Robust and Efficient Detection of Salient Convex Groups

David W. Jacobs

Abstract—This paper describes an algorithm that robustly locates *salient* convex collections of line segments in an image. The algorithm is guaranteed to find all convex sets of line segments in which the length of the gaps between segments is smaller than some fixed proportion of the total length of the lines. This enables the algorithm to find convex groups whose contours are partially occluded or missing due to noise. We give an expected case analysis of the algorithm's performance. This demonstrates that salient convexity is unlikely to occur at random, and hence is a strong clue that grouped line segments reflect underlying structure in the scene. We also show that our algorithm's run time is $O(n^2\log(n) + nm)$, when we wish to find the m most salient groups in an image with n line segments. We support this analysis with experiments on real data, and demonstrate the grouping system as part of a complete recognition system.

Index Terms—Grouping, perceptual organization, convexity, proximity, nonaccidental properties, robust, efficiency, recognition.

1 Introduction

In this paper, we consider *grouping* (or *perceptual organization*) as a bottom-up process that clusters image features into higher level organizations, each likely to come from a single object. Two significant questions arise in grouping. First, which features should we cluster together? Second, how can we locate these clusters efficiently? We approach the first problem by looking for *nonaccidental properties*. In this approach, we group together features that have some property that is frequently shared by features originating in a single object, but that is unlikely to arise at random. In particular, we consider convexity, which is a common characteristic of object parts and object faces. We show that convex sets of line segments with relatively small gaps rarely occur at random. We then show that this contributes to a solution to the second problem; the very fact that these groups are rare enables us to find them efficiently.

It has long been recognized that grouping is a difficult problem, which perhaps explains its relative neglect. Marr [41] said:

> The figure-ground "problem" may not be a single problem, being instead a mixture of several subproblems which combine to achieve figural separation, just as the different molecular interactions combine to cause a protein to fold. There is in fact no reason why a solution to the figure-ground problem should be derivable from a single underlying theory.

Marr recommended focusing on problems that have a "clean underlying theory," instead. These more manageable problems included shape-from-shading, edge detection, and object representation. More recently, Huang [23] has stated:

> Everyone in computer vision knows that segmentation is of the utmost importance. We do not see many results published not because we do not work on it but because it is such a difficult problem that it is hard to get any good results worthy of publication.

While grouping has proven a difficult problem, it deserves attention because even partial progress can be quite valuable. For example, a grouping system can focus and improve a search for an object by a recognition system by collecting together features that are more likely to come from a single object than are a random collection of features. Many recognition systems now exploit simple grouping techniques. By extending our understanding of grouping, we extend the domain in which recognition is feasible. Also, groups of features are more distinctive than individual features, and are more readily matched between images. Hence, grouping can help in solving the correspondence problem in stereo or motion.

One approach to perceptual organization is to identify common properties of image groups that originate in a single object or process, and that occur relatively infrequently in groups generated by a random process. We describe the past work that has developed this approach in Section 2. In this paper we combine proximity and convexity as grouping clues, forming what we call *salient, convex* groups. We present an algorithm that finds all collections of convex line segments where the length of the line segments accounts for at least some fixed proportion of the length of their convex hull. The *salience* of a group is therefore measured by the percentage of the convex boundary that is identified as a brightness discontinuity. By taking account of the gap between the end of the group and its beginning, this salience measures also has the effect of enforcing a *closure* constraint on the groups that it finds. We show that a random process is unlikely to produce such groups. At the same time, salient convex groups can be very useful in domains in which objects frequently have convex faces or parts. And in such domains, salient convexity will be a strong clue that

• *The author is with NEC Research Institute, 4 Independence Way, Princeton, NJ 08540. E-mail: dwj@research.nj.nec.com.*

Manuscript received June 3, 1993; revised Aug. 28, 1995. Recommended for acceptance by D.P. Huttenlocher.
For information on obtaining reprints of this article, please send e-mail to: transactions@computer.org, and reference IEEECS Log Number P95147.

0162-8828/96$05.00 ©1996 IEEE

a group of lines originate in a convex part of a single object, as we will discuss.

We present an algorithm for finding salient convex groups that has several desirable qualities. First, the algorithm is simple and we demonstrate that it is efficient, on average. Second the algorithm is robust to noise and occlusion because it allows for gaps to appear in groups. Third the algorithm is robust to clutter, because it finds all image groups that meet a simple test of salience. Since salience is measured by the overall properties of a group, the presence of extraneous lines cannot divert the system into connecting lines that appear good locally, but do not lead to groups with strong overall salience.

In addition, we determine analytically and experimentally the likelihood of salient convexity occurring by chance. For realistic situations this likelihood is low. Consequently, for images of a wide variety of objects, the grouping system we describe will reliably find groups that come from a single object, while producing groups unlikely to be due to a random process.

Our analysis and experiments on real data also show that precisely because salient convex groups are unlikely to occur at random, they can be found efficiently using a simple, backtracking search. We show that the expected time required to find the m most salient convex groups in an image containing n line segments is $O(n^2 \log(n) + mn)$. This leads to a grouping algorithm that is efficient and robust in practice. We demonstrate this by showing the performance of the algorithm on a number of real images. We also show a sample application of the algorithm to object recognition.

2 Relation to Past Work

We will emphasize three aspects of past work on perceptual organization. First, we describe previous uses of nonaccidental properties as grouping clues. We describe how our current work derives from and differs from these approaches. Second, we consider past efforts at making grouping systems efficient. Third, we describe the role that convexity has played in a variety of approaches to image understanding.

The use of nonaccidental properties in grouping has been developed theoretically by Witkin and Tenenbaum [58], Binford [5], Kanade [32], Lowe [38], and Richards and Jepson [49]. These authors argue that the visual system is sensitive to properties commonly produced by a single object or process, and rarely occurring at random. They show how this approach can explain a number of well-known Gestalt grouping phenomena, including grouping due to symmetry, parallelism, smooth continuation, and proximity (see Kohler [35] for a discussion of the Gestalt work. For more recent consideration of these ideas, see Kubovy and Pomerantz [37]).

Lowe in particular has contributed to this approach by showing how to derive probability distributions that reflect the strength of grouping clues. Lowe, for example, shows that parallel image lines are relatively likely to come from a single object. This derivation assumes that objects tend to contain parallel lines, and that lines coming from different objects have random relative orientations. One must allow for noise in detecting these properties, and so, for example, nearly parallel lines are grouped together in Lowe's system. The presence of noise, however, implies that nonaccidental properties like parallelism indicate that a grouping is likely, but not certain, to be correct. Because these types of clues are probabilistic, one expects to achieve better performance by combining many clues together, and this seems to be the experience of many researchers (Lowe [38], Jacobs [26], Shashua and Ullman [53], Denasi et al. [13], Sarkar and Boyer [50], Mohan and Nevatia [42], [43], Williams [57], and Nitzburg and Mumford [44]). For additional computational work that uses Gestalt grouping clues, see Reynolds and Beveridge [48], Parent and Zucker [45], and Trytten and Tuceryan [55].

Lowe [38] also stressed the importance of grouping to object recognition. He pointed out that by grouping together features that are likely to have been all produced by a single object, one can intelligently order one's search for that object. Others have also explicitly used grouping, or the formation of more complex features, to speed up recognition systems (Jacobs [25], [26], Califano and Mohan [8], Clemens [10], Syeda-Mahmood [54], Burns and Riseman [7], Huttenlocher and Wayner [24], and Wayner [56]). The interaction between grouping and the computational complexity of recognition is treated more theoretically by Grimson [17] and Clemens and Jacobs [11].

Most past work on contour grouping has focused on using a small set of properties identified by Gestalt psychologists: proximity, symmetry, parallelism, collinearity and smooth continuation (although Williams [57] has explored the role of topological constraints in grouping). Work in computer vision has stressed that there is zero probability that a perfect instance of any of these properties will occur as a result of some appropriate, uniformly distributed random process. For example, lines positioned by a uniform random distribution have zero probability of being parallel, or having a common end point. Convexity differs from these properties in that lines positioned with a uniform random distribution have a non-zero likelihood of being convex. As we will show, convexity provides only some probabilistic information about the correctness of a group. Since in practice, nonaccidental properties also provide only probabilistic grouping information, the difference is one of degree rather than kind.

Computational efficiency is a second important issue in perceptual grouping. Approaches have spanned a wide spectrum from efficiently computing simple local relationships between image features to more complex algorithms that attempt to integrate more global image information.

Lowe's system, SCERPO, provides an example of simple and efficient computation of pairwise relationships. It groups together pairs of lines that are parallel, collinear, or nearby. SCERPO then forms larger groups by finding chains of features connected by these pairwise relations. The system has low complexity, but the decision about what lines to group together rests essentially on the relationship between just two lines. As another example, Cox, Rehg, and Hingorani [12] describe a system that uses a Bayesian approach to finding smooth curves, based on ideas from target tracking. This method makes smoothness decisions based on the recent history of a curve. Huttenlocher and Wayner [24] also use local relationships in forming convex groups. They begin with each side of each line segment as a convex group, and then extend a group by adding the nearest neighbor that will preserve its convexity. By only making the best local extension to each group, they guarantee that the output will be linear in the size of the input, and they produce an efficient algorithm. This can, however, make their algorithm sensitive to small local perturbations in the image.

Other approaches have attempted to efficiently integrate more global information into grouping. For example, a hierarchical approach can be used that makes local decisions at a variety of scales, allowing for the integration of information from more spatially separated parts of the image. Boldt, Weiss, and Riseman [6] take this approach in detecting collinear groups, while Dolan and Riseman [14] and Saund [51] apply hierarchical methods to cocircularity detection. Mohan and Nevatia [43] integrate several gestalt grouping clues, using a hierarchical algorithm. Mahoney [40] uses parallel processing to gain efficiency in an algorithm that extracts smooth, connected curves. Other researchers have used methods such as convolution or relaxation to integrate smoothness information (Zucker [60], Finkel and Sajda [16], Heitger and Van der Heydt [19], Hérault and Horaud [20], and Guy and Medioni [18]). Other network approaches include Hopfield nets (Mohan and Nevatia [42]) and Bayes nets (Sarkar and Boyer [50]). Typically these systems are not able to formulate and then optimize a simple measure of the value of a group. Rather, clever methods are used to find groups that bear a strong, though heuristic relationship to some clearly desirable grouping criteria, or heuristic optimization techniques such as gradient descent are used.

In contrast, Shashua and Ullman [53] find the image curve that will explicitly optimize a cost function based on the total curvature and number and size of gaps in a curve. This is computed using dynamic programming, in a network. The system requires $O(N\,kn^2)$ computation, where the image is of size n^2, the system represents k discrete orientations at each pixel ($k = 16$ in their experiments), and the system finds a curve of length N. From this, they can extract the most salient curve in the image. In related work, Hu, Sakoda, and Pavlidis [22], and Alter [1] use shortest path algorithms to find globally optimal curves.

Our system differs from these approaches in that we find all groups that satisfy a global grouping criteria. For example, our salience criteria is scale independent, unlike the cost functions of the above systems. It appears to be difficult to optimize interesting scale-independent cost functions with dynamic programming. Alter [1] contains a discussion of this problem, and of other limitations of Shashua and Ullman's system.

Third, we point out that convexity and closure are significant cues in human and machine vision. Elder and Zucker [15] and Kovacs and Julesz [36] provide evidence that closure plays a significant role in human perceptual organization, and this cue has been used in the grouping systems of Mohan and Nevatia [43] and Zerroug and Nevatia [59].

Kanisza and Gerbino [33] have shown that convexity can play a strong role in human judgements of figure and background, stronger, in fact, than symmetry. Kellman and Shipley [34] also discuss the significance of inflexion points in perceptual organization, which is related to the convexity of connections. Jacobs [25], [26] presents an analysis of convexity and proximity as grouping clues that is in many ways complementary to the one presented here. That work considers the effectiveness of these cues in grouping pairs of already formed convex sets of edges. The work described here is more robust and practical, because it provides a method of finding convex chains of line segments in the presence of noise. However, the previous work provides a more thorough analysis of proximity as a grouping clue, and treats the orientations of groups of lines in general, not just orientations that are mutually convex. Kalvin et al. [34] describes a 2D recognition system that uses convex curves to index into a library of objects, for recognition. Jacobs [25], [26] also describes a system that combined several convex portions of the image to find enough information to index into a library of objects. This system demonstrated that grouping using convex parts could speed up recognition by a factor of several hundred to a thousand. Based on the work in Jacobs [25], [26], Huttenlocher and Wayner [24], and Wayner [56] use convexity as a grouping method, followed by an indexing method of object recognition. Basri and Jacobs [2] describes a recognition method that depends explicitly on matching convex regions of images to convex components of objects.

A variety of authors have proposed more general approaches to recognition that rely on finding the parts of objects. Hoffman and Richards [21], for example, suggest dividing objects into parts at concave discontinuities. And Biederman [4] suggests performing recognition using the invariant qualities of an object's parts and their relations. These parts tend to be convex. In fact, in implementing a version of Biederman's work, Bergevin and Levine [3] rely on convexity to find the parts of an object.

Convexity may be useful for other types of matching problems, such as motion analysis or stereo. Mohan and Nevatia [42], for example, perform stereo matching between groups of line segments that form partial parallelograms in each image. This reduces the combinatorics of matching. Sawhney [52] uses convexity and proximity as clues for forming groups to track in motion sequences. Such structure is likely to come from a single object face, which is necessary to the tracking method used.

To summarize, the novelty of the work presented here lies in three areas. First, we present a concrete derivation that shows how to interpret convexity within the framework of nonaccidental properties. Second, we present a new approach to efficiently implementing grouping methods. The efficiency of our approach comes not from making local grouping decisions, but from the fact that our grouping criteria are unlikely to be met by random groups. Third, we present a robust tool that may be of value in a number of approaches to image understanding that require finding convex regions of images.

3 Precise Statement of the Problem

The system begins with line segments that we obtain by running a Canny edge detector [9] (in the experiments shown, $\sigma = 2$), and then using a split-and-merge algorithm based on Horowitz and Pavlidis [46] to approximate the edges to within three pixels, by straight lines.

We call a line segment "oriented" when one endpoint is distinguished as the first endpoint. If l_i is an oriented line segment, then $l_{i,1}$ is its first endpoint, and $l_{i,2}$ is its second. The image contains n line segments, and so it has $2n$ oriented line segments. A set of oriented line segments is convex if for each oriented line segment, all the other line segments are on the same side of the oriented line segment as its normal, where we define the normal as lying to the right when we traverse the line segment from the first endpoint to the second.

Let S_n be a cycle of oriented line segments: $(l_1, l_2, \ldots l_n)$ (i.e., l_1 follows l_n). We define L_i to be the length of l_i, and we define G_i to be the distance (or *gap* between $l_{i,2}$ and $l_{i+1,1}$ (where l_{n+1} is l_1). We then let:

$$L_{1,n} = \sum_{i=1}^{n} L_i, \qquad G_{1,n} = \sum_{i=1}^{n} G_i.$$

We define the *salience fraction* of a convex group to be:

$$\frac{L_{1,n}}{L_{1,n} + G_{1,n}}$$

(see Fig. 1) and we say that S_n is *valid* if and only if connecting the line segments in sequence would create a convex polygon with a salience fraction greater than some fixed threshold, k.

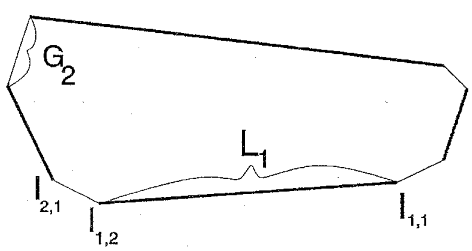

Fig. 1. The thick lines represent the lines in a group. The thin lines show the gaps between them. The salience fraction is the sum of the length of the thick lines divided by the sum of the length of all the lines.

The output of this algorithm satisfies a simple, global criterion. One way to think of the salience fraction is that if the lines forming a group originally came from a closed convex curve, the salience fraction tells us the maximum fraction of the curve's boundary which has been detected as image edges. Previous approaches to convex grouping have either grouped one line with another if it is the nearest, mutually convex line ([26], [24]) or if it is within some preset distance ([52]). Because the salience threshold depends on the overall structure of the group, our algorithm can bridge gaps whose size is dependent on the overall group size, and hence is scale independent. Furthermore, spurious lines near a group will not prevent the algorithm from considering more distant lines that lead to groups with a strong overall salience.

4 The Grouping Algorithm and its Complexity

In this section we present an algorithm for finding these salient convex groups. We begin by describing a basic backtracking algorithm. We analyze this algorithm theoretically in order to bound its expected run time and the expected size of its output. We show that the actual results of running the algorithm match our theoretical predictions. Both our experiments and theoretical analysis indicate that the expected running time of the algorithm will be $O(n^2 \log(n) + mn)$ when we use a salience threshold designed to find m groups in an image having n lines. Furthermore, our analysis demonstrates that salient convexity can be a strong clue that a configuration of lines did not occur at random.

Finally, we make some modifications to the basic algorithm, which make it more robust, but which make complexity analysis more difficult. Accordingly, we use experiments to show that these modifications do not significantly affect the algorithm's performance.

4.1 The Basic Algorithm

Our problem definition takes the form of global constraints on the groups of line segments we seek. To perform an effective backtracking search, we must convert these into local constraints whenever possible. That is, we need constraints that determine whether a sequence of line segments could possibly lead to a valid sequence, so that we may prune our search.

We provide the following recursive definition of an *acceptable* sequence, assuming that S_i is acceptable.

1) Any sequence of a singleton, oriented line segment is acceptable.
2) S_{i+1} is acceptable only if $l_{i+1} \notin S_i$ (i.e., no line may appear more than once in a group).
3) S_{i+1} is acceptable only if the oriented line segments in it are mutually convex. This will be the case if the sum of the absolute values of the angles turned is 2π when one travels from the first endpoint of the first line to each additional endpoint in turn, returning finally to the first endpoint.
4) S_{i+1} is acceptable only if: $G_i < \frac{L_{1,i}(1-k)}{k} - G_{1,i-1}$. This is equivalent to stating that $\frac{L_{1,i}}{L_{1,i}+G_{1,i}} > k$.

We prune all unacceptable sequences reached in our search. It is not hard to see that enforcing these constraints will not eliminate any correct groups, while guaranteeing that the search produces only correct groups. We prove this in [28].

To further reduce the run time of our algorithm we notice that some values are reused many times in the course of such a search, so we precompute these results and save them in tables. In particular, we often wish to know whether two oriented line segments are mutually convex, and if they are we want to know the distance from the end of one segment to the beginning of the other. It is also convenient to keep, for each oriented line segment, a list of all other oriented line segments with which it is mutually convex, sorted by the distance that separates them. Finally, we precompute the angle that is turned when going from one oriented line segment to another. Calculating this information takes $O(n^2 \log(n))$ time, because we must sort $2n$ lists that can each contain up to $2n - 2$ items.

We may now describe the backtracking search in more detail, noting how these results are used. The search begins by trying all oriented line segments in turn as singleton sequences. Given an S_i, we calculate $\frac{L_{1,i}(1-k)}{k} - G_{1,i-1}$. From constraint 4), we know that we only want to consider adding a line, l_{i+1}, when the distance from $l_{i,2}$ to $l_{i+1,1}$ is less than or equal to this quantity. Clearly we only want to add l_{i+1} if it is mutually convex with l_i. So we can find all candidates to add to the sequence by referencing the precomputed list of line segments that are convex with l_i. Since these lines are sorted by their distance from l_i we may loop through them, stopping once we reach line segments that are too distant to consider. By limiting ourselves to these candidates, we have enforced constraint 4). In addition, we check that l_{i+1} is convex with l_1 using our precomputed results.

We can then enforce constraint 3) by keeping a running count of the angles turned as we traverse the line segments in S_i. A table lookup will tell us the angles added to go from l_i to l_{i+1} and from l_{i+1} to l_1. Therefore, we can ensure that the entire sequence is mutually convex by checking that the absolute values of the angles turned in traversing it sum to 2π. And constraint 2) is simply checked explicitly.

4.2 Complexity Analysis of the Basic Algorithm

We now analyze the performance of this algorithm on randomly generated line segments. This analysis has two goals. First, we wish to understand when salient convexity is a useful grouping clue. By determining the expected number of groups produced, as the salience threshold, k, and the number of lines, n, vary, we obtain a basis for determining the relative likelihood that a group was produced by a scene structure, as opposed to a random process. This analysis will lead us to a method for choosing k appropriate to the input size, so as to produce manageable and useful output. Second, we want to understand when the algorithm will be fast. We also determine the algorithm's expected run time, and show that if we choose a salience fraction that will produce a manageable sized output, then the system's expected run time will also be reasonable.

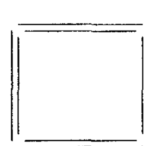

Fig. 2. A salient convex group may be formed by choosing any line from each of the four sides.

In the worst case, the algorithm will be exponential in both run time and in the size of its output, for any choice of k. As a simple example of this, in Fig. 2 we show eight lines formed into a squarish shape. Even for fairly high values of k, we may form a salient convex group using either of the two lines on each side of the square, for a total of 2^4 groups. If we formed n-tuples of lines around an m sided convex polygon we would have an output of at least m^n groups. By making the sides' endpoints close together, we can ensure that these groups are judged salient for any value of k less than 1. And the work required by the system is at least equal to the size of the output.

However, one way to understand the effectiveness of salient convexity as a nonaccidental property is to determine the expected size of the output produced with random data. We also find that performance on real data is well predicted by this analysis, and so it provides a better understanding of the system's typical performance. We need first to choose a model of random image generation. We have chosen a simple random model, for which we provide a conservative estimate of our algorithm's performance.

Perhaps the most generic possible model of random line generation would be to assume that lines have lengths, angles and positions chosen from independent uniform distributions. If we assume that camera position is random, there is no reason to expect any bias in line orientation or position so uniform distributions on these variables are particularly natural. Nonuniformity might, however, occur in line length in natural settings. For example, if three-dimensional lines were uniformly distributed in the world, one would tend to see more distant lines than nearby ones, and these distant lines would appear shorter. Furthermore, the actual distribution of line lengths we find in images will depend on our methods of detecting edges and approximating them with line segments. In the analysis that follows, we will none-the-less assume that the image contains n line segments with lengths uniformly distributed between 0 and M, the maximum allowed length. But it is straightforward to repeat the analysis for other distributions, and to see that distributions that favor shorter lines will lead to results that indicate that less work is required to find salient convex groups than our present analysis indicates.

We will simplify our analysis by assuming that the beginning point of a new oriented line segment is uniformly distributed within a circle of fixed radius, $R = M$, centered at the second endpoint of the last oriented line segment in the current sequence. This case is more tractable to analyze than the assumption that lines are uniformly distributed throughout the image, because through this new assumption the position of each line becomes independent of the structure of the previous lines in a group. This assumption will cause us to conservatively estimate the work required to find convex groups, since randomly distributed lines are closest together when distributed in a circle, leading to more saliency.

We begin by examining the effectiveness of distance and angle constraints in pruning our search. The distance constraint must be met by acceptable sequences to satisfy condition 4, the angle constraints are required by condition 3. Using these results, we compute the expected work and output size of the search, assuming a particular value of k. This allows us to see how we must adjust k as n grows to keep the output size reasonable, and to see how work depends on k, n, or the desired output size.

4.2.1 The Distance Constraint

The probability that the distance constraint is met as we add line i to a group is given by:

$$Pr\left(G_i < \frac{L_{1,i}(1-k)}{k} - G_{1,i-1}\right)$$

We define:

$$r_i = \frac{L_{1,i}(1-k)}{k} - G_{1,i-1}$$
$$s_i = r_i - G_i$$

and let $h(r_i)$ and $g(s_i)$ denote probability density functions on r_i and s_i respectively. Note that:

$$r_i = s_{i-1} + k'L_i$$

where $k' = \frac{1-k}{k}$. We may recursively compute $h(r_i)$ and $g(s_i)$.

To initialize the recursion, we note that the density function of l_i is $\frac{1}{M}$, and the density:

$$h(r_1) = h(k'l_1) = \frac{1}{Mk'}, \quad \text{for } 0 \le r_1 \le Mk'.$$

We may then show that:

$$g(s_i) = \int_{s_i}^{\max(r_i)} h(r_i)\left(\frac{2}{r_i} - \frac{2s_i}{r_i^2}\right)dr_i$$

and

$$h(r_{i+1}) = \int_{\max(0,(r_{i+1}-k'M))}^{r_{i+1}} \frac{g(s_i)}{Mk'}ds_i$$

(see [28] for details). In fact, this integral is a bit of a simplification, because in deriving it we do not take account of the fact that r_i may never be bigger than R. If we ignore this effect, we are slightly exaggerating the likelihood of an additional line meeting the distance constraint, and hence overestimating the work that the system performs.

The distance constraint is met provided that $l_{i,1}$ falls somewhere in a circle of radius r_i. This occurs with probability:

$$Pr\left(\left\|l_{(i-1),2}l_{i,1}\right\| \le r_i\right) = \int_0^R h(r_{i-1})\frac{r_{i-1}^2}{R^2}dr_{i-1}$$

which we may compute after computing $h(r_i)$.

4.2.2 The Angle Constraint

There are two parts to our treatment of the angle constraint. First, we consider the probability that a line that passes the distance constraint will be locally convex with just the previous line. When a line, l_i, passes the distance constraint, we know that $l_{i,1}$ will be uniformly distributed in a circle about $l_{(i-1),2}$. The location of $l_{i,1}$ in this circle is constrained to lie in a wedge (see Fig. 3), and so the probability of this occurring depends only on the angle of the wedge. If we define a_i to be the angle of line l_i relative to the x axis, then the angle of the wedge is $a_i - a_{i-1}$, provided that $a_i - a_{i-1} \le \pi$, and otherwise is $2\pi - (a_i - a_{i-1})$. The probability of l_i being compatible with l_{i-1} is just the angle of this wedge divided by 2π. Integrating over all angles, we find that there is a probability of $\frac{1}{4}$ that the lines will be compatible.

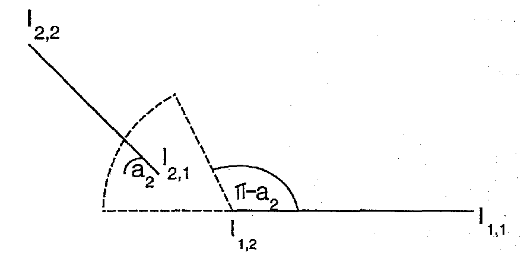

Fig. 3. The dashed circular wedge shows where $l_{2,1}$ must lie in order to satisfy the distance and convexity constraints.

We must also consider the probability that a sequence of i lines will be mutually convex. We derive a density function on the sum of the angles that will be turned as we go from one line to the next, and determine the probability that this sum is less than 2π. This is a necessary, though not a sufficient condition on convexity. The density function for each such angle is independent of the others and of the distance between the lines. So we need only consider the density on one of these changes of angle, and then convolve this density with itself i times to find the density of the sum of i such angles.

The angle of one line relative to the previous one is uniformly distributed between 0 and 2π. We need then derive a density function for the difference between the angle of one line and the next one in a group, given that the new line is convex with respect to the previous one. If we let f be this probability density function, we have:

$$f(a_i - a_{i-1}) = \begin{cases} \frac{a_i - a_{i-1}}{\pi^2} & \text{if } a_i - a_{i-1} \le \pi \\ \frac{2\pi - (a_i - a_{i-1})}{\pi^2} & \text{if } a_i - a_{i-1} > \pi \end{cases}$$

(again, this is derived in [28]).

4.2.3 Expected Work of the Algorithm

As we have stated, there is a fixed overhead of $O(n^2\log(n))$ work. We will now determine the expected number of nodes our algorithm searches. We sum over all i the amount of work that must be done when we consider extending each group of i lines by an additional line. We must consider two orientations for each line segment, so there are $2n(2n-2)\ldots(2n-2i+2)$ possible ordered sequences of i line segments. Let δ_i be the probability that the ith line will pass the distance constraint, given that the previous lines have, and let λ_i be the probability that a group of otherwise compatible lines will have angles that sum to less than 2π. Then the expected number of groups of size i that we must consider, which we will call E_i, is:

$$E_i = \lambda_i 2n(2n-2)\ldots(2n-2i+2)\delta_2\ldots\delta_i\left(\frac{1}{4}\right)^{i-1}$$

where $E_1 = 2n$.

For each group with i lines that we reach, there are potentially $2n - 2i$ lines that we must consider adding to the group. However, our preprocessing has sorted these lines, so that we only need to consider the ones that meet the distance constraints and that are convex with the last line in our group. We call the expected number of possible extensions to a group of size i, X_i, and:

TABLE 1
THIS TABLE SHOWS THE EXPECTED NUMBER OF SEARCH STEPS REQUIRED TO FIND CONVEX GROUPS

Expected Work							
Number of lines	k						
	.6	.65	.7	.75	.8	.85	.9
200	1.99×10^7	2,270,000	346,000	65,800	15,100	4,280	1,650
300	5.82×10^8	4.14×10^7	4,200,000	561,000	93,200	19,000	5,200
400	8.73×10^9	4.23×10^8	3.04×10^7	3,020,000	386,000	61,000	12,700
500	8.53×10^{10}	3.05×10^9	1.62×10^8	1.24×10^7	1,260,000	162,000	26,800
700	3.45×10^{12}	8.09×10^{10}	2.65×10^9	1.29×10^8	8,830,000	791,000	90,700
1000	2.25×10^{14}	3.76×10^{12}	7.54×10^{10}	2.11×10^9	8.77×10^7	5,060,000	378,000

TABLE 2
THIS TABLE IS AN ADJUNCT TO TABLE 1. IT SHOWS THE EXPECTED NUMBER OF NODES EXPLORED IN THE SEARCH TREE DIVIDED BY THE NUMBER OF STEPS IN A PREPROCESSING PHASE

Ratio of search to preprocessing							
Number of lines	k						
	.6	.65	.7	.75	.8	.85	.9
200	14.4	1.65	.25	.048	.011	.003	.001
300	175	12.5	1.26	.169	.028	.006	.002
400	1,410	68.6	4.92	.489	.063	.010	.002
500	8,560	306	16.3	1.25	.127	.016	.003
700	168,000	3,950	130	6.28	.431	.039	.004
1000	5,140,000	85,800	1,720	48.2	2.00	.115	.009

TABLE 3
THIS TABLE SHOWS THE ACTUAL NUMBER OF NODES IN THE SEARCH TREE THAT WERE EXPLORED WHEN FINDING CONVEX GROUPS IN RANDOMLY GENERATED IMAGES. THE LENGTHS OF THE LINES WERE GENERATED FROM A UNIFORM DISTRIBUTION FROM ZERO TO HALF THE WIDTH OF THE IMAGE. THE POINTS WERE THEN RANDOMLY LOCATED IN A SQUARE IMAGE

Actual work for random lines							
Number of lines	k						
	.6	.65	.7	.75	.8	.85	.9
200	347,000	112,000	31,300	8,590	2,270	601	123
300	3,120,000	892,000	208,000	45,600	8,760	1,670	306
400	17,400,000	4,970,000	1,130,000	172,000	27,000	4,050	692

$$X_i = \frac{1}{4}(2n - 2i)\delta_{i+1}$$

Therefore, the total amount of work that we must perform in our search, W_n, is the number of possible extensions to groups:

$$W_n = \sum_{i=1}^{n} E_i X_i$$

This expression shows that asymptotically, the expected amount of work is exponential in n. We will show, however, that run time is polynomial in the output size. To predict at what point the computation will become unmanageable, we also compute the expected amount of work for a variety of realistic situations (further details of the computation are given in [28]).

Table 1 shows the expected work of the system as n and k vary. Table 2 shows the amount of work of the system divided by $(2n)^2\log(2n)$. This tells us roughly the proportion of the system's work that is spent in search, as opposed to fixed overhead, although one step of overhead is not directly comparable to one step of search. We see that even though the search is asymptotically exponential in n, for fixed k, this exponential growth does not dominate the system's performance in many realistic situations.

Later, we compare this to the results of the full system on real data. For now, we compare this theoretically derived estimate of the system's work to simulations on random data. This determines the effect of two approximations made in our analysis. First, we assumed that the end point of one line is uniformly distributed in a circle about the end point of a previous line. However, in simulation, lines will be uniformly distributed in a fixed image. Second, our analysis did not apply the full convexity constraints, because it does not consider whether a newly added line is convex with the first line in our group. We expect that these approximations should make our analysis conservative, overestimating the work required.

In this test we generate collections of random line segments in a square. Lines have uniformly distributed angles and a length chosen from a uniform distribution between 0 and half the width of the square. We draw the location of the line from a uniform distribution, assuming that the line is completely inside the image. Table 3 shows the results of these experiments for various combinations of values of k and n.

Comparing these results with Table 1 shows that our analysis overestimates the amount of work required by the system. Since we are overestimating the constants in an exponential series, we expect the overestimate to be more severe in situations where the higher order terms of our series come into play. This occurs as k shrinks and as n grows (i.e., when the number of larger groups considered becomes substantial). This is supported by the data. When our analysis predicts that only a few thousand nodes will

TABLE 4
THIS TABLE SHOWS THE EXPECTED NUMBER OF SALIENT CONVEX GROUPS THAT OUR ALGORITHM PRODUCES

Expected number of groups produced							
Number of lines	k						
	.6	.65	.7	.75	.8	.85	.9
200	51,400	5,850	885	166	36.3	8.75	2.09
300	996,000	70,600	7,120	946	154	28.8	5.62
400	1.12×10^7	539,000	38,600	3,820	484	72.8	11.8
500	8.69×10^7	3,100,000	165,000	12,500	1,270	158	21.7
700	2.50×10^9	5.86×10^7	1,920,000	92,800	6,340	561	57.8
1000	1.14×10^{11}	1.90×10^9	3.80×10^7	1,060,000	44,100	2,530	179

TABLE 5
THE ACTUAL NUMBER OF SALIENT GROUPS THAT WERE FOUND IN IMAGES OF RANDOMLY GENERATED LINE SEGMENTS

Actual number groups for random lines							
Number of lines	k						
	.6	.65	.7	.75	.8	.85	.9
200	7,500	2,590	782	234	70	15	5
300	45,800	13,800	3,540	867	159	32	5
400	189,000	55,500	13,300	2,300	392	58	6

be searched, we find that our analysis has overestimated the required work by about a factor of 10. This gap widens to the point where, when millions of search nodes are expected, this is actually an overestimate by about a factor of 40. Considering that we are predicting the behavior of an exponential process, this seems like a good (albeit conservative) analysis. It is sufficiently accurate to provide us with a good idea of the circumstances under which our run time will be dominated by preprocessing.

4.2.4 Expected Size of the Output

It is also important to determine the size of the output of our algorithm. First it will help us to assess the significance of salient convexity as a grouping clue. Second, if we can predict the size of the output ahead of time, we can use this to decide how high to set our salience threshold based on the number of lines in the image to produce an output of the desired size. Third, it will help us to see the extent to which the system's run time depends on the size of its output.

The expressions above for E_i provide the expected number of groups of any particular length that we will encounter in our search. We could use this as a bound on the size of the output, but this would be an oversimplification. Just because a group is reached in our search does not mean it will be accepted. When we reach a group of length i in our search, we have yet to take account of the length of the ith line, or the gap between the ith line and the first one. It is difficult to determine the probability distribution of this final gap, because it is dependent on the combination of i previous processes that built up the group. But we can approximate it very simply by assuming that the distribution of the relative position of the first and last lines in the group is the same as that between any other two lines. Using this approximation, the expected number of groups that the system will produce is:

$$O_n = \sum_{i=2}^{n} \frac{1}{4} \delta_{i+1} E_i$$

We sum for i from two up because groups of size one are never salient unless the salience fraction is less than or equal to .5.

Table 4 shows the number of groups that we expect to find using this method. Table 5 shows the number of groups that we found in experiments with random line segments. There is good agreement between these results. Our analysis typically overestimates the size of the output by about a factor of 2, except when the expected output is quite large.

This result has two implications. First, it allows us to identify those circumstances where salient convexity will provide a strong clue that a group did not emerge from a random process. As a simple example of this, suppose we expect an image of 300 lines to contain 20 convex groups that come from convex objects (or parts) and that have a salience threshold of .8 or more. Table 4 shows that we would expect to find only 154 convex groups with this salience. This implies that we would have to examine only 174 groups output by our algorithm to find the 20 correct groups. Without some grouping clue we might have to examine a vastly larger number of groups of lines before finding some that originate in a single object. Obviously salient convexity provides a powerful clue that a group originates with a single object in this example. If we can choose k so that few random groups will be salient, but some of the real structure of the scene will produce salient convex groups, we will have a valuable grouping clue.

Second, we can use this result to determine how the asymptotic run time of the algorithm depends on the output size. To see this, we consider a strategy in which we adjust k as n grows to maintain a constant sized output. Note that:

$$W_n = \sum_{i=1}^{n} E_i \frac{1}{4} (2n - 2i) \delta_{i+1}$$

And therefore:

$$W_n \leq 2nO_n + E_1 X_1 \leq 2nO_n + n^2$$

This tells us that, if we denote output size by m, our search is $O(nm + n^2)$, and total run time of the algorithm is $O(n^2 \log(n) + nm)$. This confirms that the algorithm will be efficient whenever the output is kept small.

This conclusion is based on derivations of W_n and O_n that are bounds on the true values. It is conceivable that this conclusion might not hold if there were a very tight bound

TABLE 6
THIS IS THE NUMBER OF NODES EXPLORED IN THE SEARCH TREE FOR SOME REAL IMAGES. THE SECOND COLUMN INDICATES WHETHER WE USED THE MODIFICATIONS TO THE ALGORITHM DESCRIBED IN THE TEXT TO MKE IT MORE ROBUST, OR WHETHER WE USED JUST THE BASIC ALGORITHM

Actual work for real images								
No.	Alg.	k						
lines	Type	.6	.65	.7	.75	.8	.85	.9
183	basic	1,800,000	548,000	133,000	28,900	7,030	2,000	613
	complete	284,000	166,000	75,000	37,000	15,400	6,710	2,470
265	basic	5,630,000	1,660,000	420,000	102,000	27,200	6,370	1,410
	complete	496,000	288,000	136,000	55,300	18,200	7,440	2,800
271	basic			816,000	193,000	47,000	9,740	1,590
	complete			106,000	59,400	24,200	9,810	3,840
296	basic	7,200,000	1,820,000	429,000	93,300	16,300	3,350	946
	complete	273,000	163,000	89,400	41,800	14,800	6,170	2,900
375	basic	689,000	226,000	78,600	27,100	9,620	3,410	1,390
	complete	201,000	104,000	54,600	31,300	18,500	9,610	3,610
450	basic	2,090,000	696,000	227,000	69,000	21,300	6,420	2,160
	complete	295,000	163,000	92,500	48,400	24,800	11,400	4,440
461	basic			72,000	26,700	9,970	3,560	1,250
	complete			105,000	37,500	19,300	8,200	3,130

TABLE 7
THIS TABLE SHOWS THE NUMBER OF SALIENT CONVEX GROUPS PRODUCED BY THE TWO VARIATIONS OF OUR ALGORITHM, WHEN APPLIED TO A NUMBER OF REAL IMAGES

Actual number of groups found for real images								
No.	Alg.	k						
lines	Type	.6	.65	.7	.75	.8	.85	.9
183	basic	16,300	4,760	1,170	248	110	60	35
	complete	2,160	1,190	494	315	134	69	31
265	basic	32,300	8,930	1,850	404	182	98	51
	complete	1,750	982	542	276	120	47	27
271	basic			5,600	598	148	64	36
	complete			540	291	157	65	34
296	basic	47,400	12,800	2,670	474	136	73	42
	complete	2,040	1,180	620	312	152	85	42
375	basic	23,100	11,100	5,250	2,390	1,020	406	163
	complete	1,680	1,020	536	331	188	122	48
450	basic	74,100	37,900	18,500	7,840	2,960	965	293
	complete	2,160	1,340	797	430	235	125	52
461	basic			754	376	194	96	49
	complete			863	368	178	84	34

on W_n, and a very loose bound on O_n. However, our experiments indicate that this is not the case. Our data suggest that run time grows roughly linearly with the size of the output. In Fig. 4 we plot the amount of work performed by the search, for fixed values of n, as k, and therefore the output size, varies. We can see that the search does grow linearly with the output size.

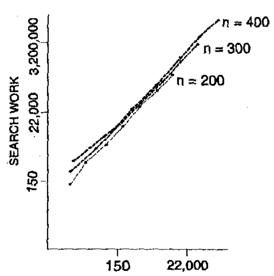

Fig. 4. The number of search steps performed on simulated data, plotted against output size. The plot is on a log-log scale. The three graphs show different values of n.

4.3 Run Time and Output Size for Real Images

Next we made a few practical modifications to our system. For robustness, we allowed for some sensing noise when computing convexity. Also, if an entire line segment violated convexity when added to a group, we consider adding only a portion of the line segment. In fact, even when convexity is preserved when an entire line segment participates in a group, we still allow the group to use only a subset of the line if that will increase the group's salience. Finally, we removed duplicate or near duplicate groups from the output of the system. See [28] for details. A Common Lisp version of this system can be obtained by anonymous ftp to external.nj.nec.com, in the directory "/pub/dwj/src/convex-grouping.tar."

We repeated our experiments to determine the effect these modifications have on the run time of the system and on the size of its output. We ran both the basic algorithm and the modified algorithm on a set of real images, so that by comparing these results to our previous results we can tell how much of the change is due to the use of real images, and how much is due to the additional constraints. We can see that the basic and modified algorithms have comparable performance.

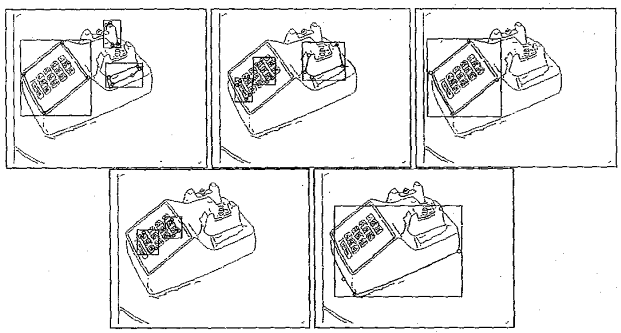

Fig. 5. This shows the most salient groups in an image of a telephone. These are all the groups that have a salience fraction of at least .75, and that consist entirely of oriented line segments that do not appear in more salient convex groups. The dotted lines show the edges of the image. Within the image there is a box around each separate group. Solid lines show the lines that form the group. Circles show the corners found in the group.

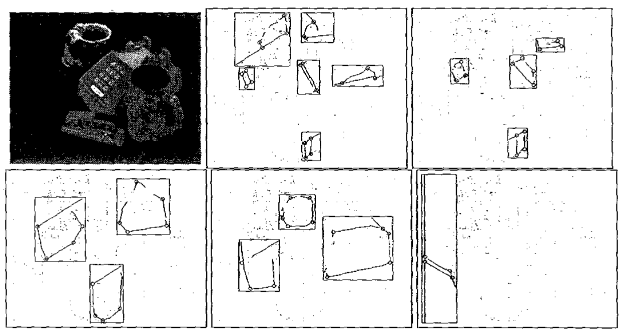

Fig. 6. Similarly, a scene is shown in the upper left corner, and the remaining pictures show all the most salient groups found.

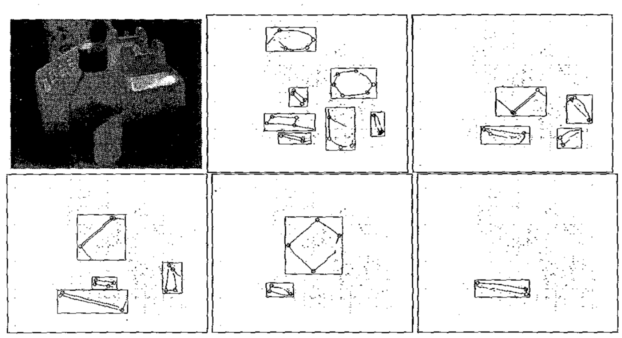

Fig. 7. The salient convex groups found in another scene.

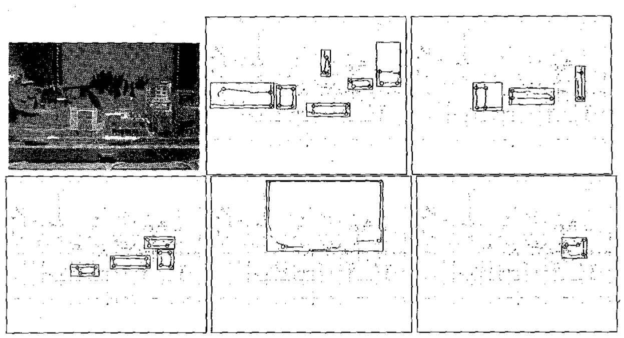

Fig. 8. A picture from the CMU calibration lab, courtesy of David Michael, along with the most salient groups found.

Section 5 shows examples of the images we used. Table 6 shows the number of nodes explored in the search, for both the basic and full systems on these and similar images. Table 7 shows the number of groups produced by both variations of the algorithm on these images.

We can see that our theoretical predictions almost always bound the performance of the system. Moreover, our analysis predicts the system's performance roughly to within an order of magnitude. For example, when fewer than one million search steps are expected, our analysis overestimates the work that will be required by the basic algorithm by roughly a factor of 2. This is quite good agreement, considering that the amount of search required by two different images of similar size may also vary by a factor of 2. Our predictions concerning the size of the output are similarly good.

We also measured the run time of our unoptimized computer implementation. The system ran on a Symbolics 3640 Lisp Machine.[1] On an image with 246 lines, the basic algorithm spent 48 seconds on preprocessing overhead. The search tree was explored at a rate of between 450 and 2,300 nodes per second. Our implementation of the complete algorithm was approximately a factor of 20 slower, largely because of additional time spent in preprocessing. However, our implementation of the complete algorithm was simple and inefficient, and we believe that most of the additional time it required could be eliminated in a more careful implementation. These numbers indicate that the overall system could be expected to run in a few minutes or less in a practical implementation.

5 Grouping Performance

So far, we have evaluated our grouping system in terms of its efficiency, and the probability that the groups it produces will reflect real scene structure, rather than a random process. We now present some examples of the system's performance on real images. In particular, we have integrated our grouping system into a complete recognition system, to demonstrate its potential value.

Our recognition system, described in detail in [28], first forms convex groups, as we have described here. Additionally, it only uses a group if none of its oriented line segments appear in a more salient convex group. This prevents it from having to consider several, similar groups. For each group, the system finds 2D point features at places where the lines in the group have an intersection point that is stable with respect to error. The recognition system then considers all pairs of salient convex groups found. The 2D points in a pair of salient groups are used to index into a lookup table, where we have represented comparable groups of 3D model points, at compile time. This indexing phase, which matches groups of 2D image points to geometrically consistent 3D model points is also described in [27]. We then explore the matches hypothesized by indexing, beginning with those generated by pairs of image groups that matched the fewest pairs of model groups. A verification step uses matches between 2D points and 3D points to determine a hypothetical location of a model in the image. It then searches for additional evidence of the model's presence at the location, to confirm or reject a hypothesis.

We first show the most salient groups found in several images, along with their associated point features. Fig. 5 shows the most salient groups found by the grouping system in an image of an isolated telephone. Many of the groups found here show up reliably in other pictures of the phone, taken from different viewpoints.

In Fig. 6, we see some groups found in a scene that includes the telephone. Almost all the telephone's convex groups are at least partially occluded in this picture. However, we find unoccluded portions of these groups, many useful groups from the stapler, and some of the salient structure of the mugs. Fig. 7 shows the groups found in another scene containing the occluded telephone.

Fig. 8 shows the results on a different scene, which was taken at the CMU calibrated image lab. Although the edges are noisy and hard even for a person to interpret, we can see that the system finds much of the rectangular structure inherent in the buildings in the scene.

In each of the pictures shown, many of the most salient groups come entirely from the convex structure of a single object, making them useful for recognition. We also see many remaining challenges to grouping, because many of the groups found either do not appear perceptually salient, or appear to either combine lines from two different objects, or to combine strong lines from one object with noisy or unstable lines. For example, in the fourth set of groups shown in Fig. 6, we can see that a strong group from the face of the telephone includes an external edge from the rim of an occluding coffee mug.

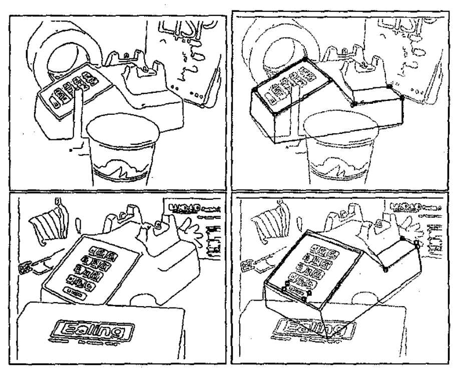

Fig. 9. This figure shows the results of a recognition system that uses salient convex grouping. On the left are edges found in two different scenes. On the right we show the correct hypothesis of the telephone's location in each scene, which is found by the system. Lines, which indicate the hypothetical location of model lines, are shown superimposed over a dotted edge map of the image. Circles indicate the location of image points that were used for indexing. Squares show the hypothesized location of the corresponding model points.

These groups, however, do enable our system to efficiently recognize objects. Although [28] details the performance of the system, Fig. 9 shows two samples of our results. Pairs of groups of points from the telephone have

1. This was a state-of-the-art computer in the MIT AI lab circa 1990.

been represented in the indexing table ahead of time. The figure shows that the system correctly locates the telephone. Circles indicate the location of image points used for indexing, which come from two salient convex groups. In each example, one group is from the side of the telephone. In the example on top, the other group comes from the rectangular face of the phone; in the example on the bottom, the second group comes from the small plate containing the phone number. In these examples, grouping produces a relatively small set of convex groups, some of which lead to correct recognition of the object sought. See [28] for further details on the overall efficiency of the recognition system, using grouping.

6 Comparison with Other Grouping Methods

In this section we will compare the effectiveness of salient convexity as a grouping clue with two other grouping methods. First, we consider Lowe's approach to grouping parallel lines. Lowe [39] measures the salience of parallel lines by considering the amount of separation between them, and the extent to which their angle deviates from perfect parallelism. Lowe suggests considering those pairs of lines first whose degree of separation and parallelism is least likely to occur at random. Following Lowe's derivation we will determine the expected number of salient parallel lines that will appear in an image, for varying degrees of salience.

Let θ be the angle between two lines, let l_1 and l_2 be the lengths of the shortest and longest lines respectively, and let s be the perpindicular distance from the midpoint of the shorter line onto the longer line. Lowe uses the expression:

$$E = \frac{4\theta l_2}{\pi l_1 h'}$$

as a measure of the inverse salience of a pair of lines, where we let $h' = \frac{l_1}{s}$. This is analogous to the definition of k' in our salient convexity grouping. Lowe makes the additional requirement that the perpindicular projection of the midpoint of the shorter line onto the longer lie inside the longer line.

Suppose we wish to find a pair of lines of equal length with an angle of θ_0 and a separation between them of s_0. We let $h'_0 = \frac{l_{1,0}}{s_0}$, i.e., h'_0 denotes the ratio of gap to line length. Let E_0 denote the inverse salience of these lines. We ask: how many pairs of lines will have a salience greater than $\frac{1}{E_0}$ in an image containing n randomly positioned lines? We assume random positioning with the same distribution as in our previous analysis, except that we assume that the midpoints of lines have uniformly distributed positions in the image, rather than the endpoints of lines.

A pair of random lines described by θ, l_1 and l_2 will have a greater salience than the lines we seek when:

$$\frac{4\theta s l_2}{\pi l_1^2} \le \frac{4\theta_0}{h'_0 \pi} \Rightarrow s \le \frac{\theta_0 l_1^2}{h'_0 \theta l_2}$$

That is, the midpoint of the shorter line must fall inside a rectangle of area:

$$2 s l_2 = \frac{2\theta_0 l_1^2}{h'_0 \theta}$$

If this area is greater than the image, we assume that the probability of inverse salience less than E_0 is 1, although this will not happen often. Therefore, given θ, l_1, l_2, the probability of inverse salience less than E_0 is:

$$\min\left(1, \frac{2\theta_0 l_1^2}{h'_0 \theta \pi R^2}\right)$$

where the image is assumed to be a circle of size πR^2. In fact, since all lengths will be taken relative to the circle size, we may from now on assume, without loss of generality that $R = 1$.

Now we may integrate over all possible combinations of l_1, l_2, and θ to determine the probability that two lines will have inverse salience less than E_0. This is:

$$\int_0^1 \int_{l_1}^1 \int_0^{\frac{\pi}{2}} \min\left(1, \frac{2\theta_0 l_1^2}{h'_0 \theta \pi}\right) \frac{4}{\pi} \, d\theta \, dl_2 \, dl_1$$

$$= \frac{8\theta_0}{h'_0 \pi^2}\left(\frac{1}{12}\ln\left(\frac{\pi}{2}\right) - \frac{1}{12}\ln\left(\frac{2\theta_0}{h'_0 \pi}\right) + \frac{13}{72}\right)$$

This derivation assumes that:

$$\frac{2\theta_0}{h'_0 \pi} \le \frac{\pi}{2}$$

which will be the case for all situations of interest in what follows.

Using this result, we can determine the expected number of false positive groups produced by Lowe's system in situations similar to those to which we have applied convex grouping. As an example, Table 8 shows the expected number of randomly generated groups with a greater salience than a designated pair of lines. This can be compared with Table 4. The designated lines are assumed to have an angle difference of $\frac{\pi}{20}$, and a separation that varies, as shown in the table. The table shows the separation in terms of a variable, $h = \frac{l_1}{l_1 + s}$, which is analogous to the variable k used in the convex grouping system.

It is easy to see that Lowe's method of grouping has asymptotically better performance than the convex grouping method. But it is also clear that when we choose thresholds so that the systems produce reasonably small sets of salient groups, Lowe's system must require lines to have only small gaps between them. In some realistic circumstances it appears that our system will tolerate larger gaps. Moreover, the gaps allowed in convex grouping are considered relative to the size of the entire group, while the gaps between parallel lines are measured relative only to the shortest line. Overall, both convexity and parallelism provide probabilistic information for grouping of roughly similar quality.

We now consider the performance of a greedy algorithm for salient convex groups similar to the one proposed by Huttenlocher and Wayner [24], or the one used as a preprocessing step in Jacobs [26]. We suppose that each oriented line initializes a group, and that the group is extended by adding the

TABLE 8
THIS TABLE SHOWS THE EXPECTED NUMBER OF SALIENT PARALLEL LINES THAT WE EXPECT LOWE'S ALGORITHM TO PRODUCE, AS THE NUMBER OF LINES AND THE ALLOWED GAP BETWEEN THE LINES VARY. WE ASSUME THAT LINES MUST HAVE A SALIENCE GREATER THAN LINES OF EQUAL LENGTH HAVING AN ANGLE OF $\frac{\pi}{20}$ AND A GAP OF s, SUCH THAT $h = \frac{l_1}{l_1+s}$

Expected number of parallel line pairs produced									
Number	h								
of lines	.25	.5	.6	.65	.7	.75	.8	.85	.9
200	2400	1000	750	630	520	420	330	250	170
300	5500	2300	1700	1400	1200	950	750	560	380
400	9700	4200	3000	2500	2100	1700	1300	990	670
500	15000	6500	4700	3900	3300	2700	2100	1600	1000
700	30000	13000	9200	7700	6400	5200	4100	3000	2100
1000	61000	26000	19000	16000	13000	11000	8400	6200	4200

nearest line to the end of the group that does not introduce a concavity in the connection between the two lines. That is, this approach does not check that the first and last points in the group can be convexly connected, allowing for spiral groups (this is allowed in [24], for example). Huttenlocher and Wayner [24] and Jacobs [26] perform variations on this method, with different specific methods of adding lines, or deciding when to stop extending a group. Huttenlocher and Wayner [24] also consider other criteria for deciding which line is best added to a group. However, the above general description of the greedy algorithm will suffice for our analysis, and provides a reasonable example of a greedy convex grouping algorithm.

False positives in cluttered environments are easily predicted for the greedy algorithm, since it is guaranteed to produce an $O(N)$ sized output. The more relevant issue is false negatives. We consider how the probability of finding a convex structure in an image will decrease as the image grows more cluttered with random lines.

Suppose we wish to locate a convex group that contains two gaps, each of at least length g_0. That is, we suppose, using our previous notation, that $\|l_{i,2}l_{i+1,1}\|$, $\|l_{j,2}l_{j+1,1}\| \geq g_0$ for some i, j. A greedy algorithm will fail to find this group if there exist a line convex with l_i whose first endpoint is a distance less than g_0 from $l_{i,2}$, and if a similar line exists for l_j.

Denote by P_b the probability that a random uniformly distributed oriented line will block a group containing l_i from extending to contain l_{i+1}. It is easily seen that:

$$p_b = \frac{1}{4}\frac{g_0^2}{R^2}$$

Assuming $2n$ oriented lines, the probability that at least one will be convex with l_i and closer to it than l_{i+1} is:

$$1 - (1 - P_b)^{2n}$$

The probability that the greedy algorithm will not bridge either of the gaps in the group, and therefore fail to collect all the group's convex lines together will be approximately:

$$(1 - (1 - P_b)^{2n})^2$$

In Table 9, we show how this probability varies with the gap size, g_0, and the number of lines, n. We can see that even with small gaps, there is a significant likelihood of missing a salient convex group with a greedy algorithm.

The greedy method is much like our backtracking search, with the search tree for salient convex groups restricted to have a branching factor of 1. Clearly this will produce greater efficiency at the cost of missing some salient convex groups. Again the appropriateness of each method may depend on its intended use. If we only expect small gaps between lines, or if we can tolerate missing some salient, convex groups, a greedy algorithm is to be preferred. Our method does have several advantages, however. The output is predictable in terms of the gaps that will be present in a group; one cannot predict with certainty whether the greedy algorithm will find a group on the basis of the group's structure alone; performance also depends on the location in the image of other, unrelated lines. Consequently, false negatives do not occur in our algorithm, given a particular tolerance. This may be important when one is seeking a few significant groups in an image. Also, our grouping method requires that a group be completely convex when the beginning and end are connected, and enforces closure. It is also possible to use hybrid methods between the two. For example, for any fixed m, the m nearest points to a line may be found in $O(n \log(n))$ time. Therefore, a backtracking search that only considered the m nearest lines could be performed with $O(n \log(n))$ preprocessing. In situations where preprocessing dominates, this could make a backtracking search very efficient, while providing greater robustness than a simple greedy algorithm.

TABLE 9
WE CONSIDER A CONVEX GROUP WITH TWO GAPS, WHOSE LENGTHS VARY IN THE COLUMNS OF THE TABLE.
THE GROUP IS IN AN IMAGE OF RADIUS 250 PIXELS.
THE TABLE SHOWS THE PROBABILITY THAT A SIMPLE GREEDY ALGORITHM WILL MISS SUCH A CONVEX GROUP

Probability of Missing a Salient, Convex Group								
Number	Gap (pixels)							
of lines	5	10	15	20	25	30	40	50
200	1.0	.98	.91	.78	0.6	.42	.15	.04
300	1.0	.95	.83	.62	0.4	.22	.04	.00
400	.99	.92	.74	.48	.25	.11	.01	.00
500	.99	.89	.65	.36	.16	.05	.00	.00
700	.98	.82	.49	0.2	.06	.01	.00	.00
1000	.97	0.7	0.3	.08	.01	.00	.00	0.0

7 CONCLUSIONS

Convexity is just one potential grouping clue, but it is an important one to understand thoroughly. Objects often contain at least some convex parts, especially in two important application areas, recognition of buildings and of manufactured objects. Something similar to the salient convexity measure that we consider is an essential part of the grouping methods used by Jacobs [25], Wayner [56], and Sawhney [52]. And related grouping methods based on convexity have been widely used in various approaches to recognition. But convexity has usually been handled in ad-hoc ways that are sensitive to local perturbations of the image.

In this paper we use a simple, global measure of a convex group's salience. A global definition of our output has the strong advantage of allowing us to anticipate our output, independent of unrelated context. We show here that much of the global constraint provided by salient convexity can be converted into a local form in which it can be applied at each step of the search, and that this allows us to build an efficient system.

Three key questions arise if we wish to understand the performance of salient convexity. First, can we compute it efficiently? Second, how likely is it that salient convex groups reflect the underlying structure of the scene, as opposed to simply resulting from a random process? Third, how robust can salient convex groups be to possible occlusions or clutter? We have performed a theoretical analysis, backed up by experiments, that allow us to answer these questions.

We show that in realistic situations, we can efficiently find the most salient convex groups of an image. Our algorithm takes $O(n^2 \log(n) + mn)$ expected time to find the m most salient groups in an image with n line segments. This theoretical result is also practical. In real images, our system runs sufficiently fast to be useful as an experimental system. The system runs quickly precisely because salient convex groups do not often arise by chance. This allows us to effectively prune our search for groups. It also allows us to show that the groups that we do find are likely to reflect some non-random scene structure.

Finally, our analysis shows how to choose a measure of salient convexity appropriate to the size of the image, and of the desired output. For realistic sized images, we show that a salience threshold ranging from about .70 to .85 will produce a fast algorithm, with a manageable output size. This tells us that our grouping system can find convex groups that have between 30% and 15% of their boundary missing due to occlusion or noise. The presence of clutter also affects the robustness of the algorithm; in an uncluttered scene with fewer line segments we could use a lower salience threshold, as we add clutter, our output will grow unless we raise this threshold. Since our output is characterized by a simple salience criteria, we can predict precisely how robust our algorithm will be.

Finally, we have demonstrated the potential value of salient convex grouping by incorporating our system into a complete recognition system. Grouping reduces the combinatorics of recognition by focusing the search for an object on subsets of the image that are likely to come from a single object. It is not necessary that every group of lines that we find in the image actually comes from the object for which we are looking. It is sufficient if we can locate enough image groups to allow us to recognize an object without having to examine too many irrelevant image groups, that is, our groups need to provide points that are more likely to come from the object for which we are looking than are randomly selected groups. The greater this likelihood is, the greater is the advantage provided by grouping. The grouping system presented here supports a very focused recognition system that requires little search.

Acknowledgments

This paper describes research done, in part, at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology, and was funded in part by an Office of Naval Research University Research Inititative grant under contract N00014-86-K-0685, and in part by the Advanced Research Projects Agency of the Department of Defense under contract N00014-91-J-4038, Army contract number DACA76-85-C-0010, and under Office of Naval Research contract N00014-85-K-0124. The author has received further support from the NEC Research Institute.

The author received valuable comments and suggestions from Ingemar Cox, Eric Grimson, Lance Williams, and anonymous referees. More preliminary versions of this work have appeared in [29], [30].

References

[1] T.D. Alter, "The role of saliency and error propagation in visual object recognition," PhD thesis, MIT, 1995.

[2] R.Basri and D. Jacobs, "Recognition using region correspondences," *Proc. IEEE Int'l Conf. on Computer Vision*, pp. 8-15, 1995.

[3] R. Bergevin and M. Levine, "Generic object recognition: Building and matching coarse descriptions from line drawings," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 1, pp. 19-36, Jan. 1993.

[4] I. Biederman, "Human image understanding: Recent research and a theory," *Computer Graphics, Vision, and Image Processing*, vol. 32, pp. 29-73, 1985.

[5] T. Binford, "Inferring surfaces from images," *Artificial Intelligence*, vol. 17, pp. 205-244, 1981.

[6] M. Boldt, R. Weiss, and E. Riseman, "Token-based extraction of straight lines," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 19, no. 6, pp. 1,581-1,594, 1989.

[7] J.B. Burns and E. Riseman, "Matching complex images to multiple 3D objects using view description networks," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 328–334, 1992.

[8] A. Califano and R. Mohan, "Multidimensional indexing for recognizing visual shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 4, pp. 373-392, Apr. 1994.

[9] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679-698, 1986.

[10] D. Clemens, "Region-based feature interpretation for recognizing 3D models in 2D images," Technical Report AI TR–1307, MIT, 1991.

[11] D. Clemens and D. Jacobs, "Space and time bounds on model indexing," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 10, pp. 1,007-1,018, Oct. 1991.

[12] I. Cox, J. Rehg, and S. Hingorani, "A Bayesian multiple-hypothesis approach to edge grouping contour segmentation," *Int'l J. Computer Vision*, vol. 11, no. 1, pp. 5–25, 1992.

[13] S. Denasi, G. Quaglia, and D. Rinaudi, "The use of perceptual organization in the prediction of geometric structures," *Pattern Recognition Letters*, vol. 13, no. 7, pp. 529-539, 1991.

[14] J. Dolan and E. Riseman, "Computing curvilinear structure by token-based grouping," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 264–270, 1992.

[15] J. Elder and S. Zucker, "A measure of closure," *Vision Research*, vol. 34, no. 24pp. 3,361-3,370.

[16] L. Finkel and P. Sajda, "Object discrimination based on depth-from-occlusion," *Neural Computation*, vol. 4, no. 6, pp. 901-921, 1992.

[17] W.E.L. Grimson, *Object Recognition by Computer: The role of geometric constraints*. Cambridge, Mass.: MIT Press, 1990.

[18] G. Guy and G. Medioni, "Inferring global perceptual contours from local features," *DARPA Image Understanding Workshop*, pp. 881–892, 1993.
[19] R. Heitger and R. Von der Heydt, "A computational model of neural contour processing, figure-ground segregation and illusory contours," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 32–40, 1993.
[20] L. Hérault and R. Houraud, "Figure-ground discrimination: A combinatorial optimization approach," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 899-914, Sept. 1993.
[21] D. Hoffman and W. Richards, "Parts of recognition," *Visual Cognition*, S. Pinker, ed. Cambridge, Mass.: MIT Press, 1984.
[22] J. Hu, B. Sakoda, and T. Pavlidis, "Interactive road finding for aerial images," *Proc. IEEE Workshop Applications of Computer Vision*, pp. 56–63, 1992.
[23] T. Huang, "Reply: Computer vision needs more experiments and applications," *Computer Vision, Graphics, and Image Processing*, vol. 53, no. 1, pp. 125-126, 1991.
[24] D. Huttenlocher and P. Wayner, "Finding convex edge groupings in an image," *Int'l J. Computer Vision*, vol. 8, no. 1, pp. 7-29, 1992.
[25] D. Jacobs, "GROPER: A grouping based object recognition system for two-dimensional objects," *Proc. IEEE Workshop Computer Vision*, pp. 164-169, 1987.
[26] D. Jacobs, "Grouping for recognition," MIT AI Memo 1177, 1989.
[27] D. Jacobs, "Space efficient 3D model indexing," *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 439–444, 1992.
[28] D. Jacobs, "Recognizing 3D objects using 2D images," MIT AI TR–1416, 1992.
[29] D. Jacobs, "Robust and efficient detection of convex groups," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. pp. 770–771, 1993.
[30] D. Jacobs, "Finding salient convex groups," *Proc. DIMACS Workshop Partitioning Data Sets: with Applications to Psychology, Vision, and Target Tracking*, 1995.
[31] A. Kalvin, E. Schonberg, J. Schwartz, and M. Sharir, "Two-dimensional, model-based, boundary matching using footprints." *Int'l J. Robotics Research*, vol. 5, no. 4, pp. 38-55, 1986.
[32] T. Kanade, "Recovery of the three-dimensional shape of an object from a single view," *Artificial Intelligence*, vol. 17, pp. 409-460, 1981.
[33] G. Kanisza and W. Gerbino, "Convexity and symmetry in figure-ground organization," *Vision and Artifact*, M. Henle, ed. New York: Springer, 1976.
[34] P. Kellman and T. Shipley, "A theory of visual interpolation in object perception," *Cognitive Psychology*, vol. 23, pp. 141-221, 1991.
[35] W. Kohler, *Gestalt Psychology*. New York: Mentor Books, 1959.
[36] I. Kovacs and B. Julesz, "A closed curve is much more than an incomplete one: Effect of closure in figure-ground segmentation." *Proc. Nat'l Academy of Science, USA*, vol. 90, pp. 7,495-7,497, 1993.
[37] M. Kubovy and J. Pomerantz, *Perceptual Organization*. Hillsdale N.J.: Lawrence Erlbaum Associates, 1981.
[38] D. Lowe, *Perceptual Organization and Visual Recognition*. The Netherlands: Kluwer Academic Publishers, 1985.
[39] D. Lowe, "Three-dimensional object recognition from single two-dimensional images," *Artificial Intelligence*, vol. 31, pp. 355-395, 1987.
[40] J. Mahoney, "Image chunking: Defining spatial building blocks for scene analysis," MIT AI TR–980, 1987.
[41] D. Marr, "Artificial intelligence—A personal view," *Artificial Intelligence*, vol. 9, pp. 37-48, 1977.
[42] R. Mohan and R. Nevatia, "Using perceptual organization to extract 3D structures," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 11, pp. 1,121-1,139, Nov. 1989.
[43] R. Mohan and R. Nevatia, "Perceptual organization for scene segmentation and description," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 6, pp. 616-635, June 1992.
[44] M. Nitzburg and D. Mumford, "The 2.1-D sketch," *Proc. IEEE ICCV*, pp. 138-144, 1990.
[45] P. Parent and S. Zucker, "Trace inference, curvature consistency and curve detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 8, pp. 823-839, Aug. 1989.
[46] T. Pavlidis and S. Horowitz, "Segmentation of plane curves," *IEEE Trans. Computers*, vol. 23, pp. 860-870, 1974.
[47] A. Pentland, "Recognition by parts." *Proc. First Int'l Conf. Computer Vision*, pp. 612-620, 1987.
[48] G. Reynolds and J.R. Beveridge, *Proc. DARPA Image Understanding Workshop*, pp. 257–271, 1987.
[49] W. Richards and A. Jepson, "What makes a good feature?" MIT AI Memo 1356, 1992.
[50] S. Sarkar and K. Boyer, "Integration, inference, and management of spatial information using Bayesian networks: Perceptual organization," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 3, pp. 256-274, Mar. 1993.
[51] E. Saund, "Identifying salient circular arcs on curves," *Computer Vision, Graphics and Image Processing: Image Understanding*, vol. 58, no. 3, pp. 327–337, 1993.
[52] H. Sawhney, "Trackability as a cue for potential obstacle identification and 3D description," *Int'l J. Computer Vision*, vol. 11, no. 3, pp. 237-266, 1993.
[53] A. Sha'ashua and S. Ullman, "Structural saliency: The detection of globally salient structures using a locally connected nNetwork," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 321-327, 1988.
[54] T. Syeda-Mahmood, "Data and model-driven selection using color regions," *Proc. Second European Conf. Computer Vision*, pp. 115-123, 1992.
[55] D. Trytten and M. Tuceryan, "Segmentation and grouping of object boundaries using energy minimization," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 730-731, 1991.
[56] P. Wayner, "Efficiently using invariant theory for model-based matching," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 473-478, 1991.
[57] L. Williams, "Perceptual organization of occluding contours," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 133-137, 1990.
[58] A. Witkin and J. Tenenbaum, "On the role of structure in vision," *Human and Machine Vision*, Beck, Hope, and Rosenfeld, eds. New York: Academic Press, 1983.
[59] M. Zerroug and R. Nevatia, "Using invariance and quasi-invariance for the segmentation and recovery of curved objects," *Applications of Invariance in Computer Vision*, J. Mundy and A. Zisserman, eds. Berlin, Heidelberg: Springer-Verlag, 1994.
[60] S. Zucker, "Cooperative grouping and early orientation selection," *Physical and Biological Processing of Images*, Braddick and Sleigh, eds. Berlin: Springer-Verlag, 1983.

David W. Jacobs received the BA degree in mathematics from Yale University in 1982, the MS degree in computer science from M.I.T. in 1988, and the PhD degree in computer science from M.I.T. in 1992. Since then he has been a Research Scientist at NEC Research Institute, where he has pursued interests in human and machine vision and in artificial intelligence.