrive at its most efficient interpretation. After an additional dialogue in which its queries, pertaining to the particular values of resistance, capacitance, etc., were satisfied. The system would be then able to "understand" and to simulate the behaviour of electronic circuits.

Moreover, the same system could be used in the fields of mechanical or civil engineering, if its command mnemonics were updated correspondingly.

ON INTELLIGENT IMAGE RETOUCHING

The possibility of another class of devices emerges here as well. In order to explain how they would work, let us consider the following problems:

- How does a photographer know how to retouch a slightly damaged negative, which s/he has never seen before?
- Suppose that you watch a TV program, the reception being particularly bad ("snow" in the picture). How do you know which parts of the image constitute the transmitted information, and which are the noise artifacts of poor reception?

Consider initially a fictitious species of intelligent vision systems, inhabiting an imaginary world made of points, line segments and rectangles. The most primitive of these is the point. In this paper till now we did not use points at all, when offering the interpretations of patterns A, B and C. Point interpretations would be grossly inefficient, being too verbose.

Initially, however, our vision systems would use only these most primitive, pointwise interpretations. Eventually, through an evolutionary mutation, some of them would notice that certain groups of points could be lumped together to be considered as line segments. Then, broken lines, being the chains of line segments would be discovered. Finally the discovery of rectangles would be made. At this stage the concept of points could be abandoned, assuming that there would be no noise in the picture.

With noise present, the situation is a little different. There could always be some free-standing points of in the picture. For each of them, the question would arise: is it a noise artifact, or is it "real"?

Humans did evolve in a different environment, made of grass, stones, trees, sky, water, etc. These are the initial elements of our image interpretation languages. As said before, the efficiency of a particular image interpretation could be described by the formula:

$$e = \frac{1}{s\,a}$$

where s is the size of the interpretation, and a is some measure of its ambiguity.

The highest value of e is for an image consisting of a single point. (Such image is unambiguous, and its representation in any LINGO is the smallest possible). Let us call this value e_0. Let us normalize the values of the efficiency, viz.

$$\eta = \frac{e}{e_0}$$

The highest possible value for η is 1, when only one point on a uniform background is seen. The lowest possible value is some η_{min}, when all the pixels on the retina must be accounted for individually (as in a picture on a TV set with no antenna). In fact for arbitrary pictures the values of η can be seen as particular values of the random variable H, of some distribution, which depends on the images seen, the interpretation language, and the elimination criterion. Whatever the distribution of this random variable, the following property holds:

$$\eta_{min} \leq H \leq 1$$

Should the above inequality be violated, the vision system would "know" that one of the following conditions occurred:

- its eyes are closed, or
- its lens cap has not been removed, or
- there is not enough light to rely on the vision system, or
- its vision hardware failed.

In fact, the typical values of the random variable H, for a vision system inhabiting a particular environment fall within a confidence interval that lies inside the interval [η_{min}, 1].

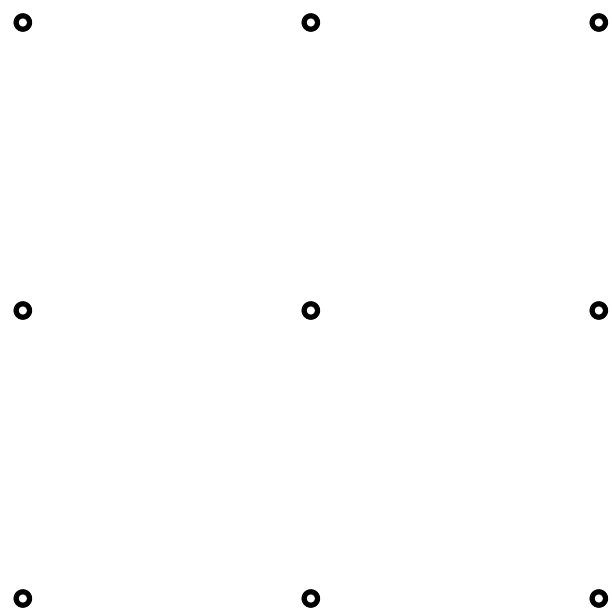
Obviously, the more complex the picture, the lower value of the random variable H. We might want to construct vision systems, which would attempt to interpret the images so that the first element recognized would account for the most pixels of the image, the second element the next most pixels of the image, etc. Having recognized each element, the system would compute the value H established so far, making sure that the value H remains within the confidence interval. Should any pixels remain, they would be dubbed "noise" and ignored.

ACKNOWLEDGMENTS

The author is indebted to Jim Bradford for his subtle but firm encouragement, to Tom Jenkyns for his idea of modifying LINGO ONE to arrive at LINGO TWO, and to Vladimir Tucakov for his help in conducting the survey and for the contribution of the following puzzle.

CONCLUDING PUZZLE

To illustrate the fact that the uncanny ability of humans to interpret raster images may sometimes hinder our problem solving abilities, please solve the following puzzle:



Given nine points as above, connect them all with a broken line made of four line segments. Please give this problem a fair moment of your time. Should you fail to find the solution, it is

contained in the Postscript® file named: **`puzzle.ps`**
stored in the directory: **`/usr/local/gopher/ftp/usr/vwojcik`**
on the computer named: **`sandcastle.cosc.brocku.ca`**

After downloading this file (using anonymous FTP, GOPHER, etc.) you may print in on any Postscript® printer, connected to almost any computer. For example, on `sandcastle`, a typical UNIX machine, the command of the form **`lp puzzle.ps`** will do the job.

APPENDIX: Detailed results of the study

Pattern A.
In the interpretation column, a square, being a special case of a rectangle, is called a rectangle. The minority interpretations are shown in **bold**.

Seq	Response	Geometric interpretation
1	window	2 adjacent rectangles
2	2 boxes beside each other	2 adjacent rectangles
3	2 squares attached to each other	2 adjacent rectangles
4	an open book	2 adjacent rectangles
5	2 equal squares joined together evenly...	2 adjacent rectangles
6	2 squares	2 adjacent rectangles
7	a rectangle cut in half	**a rectangle divided**
8	basement window	2 adjacent rectangles
9	something akin to a corner viewed straight on	2 adjacent rectangles
10	two squares	2 adjacent rectangles
11	domino	**a rectangle divided**
12	two squares side-by-side, sharing an edge...	2 adjacent rectangles
13	2 squares forming a rectangle	2 adjacent rectangles
14	two squares	2 adjacent rectangles
15	2 squares side by side	2 adjacent rectangles
16	book / magazine	2 adjacent rectangles
17	book (open)	2 adjacent rectangles
18	two equal sized squares that are attached	2 adjacent rectangles
19	2 squares	2 adjacent rectangles
20	two squares side-by-side forming a rectangle	2 adjacent rectangles
21	two squares	2 adjacent rectangles
22	2 squares & 1 rectangle	2 adjacent rectangles
23	equal	<ignored: too ambiguous>
24	two adjoining squares of equal dimensions	2 adjacent rectangles
25	2 squares adjacent (sharing a side)	2 adjacent rectangles
26	open book	2 adjacent rectangles
27	a rectangle cut in half	**a rectangle divided**
28	2 squares & 1 rectangle	2 adjacent rectangles
29	2 boxes	2 adjacent rectangles
30	window	2 adjacent rectangles
31	rectangle	**a rectangle divided**
32	windows	2 adjacent rectangles
33	domino	**a rectangle divided**
34	2 squares side by side	2 adjacent rectangles
35	two halves of a basketball court or a soccer field	2 adjacent rectangles
36	2 squares	2 adjacent rectangles
37	2 squares that have a common side	2 adjacent rectangles
38	2 squares	2 adjacent rectangles
39	two squares	2 adjacent rectangles
40	2 squares directly beside each other	2 adjacent rectangles
41	2 squares beside each other	2 adjacent rectangles

42	2 squares	2 adjacent rectangles
43	2 squares	2 adjacent rectangles
44	2 boxes side by side	2 adjacent rectangles
45	domino	**a rectangle divided**
46	two boxes side by side to form a rectangle	2 adjacent rectangles
47	two binded(?) square window	2 adjacent rectangles
48	2 squares together	2 adjacent rectangles
49	a rectangle	**a rectangle divided**
50	2 sides of a box seen with edge in middle	2 adjacent rectangles
51	two squares touching	2 adjacent rectangles
52	two squares	2 adjacent rectangles
53	2 squares	2 adjacent rectangles
54	eye	<ignored: too ambiguous>
55	two squares	2 adjacent rectangles
56	book	2 adjacent rectangles
57	2 squares	2 adjacent rectangles
58	two squares sharing one side	2 adjacent rectangles
59	two squares	2 adjacent rectangles
60	one rectangle	**a rectangle divided**
61	2 squares inside a square	**3 rectangles**
62	2 squares making up a square	2 adjacent rectangles
63	two squares side by side	2 adjacent rectangles
64	a rectangle with intersecting line	**a rectangle divided**

Pattern B.
As before, in the interpretation column, a square, being a special case of a rectangle, is called a rectangle. The minority interpretations are shown in **bold**.

Seq	**Response**	**Geometric interpretation**
1	two panes of glass, one hanging in front of the other	two overlapping rectangles
2	clear window offset from a dark background	two overlapping rectangles
3	2 squares overlapping each other by a corner	two overlapping rectangles
4	two squares intersecting	two overlapping rectangles
5	two equal squares overlapped	two overlapping rectangles
6	inter-layed tiles	<ignored: too ambiguous>
7	a square sitting on top of another one	two overlapping rectangles
8	overlapping areas (sprite collisions)	two overlapping rectangles
9	two overlapping panels of glass	two overlapping rectangles
10	2 sqrs (overlapped)	two overlapping rectangles
11	cube	<ignored: too ambiguous>
12	two squares on top of each other	two overlapping rectangles
13	box	<ignored: too ambiguous>
14	two overlapping squares	two overlapping rectangles
15	one square in front of another	two overlapping rectangles
16	icon in computer to make a screen bigger	<ignored: too ambiguous>
17	stairs / steps	<ignored: too ambiguous>
18	two equal squares with overlapping corners	two overlapping rectangles
19	3 squares	**three rectangles**

20	two squares overlapping to form a smaller square	two overlapping rectangles
21	a cube	<ignored: too ambiguous>
22	3 squares	**three rectangles**
23	union	<ignored: too ambiguous>
24	two squares with a 25% corner overlap	two overlapping rectangles
25	a square with a square overlapping the bottom right corner	two overlapping rectangles
26	cube	<ignored: too ambiguous>
27	two squares with parts overlapping	two overlapping rectangles
28	3 squares	**three rectangles**
29	3 boxes	**three rectangles**
30	kite	<ignored: too ambiguous>
31	box	<ignored: too ambiguous>
32	origami	<ignored: too ambiguous>
33	multiple squares	<ignored: too ambiguous>
34	4 squares in box formation	<ignored: contradictory>
35	two boxes joined by a common bond	two overlapping rectangles
36	3 squares	**three rectangles**
37	2 overlapping squares	two overlapping rectangles
38	2 squares overlapping	two overlapping rectangles
39	2 overlapping squares	two overlapping rectangles
40	2 squares one on top of each other	two overlapping rectangles
41	2 overlapping squares	two overlapping rectangles
42	3 squares	**three rectangles**
43	1 square with lines leading up to it	<ignored: too ambiguous>
44	2 square 1 laying on top of the other	two overlapping rectangles
45	beginnings of the drawing of a cube	<ignored: too ambiguous>
46	two big boxes which overlap to form a smaller box	two overlapping rectangles
47	one front and one rear window in some dimension	two overlapping rectangles
48	a square behind another square	two overlapping rectangles
49	plaid	<ignored: too ambiguous>
50	square overlapping another square	two overlapping rectangles
51	two overlapping squares	two overlapping rectangles
52	box	<ignored: too ambiguous>
53	2 overlapping squares	two overlapping rectangles
54	stone	<ignored: too ambiguous>
55	two squares with one overlapping the other	two overlapping rectangles
56	what a 1 square looks when I'm drunk	<ignored: too ambiguous>
57	2 squares intersecting generating a third square	two overlapping rectangles
58	overlapping squares	two overlapping rectangles
59	overlapping squares	two overlapping rectangles
60	3 squares as a result of a combination of 2 squares	two overlapping rectangles
61	two intersecting squares which make another square	two overlapping rectangles

Pattern C.
As before, in the interpretation column, a square, being a special case of a rectangle, is called a rectangle. The minority interpretations are shown in **bold**.

Seq	Response	Geometric interpretation
1	another window, this one larger than the first	4 adjacent rectangles
2	storm window frame	4 adjacent rectangles
3	1 square divided into 4 sections	**a rectangle divided twice**
4	a window	4 adjacent rectangles
5	4 equal squares together to form one big square	4 adjacent rectangles
6	the court for a game of 4 caners (4 square)	4 adjacent rectangles
7	a big square cut into four equal parts	**a rectangle divided twice**
8	positioner from forms design	<ignored: too ambiguous>
9	a window?	4 adjacent rectangles
10	four squares inside a square	**5 rectangles**
11	small bathroom tile	<ignored: too ambiguous>
12	four squares making a big square	4 adjacent rectangles
13	square	<ignored: too ambiguous>
14	one square and two intersecting lines	**a rectangle divided twice**
15	4 squares	4 adjacent rectangles
16	window	4 adjacent rectangles
17	window	4 adjacent rectangles
18	4 equal squares attached to form a large square	4 adjacent rectangles
19	4 square game	4 adjacent rectangles
20	4 squares forming a larger square	4 adjacent rectangles
21	a tower	<ignored: too ambiguous>
22	5 squares & 4 rectangles	<ignored: too ambiguous>
23	divided	<ignored: too ambiguous>
24	a single large square divided into four equal pieces	**a rectangle divided twice**
25	4 squares adjacent to each other to form 1 larger sq.	4 adjacent rectangles
26	window frame	4 adjacent rectangles
27	large square cut into four squares	**a rectangle divided twice**
28	5 squares	**5 rectangles**
29	5 boxes	**5 rectangles**
30	window	4 adjacent rectangles
31	square	<ignored: too ambiguous>
32	tile	<ignored: too ambiguous>
33	window	4 adjacent rectangles
34	2 squares with overlapping corners	<ignored: contradictory>
35	elementary school game that had four squares...	4 adjacent rectangles
36	5 squares	**5 rectangles**
37	4 squares window	4 adjacent rectangles
38	5 total squares	**5 rectangles**
39	four connecting squares	4 adjacent rectangles
40	4 boxes forming one larger square	4 adjacent rectangles
41	4 squares connected together	4 adjacent rectangles
42	5 squares	**5 rectangles**
43	4 squares	4 adjacent rectangles
44	4 boxes arranged in a grid format	4 adjacent rectangles

45	chessboard fragment	4 adjacent rectangles
46	four boxes which create one big box	4 adjacent rectangles
47	four squared window	4 adjacent rectangles
48	4 squares together	4 adjacent rectangles
49	window	4 adjacent rectangles
50	4 squares	4 adjacent rectangles
51	four squares touching	4 adjacent rectangles
52	window	4 adjacent rectangles
53	4 squares	4 adjacent rectangles
54	window	4 adjacent rectangles
55	two of pattern A. <ambiguous?> and/or 4 squares	4 adjacent rectangles
56	target screen	<ignored: too ambiguous>
57	I square containing 4 smaller squares	**5 rectangles**
58	four linked squares	4 adjacent rectangles
59	window	4 adjacent rectangles
60	4 squares making up 1 large square	4 adjacent rectangles
61	four squares	4 adjacent rectangles
62	one big one [square] with many squares inside	**5 rectangles**