Perception & Psychophysics
1993, 53 (4), 403-421

Evaluating a computational model of perceptual grouping by proximity

BRIAN J. COMPTON and GORDON D. LOGAN
University of Illinois, Urbana-Champaign, Illinois

A formal approach to the phenomenon of perceptual grouping by proximity was investigated. Grouping judgments of random dot patterns were made by the CODE algorithm (van Oeffelen & Vos, 1982) and several related algorithms, and these judgments were compared with subjects' grouping judgments for the same stimuli. Each algorithm predicted significantly more subject judgments than would be expected by chance. The more subjects agreed on how a given dot pattern should be grouped, the more successful was the algorithms' ability to match the judgments for that pattern. CODE predicted significantly fewer subject judgments than did some of the other algorithms, largely because of its overemphasis on the extent of interactivity among dots as they are being grouped.

Gestalt laws of perceptual grouping, including grouping by proximity, were initially put forth as arguments against the prevailing theories of perception at the time. In Helmholtz's atomistic sensory theory, "each fundamental point sensation was taken to be independent of its neighbors" (described in Hochberg, 1981, p. 259). The Gestaltists, on the other hand, argued that the perceptual system does not simply collect and combine incoming sensory information to give a picture of the world, but instead actively organizes it. The law of grouping by proximity dictates that "when the [stimulus] field contains a number of equal parts, those among them which are in greater proximity will be organized into a higher unit," which "must be considered as real as the organization of a homogeneous spot" (Koffka, 1935, pp. 164-165). This "higher unit" is the product of a perceptual process that actively imposes structure on sensory input. It is an interpretation of a pattern's configuration, and it cannot be derived simply by examining the pattern's constituent parts in isolation.

Although the Gestaltists emphasized the holistic nature of perception, the actual computations dictated by some of the laws of grouping, including grouping by proximity, can conceivably be computed in a bottom-up fashion, using relatively local information. Pomerantz (1981) has suggested that grouping by proximity, similarity, and common fate could be computed by using algorithms that are purely data-driven, whereas grouping based on good figure, good continuation, or Prägnanz may require top-down processing.

This research was supported by Grants BNS 88-11026 and BNS 91-09856 to G.L. from the National Science Foundation. We would like to thank Liz Murphy and Jocelyn Weiss for entering data, and Gail Heyman and Michael Walker for help with analyses. Address correspondence concerning this article to B. J. Compton, Department of Psychology, University of Illinois, 603 East Daniel St., Champaign, IL 61820.

Palmer (1975) proposed a hierarchical model of perceptual representation, in which "structural units" at lower levels of a hierarchy encode the more specific details of an image and are themselves organized into more global structural units at higher levels in the hierarchy. This approach to perception highlights an important question for any theory of perceptual grouping by proximity: at what level in the "perceptual hierarchy" does grouping by proximity operate? Figure 1 provides one answer to this question. Does the viewer see two groups, one with 11 dots and the other with 10? Or are there six groups of either 3 or 4 dots each? Clearly, both configurations can be seen. Grouping by proximity can operate under strict grouping criteria, with only the closest elements being grouped together (e.g., seeing six groups of dots in Figure 1), or with looser grouping criteria (e.g., seeing two groups of dots in Figure 1). This suggests that a formal account of proximal grouping that specifies only one "correct" way to perceive a pattern will fail to capture an important aspect of the phenomenon.

A second question for any model of proximal grouping is the extent to which elements interact with other elements as they are being grouped. Can a single element somehow influence the way relatively remote elements are grouped, or does each element only influence its very nearest neighbors? The Gestalt position does not rule out the notion of purely local computations, but it raises the possibility that relatively distant elements of the stimulus field could interact in some way as they are grouped. Any model of proximal grouping will need to consider how great an effect relatively distant regions of a stimulus field have on each other in determining perceptual groups.

The role of proximity in perceptual grouping can perhaps be studied best with the use of dot patterns as stimuli. Each element in a dot pattern differs from the others by only one attribute, that of position. Dot pattern stimuli allow the effects of proximal grouping to be studied in isolation, free from any influence of grouping based

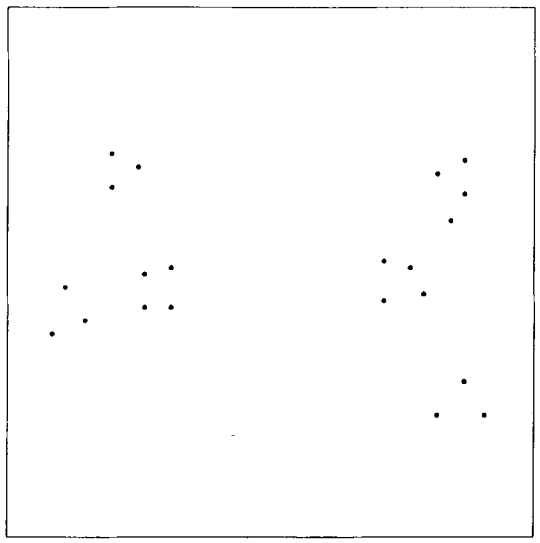

Figure 1. This pattern can be seen as containing two groups, one with 11 dots and one with 10, or as six groups of 3 or 4 dots each.

on differences of shape, size, orientation, or color among individual stimulus elements.

The evidence initially presented in support of the Gestalt laws of perception was limited to phenomenological demonstrations (Attneave, 1950). Although the validity of the Gestalt principles was intuitively apparent to observers, formal descriptions of the mechanisms underlying them were lacking. Formal models of grouping principles provide terminology for discussing and evaluating potentially different instantiations of the principles. Also, when a grouping principle is stated in formal terms, its validity can be assessed by systematically comparing its interpretations of stimuli against subjects' interpretations of those stimuli.

Few formal models of proximal grouping have been proposed. This is perhaps surprising, since grouping by proximity is one of the most well known and intuitively appealing of the Gestalt laws. One such formal model, the CODE algorithm, has recently been put forth by van Oeffelen and Vos (1982). The CODE algorithm is a purely bottom-up approach to perceptual grouping, since it assumes that all information needed to form groups is available in the stimulus itself. Its grouping mechanism is invariant across changes in scaling or rotation of the stimulus pattern.

Van Oeffelen and Vos (1982) initially tested the validity of CODE in an experiment in which subjects were shown stimuli consisting of clusters of dots for 100 msec and estimated the number of clusters of dots that they saw. The CODE algorithm agreed with the subjects' estimations more than 80% of the time when fewer than five clusters were presented, and less frequently for stimuli containing more than five clusters. The CODE algorithm has been used as a predictor of numerosity judgments (Allik & Tuulmets, 1991; Vos, van Oeffelen, Tibosch, & Allik, 1988) and has served as the basis of algorithms that attempt to account for other grouping principles besides proximity (Smits, Vos, & van Oeffelen, 1985; Vos & Helsper, 1991).

Although the CODE algorithm goes beyond the phenomenological demonstrations of the Gestalt psychologists in presenting a formal account of proximal grouping, a similarity in the two approaches remains. The proximity principle was first supported by demonstrations in which the stimuli were contrived to maximize the phenomenon of interest. Such stimuli were frequently arranged in clusters in such a way that the groups that were intended to be perceived were very obvious. The CODE algorithm, while providing a specific mechanism for proximal grouping phenomena, was tested with the use of stimuli that were contrived to contain distinct clusters.

Many formal accounts of proximal grouping can be imagined that might effectively group stimuli that contain dots arranged in tight clusters. These types of stimuli are more likely on the average to be considered "good" (which has been variously defined as simple, symmetrical, predictable, stable, and redundant; see Koffka, 1935; see also Garner, 1970, 1974), in comparison with stimuli containing dots whose positions are chosen at random. Since many models of proximal grouping are likely to be successful when goodness is high, it is the ambiguous cases, in which goodness is low, that should provide the most rigorous test of a model's behavior. In the present study, we addressed this issue by using completely random dot patterns as stimuli.

The CODE algorithm specifies a single organization for a given pattern, which may be more appropriate when stimuli are designed to have clusters, as opposed to being generated randomly. Some stimuli can be organized in several different ways, each essentially equivalent in goodness. This will occur frequently when the stimuli are random dot patterns, rather than patterns designed to suggest a particular organization. To accommodate the possibility that there can be several good organizations of a stimulus, in the present study we used variations of CODE that, among other things, are less constrained and permit a range of different organizations of proximal groups for a single stimulus.

In the present study, random dot patterns were presented to subjects who indicated how they should be grouped. These grouping judgments were then compared with grouping judgments for the same stimuli made by the CODE algorithm and related algorithms. The CODE algorithm was modified in two major ways. First, assumptions about the proper strength of grouping (the level in the "perceptual hierarchy" at which proximal grouping operates) were relaxed to produce a less constrained version of CODE, which allowed a single stimulus to be grouped in several different ways. Second, some of the design decisions of CODE were challenged by comparing the performance of CODE with a family of related grouping algorithms produced by considering alternatives to van Oeffelen and Vos's (1982) design decisions. These algorithms differ

in the degree to which elements interact with each other as they are grouped, and in the specific mechanisms of interaction. Comparison among these different algorithms allowed the role of interactivity among stimulus elements in proximal grouping to be investigated. In the present context, *interactivity* will be used to denote grouping algorithms that use relatively more global information about the stimulus pattern (e.g., those that use the position of remote stimulus elements to determine the group membership of each element), and opposed to those algorithms that rely on more local information (e.g., those that determine group membership locally, free from influence by distant regions of the stimulus pattern).

The CODE Algorithm

In the CODE algorithm, each element in the stimulus array exerts an influence on its neighboring region. This effect is strongest on regions of the stimulus array closest to the element, and diminishes to nearly zero for remote regions of the stimulus array, with a strength gradient that follows the shape of the normal distribution. CODE represents this influence as a spread function in the shape of a normal distribution, which is centered on each element in the stimulus.

The standard deviation of each spread function is set to one half the distance between the element and its nearest neighboring element. Once the shape of the spread function is found, it is rescaled so that the height of its peak equals 1. The spread functions contributed by each element in the stimulus array are then summed, to create a strength gradient for the stimulus array as a whole. Figure 2a illustrates this process, as applied to a one-dimensional stimulus array consisting of four dots on a line (which are shown just below the x-axis, and labeled A, B, C, and D). The strength of grouping is represented by the y dimension. The individual spread function for each dot is represented by dotted lines, and the sum of the functions, the strength gradient curve, is represented by the solid line. Since the standard deviation of each spread function is defined as half the distance from the dot to its nearest neighbor, some of the spread functions have different dispersion values. For example, Dots A and B are mutual nearest neighbors, and thus have identical spread functions, but Dots C and D have different spread functions, since the nearest neighbor to C is B, whereas the nearest neighbor to D is C.

Since each spread function is rescaled to have a maximum height of 1, spread functions with differing standard deviations have differing areas under their curves. For example, the spread function of Dot D has a greater standard deviation than do the spread functions for the other dots, so once it is rescaled it contains a larger area under its curve. The rescaling of the spread functions was one of several design components of the CODE algorithm that were investigated in the present study. Figure 2b shows the spread functions and their sum for a variant of the CODE algorithm that does not rescale the individual spread functions before summing them. This variation of CODE produces spread functions that have different heights, but that all have identical areas (i.e., 1) under their curves. The present discussion of the CODE algorithm will make use of both the original version (with each spread function having a height of 1) and this variation (with spread functions having differing heights).

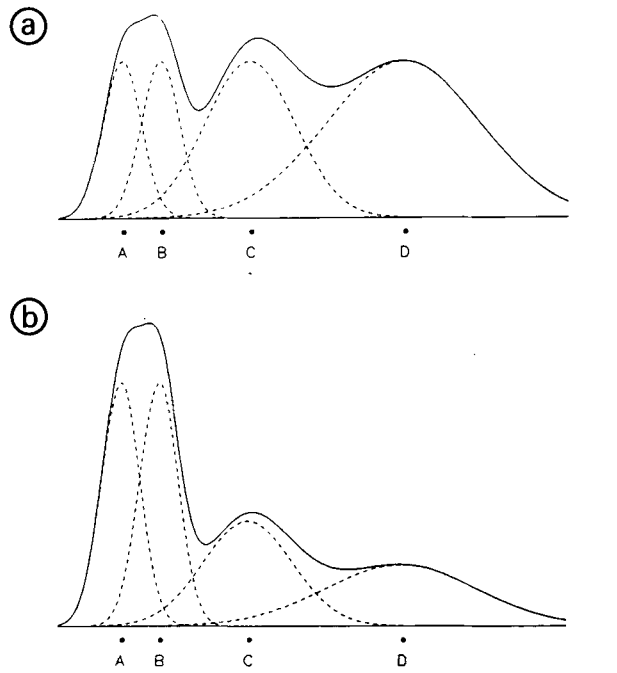

Figure 2. (a) Solid line shows the strength gradient for a one-dimensional dot pattern; dotted lines show underlying spread functions associated with each dot, which are rescaled to each have a height of 1. (b) Same as Figure 2a, but the functions are not rescaled.

The CODE algorithm operates analogously on a two-dimensional stimulus array, such as the one shown in Figure 3a. For a two-dimensional stimulus, the strength of grouping can be represented by the z dimension. Figure 3b shows the strength gradient (composed of individual spread functions that do not have rescaled peaks) for this stimulus as a surface. The surface is shown at a 45° counterclockwise rotation. The shortest peak, at the front of the surface, represents the combined strength of grouping for Dots E and F, whereas the tallest peak, at the right rear of the surface, represents Dots C and D.

According to CODE, grouping occurs when two or more elements that lie in proximity cause the strength of grouping for a region of the stimulus array to surpass a threshold value. When this occurs, all elements that are included within that region are identified as belonging to a single group. For the original CODE algorithm, this threshold is 1 (which is the height of the peaks before they are summed). Figure 4a shows this threshold as applied to the strength gradient seen in Figure 2b. This threshold indicates the set of groups {AB,C,D}. (A *set* of groups is defined as the configuration of groups that is specified by a single value of a threshold.)

While CODE specifies that a single threshold is used, the general approach is capable of generating multiple sets of groups for a single stimulus pattern, by varying

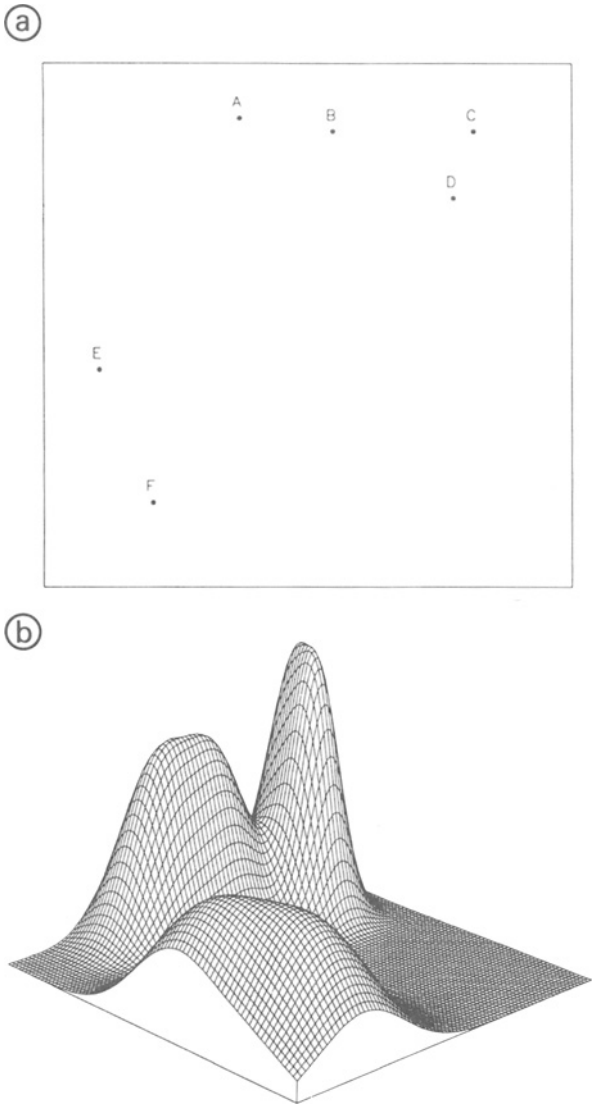

Figure 3. (a) A two-dimensional stimulus pattern. (b) Its strength gradient.

the threshold value. Figure 4b shows the different ways the stimulus seen in Figure 2b can be grouped when different threshold values are used. In this example, Threshold 1 specifies the set of groups {ABCD}; Threshold 2, {ABC,D}; Threshold 3, {AB,C,D}; and Threshold 4, {A,B,C,D}. Four sets of groups are shown, one specified by each threshold level.

Groups are identified in an analogous fashion for two dimensional stimuli. Figure 5a shows the effect of a threshold applied to the two-dimensional dot pattern shown in Figure 3a. The z values are truncated at the threshold to show the shapes of the regions that surpass the threshold. Only one threshold level is shown; it specifies the set of groups {AB,CD,EF}.

Figure 5b is a top view of the strength gradient surface for the same stimulus pattern. This figure shows the dots and the regions that result from the application of threshold values to the strength gradient surface. The regions carved out by six different threshold values are shown simultaneously in this figure. These six threshold values identify every set of groups that the algorithm can generate for this particular stimulus pattern. At each of the six threshold values, every region of the stimulus field that surpasses the threshold is shown. Threshold 1 has the lowest value on the z-axis, and Threshold 6 the highest. At Threshold 1, all dots are conglomerated into a single group, so the set of groups is {ABCDEF}. At Threshold 2, the set of groups is {ABCD,EF}; at 3, {AB,CD,EF}; at 4, {AB,CD,E,F}; at 5, {AB,C,D,E,F}; and at 6, {A,B,C,D,E,F}.

An informal exploration of the mechanics of the CODE algorithm revealed that there is a limit to the number of unique sets of groups that can be generated for any given stimulus pattern. The number of sets of groups that can be identified is always less than or equal to the number of dots that the stimulus pattern contains.[1] This limit on the number of sets of groups that can be generated by the CODE algorithm (or any of its variants discussed in this paper) will be referred to as the *algorithm constraint*.

Aim of Present Study

The purpose of the present study was to evaluate the ability of the CODE algorithm, in its original form and in several variations, to account for subjects' judgments of how a given stimulus array should be grouped. One goal of the present study was to determine under what conditions, if any, the CODE algorithm is an accurate predictor of subjects' grouping judgments. A second goal was to

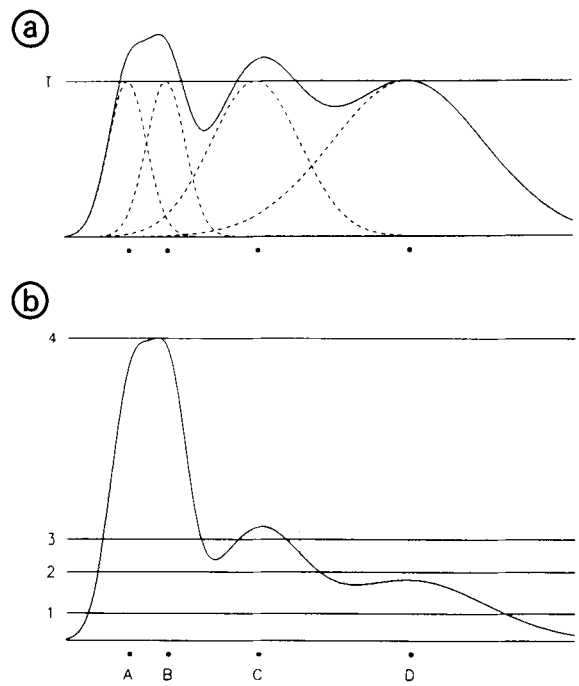

Figure 4. (a) A single threshold (labeled *T*) applied to the strength gradient seen in Figure 2a. (b) Four thresholds (labeled 1–4) applied to the strength gradient seen in Figure 2b.

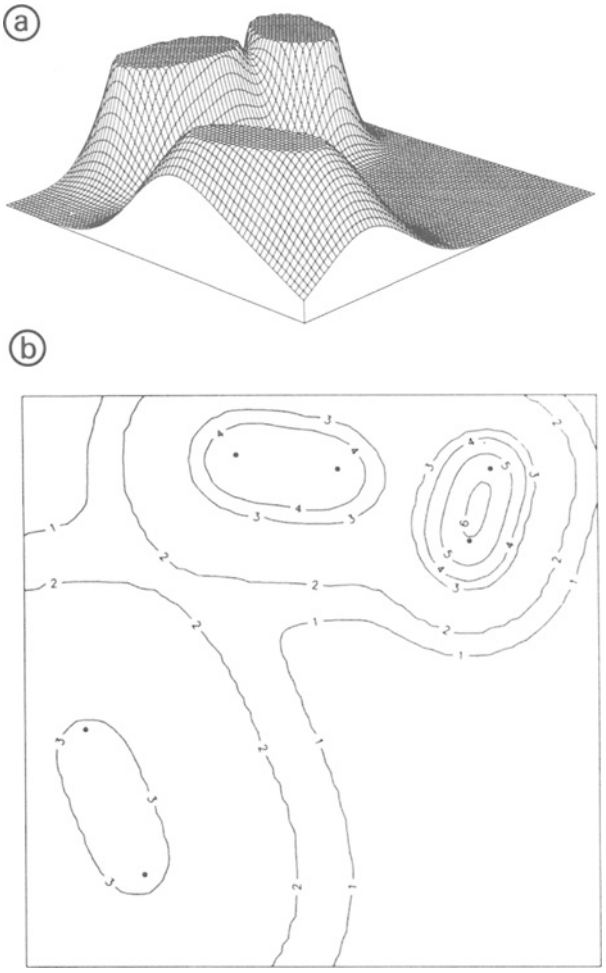

Figure 5. (a) A single threshold applied to the strength gradient seen in Figure 3b. (b) An overhead view of six thresholds (labeled 1-6) applied to the strength gradient seen in Figure 3b.

examine several of the processing assumptions of the CODE algorithm.

The five factors will now be described in detail. Although the stimuli used in the present experiment were all two-dimensional dot patterns, in our description of the different parameters we will make use of the one-dimensional dot pattern first seen in Figure 2b for purposes of illustration.

Rescaling Assumption

The CODE algorithm rescales individual spread functions so that their heights all equal 1. One variation of the CODE assumes that no such rescaling is done and that the height of each spread function is simply a function of its standard deviation. This variation leaves the area under each spread function equaling 1. This variation will be referred to as the *peaks* parameter. Figure 2b demonstrates *standard* peaks; Figure 2a, *rescaled* peaks. The examples in Figure 6a have standard peaks, while those in Figure 6b have rescaled peaks. When the peaks are not rescaled, each dot contributes a volume of 1 to the strength gradient surface. In contrast, when the peaks are rescaled to all have the same height, the volume each dot contributes to the strength gradient surface is a function of that dot's standard deviation. The purpose of the rescaling parameter was to determine what effect (if any) rescaling had on the sets of groups that are chosen by the algorithm for a given stimulus.

Intersection Assumption

The second processing assumption concerns the way in which the spread functions centered around each dot are combined to produce the strength gradient surface. The CODE algorithm handles overlap between the spread functions associated with different dots by summing their values, so that the intersection of two or more spread functions is additive. Alternatively, the algorithm could assume that at the intersection of two or more spread functions, the surface contour is determined by the spread function that has the greatest value for that point, essentially superimposing spread functions, rather than adding them. This variation will be referred to as the *intersection* parameter.

When additive intersections are used, the contributions of each spread function to a given point are combined. This is a form of interactivity that is not present when maximum intersections are used, since in the maximum intersections case only the spread function that makes the greatest contribution to a given point has any influence in determining the group membership of a dot located at that point.

In algorithms in which the intersection parameter was set to *add*, the surface contour at each x, y location was determined by summing the contributions of all spread functions in the stimulus array. When the intersection was set to *max*, the surface contour at each x, y location was set to the spread function that made the greatest contribution for that location. Figure 2b illustrated additive intersections; Figure 7 shows the same stimulus with maximum intersections. The examples in Figures 6a and 6b all have additive intersections.

Distribution Assumption

Another processing assumption concerns the shape of the spread function around each dot. The algorithm could employ distributions other than the normal for this purpose. This variation will be referred to as the *distribution* parameter. Two distributions were used, the *normal* and the *Laplace*, an exponential distribution.[2] The Laplace distribution may be more appropriate than the Gaussian for describing strength gradients (see Shepard, 1987, for an argument that exponential functions best characterize generalization gradients in psychological space). The algorithms in the first and third columns of Figures 6a and 6b use the normal distribution, and those in the second and fourth, the Laplace.

Standard Deviation Assumptions

The CODE algorithm assumes that the spread function associated with each dot in a stimulus array has its own

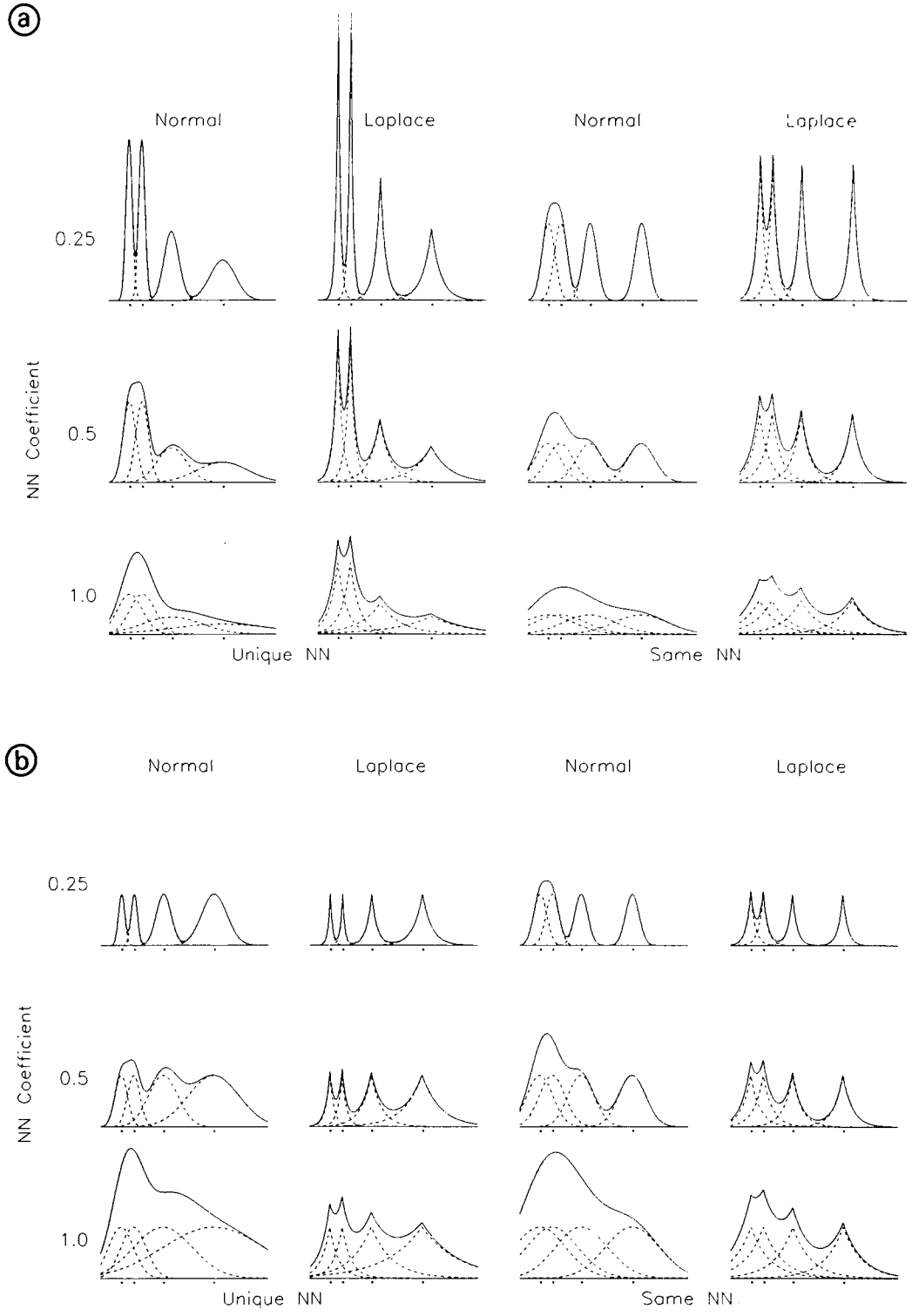

Figure 6. (a) Illustration of the 12 algorithms that had standard peaks and additive intersections. Each combination of the distribution, unique/same nearest neighbor, and nearest neighbor coefficient parameter is shown. (b) Same as Figure 6a, but with rescaled peaks.

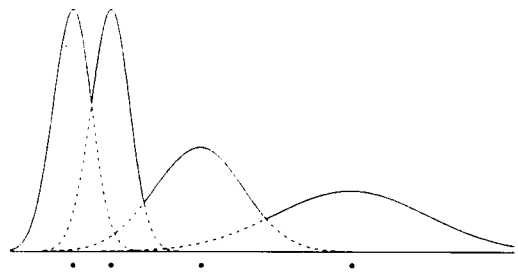

Figure 7. A strength gradient identical to the one seen in Figure 2b, but with maximum, rather than additive, intersections.

standard deviation, which is equal to one half the distance from the dot to its nearest neighboring dot. The algorithm could instead assume that the spread function for each dot is derived from the average nearest neighbor distance across all dots in the stimulus, rather than from the unique nearest neighbor distance of each dot. This variation will be referred to as the *unique/same nearest neighbor* parameter. In algorithms in which the unique/same nearest neighbor parameter was set to *unique*, the nearest neighbor contribution to each spread function was determined by the distance between the dot and its nearest neighboring dot. When the unique/same parameter was set to *same*, the nearest neighbor distance for every dot was the mean of all pairwise nearest neighbor distances. The two leftmost columns of Figures 6a and 6b show the unique/same parameter set to unique, while the rightmost two columns show it set to same.

When the nearest neighbor parameter is set to unique, the distance from each dot to its nearest neighbor contributes actively to the determination of the shapes of the spread functions, giving different spread functions different standard deviations. When the nearest neighbor parameter is set to same, all spread functions have the same standard deviation, and the nearest neighbor distance information affects all spread functions equally.

Another standard deviation assumption concerns the coefficient that, when multiplied by the nearest neighbor value, determines the standard deviation for each spread function. The algorithm could set the spread functions to multiples other than one half of the nearest neighbor distance (independently of whether the nearest neighbor value is unique for each dot or is an average of all interdot distances). This variation will be referred to as the *nearest neighbor coefficient*. The nearest neighbor coefficient specified a weighting of the nearest neighbor distance to produce the standard deviation for each spread function. The different weightings used were .25, .5, and 1. The effect of these different weightings can be seen in Figures 6a and 6b. Differences in this parameter affect all spread functions equally. As the nearest neighbor coefficient increases, spread functions get larger, and the extent to which a given dot can influence the grouping of its relatively distant neighbors increases.

Purpose of Testing Variants of CODE

The strength gradient surface of the original CODE algorithm can be described in terms of the five parameters. It has rescaled peaks, additive intersections, normally distributed spread functions, unique nearest neighbor distances, and nearest neighbor coefficients of .5. The CODE algorithm (in its expanded form, in which the threshold is allowed to vary) can therefore be seen as 1 algorithm in the set of 48 that were investigated in the present experiment.

The variants of the CODE algorithm allow some of its assumptions to be tested in order to determine whether an algorithm of the complexity of CODE is required to produce adequate predictions of subjects' grouping judgments. The different algorithms allow for different and in some cases simpler assumptions about the mechanisms underlying perceptual grouping by proximity.

For example, an algorithm that assumes that spread functions are superimposed and not added together (the intersection parameter set to max) and that each spread function in a stimulus array is identical (the unique/same parameter set to same) predicts that grouping is solely a function of the distance between dots. (Algorithms with this particular configuration of parameters will produce identical grouping judgments, regardless of the settings of the peaks, distribution, or nearest neighbor coefficient parameters.) This approach to grouping can be visualized as drawing a circle around each dot in the stimulus array (with the height of the threshold determining the size of the circles) and assigning dots whose circles overlap to a common group. This simple distance model is perhaps

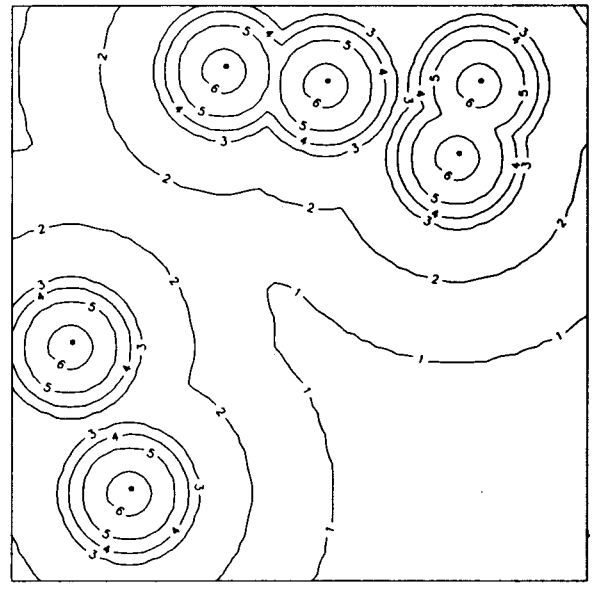

Figure 8. An overhead view of six thresholds (labeled 1–6) applied to a strength gradient that is based on the same dot pattern as that shown in Figure 5b, but that has different parameters.

one of the simplest accounts of grouping by proximity that can be imagined. Figure 8 shows the sets of groups that are found when the simple distance model is applied to the stimulus first introduced in Figure 3a. It should be noted that for this particular stimulus pattern, the sets of groups that are identified by the absolute distance model are identical to the sets of groups identified by the CODE algorithm (see Figure 5b). However, this equivalence of result does not always hold between the two algorithms when other stimuli are used, as will be discussed later.

A goal of the present study was to compare the relatively complex assumptions of the CODE algorithm with simpler assumptions, and with equally complex parallel assumptions. Through the comparison of different variants of the algorithm, from the absolute distance algorithm to more complex forms, the relative virtues of several assumptions of the CODE algorithm can be assessed.

METHOD

Subjects

The subjects were 44 undergraduate students at the University of Illinois who received course credit for their participation.

Stimuli

The stimuli consisted of 64 random dot patterns which were generated in the following manner. Dots were placed randomly within an imaginary 80×80 square grid, under the constraint that no dot be closer than four grid units to another. Eight dot arrays were generated at each of eight numerosity levels, for a total of 64 patterns. The numerosity levels were the even numbers from 6 to 20, inclusive.

The patterns were printed on 8.5×11 in. paper, one to a page, in a pseudorandom order, such that each sequence of eight patterns included one pattern from each of the eight numerosity levels.

Procedure

The subjects were tested in groups. Each subject was given a packet containing a page of instructions and the 64 dot patterns. The subjects were instructed to draw a circle or other closed form around any groups they saw on the page. They were allowed to select as many or as few groups as they wished, but dots were not to be assigned to more than one group. These rules for selecting groups will be referred to as the *selection constraint*, because they served to limit the responses that the subjects made. Written instructions were provided, which were paraphrased by the experimenter at the beginning of the session. The written instructions read as follows:

Instructions for Perceptual Judgment Experiment

The purpose of this study is to investigate how the human perceptual system organizes visual patterns into groups.

You will be asked to make grouping judgments about a series of dot patterns. Please draw a circle or other closed form around any dots you see as a group. Your grouping judgment for a single pattern may include as many or as few groups as you see fit. If you see any "stray" dots that don't seem to belong to any group, you can just ignore them.

This experiment is not a logic puzzle or an exercise in reasoning; we are merely interested in getting your perceptual intuitions. For that reason, it is not necessary to take a long time or agonize over a single pattern. The best judgments will be based upon your immediate impressions of the organization of the patterns.

You may take short breaks between patterns if you wish. When you are finished, please close your booklet and wait for further instructions from the experimenter.

The subjects used a pencil or a pen to circle the groups they saw (if any) within each pattern. They were allowed to go at their own paces, and they took between 10 and 15 min to complete their grouping judgments for the 64 patterns.

Data Analysis

In the analysis of the subject data, a group was defined as consisting of two or more dots. Any single dots that were circled by the subjects were ignored. Each subject's response on a single stimulus pattern was defined as a set of groups. A single set of groups contains zero or more groups, with the maximum number of groups being equal to one half the number of dots in the pattern (as when each dot is part of a group, and each group contains two dots).

The number of unique sets of groups that can be identified for a dot pattern of a given size, under the rules of selection that were given to the subjects, is shown in Table 1. These values, the number of theoretically possible sets of groups at each numerosity level, are determined by the selection rules given to subjects, the selection constraint, and they should not be confused with the number of sets of groups that can be generated by a specific algorithm, the *algorithm constraint*.[3] As indicated in Table 1, the size of selection-constrained sets of groups increases sharply with numerosity level.

The grouping judgments for the 44 subjects were sorted by stimulus patterns. For each stimulus pattern, subjects' responses that were identical (i.e., the sets of groups chosen were identical) were grouped together and counted. Subject data for each of the 64 stimulus patterns consisted of a list of the sets of groups that the subjects collectively generated for the pattern, and for each set of groups, the number of subjects who grouped the stimulus pattern in that particular way. The subject data were entered into a computer for comparison with the grouping predictions made by the different variations of the CODE algorithm.

Predicting Groups

The subject data were compared with the predictions made by 48 variations of the CODE algorithm. These variations resulted from the combination of five factors, in a 2 (peaks) $\times$ 2 (intersections) $\times$ 2 (distribution) $\times$ 2 (unique/same nearest neighbor) $\times$ 3 (nearest neighbor coefficient) design. As previously discussed, the CODE algorithm and its variants are capable of grouping each stimulus a number of different ways by varying the threshold. Each set of groups (for a given pattern) generated by an algorithm was taken as a single prediction, so that, in most cases, an algorithm submitted as a prediction more than one set of groups for each stimulus pattern. As a result, tests of the algorithms' performance, both against each other and against an absolute standard, involved comparing (for each stimulus pattern) several of sets of groups gener-

Table 1
Number of Unique Sets of Groups That Can Be Identified Under the Selection Constraints at Each Numerosity Level

Numerosity Level	Sets of Groups
6	203
8	4,140
10	115,975
12	4,213,597
14	190,899,322
16	10,480,142,147
18	682,076,806,159
20	51,724,158,235,372

ated by the algorithm with several sets of groups generated by the subjects.

The sets of groups found by each algorithm for each stimulus pattern were generated in the following manner. For each algorithm, the strength gradient surface for each of the 64 stimulus patterns was calculated, according to the appropriate parameters, in the following manner.

First, the standard deviation for each dot's spread function was found. The distance between each dot and its nearest neighbor was calculated, and if unique/same nearest neighbors was set to *same*, the mean of the nearest neighbor distances was used as the nearest neighbor distance for each dot. This value was then multiplied by the nearest neighbor coefficient (.25, .5, or 1) to produce the standard deviation value for each dot. Next, for each x, y position in the grid, the contribution of the spread function associated with each dot (a function using the distribution specified by the distribution parameter, standard deviation derived by the method previously described, and centered on the dot) was determined. If the peaks parameter was set to *rescaled*, the value of the spread function at each x, y position in the grid was multiplied by the reciprocal of the spread function's greatest value (i.e., the value of the function at its center). Finally, the contributions of each dot to each point in the x, y grid were combined, either additively or by taking the single greatest value, as specified by the intersection parameter.

Once the strength gradient surface was created, the threshold was varied in an attempt to identify as many sets of groups as possible. (The number of sets of groups an algorithm can generate is less than or equal to the number of dots present in the pattern; see note 2.) In order to find all possible groups quickly, the threshold was varied in an iterative fashion, in a search for still-unidentified groups, rather than simply incremented in a stepwise fashion from the bottom or top of the strength gradient surface. The nature of this iterative search process is described in detail in Appendix B.

RESULTS AND DISCUSSION

The subjects agreed with each other to varying degrees on how the dot patterns should be grouped. One indicator of the extent of agreement is the average number of subjects who selected each set of groups that the subjects produced for a given stimulus, as shown in Table 2. For example, for the stimulus pattern labeled A6, each set of groups was selected by an average of 6.3 subjects. This value is greater than the mean for Numerosity Level 6, which was 4.3, indicating better than average intersubject agreement on how Pattern A6 should be grouped. There was considerable variability in the extent of intersubject agreement within numerosity levels, and the extent of agreement declined as numerosity level increased.

Table 2
Average Number of Subjects Selecting Each Set of Groups, for Each of the 64 Stimulus Patterns

Stimulus Set	Numerosity Level								
	6	8	10	12	14	16	18	20	*M*
A	6.3	2.4	2.3	1.2	1.9	1.5	1.3	1.4	1.8
B	4.0	4.4	2.0	2.0	2.1	1.9	1.6	1.2	2.0
C	3.7	2.9	3.1	2.0	2.3	1.4	1.2	1.2	1.9
D	3.4	2.3	2.0	1.6	1.5	1.6	1.2	1.1	1.6
E	4.0	2.8	1.6	1.3	1.4	1.6	1.1	1.2	1.6
F	5.5	2.4	2.8	1.6	1.3	1.7	1.4	1.4	1.8
G	4.9	2.4	2.0	1.3	1.4	1.4	1.9	1.3	1.7
H	8.8	2.2	3.7	2.3	1.7	1.3	1.4	1.3	1.9
M	4.6	2.6	2.3	1.6	1.6	1.5	1.3	1.3	1.8

Table 3
Number of Sets of Groups Generated by Each Algorithm, With Frequencies as Percentages of the Maximum Number Possible

Intersection	Nearest Neighbor Coefficient	Unique Nearest Neighbor				Same Nearest Neighbor			
		Normal		Laplace		Normal		Laplace	
		No.	%	No.	%	No.	%	No.	%
Standard Peaks									
Additive	.25	832	100	832	100	832	100	832	100
	.5	832	100	832	100	832	100	832	100
	1	832	100	832	100	832	100	832	100
Maximum	.25	811	97	809	97	746	90	746	90
	.5	815	98	816	98	746	90	746	90
	1	820	99	816	98	746	90	746	90
Rescaled Peaks									
Additive	.25	831	100	832	100	832	100	832	100
	.5	832	100	832	100	832	100	832	100
	1	832	100	832	100	832	100	832	100
Maximum	.25	808	97	808	97	746	90	746	90
	.5	804	97	807	97	746	90	746	90
	1	804	97	815	97	746	90	746	90

Table 3 shows the number of sets of groups (aggregated over the 64 stimulus patterns) generated by each of the 48 algorithms that were tested. For 23 of the 24 versions of the algorithm that had additive intersections, the algorithm-constrained limit on the number of sets of groups that could be generated was reached for all 64 stimulus patterns. Since the maximum number of sets of groups that could be found was equal to the number of dots in the pattern (see note 2), for these 23 additive intersection algorithms the number of sets of groups found was the sum of the numerosity levels times eight stimulus sets, or 832 (the remaining algorithm generated 831 sets of groups). The 24 algorithms that had maximum intersections identified fewer sets of groups than the theoretical maximum.

There are two possible explanations for the less than perfect ability of some of the algorithms to identify all possible sets of groups. It may be that two or more dots or groups had conglomerated with a single move in threshold because two local minima in the strength gradient surface (the points at which two regions conjoin or break apart, resulting in a change in the set of groups) are located at exactly the same height. In this case, a single change in the threshold would cause two separate conglomeration events to occur simultaneously, thereby reducing the number of sets of groups that could be identified. Alternatively, it may be that because of limitations of the computer implementation, very small differences in the heights of two or more saddle points (the points in the strength gradient surface at which two groups join or break apart with a minute change in threshold) went undetected, again resulting in a reduction in the number of sets of groups that could be identified.

Table 4
Number of Stimuli, for Each Algorithm Tested, That Were Significant at the $p < .0001$ Level by the Binomial Test, With Frequencies as Percentages of Total

		Unique Nearest Neighbor				Same Nearest Neighbor			
	Nearest Neighbor	Normal		Laplace		Normal		Laplace	
Intersection	Coefficient	No.	%	No.	%	No.	%	No.	%
		Standard Peaks							
Additive	.25	63	98.4	63	98.4	63	98.4	63	98.4
	.5	63	98.4	63	98.4	63	98.4	63	98.4
	1	56	87.5	63	98.4	42	65.6	63	98.4
Maximum	.25	63	98.4	63	98.4	63	98.4	63	98.4
	.5	63	98.4	63	98.4	63	98.4	63	98.4
	1	63	98.4	64	100	63	98.4	63	98.4
		Rescaled Peaks							
Additive	.25	58	90.6	58	90.6	63	98.4	63	98.4
	.5	60	93.8	61	95.3	63	98.4	63	98.4
	1	40	62.5	57	89.1	42	65.6	63	98.4
Maximum	.25	60	93.8	60	93.8	63	98.4	63	98.4
	.5	60	93.8	60	93.8	63	98.4	63	98.4
	1	60	93.8	60	93.8	63	98.4	63	98.4

Evaluating Algorithm Performance Versus an Absolute Standard

The first question in examining the algorithms is whether the algorithms can predict subjects' judgments with any degree of accuracy. A binomial test was performed in which the frequency with which an algorithm matched subjects' judgments was compared with the probability that sets of groups selected at random (from the set of theoretically possible sets of groups; see Table 1) would match subjects' judgments.

A separate binomial test was conducted for each of the 64 stimuli, under each of the 48 algorithms, for a total of 3,072 tests. For each test, the dependent variable was the number of subjects whose judgments for the pattern were matched by one of the sets of groups generated by the algorithm for that pattern. For example, consider the case in which one of the algorithms matches 41 of the 44 subjects' grouping judgments for a certain six-dot stimulus. Imagine that this particular algorithm has generated six sets of groups for this particular stimulus. The binomial test, then, consists of comparing the actual number (41) of subject judgments that were matched, with the number of matches expected if six sets of groups were chosen at random from the set of selection constrained sets of groups (which for six-dot stimuli is 203, as shown in Table 1).

Table 4 shows, for each of the 48 algorithms, the number of stimuli (out of 64) for which the performance of the algorithm was better than chance ($p < .0001$). The 48 algorithms collectively reached the binomial criterion for success on an average of 61 of the 64 stimulus patterns. The performance of each of the algorithms is considerably better than would be expected if the sets of groups were chosen at random.

Evaluating the Algorithm Parameters

Table 5 shows the number of subjects' judgments that were matched by each algorithm, both as raw frequencies of the total and as percentages of all subject judgments. Which parameters made a difference in the ability of the algorithms to predict the subjects' judgments? To address this question, a five-way analysis of variance (ANOVA) was performed, with the five parameters that define the different algorithms as factors. The 64 stimuli were treated as subjects. For each stimulus, the dependent measure was the number of subjects whose grouping judgment for the stimulus matched one of the judgments made by the algorithm (calculated separately for each of the 48 algorithms).

A main effect of the peaks parameter was seen [$F(1,3024) = 30.17$, $MS_e = 3,154$, $p < .001$], with the algorithms having standard peaks matching an average of

Table 5
Number of Subject Judgments Matched by Each Algorithm, With Percent of Subject Judgments Matched

		Unique Nearest Neighbor				Same Nearest Neighbor			
	Nearest Neighbor	Normal		Laplace		Normal		Laplace	
Intersection	Coefficient	No.	%	No.	%	No.	%	No.	%
		Standard Peaks							
Additive	.25	924†	38.2	988†	35.1	1,015†	36.0	1,013†	36.0
	.5	986†	35.0	1,029†	36.5	985†	35.0	978†	34.7
	1	549	19.5	825	29.3	281‡	10.0	716	25.4
Maximum	.25	937†	33.3	959†	34.1	998†	35.4	998†	35.4
	.5	1,052†	37.4	1,056†	37.5	998†	35.4	998†	35.4
	1	863	30.6	1,027†	36.5	998†	35.4	998†	35.4
		Rescaled Peaks							
Additive	.25	672	23.9	654	23.2	1,015†	36.0	1,013†	36.0
	.5	701*	24.9	731	26.0	985†	35.0	978†	34.7
	1	348‡	12.4	608	21.6	281‡	10.0	716	25.4
Maximum	.25	728	25.9	728	25.9	998†	35.4	998†	35.4
	.5	728	25.9	728	25.9	998†	35.4	998†	35.4
	1	728	25.9	728	25.9	998†	35.4	998†	35.4

*CODE algorithm. †Matched significantly more judgments than did the CODE algorithm ($p < .001$). ‡Matched significantly fewer judgments than did the CODE algorithm ($p < .001$).

32.8% of the subjects' judgments, and those having rescaled peaks matching 28.2%. As shown in Table 5, each algorithm with standard peaks performed as well as or better than its counterpart with rescaled peaks.

A main effect of the intersection parameter was seen [$F(1,3024) = 32.82$, $MS_e = 3{,}431$, $p < .001$], with the algorithms having additive intersection algorithms matching an average of 28.1% of subjects' judgments, and those having maximum intersections matching 32.9%. The effect of having additive intersections was especially detrimental when the nearest neighbor coefficient was set to 1, a combination that seemed to overemphasize global factors in grouping.

A main effect for distribution of the spread function was seen [$F(1,3024) = 8.97$, $MS_e = 937$, $p < .005$], with algorithms using the normal distribution matching 29.2% of the judgments, and those using the Laplace distribution matching 31.8%. Algorithms with Laplace distributions were less susceptible than their counterparts with normal distributions when other parameters came together to overemphasize interactivity (e.g., additive intersections or a nearest neighbor coefficient of 1), as is shown in Table 5.

A main effect was seen for the unique/same nearest neighbor parameter [$F(1,3024) = 22.28$, $MS_e = 2{,}329$, $p < .001$], with algorithms having unique nearest neighbors matching 28.5% of subjects' judgments and those with the same nearest neighbors matching 32.5%. Models with the same nearest neighbors ignore nearest-neighbor relations (except to the extent that the mean nearest-neighbor value may change from pattern to pattern) and involve more local computations than do their counterparts that use unique nearest neighbors.

Table 6
Two-Way Interaction ($p < .01$) Between Peaks and Unique/Same Parameters: Percent of Subject Judgments Matched

	Unique/Same	
Peaks	Unique	Same
Standard	33.1	32.5
Rescaled	23.9	32.5

Table 7
Two-Way Interaction ($p < .05$) Between Intersection and Distribution Parameters: Percent of Subject Judgments Matched

	Distribution	
Intersection	Normal	Laplace
Additive	25.9	30.3
Maximum	32.6	33.2

Table 8
Two-Way Interaction ($p < .01$) Between Distribution and Standard Deviation Coefficient Parameters: Percent of Subject Judgments Matched

	SD Coefficient		
Distribution	.25	.5	1.0
Normal	32.3	33.0	22.4
Laplace	32.6	33.3	29.4

Table 9
Two-Way Interaction ($p < .01$) Between Intersection and Standard Deviation Coefficient Parameters: Percent of Subject Judgments Matched

	SD Coefficient		
Intersection	.25	.5	1.0
Additive	32.4	32.7	19.2
Maximum	32.6	33.5	32.6

Table 10
Two-Way Interaction ($p < .05$) Between Unique/Same and Standard Deviation Coefficient Parameters: Percent of Subject Judgments Matched

Nearest Neighbor Parameter	*SD* Coefficient		
	.25	.5	1.0
Unique	29.3	31.1	25.2
Same	35.7	35.1	26.6

Table 11
Three-Way Interaction ($p < .05$) Between Intersection Unique/Same, and Standard Deviation Coefficient Parameters: Percent of Subject Judgments Matched

Nearest Neighbor Parameter	*SD* Coefficient		
	.25	.5	1.0
Additive Intersections			
Unique	28.7	30.6	20.7
Same	36.0	34.4	17.7
Maximum Intersections			
Unique	29.8	31.6	29.7
Same	35.4	35.4	35.4

A main effect was seen for the nearest neighbor coefficient [$F(2,3024) = 30.53$, $MS_e = 3{,}192$, $p < .001$], with algorithms having a coefficient of 0.25 matching 32.5% of subjects' judgments, those with a coefficient of 0.5 matching 33.1%, and those with a coefficient of 1 matching 25.9%. Whenever there was a difference among nearest neighbor coefficients, algorithms with a coefficient of 1 performed worse than those with a coefficient of .5. These results clearly show the detrimental effect of overemphasizing interactivity among points in a stimulus, at the expense of keeping local features distinct.

In addition to the main effects, there were five two-way interactions, and two three-way interactions. These interactions are shown in Tables 6-12. The interaction of the peaks and the unique/same parameters, shown in Table 6, shows that rescaling the spread functions matters only when unique nearest neighbor distances are used. This result is to be expected, since rescaling the peaks should have no effect when each spread function already has the same height, as is the case when the nearest neighbor distances (and therefore the spread functions) are identical for all dots.

The main effects and the remaining interactions (other than the peaks by unique/same interaction previously mentioned) seem to result from related phenomena. The al-

Table 12
Three-Way Interaction ($p < .05$) Between Intersection, Distribution, and Standard Deviation Coefficient Parameters: Percent of Subject Judgments Matched

Distribution	*SD* Coefficient .25	.5	1.0
Additive Intersections			
Normal	32.2	32.5	13.0
Laplace	32.6	33.0	25.4
Maximum Intersections			
Normal	32.5	33.5	31.8
Laplace	32.7	33.6	33.3

gorithms that were more successful tended to be those that computed their groups from relatively more local information about the stimulus pattern. Across the 48 algorithms tested, the maximum intersection algorithms were more successful than the additive intersection algorithms. The maximum intersection algorithms, in which the contribution of each spread function to the strength gradient surface is limited by the contributions of other spread functions, can be said to be more local than the additive algorithms, whose spread functions contribute some quantity to the entire strength gradient surface.

The algorithms with Laplace distributions were more effective than those with normal distributions. In a Laplace algorithm, the contribution to the strength gradient surface is greater in the immediately surrounding regions than in normal algorithms.

The nearest neighbor coefficient parameter follows a similar pattern. Although the algorithms with a coefficient of 0.25, which were the most local in nature, did not perform as well as those with a coefficient of 0.5, the algorithms with a coefficient of 1, the least local, performed the worst of all.

Evaluating Performance Among the Algorithms

In addition to the test of the algorithm parameters, performance among the individual algorithms was assessed. In this test, the CODE algorithm was compared with each of the other 47 algorithms. For each comparison, a Wilcoxon signed ranks test was performed on the number of subject judgments that were matched by the algorithms on each stimulus. With 48 comparisons, the alpha level was set to $p < .001$. As indicated in Table 5, 29 algorithms performed significantly better than the CODE algorithm, 3 algorithms performed significantly worse, and 15 algorithms did not perform significantly better or worse. It is clear that the performance of CODE can be dramatically improved by modifying some of its assumptions.

Performance Distinctions Among Algorithms: An Example

Figure 9 gives an example of how different algorithm parameters can group the same stimulus pattern in different ways. Figure 9a shows one of the stimuli used in the experiment. Figures 9b and 9c show the sets of groups generated for this stimulus by two different algorithms. Both algorithms had standard peaks, additive intersections, Laplace distributions, and unique nearest neighbors, with the algorithm in Figure 9b having a nearest neighbor coefficient of .5 and the one in Figure 9c having a nearest neighbor coefficient of 1. The sets of groups specified by these two algorithms are identical, except for those specified at Threshold 2 by each algorithm. The algorithm in 9b specifies the set of groups {ABCD},{EF}, and the one in 9c specifies {ABCEF},{D}.

Table 13 lists each set of groups subjects generated for this pattern. The algorithm with a nearest neighbor coefficient of 1 fails to predict the set of groups {ABCD},{EF}. This is because the spread functions for that algorithm exert their influence over a very large area of the stimulus field. The aggregate influence of the spread functions increases the strength of grouping around the center of the dot figure, at the expense of the separateness of Group {ABCD}. This is one example of the detrimental effect of overemphasizing global factors in proximal grouping.

Performance of Single Threshold Version of CODE

The performance of the original single threshold version of CODE (with the threshold set to 1) was assessed. The single-threshold version of CODE generated one set of groups for each of the 64 stimulus patterns, in comparison with the multiple-threshold (but otherwise identical) version, which generated a total of 832 sets of groups for the 64 patterns (see Table 3). The single-threshold version of CODE matched 189 judgments, which is 6.7% of the 2,816 judgments made by all subjects. (As was shown in Table 5, the multiple-threshold version matched 701 judgments, which is 24.9% of total subject judgments.)

For the purposes of comparison, the same assessment was made of an alternative single-threshold version of CODE. This alternate version used a single variable threshold (determined separately for each stimulus pattern) which was defined as the one that matched the greatest number of subjects' judgments, rather than the fixed threshold of 1. This "best threshold" version of CODE matched 367 judgments, or 13.0% of the total. If the CODE algorithm is to be used to predict a single grouping configuration for each pattern, the process by which the threshold is selected will require further investigation.

It is clear that the multiple-threshold version of CODE outperforms the single-threshold version. To what factors can this improvement be attributed? One possibility is that the threshold that the subjects used varied from stimulus to stimulus and was determined by the stimulus alone or by an interaction between subject and stimulus. If this were the case, the multiple-threshold version of CODE would be most appropriate. A second possibility is that differences in thresholds were produced solely by individual differences among the subjects. If this were the case, a model intermediate between the single-threshold and the multiple-threshold model would be appropriate—one in which each of the subjects has a unique threshold

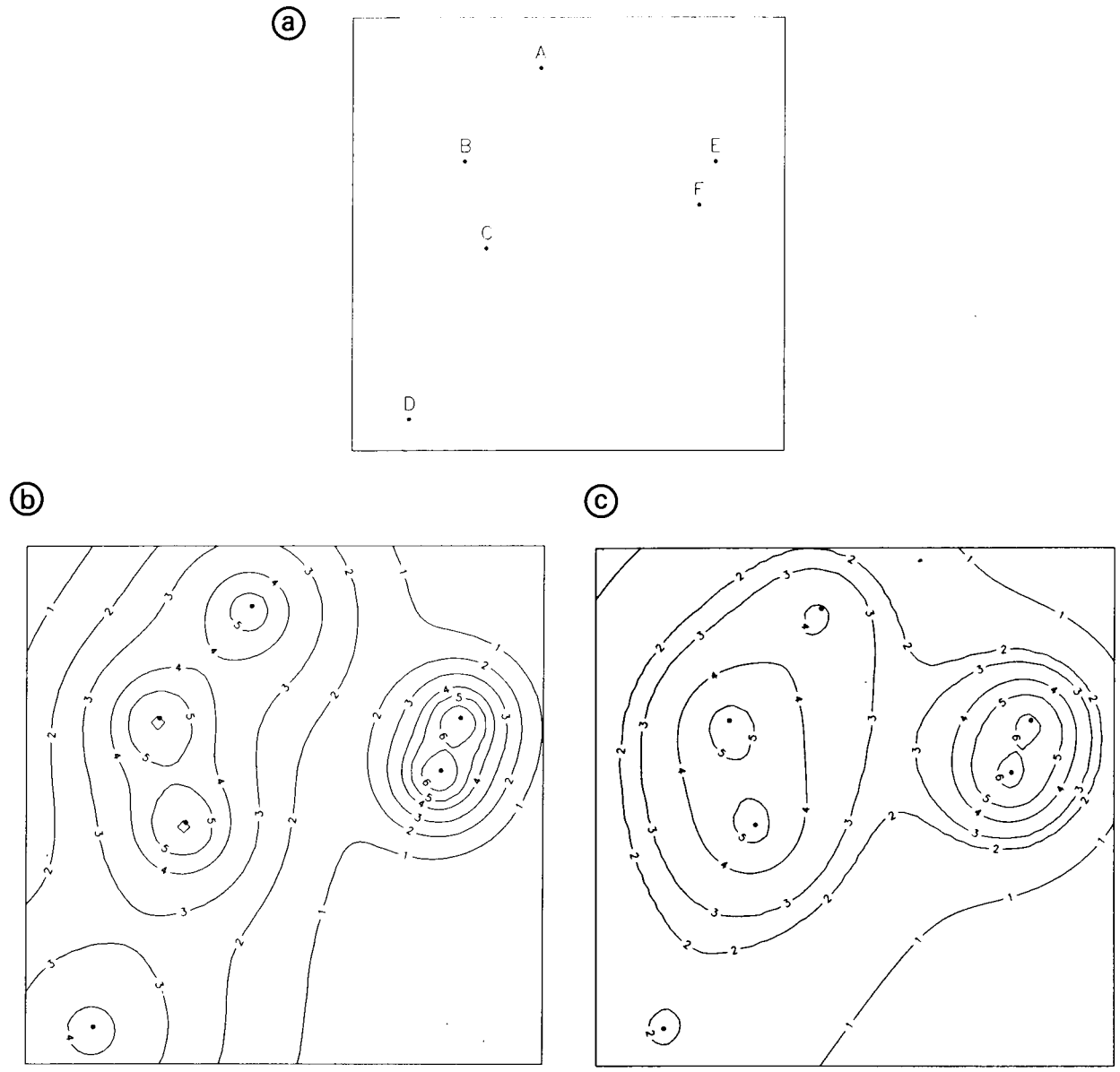

Figure 9. (a) A two-dimensional stimulus pattern. (b and c) An overhead view of six thresholds (labeled 1–6) applied to two strength gradients based on the dot pattern seen in Figure 9a. The thresholds in Figures 9b and 9c are generated by algorithms with additive intersections, Laplace distributions, and unique nearest neighbors, with that in Figure 9b having a nearest neighbor coefficient of .5 and that in Figure 9c having a nearest neighbor coefficient of 1.

that is applied to all stimulus patterns. According to this *individual differences* view, the threshold would be expected to vary only between subjects, and not within subjects.

To investigate the individual differences explanation, we performed the following analysis. Our goal was to determine whether subjects differed in their average threshold and to assess the extent of differences between subjects relative to differences within subjects. We entered the threshold that each subject used on each stimulus for which there was a match between their judgment and the algorithm (using the most successful algorithm so as to maximize the number of available data points) into a one-way ANOVA, with subjects as a factor. The effect of subjects was significant [$F(43,1{,}055) = 7.69$, $MS_e = 4{,}443{,}314$, $p < .001$], which indicates the presence of individual differences among subjects in the thresholds they used. However, the main effect of subjects accounted for only 32.7% of the variance. The remaining variance was due to differences in thresholds within subjects.

The individual differences model requires that there be differences between subjects in the thresholds that they used, but not differences within subjects. Because of the large variation of thresholds within subjects, the individual differences model would be unable to approach the success of the multiple-threshold model.

Table 13
Complete Set of Grouping Judgments Made by Subjects Versus Predictions Made by Two Algorithms for the Stimulus Pattern Seen in Figure 9A

Number of Subjects (n = 44)	Set of Groups Selected	Nearest Neighbor Coefficient .5	Nearest Neighbor Coefficient 1
14	{ABC},{EF},{D}	3	3
13	{BC},{EF},{A},{D}	4	4
10	{EF},{A},{B},{C},{D}	5	5
4	{ABCD},{EF}	2	
3	{AB},{CD},{EF}		

Note—Both algorithms have standard peaks, additive intersections, Laplace distributions, and unique nearest neighbors; the nearest neighbor coefficient is .5 for one algorithm, 1 for the other. Columns for algorithms (labeled Nearest Neighbor Coefficient) show threshold index if the set of groups was predicted. They are left blank if it was not. Judgments matched = 41 for coefficient of .5, 37 for coefficient of 1.

Table 14
Correlation Between Index of Goodness and Number of Subject Judgments Matched by Each Algorithm

Intersection	Nearest Neighbor Coefficient	Unique Nearest Neighbor Normal	Unique Nearest Neighbor Laplace	Same Nearest Neighbor Normal	Same Nearest Neighbor Laplace
		Standard Peaks			
Additive	.25	.85	.83	.85	.85
	.5	.84	.83	.86	.81
	1	.78	.82	.43	.83
Maximum	.25	.83	.84	.85	.85
	.5	.81	.81	.85	.85
	1	.81	.83	.85	.85
		Rescaled Peaks			
Additive	.25	.73	.74	.85	.85
	.5	.72	.69	.85	.85
	1	.74	.75	.85	.85
Maximum	.25	.80	.80	.85	.85
	.5	.80	.80	.85	.85
	1	.80	.80	.85	.85

Algorithm Performance and Goodness

A final question concerns the relation between the goodness of the individual stimuli and the ability of the algorithms to predict how subjects will group them. An index of goodness was derived, based on subject agreement. The more subjects who agreed on a particular grouping, the better the grouping (suggested by Hochberg & McAlister, 1953). The average number of subjects who selected each set of groups was calculated for each stimulus pattern. Stimuli were considered good to the extent that the average number of subjects selecting each set of groups was high. The correlation between the average number of subjects per set of groups and the number of subject judgments matched for each stimulus was computed. The resulting correlation is shown for each algorithm in Table 14. As was shown in Table 2, the extent of intersubject agreement declines sharply with numerosity level. Similarly, the performance of the algorithms declines sharply with numerosity level. Figure 10 shows performance by numerosity level, averaged across all algorithms, and for the single best algorithm. The results are what should be expected from proximal grouping algorithms faced with stimuli that have varying degrees of goodness: good stimuli are associated with more accurate predictions of grouping by the algorithm.

In order to ensure that the ability of the present experiment to discriminate among algorithms was not dependent upon the degree of goodness of the patterns being grouped, the following analysis was performed. The 64 stimulus patterns were assigned to either a high or a low goodness category (with 32 patterns being assigned to each category), according to the criterion of intersubject agreement. For both the high and the low goodness stimuli, the number of subject judgments matched was averaged over the 32 stimulus items and placed into the design of the five-way ANOVA that was described in the section entitled Evaluating the Algorithm Parameters. This ANOVA had the five parameters (peaks, intersection, distribution, unique/same nearest neighbors, and nearest neighbor coefficient) that defined the algorithms as factors. The correlation between the cells for the high goodness and the low goodness patterns was .93.

Two separate ANOVAs were performed; one for the high-goodness stimuli and one for the low-goodness stimuli. These ANOVAs were identical in design to the five-way ANOVA that was used to evaluate the algorithm parameters (except that they had 32, rather than 64, stimuli). The results of the two ANOVAs were extremely similar to each other and to the 64-stimulus ANOVA. The main effects were identical in all three ANOVAs, and the interactions were identical, with the following exceptions. In the high-goodness stimulus ANOVA, the two-way interaction of intersection × nearest neighbor coefficient was not significant [$F(2, 1488) = 1.96$, $MS_e = 184$, $p < .14$], although it reached significance in both the 64 stimuli ANOVA and the low goodness stimuli ANOVA. In the low-goodness stimulus ANOVA, the two-way interaction of intersection and distribution [$F(1, 1488) = 3.68$, $MS_e = 62$, $p > .05$], and the three-way interaction of intersection, distribution, and nearest neighbor coefficient

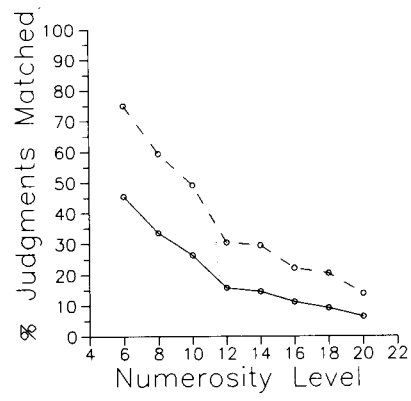

Figure 10. Percent of subject judgments matched at each numerosity level. Solid line shows average over the 48 algorithms; dotted line shows single most successful algorithm.

[$F(2, 1488) = 1.61$, $MS_e = 27$, $p > .20$], were not significant, although they were significant in both the 64-stimulus ANOVA and the high-goodness stimulus ANOVA. In the high-goodness stimulus ANOVA, the effects that agreed with those found in the other two ANOVAs accounted for 98.6% of the variance due to treatments, and in the low-goodness stimulus ANOVA, the effects that agreed with the other two ANOVAs accounted for 98.7% of the variance due to treatments. These results indicate that performance distinctions among algorithms were stable across changes in pattern goodness.

GENERAL DISCUSSION

The proximal grouping algorithms examined in this study were successful in matching subjects' judgments. This was particularly true when there was agreement among subjects on how a particular pattern should be grouped. The comparisons among the five algorithm parameters and the 48 algorithms demonstrated the unsuitability of defining proximal grouping in a way that overemphasizes interactivity between dots in relatively distant regions of the configuration. The performance of the original CODE algorithm (as measured by the pairwise comparison with its alternatives) was significantly worse than many of its alternatives because of an overemphasis on interactivity.

The most successful algorithms, however, were not the most local. Several of the algorithms that were most successful used unique nearest neighbor distances (which emphasize differences in individual nearest neighbor distances), and nearest neighbor coefficients of .5 (which is intermediate between the most local value, .25, and the most global, 1).

The test of the algorithms demonstrates that the best models of grouping by proximity employ a moderate level of interactivity among the stimulus elements during processing. Since to group elements is to establish a relationship among them, models of grouping must employ some degree of interactivity. The present study indicates that the degree of interactivity should be relatively minimal. Algorithms that use only limited interactivity should be easier to compute, since they can be performed simultaneously at a number of locations.

The various algorithm parameters (except, perhaps, the distribution parameter) can be seen as different ways of defining interactivity. Since no configuration of parameters specified a single algorithm that was vastly superior to all others, the best algorithms can be viewed as existing within a region of a space defined by the various algorithm parameters. In general, the success of an algorithm was a result of the configuration of a number of parameters, and not just the value of a single parameter. For example, consider the single most successful algorithm, which matched 1,056 subjects, and had rescaled peaks, maximum intersections, Laplace distributions, unique nearest neighbors, and nearest neighbor coefficients of .5, as shown in Table 5. With one exception, the algorithms that differ from this best algorithm by only one parameter also performed quite well. The exception is the algorithm that shared four of the parameters and had rescaled peaks, which performed rather poorly. It is interesting that the rescaling of peaks had perhaps the most dramatic effect of the strength gradient surface, suggesting that it would be computationally intensive to implement.

While the CODE algorithm was moderately successful, the present study shows it can be improved upon without making more complex processing assumptions. For relatively ambiguous stimuli, models of perceptual grouping need to account for more than one potential "good" configuration. The original formulation of the CODE algorithm, with its single prediction of grouping, is less appropriate for ambiguous stimuli than an algorithm that allows for several possible organizations.

The ability of the algorithms (in absolute terms) to predict subjects' judgments was highly correlated with the goodness of the individual stimuli, as measured by intersubject agreement. (In contrast, the relative ability of individual algorithms, in comparison with other algorithms, to predict subjects' judgments was largely unaffected by goodness.) In addition to the issue of goodness, there is another limit to the ability of a proximity-based definition of grouping to predict the groups subjects see in random dot patterns. In some patterns, grouping principles based on, for example, orientation, good continuation, or similarity may overpower organizations based on proximity alone. An example of this can be seen in the subjects' judgments for the dot pattern shown in Figure 9. One set of groups, {AB},{CD},{EF}, was selected by 3 subjects, but by neither of the two algorithms. If a line is drawn between the two dots in each group, it is clear that the three groups of dots have very similar orientations. This explanation of the set of groups {AB},{CD},{EF} suggests that proximity is not the only grouping principle in effect. A more complete model of perceptual grouping must employ other grouping principles, in addition to proximity, even when dot pattern stimuli are used. Such an approach will require a balancing of different grouping principles. Similarly, each grouping principle has its own limitations. Koffka (1935) suggested that the applicability of the principle of grouping by proximity is limited by the perceived goodness of the stimulus to which it is applied.

This study has shown that assumptions about the mechanisms of proximal grouping can make nontrivial differences in the way a stimulus is organized. It is not always the case that design decisions based on what "looks good" to the designer will lead to the most effective characterization of grouping for a number of subjects over a number of different stimuli. Most of the algorithms predicted judgments very successfully when subject agreement was high. As subject agreement declined, all of the algorithms performed less well, and the more successful algorithms continued to outperform the less successful ones.

One possible interpretation of the role of goodness in differentiating among the algorithms involves a distinction between bottom-up and top-down processing in perceptual grouping. While van Oeffelen and Vos (1982) pre-

sented the CODE algorithm as a strictly bottom-up account of grouping, the approach taken in the present paper can be seen as having both bottom-up and top-down components. The creation of the strength gradient surface and the generation of different sets of groups by applying different thresholds to the surface can be seen as purely bottom-up processes, while the selection of a single set of groups can be seen as a top-down process. Perhaps the job of the bottom-up processes is to generate a reasonable set of interpretations, from which top-down processes can select a single interpretation. From this perspective, for an algorithm to match an extremely "good" pattern, it need only produce among its alternatives the single popular interpretation that many subjects choose. Stimuli that are less good provide a more stringent test of the algorithm: for these patterns, the more successful algorithms will be those that produce several reasonable interpretations.

Marr's Approach to Grouping

CODE was designed to find proximity-based groups in a two-dimensional stimulus pattern. If the representation of spatial position described by CODE is accurate, one would expect to find analogous representations of spatial position in contexts other than perceptual grouping. Van Oeffelen and Vos (1983) compared CODE to the grouping function of Marr's (1982) primal sketch. Marr's primal sketch is one level of representation in his ambitious theory of vision, in which grouping plays only a small part.

Marr described grouping as functioning to create tokens in the primal sketch. Although both CODE and the primal sketch represent spatial position as a landscape of peaks of excitation centered on the elements present, there are two important differences. First, the component of the primal sketch analogous to the Gaussian spread functions of CODE uses a different function, $\nabla^2 G$, to represent points in the stimulus field.[4] This function, which is shown in Figure 11, has both excitatory and inhibitory components. In contrast, CODE uses a Gaussian function to represent excitation around a point. CODE lacks any inhibitory component. As was shown in the present experiment, relatively subtle changes in the design assumptions of the CODE algorithm can produce very different grouping predictions. Changes in shape of the spread function alone might produce significantly different grouping predictions, but this remains to be seen.

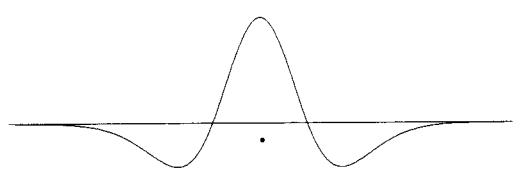

Figure 11. The shape of the $\nabla^2 G$ function used in Marr's primal sketch, approximated as the difference of two Gaussian functions in a ratio of 1:1.6.

One aspect of the use of the $\nabla^2 G$ function by the primal sketch is that the net level of excitation averaged across the stimulus field is zero. As with CODE, groups can then be identified as elements contained within contiguous regions that surpass the threshold (which is zero). The "strength of grouping" factor, which is represented in the modified CODE algorithm by varying the threshold, could be represented in the primal sketch by varying the size of the $\nabla^2 G$ filter around each point (see Watt, 1988, p. 116). When the spread of the filter is large, the strength of grouping is high, and when it is small, the strength of grouping is low. The primal sketch model assumes that changes in the strength of grouping are accomplished by changing the surface that represents the stimulus and leaving the threshold the same. In contrast, CODE indicates that changes in the strength of grouping are accomplished by leaving the surface representing the stimulus the same, and by changing the threshold. These two approaches to characterizing the strength of grouping may not produce equivalent results. A useful comparison could be made between CODE and the primal sketch, using the approach presented in this study. This would require that certain details of the primal sketch be made explicit: for example, the way in which the spread of the $\nabla^2 G$ filter changes to create different grouping configurations. Such a comparison is beyond the scope of the present study.

Grouping Effects in Cognition and Memory

The approach to representing spatial position taken by CODE is similar to that taken by Ratcliff (1981) in a model of order relations in perceptual matching. In Ratcliff's model, the spatial position of each letter in memory is represented with a normal distribution. As the delay between study and test was increased, the standard deviation of each spatial position increased, and the spatial position of each letter became less distinct. As a result, it becomes more difficult to distinguish between stimuli that differ by a transposition of adjacent elements. Ratcliff presented subjects with a string of letters, which was then masked, and a test string was presented for a *same/different* discrimination after a delay. The discrimination was more difficult when adjacent letters in the test string were transposed than it was when nonadjacent letters were transposed. The phenomenon of spatial generalization, which leads to performance decrements in this perceptual matching task, may be similar to the ability to generalize about position in order to determine what groups are present in a stimulus field.

The way in which CODE represents spatial positions (as distributions centered on each element) is similar to the way in which temporal positions are represented in Glenberg and Swanson's (1986) theory of temporal distinctiveness in memory retrieval. In this theory, memory traces include the time at which the item is presented. An advantage of the auditory versus visual modality for recently presented items is attributed to temporal information being represented more precisely for the auditory information (i.e., the spread function representing each

temporal position is smaller for auditory as opposed to visual items). This is similar to the observation that the variations of CODE that performed the most poorly had the largest spread functions and thus overgeneralized positional information.

It has been noted in many studies that pattern configuration affects performance on a number of tasks. Configuration has been shown to influence numerosity judgments (see, e.g., Bevan, Maier, & Helson, 1963; van Oeffelen & Vos, 1982). Banks and Prinzmetal (1976; Banks, Larson, & Prinzmetal, 1979) found effects of the proximal grouping of distractors in visual attention tasks. Using dot patterns in a series of experiments, Hock, Tromley, and Polmann (1988) found similarity, categorization, and memory effects that involved higher order perceptual units and could not be explained in terms of differences between patterns at the level of individual elements. The role of perceptual organization in studies such as these could be illuminated by the use of a formal grouping algorithm, such as CODE, to describe pattern configuration.

This study has shown that grouping by proximity can be modeled with relatively local computations. The approach that CODE takes to representing positional information is analogous to the ways in which spatial (or even temporal) position have been represented in other contexts. It may be that other grouping principles can also be described with relatively local computations, and that they make use of similar representations of position. A more complete model of perceptual grouping will include a number of grouping principles, with each principle fully specified, and tested against subjects' judgments. Such an approach, if successful, will tell us much about how position is represented, and about the types of computations that are required to impose organization on a stimulus field. A successful computational model of perceptual organization would also be a valuable methodological tool in any experimental setting in which pattern configuration plays a role.

A number of studies have shown that stimuli cannot be fully understood from their physical descriptions alone; their representations at the psychological level must be understood as well. To understand the role that pattern configuration plays in various contexts, one must understand the principles of perceptual organization in detail.

REFERENCES

ALLIK, J., & TUULMETS, T. (1991). Occupancy model of perceived numerosity. *Perception & Psychophysics*, **49**, 303-314.

ATTNEAVE, F. (1950). Dimensions of similarity. *American Journal of Psychology*, **63**, 516-556.

BANKS, W. P., LARSON, D. W., & PRINZMETAL, W. (1979). Asymmetry of visual interference. *Perception & Psychophysics*, **25**, 447-456.

BANKS, W. P., & PRINZMETAL, W. (1976). Configurational effects in visual information processing. *Perception & Psychophysics*, **19**, 361-367.

BEVAN, W., MAIER, R., & HELSON, H. (1963). The influence of context upon the estimation of number. *American Journal of Psychology*, **76**, 464-469.

GARNER, W. R. (1970). Good patterns have few alternatives. *American Scientist*, **58**, 34-42.

GARNER, W. R. (1974). *The processing of information and structure*. Potomac, MD: Erlbaum.

GLENBERG, A. M., & SWANSON, N. G. (1986). A temporal distinctiveness theory of recency and modality effects. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **12**, 3-15.

HOCHBERG, J. E. (1981). Levels of perceptual organization. In M. Kubovy & J. R. Pomerantz (Eds.), *Perceptual organization* (pp. 141-180). Hillsdale, NJ: Erlbaum.

HOCHBERG, J. E., & MCALISTER, E. (1953). A quantitative approach to figural "goodness." *Journal of Experimental Psychology*, **46**, 361-364.

HOCK, H. S., TROMLEY, C., & POLMANN, L. (1988). Perceptual units in the acquisition of visual categories. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **14**, 75-84.

KOFFKA, K. (1935). *Principles of Gestalt psychology*. New York: Harcourt, Brace.

MARR, D. (1982). *Vision*. San Francisco: W. H. Freeman.

PALMER, S. E. (1975). Visual perception and world knowledge: Notes on a model of sensory-cognitive interaction. In D. A. Norman & D. E. Rumelhart (Eds.), *Explorations in cognition* (pp. 279-307). San Francisco: W. H. Freeman.

POMERANTZ, J. R. (1981). Perceptual organization in information processing. In M. Kubovy & J. R. Pomerantz (Eds.), *Perceptual organization* (pp. 141-180). Hillsdale, NJ: Erlbaum.

RATCLIFF, R. (1981). A theory of order relations in perceptual matching. *Psychological Review*, **88**, 552-572.

SHEPARD, R. N. (1987). Toward a universal law of generalization for psychological science. *Science*, **237**, 1317-1323.

SMITS, J. T., VOS, P. G., & VAN OEFFELEN, M. P. (1985). The perception of a dotted line in noise: A model of good continuation and some experimental results. *Spatial Vision*, **1**, 163-177.

VAN OEFFELEN, M. P., & VOS, P. G. (1982). Configurational effects on the enumeration of dots: Counting by groups. *Memory & Cognition*, **10**, 396-404.

VAN OEFFELEN, M. P., & VOS, P. G. (1983). An algorithm for pattern description on the level of relative proximity. *Pattern Recognition*, **16**, 341-348.

VOS, P. G., & HELSPER, E. L. (1991). Modelling detection of linear structure in dot patterns. In I. Dvorak & A. V. Holden (Eds.), *Mathematical approaches to brain functioning diagnostics* (pp. 429-441). Manchester, UK: Manchester University Press.

VOS, P. G., VAN OEFFELEN, M. P., TIBOSCH, H. J, & ALLIK, J. (1988). Interactions between area and numerosity. *Psychological Research*, **50**, 148-154.

WATT, R. J. (1988). *Visual processing: Computational, psychophysical, and cognitive research*. East Sussex, UK: Erlbaum.

NOTES

1. As a demonstration of this constraint, consider the case in which none of the dots in a stimulus array are grouped together, as when the grouping threshold is set to a very high value. As this threshold value is lowered, dots conglomerate to form groups of more than one dot. Each time a single dot or group conglomerates with another single dot or group, a new set of groups is found. As each new set of groups is identified, the remaining number of single dots or groups that can potentially be conglomerated to create a new set of groups is reduced by one. Therefore, at each point in the process, the number of remaining sets of groups that can potentially be identified is constrained by the number of remaining single dots or groups. If more than one single dot or group conglomerates at a single increment of the threshold, the number of sets of groups that will be produced will be less than the number of dots the stimulus contains. This constraint applies both to the original CODE algorithm, and to all its variants that are discussed in the present paper.

2. The three-dimensional probability distribution for the normal distribution is

$$f(r) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(1/2)(r/\sigma)^2},$$

and the corresponding formula for the Laplace distribution is

$$f(r) = \frac{1}{\sigma\sqrt{2}} e^{(-1\sqrt{2}r)/\sigma}.$$

In each formula, σ is the standard deviation and r is the radial distance from the origin.

3. The selection constraint is discussed in Appendix A; the algorithm constraint is discussed in note 1. Since the algorithms followed the rules of selection in generating their sets of groups, each algorithm-constrained set of groups is a subset of the selection-constrained set of groups for the appropriate numerosity level. The set of groups generated by an algorithm is both selection constrained and algorithm constrained.

4. The formula for the $\nabla^2 G$ function is given by Marr as

$$\nabla^2 G(r) = \frac{-1}{\pi\sigma^4}\left(1 - \left(\frac{r^2}{2\sigma^2}\right)\right) e^{-r^2/2\sigma^2},$$

where r is the radial distance from the origin. This function can be closely approximated by the difference of two Gaussian functions in a ratio of 1:6, as is shown in Figure 11.

APPENDIX A

To determine whether the algorithms examined in the present study performed better than would be expected by chance, it was necessary to calculate the number of logically possible ways of grouping a stimulus pattern. As numerosity increases, the number of possible ways of grouping the stimulus pattern increases, as can be seen in Table 1. The number of potential sets of groups was calculated separately for each numerosity level.

Under the selection constraint, a set of groups can contain zero or more groups, with each group containing 2 or more dots. The number of groups and their sizes are limited by the number of dots in the stimulus. For each numerosity level, an exhaustive list of group-size configurations was generated and rank ordered by group size. Table A1 shows the complete set of group-size configurations that can be found for patterns with 6 dots. The number of different group-size configurations varies with numerosity; there are 11 group-size configurations for 6-dot stimuli, and 627 group-size configurations for 20-dot stimuli.

The function relating group-size configuration G_i to the number of sets of groups it specifies is defined as $f(G_i)$. Table A1 shows the values of $f(G_i)$ from $i = 1$ to m (where m is the number of G at numerosity level n, for example, when $n = 6$, $m = 11$; see Table A1), for numerosity level $n = 6$. T_n, the number of selection-constrained sets of groups at numerosity level n, is defined as

$$T_n = \sum_{i=1}^{m} f(G_i).$$

When $k_i = 0$, where k_i is the number of groups in G_i, $f(G_i) = 1$. When $k > 0$,

$$f(G_i) = \frac{\prod_{j=1}^{k} \binom{n-q}{G_{i(j)}}}{D},$$

where q indicates the number of dots that are no longer available to be selected. When $j = 1$, $q = 0$. When $j > 1$,

$$q = \sum_{b=1}^{j-1} G_{i(b)}.$$

The denominator D addresses the situation in which G_i contains one or more sets of two or more groups of equal size. Since the algorithms under study are said to operate in parallel, differences among sets of groups that are due to the order in which identically sized groups are selected during the computation of $f(G_i)$ must be factored out. If G_i contains no sequences in which $G_{i(j)} = G_{i(j+1)}$, the denominator $D = 1$. If any such sequences are present, the denominator D is defined as

$$D = \prod_{w=1}^{v} R_w!,$$

where v is the number of sequences of identically sized groups in G_i, and R_w is the length of each such sequence. For example, Table A1 indicates that for $n = 6$ and $i = 11$, $v = 1$ and $R_1 = 3$, so for $f(G_{11})$, $D = 6$.

Table A1
Complete Set of Group-Size Configurations for Patterns Containing Six Dots

i	$G_{i(j)}$	k_i	$f(G_i)$
1		0	1
2	2	1	15
3	3	1	20
4	4	1	15
5	5	1	6
6	6	1	1
7	2, 2	2	45
8	3, 2	2	60
9	3, 3	2	10
10	4, 2	2	15
11	2, 2, 2	3	15
			$T_6 = 203$

Note—Index to group size configurations (i), group sizes ($G_{i(j)}$), number of groups in G_i, (k_i), and $f(G_i)$, the number of sets of groups that can be derived from G_i, are shown.

APPENDIX B

The grouping algorithms in this study created a strength gradient surface to represent the positional structure of each dot pattern. Once a surface was created, a search process began in an attempt to find, for that stimulus, as many sets of groups as possible. This search process took advantage of the monotonically decreasing nature of the spread functions used. Imagine a situation in which an algorithm has identified all possible sets of groups (as specified by the algorithm constraint) for a particular stimulus (e.g., the one with dots ABCD shown in Figure 2B). There will exist at the lowest threshold a set of groups that specifies a single group containing all of the dots in the pattern {ABCD}, a set of groups with two distinct groups or single dots at the next highest threshold {ABC},{D}, and so on until the last set of groups, which contains only single dots {A},{B},{C},{D}, is identified at the highest threshold. This ordered relation between the threshold value and the number of groups and single dots that threshold specified allows the search algorithm to search for any potential set of groups that may lie between the threshold values associated with two previously identified sets of groups. For example, if a set of groups has been identified that contains two groups or individual dots, and if another has been identified that contains four groups or individual dots, the search can begin for the potential set of groups that contains three groups or individual dots.

At each iteration of the search process, a new threshold is set half way between the highest known threshold of the set of groups with the lower threshold and the lowest known threshold of the set of groups with the higher threshold. If this threshold specifies not a new set of groups, but instead one of the adjacent sets of groups, the known threshold boundary of the appropriate set of groups is updated, and the search process repeats until a new set of groups is identified (or until the process terminates, after 40 consecutive searches for the same potential set of groups). The search process must be terminated after 40 iterations, because at this point the increment at which the threshold is being changed is so small that it begins to approach the resolution limit of the strength gradient surface (represented in the computer implementation as 8-byte double-precision real numbers). The iterative search process allows for the detection of differences between sets of groups less than 2^{-40} times the distance from the base to the maximum height of the strength gradient surface.

(Manuscript received February 3, 1992;
revision accepted for publication October 1, 1992.)

Forthcoming Articles

The following is a list of forthcoming *Perception & Psychophysics* articles that are currently in press. They are given in approximate order of acceptance. Each entry includes the name and address of the author with whom to communicate for further prepublication information.

"Time estimation and concurrent nontemporal processing: Specific interference from short-term memory demands" by C. Fortin, R. Rousseau, P. Bourque, & E. Kirouac (C.F., Ecole de Psychologie, Univ. Laval, Quebec G1K 7P4, Canada)

"Odor intensity interaction in binary and ternary mixtures" by B. Berglund & M.J. Olsson (B.B., Dept. of Psychology, Stockholm Univ., S-106 91 Stockholm, Sweden)

"Further evidence for phonological constraints on visual lexical access: Towed primes FROG" by G. Lukatela, K. Lukatela, & M.T. Turvey (G.L., Haskins Labs., 270 Crown St., New Haven, CT 06511)

"Long-term training, transfer, and retention in learning to lipread" by D.W. Massaro, M.M. Cohen, & A.T. Gesi (D.W.M., Program in Experimental Psych., Univ. of California, Santa Cruz, CA 95064)

"Two effects of context on the presence/absence of connecting segments" by D.L. King & J. Thomas (D.L.K., Dept. of Psychology, Howard Univ., 525 Bryant St. N.W., Washington, DC 20059)

"Individual differences in perceived pinch force and bite force" by C.L. Van Doren (MetroHealth Medical Ctr., H-611, 2500 MetroHealth Dr., Cleveland, OH 44109-1998)

"Self-terminating vs. exhaustive processes in rapid visual and memory search: An evaluative review" by T. Van Zandt & J.T. Townsend (J.T.T., Indiana Univ., Dept. of Psychology, Bloomington, IN 47405)

"A quantitative analysis of illusion magnitude predicted by several averaging theories of the Müller-Lyer illusion" by P.R. DeLucia (Psychology Dept., Texas Tech Univ., Lubbock, TX 79409-2051)

"Cortical involvement in visual scan in the monkey" by R.B. Bolster & K.H. Pribram (K.H.P., Radford Univ., Box 6977, Radford, VA 24073)

"Redundancy gains and coactivation with two different targets: The problem of target preferences and the effects of display frequency" by J.T. Mordkoff & J. Miller (J.T.M., Ctr. for Human Information Processing, Univ. of California, San Diego, La Jolla, CA 92093-0109)

"Satiation or availability?: Effects of attention, memory, and imagery on the perception of ambiguous figures" by K.L. Horlitz & A. O'Leary (A.O., Dept. of Psychology, Tillet Hall, Kilmer Campus, Rutgers Univ., New Brunswick, NJ 08903)

"Spatial context affects the Poggendorff illusion" by M. Spivey-Knowlton & B. Bridgeman (M.S., Ctr. for Visual Science, Meliora Hall, Univ. of Rochester, Rochester, NY 14627)

"The spatial signal for saccadic eye movements emphasizes visual boundaries" by J.M. Findlay, D. Brogan, & M.G. Wenban-Smith (J.M.F., Dept. of Psychology, Univ. of Durham, South Rd., Durham DH1 3LE, England)

"Perceptual enhancement of tumor targets in chest X-ray images" by E.A. Krupinski, C.F. Nodine, & H.L. Kundel (E.A.K., Dept. of Radiology, Radiology Research Division, Univ. of Arizona Health Sciences Ctr., Tucson, AZ 85724)

"The generation of temporal and melodic expectancies during musical listening" by M.G. Boltz (Dept. of Psychology, Haverford College, Haverford, PA 19041)

"The indispensability of precollision kinematics in the visual perception of relative mass" by S. Runeson & D. Vedeler (S.R., Dept. of Psychology, Uppsala Univ. Box 1854, Uppsala S-75148, Sweden)

"The effect of edge orientation and movement direction on the apparent misalignment of collinear bars" by R.H. Day, E.J. Stecher, & A.L. Parker (R.H.D., Monash Univ., Clayton Victoria 3168, Australia)

"Testing the speech unit hypothesis with the primed matching task: Phoneme categories are perceptually basic" by S. Decoene (Centrum voor matematische psych, en psych metodologie, Tiensestraat 102, 3000 Leuven, Belgium)

"Distinctness of perceived three-dimensional form induced by modulated illumination: Effects of certain display and modulation conditions" by S.S. Bergstrom, K. Gustafsson, & T. Jakobsson (S.S.B., Dept. of Applied Psychology, University of Umeå, S-901 87 Umeå, Sweden)

"Factors affecting the onset and magnitude of linear vection" by L. Telford & B.J. Frost (L.T., Dept. of Neurology, Box 605, Univ. of Rochester, 601 Elmwood Ave., Rochester, NY 14642)

"Limits of focused attention in three-dimensional space" by G.J. Anderson & A.F. Kramer (G.J.A., Dept. of Psychology, Univ. of California, Riverside, Riverside, CA 92521-5242)

Continued on page 428