Section 2: A Generative Transformational Approach to Modelling Visual Perception

Chapter 7: Fractal Decomposition

This section is chiefly concerned with developing a generative transformation model of visual perception, although it addresses related matters and other concerns in later chapters. Most of the patterns described in Chapter 1 are employed by the model in pattern discrimination, and some new patterns are also employed. Most of the work is original, and work anywhere in the section that is not original is clearly referenced.

Brief summary of chapter

In this chapter a method for decomposing fractal objects is developed, which is achieved by employing affine transformations. Two copies, in the form of point sets, are made of the objects. One of these is superposed and repeatedly contracted with respect to the other. At each contraction, the sets are searched for optimal match from the contracted set to the original set.

Identification of transformation angle(s) is performed either during establishment of the proper contraction or afterwards, depending on whether or not the first segment is angled away from a reference direction. This is done by rotating and translating the contracted copy with respect to the original copy, and searching for the same optimal match.

A method for identifying the number of replacements depends on counting repetitions of the same optimal match during reduced copy translations and rotations to the discovered angle(s). Once all the parameters have been identified, the original fractal can then be reproduced. Using these techniques, the method can successfully decompose a variety of both regular and probabilistic geometric fractals, whether they be simple line shapes or closed figures.

The work outlined in this chapter, with the exception of that indicated of others along with basic fractal generation techniques, is original. Particularly, the fractal decomposition methods are original, and all software used in the endeavour is original.

Invariance and the fractal geometry of natural objects and scenes

Pentland (1986) has claimed that the most important role of perception is to recognise lawful regularities that indicate causal structure in the environment, thereby allowing the observer to anticipate events and predict consequences. Because the interpretation of a visual image is typically underdetermined by information available only from the image, such recognition must make use of some model of the environment that produced the image. As Pentland (1986) expresses it, '[a] theory of visual function that has no model of the world also has no meaning' (p. ix).

Models incorporated in theories of vision vary widely in their specificity. For example, one class of model employs a development of Plato's theory of ideal forms, in which objects are described in terms of a small number of regular three-dimensional shapes, such as spheres, cylinders, and cubes, with smooth surfaces. The models may work reasonably when applied to images of built environments. However, they become awkward and complex when applied to irregular surfaces typical of natural scenes.

According to Pentland (1984, 1986), the iterated functions of fractal geometry provide a useful model of the structure exhibited by many—if not most—natural surfaces. A similar standpoint has been developed by a number of researchers in a range of disciplines. Among the most notable is Mandelbrot (1977, 1983), who has extensively analyzed and systematized a novel class of mathematical functions known as *fractals*. Also prominent are biologists, such as Lindenmayer and Prusinkiewicz (1990), who have developed a computational (L-system) language for describing the morphology and growth of organic structures, such as plants, including bushes and trees.

Recently, researchers concerned with computer graphics and with descriptions of the natural world have recognized the importance of non-linear dynamical aspects of the systems under study and the fertility of descriptions based on fractal geometry. In particular, work on iterated function systems has demonstrated that visual images of complex natural structures, such as trees or mountains, can be encoded in a compressed form by the parameters of a collage of affine transformations applied to a number of reduced copies of a single seed element, such as a line or rectangle (Barnsley, 1988a, 1988b; Barnsley & Anson, 1993; Barnsley & Hurd, 1993; Jacquin, 1992). Conversely, an approximation of the original image can be unambiguously and efficiently regenerated to any desired level of precision by the continued probabilistic iteration of the same collage of transformations. Such processes of fractal encoding are now finding increasing commercial application to the compression and generation of visual imagery (e.g. feature films) on to media such as compact discs (Anson, 1993).

Fractal encoding as a model for visual perception

The above developments mainly have to do with the intrinsic structure of the objects rather than with the optical structuring of the image resulting from the interrelations between the object, the light source, and the observer. Vickers (1979) argued that the perception of such inherent structure might be mediated by a sequence of constructive processes—mimicking those in the natural world—the results of which were then subjected to various projective transformations. These operated so as to maximize invariance in the stimulus pattern (under certain constraints), thereby giving rise to the perception of orientation and layout. Vickers' tentative suggestions predated the conceptual advances made possible by the above developments. However, we believe that these ideas can now be well formulated.

It can be proved that any image, however complex, can be represented by the parameters of a set of fractal generation processes. This suggests a very economical way of encoding visual images. As Peitgen, Jürgens, and Saupe (1992a) remark, 'Fractal geometry offers a totally new and powerful modelling framework for such encoding problems. In fact, we could speculate that our brain uses fractal-like encoding schemes' (p. 259).[1] Specifically, it could be speculated that, in its tendency towards simplicity and economy, the human visual system might employ something like a fractal encoding process. One aim of the thesis is to explore this possibility.

For example, the perception of 'what' (perceived intrinsic structure) may be mediated by neural networks collating the outputs of probabilistically iterated transformations, the parameters of which are rapidly tuned so as to minimize any discrepancy between their attractor-like outputs and information from the retina. Since knowing the algorithm underlying a natural structure is akin to understanding that structure, this would endow perception itself with some intelligence, independently of its interactions with higher cognitive processes.

Turning to the perception of 'where' (topographical perception); within the field of computer vision there has been a recent resurgence of interest in the importance of geometric invariants under different groups of (affine and projective) transformations and the

[1] For a lucid account of the various fractals and their generation, the two books by Peitgen, Jürgens, and Saupe (1992a, 1992b) are recommended.

formulation of a systematic and coherent programme of research (Mundy & Zisserman, 1992a). Within human perception, there has also been a long tradition of interest in the importance of projective invariants (exemplified notably by the work of Gibson (1968) and Cutting (1986).

As pointed out by Van Gool, Moons, Pauwels, and Wagemans (1994), while some of the earlier search for reliance on invariants in human perception may have yielded a lean harvest, many of the recent approaches in computer vision are efficient and robust but seem still unknown to the perception community. For example, recent work reviewed by Mundy and Zisserman (1992b) has demonstrated that affine invariants may be obtained from as few as two distinct views of a structure. Moreover, some earlier studies may have looked for invariance at an inappropriate stage. Recently, for example, Farah, Rochlin, and Klein (1994) have replicated previous failures to find orientation invariance with wire forms (contours), but have found good or perfect orientation invariance with equivalently shaped surfaces.

Given the obvious utility and importance of invariants in computer vision, the notion that invariants under affine and projective transformations might provide a basis for topographical perception seems plausible. In addition, the operation of the perceptual system, as conceptualized in the transformational approach, is consistent with most of the basic notions of the four approaches to perception listed earlier (i.e., the neurophysiological, the ecological, the inferential approaches based on likelihood, and those based on economy of coding).

This conceptualization suggests that the perception of ‘what’ and the perception of ‘where’ might be mediated by two interlaced but distinguishable systems of transformations. More particularly, this way of looking at visual perception places a number of problems in perception into an entirely new perspective. To take but one example, the contraction mapping principle, which makes possible the iterated function system approach to encoding, depends upon the transformations being affine. This condition is ensured by a metric in which distances in the horizontal and vertical directions are differentially weighted. In turn, the requirement for differential weighting raises the possibility that the *horizontal-vertical illusion* (in which a vertical line is seen as longer than a horizontal line although both are physically equal in length) may be not so much a perceptual oddity as a necessary and inherent feature of the encoding process.

Decomposing fractal curves, outline figures, and surfaces

All the information necessary to generate a fractal curve can be represented by the parameters of the set (or collage) of some half dozen transformations. With the above as background, we set about designing a system that was plausible from an evolutionary perspective, and that was capable of solving the inverse problem of finding the fractal encoding for a visual array. We set about designing a system that was capable of using a constrained sequence of transformations to work out how a wide range of fractal curves are generated, and that was capable also of generating a copy of the curve and matching it against the original. (The underlying idea is that what we experience as perceptual organization is the output of a generative process that applies multiple transformations to stimulus elements, thereby producing an output that can be continually tuned and steered by matching it with the current visual input.)

Hausdorff distance

Hausdorff distance is the pattern matching mechanism we chose to implement our various strategies. Hausdorff distance is the largest nearest neighbour distance between a point in one set and a point in another set. Any point in one set is less than or equal to such a distance from a point in the other set. Partial Hausdorff distance, as it applies to the model, operates on parts of sets that are within common regions. Many fractals, including all line fractals, can be decomposed using partial Hausdorff distance as a measure of match involving affine transformations.[2]

Detailed description of Hausdorff distance

Hausdorff distance is named after Felix Hausdorff (1868-1942). Firstly, directed Hausdorff distance h(A→B) corresponds to the length of the longest nearest neighbour link among all the links from the points in a set A to their nearest point(s) in a set B. If h(A→B) = *d,* then all points in set A are less than or equal to distance *d* of a point in set B. Of course h(B→A) is the other directed Hausdorff distance, and corresponds to the length of the longest nearest neighbour link among all the links from the points in set B to their nearest point(s) in set A. If h(B→A) = *d,* then all points in set B are less than or equal to distance *d* of a point in set A.

Hausdorff distance H(A↔B) corresponds to the longer of the two longest nearest neighbour links. If H(A↔B) = *d,* then all points in set A are less than or equal to distance *d* of a point in set B, and vice versa. Essentially, directed Hausdorff distance refers to the largest nearest neighbour distance *from* one set *to* another, and Hausdorff distance refers to the largest nearest neighbour distance *between* two sets. Hausdorff distance is suited to pattern detection because it applies to all bounded sets, is non-negative, and obeys the properties of identity, symmetry, and triangle inequality (Cornell University, 1994).[3] Hausdorff distance is calculated as per Table 7.1, which refers to Figure 7.1.

[2] Strictly, Hausdorff distance is a measure of mismatch, such that minimum Hausdorff distance corresponds to minimum mismatch.

[3] The triangle inequality states that if two shapes are highly dissimilar, they cannot both be similar to a third shape.

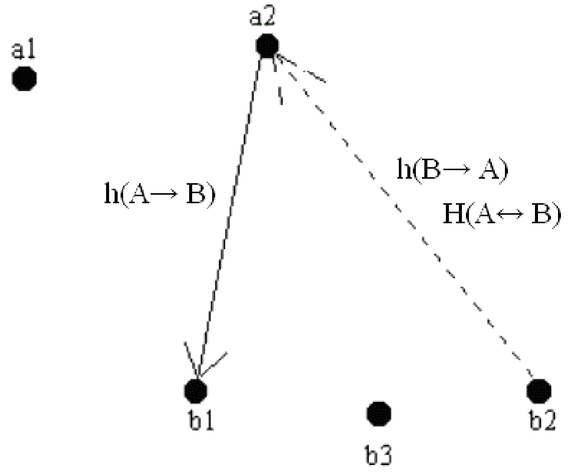


Figure 7.1: Two point sets, A and B. All points in set A are less than or equal to distance H(A↔B) of a point in set B, and vice versa. H(A↔B) is the Hausdorff distance.

Table 7.1: In reference to Figure 7.1, the first and second columns show directed Hausdorff distances in boldface type. The last column shows Hausdorff distance, which is the greater of the two directed distances. A and B are finite point sets, and # indicates a least Euclidean distance in arbitrary units.

Calculating Hausdorff distance from A to B	Calculating Hausdorff distance from B to A	**Hausdorff distance** (between A and B)
a1 b1 2086 #	b1 a1 2086 #	**H(A↔B) 2647**
a1 b2 3416	b1 a2 2146	
a1 b3 2805		
	b2 a1 3416	
a2 b1 2146 #	b2 a2 2647 #	
a2 b2 2647		
a2 b3 2416	b3 a1 2805	
	b3 a2 2416 #	
h(A→B) 2146		
	h(B→A) 2647	

Distances from points A to points B marked with crosshatches in the first column are nearest neighbour distances of points in set A to points in set B. And distances from points B to points A marked with crosshatches in the second column are nearest neighbour distances of points in set B to points in set A. Note that all distances A to B must have equals in all distances B to A, but in the case of point sets of different size the set structure of the A to B and B to A columns is different. Note, also, that distances within each of the columns need not be repeated.

If a set is superposed on another identical set, then nearest neighbour distances can be considered to be within the one set rather than between two sets. (Here, zero distances are omitted when doing h(A→B) or h(B→A) calculations.) For superposed, identical point sets, the set structure of the A to B and B to A columns is identical, and would the same as that for A to A or B to B columns. Since all distances A to B must have equals in all distances B to A, the columns are identical; hence only calculations for any one column need be performed.

Moreover, this yields a column within which each distance is duplicated by its reciprocal. In any event, the underlying consideration in Hausdorff calculations, which are pivotal to research reported in this section, is nearest neighbour yet again.

Partial Hausdorff distance

If one set partly overlies another, or one set lies inside another, then the distance between the common regions is obtained by performing a Hausdorff distance calculation that utilizes points from those regions only. This is called partial Hausdorff distance. Directed partial Hausdorff distance h(A→B)p is the measure most used for fractal decomposition in the generative transformation model, and directed Hausdorff distance h(A→B) is the measure most used for structure detection.

Fractals

Fractals have helped investigators to demonstrate the regular underlying features of many processes and related structures found in nature. For example, a fern has a fractal structure. A fern consists of leaflets that have the shape of the fern itself, and each leaflet in turn is made up of smaller leaflets that again have the same shape as the fern.

Many plants, including cauliflower and broccoli, have fractal structures. The structure of the blood vessels of the heart is also that of a fractal: large vessels branch into smaller ones that in turn branch into even smaller ones. Likewise for the blood vessels in the kidneys; and the same principle applies to the airways in the lungs.

A corresponding principle applies to natural processes. Heartbeats, for example, show fractal patterns (Thomas, 1999). If the heart speeds up over a short sequence of beats, it will slow down over the next few beats. The pattern of quickening and slowing can be seen over both short and long time scales: it looks the same over a few beats as it does over many beats. Fractal process and structure exist at all scales in the universe, from the tiny through to galactic clusters and beyond (Hamilton, 1988; Ribeiro, 2005). By and large, natural objects possess self-symmetry or *self-similarity,* which property is *scale invariant.*

There are two main types of fractals: regular fractals and random fractals, from which images can be produced by operations on dots, lines, areas, and volumes. Regular fractals, also called geometric fractals, have large and small structures that, except for their size, are copies of one another. For example, a regular fractal known as the Koch curve has small triangles added to the sides of larger triangles. Each time the fractal generator is iterated, the entire structure is contracted by a factor of three and repeated four times to make the next order curve. See Figures 7.2 and 7.3.

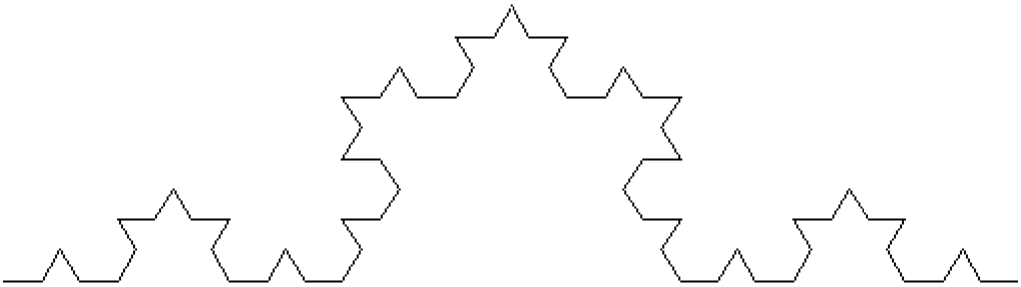


Figure 7.2: Koch curve at iteration level three.

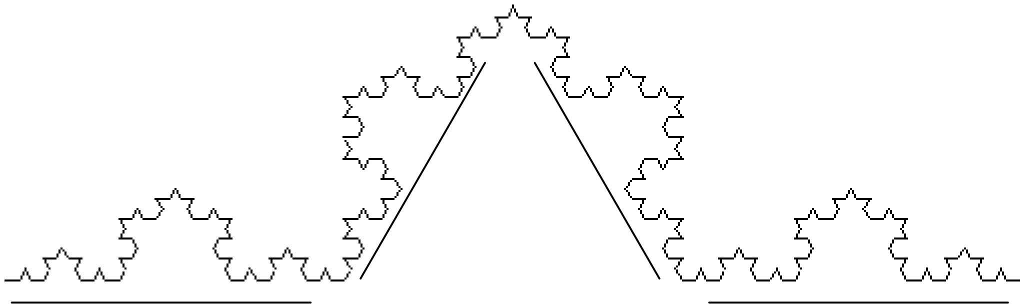


Figure 7.3: Koch curve at iteration level four. The section above each line is a contracted level-three Koch curve.

Such fractals are called 'curves' because, in the limit, the lines become infinitely short and together make 'curves'. Nonetheless they are quite the opposite in kind to the continuous functions belonging to calculus because they are not strictly differentiable.

For random fractals the large- and small-scale structures are mathematically related, but may differ in detail. Random fractals represent irregular patterns found in nature. The shapes of coastlines, mountains, clouds, landscapes in general, and numerous plants, for example, can be represented by random fractals. Figure 7.4 shows a fern, a maple leaf, and a stick-like weed.

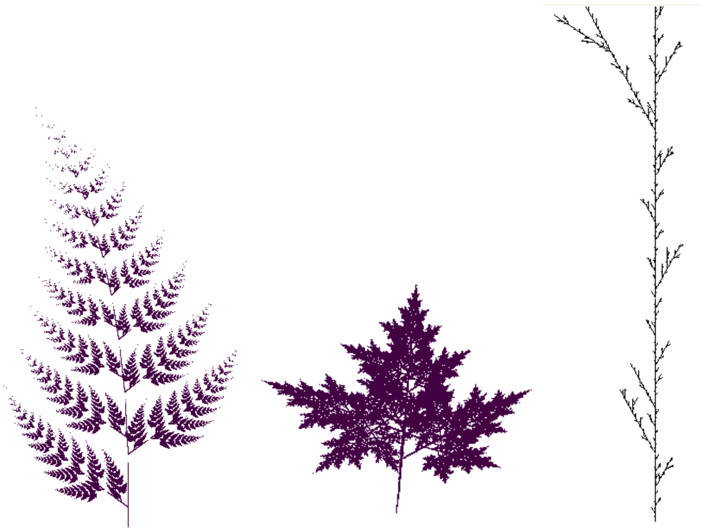


Figure 7.4: Fern, maple leaf, and stick-like weed implemented by probabilistic iterated functions.

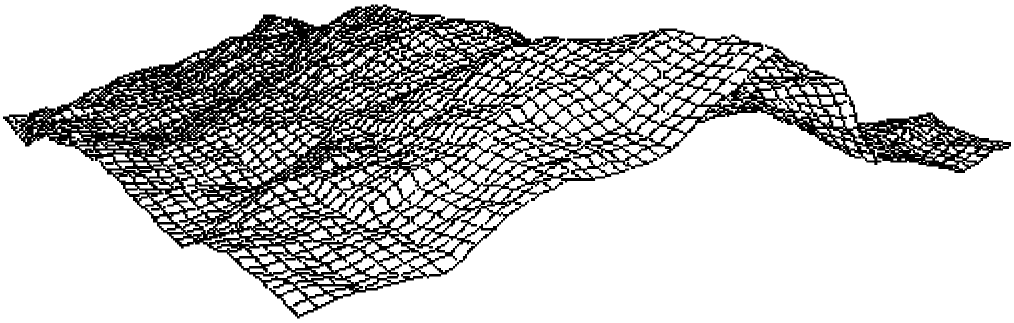


Figure 7.5: Fractal landform implemented by a probabilistic iterated function.

The fern and maple leaf were each generated by a sequence of dots positioned under the control of a probabilistic iterated function. Each *x*-coordinate is a function of the immediately previous *xy*-coordinate pair and each *y*-coordinate is also a function of the immediately previous *xy*-coordinate pair. The stick-like weed was generated by a sequence of lines positioned under the control of another form of probabilistic iterated function. Each *x*-coordinate, denoting the endpoint of a line, is a function of the immediately previous *xy*-coordinate pair in some subsequence of coordinate pairs. Each *y*-coordinate, denoting the endpoint of a line, is also a function of the immediately previous *xy*-coordinate pair in the subsequence of coordinate pairs. And a subsequence, itself, works in the same way by branching from a superordinate sequence. Figure 7.5 shows a fractal landform, which was generated by a sequence of lines, as sides of triangles, with midpoints displaced upwards or downwards under the control of yet another form of probabilistic iterated function. (So as to not congest the figure, only two of the three sides of the triangles are shown.) Each *xy*-coordinate, specifying the endpoint of a line, is a function of the pair of *xy*-coordinates immediately adjacent. These demonstrate a recursive process, whereupon an output is fed back as input to produce a new output, and so on.

Brownian motion—the random motion of a particle in a fluid or gas due to the spread of molecular kinetic energies—can be represented by a random fractal. Other examples of random fractal process are the growth of plants, the paths followed by lightning, and viscous fingering: the diffusion of one liquid into another in which it cannot dissolve, such as oil in water. Numerous aspects of the natural world can be modelled by fractal geometry, and any image can be produced by fractal geometry. Only special and limiting cases are modelled by regular geometry, such as idealized trajectories and ‘perfect’ shapes based on continuous functions.

By and large, forms associated with man-made objects belong to regular geometry, which involves lines, planes, cubics, arcs, spheres, cylinders, etc. And these relatively simple, ideal forms are used in one or another model to approximate the natural world, when all the while fractals accommodate the natural world with immeasurably greater precision. If fractals lend themselves so readily to modelling the natural world, then might a visual system, which has so long perceived the natural world and is a part of the natural world, be modelled in a complementary fashion? Might the visual system encode an image in terms of the parameters of the simplest set of transformations required to construct a model of the image that has the greatest amount of correspondence with it?

The half split method

The half split method repeatedly narrows a domain by splitting what remains in half. (Like locating a name in a telephone directory by first going to the middle page, then by going either forward or backward, depending on the initial letter, by half of what is left, and so on.) The simple method outlined below proves satisfactory for the applications considered herein. See Figure 7.6 in connection with the description. An ordinate is a *y*-axis value and an abscissa is an *x*-axis value.

After recording ordinates at discrete abscissa intervals (where the graph intersects the dotted lines), a check for the lowest value that intersects a discrete location is performed. The abscissa is then shifted halfway between a neighbouring discrete location and the current discrete location in an attempt to reduce the ordinate. Each horizontal direction is checked and the direction that results in the lower of the two ordinates is selected. The abscissa is then shifted halfway between the bounds of what is left in the same fashion as before, and so on, for some number of iterations that results in acceptable precision.

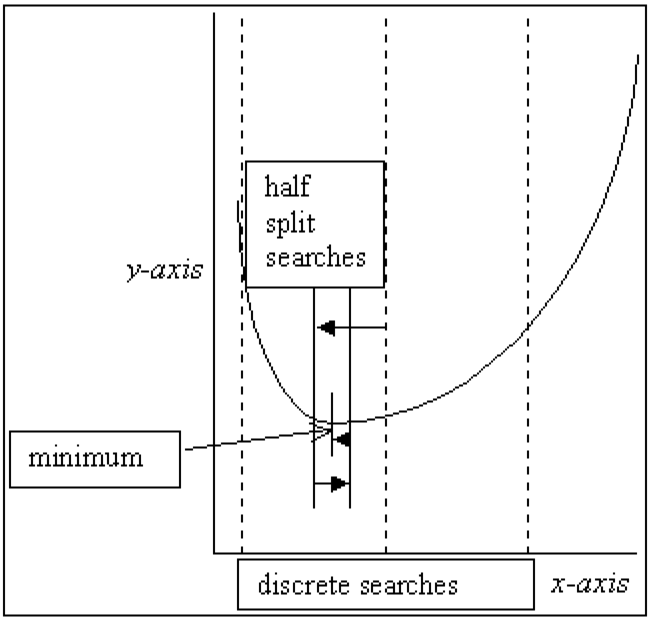


Figure 7.6: Direction of the half split search sequence is indicated by the solid arrows, and the order of the sequence is indicated by reduction in length of the arrows.

Where decomposing fractal objects is concerned, the x-axis would have contraction ratios as values and the y-axis would have values based on Hausdorff distance. In terms of fractal decomposition, in which one copy of a fractal is contracted against another (originally identical) copy by small discrete amounts, additional smaller contractions using the half split method indicate if a further lowering of a measure based on Hausdorff distance can be attained by shifting the contraction. It may be that no further lowering beyond a preceding discrete search is found. Indeed it is usually the case with a fractal that a coarse grained, discrete search alone finds the minimum.

Details of the model

A computational model of visual perception, based on fractal decomposition and encoding methods, and with some attention to Gestalt principles, is now described. A key factor is that of symmetry, which, in the context of the model, means an object has some invariance under affine transformation. This encapsulates much of the philosophy underpinning fractal decomposition and pattern detection.[4]

The operations employed are elementary, completely general, and were chosen to accord with the kinds of operations that could be asserted for networks of neurons. They include affine transformations: translations, rotations in the plane and in depth, contractions and dilations, and the like. They also include simple pattern matching operations; the most versatile and widely used being based upon minimization of some form of Hausdorff distance.

The model works with point sets of data, containing enough points to represent the designated images adequately. Operations initially involve loading two superposed, identical point sets, each into a separate array. One is transformed with respect to the other in some way, and then compared; usually by using results from the Hausdorff distance calculation. See Figure 7.7, in which one point set exactly overlies the other. (Figures are screen shots of a computer implementation of the model.)

[4] Non-rigid transformations are not treated here, but the general philosophy developed by the model can still apply to these.

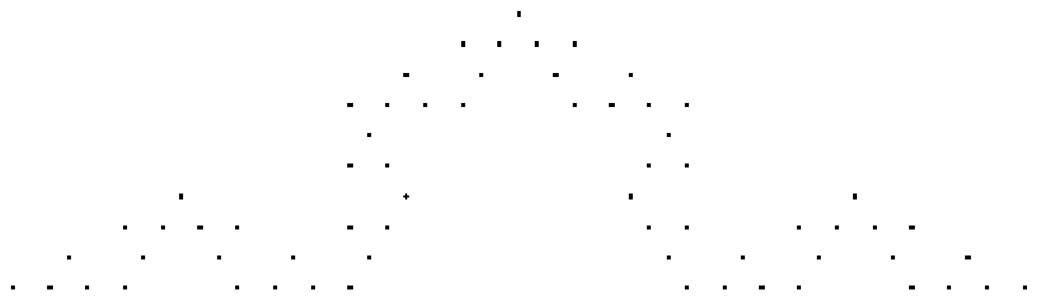


Figure 7.7: Point sets for the Koch curve, which is shown at iteration level three.

The two methods used in the model can be briefly described as 'reconciliation of transformations over images', and 'reconciliation of spatial and/or temporal relationships by proximity and relative frequency' respectively. The first is used to decompose fractal objects into just a handful of parameters per object. The parameters can then be used to reproduce the objects. The second is used in pattern detection, as per the next two chapters, with particular emphasis on the Gestalt, or whole, as the eye immediately perceives holistic order against some background.

Fractal decomposition

Fractal decomposition is an example of solving an inverse problem. The problem has enough constraints to result in a correct solution. Upon loading a regular fractal object, options are provided for decomposition. The large group of line fractals for which the Koch curve may be considered a prototype, can be decomposed by using a method that is slightly different to that for fractals based upon area, such as the Serpinski triangle, Any method shares the elements:

- find contraction factor and angle;
- find number of replacements.

Matching a contracted copy of a fractal against a segment of its original copy is equivalent to matching an original copy against a segment of its dilated copy. Terms applicable only to the former are employed herein.

Finding contraction factor and angle

Copy A of a fractal is contracted against copy B by small discrete amounts, down to an arbitrary low value of directed partial Hausdorff distance, h(A→B)p. At each contraction, h(A→B)p is saved. Afterwards, the earliest minimum value of h(A→B)p is chosen, and this often proves to be an absolute minimum. The use of discrete transformations, with values chosen at defined intervals, keeps transformations tested to manageable numbers and is justified by the restricted number of values typically found in both artificial and natural fractal structures. Nonetheless, contractions such as $1/\sqrt{2}$ are sometimes required. Further searching using the half split method indicates if a further lowering of h(A→B)p can be implemented by shifting the contraction.

The earliest minimum corresponds to a first order contraction, and is chosen because the same minimum h(A→B)p value is later repeated for the next order down, and so on, where such self-similar objects are concerned. See Figures 7.8 and 7.9. The numbers quoted for minima are in screen units, and have no absolute significance.[5]

[5] In figures of this kind, the reduced copy is placed on top of a segment of the original copy. For clarity, figures are usually shown with lines connecting the points.

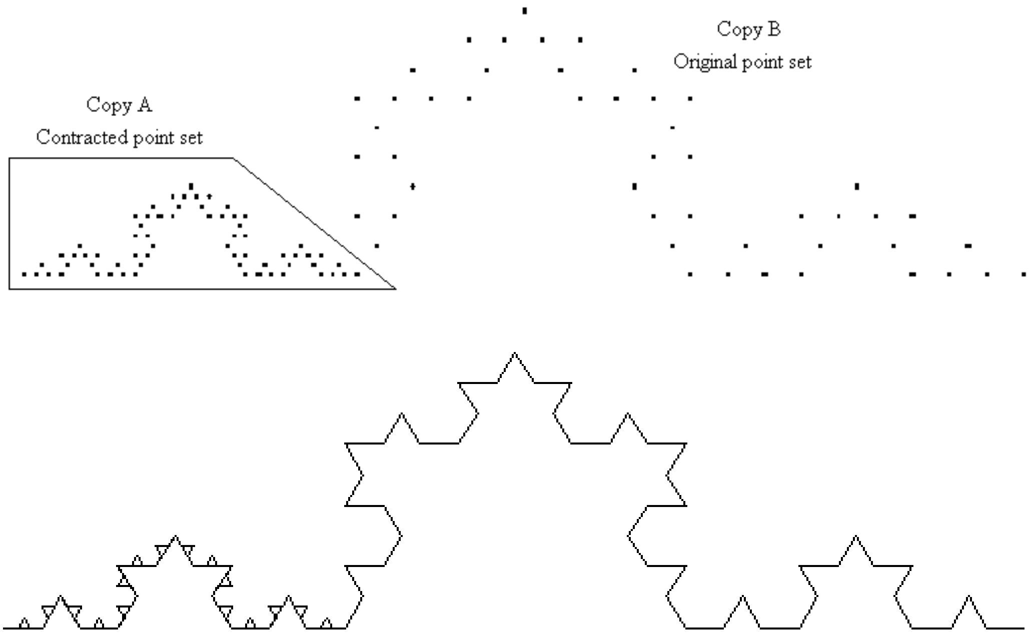


Figure 7.8: Upper panel: point representation of Koch curve at iteration level three. Lower panel: points joined by lines for added clarity. Earliest minimum directed partial Hausdorff distance (of 6.42) for the Koch curve corresponds to a contraction factor of three.

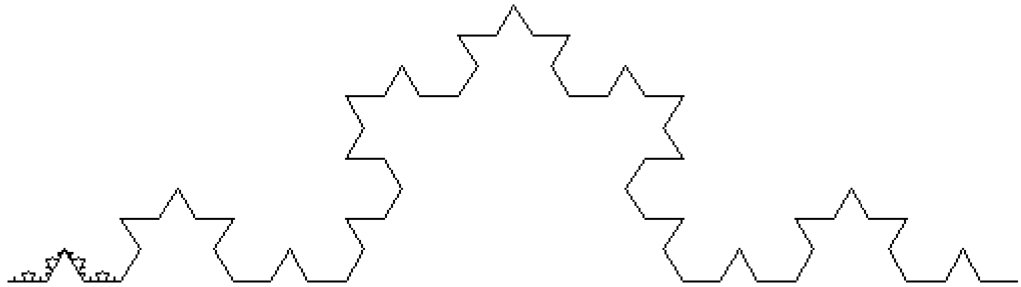


Figure 7.9: Next minimum directed Hausdorff partial distance (still 6.42) for the Koch curve corresponds to a contraction factor of nine.

After enough contractions, structure represented by points occupying a common region is depleted by copy B having only one point—the starting point—in the region common with copy A. See Figure 7.10. This state of affairs grants some arbitrary low value ultimatum for a search.

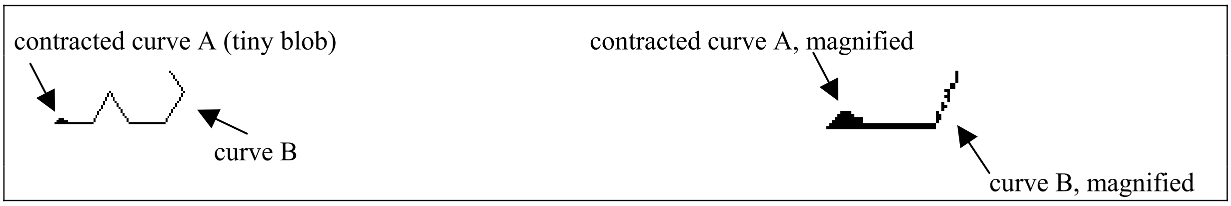


Figure 7.10: The point set represented by the region of curve B that is common with curve A has only one point, while the point set that represents curve A consists of all its points.

The iteration level for a fractal is one more than the number of times the contraction finding phase repeats the minimum h(A→B)p its way to the limit marked by point set oblivion. The original fractal (contraction factor equal to one, therefore zero h(A→B)) represents one iteration, and the number of matches at minimum h(A→B)p represents the others. See Figure 7.11, which shows the Koch curve at iteration level five and a graph of the h(A→B)p during the contraction finding phase. Nonetheless a simple method of calculating iteration level, which involves no additional housekeeping for the model, is left until the number of replacements is known.

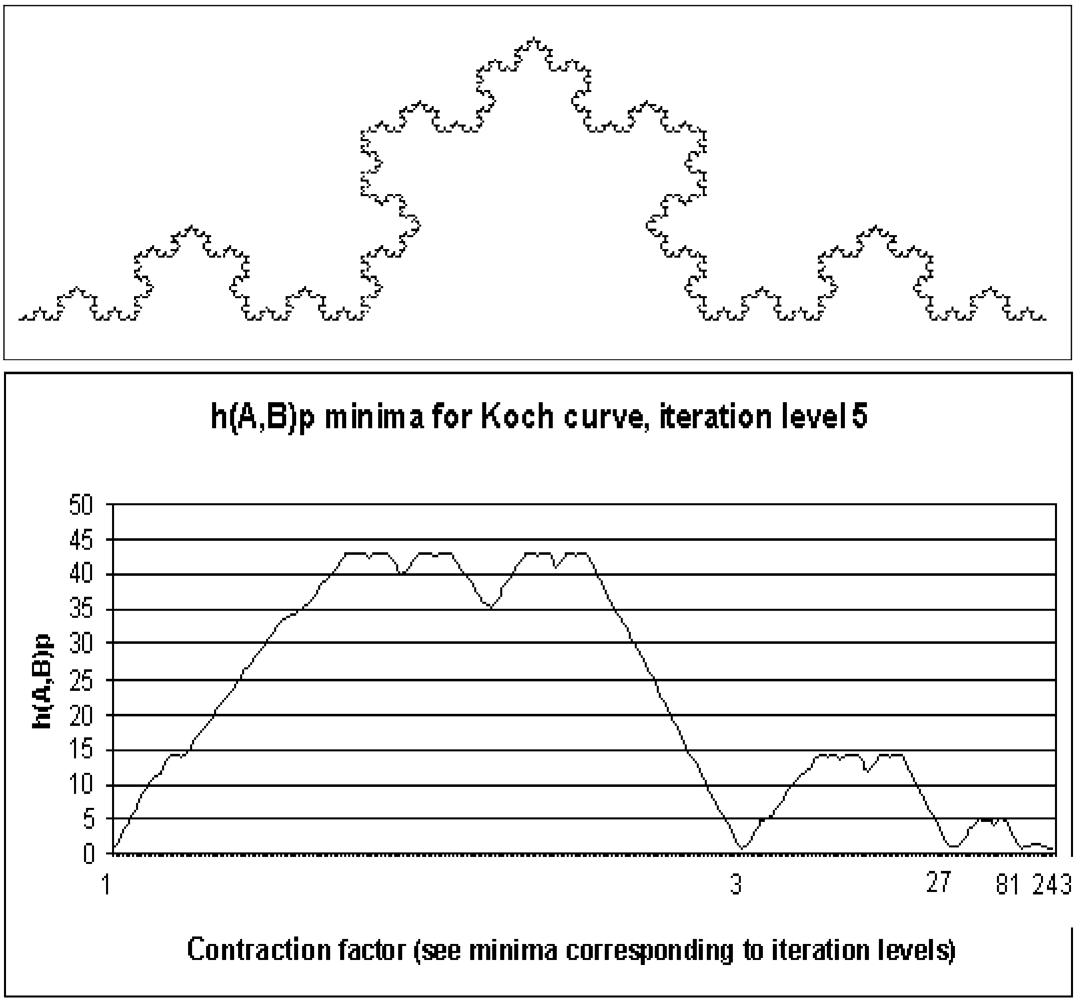
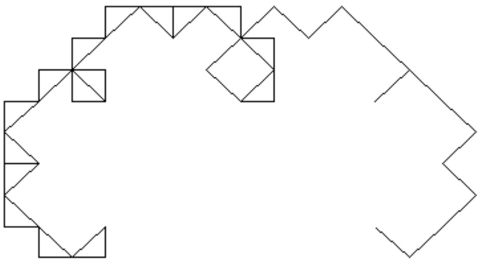
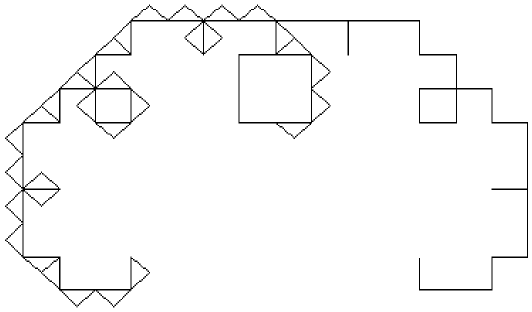


Figure 7.11: Upper panel: Koch curve at iteration level five. Lower panel: graph of the h(A→B)p during the contraction finding phase. Original fractal (contraction factor equals 1) plus four minima indicates 5 iterations.

Sometimes a contraction cannot be established at the horizontal—because the first segment is not horizontal—hence not only contraction ratio, but also angle needs to be searched. Again the search is discrete, with fine tuning by the half split method implemented later if necessary. See Figure 7.12 for an example involving Lévy’s fractal.



Lévy’s fractal, iteration five



Lévy’s fractal, iteration six

Figure 7.12: The first minimum directed partial Hausdorff distances (15.00 and 10.61) for Lévy’s fractal, at levels five and six, correspond to a contraction factor of √2, and the reduced copies are angled at 45° to the horizontal.

By exploiting the likelihood approach, included in the system of constraints, the model looks for an h(A→B)p repetition pattern for any fractal first at the horizontal, then ± 60°, ± 45°, ± 90°, then in ± 5° steps between ± 90° for the others.[6] In getting close to the correct angle, repetition becomes apparent, and the half-split method can be used to further lower h(A→B)p if necessary. Correct repetitions repeat the minimum h(A→B)p value exactly.

The first segment of Levy’s fractal is angled away from the horizontal by 45°, hence its angle is found early in the search sequence. However, the contraction factor for Levy’s fractal falls between two discrete contractions, and is located by a further lowering of h(A→B)p as determined by the half-split method. Such a fractal presents one more problem in that repetition of the minimum h(A→B)p is observed not by contraction at some fixed angle, but by further rotating the contacted copy in increments of the discovered angle.

Even without prioritising angles in the search sequence, as per the likelihood approach, an angle search using the half split method in conjunction with reducing value of partial directed Hausdorff distance proves satisfactory. For Lévy’s fractal the angle search takes the values 0°, ±90°, +45°, for example. Another example, in which a 60° angle is involved, takes the values 0°, ±90°, ±45°, ±67.5°, ±56.25°, ±61.875°, ±59.0625°, ±60.46875°, ±59.765625°, ±60.1171875°, etc., continued iteration of which lowers partial Hausdorff distance to within some arbitrarily small value of the minimum. Additionally it proves sometimes faster to test all contractions at each angle, and other times faster to test all angles

[6] Again, the likelihood approach derives from Helmholtz (1867-1925). Sensory elements are supposed to be organised by unconscious inference into the most likely hypotheses concerning their source (Albert & Hoffman, 1995; Brunswik, 1956; Gregory, 1970).

at each contraction. Returning to the Koch curve by way of example, its parameters can be determined as follows.

Upon finding the proper contraction, the *x*-coordinate value at the beginning of the reduced copy is subtracted from the *x*-coordinate value at the end of the reduced copy. The same is done for the *y*-coordinate. The *x* value obtained is then added to all the reduced copy *x*-coordinate values and, again, the same is done for the *y*-coordinate. This shifts the reduced copy along in a direction-preserving manner, such that its beginning is located at its former endpoint. If there is now a match as indicted by h(A→B)p, then a straight-ahead direction is recorded. If not, then the rotation sequence, using the beginning of the reduced copy as the pivot, is executed in search of a match. The h(A→B)p for which to search is the same as the minimum for the contraction factor. Figure 7.13 shows an angle of 60° to the horizontal found for the Koch curve and an angle of 90° to the horizontal found for the Minkowski fractal.

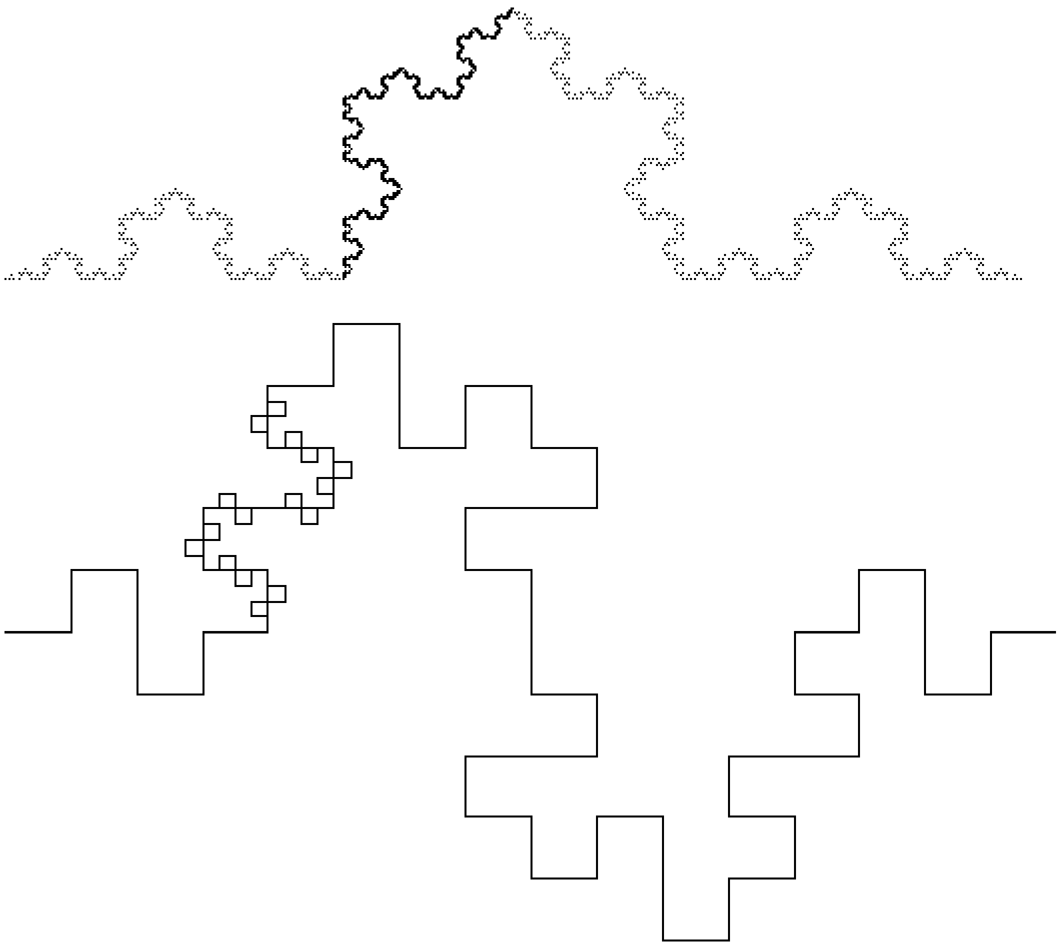


Figure 7.13: Upper panel: the contraction factor for the Koch curve is three, and the angle found is 60°. Distance h(A→B)p = 0.71. Lower panel: the contraction factor for the Minkowski fractal is four, and the angle found is 90°. Distance h(A→B)p = 13.98.

Upon finding the angle, the reduced copy is again shifted in the 'first end goes to second end' direction preserving manner described above. However, upon these subsequent displacements, angles can be matched more efficiently by invoking complements of the discovered angle. This is justified by the observation that regular fractals have complementary positive and negative angles.

Finding number of replacements

Finding the number of replacements is bound up with finding angles. Here, the description is given again; this time with emphasis on counting the number of replacements. This involves positioning the beginning of the reduced copy over the beginning of the original copy and presenting horizontally. Direction is recorded, the match count is incremented, and direction is maintained for the moment. Next, the reduced copy is shifted along in a direction-preserving manner, such that its beginning is located at its former endpoint. If there is now a match, then the match count is incremented. If not, then the rotation sequence, beginning at the horizontal, is again executed, and so on, until the reduced copy moves off the end of the original copy. The terminal match count is then equal to number of replacements.

Of course, the same process applies to fractals with angled initial segments, only the beginning of the reduced copy is positioned over the beginning of the original copy and presented at the discovered angle. From there on its simply a matter of translating and rotating the reduced copy along the original copy. Note that a correct solution, resulting from finding a proper contraction factor, is not a unique solution, only the first encountered in some (limited) series of solutions. A correct solution is also associated with the biggest shift to a minimum directed partial Hausdorff distance in the contraction factor stage, as per Figure 7.11, lower panel.

Other regular fractals

For fractals that involve operations on areas, rather than on lines, angle and replacement searches are slightly different. A 360° rotation of the reduced copy about its centre of mass, (average *x* and *y*-coordinate values) while saving the h(A→B)p at small regular intervals reveals characteristic rotational symmetry. An equilateral triangle, for example, has equal minimum h(A→B)p at 0°, 120°, and 240°. And, of course, this same procedure can be performed before the copy is reduced, if desired. The Serpinski triangle is a case in point, and is created by repeated removal of triangular sections. See Figure 7.14, which also shows a reduced copy with a contraction factor of two, for distance between vertices, at the first minimum.

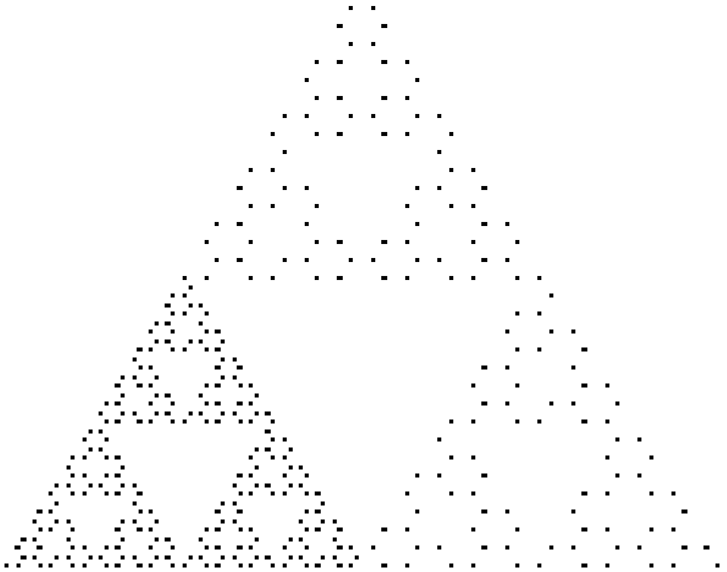


Figure 7.14: Serpinski triangle. The reduced copy (lower left) is translated wholly from sites of minimum h(A→B)p.

The number of replacements is searched by translating the reduced copy wholly from the site of minimum h(A→B)p, in a fashion analogous to that mentioned for line fractals. Upon translation, a check for another minimum, equal to the previous, is done. The idea is to visit all non overlapping sites and check for the minimum at each site. In another scheme the translation strategy uses centre-of-mass of the original copy to initiate translation directions for the reduced copy, which are sustained until limits of the original copy are encountered.

It should be mentioned that the h(A→B)p between a proper contraction of a fractal and its original, would tend to zero as the fractal iteration level tended to infinity. Nonetheless if computers were powerful enough to work effectively with fractals at iteration levels of just a few more than those treated herein, the h(A→B)p would be so close to zero as to cause the search minimum to become delineated by some tiny constant for any such fractal. This would simplify a model of visual perception that features fractal encoding at marginally higher iteration levels. Compare, for example, Figures 7.8 and 7.9, h(A→B)p = 6.42, with Figure 7.13 upper panel at just two more iterations, h(A→B)p = 0.71.

String rewriting

As already stated, characteristics of fractals are typical of nature. Lindenmayer (1968) formalized a description of plant growth, which is appropriate for computer implementation and is now developed as a dialect called *L-systems*. (Again see Lindenmayer and Prusinkiewicz, 1990). L-systems feature the so-called ‘string rewriting’ method for generating biological forms and are ideal for producing fractals.

A string of information, or generator, suitable for reproducing a particular fractal, is recorded by the model during decomposition. A copy of the generator string along with the other essential parameters for the Koch curve shown in Figure 7.11, upper panel, is output by the model.

Contraction factor = 3
Angle = 60^{o}
Number of replacements = 4
Generator = F+F--F+F
Iteration level = 5

- F stands for ‘inscribe with the drawing, or move the generating, tool a fixed distance forward; usually to the right or in an upward direction’. (F draws the replacements.)
- + stands for ‘turn in a direction and magnitude indicated by a positive angle’.
- - stands for ‘turn in a direction and magnitude indicated by a negative angle’.
- No prefix stands for a reference direction, usually to the right in a horizontal direction or upwards in a vertical direction.

Starting in the horizontal direction the drawing tool inscribes over a fixed distance forward, corresponding to the length of a replacement. It then turns left by 60^{o} and again inscribes over the fixed distance forward, after which it turns twice right by 60^{o} (120^{o}) and inscribes over the fixed distance forward. Lastly it turns left by 60^{o} and inscribes over the fixed distance forward. See Figure 7.15, which is a screen shot of the model output for the Koch curve at iteration level five.

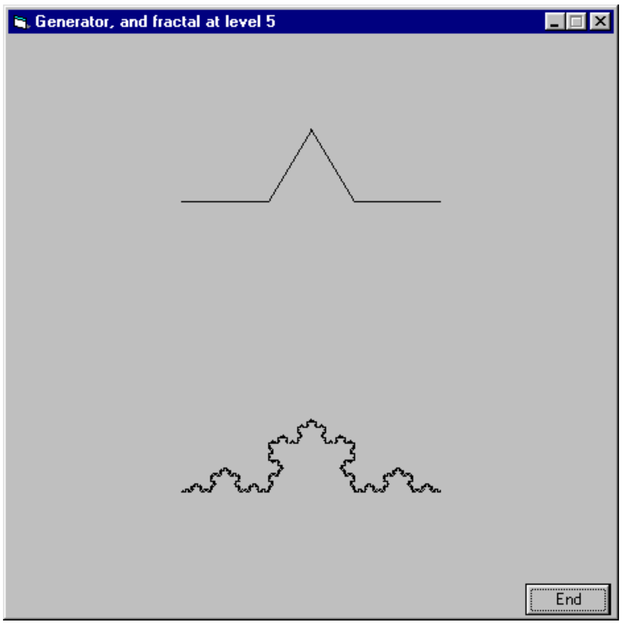


Figure 7.15: A reproduction of the Koch curve generator, and Koch curve at iteration level five.

This might be fine for explaining the generator, but how does it explain the generation of the fractal? This is where the notion of recursion is important. For each iteration, each F is simply replaced by the generator string. The axiom, or zero iteration level string, is F, or a *straight line.* The generator, or first iteration level, string replacement (for F) is F+F--F+F. The second iteration level string replacement (for F+F--F+F) is F+F--F+F + F+F--F+F+ F+F--F+F+ F+F--F+F, and so on.

Now iteration level must be proportional to the number of Fs, which in turn is equal to the size of the array required to hold the point set. A simple method, alluded to earlier, for finding iteration level with no additional housekeeping involves the number of replacements. It turns out that the number, R, of replacements raised to a power equal to the iteration level, I, equals the number, N, of Fs.

$$R^I = N$$

therefore

$$I = \log (N) / \log (R)$$

Iteration level equals log of the number of Fs divided by log of the number of replacements. And fractal dimension can be calculated from the parameters too.[7]

Fractal dimension = log (number of replacements) / log (contraction factor).

(See Voss, 1988, for example.) The term ‘fractal dimension’ comes from ‘fractional dimension’. The fractal dimension of an object is represented by a nonnegative real number instead of a nonnegative integer, which ordinarily applies to a point (zero-dimensional), line (one-dimensional), plane (two-dimensional), or solid (three-dimensional), but need not stop

[7] Logarithms of any base can be used in such a situation because they can be calculated for any number by dividing the natural logarithm of the number by the natural logarithm of the base. Consider $\log_x(N) = \ln(N) / \ln(x)$ and $\log_x(R) = \ln(R) / \ln(x)$, then $I = \ln(N) / \ln(x) \times \ln(x) / \ln(R) = \ln(N) / \ln(R)$, no matter what value the base x might have.

there. I state ‘nonnegative’, and include whole numbers, because regular objects can be generated by appropriate constraints on a suitable fractal generator (see Pentland, 1988, for example). Hence regular objects can be considered special cases of fractal objects.

The fractal dimension of the Koch curve equals 1.26, which is more than one dimensional as per a line, but less than two dimensional as per an area. Of course the replacements have to fit into the Koch curve without making it shrink or grow, hence there needs to be a unique contraction, or reduction, factor. (In modelling natural forms a contraction can allow for change in size, in which case it needs to converge to some limit.)

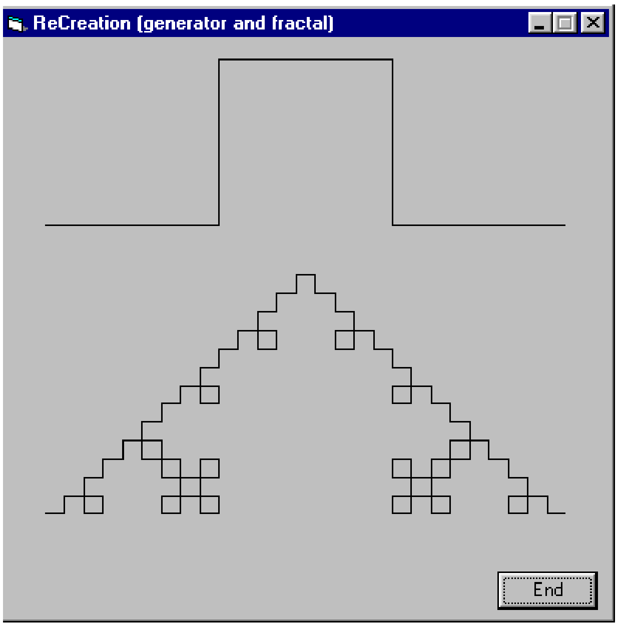


Figure 7.16: Reproduction of Koch quadratic generator, and Koch quadratic curve at iteration level three. Generator = F+F-F-F+F.

For another example—this time a Koch quadratic fractal—see Figure 7.16. For yet another example, see Figure 7.17. It has a starting, and therefore ending, direction away from the horizontal and vertical. This corresponds to ‘turn left by the prescribed angle before drawing the first line and after drawing the last line’. The fractal also has no change in direction for some number of consecutive segments. This is shown by the contiguous lines, i.e. two consecutive lines in the same direction.

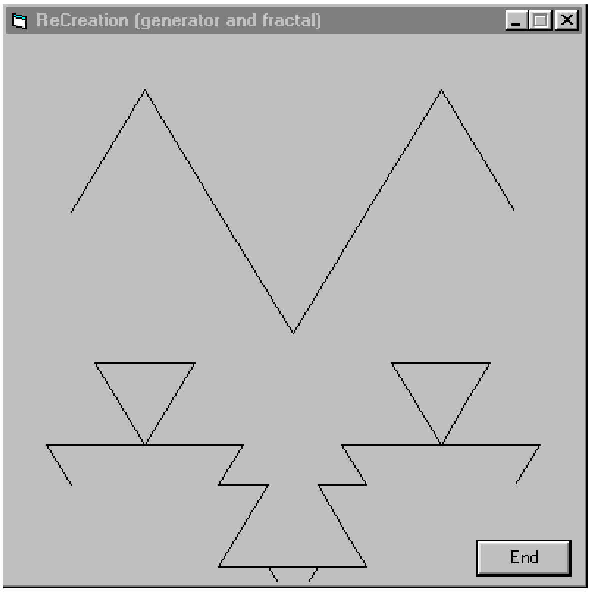


Figure 7.17: Reproduction of experimental curve generator, and experimental curve at iteration level two. Generator = +F--FF++FF--F+.

Having described a major part of the methodology for decomposing different fractals, it can now be declared that an initial test for reflection symmetry on a point set can shorten the decomposition process. If a point set is symmetric about a central axis—relative to the point set—then the process need only be confined to one half, and results for the half extended to the whole by way of complementing appropriate parameter values.

The test for reflection symmetry is fast and simple. Array contents for the coordinate of interest are indexed out to a maximum of half the array dimension value, integer truncated, and the same index is modified for the complementary offset, i.e. array dimension value minus index. Contents of the pairs of array locations so indexed are then compared (one element of a pair with the other element of the pair) for effective equality of coordinate values until either an unequal value or terminal index value is reached, without such inequality. If terminal index value is reached without inequality, then reflection symmetry exists about the complementary coordinate.

Random fractals, regular fractals, and fractal decomposition without Hausdorff distance

The above-described methodology, using Hausdorff distance, does not yield decomposition of random fractals. Before detailing why, see Figure 7.18, which shows a sequence of randomised iterations; in this instance for a Koch curve. A random fractal models the general appearance or essence of a type, not the details of a particular individual. The Koch generator was randomised on a 50-50 basis, i.e. each angled segment had an even chance for left or right displacement.[8]

[8] The randomising split does not have to be 50-50.

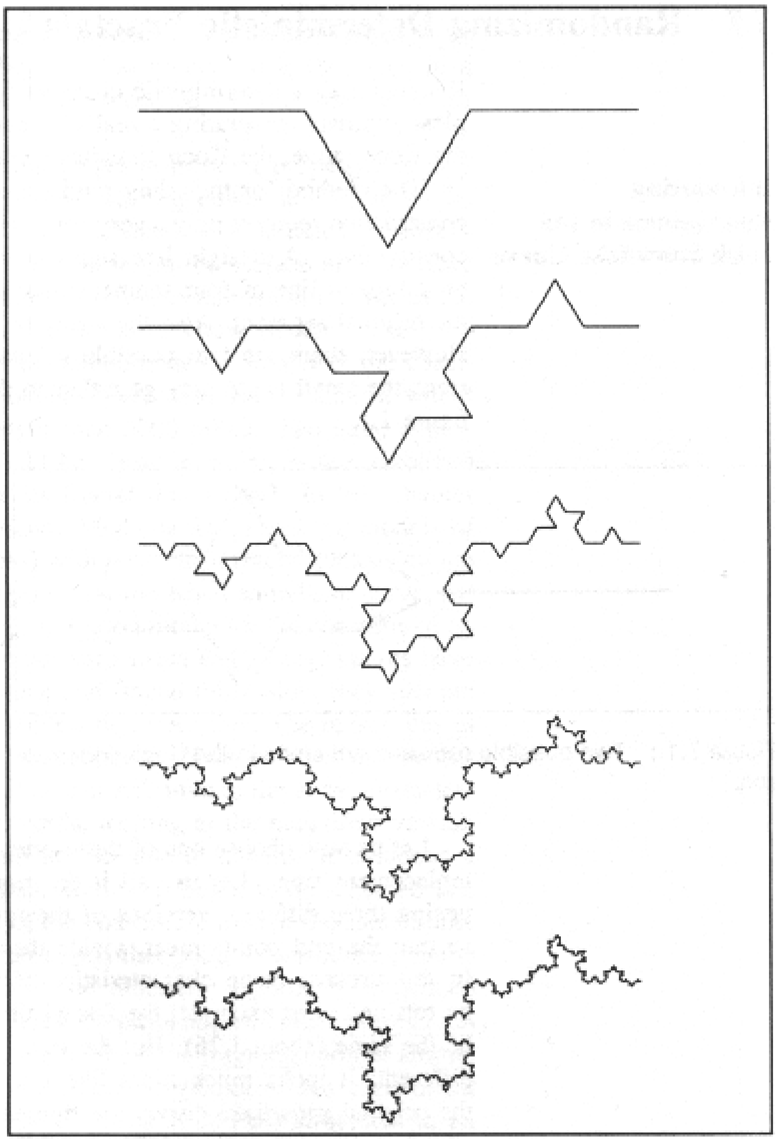


Figure 7.18: Series of randomised Koch curve iterations.

A correctly contracted copy of a random fractal does not necessarily lie snugly along a segment of the original. See Figure 7.19, which shows a random Koch curve at iteration level four. In this instance the mismatch of the contracted copy with the first segment of the original copy is evident. If the contracted copy were to be advanced from segment to segment of the original copy, then note also that the mismatch with the second segment would be greater than the mismatch with either of the last two segments. The crux of the problem is that there is no *consistent* minimum Hausdorff distance mismatch.

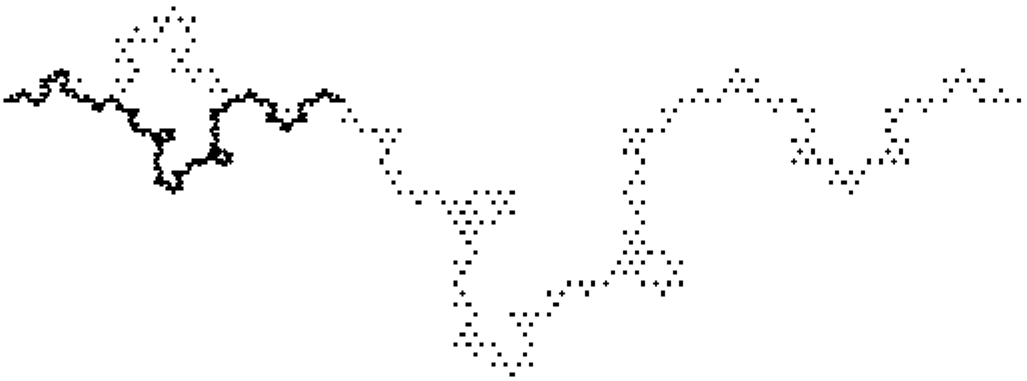


Figure 7.19: Variable mismatch of correctly contracted copy.

Discarding Hausdorff distance

The matching problem can be overcome by discarding Hausdorff distance for a scheme that does not include matching for *shape.*[9] The shape matching inherent in Hausdorff distance precludes the flexibility required for decomposition of random fractals.

The alternative matching scheme exploits 'invariance under randomisation'. Examination of Figure 7.19 reveals hierarchies of points that do not deviate with respect to copies of the fractal related to the orders of recursion in which they are embedded. For any line fractal, such invariance points are used in exact—effectively zero distance—matching. Matching for form has been discarded and matching for common elements across otherwise random elements is employed. And since regular fractals can be considered specific instances of random fractals, then regular fractals can be decomposed by the same method.

Invariance points for a copy lie along a straight line. The start and end points are included, and define a straight line along which intermediate invariance points lie. Intermediate points are identified by a maximum number of matches, which includes matching for endpoints, during the contraction finding phase.

Finding the proper contraction is similar in principle to that outlined for Hausdorff matching. Contractions are tried, and the proper contraction is the *first* one in which a maximum number of invariance points match. For the example shown in Figure 7.19, eight invariance points match at a contraction factor of three. On the way to this contraction there were no other instances in which as many as eight points matched. In continuing with contractions the same pattern is repeated, ending in a contraction factor of nine; then 27, and so on, until the limit, marked by point set oblivion, is reached.

Some fractals have angled initial segments, and for these there may be no cyclic trend along the horizontal. If trying contractions along the horizontal yields no success, then trying angles along with contractions is expedient. If trying contractions along the horizontal does yield success, as per the Koch curve in this example, then the angle is found by first advancing the contracted copy by an increment of its length along the original copy. The contracted copy is then rotated from the left until its endpoint matches the corresponding point belonging to the segment of the original copy. The contracted copy is then returned to the first segment in readiness for string rewriting and finding number of replacements. If the first segment is angled then, of course, the contracted copy is already there. Note that only the invariance point coincident with the endpoint of a proper contraction is now utilized.

Finding the number of replacements is just a matter of advancing the correctly contracted copy in increments of its length along the original copy, and counting the number

[9] Even so, the alternative scheme could be considered a simplifying instance of Hausdorff distance.

of matches between its endpoint and the corresponding segment points of the original copy. Since the match with the first segment implicitly defines the first replacement, the matches are counted from two onwards. (Note: points of conjunction are members of a set of *invariant* points.) As the contracted copy is advanced, the Fs, as increments, along with respective interspersed directions, are recorded by the string rewriting system.

Statistical similarity and form

To get a statistically similar reproduction of a random fractal, and one that is necessarily similar in outward form, involves a somewhat philosophical consideration that concerns the scale at which things are 'meaningfully' apprehended. Perception of the form of a particular tree, for example, might be dictated by the character of the trunk and just a few major branches. Progressively finer (and progressively more numerous and similar) details are (necessarily) regarded with increasingly less view to distinction. They become 'touched upon' in a general and encompassing manner.

In this sense overall shape of a random fractal needs to be considered for the purpose of reproduction, but finer detail can be random. This is accomplished by making one or more 'structured iterations'—as opposed to 'random iterations'—of the fractal generator employed in reproduction. The structured iterations deviate the larger scale angled segments in the same directions as those of the original fractal. The number of structured iterations depends upon the scale of readily noticed detail, and is in the range one to three; typically two. Since the string rewriting system records the direction of angles in the generator, and can do the same for another order, this presents little difficulty.

Line fractals that enclose areas

Several line fractals can be generated at angles to one another such that they form a closed loop; sometimes called an 'island'. Random fractals are sometimes generated in loops or some part thereof for the purpose of emulating natural features, such as coastlines. See Figure 7.20, which is, again, by way of example, based on the Koch curve. It consists of three random Koch curves, orientated at 120^o to one another, forming a closed loop. Figure 7.21 shows a statistically, and form-wise, similar reproduction of the fractal depicted in Figure 7.20.

In the case in which regular line fractals form such a loop, one of the line fractals is not only tested for reflection symmetry as described on page 136, but one copy of the loop is rotated on its centre through 360^o with respect to the other. This is performed in increments related to the number of dots in a representation, upon which repetition of directed Hausdorff distance reveals degree of rotational symmetry (threefold, fourfold, or whatever). Along with these tests for symmetry, only one half of one line fractal need be decomposed in order to get enough information to reproduce the entire loop.

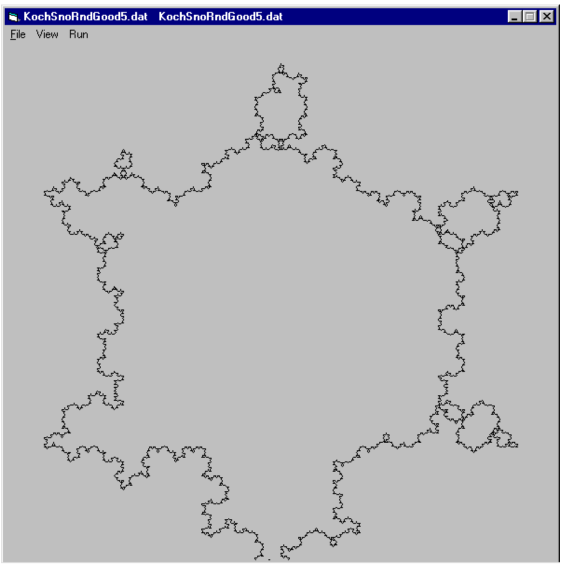


Figure 7.20: Random Koch island.

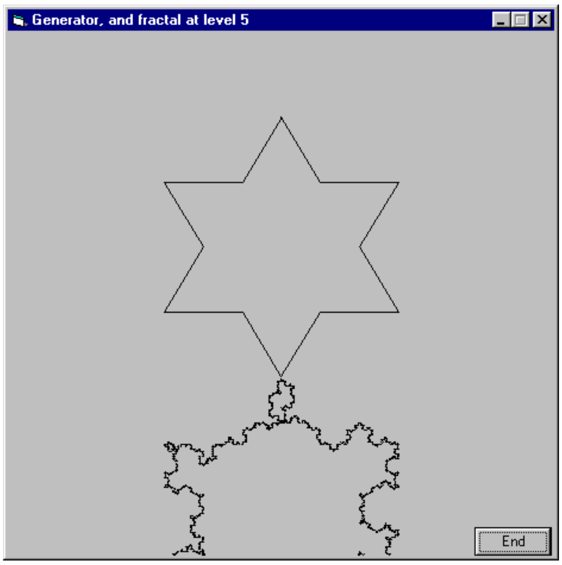


Figure 7.21: Reproduction of generator and random Koch island. (Note that any of the angled segments of the generator could point inwards, but the randomising process resulted in the option where all the angled segments point outwards.)

Brownian fractals

Brownian motion—after biologist Robert Brown (1773-1858)—is a fractal phenomenon. (In one variable, it constitutes the simplest random fractal.) Fluid or gas molecules can continually bump against much bigger particles. Given the spread of molecular kinetic energies, the bigger particles exhibit erratic, unpredictable movements. A range of particle sizes, or scales, can be observed, as if through a microscope, and the same erratic movements are evident. The principle of self-similarity across scales holds in a probabilistic manner.

The simplest Brownian fractal can be modelled by a 'random walk'. For a random walk there is an even chance of taking a step, of Gaussian distributed random length, in one of two complementary directions. Upon taking the step, there is again an even chance and so on. This results in a curve that ultimately wanders about the mean. Larger deviations sit on smaller deviations sit on smaller deviations. If the curve is stretched or shrunk in the *x* direction, then provided it is scaled correctly in the *y* direction the curve exhibits scaling invariance. With regard to roughness or smoothness, any part looks like the whole.

If the curve is scaled up by 2 in the *x* direction then scaling up by √2 in the *y* direction, i.e. by $2^{1/2}$, preserves scaling invariance. Exponents greater than or equal to zero but less that $^1/_2$ result in the curve being too smooth, and exponents greater than $^1/_2$ result in the curve being too bumpy.

This raises the question: 'Are there such probabilistic fractals for which other exponents preserve scaling invariance?' The answer is 'yes'. Brownian fractals can be divided into three categories: exponents < $^1/_2$, exponent = $^1/_2$, and exponents > $^1/_2$. The case for which the exponent = $^1/_2$ represents ordinary Brownian motion, which has independent increments (i.e. their correlation is zero). The other exponents represent 'fractional' Brownian motion. For exponents < $^1/_2$ there are positive correlations between increments (i.e. if *y* increases for some opening *x* then it tends to increase for some subsequent *x*, and likewise for decrease). For exponents > $^1/_2$ there are negative correlations between increments and the curves oscillate more erratically. The exponents are called 'Hurst exponents', usually written as *H*. *H* is related to fractal dimension by

$$H = \text{Euclidean dimension} + 1 - \text{fractal dimension}$$

where Euclidean dimension equals 0 for a point, 1 for a line, and 2 for a surface. Hence fractal dimension for a line fractal equals $2 - H$. (See Pentland, 1988;Voss, 1988, for example.)

By locating the various Brownian fractals about different initiators (averages of different geometric shapes) realistic profiles of assorted landscapes can be created. For instance, if a fractional Brownian fractal with an exponent of 0.7 is located about an inverted V of an appropriate height-to-width ratio, then a realistic profile of a mountain is created. For a realistic profile of distant hills, the exponent needs to be less and the height-to-width ratio of the inverted V needs to be less. See Figures 7.22 and 7.23, in which 'Level', on a 10-point scale, is linearly proportional to the exponent and 'Scale Factor' is of the order of height-to-width ratio. (By invoking a little more technical detail, realistic three-dimensional projections can be also produced.)

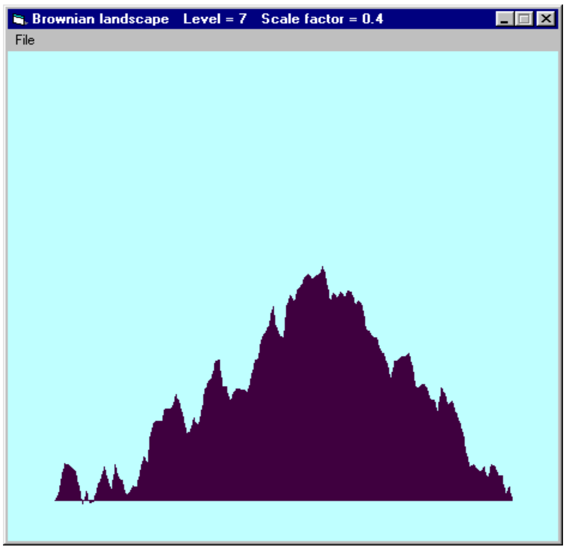

Brownian landscape Level = 7 Scale factor = 0.4

Figure 7.22: Profile of a jagged mountain.

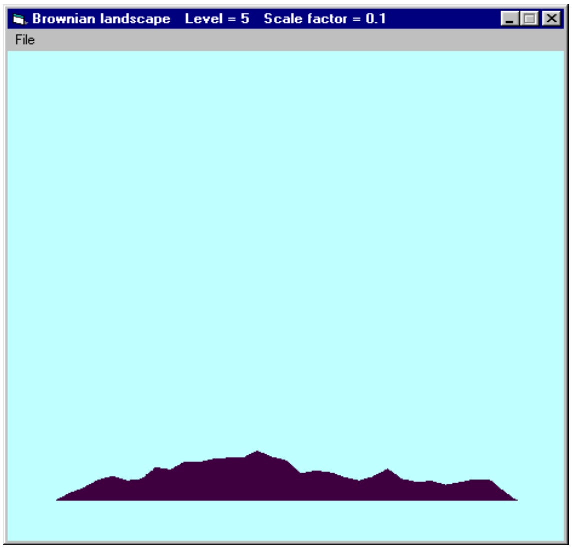

Brownian landscape Level = 5 Scale factor = 0.1

Figure 7.23: Profile of distant hills, possibly worn smooth by erosion.

The model can decompose any such probabilistic fractal by first matching for the correct initiator. Where the initiator is an inverted V, this is done by incrementally increasing the height-to-width ratio and matching for minimum h(A→B) to the landscape profile. A test profile is then incremented by increasing its exponent, and the profile for which h(A→B) is closest to h(B→A) is chosen. (In other words the profile for which the landscape-to-test profile is closest to the test-to-landscape profile, is chosen.) As an alternative to Hausdorff distance matching, the test profile can be also matched with the landscape profile for variance after stripping variance due to the initiator.

More generally, just two parameters are required to reproduce a Brownian form. These initialise a simple recursive process, which results in a form that is statistically similar to the original. The first parameter defines a smooth geometric outline of best fit, and the second parameter defines a degree of roughness for the outline. The transformations used in this section of the model are best described as 'scaling transformations'.

Variable contraction size

In modelling a form for which contractions must allow growth or decay, a change in size of contracted copies along with a convergence factor is needed. In the case of rudimentary growth, for example, successive contracted copies increase the size of a form while converging on some limit, whereupon there is no longer any increase in size. The Koch curve could be made to grow something like this if the initial four replacements were contracted by 1/3 × 1.06, say, and the second four by 1/3 × 1.03, and the third four by 1/3 × 1.01, and the fourth four by 1/3 × 1, whereupon there is no longer an increase in size. Hence any constant size fractal is just a special case in which the multiplier is 1.

Summary and conclusions

By using some form of Hausdorff distance to guide the matching process between original and transformed arrays, the model can decompose a range of fractal objects. This is achieved by employing discrete affine transformations. Affine transformations (which correspond to the changes in appearance of a planar object as it translates and rotates rigidly in three dimensions) preserve straight lines and ensure that the regenerated fractal will not grow explosively or shrink to a point, but will approach a defined, invariant image. The use of discrete transformations (i.e., with values chosen at defined intervals) keeps the number of transformations tested to manageable proportions, and can be justified by the restricted number of values typically found in both artificial and natural fractal structures.

The analysis of fractal objects reflects the same contract-multiply-transform steps that are used to generate such objects. Specifically, the program that implements the model begins by finding the contraction factor used to generate the fractal object in question. Two copies, in the form of point sets, are made of the object. One of these is superposed and repeatedly contracted with respect to the other. At each contraction, the sets are searched for minimum partial Hausdorff distance from the contracted set to the original set.

Identification of transformation angle(s) is performed either during establishment of the proper contraction or afterwards, depending on whether or not the first segment is angled away from a reference direction. This is done by rotating and translating the contracted copy along segments of the original copy and matching for the same minimum partial Hausdorff distance. Hence, as for contractions, the transformations are selected in a constrained manner that keeps the search process within comfortable bounds.

One method for identifying the number of replacements depends on counting repetitions of the same minimum partial Hausdorff distance during reduced copy translations and rotations to the discovered angle(s). The reduced copy is displaced in steps along the original in a ‘first end goes to second end’ manner. Once all the parameters have been identified in these ways, the original fractal can then be reproduced.

Using these techniques, the program can successfully decompose a variety of both regular and probabilistic geometric fractals, whether they be simple line shapes or closed figures. The program can also decompose random fractals into their initiating seed component and the parameters controlling the random walk process by which such fractals are generated.

In the case of regular geometric fractals, the program not only decomposes the original image but is capable of generating an exact replica of it. At low levels of recursion, this is not an implausible model of the way in which human visual perception might operate. However, at higher levels of recursion, it may be questioned whether the human visual system is capable of such detailed, accurate perception. Of course, it is always possible for an observer to focus more closely on any part of a pattern and to see it ‘veridically’. It is also not difficult to modify the model so as to include, for example, a more probabilistic execution of the generation process, especially at the higher orders of recursion. In principle, therefore, the model can be developed to simulate human performance with a range of degrees of fidelity. (One problem is that there are no known studies of such veridical perception, particularly of geometric fractal objects.)

Finally, it should be emphasized that the present account of fractal decomposition is intended principally as a ‘toy’ model, to explore a working version of a generative transformation model. Although human visual perception may arguably incorporate some of its essential ingredients, it would be naïve to suppose that the present theory represents much more than a demonstration that this approach is feasible.

A test of the usefulness of the model lies in its potential for generalization to a wider range of different situations and tasks. Therefore the following chapters are concerned with the potential of the generative transformational approach to account for different perceptual phenomena, including the perception of non-fractal structure, the perception of motion, and certain dynamic or adaptive effects.

Chapter 8: Static Pattern Detection

Brief summary of chapter

Relationships based on proximity and relative frequency of picture elements are used in a reconciliation process. In this chapter, such relationships are spatially disposed. Temporally disposed relationships are addressed in the next chapter. Sets of distances—each set containing distances from a different point in a pattern set to all points in another pattern set—are produced during calculation of Hausdorff distance. It turns out that a pair of overlying, identical pattern sets yields sets of distances replete with structure. Glass translation patterns, for example, have consistent distances from original to transformed members of pairs of elements across all sets of distances, whereas combinations of distances between pairs of elements lack such consistency. For Glass rotations and dilations, as well as arcs and circles, for example, consistent distances are identified by a transformation search.

Frequencies of occurrence of orders of regularities between sets of distances are exploited in identifying any precise geometric figure placed anywhere in a field of noise (random dots). Moreover hand-composed figures constituted of lines and arcs of points somewhat irregularly spaced, and placed anywhere in a field of noise, are identified. A pattern set that is rejected by the Glass pattern search could be wholly or partially random. If wholly random, then frequencies of occurrence of 'regularities' are not significant by one measure or another. If they are significant, then a set might contain structure. Regular distances are probed in ascending order of magnitude and descending order of frequency for point-to-point connection, until a connected object is yielded or a threshold for distance probing is reached. Detection of less regular patterns is facilitated by examining coordinate sets of half-distance points for maximum frequency of visits to a limited region (of size related to distance spread) and then working back to find what distances produced them. (Precise patterns can be also detected using this principle, but the half-distance maximum visit region may be consigned to a point instead of an area.) Having stated the case for consistent distances, with little extra overhead corresponding orientations are frequently obtained and used in conjunction with the distances.

In this and the next chapter, a variety of pattern types and situations are employed so as to demonstrate the versatility of what amounts to just a handful of underlying principles. These are summarized generally in a model that uses information about relative positions of image elements, where relationships of economy that prove most suitable are utilized to select transformations that minimize dissimilarity between arrays, and in which the trajectories of the selected transformations then correspond to perceived structure. Boundary conditions are determined by the amount of slack, or bin-width, allowed for consistencies, and by the outcome of one or another statistical analysis performed on a pattern. For example, a limit on the lengths of structural links might be defined by Ripley's *K* function. The work outlined in this chapter, with the exception of that indicated of other researchers, is original. Particularly, the greyscale charts and pattern detection methods are original.

Background

Among the patterns used to test the generative transformation model, Glass patterns are of special theoretical interest. Since Glass patterns are constituted of some elementary transformation on random elements they have low order structure, which leaves the transformed elements proximate to the original. To perceive structure in such textures it has seemed necessary to suppose the visual system must solve some form of massive correspondence problem, by determining which dots go with which.

Perhaps the simplest model proposed to explain how the visual system might detect structure in Glass patterns is the autocorrelation model of Maloney, Mitchison, and Barlow

(1987), which, in turn, is based on the model of Barlow and Reeves (1979) for detection of mirror symmetry in random dot displays. According to the autocorrelation model, the stimulus is examined for occurrence of signal pairs of points related by a common distance and/or orientation. This involves comparing all possible pairs of points and deciding whether the proportion of signal to noise pairs departs substantially from that expected if all points were distributed at random.

The autocorrelation model works well by producing a roughly linear relationship between empirical measures for Glass patterns and theoretical measures; the latter based upon signal detection theory (Green & Swets, 1966). However, if paired elements in Glass patterns are perturbed by random angles, for example, then global structure is lost, but the model of Maloney et al. (1987) cannot distinguish this situation from that prior to perturbation.

The neural filter model (see Wilson & Wilkinson, 1998; Wilson, Wilkinson, & Assad, 1997), with its reliance on multiple, richly distributed filters, each specialized for detection of some specific feature, has encountered difficulties reconciling different kinds of Glass patterns.

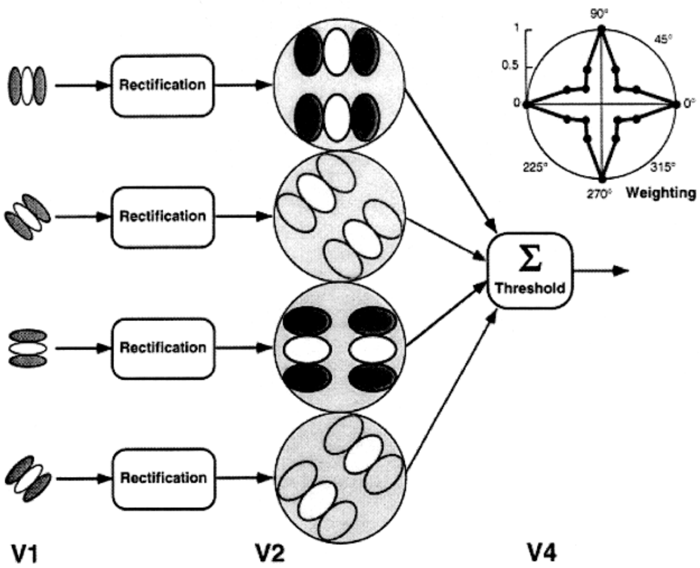


Figure 8.1: Neural model for radial Glass pattern detection consists of 8 parallel pathways (only 4 are shown). Each pathway involves oriented filtering, rectification, and second-stage oriented filtering at the same orientation. Excitatory zones of each filter are shown in white and inhibitory zones in darker shades. Oblique pathways receive weaker weighting than horizontal and vertical pathways. The weighting function is shown at upper right. Outputs from second-stage filters are summed and passed through a threshold at the final stage. (From Wilson & Wilkinson, 1998, p. 2940.)

Figure 8.1 shows an example of one such recent model, proposed by Wilson and Wilkinson (1998), to account for the perception of radial Glass patterns. The first stage consists of eight types of orientation filter (only 4 are shown), differing by multiples of 22.5°, which includes vertical and horizontal orientations. Outputs from these units are combined with responses by sixteen second stage filters that have the same orientations as the first stage but are maximally sensitive to displacements of a certain magnitude along the same orientation as the corresponding first stage units. The weighted output from these second stage filters is then summed and compared with a threshold value to determine whether a radial pattern has been detected.

The general outline of the neural filter model of Wilson et al. is similar to those of many theoretical approaches to low-level vision. Not unreasonably, these approaches stem from research in neurophysiology (Hubel & Wiesel, 1959, for example), which suggests a theory in which feature detector cells are tuned to respond to fixed attributes, not attributes scaled to pattern configuration. The neural filter model makes specific assumptions about scale of dot separations and orientations to which filters are responsive. Although the assumptions may be justified within the context of a particular study, the human visual system is demonstrably sensitive to Glass pattern structure that varies over a wide range of different scales from one stimulus to the next or even within a single stimulus (Dakin, 1997; Prazdny, 1986. See also Preiss & Vickers, 2005, Appendix A).

At the least, the neural filter model requires the addition of a principled account of the setting of filter sizes that can be demonstrated to apply across a realistically wide range of scales. Meanwhile, the necessity for filter sizes to be hand-tuned to the characteristics of each different type of display severely restricts the generality of the model and adds to its complexity, even supposing that a suitably principled account of filter setting can be supplied.

An underlying difficulty with this neural approach in terms of hierarchies of filtering units can be appreciated by reference to the model for detection of radial Glass structure, for example. The model is constructed so as to detect other, similarly oriented dipoles, lying along the same direction as a dipole detected by a first-stage filter. Beyond that immediate neighbourhood, however, the second-stage unit is unresponsive. This means that we could proceed through the radial array, delimiting all such locally similar sets of dipoles and redistribute such sets randomly to make a new array. Such an array would appear to have structure (in the form of a patchwork of sets of dipoles with locally similar and collinear orientations). However, it would be perceptually and geometrically quite dissimilar from a radial pattern (cf. the random perturbation of paired elements for the model of Maloney et al., 1987, but on a larger spatial scale.) In other words, the neural model of Wilson et al. (1997) is insensitive to the global structure in Glass patterns.

The generative transformation model obviates these problems by looking for significant frequencies of point-to-point distance and orientation relationships that define a suitable transformation, which fits patterns within some tolerance based on variance of the defining relationships. Essentially the model looks for distinguishing frequencies of identifying consistencies. Orientation of displacement is inherent in the Pythagorean treatment of *xy*-coordinates for calculating distance, and is given by arctangent (*x* distance / *y* distance) with the exception of *y* distance equal to zero, in which case orientation of displacement is horizontal. Indeed, a common set of transformation routines applies to a variety of spatial and temporal pattern types, not just to Glass patterns alone, and these are routines that implement the affine group of transformations. The strategy for detection of Glass patterns forms a substantial basis for detection of other types of pattern.

Contribution of nearest neighbour to the phenomena of Glass patterns

The following is a step-by-step exposition, which indicates the extent of utility in the nearest neighbour approach to discriminating between Glass patterns and random point distributions; the latter more commonly called noise. First, the relationship between *Hausdorff distance* and nearest neighbour is examined.

Consider that a form contains *N* randomly distributed points belonging to a set A, and *N* randomly distributed points belonging to a set B. Then a set of distances radiates from one point in set A to every point in set B, and there is a *least distance (nearest neighbour)* within the set. Another set of distances radiates from another point in set A to every point in set B, and there is another *least distance (nearest neighbour)* within this set; and so on for each point of set A. This forms a set of *least distances (nearest neighbours),* and there is a *largest* *distance (largest nearest neighbour)* within the set. This is the *directed Hausdorff distance* from set A to set B, denoted h(A→B).

A set of distances also radiates from one point in set B to every point in set A, and there is a *least distance (nearest neighbour)* within the set. Another set of distances radiates from another point in set B to every point in set A, and there is another *least distance (nearest neighbour)* within this set; and so on for each point of set B. This forms another set of *least distances (nearest neighbours),* and there is a *largest distance (largest nearest neighbour)* within this set. This is the *directed Hausdorff distance* from set B to set A, denoted h(B→A).

Now there is a set—made up of just two distances in this case—of largest least distances (largest nearest neighbours), and there is a largest distance (largest of the two largest nearest neighbours) within this set. This distance is the *Hausdorff distance,* denoted H(A↔B).

If there are *N* randomly distributed points in a set A and *N* randomly distributed points in a set B, then the point density on a form is half that for which there are *2N* randomly distributed points in set A and *2N* randomly distributed points in set B. The greater the point density, the greater the chance of tighter point proximity. Tighter point proximity corresponds to smaller *least distances* between sets and therefore to smaller *largest least distances,* which corresponds to a smaller *Hausdorff distance.*

Consider that a form contains *N/2* randomly distributed points paired with another *N/2* points under some transformation, with distances between transformation partners kept relatively small, as per any Glass pattern, and these constitute one set. The form also contains a further *N/2* randomly distributed points paired with another *N/2* points under the same transformation, and these constitute another set. The form has *2N* points, but because of pair-wise correspondences the inter-point distance distribution between sets is of the order for that belonging to a form containing *N* randomly distributed points. A form with *2N* point density under Glass patterning can exhibit (larger) inter-point distances of an order for that of a random distribution at half the point density. And similar reasoning can be extended to triplets of points and so on. In statistical terms, *Hausdorff distance* is larger for Glass patterns than for the same numbers of randomly distributed points; simply because the larger nearest neighbour gaps are larger between points of the patterned sets.

Mean distance for a set of *least distances* is related to directed Hausdorff distance, h(A→B) or h(B→A), and these are related to Hausdorff distance, H(A↔B). However, Hausdorff distance, directed or otherwise, can represent an outlier, particularly where not many points are concerned; hence mean distance provides a better measure by which to discriminate Glass patterns. (Nonetheless the representativeness embodied in a reasonable number of points ensures that Hausdorff distance correlates well with mean distance.)

There are *two important convergences,* in which the *mean* distances for sets of least distances involving Glass patterns approach those involving random distributions. The convergences are bracketed by the upper and lower plots of Figure 8.2 as follows.

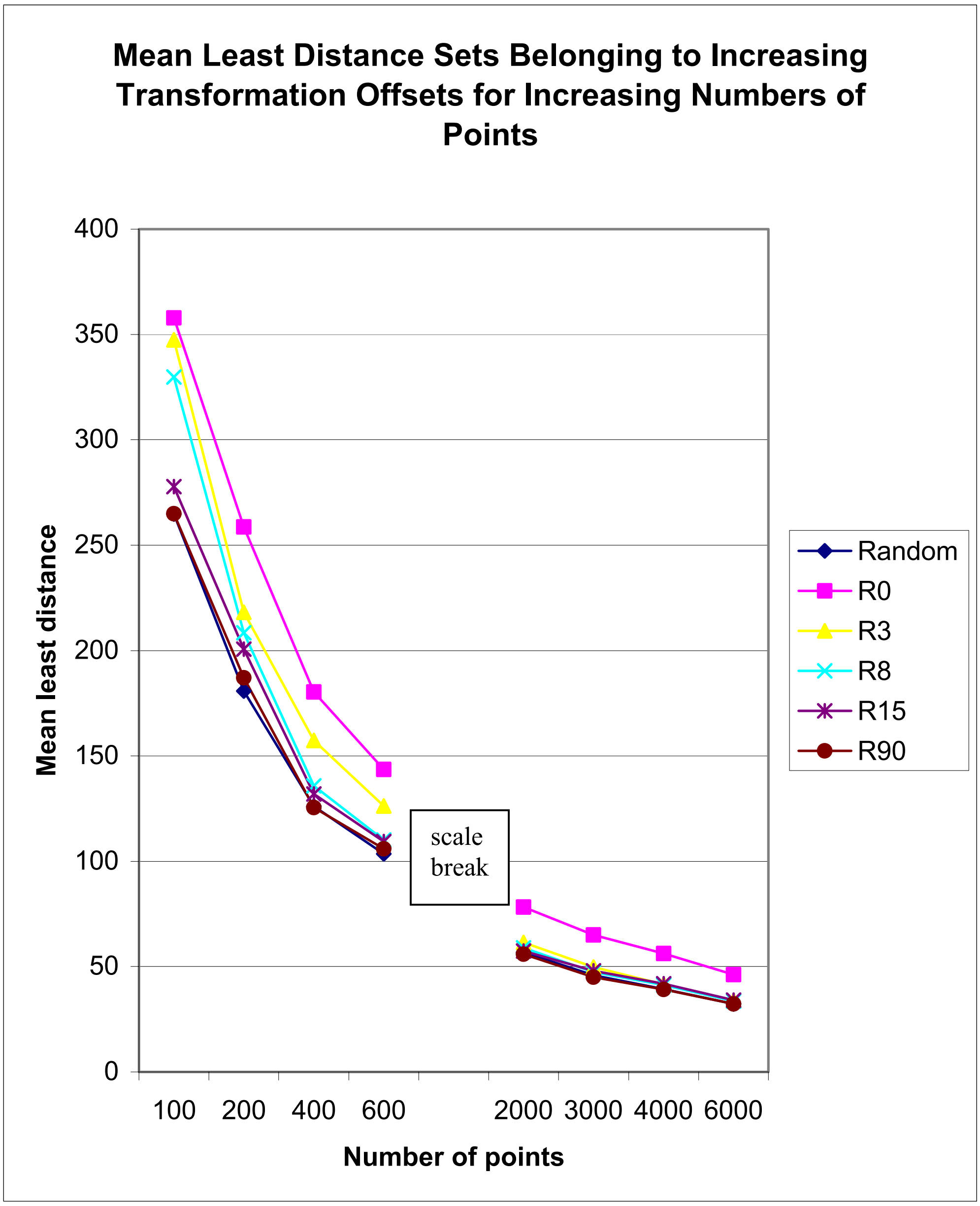


Figure 8.2: Mean least distance (average nearest neighbour distance) decreases with increasing point density and increasing transformational offset.

1. As intra-pair distances reduce, the mean distance for a set of *least distances* between two like Glass patterns intermingled on a form, and each having *N* points, approaches that of two random distributions intermingled on a form, and each having *N/2* points. To see this, as transformation distances reduce to zero the total of *2N* points belonging to the 'Glass patterns' becomes equivalent to the *N* points belonging to the random distributions. The situation is indicated by the plot labelled 'R0'. Here, R0 indicates a rotational 'Glass pattern' with 0^o rotation between transformation partners; which equates to half as many noise points. (Equivalent results pertain to any type of Glass pattern: a rotational glass pattern is arbitrarily chosen for this exposition.)
2. As intra-pair distances become large relative to inter-pair distances, then the mean distance for a set of *least distances* between two like 'Glass patterns' intermingled on a form, and each having *N* points, approaches that of two random distributions intermingled on a form, and each having *N* points. To see this, consider the lengths of

the distance radiations from any point in a set A to every point in a set B. See Figure 8.3. For relatively short intra-pair distances there is a good approximation for equivalent lengths from A_i to B_j and A_i to B_j' (where B_j' is the displacement from B_j). And these two distances are close in length to the two distances A_i' to B_j and A_i' to B_j', and so the different lengths of the distance radiations are grouped. This weakens as intra-pair distances become larger, and for relatively large intra-pair distances there are no such groupings. Hence, as transformation distances increase from zero to some large magnitude, the mean distance for a set of *least distances* approaches (from above) that of two random sets each having *N* points, indicated in Figure 8.2 by the plot labelled 'Random'.

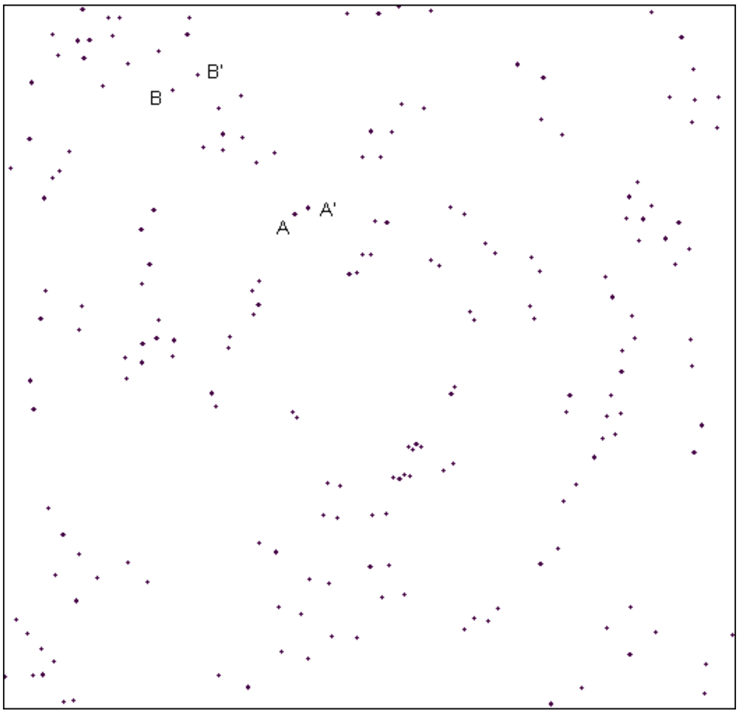


Figure 8.3: Rotational Glass pattern showing a grouping of distances: A_i to B_j, A_i to B_j', A_i' to B_j, and A_i' to B_j'.

The other plots, R3 through to R90, are for patterns with 3° through to 90° of rotation respectively between transformation partners. These plots indicate decrease in discriminability with increasing density and increasing transformational offset. Observe them in relation to the *N/2* situation for noise (top plot: R0) and the *N* situation for noise (bottom plot: Random), which correspond respectively to the two convergences just explained.

The second convergence explains what might appear to be two phenomena, but in fact are one and the same. They are:

1. As 'within pair' distances increase, the pattern effect is first degraded and then lost.
2. As pair density increases, the pattern effect is first degraded and then lost.

The greater the density of pairs, the shorter intra-pair distances must become to maintain an accordant pattern effect. This reflects the 'tension' between intra-pair distances and inter-pair distances. Figure 8.4 shows the average of four subject's results from responding to a pilot study prescribed by Vickers (2002). Subjects were asked to discriminate Glass translation patterns from equivalent density noise displays in a single interval, two alternative, forced choice design. Numbers of dots in the collection of Glass patterns ranged from 50 to 3000 in steps of 50, and transformation distances ranged from 50 to 250 screen units in steps of 50. In the sequence of displays, dot density levels in combination with transformation magnitude were reached such that the Glass effect was obviated, and could no longer be distinguished from noise. This situation is indicated in Figure 8.4 by the dots, at which points subjects were operating at chance level. Any ordinate point below any dot in each series indicates perception of Glass effect by at least one subject. Ordinate points above each series of dots correspond to no detection of Glass effect by any subject. Results indicate the exponential relationship of the tension between intra-pair distances and inter-pair distances.

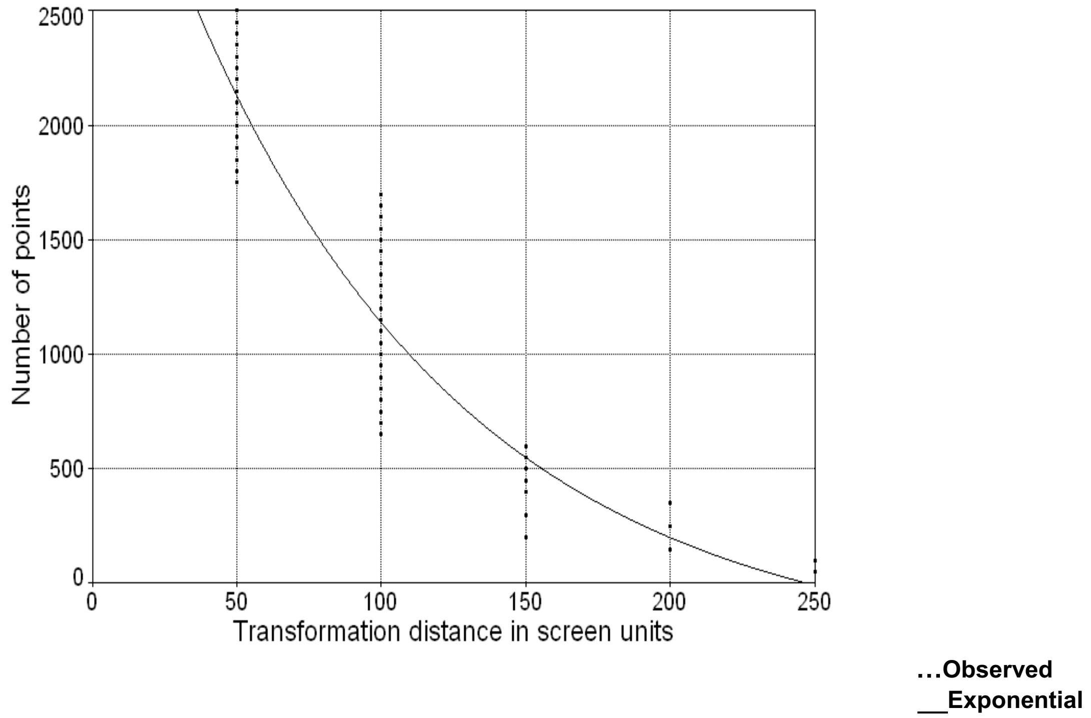


Figure 8.4: The dots show average results of four subjects for discriminating translation Glass patterns in which the number of dots and ‘within pair’ distance were varied (derived from Vickers, 2002). The solid line (curve fit) shows an exponential decay.

Note: correlation between members of pairs has little to do with the argument. Such correlation exists for any intra-pair separation, no matter how large. The measure outlined above discriminates ‘correspondences’ between members of pairs, which only become correspondences, as such, at relatively short intra-pair distances. It allows for tension between intra-pair distances and inter-pair distances.

The intermingled point sets employed above simply serve to facilitate an explanation. Two identically located, identical point sets yield an equivalent outcome. Here, distances between set A and set B are identical to distances between set A and set A, which are identical to distances between set B and set B. In other words, distances between points *within* a set are the same as distances between points *between* sets. Hence one point set of a kind produces an equivalent result, but in this case distances of zero (distance between any point and itself) are discounted.

This account accords with the account given in Chapter 2 inasmuch as nearest neighbours have a fundamental relevance to Delaunay neighbours. It does not go the whole way in explaining the phenomenon of Glass patterns, but it does make a useful contribution.

Pattern detection

Glass translation patterns, and some other Glass patterns, have consistent distances between original and transformed elements across all sets of distances, whereas combinations of distances between elements lack such consistency. Moreover such patterns embedded in noise (random dots) have consistent distances between original and transformed elements across some distinguishing proportion of sets of distances. *Consistency is an important factor in pattern detection.* Any Glass pattern with a non-constant separation between transformation partners—Glass rotations and perspective dilations, for example—show consistency, which can be detected by extending the transformation search.

For the following example, see Figure 8.5. A subset of normals is found from all normals intersecting the midpoints of all point-to-point distances. The subset is distinguished by the locus of convergence it forms over some nominally small area. (Only segments of just two normals are shown in Figure 8.5, by way of example.) Corresponding point-to-point distances can then be further checked for a constant relationship defined by *point-to-point distance* divided by *distance to convergence*. The same can be done for Glass dilations, only a locus of convergence is formed by a subset of parallels instead of normals: see Figure 8.6, in which the pairs of Figure 8.5 have been rotated about midpoints by 90^{o}. Indeed the transformation search decrements angles of virtual lines intersecting the midpoints of all point-to-point distances by some nominal gradation over the range from perpendicular to parallel, all the while looking for convergence.

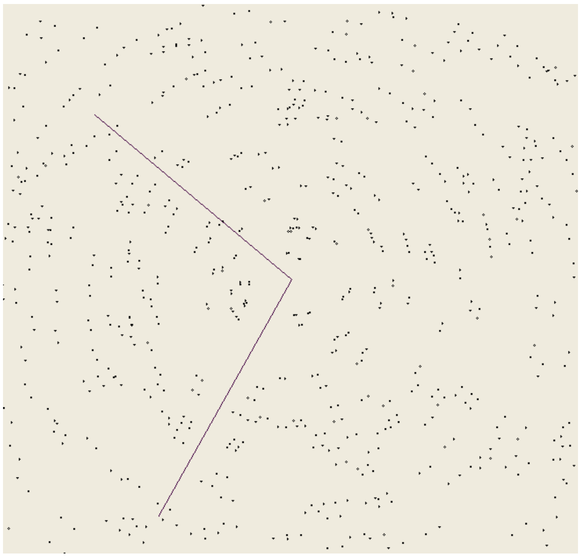


Figure 8.5: Glass rotation pattern.

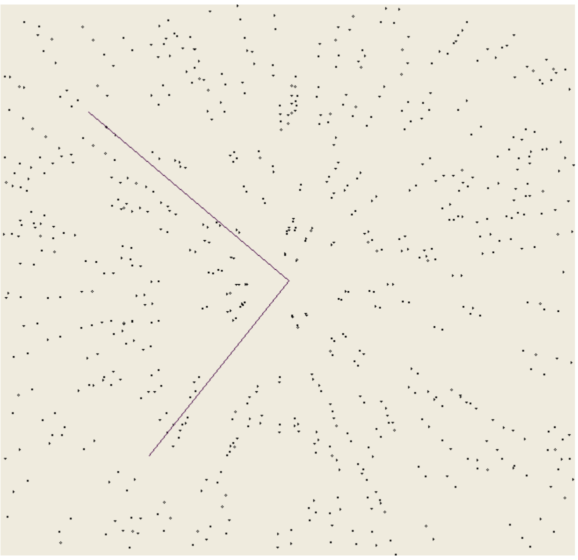


Figure 8.6: Glass dilation pattern

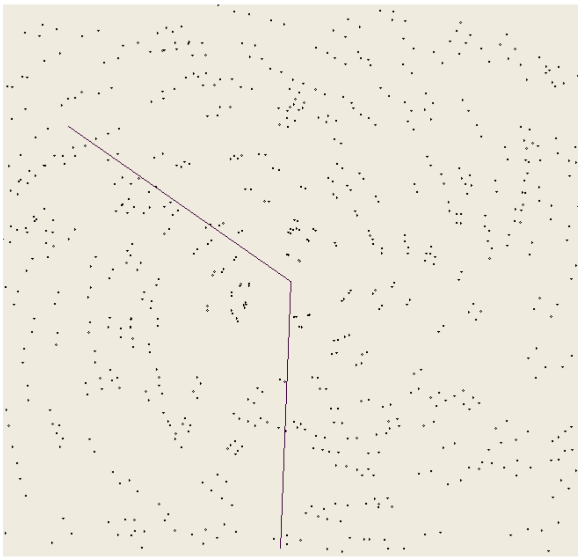


Figure 8.7: Glass screw pattern.

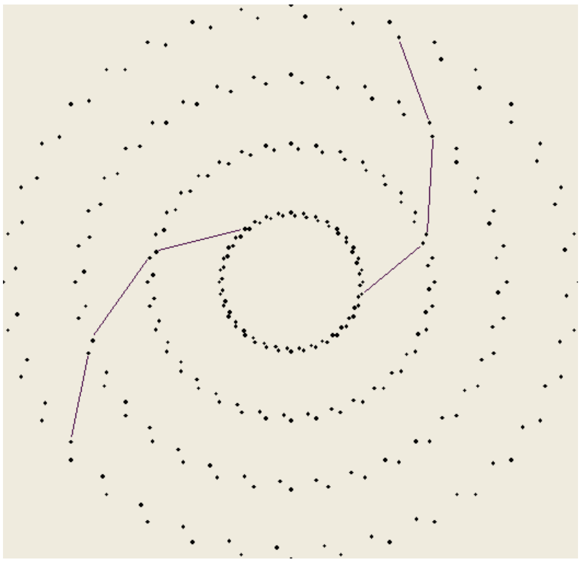


Figure 8.8: Test pattern for Glass screw pattern.

A Glass screw pattern can be generated by taking a Glass rotation, or dilation, pattern and rotating the pairs by some constant amount: see Figure 8.7, in which the pairs of Figure 8.5 have been rotated about midpoints by 20°. The transformation search now finds virtual line convergence at 20° from perpendicular, and somewhat similarly for hyperbolic transformations—indeed for any mathematical transformation—by suitable analysis of resultant geometry. A pattern can be embedded in noise and be geometrically off-centre, and the method performs equally as well. And, of course, arcs, circles, and other regular forms also can be found by this method.

A point of interest is now raised in support of the notion of consistency in distance and orientation. The Glass screw pattern shown in Figure 8.7 was generated by rotating the pairs of the Glass rotation pattern shown above it, in Figure 8.5. The pattern shown in Figure 8.8 is composed of regular, rather than random, elements. These are formed by concentric circles of dots, with pairs rotated by some constant amount. And the only way to get a screw interpretation from this is to trace virtual lines of optimal consistency in distance and orientation. (Only segments of just two virtual traces are shown in Figure 8.8, by way of example.) Hence the screw interpretation of Figure 8.7 seems to be an optimal (and natural) response to the evidence and, owing to the myriad alternative possible interpretations, is not forced upon an observer as the only possible interpretation.

At this stage, the utility of a transformational approach can be made more obvious. At the end of Chapter 5, page 99, it was stated that reflection symmetry defies classification methods that differentiate point sets. However, reflection symmetry poses no problem for a transformational approach because it, like other structural symmetries, is inherent in such an approach. The transformation approach deals with pattern matching through transformations, by looking to minimize pattern mismatch. By using the heuristic of grading tasks from mathematically simple to mathematically more extensive, any pattern is first tested for reflection symmetry across a vertical line through its *x*-coordinate average, which simply involves a transposition of *x*-coordinates followed by a test for match with the original pattern. If there is no match, then the corresponding procedure is also performed across a horizontal line through the *y*-coordinate average. If there is no match, then the *x*-coordinates are transposed again; which means they are returned to their original positions. (Hence matching is tried in the order of *x*-coordinate transposition, *x* and *y*-coordinate transposition, and then *y*-coordinate transposition.) If there is still no match after the last transposition, then the search for Glass patterns is tried, and so on, towards patterns that are mathematically more complex: the point being that a transformational approach manages elementary reflection symmetry with ease. More complicated symmetries, further along the path to complexity, are addressed at the beginning of the next chapter.

If dots constituting Glass and other patterns, especially those embedded in noise dots, are not suitably proximate (relative to form scale) then pattern effect is degraded or lost. However, as the model stands, a distinguishing frequency of some identifying consistency is still realized. To resolve this situation a pattern may be subjected to a test for Delaunay neighbour loading, as per Chapter 2, upon which any useful point-to-point distance must then fall in the range of the Delaunay neighbourhood. Alternatively, a pattern may be subjected to Ripley's *K* function, as per Chapter 4, which indicates a distance at some level of significance for the disposition of pattern elements with respect to an equivalent random situation. Any useful point-to-point distance must then fall in this range.

By and large, if reflection symmetry is not affirmed then a pattern is subjected to one or another statistical analysis for randomness, as outlined in the last section. If random, then analysis halts or can proceed tentatively. And if analysis proceeds then pattern detection needs to be realistic in its characteristics.

A threshold system for pattern detection in noise dots

In the endeavour to illustrate realistic pattern detection, a relatively simple instance is advanced. The example utilizes Glass translation pairs, which have constant intra-pair distance and orientation. Only distance is here dealt with. The parameters that are fixed in this example produce results equivalent to other combinations of such parameters and still further combinations could produce different results, hence no importance should be placed on absolute values given to parameters.

The example explains how the model can emulate Glass pattern detection by subjects, who responded to a task in which Glass translation pairs were obscured by noise (random) dots (Dry, Vickers, Lee, & Hughes, 2004). One configuration of the task presented to the model utilized 300 dots in all, of which the ratio of Glass dots to noise dots was increased from .2 to .45 in small, equal steps. This range was extrapolated from proportions of pattern dots in noise as outlined in Dry et al., and is a good estimate for subjects picking out differently orientated Glass translation patterns at chance level to complete accuracy, in effect. Dry et al. provided subjects with a three-way choice: left down, right down, or noise. The stimuli were Glass translation patterns oriented at 45^{o} and embedded in noise, Glass translation patterns oriented at 135^{o} and embedded in noise, or noise with no pattern. At each step, the same number of independently generated categories were presented. The way in which the model deals with the situation is as follows.

All inter-dot distances are perturbed such that they vary randomly, independent of one another, in the range zero to ±10%, say. If the tolerance for consistent distances is set to ±5%, say, then, on average, half of the maximum possible number of consistent distances in a pattern is found. Moreover if many different patterns are presented, then for about half the presentations the consistent distances found in each pattern are not less than 50% of the maximum possible number of consistent distances. (The model does not require a number of identical distances before detection occurs. As long as the distances fall within a tolerance range then detection occurs. Additionally, perturbation range and tolerance need not be set at 10% and 5%: an enormous range of workable combinations is possible.)

The Glass to noise ratio around which most variation is likely to occur in response accuracy by subjects is .32 (halfway between chance response and complete accuracy, in effect). This corresponds to a response accuracy of 66%. The number of consistent distances the model needs to detect in order to perform at 66% accuracy for a Glass to noise ratio of .32

can be declared a threshold for whether or not a pattern is detected. In this instance the threshold is 25 in absolute terms (and the same figure could have been produced at any other ratio around which a lot of variation is likely to occur, but these situations may not be optimal for accuracy). If the threshold is increased, then a pattern is detected for less than 66% of the presentations. For example, if 70% of the maximum possible number of consistent distances within a pattern is required to fall within a tolerance of ±5% before the pattern is detected, then detection occurs less often in a sequence of such patterns. Conversely, if 62% of the maximum possible number of consistent distances within a pattern is required to fall within the tolerance of ±5% before the pattern is detected, then detection occurs more often in a sequence of such patterns. The following provides a concrete example, initially set at the Glass to noise ratio at which subjects performed at chance level.

Let a translation Glass pattern have 30 pairs of dots. Then there are 30 consistent intra-pair distances, along with the inconsistent inter-dot distances for the pairs. The pattern is obscured by 240 randomly situated dots, which, along with the 30 pairs, generate many inconsistent distances. This is a situation in which 20% of dots contribute to pattern structure. If many such displays are presented then the relevant question is ‘How many times will 25 or more inter-dot distances fall within a range of ±5%?’ The answer is ‘Rarely’, but if 45% of dots contribute to structure then the answer is ‘Nearly always’.

The number ‘25’ is a threshold, or least, amount of consistent distances to be found before a pattern is detected. For 30 possible consistent distances (20% structure) it is most unlikely that 25 of them will fall in the ±5% range if they are perturbed by a maximum of ±10%. But for 67 possible consistent distances (45% structure) it is very likely that 25 of them will fall in the ±5% range. Between the lower and upper limits, likelihood of pattern detection increases in a sigmoidal way (about which more presently).

How does the threshold change with total number of dots in a display? It changes as per the expression N / p, where N is the total number of dots and p is a constant, which dictates the proportion of total number of dots that constitutes the threshold. The constant can be made less for poorer performance and more for better performance, and can be adjusted to emulate any particular subject’s performance. This ensures that for sparse displays, detection performance does not significantly alter. However, for dense displays detection performance falls off, as would be expected, when mean nearest neighbour distance becomes significantly less than Glass translation distance.

The effect of the threshold on the proportion of patterns detected for increasing numbers of structure dots is to produce a sigmoid shaped function. The function indicates that for relatively small and large percentages of structure dots, very little change in detection rate occurs, but over some intermediate range a steady change in detection rate occurs. This, of course, is to be expected in any plausible pattern detection model. The sigmoid function provides a range of probabilities for detection (dependent variable) versus number of structure dots (independent variable).

An equivalent process to that of distance exists for orientation of Glass translation pairs. For the types of Glass pattern in which distance and orientation are not constant, then the intermediary of the transformation search resolves the issue. More generally a similar process could be applied to detect many regular patterns obscured by noise.

Parameter values do not mean much in the absolute sense because of the enormous, but constrained, combination that can be tuned to get equivalent results. What is important is that the model performs a number of trials for each number of Glass pairs. Each Glass pair has a random perturbation, constrained by a common limit, imposed upon its transformation distance. Every other dot-to-dot distance also has imposed a random perturbation, constrained by the same limit. Going from a lower bound, the number of Glass pairs is increased in steps over a range. Given some number of Glass pairs, the model discovers how many times transformation distances over range 'plus and minus tolerance for consistent distances' are greater than or equal to 'detection threshold'. In the light of results from Dry et al., it should be virtually no times for Glass pair populations of 20% and virtually every time for Glass pair populations of 45%. 'Detection threshold' operates over a contiguous range of Glass dots at some fixed total number of dots: both Glass and noise. It increases for increasing total numbers of dots.

Basically, results for any subject can be emulated by adjusting the limit level of random perturbation, or noise, for dot-to-dot distances, with greater levels of noise corresponding to poorer results. Maybe it is convenient to consider more inept performances from subjects to be clouded by more noise in the form of anything that reduces application to the task at hand. In any event, noise provides a very flexible 'device' by which the model can be tuned to emulate performance by subjects.

Figure 8.9 shows model results for the concrete example given above. The abscissas range in value from 20% to 45% for proportion of structure dots, while the ordinates range in value from .3 to 1 for detection rate: just below chance level to always correct, in effect. Responses describe a sigmoid shape, which becomes smoother as the number of trials at each Glass to noise ratio is increased from top left to bottom right.

Figure 8.10 shows the subject's results for the three-way choice. Here, the abscissas range in value from 14% to 30% for proportion of structure dots. However, values that relate to detection by the model range from 20% to 30%. Values before this were of little use because detection rate by subjects remained essentially at chance level. The ordinates in Figure 8.10 range in value from .2 to .8, with values relevant to detection rate by subjects ranging from .3 to .6: just below chance level to correct for 60% of presentations. Inasmuch as the model responded to similar cases and numbers of displays as were presented to subjects, results generally concurred. In any event, both the model and subjects show about 56% detection rate for 30% of structure dots.

Of course the difference in step size for percentage of structure dots needs to be considered when comparing the graphs, in addition to the range 14% to 30% of structure dots in Figure 8.10. However, Dry et al. took the limit to 40% of structure dots for a subsequent experiment reported in their paper, and inasmuch as translation distance matched that of the Glass patterns associated with the graph of Figure 8.10, the result generally concurred with that of the model. (Dry et al. found that loading on first and second nearest neighbour distances and orientations in the range $\pm 5^{o}$ to $\pm 15^{o}$ of Glass pattern orientation nicely explained subject's responses.)

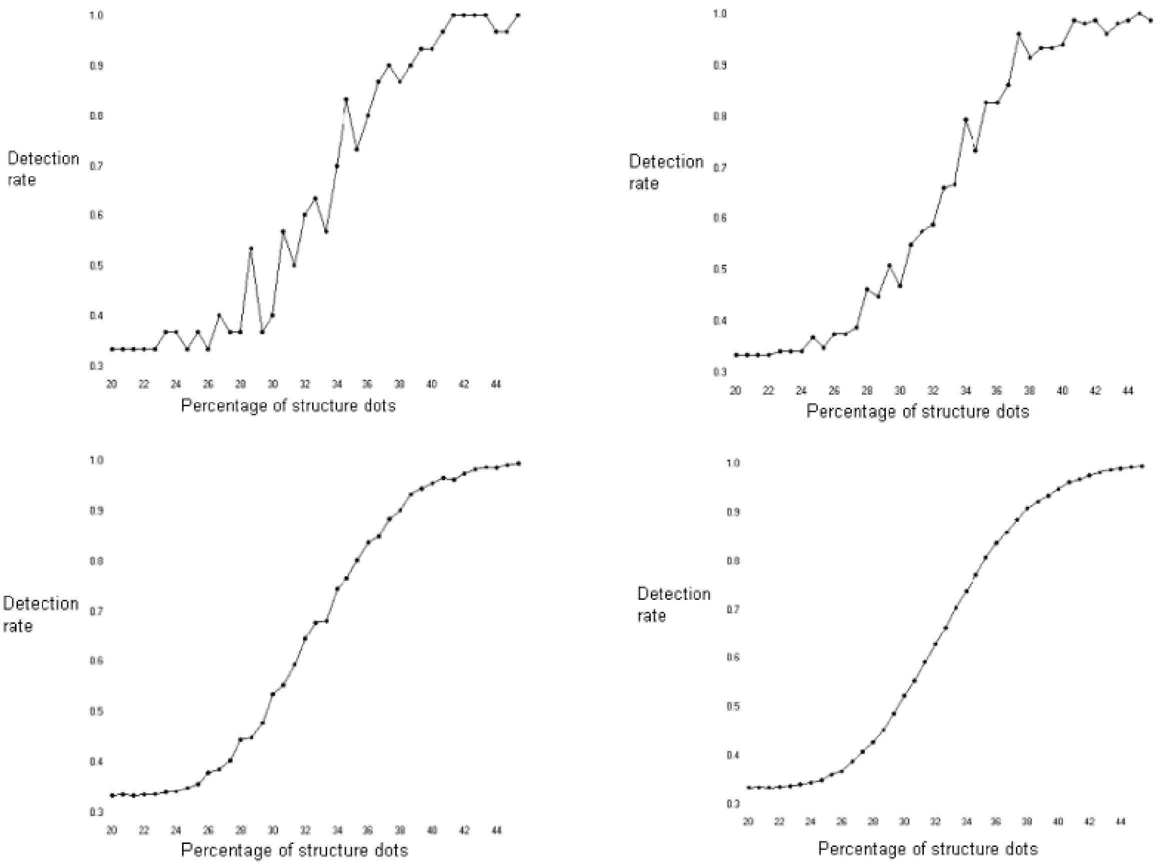


Figure 8.9: Glass pattern detection rates for the model become smoother as the number of trails at each Glass to noise ratio is increased in the order top left, top right, bottom left, bottom right.

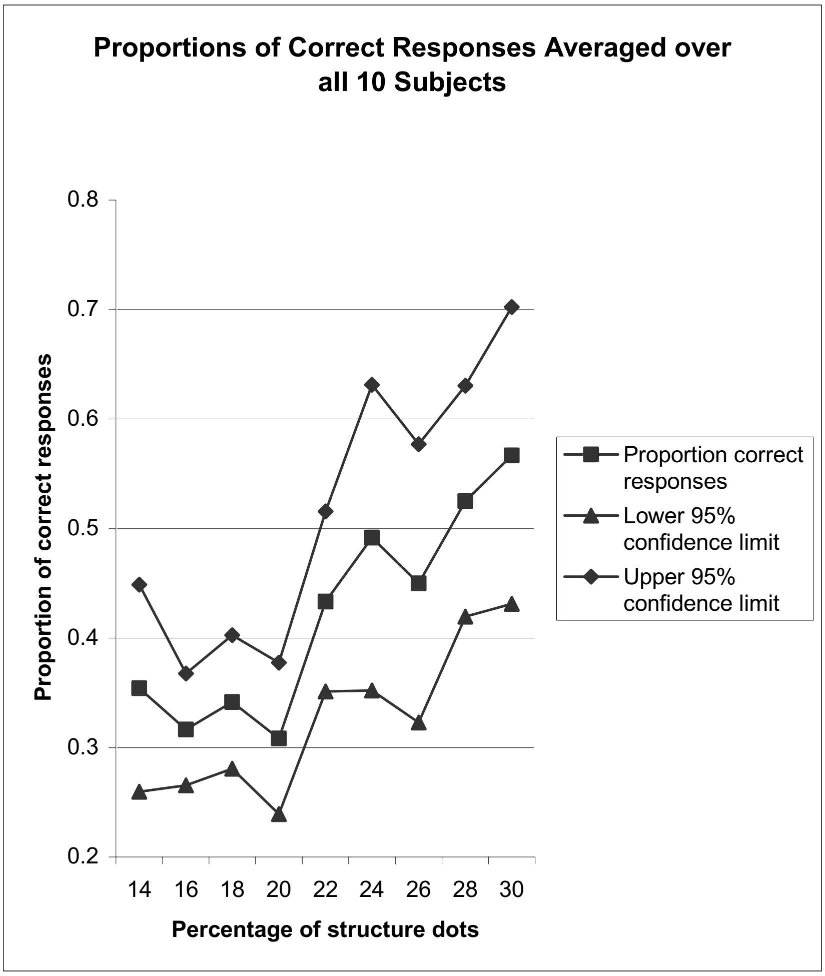


Figure 8.10: Averages of proportions of correct responses made by subjects in a pattern discrimination experiment by Dry, Vickers, Lee, and Hughes (2004).

The Hausdorff distance procedure and greyscale charts

In calculating Hausdorff distance, sets of distances are produced. The example illustrated in table 7.1, page 122, shows the sets in regular font style. Taking the directed Hausdorff distance from A to B, each set shows the distances from one point in pattern set A to all points in pattern set B. The method of pattern detection employed herein involves searching sets of distances not only for consistencies, but also for relative frequency of consistencies from which frequency ordering of hierarchies of Euclidean distances, with corresponding orientations, can be obtained.

In pattern sets of equal size, all sets of distances are of equal size and the h(A→B) and h(B→A) sets of distances are the same. In the model, the two pattern sets are identical and one pattern set directly overlies the other; which equates to checking a pattern against itself.

As a benchmark, Figure 8.11 gives a greyscale representation of sets of distances for some randomly distributed dots. (Greyscale charts are employed to illustrate a principle, and, as such, just distances are chosen by way of example.) Shading is scaled linearly with point-to-point distance from white to black. The greatest distance equates with white (represented by red, green, and blue components, each with a value of 255) and the least distance (zero) equates with black (represented by red, green, and blue components, each with a value of zero).[1] Because the sets are superposed, the diagonal is always composed of zero distances (distance between each dot and itself), and each distance is duplicated by its reciprocal.

For 32 randomly distributed dots there are 32 sets of 32 distances in a calculation column of the Hausdorff type. The sets of distances are arranged in 32 columns across the corresponding greyscale chart, each column being 32 squares high and 1 square wide. (The model scales to deal with many more elements, but these have been necessarily limited for the greyscale display.)

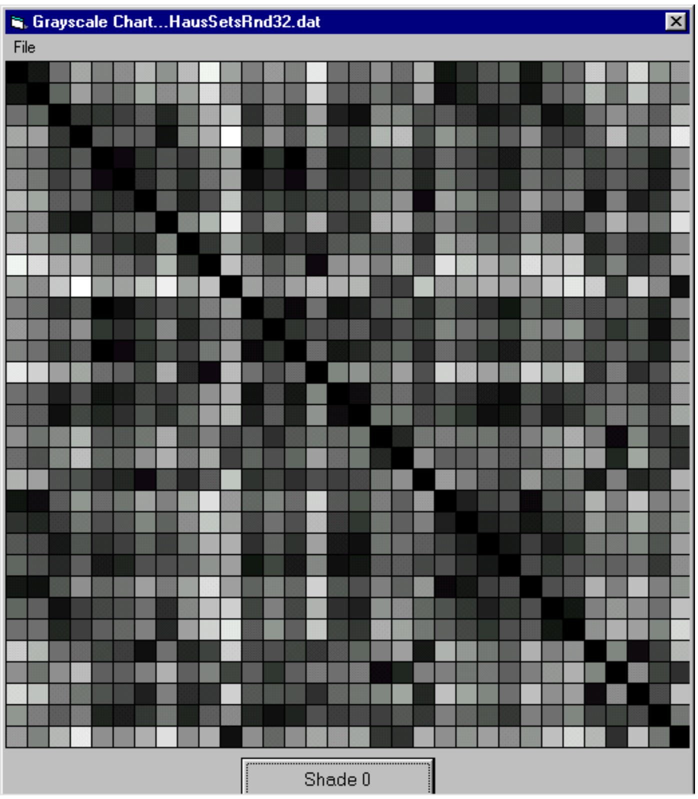


Figure 8.11: Greyscale chart for randomly distributed dots.

Detailed explanation of of greyscale chart organization

Before explaining how the rows and columns of a greyscale chart are ordered, it is expedient to discuss array representions. Correspondence with a retinal array can be achieved by implementing a two dimensional array in computer memory. The display scaling used mostly throughout this thesis is 5000 × 5000 screen units, which equates to twenty-five million array cells made up of 5000 columns, each column consisting of 5000 cells. Active cells contain ones, and quiescent cells contain zeros. With this representation, the values of *xy* indexing used to access the cells in computer memory provide position information, from which inter-point distances can be calculated.

However, there is a more memory efficient representation, which only has as many pairs of array cells as there are active cells in the previously described representation. In this abbreviated representation, the first element of each pair contains the *x* position information for an active cell and the second element of each pair contains the *y* position information for the active cell. Nothing is allocated for quiescent cells, and the *xy* indexing values to the array cells do not provide position information.

[1] The RGB model is used to specify colours for displays and other light emitting devices. Red, green, and blue values each range from 0 through 255, with 0 indicating minimum intensity and 255 indicating maximum intensity. Other colours are produced by different intensities of these primary colours.

Active cells are arranged in the latter representaion in order of distance from an arbitrary location in the the former represention. Indexing for the former representation goes 1,1, 1,2, to 1,5000, then 2,1, 2,2,…,2,5000, then 3,1, 3,2,…,3,5000, etc, up to 5000,1, 5000,2,…,5000,5000. For the latter representation indexing goes 1,1, 1,2, 2,1, 2,2, 3,1, 3,2, etc, up to *n*,1, *n*,2, where *n* is the number of active cells. Such indexing of computer arrays is implemented by nested loops, which are loops inside of loops. Since there is an equivalence between the two represention*s*, the latter, more memory efficient, representation is generally used.

I now explain how the rows and columns of a greyscale chart are ordered. First, elements of static visual arrays are generally distributed in a spatially structured fashion. (Elements of structure are generally located contiguously within segments of arrays). Concerning the examples indicated for the greyscale charts, structures may be reasonably (and often are) located contiguously in respective computer arrays. The number of point-to-point distance combinations for *n* points is n^2. Because the displays under consideration are limited to just 32 elements, there are 32 sets of 32 distances produced by the Hausdorff calculation. The order of distances within such sets, along with the order of the sets, are primarily dictated by the order of the position defining elements in the computer array.

Distance representations are generally arranged in a greyscale chart in the order produced by the Hausdorff calculation (which indexes an array in numerical order via a nested loop). The distances are arranged in columns 32 squares high and one square wide. Since there are 32 of these 32 element columns (located side by side across the greyscale chart) they accommodate all distances in the sets produced by the Hausdorff calculation.

It might be (reasonably) asserted that contiguous elements of patterns contained in visual arrays can be occluded by noise elements. This situation has a correspondence in the computer model. First, note that an element of the data array specifies a specific position of an element on the computer display. This is the case no matter where in the data array the element is located. Thus elements that specify positions of noise on the display can interrupt contiguity in the array of elements that specify positions in a pattern on the display.

Again, whether or not the elements that specify positions of noise on the display result in pattern occlusion depends on the position information contained in the elements, not their locations in the data array. However, ordering of distance elements in the Hausdorff calculation is directly related to locations of position elements in the data array, and therefore to the layout of the greyscale chart.

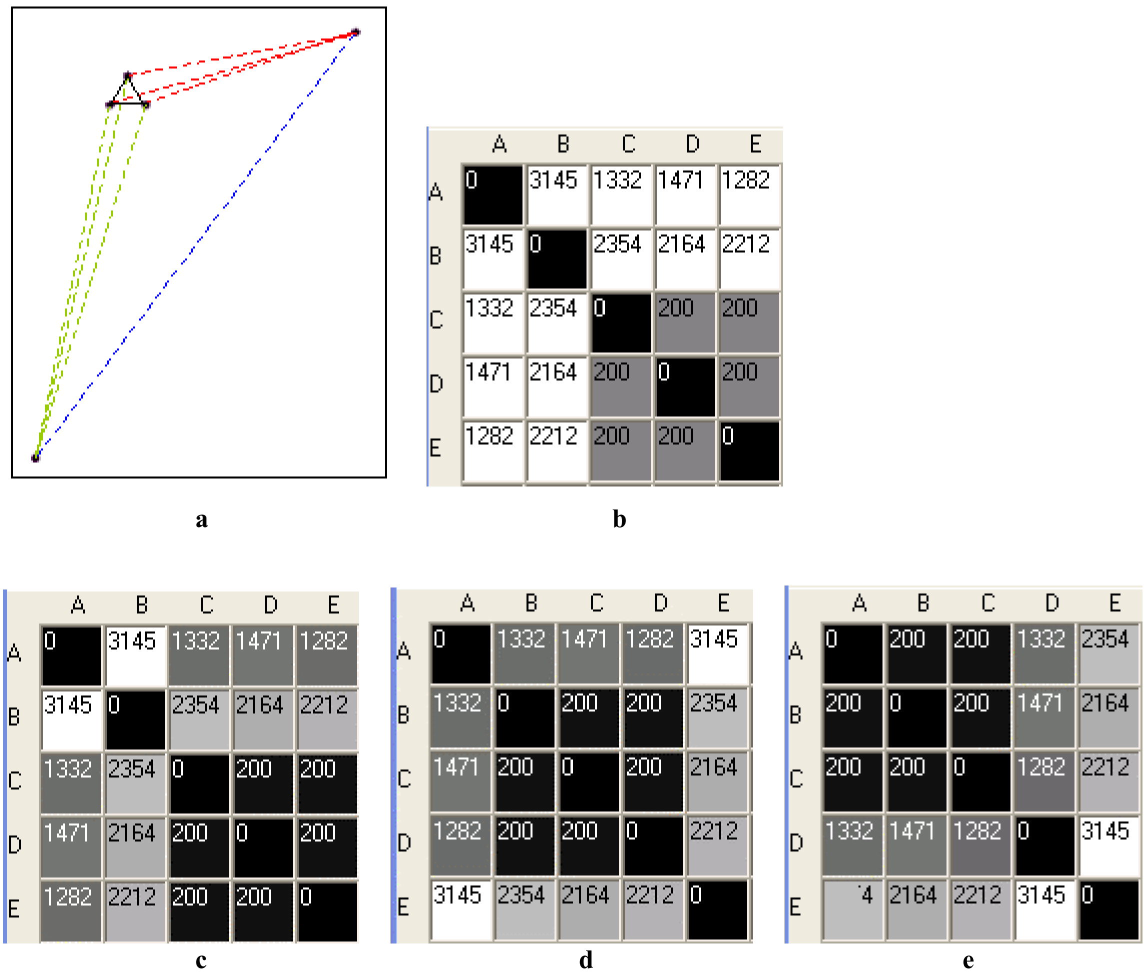
Further it may be (unreasonably) asserted that the model should detect patterns no matter where (position defining) elements are located in data arrays. Though this situation may have little correspondence to the general pattern contiguity of visual arrays, the model can deal with such a situation. If pattern contiguity is compromised, structure is still found by the pattern detection methods outlined herein, and distances arranged to represent contiguous structure.

For cases in which pattern contiguity in an array is compromised in whatever way, the distance elements produced by the Hausdorff calculation need ordering to reflect the discovered structure so that it can be seen as a contiguous whole in the greyscale chart. In delineating structure the model finds relevant indices to the data array that holds the distance sets (which are matched to those of the data array that holds the position set, making it easy to delineate structure on the display.) Thus it is a relatively simple matter of arranging the distances that represent structure contiguously in the distance array so that it can be seen as such on a greyscale chart. In so arranging these distances, and to accord with the Gestalt proximity principal, each distance juxtaposes the next at the lowest consistent—within some tolerance—non-zero distance.

Figure 8.12 (a) shows a very simple instance of a regular object, namely dots in the form of an equilateral triangle against a background of just two noise dots. This keeps the number of point-to-point distances, indicated by the lines joining the dots, within reason for the purpose of explanation. Additionally, pattern contiguity in this simple instance is not compromised. Columns are arranged in the greyscale charts in the order produced by the Hausdorff calculations.

The consistent, relatively low distance spacing, corresponding to darker shades, is revealed by ‘bleaching’ the (often mottled) shading. This has a differentiating effect on shades at the darker end of the spectrum and a homogenizing effect on shades at the lighter end. Bleaching is implemented by multiplying each value for a shade by some small fixed amount greater than one, such that it becomes lighter. This is done repeatedly as the shades are incremented. The opposite is implemented by dividing each value for a shade by the same small fixed amount, such that it becomes darker. Again, this is done repeatedly as the shades are decremented. There are 14 bleach levels, which range from 1 (darkest, or least bleached) to 14 (lightest, or most bleached). Level 0 represents the original shading and is darker than level 1.

Structure in Figure 8.12 (a) is highlighted at shade level 6 by the grey squares in Figure 8.12 (b), which show distances of 200 screen units. These are the distances between the triangle’s vertices. The two distances about the diagonal at the top left are from one random dot to the other and vice versa. The other distances are cross terms, which are distances between structured and random elements. The same distances are also seen in Figure 8.12 (c) at shade level 0. In Figure 8.12 (d) at shade level 0 the structure has been shifted in the data array, and in Figure 8.12 (e) at shade level 0 it has been shifted again, thus exhausting all possible combinations of contiguous positions for the structure in the data array.



a

b

	A	B	C	D	E
A	0	3145	1332	1471	1282
B	3145	0	2354	2164	2212
C	1332	2354	0	200	200
D	1471	2164	200	0	200
E	1282	2212	200	200	0

c

	A	B	C	D	E
A	0	3145	1332	1471	1282
B	3145	0	2354	2164	2212
C	1332	2354	0	200	200
D	1471	2164	200	0	200
E	1282	2212	200	200	0

d

	A	B	C	D	E
A	0	1332	1471	1282	3145
B	1332	0	200	200	2354
C	1471	200	0	200	2164
D	1282	200	200	0	2212
E	3145	2354	2164	2212	0

e

	A	B	C	D	E
A	0	200	200	1332	2354
B	200	0	200	1471	2164
C	200	200	0	1282	2212
D	1332	1471	1282	0	3145
E	'4	2164	2212	3145	0

Figure 8.12: Dots representing a triangle in (a) against a background of just two noise dots. Structure in (a) is highlighted at shade level 6 by the grey squares in (b). These show the distances in each direction between the triangle's vertices, which are delineated by the black lines in (a). The two distances about the diagonal at top left are from one random dot to the other and vice versa. These are delineated by the blue line in (a). The other distances are cross terms between the structure dots and the random dots. These are delineated by the red and green lines in (a). The same distances are also seen in (c) at shade level 0. In (d) at shade level 0 the structure has been shifted in the data array, and in (e) at shade level 0 it has been shifted again, thus exhausting all possible combinations of contiguous positions for the structure in the data array.

Greyscale charts of some patterns

Figure 8.14 gives a greyscale representation of sets of distances for the Glass translation pattern shown in Figure 8.13. In this instance there are 32 sets of distances for 16 translation pairs. Observe that the Glass translation pattern is composed of randomly located pairs with equally spaced transformation partners. A feature that is not random about the pattern is the spacing between transformation partners, which is masked by the overall random shading of the greyscale chart. The consistent, relatively low distance spacing is revealed by 'bleaching' the mottled shading, as shown in Figure 8.15.

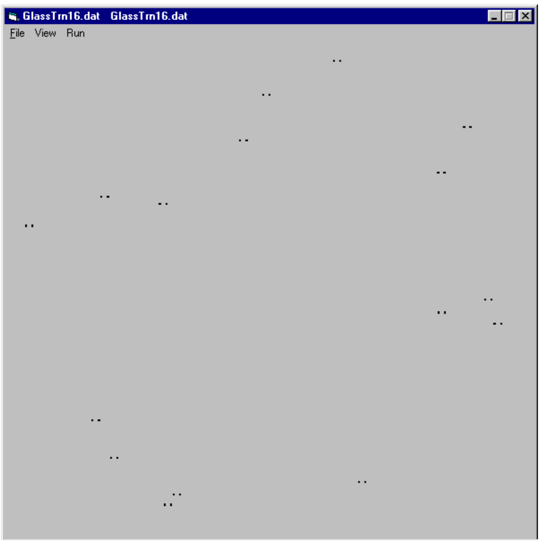


Figure 8.13: Glass translation pattern

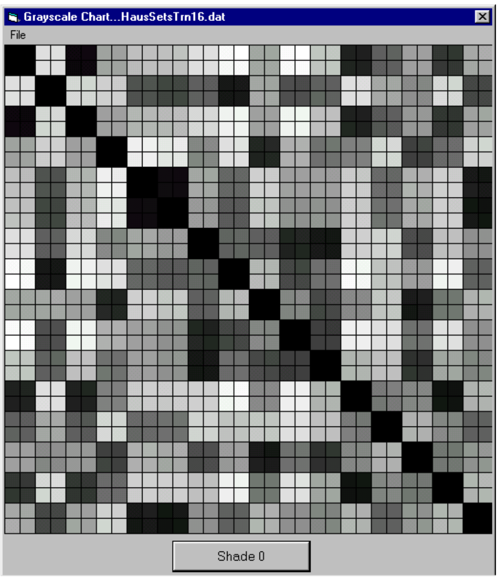


Figure 8.14: Greyscale chart for the Glass translation pattern shown in Figure 8.13.

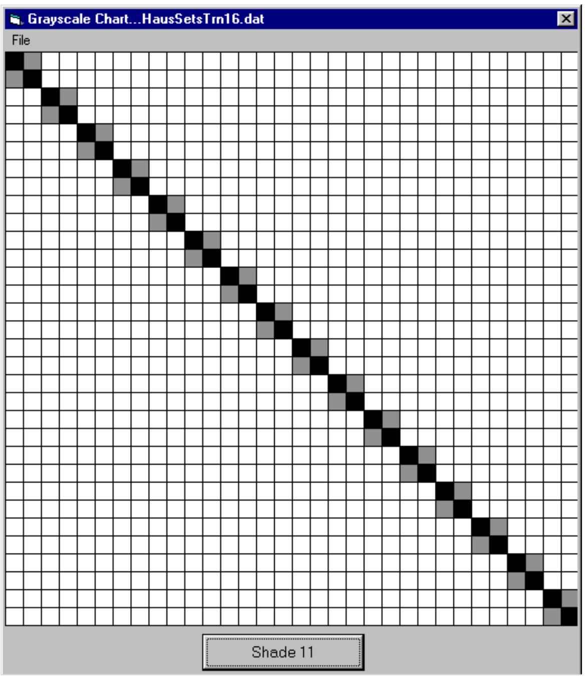


Figure 8.15: Consistent elements of the Glass translation pattern shown in Figure 8.13.

Given the range of distances, and that here the Glass translation distance is at the bottom end of the range, shade level 11 out of a possible 14 was invoked to clearly contrast structure. As the different shades, shown as they originally were in Figure 8.14, become white (red, green, and blue values equal to or greater than 255) they cannot get any whiter, while the remaining darker shades continue to be differentiated into lighter grades. In this way, detail that was buried in the previously darker shades is revealed. More generally, by trying increments and decrements of shades for best effect regarding uniformities and differentials of shades, structure is uncovered. Owing to the Gestalt proximity principal, structural elements generally prevail more with reducing distance from the diagonal.

The two greyscale charts shown in Figures 8.16 and 8.17 have been bleached to shade levels 3 and 2 respectively. They represent a square and a circle, along with the noise in which they are embedded. These two greyscale charts derive from Figures 8.18 and 8.19. Structure (lower right) and randomness (upper left), can be clearly seen as separate. It is this kind of detail that the model is programmed to discern. The remaining two areas show cross terms, i.e. distances between regular and random elements.

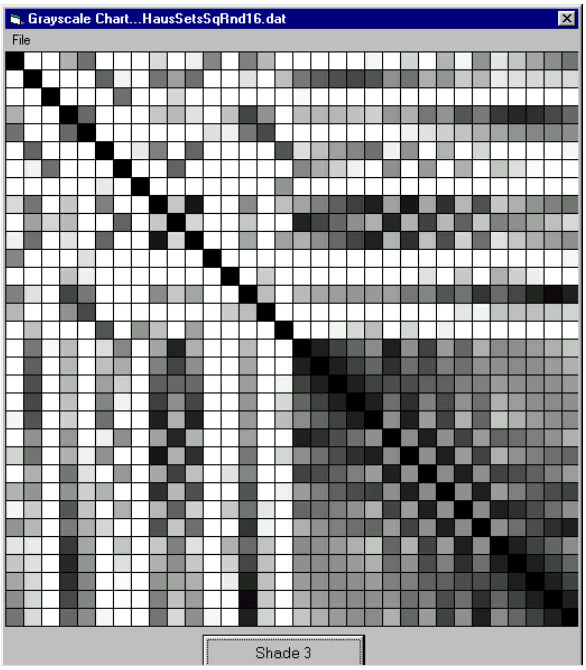


Figure 8.16: Greyscale chart for regular square in random dots.

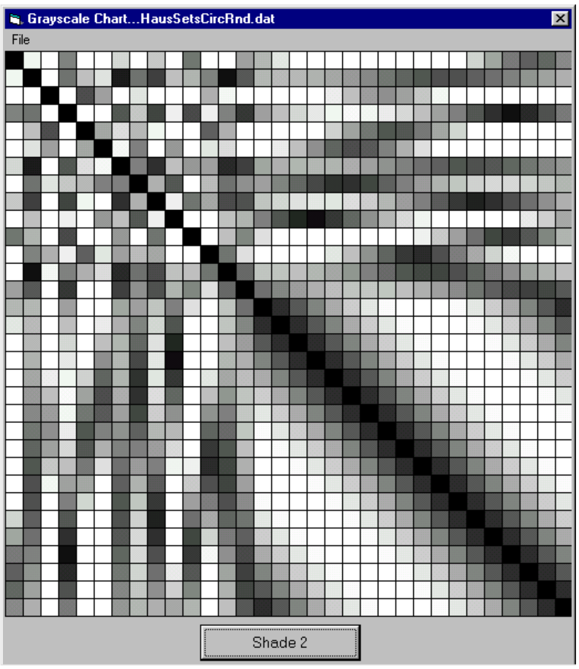


Figure 8.17: Greyscale chart for regular circle in random dots. Note the highly regular structure of the circle and the haphazard structure of the random elements.

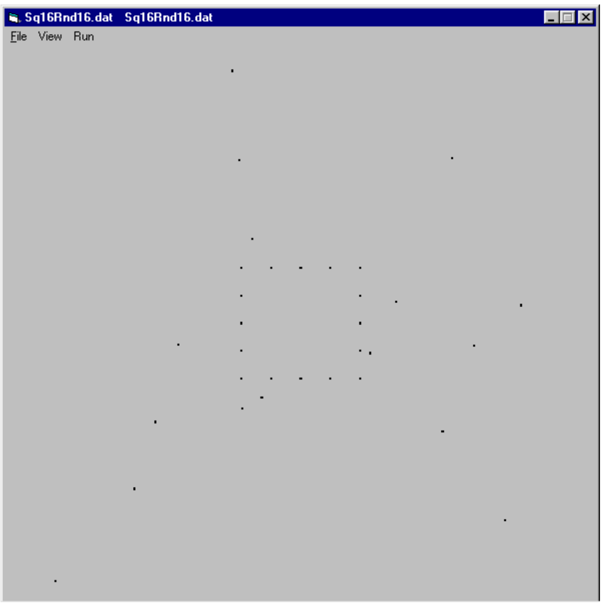


Figure 8.18: Square in noise.

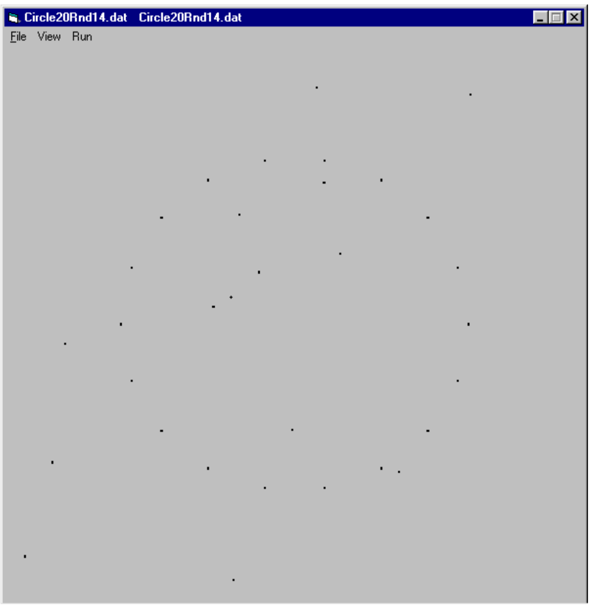


Figure 8.19: Circle in noise.

More on detection of structure

Frequencies of occurrence of orders of regularities between sets of distances are exploited in identifying any precise geometric shape placed anywhere in a field of noise.

Moreover hand-composed shapes constituted of lines and arcs of points somewhat irregularly spaced, and placed anywhere in a field of noise, are identified.

Regular distances are probed in ascending order of magnitude and descending order of frequency for point-to-point connection, until a connected object is yielded or the 'distance probe' threshold is reached. The distance probe threshold is set to some (relatively) longer distance corresponding to a significantly repeated distance, within some tolerance spread, at the upper end of the distribution of frequencies of regular or semi-regular distances. Here, the use of Ripley's *K* function is expedient for establishing such a threshold. Ripley's *K* indicates a distance at some level of significance for the disposition of pattern elements with respect to an equivalent random situation. Any of the distance measures covered in Section 1 of the thesis can provide an important distance limit or threshold for the generative transformation model, but that provided by Ripley's *K* is arguably the most generally useful.

What a *connected* object is depends on the results of a reconnaissance process. Before any connections are made, point-to-point distances are scouted to determine the number of orders of regularity and for the purpose of setting the distance probe threshold. If dealing with regularly spaced dots of singular low distance order ('single-consistency'), then each dot has in-degree one and out-degree one. If dealing with regularly spaced dots of multiple low distance orders ('multi-consistency'), then each dot has a related in-degree and out-degree. In/out-degrees can be under matched, but may not be exceeded. The distance probe threshold is also related to in/out-degree.

The model can detect grid structure at any height-to-width ratio by appeal to multi-consistency. By way of contrast, an empty square, for example, constituted of equally spaced dots has in/out-degree one and a distance probe with a low distance threshold. It is therefore connected by the lowest consistent distances greater than zero. The object exhibits single-consistency. Concerning multi-consistency, there is a number of low order consistent distances at one distance and a number of low order consistent distances at another distance, and so on.

For a grid of unequal height-to-width ratio the lowest order, consistent, non zero distances are utilized for the dots orientated along the lines of maximal dot proximity. Because the distance probe threshold is not yet reached, then consistent, non zero distances of an appropriate lesser low order are utilized for the dots orientated along the perpendicular lines. See Figure 8.20.

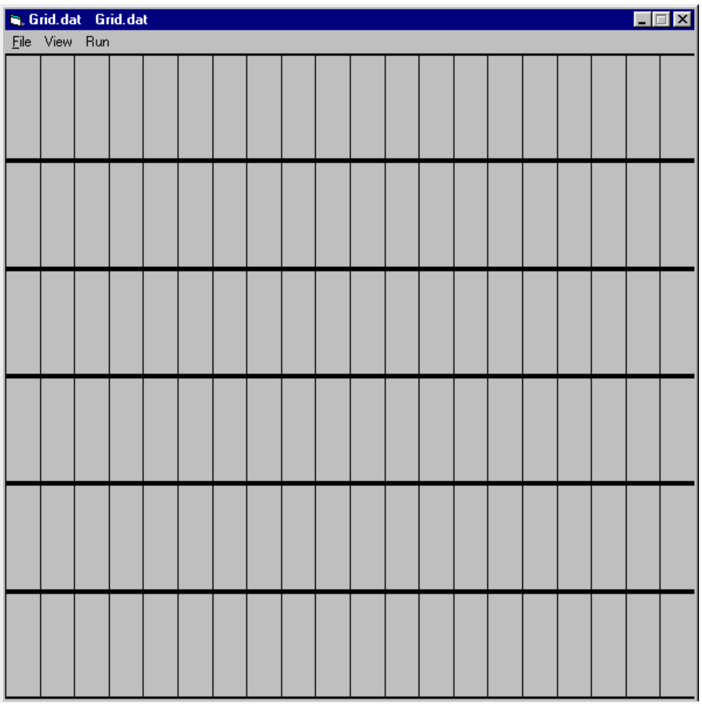


Figure 8.20: Grid structure found by the model in an input dot pattern. (Line thickness is related to dot proximity.)

The next greyscale chart, Figure 8.21, at shade level 2, is derived from distance sets produced by the somewhat irregular, hand-composed square shown in Figure 8.22. As can be seen there is still considerable regularity, relatively speaking, that can be exploited for detection of imprecise figures. Nonetheless there is a better way of dealing with imprecise figures, which also works well for precise figures.

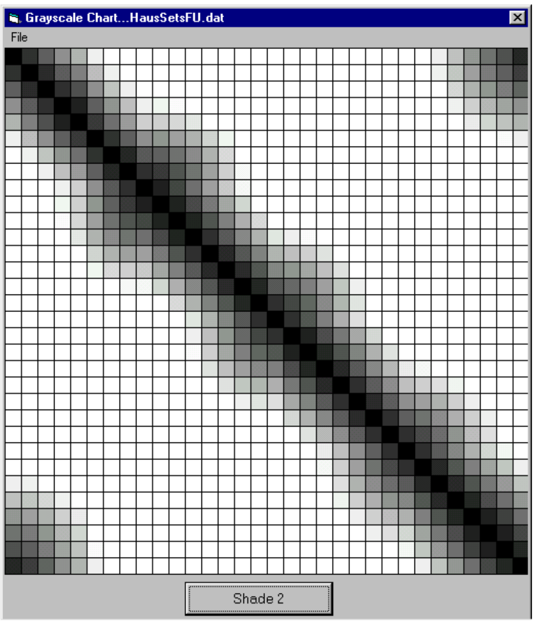


Figure 8.21: Greyscale chart for hand-composed square.

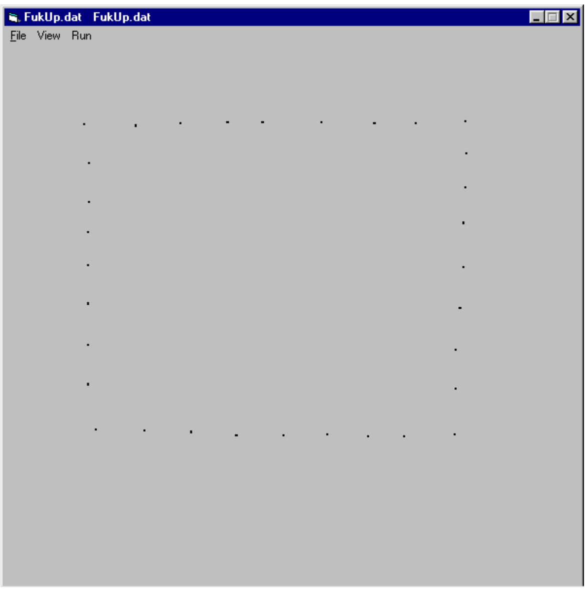


Figure 8.22: Hand-composed square.

Half-distance points

Half-distance points are located halfway between every possible pair of dots without replacement. This means that half-distance points are plotted from a dot to every other dot, and then that dot is excluded from consideration. Then half-distance points are plotted from another dot to every other dot, and that dot is excluded from consideration, and so on down to the last two dots; which results in $\frac{N^2 - N}{2}$ half-distance points.

Detection of less regular patterns is facilitated by examining coordinate sets of half-distance points for maximum frequency of visits to a limited region, and then working back to find from what dots they were produced. The limited region is a circular area related to overall dot density. It has a radius that is a small proportion of the mean nearest neighbour distance for the equivalent Poisson distribution (given by expression 5 in Chapter 3). This has been empirically tested at values in the range 0.125 to 0.0125 × $0.5\sqrt{A/N}$, and 0.05 × $0.5\sqrt{A/N}$ is typically used. Next the number of half-distance visits to each limited region, centred on each half-distance point, is counted as per the description given in Chapter 4, pages 64 and 65, for counting distance lags. And at length the region with the most half-distance visits is chosen. Working back to find from what dots half-distance points were produced is done for each half-distance point in the limited region as follows.

The distance of every dot from a half-distance point is calculated, and inserted into a list built in ascending order. In the process, the coordinates of the corresponding dots are indexed by the index to the list of ascending distances. The pair of dots for which each is equidistant from the half-distance point is then identified in the sorted list. This is done by indexing the list in ascending order until two consecutive, same values are found. The corresponding coordinates are then checked to see if they are diametrically opposed either side of the half-distance point. If not, which must be rarely the case, then the same procedure is available to move further along the list from there. Once identified, a pair of dots is transferred from the list to an array set up just for dots involved with structure.

In the course of joining structure dots in a nearest neighbour manner, the occasional outlier that might occur is identified as being far from a nearest neighbour, and is therefore discounted. Of course the whole procedure can be refined in many ways. Joining structure dots could be based on all Delaunay triangle edges, for example. However, this relatively high cost but simplistic version serves to illustrate the point.

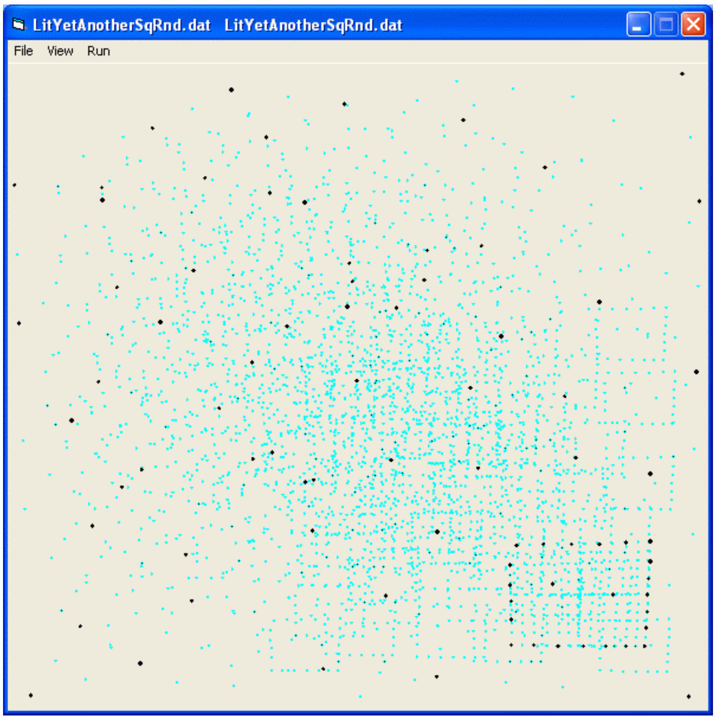


Figure 8.23: Hand-composed square in noise (black dots). The screen has been 'dusted' for half distances (blue flecks).

Figure 8.23 shows the computer display 'dusted' for half distances for explanatory purposes. The area near the centre of the rectangle is the most frequently visited (with considerable overlapping) hence the model will identify it by a cross. Note the half distances between random and structured elements. These are cross terms, and will be dealt with later when the detection of multiple structured objects in a field of noise is addressed.

For Figure 8.24 the same input file was used but the half distance dust is not displayed (although half distance still figured in detection). This time the 'pinpoint' and 'draw' options were selected, hence the model found the rectangle and then connected the relevant dots. The pinpoint option enables the model to find patterns offset with respect to form centre. The model connected the correct dots by working back to see what dots produced half distances of greatest density over a limited region.[2] The number next to the cross shows frequency of visits to the limited region.

[2] As stated earlier, precise patterns can be also detected using this principle, but the half-distance maximum visit region may be consigned to an approximation of a point instead of an area.

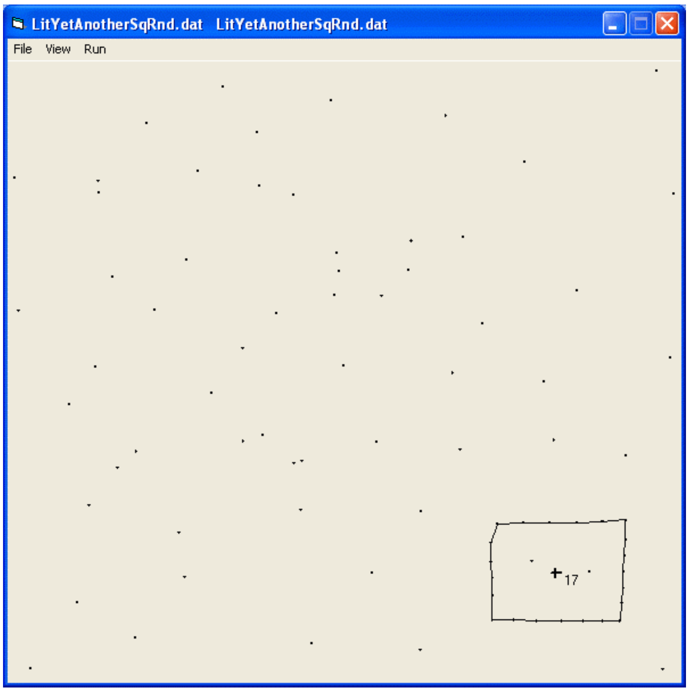


Figure 8.24: Hand-composed square in noise, pinpointed and drawn by the model.

Figure 8.25 shows a freehand elephant masked by noise. The elephant is pinpointed by the model. This example shows that the half-distance method can deal with coarse approximation to regularity. Note that some random dots overlie the elephant.

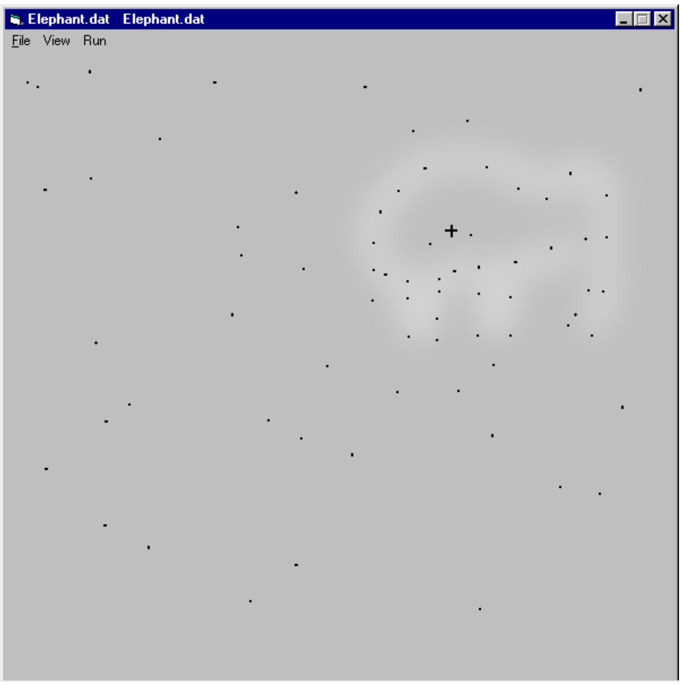


Figure 8.25: Elephant in noise, pinpointed by the model. (Outline highlighting is not part of the model response.)

Detection of multiple structured objects in a field of noise

Pinpointing multiple structured objects in a field of noise is implemented by examining coordinate sets of half-distance points for maximum frequencies of visits to multiple limited regions. Cross terms resulting from more than one structured object have significant half-distance densities, hence more processing is required.

Frequencies, representing numbers of visits to limited regions, are associated with maximum radii for the objects that produced them. Frequencies representing numbers of visits to limited regions *between* objects are associated with maximum radii greater than for those *within* objects. Before pinpointing regions of maximum frequency—and there could be more than one region with the same maximum frequency—checks for lowest associated maximum radii are performed. Objects tied in with these are then pinpointed. Once an object is pinpointed then no other pinpoint can be located within the object. Figures 8.26 and 8.27 provide examples of hand-composed patterns in noise, pinpointed by the model.

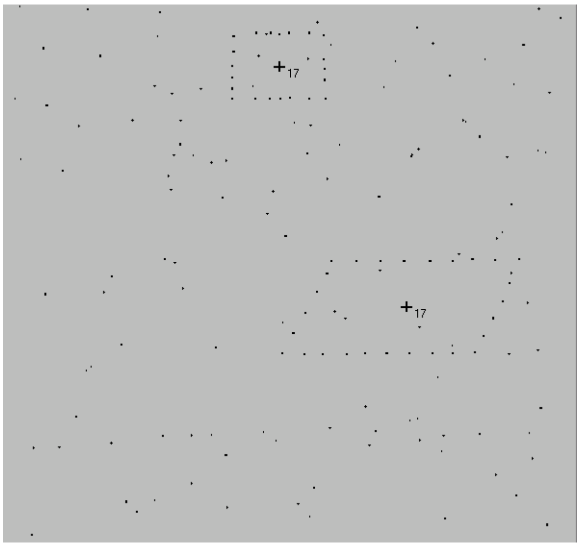


Figure 8.26: Hand-composed patterns in noise, pinpointed by the model.

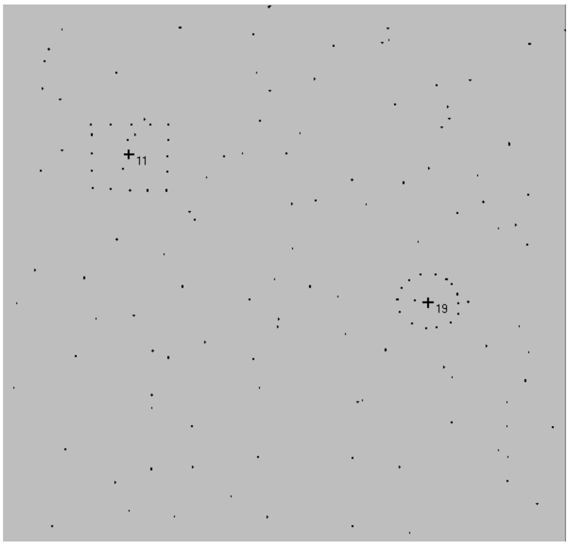


Figure 8.27: Hand-composed patterns in noise, pinpointed by the model.

The numbers next to the crosses show frequency of visits to the apposite limited regions. The maximum frequency, lowest maximum radii sites, indicated by the crosses, have been found by the model and, of course, are symmetric about the centres of the crosses.[3]

Selected screen shots

Figures 8.28 to 8.33 show the model's interpretation of several stimulus patterns. Figure 8.28 contains a circular pattern offset from the centre of the form, and embedded in noise. Figure 8.29 shows the model's response, with the pinpoint and draw options selected. Figure 8.30 contains a square pattern embedded in more noise than that of the previous pattern, and Figure 8.31 shows the model's response. Figure 8.32 contains a circular pattern embedded in even more noise, and Figure 8.33 shows the model's response. Throughout the series of patterns, increasing competition from the Gestalt proximity principle has not confounded detection outcome.

[3] Tuning of limited region size results in greater accuracy over different patterns, but has not (yet) been included as an automatic feature in the model.

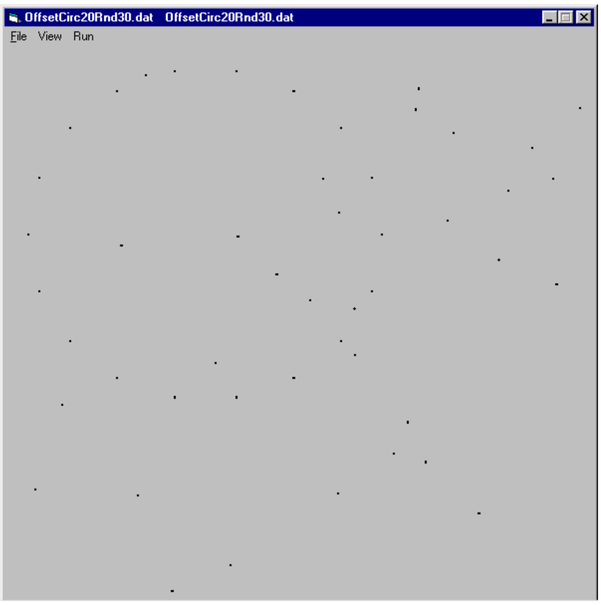


Figure 8.28: Stimulus pattern (corresponding to input file) of offset regular figure in noise.

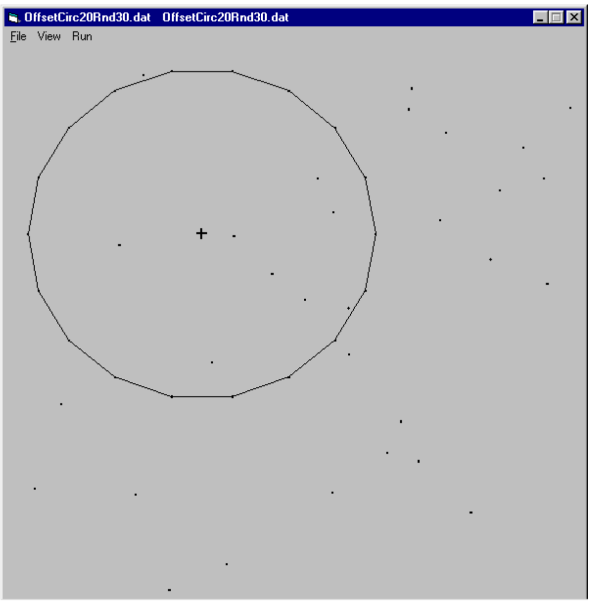


Figure 8.29: Offset regular figure in noise, pinpointed and identified by the model.

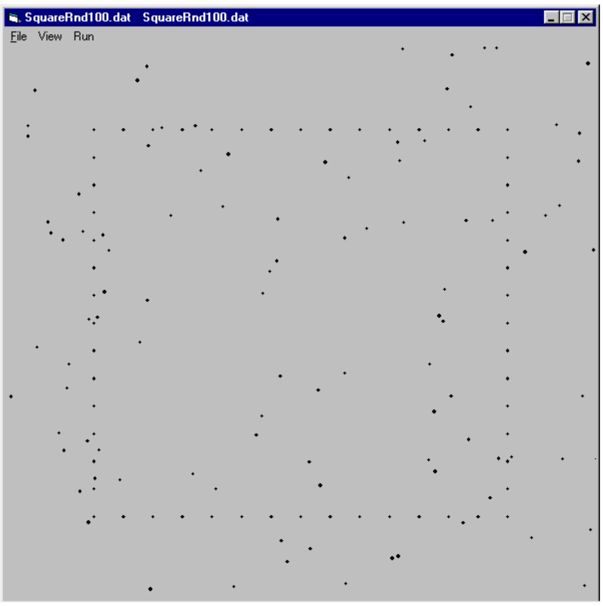


Figure 8.30: Stimulus pattern (corresponding to input file) of regular figure in noise.

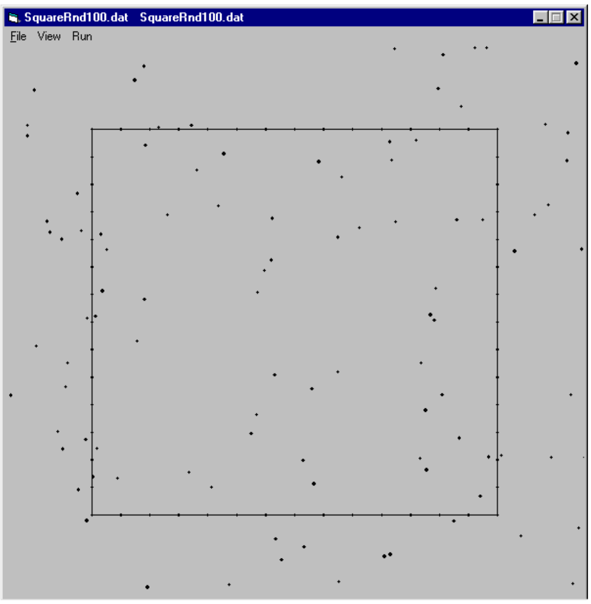


Figure 8.31: Regular figure in noise, identified by the model.

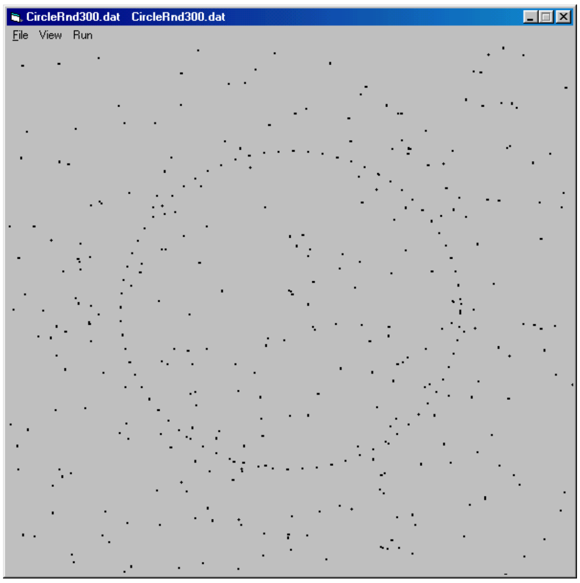


Figure 8.32: Stimulus pattern (corresponding to input file) of regular figure in noise.

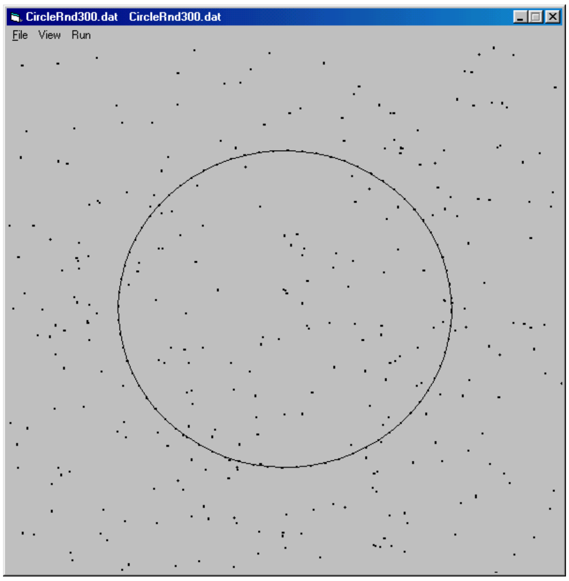


Figure 8.33: Regular figure in noise, identified by the model.

Figures 8.34 to 8.36 show the model's interpretation of a translation, a rotation, and a dilation Glass pattern. In each case there is competition from the proximity principle, which, again, has not confounded detection outcome.

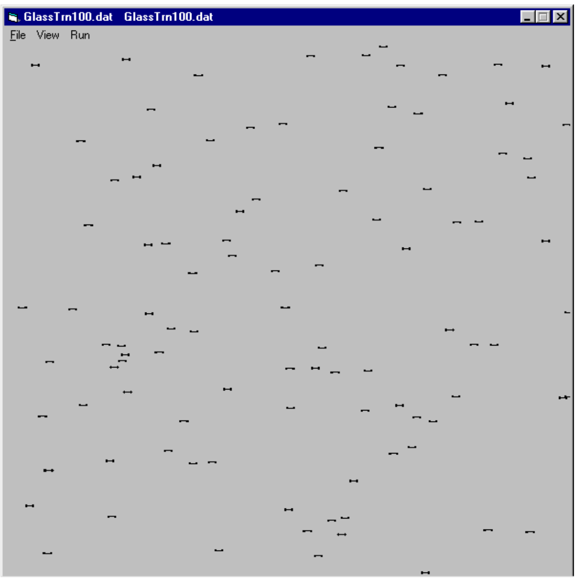


Figure 8.34: The model correctly connected the Glass translation pairs.

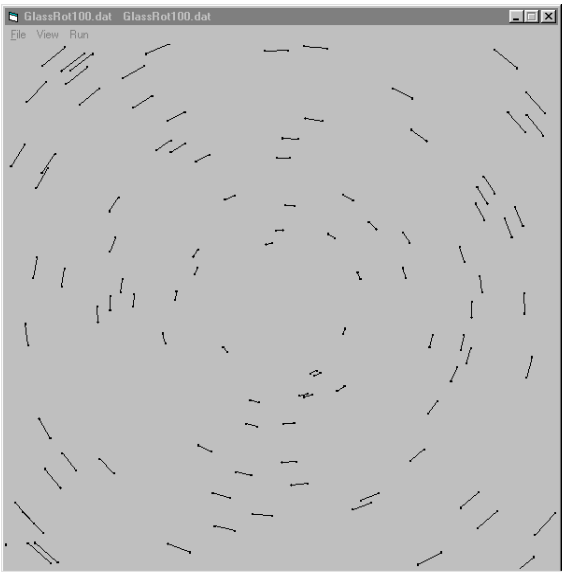


Figure 8.35: The model correctly connected the Glass rotation pairs.

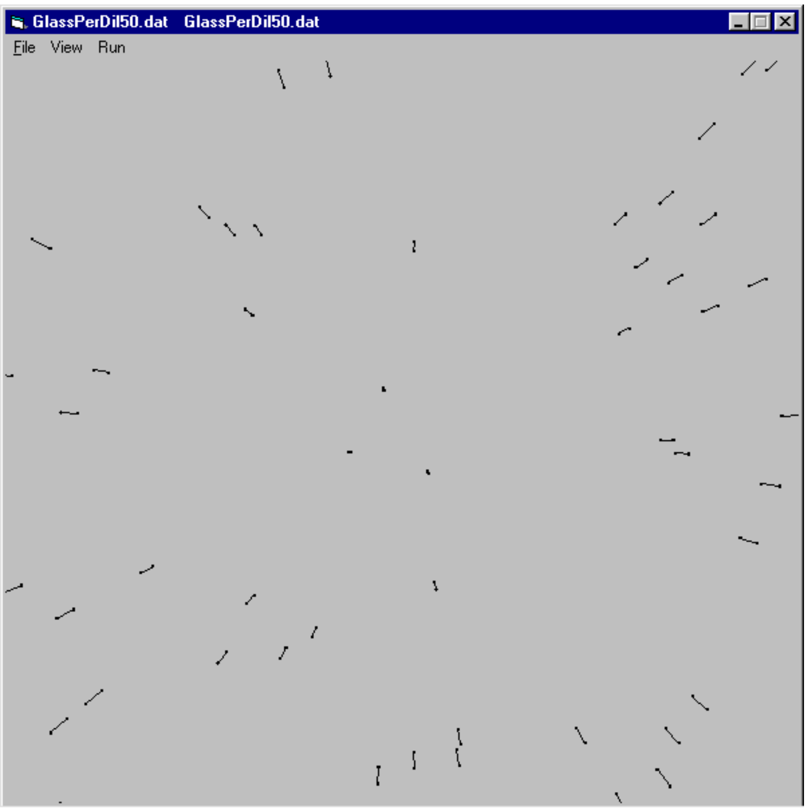


Figure 8.36: The model correctly connected the Glass perspective dilation pairs.

Figure 8.37 shows the model's interpretation of close concentric dots in noise. Again, the competition from the proximity principle has not confounded detection outcome.

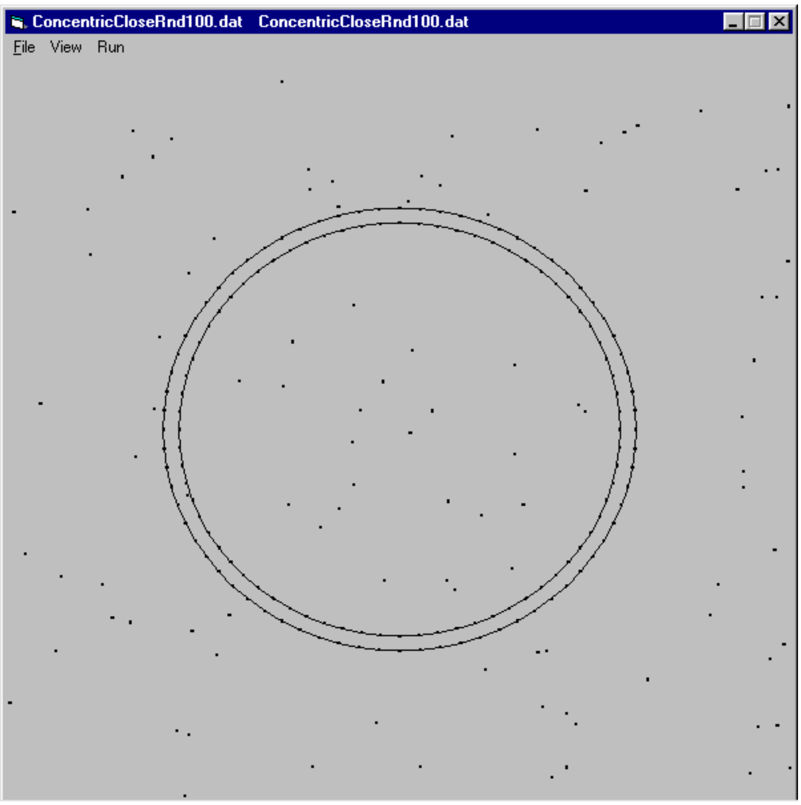


Figure 8.37: Close concentric circles in noise, identified by the model.

Finally, all the patterns depicted under the present subheading were generated using transformations. Polygons can be generated using translations and circles can be generated using rotations, for example. Patterns such as these simply consist of special arrangements of Glass pattern elements. Hence a single routine was used to detect and interpret all the patterns. This was the Glass pattern detection routine, which can serve as a basis for detection of a variety of low-level structures.

Hausdorff distance and the 3-D, 2-D, 3-D problem

Using seemingly limited information, the visual system is able to see the world. Visual information is made available at the retina in the form of a two-dimensional punctate array. This array corresponds to intensity values for separate points of light, encoded in the rods and cones. Yet, given this surface representation, the perceptual system can create a three-dimensional representation of the environment.

Movement makes the perceptual feat even more puzzling. Movement corresponds to the changing pattern of the array, and has to be organised into the correct motion despite the different organizations that could fit the changing pattern. How is a complex, two-dimensional, contorting, punctate pattern, which is encoded in the rods and cones, perceived as a constant, three-dimensional body, which is rotating for example?

Whether stationary or moving, a pattern can be highly ambiguous. The projection of a cube, for example, onto the retina nearly always makes an asymmetric retinal image, with edges of different lengths and angles all over the place, but on the three-dimensional cube in the environment the edges are the same length and the angles are equal. From a two-dimensional proximal image, the perceptual system infers the structure and dynamics of a corresponding three-dimensional body.

The process of image formation is what is known in mathematics as a forward problem, and is well understood in terms of optical physics. The process of vision is an inverse problem, and the solution to this inverse problem is apparently under-determined (Pizlo, 2001; Poggio, Torre, & Koch, 1985). An infinite number of appropriately constrained physical scenes could give rise to identical retinal images. As such, recovery of shape and structure is a problem for which it does not seem possible to devise an exact algorithm. It is generally regarded as an intractable, combinatorial optimisation problem.

Hausdorff distance and the visual system

Directed Hausdorff distance involving two sets of n elements, or one set of n elements in conjunction with itself, produces n sets of n distances, hence the algorithm is of order n^2. If the visual system used something similar, for arguments sake, then from a scene featuring $2n$ bits of information, n^2 bits of information would be represented on the retina. From n^2 bits of two-dimensional information, $2n$ bits of three-dimensional information may be reconstructed. Indeed $2n$ is the derivative of n^2, hence there would be first an integration from the environment onto the retina and then a differentiation from the retina onto higher levels of visual processing. The task of low-level visual perception would centre on dealing with rates of change, that is, with frequency differences associated with things, and this is central to the methodology outlined here.

Given a Hausdorff distance approach, the process of vision may turn out to be over-determined rather than under-determined. Some information could be omitted and still the visual system would cope. In order to see this, refer back to the greyscale charts of the square and circle in noise: see Figures 8.16 and 8.17, page 163. These derive from input patterns that are square-on to the viewer. The corresponding greyscale charts, Figures 8.38 to 8.41 below, derive from the same input patterns, only they have been rotated in depth (a common transformation encountered in daily visual experience). Figure 8.38 represents the square embedded in noise, all rotated in depth by 60°, and Figure 8.39 represents the same square embedded in noise, all rotated to the edge-on position; that is, by 90°. The same scheme applies to the circle in noise as shown in Figures 8.40 and 8.41.

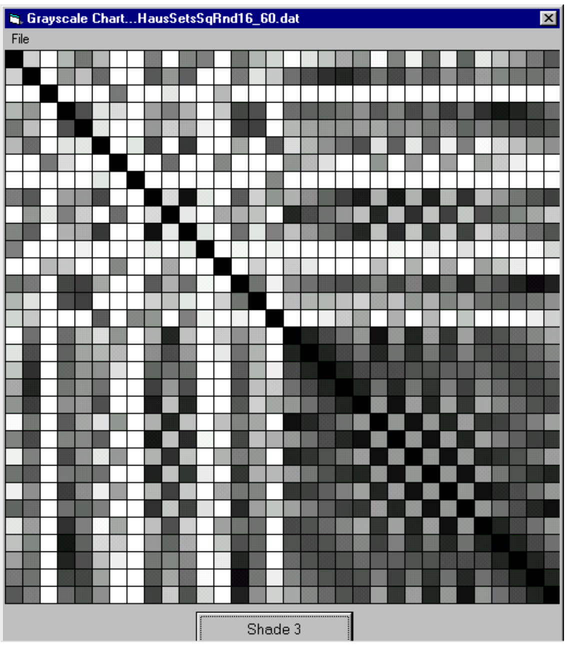


Figure 8.38: Square in noise rotated in depth by 60°.

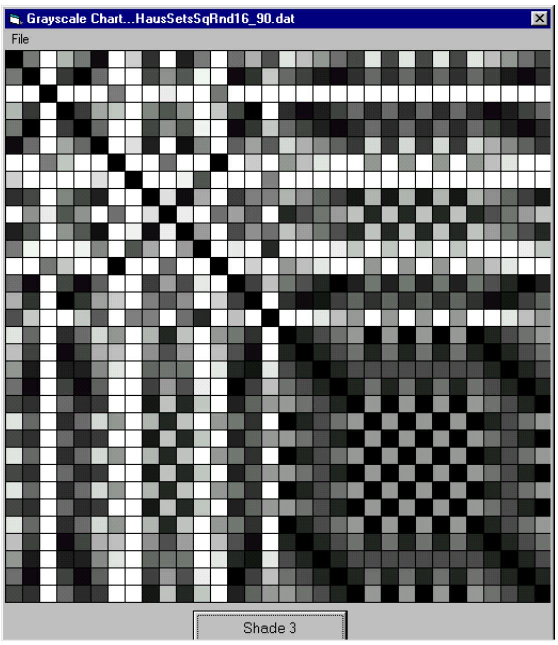


Figure 8.39: Square in noise rotated to edge-on position.

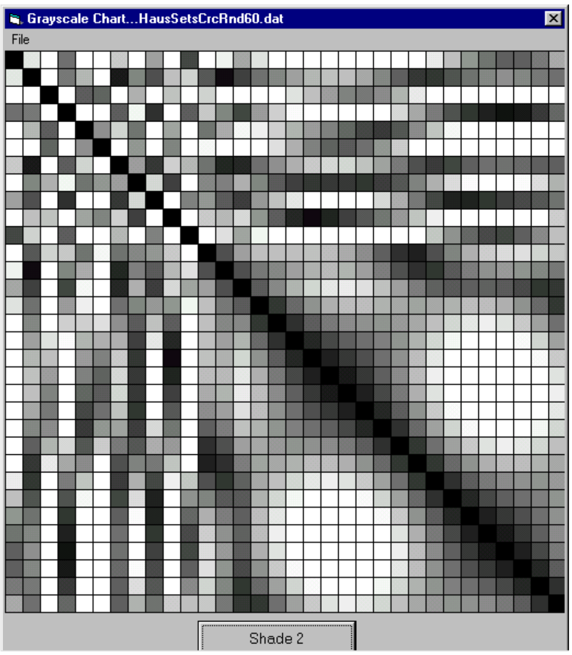


Figure 8.40: Circle in noise rotated in depth by 60°.

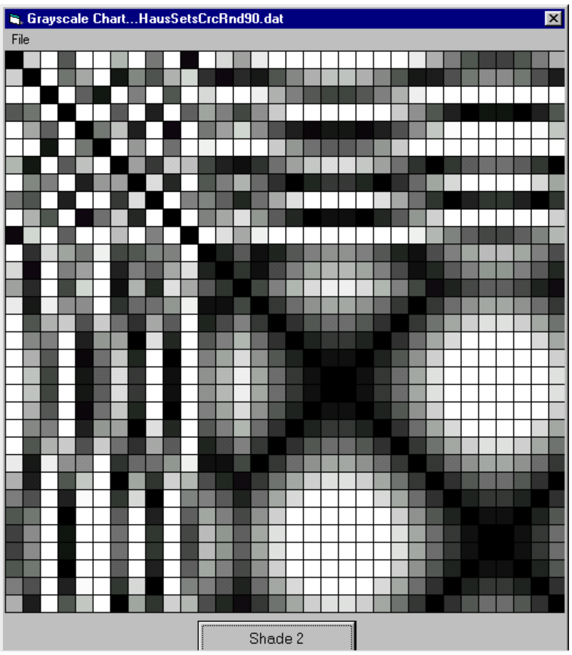


Figure 8.41: Circle in noise rotated to edge-on position.

Note that the greyscale charts remain unambiguous about what they represent. The square is still unambiguously a square and the noise is unambiguously noise (related to the square and noise in Figure 8.16; the square-on greyscale chart). The same applies to the circle in noise. Bits of the square or circle can be omitted and still the integrity of the greyscale charts is preserved. And this is particularly impressive for the edge-on transformed ensembles, since each appears as a dotted linearity on the computer display. Large transformations of the input pattern result in compressed changes in the greyscale chart. Hausdorff methodology has an important measure of invariance to a common image transformation associated with the visual system. It also has a conspicuous measure of invariance to shearing, and in the same context it is, of course, completely invariant to translation, planar rotation, dilation-contraction, and reflection.

Hausdorff methodology comes under the umbrella of 'distance measuring functions', and other distance measuring functions, in their various guises, are shown to be affine, projective, and Euclidean invariant (see Verri & Uras, 1994). Concerning Hausdorff methodology, as a set of points is affine transformed the absolute distances between points can change, but distance ratios (relative distances) have an important measure of invariance.

This measure of invariance is exemplified by Figure 8.42, which shows square-on and rotated views of a sparse version of the maple leaf shown in Chapter 7, page 124. Each view shows just two of many possible corresponding lines, each connecting one dot to another via an intermediate dot, for example. The ratios of the lengths of the lower two lines in each view are virtually the same and the ratios of the lengths of the upper two lines in each view are precisely the same. Degree of offset from colinearity of connected dots is accompanied by a principled proportion of ratio change.

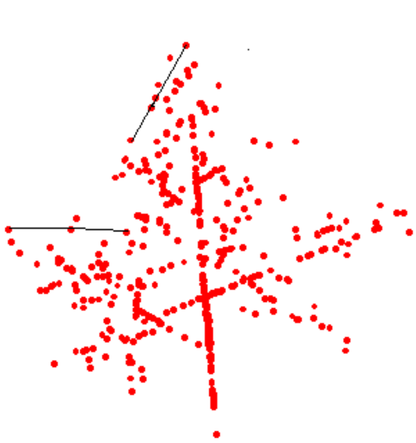


Figure 8.42: The skeletal maple leaf on the left is square-on to the viewer and the one on the right is depth rotated by 60°. The ratios of the lengths of the lower two lines in each view are virtually the same and the ratios of the lengths of the upper two lines in each view are precisely the same.

Figure 8.43 shows an outline of a leaf, with a line extending from left perimeter to right perimeter via its centroid. The ratio formed by the length of a segment of this line from perimeter to centroid, divided by the length of the whole line, for example, is invariant under affine transformations. If, by way of choosing another affine transformation, the leaf is planar rotated on its centroid over the stationary line that extends from perimeter to perimeter, a graph of ratios is obtained, as shown in Figure 8.44. This graph describes the planar rotational asymmetry of the leaf shape; and, by extension, the graph is affine invariant as well.

Of course any shape that is left-right and upper-lower symmetric gives a straight line graph at ratio = .5, hence the graph cannot be called a shape descriptor. It might be considered a shape classifier, however; especially since different symmetrical shapes can be contrived to give the same shaped graphs.

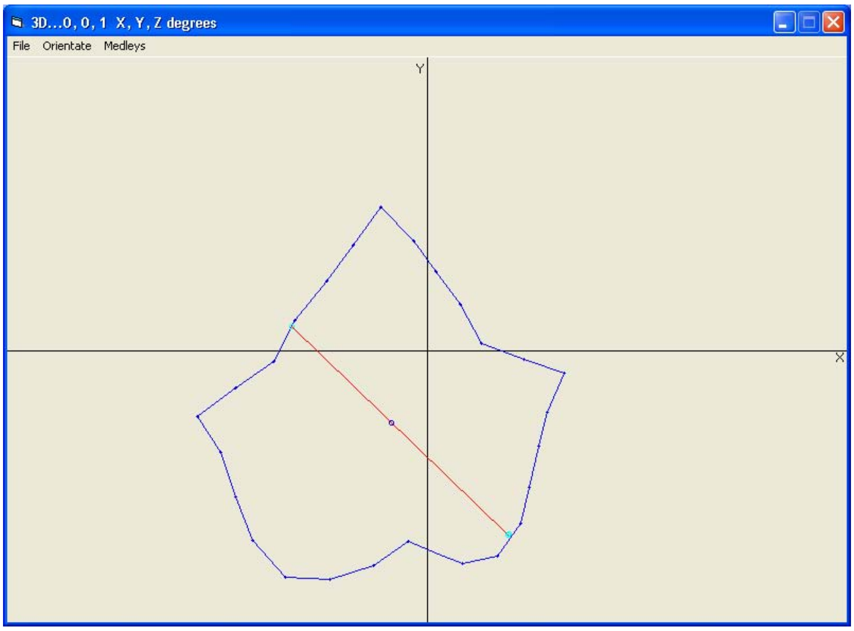


Figure 8.43: Outline of a leaf, with a line extending from left perimeter to right perimeter via its centroid. The ratio formed by the length of a segment of this line, from perimeter to centroid, divided by the length of the whole line, for example, is invariant under affine transformations on the leaf (after Verri & Uras, 1994).

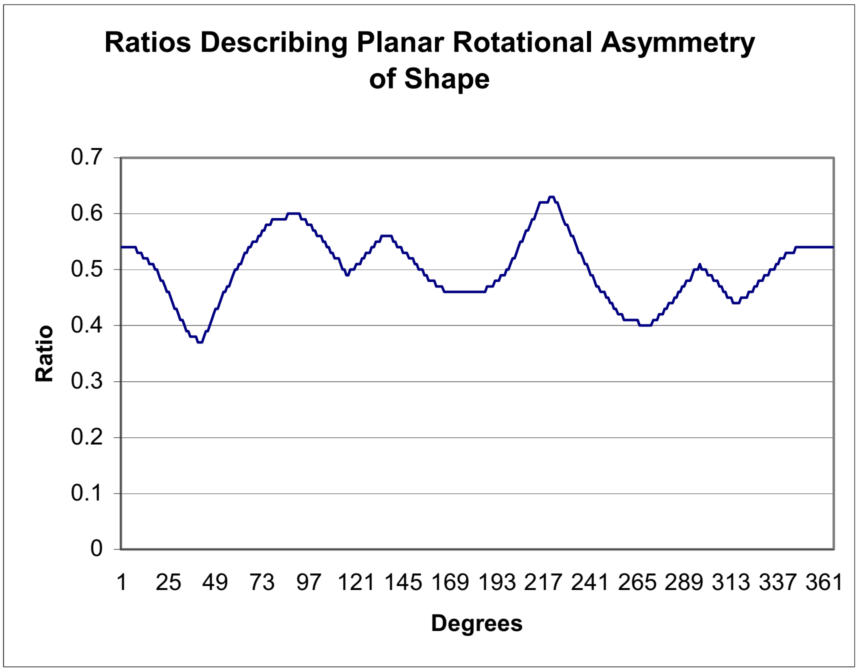


Figure 8.44: Graph of ratios obtained by planar rotating the leaf on its centroid over the line that extends from perimeter to perimeter. It describes the planar rotational asymmetry of the leaf shape. Note the left-right reflection symmetry (180° separated). Importantly, the graph is affine invariant.

From observing the greyscale charts, it could be argued that base results from the procedure used to calculate Hausdorff distance grant a useful size and shape constancy. These are not computational problems, but are guaranteed by the procedure. This suggests another way in which the visual system might recover three-dimensional shape from two views; or even from one view. All that is required is the assumption that, by applying suitable transformations, a shape can be found that preserves relative distance information of the original image, but maximizes the symmetry of the shape. This would mean, for example, that observers should have a tendency to perceive somewhat ambiguous retinal projections as produced by more symmetric shapes where possible. And this would explain why observers tend to see ambiguous projections as circles or squares rather than as ellipses or trapezoids (King, Meyer, Tangney, & Biedemann, 1976; Palmer, 1999, p. 329).

This feature of the Hausdorff metric may also explain the comparative ease with which human observers perceive the motion of a rigid structure from as few as two views. As suggested by Todd (1995), it seems likely that human observers make use of relative, rather than absolute, distance information.

Chapter 9: Dynamic Pattern Processing

Brief summary of chapter

A measure of relative symmetry for any shape is implemented by overlying one copy of a pattern set with an identical other copy and then rotating one copy about its centroid, all the while recording directed Hausdorff distance at appropriately small intervals of rotation. Relative symmetry is then calculated on the basis of repetition and depth of oscillations in directed Hausdorff distance.

The chapter then proceeds to treat dynamic—or temporal—patterns, as distinguished from static—or spatial—patterns, which were addressed in the last chapter. Successive frames over which there is some change of pattern are considered. Once detected, localized structures can be tracked as they are moved through fields of noise. They can be also tracked in a way that admits representational momentum, as outlined by Freyd and Finke (1984). Additionally, three-dimensional objects are addressed, and a tracking system is developed for them.

Again, a variety of conditions is treated so as to demonstrate the versatility of what amounts to just a handful of underlying principles. These principles, encountered in the last chapter, apply to both static and dynamic patterns. The work outlined in this chapter, with the exception of that indicated of others, is original. Particularly, the relative symmetry and tracking methods are original.

Relative symmetry

According to the traditional conception of symmetry, a figure either has a particular symmetry or it does not. However, most naturally occurring forms are characterized by imperfect symmetries that seem to vary in a continuous way. Figure 9.1 shows the change in *relative symmetry* with increasing rotational symmetry, as determined by Vickers, Navarro, and Lee (2000).

Their measure was based on results from a search for a transformation that maximized correspondence between an original configuration of dot stimulus and one or other randomly transformed version. Results were generated in a connection matrix by mapping locations from the original stimulus for which nearest neighbour distances to transformed versions were less than some tolerance. Values in the connection matrix were weighted according to the number of mappings made to a location, and then normalized. Although structure was summarized via the connection matrix, the issue of concern here involves a meaure for relative symmetry. Their measure was determined in the range 0 for no mappings in the connection matrix, to 1 for the maximum possible mappings.

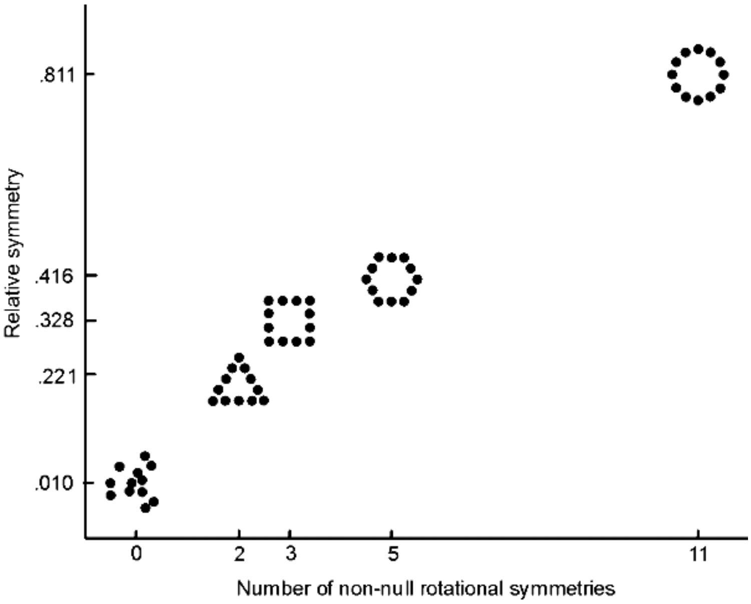


Figure 9.1: Change in relative symmetry with increasing rotational symmetry. (Based on Vickers, Navarro, & Lee, 2000.)

The generative transformation model also provides a continual measure of relative symmetry. A measure for any planar representation is implemented by reference to the number and depth of oscillations in directed Hausdorff distance. By overlying one copy of a pattern set with an identical other copy and then rotating one copy in the plane in small increments, of 1° say, about its centroid (coordinate mean), an oscillating sequence of h(A→B) is produced. The sequence comprises the directed Hausdorff distances from one copy to the other during rotation. An *initial* measure of relative symmetry is then directly related to the number of oscillations in the sequence.

This measure is normalized in the range 0 to 1 by fixing a reference. In rough accordance with the values determined by Vickers, Navarro, and Lee (2000), a circle represented by more than 10 equally spaced dots has been chosen to have a relative symmetry of 1, with a random distribution of dots having a relative symmetry of 0 in principle.[1] Since a circle composed of more than 10 equally spaced dots has at least 10 non-null rotational symmetries, a regular hexagon then has a relative symmetry of .5 (because it has five non-null rotational symmetries).

The averages for the maximum and minimum excursions of directed Hausdorff distances are then calculated, from which the absolute difference average between the two excursions is established. Finally, the initial measure of relative symmetry (normalized on the number of h(A→B) oscillations) is multiplied by the proportion of the maximum excursion average indicated by this difference. That is, the initial measure of relative symmetry is multiplied by the ratio formed by the difference average divided by the maximum excursion average. In equation form

Relative symmetry = Initial relative symmetry × (maximum excursion average – minimum excursion average) / maximum excursion average.

[1] Of course the limit for the number of equally spaced dots representing a circle of relative symmetry equal to1 could be increased according to the number of desired increments in the range 0 to 1.

If all the minimum excursions extend to zero, i.e. the minimum excursion average is zero, then relative symmetry is proportional only to the number of oscillations (because the ratio equals one). Minimum excursions further and further from zero progressively degrade relative symmetry (because the ratio becomes progressively less).

Figures 9.2 to 9.6 show a selection of waveforms along with the polygon views, defined on the unit square, from which they were produced. These were generated by overlying one copy of the corresponding polygonal dot patterns with an identical other copy and then rotating one copy in the plane in small increments about its centroid. All waveforms intrinsically relate to two-dimensional figural symmetry.

The difficulty is one of accurately quantifying a waveform with a single real number; namely *relative symmetry.* Differences between single real numbers need to reflect differences between corresponding waveforms accurately. Figures 9.2 and 9.5 are special, maximally symmetric, views of the tetrahedron and cube respectively. The relative symmetry measure for the tetrahedron view of Figure 9.2 is .200, as it should be. However, the relative symmetry measure for the cube view of Figure 9.5 is slightly less than .500, when it should be exactly .500. In rotating the tetrahedron view around the centroid, the stationary view was effectively eclipsed at 120^{o} rotation offsets. For the cube, complete eclipse occurred only at 180^{o}. Owing to a small height to width asymmetry of the dot pattern, eclipses expected at other rotation offsets of 60^{o} were either incomplete or the dots narrowly missed one another. Hence directed Hausdorff distance only came close to the reference minimum for all but the 180^{o} position, where it effectively reached the reference minimum.

Another difficulty arises when considering relative symmetry *between* polygonal categories. Is the 'relative symmetry' of the cube view depicted in Figure 9.5 really greater than the 'relative symmetry' of the tetrahedron view depicted in Figure 9.2? Nonetheless the measure appears to be consistent *within* polygonal categories.

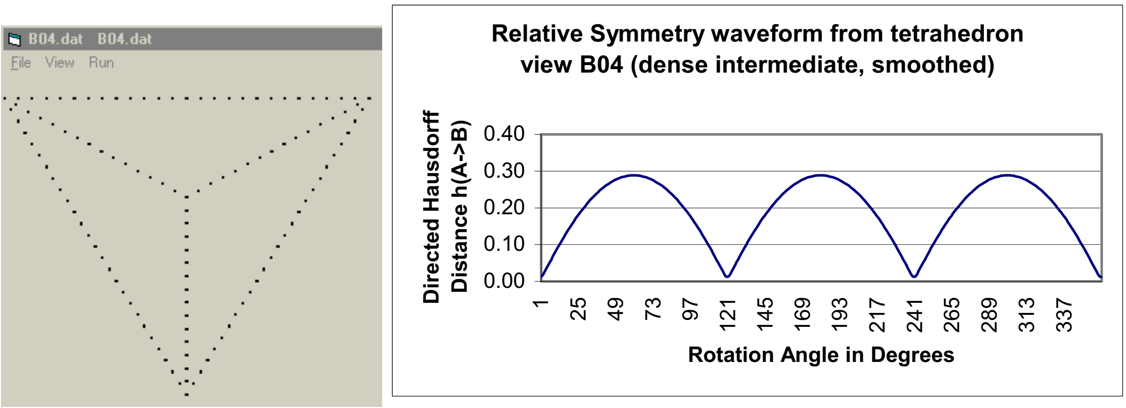


Figure 9.2: Left - View of tetrahedron. Right - corresponding waveform. Relative symmetry = .200.

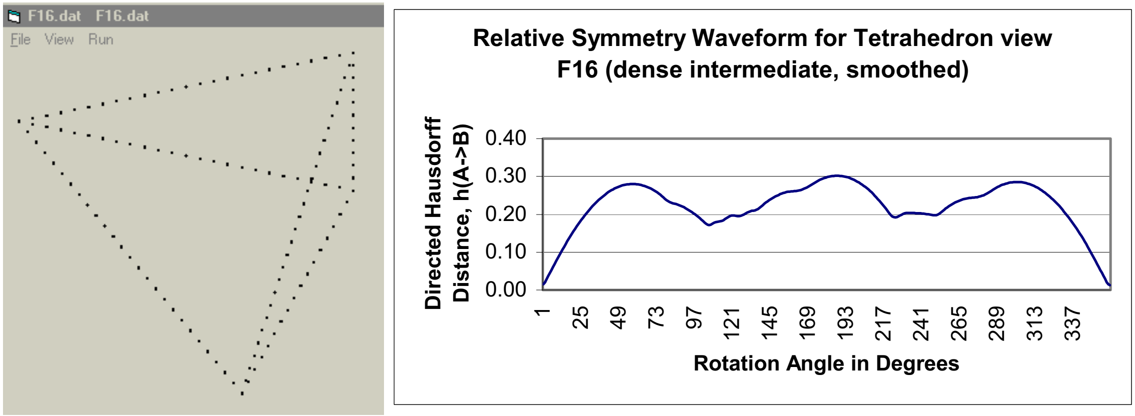


Figure 9.3: Left - view of tetrahedron. Right - corresponding waveform. Relative symmetry = .107.

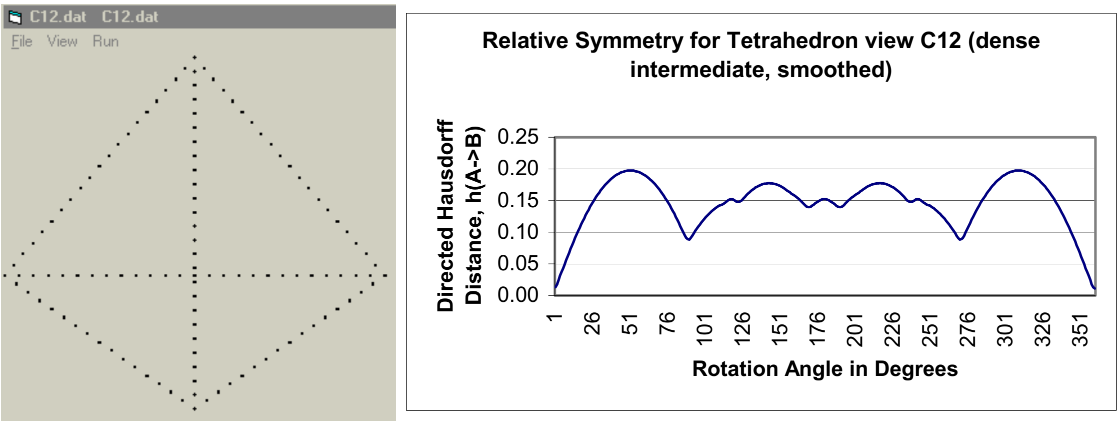


Figure 9.4: Left - view of tetrahedron. Right - corresponding waveform. Relative symmetry = .175.

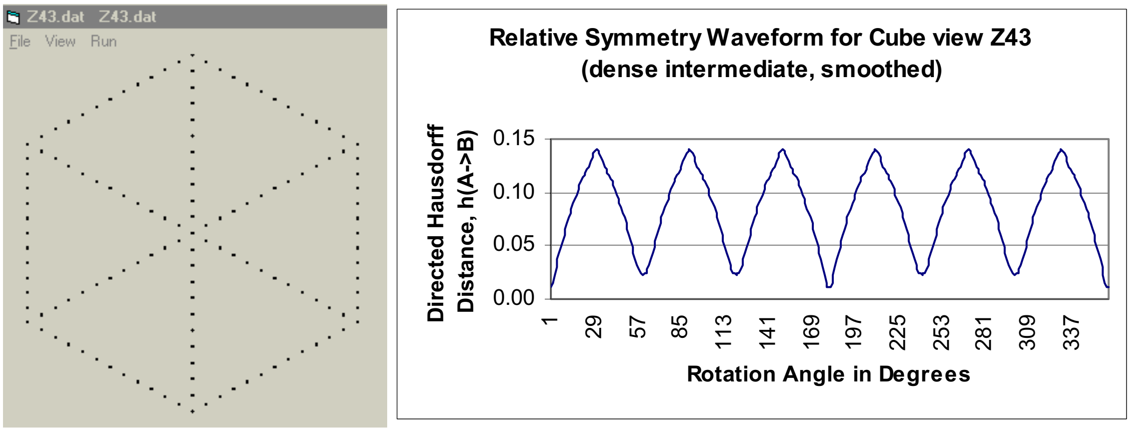


Figure 9.5: Left - view of cube. Right - corresponding waveform. Relative symmetry = .460.

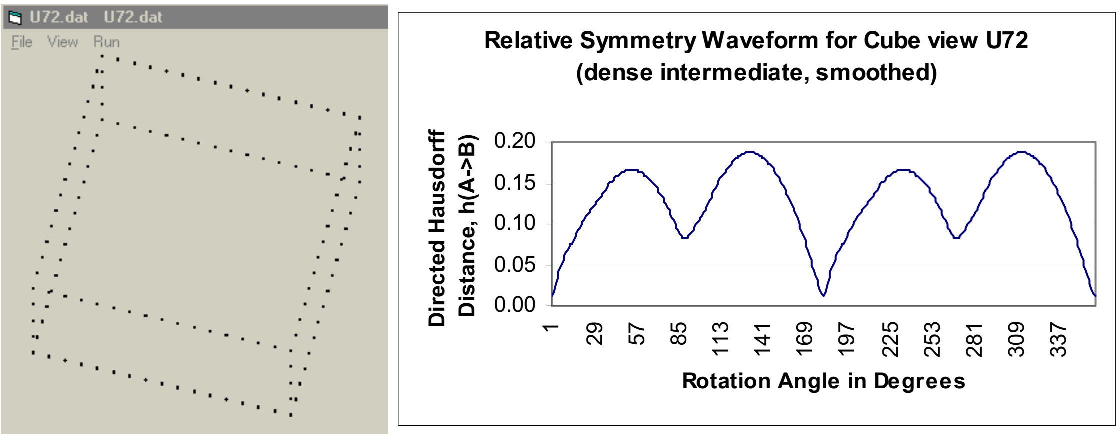


Figure 9.6: Left - view of cube. Right - corresponding waveform. Relative symmetry = .213.

Owing to precision limitations, some original polygon figures employed by the relative symmetry process have small errors in geometry. Consequently, detection of waveform extrema related to figural form is sometimes confounded by waveform glitches caused by minor misalignments of dots. This can be largely overcome by waveform smoothing, which is attained by taking the average of a small range of values around, and including, each point in the original waveform and then generating a new waveform. The range of values utilised for averaging consists of some *odd* number of elements, *n,* with the point of interest as the midpoint.

The beginning and end of the waveform require special treatment. The beginning point—corresponding to 1^{o} of rotation offset—is calculated by taking the last (n – 1) / 2 points of the waveform and then including these in the averaging process with the point itself and the next (n – 1) / 2 points. Similarly, the end point—corresponding to 360^{o} of rotation—is calculated by taking the (n – 1) / 2 points preceding the point, along with the point itself, and then including these in the averaging process with the first (n – 1) / 2 points.

The smoothing process can lift the minima and drop the maxima a little. Importantly the reference point corresponding to 0^{o} of rotation offset (end point) can be lifted from zero to some magnitude of directed Hausdorff distance. If so, then this becomes the reference minimum, instead of zero, to which minimum waveform excursions can extend.

The width of the smoothing window (the number of points included in the waveform averaging process) is a parameter that can vary proportionally to the density of dots in a figure. Another parameter that can vary proportionally to dot density is the rotation increment. Nonetheless these two are mostly fixed at *five* and *one* respectively, in this particular application. A parameter that always varies proportionally to dot density is the search width for extrema: the number of points checked each side of a point for monotonic increase, decrease.

Perception of orthographic projections as two- or three-dimensional

In an early experiment, Hochberg and Brooks (1960) showed that the tendency to see outline figures as two- or three-dimensional was a function of the number of angles and line segments required to specify them in two or three dimensions. The approach followed by Hochberg and colleagues has been widely cited, and is still considered very successful. It is represented as one of the first expressions of a form of minimum hypothesis; namely that visual perception consists of an economical coding of the information provided by patterns of retinal stimulation (e.g., Hatfield & Epstein, 1985; Pomerantz & Kubovy, 1986).

However, one problem with such an approach is that it does not provide an account of the psychological processes by which the perceptual system might achieve the hypothesised optimisation in question. Another problem is that the figures investigated by Hochberg and later researchers have been selected in an unprincipled way, so that no conclusion can be drawn for a general class of figures. Consequently, there are no strong grounds for supposing that theoretical conclusions should apply beyond the particular set of stimuli examined.

This problem can be addressed by employing stimuli that are representative of major classes of geometrical object. Specifically, within the general class of polyhedra an important sub-class of objects are the regular polyhedra, otherwise known as the Platonic solids. They are the cube, tetrahedron, octahedron, dodecahedron, and icosahedron. These are the only five solids bound by regular, convex polygons that have the same number of polygons meet at each vertex. However, despite their geometric and historical importance, it is only recently that a subset of these regular polyhedra has received explicit attention in perceptual research (e.g., Pani, Zhou, & Friend, 1997). Arguably, an assessment of any approach to the perception of ambiguous outline figures should apply at least to a representative sample of projections of the regular polyhedra.

According to a transformational approach, the perception associated with more symmetry-preserving transformations will occur more readily than one associated with fewer such transformations. If we adopt a principle of symmetry maximisation, then we need to specify what is to be made more symmetrical. One possibility is that it is the symmetry of the object in question (whether two- or three-dimensional) that counts. A second possibility is that it is the symmetry of the projected image that is to be maximised. On the first hypothesis, we should expect that all projections of an outline cube, similar to Figure 9.6, should be seen as three-dimensional cubes, rather than as two-dimensional patterns. However, this does not account for our tendency to see Figure 9.5 as a flat, hexagonal figure, rather than a cube.

For the present, therefore, it seems to be worth pursuing the second hypothesis: namely that the human visual system seeks to find some specification of an imaged object that, under a suitable set of transformations, can project a two-dimensional image that is as symmetrical as possible. In other words, an imaged object is seen as belonging to different families of possible projections arising from different objects. It is seen as belonging to the object-family that, under a suitable transformation, will produce the most symmetrical projection.

The following unpublished investigation (Vickers, n.d.) was designed to explore the transformational approach by employing a set of outline figures, based on systematically varying orthographic projections of the tetrahedron and the cube (the first two Platonic solids). The hypothesis tested was that the observer's preference for a two- or a three-dimensional interpretation would be predicted by the continuous measure of relative symmetry as determined by the transformational approach of Vickers, Navarro, and Lee (2000).

Experimental method

A group of 50 student subjects was presented with computer-generated views of families of projections, one at a time, and were asked to assign a rating of either two- or three-dimensionality to each.

Two sets of stimuli were employed. The first set comprised 16 orthographic projections of an outline cube. The second set comprised 18 orthographic projections of an outline tetrahedron. The outline polyhedra were rotated by varying amounts, first around the *z*-axis and then around the *x*-axis. In the case of the cube, the figure was rotated by multiples of 15^o around both *z*- and *x*-axes. The tetrahedron was rotated by multiples of 15^o about the *z*-axis and by multiples of 22.5^o around the *x*-axis. In addition, for the cube a further rotation of 45^o around the *z*-axis and 35.26^o around the *x*-axis was included because of the obviously high degree of symmetry possessed by this two-dimensional projection.

Each projection was printed about the centre of a sheet of A4 paper, so that the maximum extent of each projection in either the *x*- or the *y*-dimension was a constant. At the foot of each page was an 11-point bipolar Likert scale, ranging from –5, through 0, up to +5, in steps of 1, which subjects used to record their ratings. Each set of projections was presented as a booklet.

Half the subjects made judgments on the tetrahedra first and then on the cubes, while the other half of the subjects made their judgments on the cubes first and then on the tetrahedra. Both sets of judgments were then repeated. Within each set of projections, the stimuli were presented in a random order that was different for each subject and each presentation.

Results

Mean ratings of two- or three-dimensionality for each projection of the cube were 2.71 (SD = √2.37) and 2.46 (SD = √2.56) for sessions 1 and 2, respectively. When compared across the two sessions, mean ratings showed a small, but significant, decrease in judged three-dimensionality (t = 3.35; p < .005; two tails, related samples). However, mean ratings also correlated highly across the two sessions (Pearson r = .98).

Mean ratings of two- or three-dimensionality for each projection of the tetrahedron were 0.15 (SD = √4.42) and 0.12 (SD = √4.65) for sessions 1 and 2, respectively. When compared across the two sessions, mean ratings showed a slight, but insignificant, decrease in judged three-dimensionality (t = 0.43; p > .675; two tails, related samples). However, mean ratings of three-dimensionality correlated almost perfectly across the two sessions (Pearson r = .99).

As can be seen from these figures, mean ratings of judged tridimensionality of the tetrahedron were markedly lower than those for the cube, in agreement with the mean approximate (two-dimensional) symmetry measures produced by the model. In addition, predicted symmetry measures correlated well with judged tridimensionality (Pearson r = -0.74 and -0.51 for the cube and tetrahedron; N = 16 and 18, respectively; p < 0.05 in both cases).

Discussion

Results show that the proposed measure of approximate symmetry provides a reasonably good prediction of the tendency to see a range of orthographic projections of a cube and a tetrahedron as three-dimensional figures, and is consistent with the generally stronger tendency to see the cube projections as three-dimensional. For an indication of the utility of symmetry in the tendency to see outline figures as two- or three-dimensional, see the submission by Vickers and Preiss (2000), Appendix C.

Despite the results, we are not convinced that the measure of relative symmetry captures the full complexity of judgements of bi- or tri-dimensionality. There may be better predictive measures based in Hausdorff distance. In particular, it might be usefull to study how the distributions of inter-point distances behave when a structure is transformed in such a way as to maximize its symmetry under the constraint that its Hausdorff contour remains constant.

Pattern detection

Notwithstanding pattern detection involving reconciliation of spatial and/or temporal relationships based on proximity and relative frequency, pattern detection can be also implemented using reconciliation of transformations over images, as per Chapter 7. Pattern sets, originally with identical boundaries and one set overlying the other, can be decomposed by transforming one set with respect to the other. For a circle, rotation about its centre results in repetition of h(A→B). For a square or grid, orthogonal translations result in repetition of form at increasing magnitude of h(A→B)p. Contractions similarly reveal structure. The idea is to try various transformations until some degree of repetition is observed, and then attempt to optimise that repetition. Again, the half split method may be used if necessary. Structure is then discerned by the transformation(s) required to optimise repetition.

Further to this, an object undergoing transformation against some randomly moving background has its structure revealed by some form of consistency of distances and directions of elements between changing frames. This will be dealt with presently under the heading 'Detection Based on Structure of Motion'.

Transformations and the human perceptual system

Transformations are utterly transparent to the human perceptual system, which recognizes multifariously transformed versions of an object. It also tracks a transforming object. Some common transformations include translation, scaling, and rotation. Consider the simple act of walking past a park bench, for example. On approach its distance reduces, its scale becomes larger, and it rotates in depth as well as downwards, relatively speaking, more rapidly with observer proximity. Driving a car involves continuous response to transformation. Transformations are effectively intrinsic to relative motion, hence it is worth including a related dynamical capability in the model.

Tracking

Once detected, a localized structure can be tracked as it moves repeatedly from frame to frame; even as it moves through a field of noise. An object can be tracked as it undergoes any or all of the affine transformations. Tracking involves the generation of terms indicative of error between the present transformational state of an object and its last transformational state. Error terms are then systematically reduced beyond a small arbitrary threshold, which brings the last state into operative alignment with the present state. Large error terms produce coarse tracking movements and small error terms produce fine tracking movements. Given this scenario, visually following a moving object corresponds to tracking the object. Tracking movements remain proportional to error terms; they become smaller with proximity.

Centre-of-mass tracking

For translations, tracking is predicated upon 'centre of mass', and for other affine transformations tracking is predicated upon shape and size matching. Tracking something that moves from one location to another is accomplished by a transformation that reduces a degenerate form of Hausdorff distance between the centroids of the successive instances. Since only single points for the last and present location states are considered, Hausdorff distance is simply the Euclidean distance between them.

The centroid corresponding to the last location state is moved in the direction of maximum rate of distance reduction. Distance is reduced beyond some small arbitrary value, which brings the centroid corresponding to the last location state into operative alignment with the centroid corresponding to the present location state. (Of course all transformations are checked for their respective null transformations first, as effectively zero errors indicate termination of tracking.) Figure 9.7 shows an instance of an object that has moved from the last transformational state at upper left, dashed outline, to the present transformational state at lower right, solid outline. The object has been tracked from the last location to arbitrarily close to the present location.

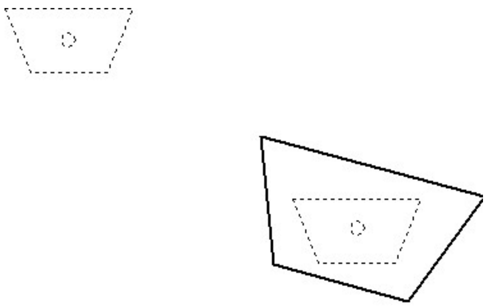


Figure 9.7: Instance of an object that has been tracked from the last location at upper left to arbitrarily close to the present location at lower right.

Shape and size matching

Shape and size matching is addressed by rotations in the plane, horizontal and vertical depth rotations, and dilations and contractions, with combinations of these and translations providing compound transformations. If an object rotates out of position in the plane, the last instance can be can be operatively re-matched with the present instance by performing transformations that reduce Hausdorff distance from the last to the present instance (or from the present to the last instance). The transformations continue until Hausdorff distance becomes less than some small arbitrary value. Transformation magnitude is proportional to Hausdorff distance. The instance of the object at the lower right of Figure 9.7, dashed outline, appears at the upper left of Figure 9.8, which shows the present transformational instance of the object in solid outline. Symmetry of an object is implicitly considered in that Hausdorff distance falls off as any matching view is moved to alignment, not just the original view.

Since direction is not implicit in Hausdorff distance (it is a scalar quantity) then some method for addressing transformation direction is required. Any of the transformations listed in the previous paragraph has just two possible, and complementary, directions. It is a simple matter of selecting a direction and performing a small test transformation to see what Hausdorff distance does.

It can increase in one direction and decrease in the other. It can also increase or decrease by the same or different amounts in both directions. If it increases in one direction and decreases in the other, which is commonly the case, then the direction of decrease is implemented in the tracking transformations. If both directions indicate increases on the original, then the magnitude of the test transformation is incremented and the test transformation is tried again and so on, until a decrease in one or both directions is forthcoming. If the decrease is in one direction, which, again, is commonly the case, then that direction is implemented in the tracking transformations.

If, at the outset, or after the procedure just described, both directions indicate different decreased results then the direction of the lowest is chosen. If, however, both directions indicate effectively the same decreased result over several trials then the object is either 180^o rotated and symmetric, or has some odd complementary asymmetry. In this case one direction of tracking transformation is consistently chosen.

The scheme just outlined is necessary to deal adequately with contingencies in planar rotations. If an object rotates in depth or changes size, then it can be operatively re-matched by using just part of the above scheme. Where there is an increase in one direction and decrease in the other, the direction of decrease is chosen. In the event of an uneven decrease in both directions, the direction of lowest decrease is chosen. In the event of an even decrease, a consistent direction is chosen. Unlike tracking transformations, test transformations are not displayed.

The tracking strategies outlined, together allow for tracking the affine transformations in a way that minimizes tracking movements. This may be useful in explaining certain illusions, and is useful in demonstrating the wagon-wheel illusion; in which the spokes appear to reverse direction.

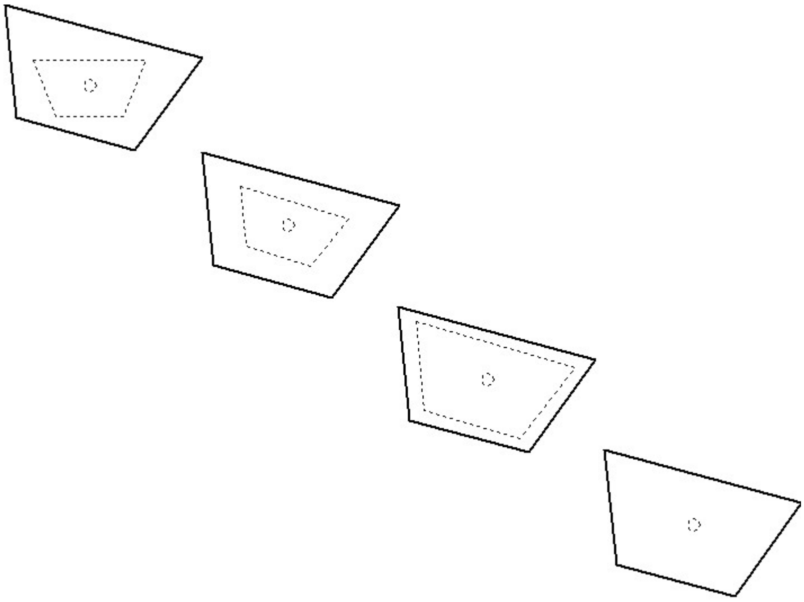


Figure 9.8: After locating, as per Figure 9.7, planar rotation and scaling are invoked by the Hausdorff reduction procedure to bring the last instance of the object into operative alignment with the present instance. When this present instance of the object undergoes further transformation, it will be immediately brought into alignment with the further transformational state by the Hausdorff reduction procedure, which applies transformations that reduce the tracking error beyond some small arbitrary value.

Representational momentum

Representational momentum (Freyd & Finke, 1984, 1985) is the tendency to mentally continue some implied motion or rigid transformation of an object. In a prototypic experiment, Freyd and Finke presented subjects with a sequence of displays in which a rectangle was progressively rotated in one direction by equal increments, and in a related experiment the rectangle had a mixed order of rotational direction. Subjects were then presented with the rectangle either at the same orientation as that last seen, or further rotated by a small amount in either direction, upon which they were required to respond ‘same’ or ‘different’. In the progressive rotation situation, subjects found it much harder to detect differences when the test rectangle was presented further rotated in the direction of implicit motion. In the mixed rotation situation (in which there was no such implicit motion) they indicated no directional bias in detecting differences.

Representational momentum is consistent with a generative transformational process designed to capture structure in a sequence of stimuli in terms of the simplest set of transformations required to map a stimulus at one instant onto the stimulus at a later instant. And objects can be tracked in a way that admits representational momentum.

The tracking system outlined above is modified such that it takes into account the history of target movements. Such history can involve just the last movement, or a string of contiguous movements up to, and including, the last movement. Patterning on the basis of history tends to stabilize the tracking system by way of habituating it to regularities in transformations made by a target. And the history has the effect of biasing tracking situations in which a target moves in an unexpected manner: hence simulating representational momentum. The tracking system tends to fall short of, or overshoots, the target.

Detection based on structure of motion

Grzywacz, Watamaniuk, and McKee (1995) have studied the perception of consistent motion of a single dot in a field of randomly moving dots generated by a rapid succession of static images. Although people can detect such motion, it poses a severe problem for established models (e.g., motion-energy and Reichardt models) because of the dilemma concerning which dots in one frame correspond with which dots in a subsequent frame.

Grzywacz et al. propose an elaborate, computationally intensive model, consisting of multiple arrays of detectors. However, the model has numerous explicit (as well as implicit) parameters, is slow and unwieldy, and needs to know in advance what the extent and nature of the motion will be (e.g., straight or circular). The generative transformation model using the Hausdorff procedure—an n^2 computation at worst—solves this problem easily and in real time.

The last chapter dealt with pattern detection that operates on the basis of spatial, or static, structure (i.e., on regularities that exist at once). The opposite consideration is that of no spatial structure, only structure over time due to form of motion. (Of course, the environment generally exhibits both kinds of structure in varying degrees and in concert.) Any single display in a sequence with a dot moving in a regular fashion against a background of identical dots moving in random fashion provides no spatial structure. Temporal, or dynamic, structure emerges during the sequence.[2]

Consistent increments of movement, consistent direction, and consistent change in direction, all within some adjustable spread (tolerance), can be discerned by the model. (In fact any change for which a transformation search finds consistency can be considered structure.) Each time the model display is updated, sets of distances—each set containing distances from a different point in the previous pattern set to all points in the current pattern set or vice versa—are produced upon calculation of Hausdorff sets. It turns out that successive pairs of pattern sets (1-2, 2-3, 3-4,…, *n*–1-*n*) yield sets of distances that, again, highlight structure. (For ease of comprehension, patterns 1 and 2 can be pictured as being on the same display, upon which Hausdorff sets are calculated from the dots in pattern 1 to the dots in pattern 2 or vice versa, then patterns 2 and 3 are likewise considered, and so on.)

It is not difficult to see that over a number of successive display updates the only consistencies across all sets of distances are those due to the consistently moving dot. All other distances show random variation. A frequency count of the elements (or some transformation thereof) comprising an entire data set can show a large frequency for a consistent increment of movement, and frequencies of unity, or thereabouts, for random increments.

For each distance associated with a datum pair, the model also records a direction. The distance from the *x*-coordinate for a dot in the previous display to the *x*-coordinate for a dot in the current display or vice versa, is calculated. (This is a directed, horizontal difference.) The same goes for the *y*-coordinate. (This is a directed, vertical difference.) The arctangent of the ratio obtained by dividing the *x* distance by the *y* distance then yields a direction. If the *y* distance is zero, of course, the maths fails, and the direction lies along the horizontal. Again, directions, or derivatives of directions; such as *differences* between successive directions, show consistency only for the consistently moving dot. All other directions show random variation.

[2] Detection based on structure of motion should not be confused with *structure from motion.* The latter usually refers to the reconciliation of common image features realized by an observer across different views of a three-dimensional body due to relative motion between the observer and body.

Note that the same methods and similar results are associated with modelling of spatial and temporal structure. The model has not been changed to accommodate the latter, but employs the same routines. Note also that temporal structure begins to emerge after just several display presentations, in the same way that a human observer needs just a few samples of change over time before realizing structure. In this vein, the model locates a regularly moving dot when a statistical test applied to successive frequency distributions for displays against that of a frequency distribution for total randomness, reaches some level of significance.

For the sake of the greyscale chart, Figures 9.9 to 9.11 below depict a severely limited data set. The data set is associated with a dot that made 64 regular movements in distance and direction against a background of 3 dots, each of which made 64 random movements in distance and direction. The greyscale chart deals with distances only (although it could deal with directions). The two line graphs show peaks in frequencies for consistent distances and directions respectively.

The greyscale chart shows two uniform grey bands (rows 16 and 32), collectively having 64 cells, the shading of which corresponds to consistent distances between dots in successive displays. The 64 cells belonging to each of the other three pairs of bands (rows 1 and 17, 6 and 22, 11 and 27), show shading that corresponds to inconsistent distances between dots in successive displays. And, again, these appear contiguously, in bands, because while each dot's coordinates change across displays, their array indices do not; and Hausdorff sets are arranged in order determined by array indices.

Now the number of distance combinations among 4 dots is 16, and given the 64 movements across frames then 16 × 64 = 1024 cells are required, each with a shading corresponding to some distance. These are accommodated by the regular greyscale chart, which has 32 × 32 = 1024 cells, but all the lighter shades, which are associated with longer distances, have been bleached at shade level 11 to white. This was a good level of bleaching to bring out the structure among the shorter distances—which were hidden in the darker shades—and to minimize the visual impact of the shades associated with the longer distances.[3]

[3] Undoubtedly, structure at any distance can be highlighted by such methodology. Moreover the negative of a greyscale chart can be subjected to such shade scaling in order to see the same effects for long distances, if, indeed, long distances are relevant to an inquiry.

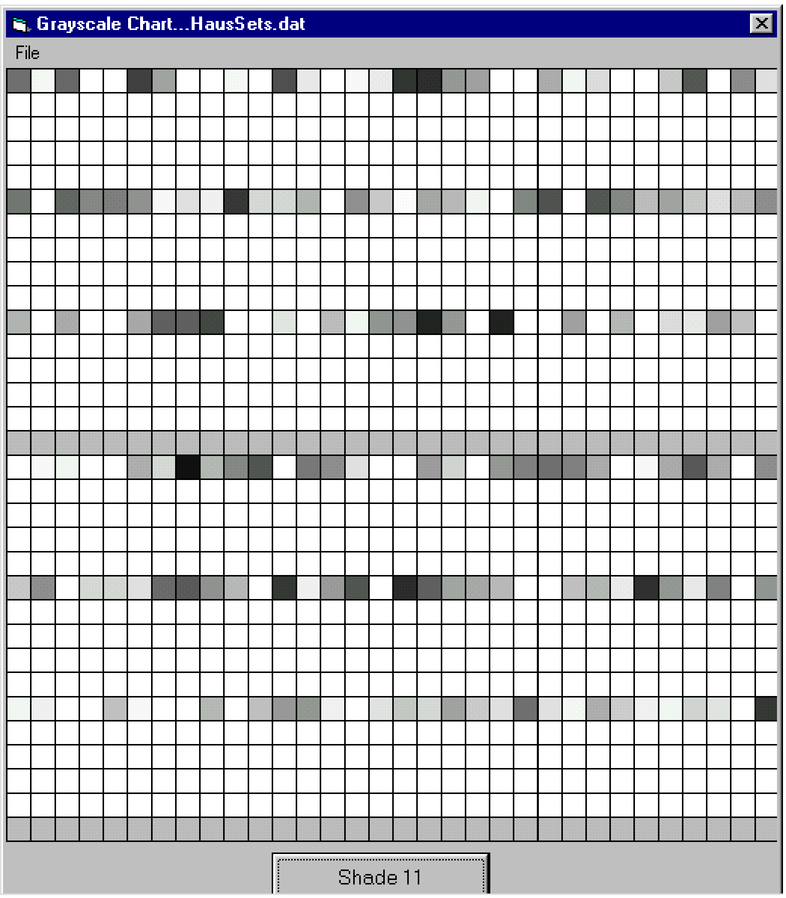


Figure 9.9: The 64 cells belonging collectively to the two uniform grey bands show shading that corresponds to consistent distances between dots in successive displays. The 64 cells belonging to each of the other three pairs of bands show shading that corresponds to inconsistent distances between dots in successive displays. (Ignore grey areas beyond the grid: they are part of the Visual Basic form object.)

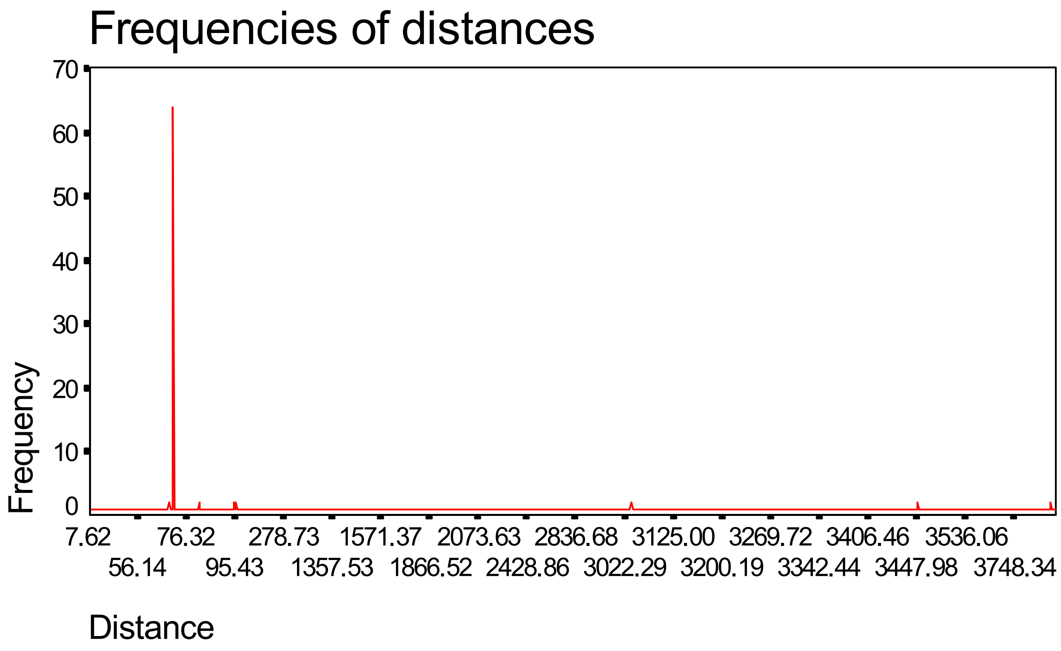


Figure 9.10: Peak in frequency of consistent distances between successive displays for a moving dot. The tiny peaks show frequencies of chance consistencies in distances between successive displays for randomly moving dots.

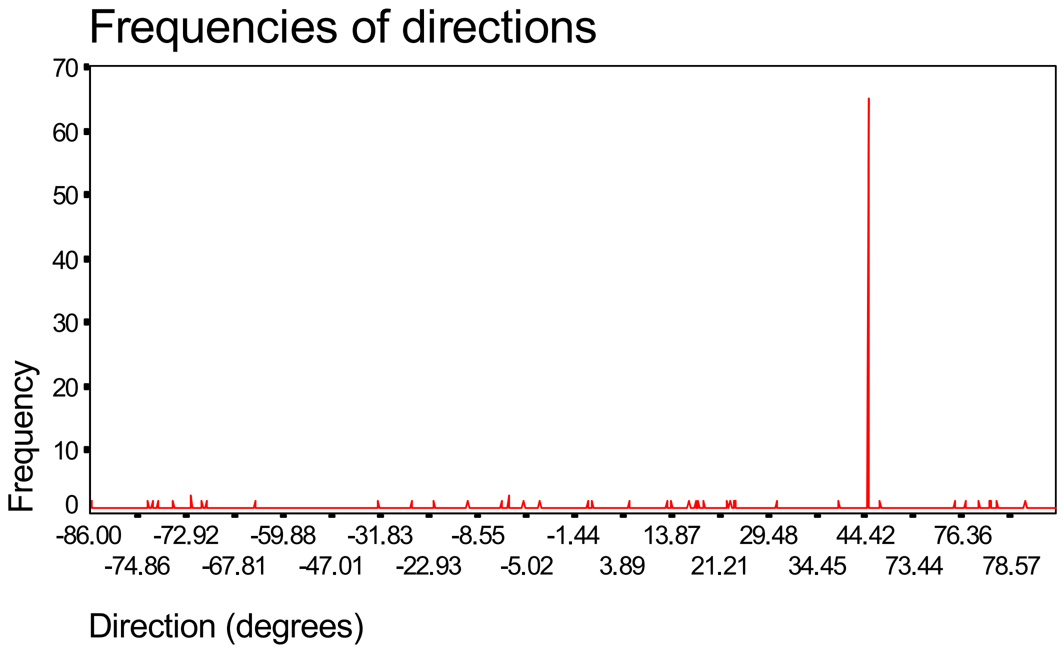


Figure 9.11: Peak in frequency of consistent directions between successive displays for a moving dot. The tiny peaks show frequencies of chance consistencies in directions between successive displays for randomly moving dots.

The model has been tested using 16, 32, and 100 dots, with the result that it located the regularly moving dot every time. A regularly moving dot is readily discerned by an observer, and detected by the model, against a background of 15 and 31 randomly moving dots. Yet it is not so readily discerned by an observer, in spite of being detected by the model, against a background of 99 randomly moving dots. Form and dot size for this kind of observation are not critical, and, disregarding the outline figure of the airplane, Figure 9.12 provides an indication of the kind of form and dots used.

It is obvious that consistent distances in both spatial and temporal patterns embedded in noise can only be discerned out to some magnitude related to amount of noise interference, upon which detection should rapidly decline. Hence for a more realistic model, frequencies of consistencies should ultimately become masked by frequencies of chance, close values in the Hausdorff sets produced by the random movement of many dots. This can be accomplished by relaxing the (unreasonable) constraint that consistencies be (exactly) equal. A histogram, with a bin-width that restricts distances and directions within a bin to be within some proximate value of one another, can be tuned by bin width to do the job. For some density of randomly moving dots, there are not enough chance, close values to mask the long term regularity of a consistently moving dot, but given enough randomly moving dots, the consistently moving dot gets lost in like, but transient, regularities.

Since counting consistencies is essentially a procedure akin to Ripley's *K,* then, again, for a more realistic model, in terms of both pattern perception and processing overhead, each dot should be checked against every other dot out to some cut off distance. For Ripley's *K,* this is about half the length of the shorter side of a rectangular study area. (See end of Chapter 4, page 80.) In any event, because of the relevance of Delaunay neighbours to pattern perception each dot should be checked against every other dot out to at least its largest Delaunay neighbour distance. In either case, the counting process is facilitated by a list of distances sorted in ascending order.

Technical description of detection of coherent motion

The following technical description complements the general description given above. Initially the first two arrays of dots, each corresponding to a frame, are loaded and displayed in sequence. Distances, found using Hausdorff sets, from dot positions in the previous display (first display) to dot positions in the current display (second display) are saved in an array. Upon update of the display to a new (current) display, again distances from dot positions in the previous display (old 'current' display) to dot positions in the new (current) display are saved in another array. Each time the display is updated, what was the current array becomes the previous array and the other array is overwritten to become the current array. In this fashion, only two consecutive display states are considered at a time. The same scheme is implemented for directions, but for explanatory purpose just distances are dealt with in the following.

If distances for corresponding elements of the two arrays are the same, or fall within some tolerance for similarity, then corresponding 'counts' are incremented in a third array. This array has a cell associated with each pair of distances. Distances considered to be different have values for their counts reduced by fractional amounts that diminish with each display, such that a result cannot be negative. For anything that moves consistently it is easy to see that its count escalates with each display, and for anything that moves erratically its count stays relatively low; all subject to tolerance for similarity. Detection is based upon some threshold for counts, and can be initially established using standard statistical methodology. Here, the application of Pearson's chi-square statistic is appropriate.

Frequency categories of successive distributions are tested for significance against corresponding frequency categories of a distribution for complete random motion among the same number of dots. At the display in the sequence where the statistic becomes significant, the highest frequency count in its distribution is chosen as the initial threshold. Upon establishing the threshold, such testing is not applied to subsequent displays. From here on the threshold becomes adaptive, and in such a way as to admit representational momentum.

Ongoing detection continues to be based upon the state of just two arrays, and after initial detection the threshold is steadily increased in a manner such that the escalating count for the consistently moving dot easily surmounts it. Concerning representational momentum, if the dot moves in too much of an unexpected mannner its count drops below the elevated threshold and the dot is lost, upon which the threshold decreases towards the asymptote of its initial value until another detection occurs. And for any run of transient consistencies inevitably encountered in dense displays of randomly moving dots, an increasing threshold due to any regularly mitigates the chance of false detections. Bin width tuning mitigates initial detection of a regularly moving dot in dense displays, but once detected then ongoing detection is given some preference. In other words, once a regularly moving dot is spotted then it is not easily lost.

Detection of extended structure

At this stage it can be seen that there is no division of methodology; just the one methodology applied to variants of application. By moving an entire structure—either rigidly or differentially under transformation—against a randomly moving background, figural form can be detected by the method outlined above. Figure 9.12 shows an airplane outlined by the detection system. The airplane was detected as it arced across the form; i.e., as it translated and rotated.

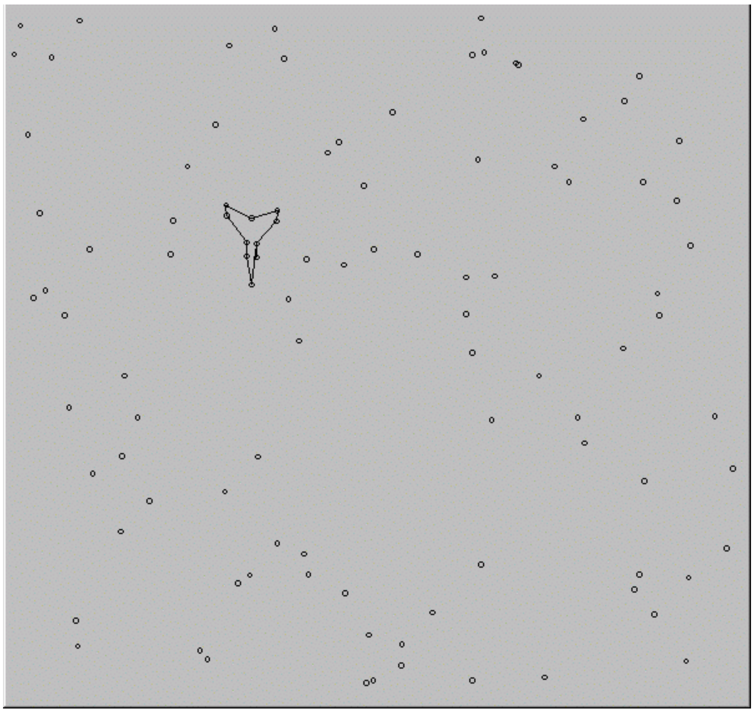


Figure 9.12: Against a background of randomly jiggling dots, the consistently moving dots belonging to the airplane under any affine transformation are easy to distinguish.

For a structure undergoing the same kind of motion as the above-described consistently moving, single dot, all its elements show consistency of distance and direction against the random elements. In this way, all structural elements can be detected by examination of the elements of the Hausdorff sets, and through examination of corresponding directions. Not only consistent displacements of the *same* elements from frame to frame, but consistent distances between *different* elements from frame to frame, indicate structure in a demonstration of the Gestalt principle of common fate. And ‘consistent’ need not only refer to 'constant', within some dynamically adjusted tolerance, as for translation, but to transformational consistency across other affine transformations as well. Given that both static and dynamic structure can be differentiated by processing components of Hausdorff distance—which have a considerable measure of invariance under depth rotations and affine transformations generally—Hausdorff distance methodology everywhere furnishes redundancy.

Wagon-wheel illusion

The illusion of a wagon-wheel reversing direction while the wagon continues forward, as commonly seen in Western movies, occurs because of synchronization effects involving frame rates relative to rotational rates, and the story concerning shortest transformational distance. The purpose of including an example of such reversing phenomena along with analyses of spatio-temporal structure is that can be explained simply in terms of the Gestalt proximity principle.

For the purpose of this explanation a 'tagging object' is a duplication of a 'target object', established initially upon pattern detection, and the objects have some planar rotational symmetry. From its initial orientation the tagging object is brought into operative alignment with the rotated target object by a principled minimization of directed Hausdorff distance. Each time the target object is rotated, the tagging object is brought into operative alignment—by successive minimization from its previous orientation—via the transformation that moves it though the shortest distance.

For example let the object have ten-fold symmetry, i.e. at each angular change of 36^o its orientation looks to be unchanged. Accordingly, when the target object is rotated through more than 18^o and less than 36^o from the tagging object, then the shortest distance for alignment of the tagging object with the target object lies in the opposite rotational direction to that when the target object was rotated less than 18^o.

For a computer implementation of the wagon-wheel illusion, first see Figure 9.13. The circle and annotations, which do not appear on the actual computer display, are for explanatory purpose only. Initially an object is loaded into the program. By way of simplifying the explanation, let the object, which is outlined by dots, be a square on this occasion. The cross facilitates seeing the wagon-wheel reversal.

The square is moved around the circumference of the circle in 1^o increments. After each movement, it is rotated consistently in the one direction on its centroid by the number of degrees it has moved around the circle. Hence, after it has moved around the circle by 1^o it is rotated on its centroid by 1^o. After it has moved around the circle by one more degree it is rotated on its centroid by a further 2^o. After it has moved around the circle to the 45^o position, it is rotated by 45^o on its centroid from its current orientation, and when it has finally moved around the circle to the 360^o position, it is rotated by 360^o on its centroid from its current orientation, which, of course, leaves it at the starting position and orientation.[4]

Before the tagging object is brought into operative alignment with the rotated target object, the translational discrepancy between the two is described by 1^o of offset around the circle and the rotational discrepancy depends upon where the target object is on the circle.

While the tagging object tracks the target object at less than 45^o of rotational discrepancy, it appears to rotate in one direction, and after 45^o and before 90^o of rotational discrepancy it appears to rotate in the opposite direction. And so it appears to reverse direction at the boundary of each 45^o segment around the circle, which can be explained by the shortest rotational distance switching at each boundary.

[4] Do not confuse the rotation of the square on its centroid with the notion that it remains tangent to the circle. Rather, each time the square moves around the circle it rotates more than it did previously.

If the object is left to rotate beyond 360^{o}, rather than end program execution at 360^{o}, then eight reversals are seen around the circle; instead of seven as is the case for program termination at 360^{o}. Since the object has four-fold symmetry, one might expect to see 20 reversals around the circle for an object with 10-fold symmetry. This is exactly what is seen; there is a reversal every 18^{o}. And some measure of relative symmetry could be also established from this procedure.

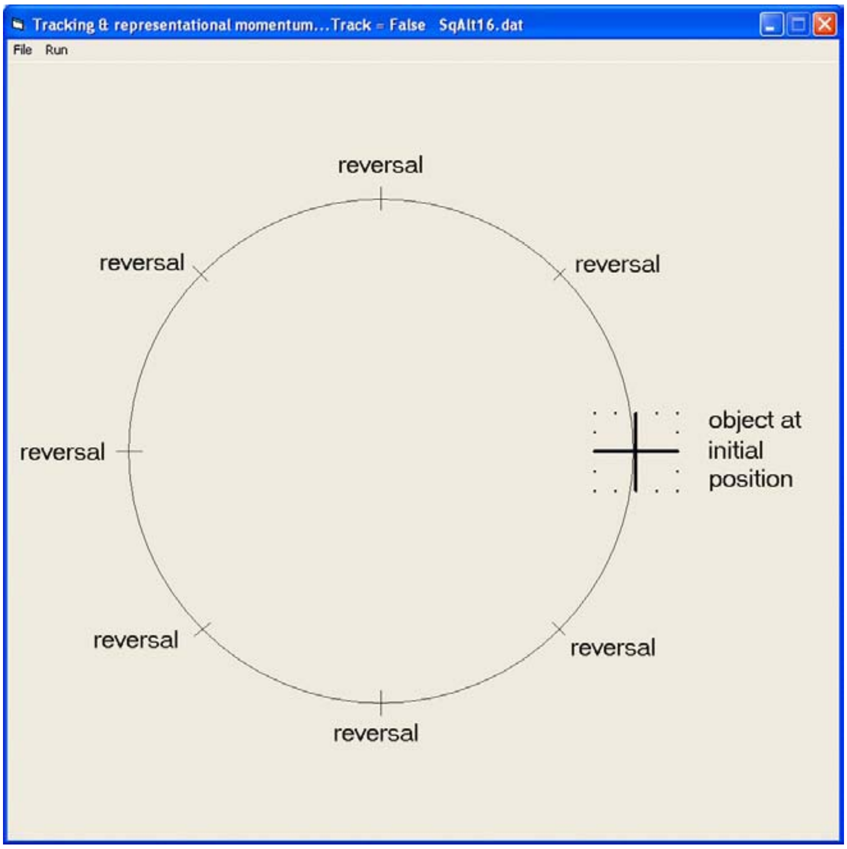


Figure 9.13: Object, in the form of a square outlined by dots, loaded in readiness for demonstrating the wagon-wheel illusion.

Application to bitmap images

The methods discussed can be applied to more general visual stimuli, such as bitmap images. Bitmap images are used by various media to accurately represent images. A bitmap is essentially an array of values for a grid of pixels, or pixel regions. An $m \times n$ image is described by a single dimensional array of length equal to the number of storage elements per colour value × $m \times n$. It now becomes a simple matter of moving a (possibly resizable) window, which spans some few elements, along the array in order to discover the regions of change in the bitmap image. If all elements spanned by the window are sufficiently near to any one value, then the region is relatively consistent. On the other hand if any elements spanned by the window are sufficiently different, then the region indicates change. The coordinates of the region are then included in an array, which will contain coordinates of other regions of change.

The situation now corresponds to the dots in the relatively sparse dot representation of images and background employed by this thesis. The same methods described for dot representation are applied to something that corresponds to salient parts of a bitmap image; which is just a more densely packed dot representation anyway.[5]

Transformations involving 3D objects and their 2D projections

Retinal images are two-dimensional projections, usually from three-dimensional objects; hence the model needs to deal with transformations involving 3D objects and their 2D projections. Views of 3D objects typically change due to relative displacement between observers and objects. For example, to see the face of a watch we can move to an appropriate position relative to the watch, or we can move the watch to an appropriate position relative to us, or, as is usually the case, some degree of both can be implemented. With regard to transformations, only relative displacement matters; not which entity is displaced.

Because of extra complexity in dealing with 3D situations, it is expedient to use matrix algebra representations of 'coordinate sets' and 'transformation sets'. Coordinate sets represent an object at some orientation and transformation sets operate on coordinate sets to change the view of the object. If employing rectangular coordinates, then a 3D object is embedded in a coordinate system comprising *xyz*-sets of coordinates. For the present application, the coordinate system is chosen to be object centred and the *xy*-plane coincides with the plane of a viewing screen. Note that there is no absolute coordinate specification for a 3D object, only coordinate specifications for different views.

A view of a 3D object can be represented by a matrix of three-element, columnar coordinate sets, each set (triplet) defining a point in 3D space. The number of columns in the matrix is the same as the number of points in 3D space that require representation. The first row of the matrix contains *x* coordinate values, the second row, *y* coordinate values, and the last row, *z* coordinate values. Since a view of a 3D object is confined to the plane of a screen, only the projection is seen; which projection is specified by the *xy*-components of the matrix.

$$\begin{pmatrix} 1 & .5 & -.5 & -1 & -.5 & .5 & .84 & .315 & -.21 & -.36 & -.21 & .315 \\ -.8 & -.8 & -.8 & -.8 & -.8 & -.8 & -.4 & .125 & .65 & .8 & .65 & .125 \\ 0 & -.867 & -.867 & 0 & .867 & .867 & 0 & -.546 & -.364 & 0 & .36 & .546 \end{pmatrix}$$

Figure 9.14: Coordinate matrix for truncated pyramid.

The matrix shown in Figure 9.14 was taken from Anton and Rorres (1987), and represents twelve points in normalized 3D space. If pairs of points are to be connected by lines, then a specification of which pairs are to be connected can be held in an array. The matrix corresponds to a canonical view of a truncated pyramid, the projection of which is shown in Figure 9.15(a). The truncated pyramid shown in Figure 9.15(b) is a projection stemming from the matrix.

[5] Bitmap image transformations and other manipulations are accomplished via the use of a windows API (Application Programming Interface) called BitBlt, which is an acronym for Bit Block Transfer. (It is an API function exposed by the GDI32.dll that performs a transfer of a rectangular area of an image to another rectangular area of equal size. This includes copying, combining, or merging images, and displaying the result.)

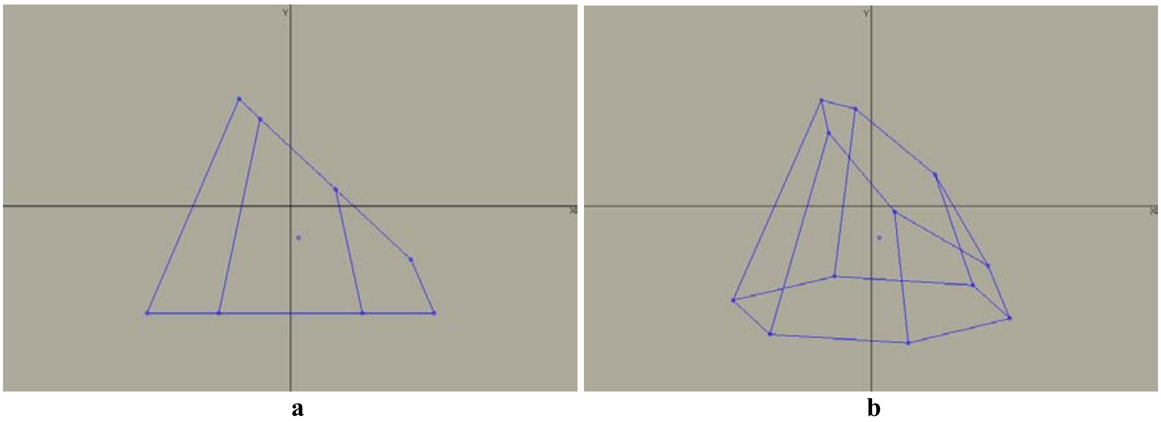


Figure 9.15: Truncated pyramid. Canonical view (a) and rotated view (b).

It has been multiplied by a transformation matrix, resulting in a view rotated from the canonical view by 15$^{\mathrm{o}}$ around the *x*-axis. It has also been multiplied by another transformation matrix, resulting in a view further rotated from the canonical view by -15$^{\mathrm{o}}$ around the *y*-axis. The two matrices of pure rotations could have been first multiplied together to get a matrix of a compound rotation, which would give the same result in one operation on the coordinate matrix.

Both *x*- and *y*-rotations are depth rotations. The *z*-axis can be visualized as a line that runs perpendicularly into, and out the other side of the screen, through the origin of the *xy*-axes. Hence, one way that *z*-coordinates figure in transformations involves planar rotations. Effects of transformations involving *z*-coordinates are not always visible in the *xy*-plane, however, and sometimes they only become visible when followed by certain other transformations. For example, dilation of an object in the *z*-plane can only be seen if it is subsequently rotated in depth.

The nature of transformation matrices

Affine transformations can be represented by $n \times n$ matrices that are elementary variations on the matrix

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

This matrix is an identity matrix, and does not change a $3 \times n$ matrix that is multiplied by it. The following is a scaling matrix:

$$\begin{pmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & \gamma \end{pmatrix}$$

The first two of the following 2×2 matrices represent shears along coordinate axes, and the third represents a reflection about $y = x$. The last two represent dilations or contractions along coordinate axes:

$$\begin{pmatrix} 1 & 0 \\ k & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & k \\ 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \begin{pmatrix} k & 0 \\ 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 \\ 0 & k \end{pmatrix}$$

Corresponding other $n \times n$ matrices are simple extensions of these.

The following is a rotation matrix, about the y-axis:

$$\begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix}$$

Note the symmetries implicit in these transformation matrices; which matrices are also taken from Anton and Rorres (1987).

Constraints on transformation matrices

An elementary product of an $n \times n$ matrix is any product of n entries, no two of which come from the same row or same column. The *determinant* of an $n \times n$ matrix is the sum of all signed elementary products. Note that the determinant of an identity matrix is 1.

Except for scaling matrices, it turns out that the transformation matrices have determinants of ±1. Transformation matrices multiplied by transformation matrices have determinants of ±1. Rotation matrices have determinants of 1, and a matrix with a rotation combined with a reflection in a coordinate plane has a determinant of –1, for example. Since a determinant can be any real number, this constitutes a considerable constraint to be associated with transformations.

Scaling matrices have determinants with symmetry about the number one. If a scaling matrix has determinant *d,* and scales a coordinate matrix by amount *s,* then the scaling matrix that scales the coordinate matrix by amount $1/s$ has determinant $1/d$. Where any one coordinate is scaled, the determinant equals the scaling factor. Where more than one coordinate is scaled, the determinant equals the product of the scaling factors.

A nonzero determinant indicates that an $n \times n$ matrix is invertible. The inverse of an $n \times n$ matrix is the matrix by which it needs to be multiplied to get an identity matrix. If applied after a transformation by a matrix, the inverse matrix inverts, or undoes, the transformation. A constraint on a transformation matrix then, is that it be invertible. Other constraints ensue from the following considerations.

An orthogonal matrix is one in which the coordinates are linearly independent. (In a 2D or 3D system, lines from the origin to the coordinates are at right angles.) The determinant of an orthogonal matrix is always ±1. (Note: this does *not* mean that if the determinant of a matrix is ±1, then its is orthogonal.)

Transposing an $n \times n$ matrix involves changing rows 1 to n into columns 1 to n. An $n \times n$ matrix is orthogonal if its inverse equals its transpose. Except for scaling, shearing, and translation matrices, the transposes of the transformation matrices equal their inverses. In other words, except for scaling, shearing, and translation matrices, the transformation matrices are orthogonal. (Note that shearing and translation matrices still have determinants of ±1.)

If every vector in a vector space is some linear combination of a subset of vectors belonging to that vector space, then the subset of vectors *spans* the vector space. Further, if the vectors of that subset are orthogonal then they form a *basis* for the vector space. For transformations in which transposes of the transformation matrices always equal their inverses, it is the basis that is rotated. The vector space changes *orientation* about an invariant pivot, which is the origin of the object's coordinate system.

What about transformations for which transposes of the transformation matrices do not equal their inverses? In the event of scaling, what were the basis vectors become changed in length, and in the event of shearing, what were the basis vectors become changed in angle. For scaling, the space of an object is scaled, either equally or differently in two or more orthogonal directions, and for shearing the space of an object is uniformly sheared such that what were right angles become inclined. For these two transformations, the vector space changes *dimension* about an invariant pivot. For translations, an object is shifted while its orientation and dimension remain invariant. Because the object has shifted to a new coordinate reference, the origin of the object's coordinate system is not invariant.

These considerations—and more, which have not yet been adequately developed—constitute constraints on matrices of affine transformations, whether pure or compound. As stated in the Introduction, pages viii and ix, constraints are necessary for solving inverse problems, and since the process of vision is an inverse problem then determining constraints is relevant.

Hausdorff distance applied to 2D projections of 3D objects

Hausdorff distance is minimized between coordinate points of a 2D projection of a 3D object and coordinate points of a 2D projection corresponding to combinations of affine transformations of the 3D object. This brings the transformed projection into alignment with the original projection (or vice versa). Established Hausdorff tracking methodology is, essentially, all that is necessary to accomplished the task. Figure 9.16 provides an example using the truncated pyramid, following from which is the explanation of methodology as it applies to Hausdorff tracking in three dimensions.

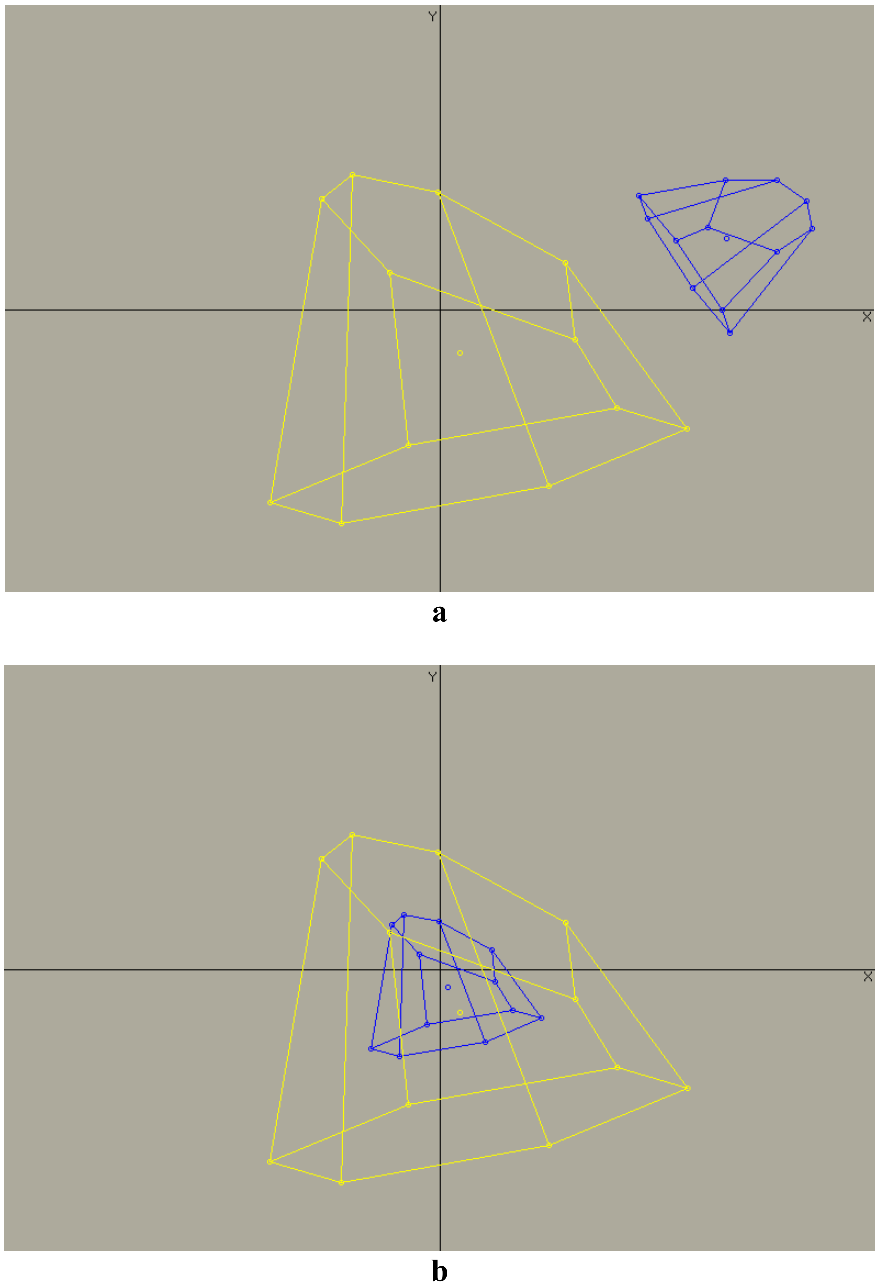


Figure 9.16: The blue object is depicted in (a) after having undergone some combination of just some affine transformations on the yellow object. The Hausdorff minimization procedure described in the text finds the transformations that effectively eclipse the yellow object with the transformed blue object. If the scaling option is disabled, however, the blue object assumes the same orientation as the yellow object, but sits inside as shown in (b). Its position corresponds to the minimum Hausdorff distance between the original projection and the scaled down projection.

The method involves repeated Hausdorff distance minimizations. Transformations that use only *x*-coordinate manipulations to minimize Hausdorff distance between the projections are first applied to the 3D object, then transformations that use only *y*-coordinate manipulations to further minimize Hausdorff distance are applied, and then transformations that use only *z*-coordinate manipulations to still further minimize Hausdorff distance are applied. This *xyz*-minimization process is iterated until no more reduction in Hausdorff distance is found. If the minimum is nominally close to zero, then a set of transformations has been found that brings the transformed view into operative alignment with the original view.

If the minimum is not nominally close to zero, then it is a local minimum rather than the sought-after global minimum. Accordingly the 3D object is 'nudged' from the local minimum in readiness for another minimization run on its 2D projection. ('Nudging' is accomplished by a 'noise' generator, linked to a random number generator drawn from a uniform distribution, which produces *xyz*-'nudging' transformations of random magnitudes.) The process is repeated until the global minimum is found, and whether the global minimum is unique or not depends upon the symmetry of the 3D object. The more symmetric the object, the quicker a global minimum is found; in a statistical sense.

Although this process works satisfactorily it does not necessarily yield a smooth reducing sequence of Hausdorff distances; which corresponds to a smooth sequence of transformations. However, if each *xyz*-nudging transformation has the random magnitudes of its components constrained to correspond to a Hausdorff distance less than that associated with the previous nudging transformation—apart from the initial nudging transformation in a succession, which has no previous of course—then a smooth reducing sequence is produced. Nonetheless such refinement results in the algorithm occasionally getting stuck in a local minimum. This can be overcome by starting another reducing sequence from another initial nudging transformation.

Notwithstanding the utility of a minimum Hausdorff distance that is nominally close to zero, larger minima may be sought where it is desired to match a shape that cannot be aligned by superposing. For instance, Figure 9.16(a) shows the projection of a uniformly scaled down object that has been translated and rotated. If the scale matching facility is disabled, the projection is brought into a matching orientation inside the original projection as shown in Figure 9.16(b). Its position corresponds to the minimum Hausdorff distance between the original projection and the scaled down projection.

The Hausdorff minimization scheme just outlined automatically includes (the intrinsically abstract) area and line objects (2D and 1D objects respectively). These objects are transformed similarly in 3D space and their projections are matched similarly by Hausdorff minimization. Resolution occurs more quickly with lesser dimensional objects. It appears to also occur more quickly as objects are rotated, scaled, or sheared closer to their centres of gravity.

Importantly, a single, compact Hausdorff minimization algorithm resolves all affine transformations for all objects. Composite rotations require all the features of the algorithm, which include 'nudging' from local minima. Scaling does not require 'nudging' because there is only one scaling minimum. The same applies to shears. However, they both require appropriate repeated transformations until the Hausdorff minimum is found. Because reflections only require one of two possible transformations for each coordinate, they do not even require that much.

Generative transformational explanation of object recognition

The above description requires that 3D representations of objects are stored in memory. Further, it is desirable (but not necessary) that these be representations corresponding to cannonical views so as accord with simplicity and the minimum principle. The task of a generative transformational system then becomes one of manipulating a stored object such that its 2D projection best matches that of the 2D input image. And on the assumption that stored representations form the basis for categorization, then such transformation provides a mapping from the general to a specific: from a category to some instance.

But how might 3D representations of objects be realized in the first place? They may be built by sensory experience of the environment. They may also be facilitated by further processing of 2D images.

Reconstruction of 3D from 2D

At this stage the reconstruction of 3D images from 2D projections is anticipated. Binocular and monocular vision is considered. 3D reconstruction is based upon structure perceived from different (temporally displaced) views due to relative motion. It is also based upon structure due to different (laterally displaced) views related to the placement of each eye. Given two (or more) views of a set of 3D features, the corresponding 2D features are first brought to match each other; as described above. The relative locations of the eyes are then recovered from these matches, and, finally, locations of 3D features are recovered by triangulation (Gershon & Benady, n.d.).

Otherwise stated, the inversion process—going from 2D feature locations across multiple views to the corresponding 3D feature locations—requires two elements. The first is to accurately locate features from each image and accurately match those features across multiple images. The second is to extract the relative locations of the eyes from the matching 2D points, whereupon locations of 3D features are recovered by triangulation (Gershon & Benady, n.d.).

Planar distances of the features are evident in Figure 9.17. Depth-wise distances of the features are proportional to angles α and β, which can be determined by distances travelled in the process of rotating the view from one eye into alignment with the view from the other eye. A table of *xyz*-coordinates can then be constructed.

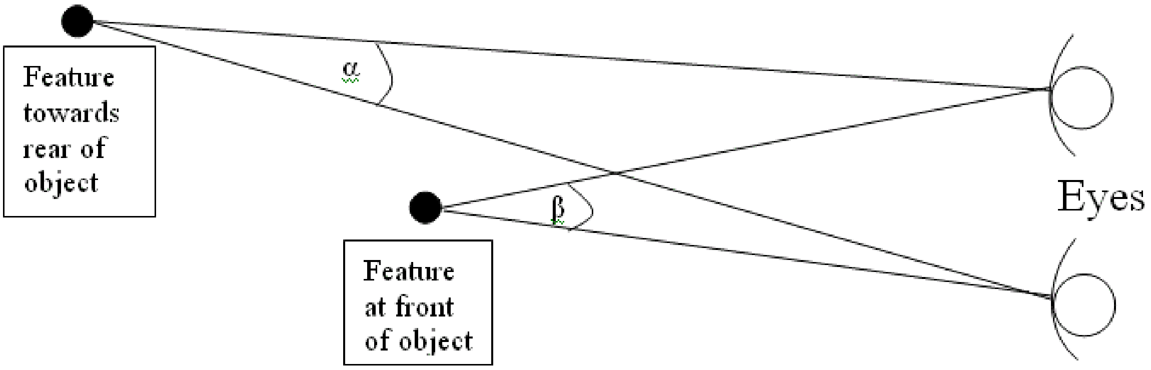


Figure 9.17: Besides planar (*xy*-coordinate) placement, depth (*z*-coordinate) placement can be determined.

Single eye 3D would exploit 3D geometry and principles of perspective distortion to calculate the 3D pose of an object using a single, stationary image (Braintech Inc., n.d.). The 3D pose of an object consists of its *xyz*-position, and roll, pitch, and yaw angles; all with respect to their referents. (Roll, pitch, and yaw angles are angles of rotation around respective axes.)

Upon reconstruction of 3D from 2D, the task of a generative transformational system might arguably be one of manipulating the 3D canonical representation of an object, as opposed to its 2D projection previously cited, such that it best matches that of the reconstructed input image. However, this route to realization requires a more stringent observation process each time something is apprehended. And on the grounds given above for seeing an imaged object as belonging to the object-family that, under a suitable transformation, produces the most symmetrical projection (involving Vickers n.d., unpublished study), we hypothesize that the transformational system manipulates the 3D canonical representation of an object such that its 2D projection best matches that of the 2D input image.

In any event the underlying methodology is here provided by the generative transformational approach to visual perception, using Hausdorff distance. The Hausdorff distance measure is ideal for reconciling transformed views and is conducive to reconstruction of 3D from 2D, which, considered along with the work involving only planar coordinates, points to a more or less full theory dealing with the organization of elements into structures.

Chapter 10: Pattern Differentiation by Half-Distance Points

Brief summary of chapter

Half-distance points can be determined at negligible extra cost during the calculation of Hausdorff distance where distance from each dot to every other dot is calculated. In this short chapter, structure in half-distance points is examined for ability to differentiate patterns from noise, and, further, to tell apart different patterns. The work outlined here remains tentative, nonetheless some of the measures that were tried merit a mention. All the work is original.

Half-distance structure

As stated in Chapter 8, pages 165 and 166, half-distance points are located halfway between every possible pair of dots. They institute every possible reflection symmetry, and raise the possibility of a pattern differentiation device couched in terms of half-distance granularity. The more regular a dot pattern is, the more the half-distance points tend to little structured 'islands' of proximity, in which they can overlie one another. The more irregular a dot pattern is, the more the half-distance points disperse towards amorphousness, and the likelihood of one overlying another is diminished.

A suitable small, regular region (subset) of half-distance points depicted in some unique colour is subjected to a pixel count for that colour. One way of getting a suitable sampling region is by calculating the 'inverse of the number of dots' × 'overall display area', with sides the same ratio as the overall display area. Half-distance points are most dense at the centre of mass of a display and least dense at the peripheral of mass. The sampling region is best located where there are good proportions of uniquely coloured pixels and background pixels. A simple scheme finds a region halfway between the centre of mass and peripheral of mass, and a more complex scheme uses dot distribution to find a suitable region.

A pixel count in the region of choice for the colour associated with half-distance points reveals the density of coloured pixels. The colour can be also scanned for regularity of visual aspect over the region (structure). One way of doing this is through a line-by-line, raster-type, scan that produces a binary string: one value for half-distance colour and another value for any other colour. The string is then subjected to an autocorrelation procedure, in which a range of within-line pixel lags is used as well as between-line pixel lags. These measures can be used to differentiate a Glass pattern, for example, from a noise display.

For examples of half-distance displays, see Figures 10.1 and 10.2. An important consideration with such displays is that of 'spatial non-locality'. One or another small part of the display says something about the overall display.

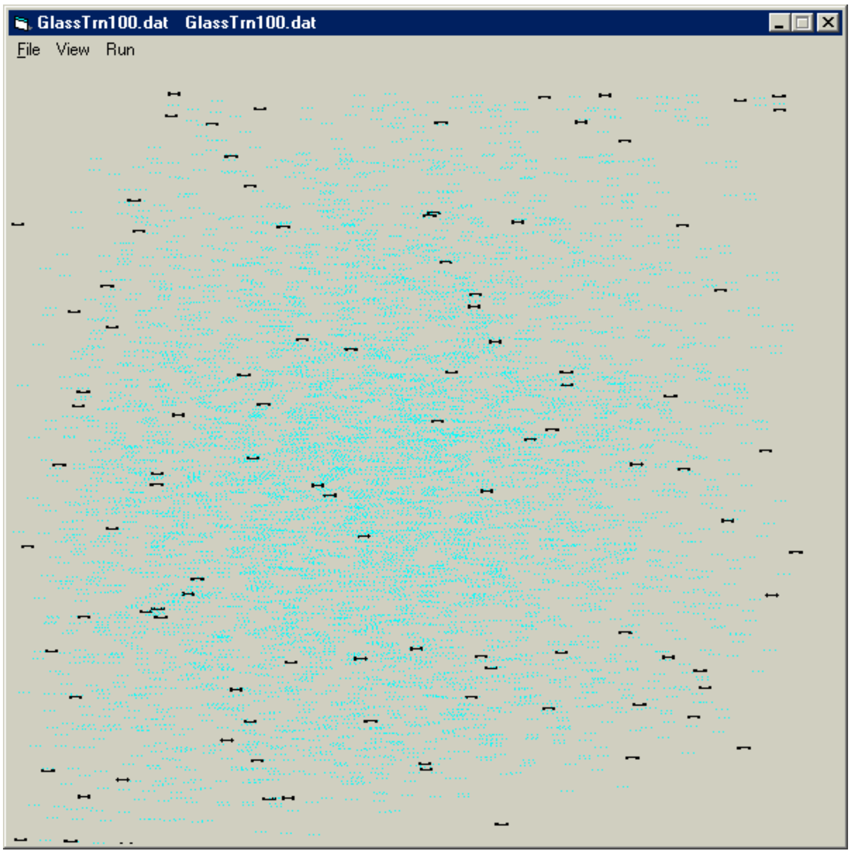


Figure 10.1: Glass pattern with 200 dots, dusted for half-distances.

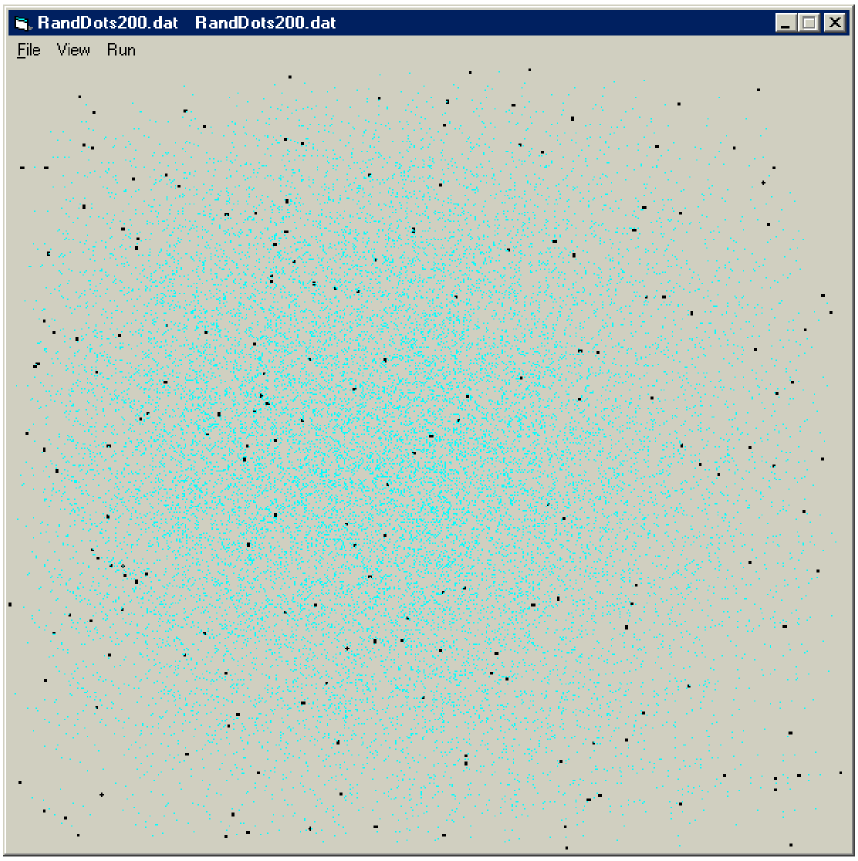


Figure 10.2: Noise display with 200 dots, dusted for half-distances.

Half-distances applied to Marroquin patterns

As stated earlier, (Chapter 1, page 3; Chapter 4, pages 70 and 71) Marroquin patterns are generated by transformations on structured arrays, and the transformations are typically moved through any magnitude from relatively small to relatively large. See Figures 10.3 and 10.4 for 'good' and 'poor' Marroquin patterns respectively. Note that the more regular a Marroquin pattern is, the more the half-distance points tend to little 'islands' of proximity, in which they can overlie one another. The more irregular a pattern is, the more the half-distance points disperse towards amorphousness. Half-distance measures can differentiate a good Marroquin pattern from a poor one. Note the disposition of the blue pixels, which are coincident with half-distance points, and note the 'non-local' effect, in which the behaviour of smaller areas corresponds to that of larger and larger areas.

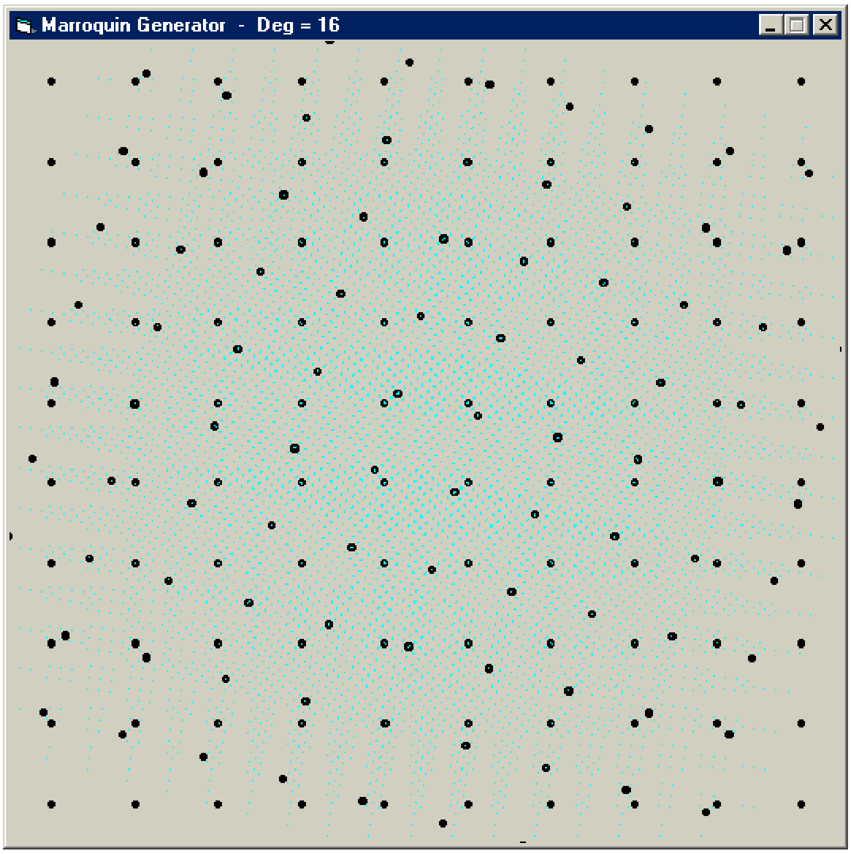


Figure 10.3: 'Good' Marroquin pattern.

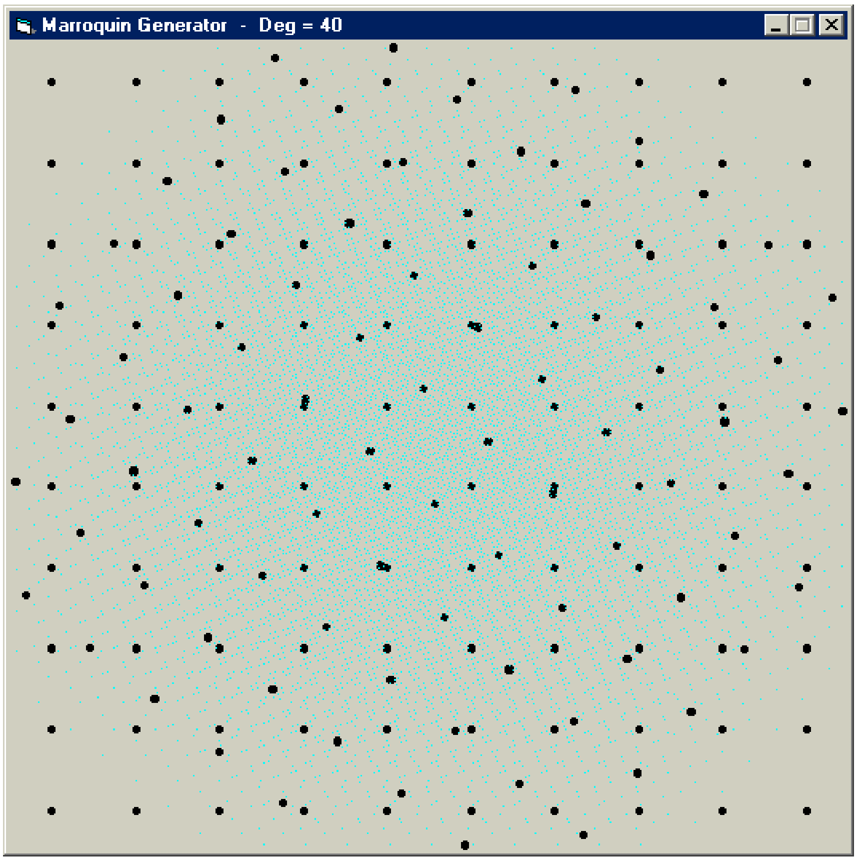


Figure 10.4: 'Poor' Marroquin pattern.

Note that the structure of half-distance points for the Glass pattern, Figure 10.1, is different to the structure of half-distance points for the 'good' Marroquin pattern, Figure 10.3. The usefulness of half-distance points might be extended to also distinguish between different types of pattern. Half-distance points were described in Chapter 8 as a means of pinpointing and delineating structure; particularly somewhat imprecise structure. They involve little extra processing overhead in any distance defining procedure.

Application to human visual perception

Half-distance sampling regions can indicate degrees of elongation or symmetry across a range of inter-point distance scales for a display. They can provide a compact description related to spatial arrangement, orientation, and size, which is what the medial and submedial axes of Chapter 6 do in a somewhat affiliated way. And as per Chapter 6, spatial arrangement, orientation, and size are shown in neurophysiological studies to be associated with the functional role of the primary visual cortex. Again, see Lee, Mumford, Romero, and Lamme (1998), for example.