

It is interesting to note that virtually all proposed algorithms use *local* operators to infer more global structures. Also note that many of the schemes are iterative, relying on one relaxation (or minimization) scheme or another, and are similar in that sense. The main differences are in the choice of the compatibility measures or the function to minimize.

Complexity comparison

It is hard to compare the complexity of the various algorithms since many of them are iterative in nature. The number of iterations required for a given algorithm was only stated empirically, and not as a function of the data, as is often the case.

Also, the different features chosen do not allow for a meaningful comparison on a standard scale.

2.4 Comparison and summary

The main features of some of the more important works are summarized in the following table:

Table 1: Comparison of different grouping techniques

	Lowe[42]	Ahuja & Tuceryan [2]	Dolan & Weiss[17]	Mohan & Nevatia [48]	Ullman & Sha'ashua [61]	Parent-Zucker [50]	Heitger & von der Heydt [26]	Our scheme
Operator	Local	Local	Local	Local	Extensible (local)	local	Global	Global
Primitives	Straight lines	Dots	Straights	curves	straight lines and curves	curves	end-points and T-junctions	dots, lines and curves
Control	One pass	Iterative	Iterative	Relaxation Progressive	Parallel-progressive (iterative)	relaxation	one pass	One-pass convolution
Noise immunity	Not clear	Good	Good	Good	Moderate	Good	Not handled	Very good
Scale	One	One	Hierarchy	One	One	One	One	One
Parameters	Yes	No	Yes	Yes	Yes	Yes	Yes	None
Pre-attentive (Domain free)	Yes	Yes	Yes	No (yes)	Yes	Yes	Yes	Yes
Special feature	First	Dot clustering, parameter free	Multi-resolution	Symmetry, high-level con.	Saliency map	local kernels		Saliency map, unified, parameter-free
Sensitive computations	Yes	No	Yes?	Yes	Yes	Yes	No	None

Kanizsa square since the input can be explained nicely by a bright square surface floating above four dark disks.

In a recent work [83], Williams *et al.* adopt an approach which is based on ours. They employ vector field convolutions with stochastic extension properties to assign saliency to image locations. Williams departs from the Gestalt constraints, and relies on a single constraint, namely, that the prior probability distribution of boundary completion shape can be modeled by a random walk in a lattice whose points are positions and orientations in the image plane. With the above assumption, Williams derives fields which are surprisingly similar to ours. All examples shown deal with perfect data.

2.3.9 Parent and Zucker

Parent and Zucker [50] describe a relaxation labeling where local kernels are used to estimate tangent and curvature. These kernels use support functions based on co-circularity. Somewhat similar kernels are used in our scheme, but applied in a very different way.

2.3.10 Heitger and von der Heydt

Heitger and von der Heydt [26] make use of anisotropic selective filters which are combined pair-wise to recover mainly *occluding* contours. The scheme takes as input endpoints and T-junctions, and results in the most natural connections of those. It is based on neurophysiological observations, but does not handle noise, and assumes that endpoints and T-junctions are available by some other means. The authors present convincing results on real images.

2.3.11 Williams

Williams [82] takes a more global look at completion of illusory contours. In his system, the input data is described as a set of occluding surfaces and the interactions between them. This is clearly a more global view of the problem, since the derivation of surface involves ‘looking’ everywhere in the input image. The mechanics of occlusion of one surface by another are described by a set of integer linear constraints. Out of the feasible solutions that this system produces, it is possible to select the one the best explains the image structure. Such approach works well for inputs similar to the

2.3.7 Parvin

In this work ([51]), a set of simultaneous differential equations are set to encode weak smoothness constraints. Every segment and vertex in the original image is assigned a token and a decay rule (a differential equation). The dynamic system is then solved by one of the standard methods (e.g. simulated annealing). Noise is handled in a consistent way. The input is in the form of range data, which removes some of the complexities inherent in 2-D images. Also, a consistent labeling scheme is incorporated into the system, thus allowing only valid objects to ‘survive’ the relaxation. Similar to Lowe’s work, the network ‘connections’ are between pairs of nodes only, and global structures due to co-curvilinearity cannot be revealed.

2.3.8 Huttenlocher and Wayner

In this work [31], a different property of natural scenes is exploited to find groupings, namely, *convexity*. Objects are usually composed of closed and convex areas, and the segmentation scheme generates a description in terms of these convex areas. The method is able to segment an edge image into an optimal set of convex groups of edges. It uses a more general form of the Voronoi tessellation for edges, thus creating a parameter-free neighborhood system. The complexity of finding the convex groups is now reasonable. The issue of noise is not addressed, and all lines from the original image are considered in the description. This makes the algorithm highly unstable with respect to noise.

logical saliency, and relies on constraints such as smoothness, length, and constancy of curvature. A typical input image is depicted in Figure 2.5 (similar to the images we

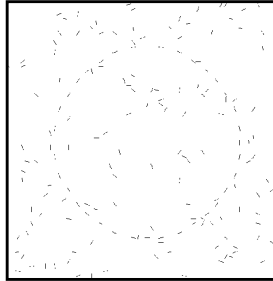


Figure 2.5 A typical input image (after [61]). The algorithm will assign high values of saliency along the fragmented circle.

handle). The scheme prefers long curves with low total curvature, and does that by using an incremental optimization scheme (similar to dynamic programming). This is done in order to reduce the exponential complexity involved in picking subsets from a large set of data. Because of this restriction, an extensible, rather than a global operator is defined. This operator is capable of *locally* choosing the best continuation from a given segment. Locality is defined in terms of the number of neighbors. As before, such a scheme cannot account for large gaps, and can be fooled by erroneous segments along a correct curve.

The process runs iteratively on a locally connected network, which sets the size and the number of the neighbors as the number of connections to each node of the net. The number of iterations equals the length of the minimal salient curve found, and is reported to be in the order of dozens.

interact to correct possible errors in a previous step. Many empirical parameters are used throughout the process, but they claim tolerance to them.

The representation extracted so far has to be treated now to eliminate competing or contradicting features. For example, two rectangles sharing an edge are considered competitive, while features which are part-of other features are considered supportive.

A constraint satisfaction network scheme using the above ‘rules’ plus measures of belief for each group is used to extract the ‘correct’ and most meaningful rectangles.

In a later work ([48]), Mohan and Nevatia present a more general scheme which makes use of symmetries in the image. Symmetries (parallel or mirror) between curved lines impose very strong constraints on groupings. This is used here to segment a scene into ribbons. All symmetry axes are extracted and are fed into a constraint satisfaction net to resolve competitions. The system works well when the curves are long and the amount of noise is small. Again, some thresholds are used to reject hypotheses, in order to reduce the complexity of the process.

2.3.6 Sha’ashua and Ullman

Sha’ashua and Ullman [61] suggest the use of a *saliency map* to guide the grouping process, and to select features in the image. A saliency map is a dense map having the size of the image, where the value of each point corresponds to the degree of importance that point plays in the image. ‘Importance’ is measured in terms of psycho-

Some geometrical properties are defined on *each* of the Voronoi polygons³, and relaxation labeling nets are constructed to reach a steady state. Several different nets are set up to relax various constraints. In some cases, interaction between the nets is enabled, to reach the final result of a consistent labeling of all tokens. Noise points are labeled 'Isolated' by this scheme.

The results shown in [2] seem superior when compared to previous attempts at clustering dot images[33]. The relaxation scheme offers a high degree of robustness, and allows for correction of results generated at an early step by a later step.

2.3.5 Mohan and Nevatia

Mohan and Nevatia's [47] original work aims at grouping straight edges in an image, but assumes *apriori knowledge* of the contents of the scene. A model of the desired features is defined, and the grouping is guided according to that model. This process is more robust than other general methods, since it basically searches for occurrences of the desired shape and not for just any 'acceptable' shape. In their later work [48], the goal is to detect rectangles (or combination of rectangles) which represent buildings as seen from aerial photographs.

The approach works in a bottom-up fashion, first grouping line fragments into lines, then grouping lines into parallels and eventually, grouping parallels into U-shaped groups and (wherever possible) complete rectangles. The different levels can

3. Like Compactness, Area, Elongation, Eccentricity, and Squeezedness.

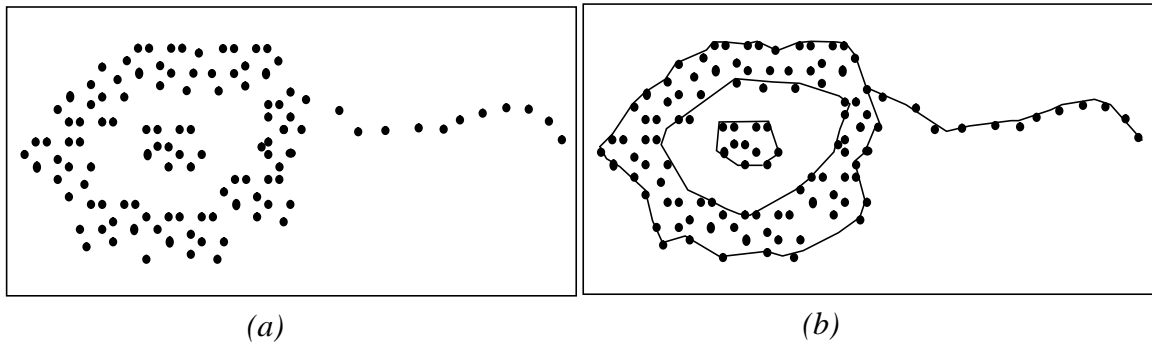


Figure 2.4 A typical input handled by Ahuja and Tuceryan [1] in (a), and the result of applying their method in (b).

Previous work in that direction (as summarized in [2]) attempted to find a partition of the space, such that an objective function is maximized. This can be achieved in several ways. Usually an initial labeling of the data is performed according to some local maximum likelihood criteria. Then, a relaxation scheme is used to iteratively satisfy the global constraints. Hierarchical methods start from a degenerate partition where each feature belongs to a different class, after which features are merged to create higher-level representations (bottom-up method).

The reverse approach (top-down) is also widely used. The actual objective function consists of a set of similarity measures (or a compatibility metric) of highly heuristic nature.

Ahuja and Tuceryan use the Voronoi tessellation and its dual, the Delaunay graph as the basic geometry on which all compatibility measures are defined. The Voronoi graph defines a scale independent (and parameter free) neighborhood system, where two points are considered neighbors if they share a polygon side.

2.3.3 Zucker

Zucker was the first to address the problem of dot clustering [85]. In his work, each dot in the image is labeled as being either an edge, interior, or noise dot. A relaxation process for labeling the functional roles that the dots appear to be playing in a given arrangement is presented. This labeling information can later be used to compute global shape descriptions. A more detailed description of the algorithm is presented in Ahuja and Tuceryan's work [2], which is described next and builds on top of Zucker's algorithm.

In a later work [86], Zucker *et al.* propose a new method of curve detection. A coarse tangent is inferred along curves, and a subsequent stage of spline fitting is then applied to this coarse tangent field. The algorithm is capable of producing spline representation of the desired curves.

2.3.4 Ahuja and Tuceryan

Ahuja and Tuceryan [2] suggest methods for clustering and grouping sets of *points* having an underlying perceptual pattern. A typical input is shown in Figure 2.4.

The authors attempt to label the points as either *Interior*, *Border*, *Curve*, or *Isolated*. Furthermore, the Delaunay edges² are also labeled as being either a *Border*, *Non Border*, or *Curve*. Other constraints incorporated explicitly into the algorithm are aimed at smoothing of borders and curves.

2. Derived as the dual from the Voronoi Tessellation (for a survey see [3]).

Computational Paradigm

The authors describe a three step iterative algorithm to extract the hierarchical representation. The cycle consists of:

- 1) Linking of line segments,
- 2) Selecting acceptable groups,
- 3) Replacing a group of several segments by one higher-level segment.

These three steps are repeated, starting at the finest level (of the initial tokens) and up to the coarsest representation of the image, which might consist of only a handful of tokens, being a rough approximation of the image edges.

An important notion is the *perceptual window*. This circular window determines the scope of grouping from one level of the hierarchy to the next. Only tokens within a window are participating in the choice of the higher-level token. This window can and does change its size as the level becomes coarser.

The constraints used for linking two segments together are very similar to what was used by Lowe, namely, proximity, angular compatibility, and continuation. Again the constraints are implemented locally only.

Other details from the work are not relevant to our research and are omitted here.

The (somewhat heuristic) significance measures for the three types of groupings do not seem to behave well in all situations. To best evaluate the suitability of the measures, one could ‘fix’ all but one parameter in a given measure, and test the behavior by changing the free parameter to its extreme bounds. We consider, for example, the colinearity relation. Suppose we fix all parameters but s . When $s = 0$ the measure equals zero regardless of all other parameters. This is clearly inaccurate, since the gap can now be huge, but the grouping would still be considered highly significant.

2.3.2 Dolan and Weiss

Dolan and Weiss [17] demonstrate a hierarchical approach to grouping. They emphasize the scale issue which makes a hierarchical approach very appealing. It allows gradual construction of the original image from coarse to fine, and offers views at different scales, which could be used for later processing.

The tokens used in the tree representation are straight lines and conics (in spline form). With these two tokens (in combination), it is possible to describe cusps, inflection points, corners, and of course straight lines.

The initial model for the input is in the form of unit length tangent segments extracted from a directional edge detector.

2.3.1.3 Collinearity Grouping

The same parameters as before are used here, and the most important parameter is the positive gap (g) between the two segments. In other words, we consider only pairs of segments which do not overlap in the direction of the assumed collinearity. Here only the shorter of the two lines is used in the measure: $E = (\Theta s (g + l_1)) / (\pi l_1^2)$. The smaller E is, the better the fit. As before the rationale

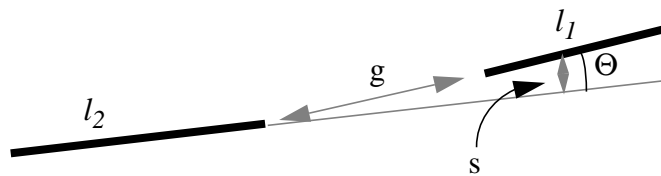


Figure 2.3 Parameters used in computing the collinearity of two segments.

is to put parameters that improve the overall likelihood of the grouping in the denominator and vice versa.

At this point, Lowe suggests an exhaustive search to extract all possible groupings in a given image. He also mentions some possible shortcuts. In his paper [43], he proceeds by performing model-based object recognition, one of the first attempts at incorporating perceptual grouping to that field.

The above grouping method computes a compatibility figure between any two segments in the image. As such, it cannot reveal global structures which are a composition of many segments, and obviously cannot support curved lines.

2.3.1.2 Parallelism Grouping

By the same method, a measure of parallelism is defined between two lines in the image. Several parameters are considered here. First, the angle Θ between the two

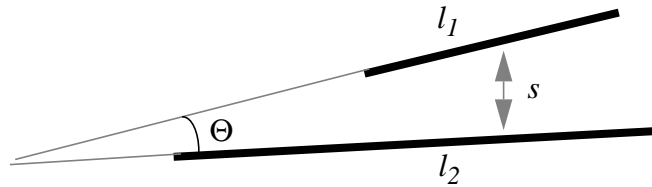


Figure 2.2 Parameters used in computing the parallelism of two segments.

lines is obviously the most important parameter. Then, the perpendicular distance¹ s is important, since the farther apart the lines are, the less likely it is that the lines form a group. Lastly, the actual lengths of the segments (l_1 and l_2) must play a role in the measure (we would want them to be roughly the same length). Lowe suggests the following measure, $E = (\Theta s l_2) / (\pi l_1^2)$, ($l_2 \geq l_1$). When E is close to zero, we have a strong parallelism relation and vice-versa. When lines are exactly parallel (which is rare in real live images), the angle Θ equates to zero, and $E=0$, meaning the highest significance possible. The logic behind the construction of the above equation is the following: parameters which intuitively increase the likelihood of the desired relationship as they grow are put in the denominator, while parameters which decrease it are put in the numerator.

1. measured from the center of the shorter line perpendicular to the longer line.

2.3.1.1 Proximity Grouping

Lowe [43] suggests a metric to quantify the *significance of proximity* of two endpoints of line segments. Significance is a loose term for probability, where no attempt is done to normalize the entire sample space so that it integrates to 1. As such, it only provides a *relative* measure of importance. The assumption is that no a priori knowledge of the scene is available. Thus, the 2-D model assumed is of line segments uniformly distributed with respect to orientation, position, and scale.

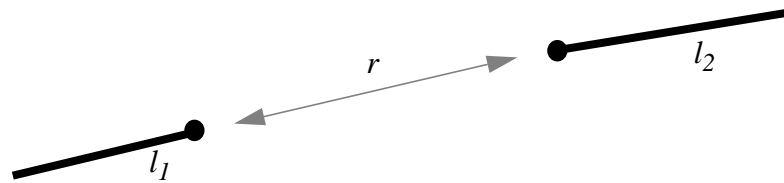


Figure 2.1 Parameters used in computing the proximity of two end-points.

With these assumptions, the (un-normalized) ‘probability’ N , of proximity being accidental is given by $(N = d\pi r^2) / \min(l_1, l_2)^2$, where r is the distance between endpoints and d is the density of endpoints per unit area. Clearly, the significance is inversely proportional to the square of the distance. Since d is *not* independent of the length of the line segments, further normalization is possible. Lowe suggests dividing the above by the square of the length of the shorter of the two line segments involved. This makes the measure invariant to scaling and also favors longer segments with greater gaps.

2.3 Edge-based organization

The largest body of research in perceptual organization deals with images represented by their edges. It has been shown (see Barrow and Tenenbaum [5]) that in many cases, edges convey most of the required information and other cues (such as shading, color etc.) are secondary. Moreover, we tend to prefer edge information over other cues when such cues are conflicting (see [5]). Also, line drawing of complex objects are recognized as quickly as full color images

2.3.1 Lowe

Lowe [43] discusses the Gestalt notions of co-linearity, co-curvilinearity and simplicity as important in perceptual grouping and, in fact, was the first to introduce them to the computer vision community. His claim is that these properties are invariant to the point of view, and are unlikely to appear by accident.

Assuming a projective model of the camera, colinear lines in the real world would project to co-linear lines in the image. Proximity has the same property with the complicating exception that far away points in real life may project to close by points with any arbitrary probability.

Since there is no hope of finding exact co-linear lines or perfect parallels, Lowe employs various functions to estimate the *degree of significance* of a desired relation.

We will now describe in some detail the basic ideas and assumptions suggested by Lowe and used by many other researchers with minor changes.

symmetry. They make use of local grey-level differences to pair up locations in the image that share a symmetry relation. The claim is that such locations have special meaning to humans, and as such can be used as a focus-of-attention areas. Computing this symmetry operator on images with faces results in the detection of points near the eyes and mouth, which are highly symmetrical. It is not clear from their paper whether the technique works when the faces are not straight ahead.

Zielke [84] has demonstrated a fast algorithm to detect cars in traffic through their inherent symmetry, again using grey-level information. The general approach is here is to hypothesize groupings between similar derivative along horizontal scan lines. The assumption is that the typical car has a vertical symmetry axis. A subsequent step attempts to detect straight vertical symmetry axis, and derive the extent of the car from that.

Ahuja [1] has devised a multi-scale skeleton and edge contours extraction system that works directly on grey-level images, and provides some degree of gap filling. The emphasis, though, is on properties of the approach rather than on experiments on real or simulated data. Here, a distance transform is computed, and properties of the resulting skeleton are evaluated and back-projected to bridge gaps. The examples are on very simple two (or three) shade inputs, making it unclear as to how the technique is applied to real grey-scale images.

Manjunath and Chellappa (in [45]), and Malik and Perona (in [44]) use a basis set of even and odd directional wavelet masks (with different orientations and scales). This set is convolved with the image to provide a representation of the image which is orientation selective and has optimal localization properties. An inhibition net is then set to converge to the desired features followed by grouping of peaks in the gradient response.

Perry and Lowe [54] use a similar decomposition scheme (wavelets) but proceed with a region-growing strategy to segment the image. Bovik *et al.* [12] and Farrokhnia and Jain [20] suggest a multichannel (hierarchical) approach where the Gabor functions are tuned to a set of selected bands, corresponding to the hierarchy.

Reed and Wechsler [56] present a comparison between several commonly used representations and demonstrate the feasibility of such schemes to pre-attentive vision.

Recently, Manjunath and Ma [46] have proposed a scheme that makes use of Gabor functions in content-based retrieval of images from large imagery databases. They report high success rates.

2.2 Grey-level organization

Some interesting work has been done on grey-level images, almost without any stage of preprocessing. This section is brought here since part of our work can be applied to grey-level images. Reisfeld *et al.* [57] suggest a symmetry operator that responds to positions in the image that are on a symmetry axis, especially for circular

Texture segmentation attempts to find boundaries between neighboring texture patches, where physical edges do not exist. Texture classification, on the other hand, is not only interested in locating the boundaries, but also to characterize the texture in some way (statistical or structural), to facilitate classification.

Most work in segmentation and classification of textures uses *local* properties measurements, which are later used in one of the classical segmentation methods, devised for grey-levels or color.

Among the most popular properties measured are the response to directional masks, especially Gabor functions which have the property of being optimally localized both in the frequency and position domains. This allows for accurate segmentation in *image* domain, based on measurements in *frequency* domain.

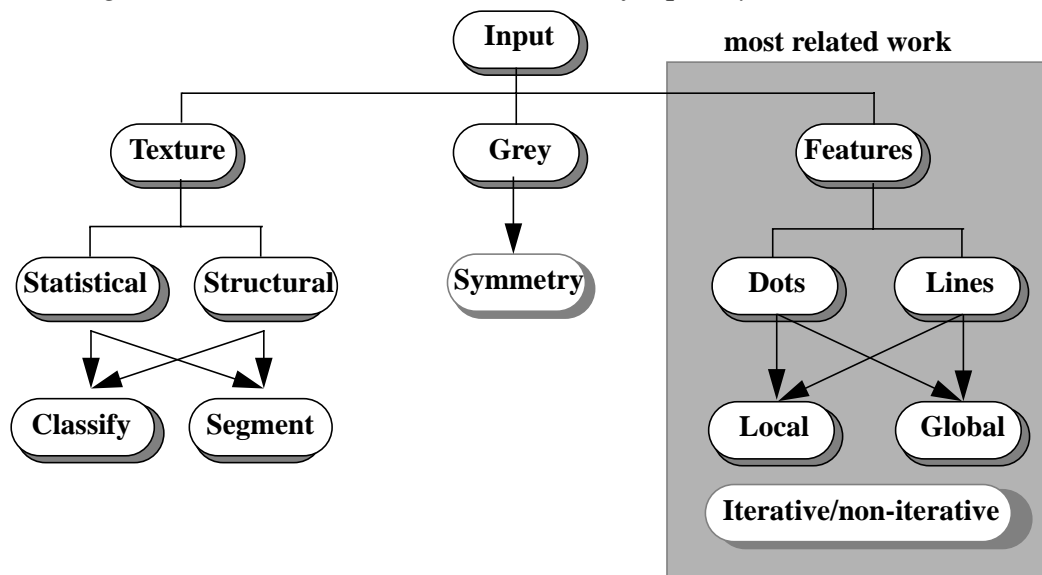


Chart 2.1 A classification of work in Perceptual Grouping

Chapter 2

Survey of Related Research (2-D)

Work in the field of perceptual organization is fairly limited in the computer vision community. Numerous researchers have used ad hoc methods somewhat similar to what Lowe [43] proposed (See “Lowe” on page 14), in order to build systems in computer vision. Such efforts do not fall into any of the categories we are about to describe, and will not be further discussed.

Attempting to classify work done in perceptual grouping can be done along many different axes. Among the most natural are: type of input, nature of the operator used, mechanism of the algorithm, and model of the output. This survey of related work is arranged as a ‘tree’ by the order described above (see Chart 2.1). Following the above categorization precisely is not possible, since the boundaries between the categories are not sharp, so some diversions will occur.

2.1 Texture segmentation and Classification

We do not attempt to handle inputs in the form of statistical textures, even though they can be classified correctly as being a perceptual grouping phenomena. A brief discussion in this subsection is provided for the sake of completeness.

to find the ‘correct’ scale of perception, but rather offer the perception at all different scales (See “Multiple Resolution” on page 63).

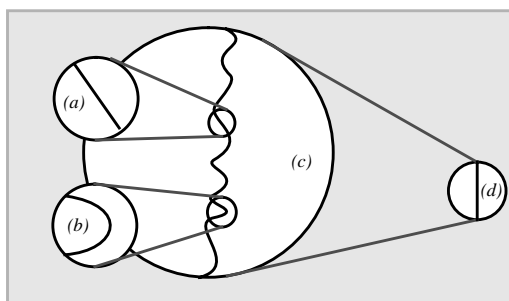


Figure 1.6 *Different perceptions at different scales. (after [17]). The crooked line in (c) is seen as a **straight** vertical line (d) at a coarser scale, and as either a **straight tilted** line (a) or a **curved** segment in (b).*

ferent from scale to scale, and in general it is hard to determine what the correct scale is, or even whether one scale is suitable for a given image. Nevertheless, in many cases a single scale is assumed by humans, based on their expectations from the scene. Other clues, such as a familiar object or a familiar context, also help reach a decision regarding the correct scale. These observations are supported by Wertheimer [81] and other Gestalt researchers. Subriana-Vilanova and Richards [67] also argue that although the pre-attentive process is essentially a bottom-up one, in some difficult situations, recognition (or even lower-level segmentation) is not possible without the high-level help (or hint), which is in a sense a top-down process. (The dalmation dog is a good example of this top-level hint).

Another important parameter related to the scaling issue is the viewing distance. Clearly, perceptual grouping is not just an objective phenomenon inherent to the image itself. It is a combination of the image and (among other things) the viewing distance. We discuss some of these issues in the following chapters. However, we do not attempt

and good continuation as depicted in Figure 1.4. To see how these laws can conflict, consider, for example, Figure 1.5. Here, groupings due to proximity conflict with

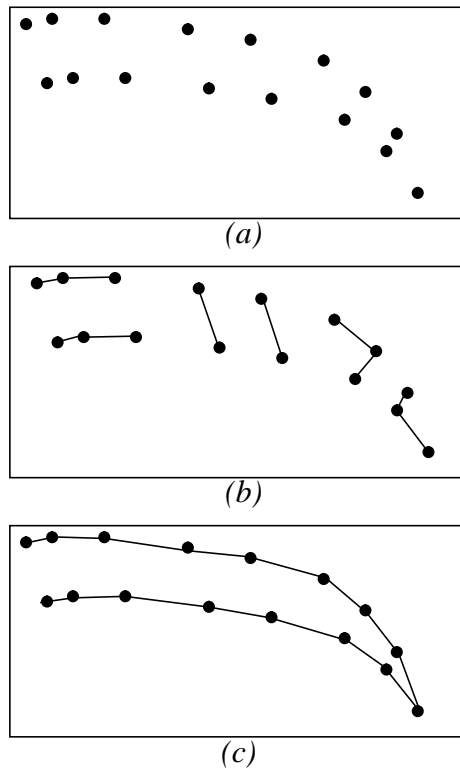


figure 1.5 Conflict between proximity grouping and good continuation. (a) a set of input points. (b) The results of connecting each point to its nearest neighbor. (c) the correct connections can only be inferred based on a more global view of the scene.

groupings based on co-curvilinearity. Towards the right end of the formation, applying the proximity rule yields one grouping (wrong in this case), while applying the good continuation yields another (correct).

1.3 Scale Dependency

Both attentive and pre-attentive perceptions occur at many scales. Figure 1.6 shows a curve when viewed at different scales [17]. The perception is obviously dif-

tours is found in the Kanizsa illusion [34] shown in Figure 1.3(c). Here we perceive edges which have no physical support whatsoever in the original signal. Figure 1.3(d) depicts a dot formation. Again, grouping of certain dots is possible, and salient curves are noticeable.

Among the first to address the issues of pre-attentive perception were the Gestalt psychologists (e.g. [81,7]). Many ‘laws of grouping’ were formulated (One hundred and fourteen, to be exact [27]), but none put in any computational (or algorithmic) language. Furthermore, the rules tend to supply conflicting explanations to many stimuli. This makes the computational implementation of such laws non-trivial.

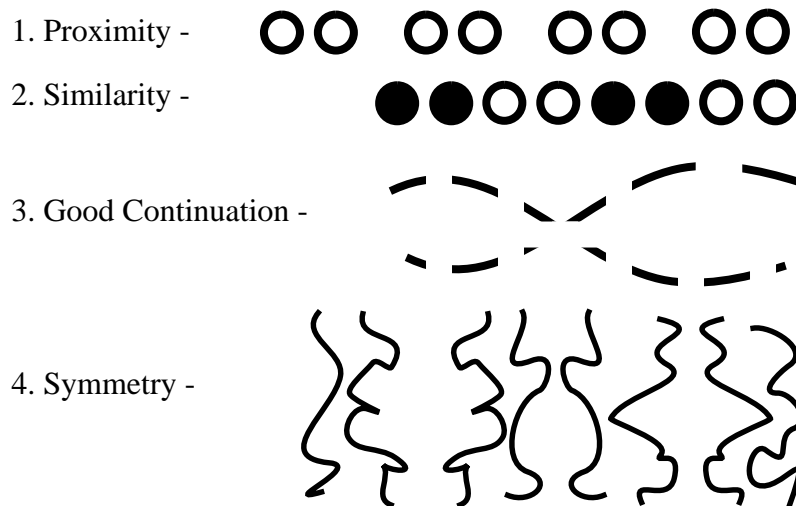


Figure 1.4 Important laws of Gestalt we use in our work.(after [58])
 1. Proximity tends to group close-by objects.
 2. Similarity will group similar objects.
 3. The dashed lines group based on good continuation properties.
 4. Symmetric features tend to group.

Figure 1.4 lists some important laws of grouping. Since most of our work deals with input in the form of edges, the laws most relevant to our work relate to proximity

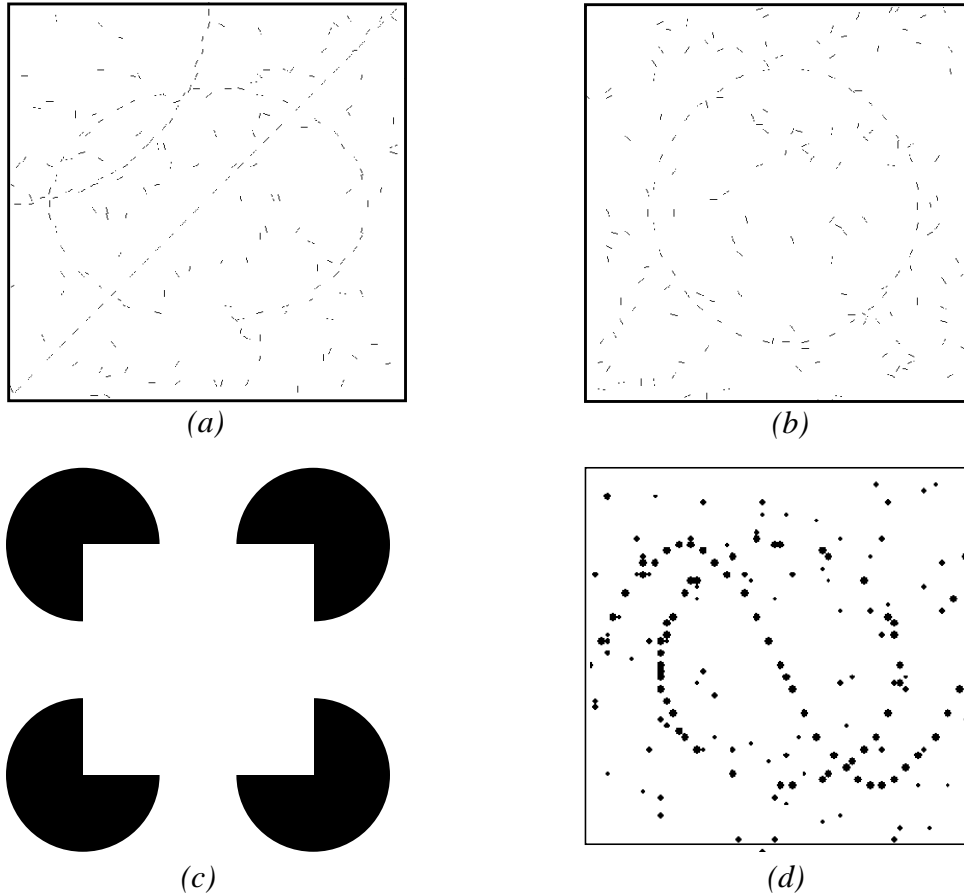


Figure 1.3 (a) & (b) Two instances of perceptual arrangements. (c) The Kanizsa Square. (d) A dot formation.

known to only take several hundreds of milliseconds (200-500 ms) to complete, and are thus not likely to utilize any high-level reasoning mechanism in the brain [7].

The circle in the middle of figure 1.3(a) is easily distinguishable from its noisy background. Furthermore, we tend to fill the gaps and accept the fragmented circle as a complete one. More precisely, we are able to complete the circle mentally. The same holds for the geometrical patterns in 1.3(b). A more striking example of illusory con-

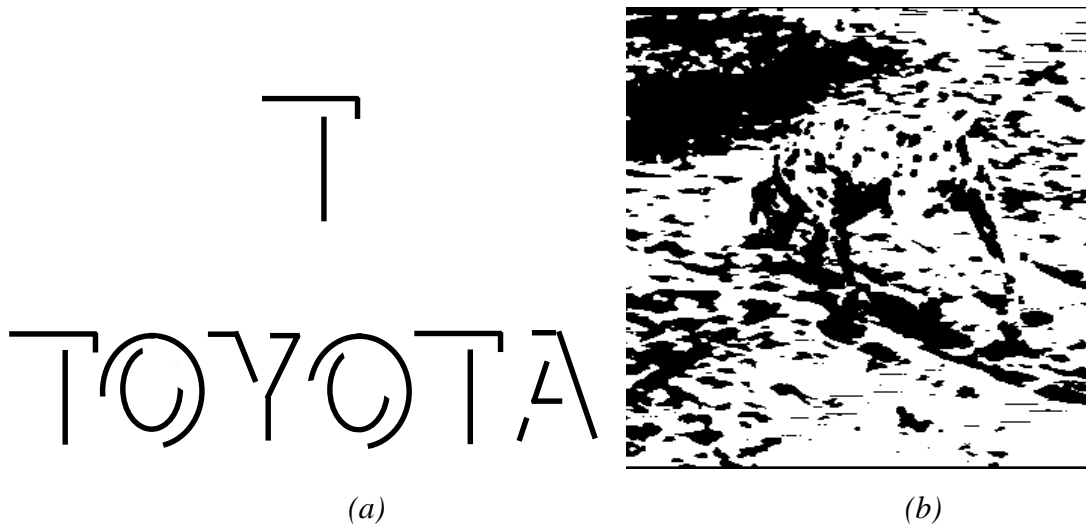


Figure 1.2 Examples of attentive perceptual groupings. (From [60] and [22])

familiar word TOYOTA. The letters become recognizable when put in the familiar formation of a word, and with the help of the word being a common one (after [60]).

Figure 1.2(b) also makes use of our representation of a dog in order to recognize the dalmation dog. Any part of the image by itself is not salient (from [22]).

Attentive processes take considerably longer than the pre-attentive ones. Examples are known (e.g. in [43]) where perception takes as much as several *minutes*.

Our work does not attempt to replicate scenarii which are of the attentive kind. It does, however, tries to computationally mimic perceptual phenomena of pre-attentive nature. Figure 1.3 depicts examples of perceptual groupings which are of interest to us, and considered to be the result of a pre-attentive process. Such processes are

recognition schemes (like [65]) rely on at least partial connectedness of the edges, and cannot function if the edge image is very fragmented. Also, the amount of noise is directly proportional to the computational cost of finding ‘real’ objects in a scene, since all features (true and noise) have to be checked against the database of valid objects.

Using global perceptual considerations when attempting to connect fragmented edge images can alleviate many of the above problems, as we show later in this document.

1.2 Perceptual Grouping

Perceptual Grouping refers to a class of visual phenomena where grouping of physically non-connected elements in the image occurs. This task turns into a figure-ground problem when patterns are embedded in noise. A simple every-day example would be a car moving behind a tree. Humans have no problem deciding that the two ‘half-cars’ visible from both sides of the tree belong to the same object, and in a sense we group the elements into one entity. Many rules (or mechanisms) have been proposed to explain our grouping decisions, as we discuss next.

We divide perceptual grouping into two main branches, *attentive* and *pre-attentive* brain processes.

An *attentive* process is one which needs to use previously acquired knowledge in order to perceive an object in an image. Figure 1.2 shows two examples of attentive groupings. The character T (Figure 1.2(a)) is not recognizable when isolated from the

1.1 Motivation

Consider, for example, the scene shown in Figure 1.1(a) and its edge image¹ as shown in Figure 1.1(b). As the edge image clearly shows, the outlines of the objects are not perfect, noise is present, and in many areas in the image no local support is present to produce an edge. Furthermore, in most cases, manual tuning of thresholds is required in order to get acceptable edges.

Methods for edge labeling (like [28,15,80]) assume perfect segmentation and connectivity, and define constraints which are only valid under these assumptions. These methods cannot work on such edges. Other methods, like shape from contour [78], and representation of objects [59, 68], also rely heavily on the connectedness of the edges, and can benefit from the removal of noise (erroneous segments). Pattern

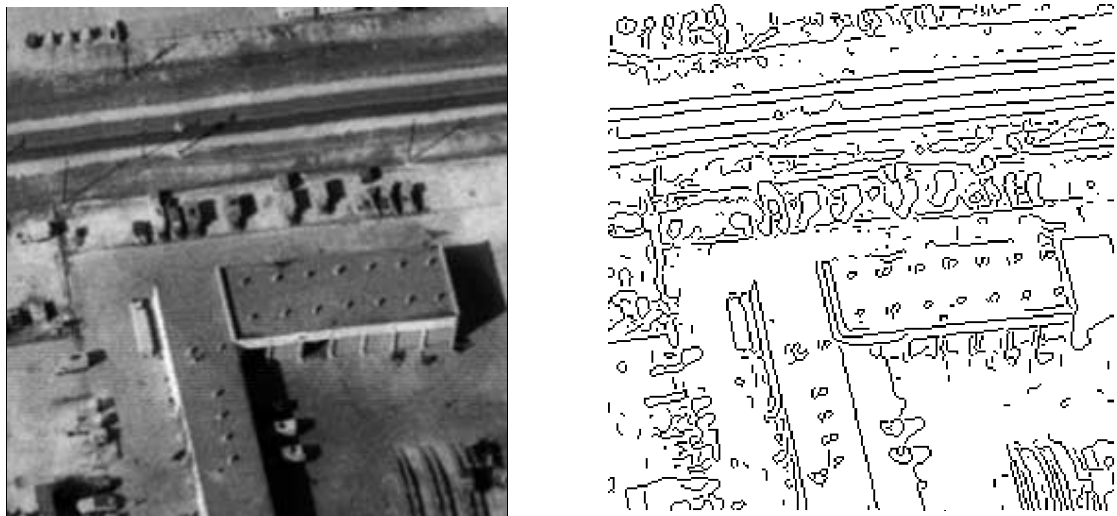


Figure 1.1 An image and its edges as produced by state-of-the-art edge detectors.

1. Using a Canny edge detector [14] with a high threshold of 2 (CME system)

Chapter 1

Inference in 2-D

Introduction

This work aims at bridging the gap between what is produced by state-of-the-art low-level algorithms (such as edge detectors) and what is desired as input to high level algorithms (perfect contours, no noise, no fragmentation, etc.). Many researchers resort to using synthetic data as their input because of these weaknesses. We believe that this gap can be bridged only by imposing constraints which are non-local. The constraints we impose here are derived from perceptual and computational considerations.

We treat the problem in 2-D and in 3-D. Chapters 1 through 3 discuss the 2-D case where input is expected in the form of a sparse edge map and/or a set of un-oriented points, and our output describes the same scene in terms of connected curves. In the 3-D case (Chapter 4) we expect a 3-D cloud of points with or without normal information. Here our approach attempts to describe the scene in terms of triangulated surfaces, space curves, and 3-D junctions.

The scheme is non-iterative, parameter-free, can handle multiple objects, each with any size genus, and does not require an initial guess. Moreover, it can handle large amounts of noise, both in the form of erroneous input primitives, and in the localization accuracy of valid input samples.

We present results on synthetic and real data.

Abstract

We address the problem of inferring high-level descriptions from sparse and noisy input data in 2-D and in 3-D. We claim that a local process cannot always capture meaningful structure among input data points, and more global constraints need to be imposed. Our system employs a global voting scheme that makes use of perceptual grouping constraints. The constraints relate to properties such as smoothness, co-curvilinearity, proximity, and curvature. These are captured into a *single* vector field applied at each input site. Thus, the voting process becomes a superposition of these fields over the entire input space.

The result of the voting phase is represented in a compact way, by keeping a covariance matrix at each site (2×2 in 2-D, and 3×3 in 3-D).

The interpretation of the voting phase generates a dense saliency map which lends itself to easy extraction of high-level primitives. These include junctions and edges in 2-D, and junctions, space curves, and surfaces in 3-D.

The result is in the form of dense saliency maps for curves and junctions (in 2-D), and surfaces, intersections between surfaces, and 3-D junctions (in 3-D). These saliency maps are then used to guide a ‘marching’ process to generate a high-level description. In the 2-D case, the description is in terms of connected curves and junctions, while in the 3-D case it consists of polygonal meshes, polygonal space curves, and 3-D junctions.

List of Tables

Table 1: Comparison of different grouping techniques	28
Table 2: Comparison between the Hough Transform and Our Scheme	89
Table 3: Saliency measures.	123

Figure 6.23	Dealing with a genus 1 object. (a) sampled torus (only the area of the middle hole is present (about 300 points). (b) resulting mesh. Note the ghost surface in the center. Its relative saliency is about 30% that of the torus itself. .	147
Figure 6.24	A 3D digitizer can return the location of the tip and normal vector of the stylus (MicroScribe-3D from Immersion Inc.)	148
Figure 6.25	(a) the egg. (b) estimated normals. (c) resulting mesh	149
Figure 6.26	(a) The cashew nut, (b) input point set, and (c) the resulting polygonal mesh.	151
Figure 6.27	(a) the curved block. (b) the original sweep. (c) a thresholded version of the curve saliency map. (d) explicit curves and junctions. (e) polygonal mesh.	152
Figure 6.28	The wooden piece. (a) input points. (b) recovered normals. (c) extracted surfaces (two views).	154
Figure 6.29	The parameter t grows from 0 to $\pi/2$ for all curves connecting and tangent to the two segments.	165
Figure 6.30	Moiré Patterns in 1-D. The original high frequency signal is sampled at low frequency creating a completely new signal (bold line).	169
Figure 6.31	A 2-D Moiré illustrating a pattern similar to our extension field, by combining two radial repeating patterns situated side by side (point fields!). This is another way to construct the basic extension field.	169

	section of the lines connecting opposite zero-crossings. (c) only three zero-crossings are found, no curve passing through that face.	130
Figure 6.12	(a) a surface patch with its normal. (b) the saliency along a line tangent to the normal, and (c) the derivative of that saliency.	133
Figure 6.13	a few of the more common situations in Marching cubes. (a) All values are positive, no surface, (b) a triangular patch describes one negative value, (c) two adjacent negative values define a rectangular patch (or two triangles), (d) four negative values define a rectangle, again.(e) three negatives generate a five-sided polygon, (f) it is possible to get two independent surface patches from a single voxel.	134
Figure 6.14	(a) A schematic model of input (The brighter lines denote the intersections between surfaces, and the dot is the 3D junction). (b) Projection of input samples.	138
Figure 6.15	Recovering surfaces, intersections, and junctions, independently.	139
Figure 6.16	Final description of the 3 planes. (a) curves and junction. (b) 12 polygonal meshes.	140
Figure 6.17	Detecting intersection of smooth surfaces	141
Figure 6.18	Recovering a complex non-constant curvature sheet surface. (a) sampled saddle function ($f(x,y)=x^2-y^2$) about a 100 points + 50 random points. (b) recover normals. (c) recovered mesh going through the maximal values of the saliency map.	143
Figure 6.19	Evaluating the surface reconstruction error. A perfect saddle, an approximated surface from sparse data, and a difference surface.	144
Figure 6.20	Recovering the “peanut”.(a) Sampled peanut (~100 points), (b) the largest values of the surface saliency map. (c) a polygonal mesh.	145
Figure 6.21	final mesh. (a)Two spheres (side by side). (b) one inside the other.	146
Figure 6.22	(a) A set of points on a cylinder. (b) the recovered surface.	147

Figure 6.2	What is the most likely normal to a surface passing through point P and at the same time tangent to the patch at the origin?	110
Figure 6.4	The parameters r , r and q in equation (6.1).	111
Figure 6.3	(a) The general shape of the Diabolo Field. The lower part is a mirror image of the top, and omitted here for clarity. Field vectors (not shown) are normal to the “bowl” surfaces shown. (b) a cross-section through the $y=0$ plane, with actual voting vectors.	112
Figure 6.5	(a) All normals (thin black arrows) at point p are equally likely. We choose to represent all of them with one vector (grey thick arrow) perpendicular to the plane they all lie on. (b) A cross-section of the Point Field at $y=0$. .	114
Figure 6.6	The general shape of the curve segment field. (a) All planes go to infinity, with diminishing strength. (b) A cross-section at $z=0$. The field elements are normals to the drawn planes	116
Figure 6.7	The three important voting ellipsoids. (a) $\lambda_{\max} \gg \lambda_{\text{mid}} = \lambda_{\min}$, high agreement in exactly one direction (a surface). (b) $\lambda_{\max} = \lambda_{\text{mid}} \gg \lambda_{\min}$, high agreement in exactly two orientations (an intersection, or 3-D curve). (c) $\lambda_{\max} = \lambda_{\text{mid}} = \lambda_{\min}$, votes are coming from all directions (a 3D junction).	121
Figure 6.8	(a) Data obtained by sweeping a 3-D digitizer along an egg. (b) Estimating tangent curves by connecting every two adjacent points.	125
Figure 6.9	The desired description. (a) A 3D object. (b) description in terms of surfaces, curves, and junctions.	127
Figure 6.10	(a) saliency projected onto a plane perpendicular to the curves’ tangent. (b) a change in derivatives signs in both v and u indicate a curve is passing through that face.	130
Figure 6.11	Different possible labels of voxel faces. (a) and (b) All four sides have a zero-crossing (denoted by the void circle) and the curve passes at the inter-	

Figure 3.33	(a)-(e) extracted curves for 2,4,6,8 and 10% of additive noise. (f) input with 10% of noise	79
Figure 3.34	An end-point formation. (a) A center egg-like shape is not only perceived but also looks whiter (after [7]). (b) An invisible circle occludes lines. No sensation of a circle is evident, because angles of intersection are not suitable. (c) The inner circle is perceived, but the outer one is not!	81
Figure 3.35	Given two segments A and B, the grey area is proportional to the probability that A and B intersect for a certain angle α	82
Figure 3.36	Convex and concave T-junctions. The Kanizsa square has two valid 3-D interpretations, but only one of them is perceived (b). The other one in (a) requires us to imagine concave T-junctions.	84
Figure 3.37	A multi-directional edge constructor for the End-Point Field. (envelope shown).	85
Figure 3.38	Results of applying the end-point field. Note that the outer circle is not highlighted! (a) Original image. (b) Saliency map. (c) after thresholding single votes.	85
Figure 4.1	A sparse set of 3-D points (with normals) are sampled from a plane intersecting a sphere. The grey line represents the intersection contour and is not explicit in the data.	92
Figure 4.2	Examples where local techniques fail. (For illustration purposes we show a 2 dimensional scenario. The 3D scenario is at best similar, and in most cases even harder). (a) a set of input points. (b) The results of connecting each point to its nearest neighbor. (c) the “correct” connections can only be inferred based on a more global view of the scene.	95
Figure 4.3	(a) A blob inside another blob (e.g. in medical imaging). (b) linked tori.	97
Figure 6.1	Flow chart of our system.	107

Figure 3.20	(a) Eccentricity only map of a saliency map of a straight line. (b) raw saliency map. (c) Product of (a) and (b). (d) -(f) same for a perfect circle.	60
Figure 3.21	Junction sites are formed when two or more sets of edgels vote for a single site, but from two (or more) distinct directions.	63
Figure 3.22	Some typical scenarios encountered by the Marching Squares algorithm. (a) a curve crosses the site diagonally. (b) top to bottom. (c) no curve since all vertices are of the same sign. (d) an ambiguous case where two solutions exist, and cannot be resolved locally.	66
Figure 3.23	(plot 1) Distribution of votes for a structured image (with real underlying curves), and (plot 2) a random image. The standard deviation (in the Y direction) for plot 1 is 334.59 compared to 122.275 for plot 2.	69
Figure 3.24	(a) an input image with edgels separated by increasing number of pixels. (b) a plot of the saliency profile. Note that the first 3 pairs from the top give rise to a single-peak profile, while the rest 'sag' in the middle.	71
Figure 3.25	(a) input. (b) result of marching along the highest ridges.	72
Figure 3.26	(a) Two converging lines. (b) the result of marching along the ridges. Note how the lines merge into one when the distance is about 3 sites apart. . .	73
Figure 3.27	The Saliency maps of images in figure 1.2.	73
Figure 3.28	Extracting the most salient features. (a) Largest eigenvalue strength map. (b) Eccentricity enhanced map. (c) Junction saliency map, and (d) linking. .	74
Figure 3.29	The result of extracting the strongest curve.(b) All other curves extracted. (c) the relative strength as a 3-D plot.	75
Figure 3.30	(a) a plot of the saliency map for the Kanizsa square. (b) the result of marching along the strongest ridges	76
Figure 3.31	(a) A non-directional input image. (b) Saliency map, after applying the Point field followed by the directional extension field. (c) all extracted curves. (d) Saliency plot.	77
Figure 3.32	The enhanced saliency map when applying the straight field.	78

Figure 3.5	Both scenarios have same first order compatibility figure. We would, however, want to group the left curves, but not the right.	39
Figure 3.6	The basic Extension Field. (a) Direction, and (b) Strength.	40
Figure 3.7	Assigning a direction for every point in space	41
Figure 3.8	When an edge and a point form an angle which is greater than 90 degrees (along a circular arc connecting them) an elliptic connection has a lower total curvature. (a) a circular connection. (b) the elliptical connection. . . .	43
Figure 3.9	5 arrangements of two segments, each with a different separation angle.	44
Figure 3.10	Loss of orientation \Rightarrow Loss of perception. The segments of (a) were replaced by dots in (b). The perception of the circle is weakened.	46
Figure 3.11	(a) A typical dot formation. (b) A multi-directional edge and the resulting point field (c).	47
Figure 3.12	A continuum of field constructors. From the maximum certainty in orientation (left), to maximum uncertainty (right).	48
Figure 3.13	The shape of a semi-deterministic field. Such a field encodes some degree of uncertainty in the orientation of the input edge (± 1 radian).	49
Figure 3.14	One way of constructing the straight line Extension Field.	50
Figure 3.15	A Straight Field generated by convolving the Extension field with a straight line.	50
Figure 3.16	Superposition of two Extension Fields over a scenario of two co-linear short segments (dark lines).	51
Figure 3.17	Superposition of two Extension Fields over a scenario of two co-circular short segments (dark lines).	52
Figure 3.18	The principal axis of the votes collected at a site is taken as an approximation of the preferred direction.	55
Figure 3.19	Saliency of a line does not grow forever. It converges to some value which is the infinite straight integral of the extension field.	59

List of Figures

Figure 1.1	An image and its edges as produced by state-of-the-art edge detectors... 2
Figure 1.2	Examples of attentive perceptual groupings. (From [60] and [22]) 4
Figure 1.3	(a) & (b) Two instances of perceptual arrangements. (c) The Kanizsa Square. (d) A dot formation. 5
Figure 1.4	Important laws of Gestalt we use in our work.(after [58]) 6
Figure 1.5	Conflict between proximity grouping and good continuation. (a) a set of input points. (b) The results of connecting each point to its nearest neighbor. (c) the correct connections can only be inferred based on a more global view of the scene. 7
Figure 1.6	Different perceptions at different scales. (after [17]). The crooked line in (c) is seen as a straight vertical line (d) at a coarser scale, and as either a straight tilted line (a) or a curved segment in (b). 8
Figure 2.1	Parameters used in computing the proximity of two end-points. 15
Figure 2.2	Parameters used in computing the parallelism of two segments. 16
Figure 2.3	Parameters used in computing the collinearity of two segments. 17
Figure 2.4	A typical input handled by Ahuja and Tuceryan [1] in (a), and the result of applying their method in (b). 21
Figure 2.5	A typical input image (after [61]). The algorithm will assign high values of saliency along the fragmented circle. 24
Figure 3.1	A simple input site model. Every site is associated with a preferred direction, strength and eccentricity (or uncertainty). 33
Figure 3.2	An obscured figure (a) triggers the perception of simple shapes (b), instead of the more complex (c) or (d). (From [58]). 36
Figure 3.3	What is the shape of the most ‘natural’ extension to a given curve? 37
Figure 3.4	What is the best path between these two curves? 38

6.2.1	The Diabolo Field	109
6.2.2	The 3-D Point field	113
6.2.3	The curve segment field.	113
6.2.4	Unification	115
6.3	Implementation of the Algorithm.	117
6.3.1	Vector Convolution and Vote Aggregation.	117
6.3.2	Combination at each voxel	119
6.3.3	Vote Interpretation	119
	6.3.3.1 The oriented case	120
	6.3.3.2 The non-oriented case	124
	6.3.3.3 The partially-oriented case	124
6.3.4	Noise tolerance	126
6.4	High-level Description	126
6.4.1	Recovering 3D junctions from a junction saliency map	127
6.4.2	Recovering space curves from a curve saliency map	128
6.4.3	Recovering triangulated surfaces from a surface saliency map by applying the Marching Cubes algorithm	131
	6.4.3.1 Our modifications	131
	6.4.3.2 The classical algorithm	134
6.4.4	Integrating junction, curve, and surface information	135
6.5	Complexity	136
6.6	Results.	137
6.6.1	Oriented input	137
6.6.2	Non-oriented input	142
6.6.3	Real data.	148
6.7	Limitations	153
6.8	Conclusion and Future Research	155

8 References 158

3.9.1.1	Extracting Junctions	64
3.9.1.2	Extracting curves	65
3.9.2	Determining absolute saliency and noise threshold.	68
3.9.3	Experimental Results	70
3.9.3.1	Basic arrangements	70
3.9.3.2	Synthetic Images	73
3.9.3.3	Point Field	76
3.9.3.4	Straight line Field	76
3.9.3.5	Noise Breakdown Point	77
3.9.4	Complexity Issues.	80
3.10	Application of scheme to end points (End-point field)	80
3.10.1	Straight angles in T-Junctions are more likely than any other	81
3.10.2	Convex T-junctions are more common	83
3.10.3	Building the End-Point Field	84
3.10.4	Experimenting with the End-Point field	85
3.10.5	End-point and Extension field interaction	85
3.11	Comparison With the Hough Transform	86
3.11.1	The classical Hough transform	86
3.11.2	Our scheme as a Hough transform	87
3.12	Conclusion and Future Work	89
5	Inference in 3-D	91
4.1	Introduction	91
4.1.1	Issues in 3D reconstruction	94
4.1.2	Our Approach	101
6	Survey of Related Work (3-D)	102
5.1	Related Work	102
5.1.1	Computational-geometry based approach.	104
5.1.2	Discussion.	105
7	Our Approach (3-D)	106
6.1	Overview	106
6.2	The design of the fields	109

	2.3.11 Williams	26
	2.4 Comparison and summary	28
4	Our Approach (2-D)	30
	3.1 Overview	30
	3.2 Philosophy	32
	3.3 Model of the Input	33
	3.4 Model of the Output.	34
	3.5 Rationale for the Extension Field	35
	3.5.1 The perceptual constraints	35
	3.5.2 Extending a Curve.	36
	3.5.3 Best Connection between Two Line Segments	38
	3.6 The Extension Field: design and implementation.	39
	3.6.1 Design of the Extension Field (Orientation and Strength)	40
	3.6.1.1 Shape and Orientation	41
	3.6.1.2 Strength	43
	3.6.2 Other fields	46
	3.6.3 Unifying the field concept	48
	3.6.4 Special Purpose Fields: The Straight Field	48
	3.7 Computation of the Saliency Map	51
	3.7.1 Vote Accumulation	51
	3.7.2 Vote representation	53
	3.7.3 Saliency measure	55
	3.7.4 Properties of the extension field	59
	3.7.4.1 A longer line implies a stronger and more directed field, but up to a point.	59
	3.7.4.2 The Non-maximum Suppression Phenomenon	60
	3.7.5 Detection of Junctions.	62
	3.8 Multiple Resolution	63
	3.9 High-level Feature Extraction.	64
	3.9.1 Description (Curves And Junctions)	64

Table of Contents

Acknowledgments	iii
Table of Contents	iv
List of Figures	viii
List of Tables	xv
Abstract	xvi
1 Inference in 2-D	1
2 Introduction	1
1.1 Motivation	2
1.2 Perceptual Grouping	3
1.3 Scale Dependency	7
3 Survey of Related Research (2-D)	10
2.1 Texture segmentation and Classification	10
2.2 Grey-level organization	12
2.3 Edge-based organization	14
2.3.1 Lowe	14
2.3.1.1 Proximity Grouping	15
2.3.1.2 Parallelism Grouping	16
2.3.1.3 Collinearity Grouping	17
2.3.2 Dolan and Weiss	18
2.3.3 Zucker	20
2.3.4 Ahuja and Tuceryan	20
2.3.5 Mohan and Nevatia	22
2.3.6 Sha'ashua and Ullman	23
2.3.7 Parvin	25
2.3.8 Huttenlocher and Wayner	25
2.3.9 Parent and Zucker	26
2.3.10 Heitger and von der Heydt	26
	iv

Acknowledgments

I thank my advisor and chairman of my thesis committee, Dr. Gérard Medioni, for all the support and advice throughout my studies at USC. I thank Dr. Ulrich Neumann and Dr. Irving Biederman for finding time to serve on my thesis committee and provide invaluable suggestions to improve the final dissertation.

Special thanks go to Dr. Keith Price, Andres Huertas, and the rest of the IRIS gang, Parag Havaladar, Mi-Suen Lee, Sanjay Noronha, Chia-Wei Liao, and all the other members, for various discussions, and challenging Tetris tournaments...

I also want to thank Delsa Castello for helping me sort through the bureaucratic jungle at USC.

I thank Jean Ponce for triggering the taught process that led to the local zero-crossing formulation for surface extraction.

My deepest gratitude goes to my wife Talia who managed to put up with me for that long, and my supportive family across the ocean in Israel.

**This work is dedicated with love to my
daughters, Maya and Li-Elle**

Inference of Multiple Curves and Surfaces from Sparse Data

Gideon Guy

A Dissertation Presented to the

FACULTY OF THE GRADUATE SCHOOL

UNIVERSITY OF SOUTHERN CALIFORNIA

In Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

(Electrical Engineering)

Institute for Robotics and Intelligent Systems

School of Engineering

Powell Hall Room 204 - MC 0273

University of Southern California

Los Angeles, California 90089-0273

December 1995

Copyright © 1995 Gideon Guy