Surface Orientation and Segmentation from Perspective Views of Parallel-Line Textures

Mark L. Moerdler
John R. Kender

CUCS-159-85

450 Computer Science
Columbia University
New York, N.Y. 10027

Table of Contents

1. INTRODUCTION 3
2. MATHEMATICAL BASIS 4
3. METHODOLOGY 6
4. RESULTS AND ANALYSIS 7
    4.1. Pure Synthetic Image 8
    4.2. Random Noise 8
    4.3. Textural Variations 9
    4.4. Multiple Adjacent Surfaces 10
    4.5. Multiple Overlapping Surfaces 11
    4.6. Multiple "Transparent" surfaces 11
    4.7. Perspective-Based Segmentation 12
5. CONCLUSION AND FUTURE WORK 13

Surface Orientation and Segmentation from Perspective Views of Parallel-Line Textures

ABSTRACT

This paper describes a particular shape-from-texture algorithm that constrains and defines surface orientations with little a priori knowledge. It has been found to be robust under a variety of conditions. The method uses the change in the spacing of parallel surface markings to derive the orientation and shape of multiple surfaces in synthetic noisy scenes. We first describe the problem domain and the representational approach. Next we outline the mathetical basis of the method, and its straightforward graphical interpretation. Third, we discuss the implementation methodology employed and some concrete implementational issues. We explain the algorithm's response to a number of images, in which various forms of noise or surface perturbation are handled: texel loss or distortion, and multiple or overlapping surfaces. We discuss the relationship between the various forms of noise and the quality of the results. Fourth, we demonstrate perliminary findings of texture-driven segmentation in which this method not only segments adjacent surfaces but also separates two overlapping "transparent" surfaces. We conclude with our future research plans.

1. INTRODUCTION

One of the key questions of image understanding is how to recover the three dimensional structure of a surface. Although many algorithms for discovering surface constraints have been proposed and implemented, most of them are based on a single surface cue. Considering the diversity of potential image surfaces and the fact that these methods are only partially successful even in a limited domain, a more broad based approach would seem necessary. Robustness and generality can only be achieved by means of a multi-module design containing many components, each of which uses a different surface cue.

Such a multi-module approach has been implemented in a wide range of vision domains (see [1], [4]). These systems contain either several distinct modules, or one module whose various instantiations act independently on subimages as if they were several distinct modules. In this paper, we shall discuss the results of one such module which simultaneously induces a segmentation and performs an initial surface analysis. It attempts to recover surface constraints from the apparent change in spacing between textual primitives which are known to be evenly spaced in the scene (Diagram 1).

These textural primitives, since they are spacings rather than lengths, are virtual textural elements. From a viewpoint normal to the surface, these spacings can be considered to be the equally-spaced intersections of actual textural elements with an arbitrary virtual line. When the surface is viewed

in perspective to the viewer, the spacing between virtual primitives changes in an orderly way. This module uses the detection of a change in texel spacing to calculate local surface orientation by finding an equivalent representation of local surface gradient: the local vanishing line of perspective.

2. MATHEMATICAL BASIS

The derivation of surface constraints from textural cues can be performed by back-projecting the textural primitives onto a hypothesized local surface ([2], [3]). Using the image position of three or more textural elements and the hypothesized surface parameters, we are able to recover constraints on surface orientation, under the assumption that the back-projected features are regularly distributed in real space.

This equal-spacing module operates in the following way. It constructs multiple straight virtual lines through the region of interest. Each of these lines passes through a number of texel primitives; each adjacent pair of intersections with the texels produces a virtual spacing. For every group of three of more texels on the virtual line (that is, for every pair of spacings) it then computes the appropriate vanishing point implied by the ratios of the spacings.

The relationship of these spacings to the local vanishing line can also be obtained by a simple graphic construction. Given an adjacent pair of

spacings, it cuts the virtual line at the middle texel, and pivots each of the two (unequal) halves up off the surface of the image plane and perpendicular to it (see Diagram 2). The method resembles the opening of a jack knife. The line through the top of each of these 'jack knife lines' intersects the virtual line at the locally determined vanishing point. Note that if the lines do not intersect, it must be that the spacing between texels are equal and the local vanishing point must be infinitely distant to the line of sight (that is, the surface is perpendicular to the line of sight). Each vanishing point constrains the local surface to one degree of freedom in its surface orientation; two independent vanishing points (from spacings derived from two independent virtual lines) uniquely determine local surface orientation.

In practice, these constructions are not done graphically but rather by means of the back-projection formula relating texel position and the slopes of the constructed lines. Given any two texel locations a and b, if the distance from a to the midtexel is equal to L and the distance from b to the same midtexel is equal to R, the vanishing point distance is given by [3]:

$$[X - a]/L = [X - b]/R$$

where X is the vanishing point distance. Rewriting the equation we have:

$$X = [La - Rb]/[L-R]$$

In the next section we discuss some of the implementation issues that arise from this ideal mathematical basis.

3. METHODOLOGY

The quality of the algorithm's results depends on a number of factors including the location and direction of placement of the virtual lines, the determinator of what constitutes a texel as opposed to noise, and which groups of texels to use to generate vanishing points.

The optimal choice of the orientation, placement, and number of virtual lines is also related to the specific image parameters. As a general rule we have found that four equally positioned orientations of virtual lines are necessary. (They lie in the direction of a pixel's eight neighbors: see Diagram 3.) If there were only one orientation, then surface orientation would be underdetermined except for the invocation of additional assumptions such as medium-scale planarity. Additionally, with only one direction, no vanishing points can be found when the virtual lines and the texture lines are parallel. Although two orientations are theoretically sufficient, results are often inadequate when an image contains converging lines that nearly parallel one of the directions (see Diagram 4). With three orientations, it is difficult to quantize the direction of the virtual lines; with four it is straight-forward.

Given a virtual line, the algorithm aggregates the virtual texels upon it into all the possible groupings that have an even number of adjacent spacings (and therefore that even number, plus one, of virtual texels). Each such group then has two end texels, and a middle pivoting texel (see Diagram 2). Since all spacings are assumed equal, grouping together

adjacent runs of them creates new virtual spacings; if two adjacent groups have the same spacing count, they two must represent equal spacings on the surface itself. Although the theory does not strictly require equality of grouping (as long as the virtual spacings are properly normalized in the vanishing point calculation), the current method operates sufficiently well, and has the advantage of straigthforward simplicity.

We operate on images in the following way. In the synthetic, noisy images we have used, we approximate the initial edge finding step (and reduce noise as well) by a simple heurisitic thresholding of each pixel. Edge linking is also approximated: pixels above threshold are considered parts of a linear surface marking if at least one neighboring pixel is also above threshold. Since virtual spacings are highly sensitive to placement and digitization error, we further limit texels to those that are spaced greater than two pixels apart.

4. RESULTS AND ANALYSIS

A number of synthetic images have been generated and used to test the capabilities of the program. Each was choosen to show a form of image perterbation that is likely to occur in real-world textures. These images test the effects on the method of random noise, texture loss, mulitiple sufaces, and overlapping transparent surfaces. It is important to note that in each case the parallel line spacing ("PLS") algorithm has been given the image as a whole. No a priori information is supplied about the number of surfaces,

minimum noise levels, or the existence of multiple textures.

4.1. Pure Synthetic Image

The first image (the "Manhattan sunrise": see Figure 1), tests the program on a pure synthetic image. The middle portion of Figure 1, (the "waves"), contains a surface textured with horizontal parallel lines, while the upper portion (the "sun") contains converging lines. Both textured surfaces would be parallel and equal spaced when seen in the normal direction. In both of these surfaces, and in other "clean" surfaces containing no noise, the program correctly found the true vanishing line with a high degree of certainty and accuracy.

4.2. Random Noise

Random noise can either create false texels or remove true ones. The creation of false texels is partially compensated for in the algorithm by the use of the neighbor-checking step. In a sparsely textured image this does remove much of the random noise; in an image where the texture is more closely spaced or the noise is more prevalent, false texels can be retained since they fall next to other retained texels (which may be true texels or false ones).

To further decrease the effects of small numbers of false texels, we generate a large number of candidate vanishing points per virtual line: there are multiple ways in which texels can be aggregated into equal adjacent pairs of groupings. Some of these groupings will include false texels. This is

a form of noise averaging; small amounts of random noise will still create incorrect vanishing points but it appears experimentally that a greater proportion of the vanishing points generated will be correct. Note, too, that when the noise is random the noisy vanishing points will also be randomly located off the true vanishing line (where they will be ignored), although their spatial distribution is hard to quantify or analyze even under the simplest of assumptions.

Noise can also cause the loss of true texels. Figure 2 is an example of a textured surface in which a substantial portion of the horizontal parallel lines have been lost. The PLS algorithm however was able to find the correct vanishing line reliabley (see Figure 3). Other incorrect vanishing lines were also indicated, but they had less than half as many points as the true vanishing line.

Similar results occurred in every other texture-loss image, with the quality of the results, as expected, decreasing as the percentage of texture loss increases. The effect of texture loss appears greatest if the loss occurs in regions where many texture lines are seperated by only few pixels (that is, near the vanishing line itself).

4.3. Textural Variations

Another form of perturbation that can occur is the skewing of texel orientation. Figure 4 is comprised of the missing-texture image of Figure 2, with the added difficulty of a random change in orientation of every texture

line; the slope is altered by one or two pixels per hundred. This image still contains sufficient correct information such that the program found the largest density of potential vanishing points on the true vanishing line (see Figure 5).

4.4. Multiple Adjacent Surfaces

The next group of test images contains multiple non-overlapping surfaces. Since no a priori information is supplied to the PLS program, all the surfaces are initially treated as a single planar surface.

When run on the "Manhanttan sunrise" image as a whole (Figure 1), the algorithm treats much of the "sand" as random noise, due in part to the lack of neighbors above threshold, and in part because the inter-texel spacing is often too small. Thus the results obtained are mainly due to the horizontal-line and converging-line surfaces, both of which point to the same true vanishing line (that is, the surfaces are parallel in three-space). The results are seen as white dots on Figure 6, and Chart 1 maps the degree of success of the program; it found most of its vanishing points on the true vanishing line.

The next image (the "corridor"; see Figure 7) contains four non-overlapping surfaces, each of which generates a different vanishing line. Again the PLS program mechanically draws virtual lines across the image; they often intersect the texels of more than one surface. If the algorithm chooses its texture groups all within the same planar surface, a true vanishing point

is computed. If one of the texels is on a different surface instead, a false vanishing point is computed. However, the distribution of the false vanishing points is heavily dependent on the location of the virtual line and the surfaces that it travels through. These false points tend not to fall in regular patterns, nor do a sufficient number randomly fall on any one line with a frequency approaching the number of points on the true vanishing line (see Figure 8 and Chart 2).

4.5. Multiple Overlapping Surfaces

The next class of images contain an even more difficult surface constraint problem, that of overlapping transparent surfaces. Figure 9 shows two surfaces, one "on top" of the other. The first is imaged as containing parallel horizontal lines, while the second is imaged as converging lines. Both surfaces, however, have the same orientation and vanishing line.

The PLS program treats the two as a single texture; the human observer also often envisions it as a single brick floor going off to infinity. The results of this surface are shown in Chart 3 (see Figure 10). The program was able to find the vanishing line accurately, with the proper line having twice the number of vanishing points as any other candidate.

4.6. Multiple "Transparent" surfaces

Next we distort this image by changing the orientation of the converging-line plane. The resulting image, Figure 11, does create difficulties for the program, but it also often puzzles human observers. The

horizontally-lined surface's vanishing points are easily found, whereas the converging-line surfaces's are not. This appears to be because the horizontal lines at the bottom of the image are little encumbered by the converging lines; they strongly indicate their vanishing line. However, the middle section of the image, which contains the best converging-line textures, is noisy due to the co-existence of both textures.

Without additional processing--such as direction-sensitive filtration of linear texels--the PLS program is unable to find with conviction the converging-line surface's orientation.

4.7. Perspective-Based Segmentation

The PLS program is also capable of some limited perspective-based segmentation. By using the same basic algorithm, with some additional information we are able to find which texels contribute to which vanishing line. Given the equation of a vanishing line, the program can label a given texel group as belonging to the local surface or not. By including all texels that generate correct vanishing points in a seperate image, we are able to partially define and remove one surface at a time from the image.

By using this method, we were able to segment all of the previous images in which there existed more than one vanishing line. For the overlapping image of Figure 11, the PLS program finds the vanishing line of the horizontally-textured surface (see Figure 12 and Chart 4), and uses this to generate Figure 13, which contains much of the horizontally-textured

surface. Subtracting Figure 13 from Figure 11 gives Figure 14, which now is sufficiently "clean" to allow the detection of the converging-line surface's vanishing line.

5. CONCLUSION AND FUTURE WORK

The PLS algorithm has been found to be robust when used on many images of synthetic parallel-line textured surfaces. One good measure of the quality of an algorithm is how much one gets "for free". Here, segmentation does comes freely: not only are we able to constrain noisy difficult images and images with multiple surfaces, but we are able to separate two transparent images.

However, combinations of noise, texture loss, and surface overlap can cause the algorithm's performance to degrade or fail completely. These situations can probably be handled best by the concurrent use of other similar texture-based modules. Future research will include other such shape-from-texture methods, and should allow us to constrain a larger range of both synthetic and simple real images. This multi-module approach should address many of the problems of perspective-based surface segmentation and surface orientation analysis.

APPENDIX A - Charts Figures and Diagrams

Chart 1: Manattan Sunrise[1]

```
Row #       # of points
----------------------
231             2736

239             1188
240             2365
241             5606  -- the true vanishing line
242             1626
243             3385
 No rows over 2000

no cols with over 2000 points
```

Chart 2: four wall image

```
vanishing line  points
----------------------
row 240         3411
row 271.5       3684 ---  1/2 row position errror

col 239.5       3674 ---  1/2 row position errror
col 271         3414

        These are all the rows or columns over 2500 points
```

Chart 3: Overlaping image

```
Row #       # of points
----------------------
230             1161
237             1127
239             1026
240             2261  -- the true vanishing line
240             1006
260             1005
 All rows over 1000

no cols with over 1000 points
```

[1]The algorithm returns potential vanishing points; they are currently processed either by a horizontal or vertical best-fit line finder, or by a Hough transform module. The results obtained by either method agree sufficiently closely in these examples to be used interchangeably.

Chart 4: Overlapping skewed images

Row #	# of points	
230	2071	
240	3604	--- The True vanishing line
242	1842	
260	1800	

All rows over 1800

no cols with over 1800 points

Diagram 1: Parallel line textured surface

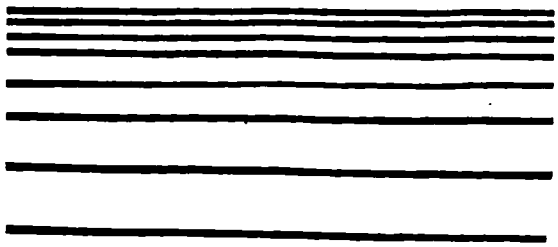


Diagram 2: Vanishing point finding using eqaul spacing method

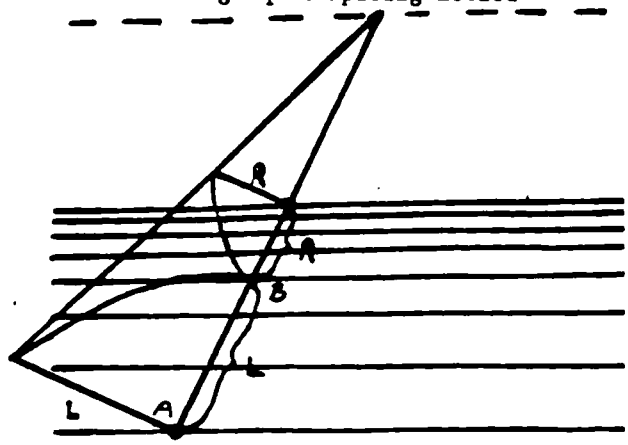


Diagram 3: Four Orientations of virtual line templates

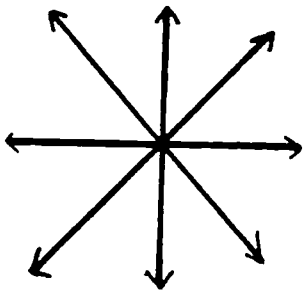


Diagram 4: Converging line texture

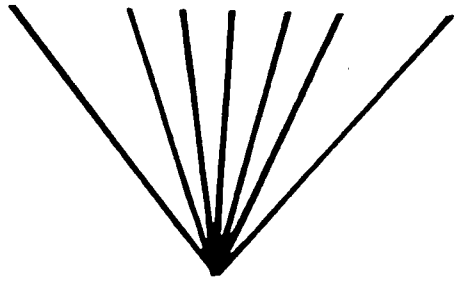


Figure 1: Manhattan Sunrise

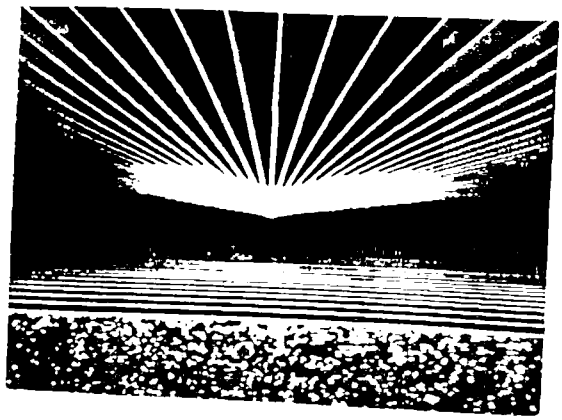


Figure 2: Horizontal line image with partial texture loss

Figure 3: Partial loss texture with result points

Figure 4: Horizontal line image with partial texture loss and position skewing

Figure 5: Previous image with result points

Figure 6: Manhattan Sunrise image with result points

Figure 7: Multiple Surface image

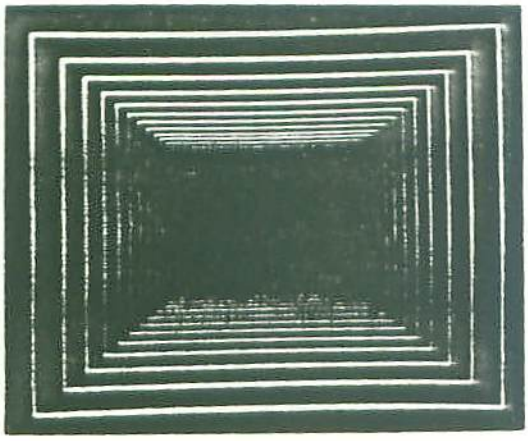


Figure 8: Multiple surface image with result points

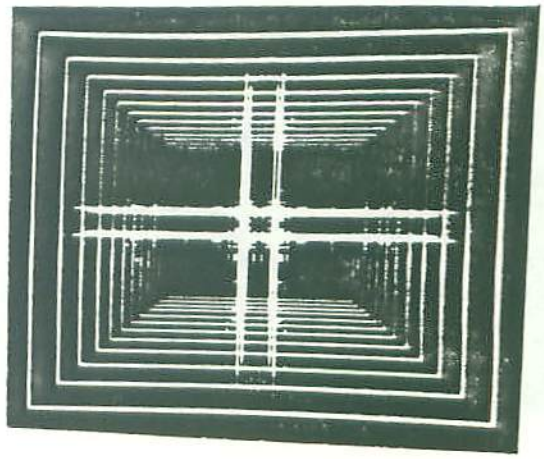


Figure 9: Two overlaping surface 'Brick Floor' Image

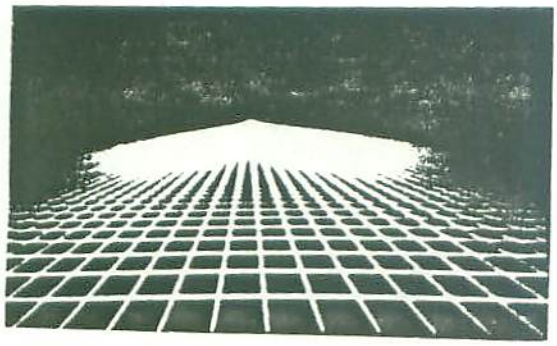


Figure 10:Brick Floor image with results

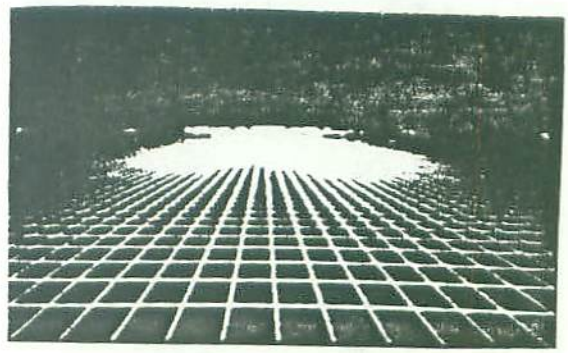


Figure 11:Two Transparent surfaces image

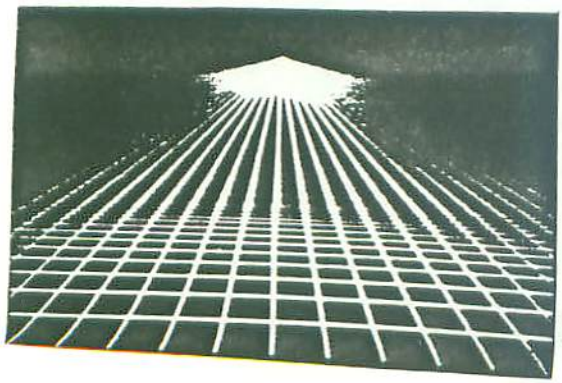


Figure 12:Two Transparent surface image results

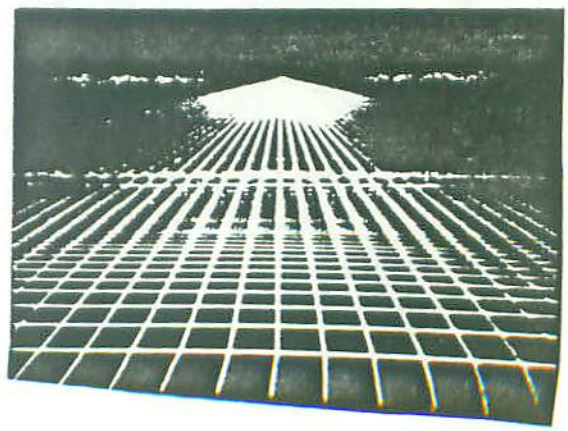


Figure 13:Horizon surface points found

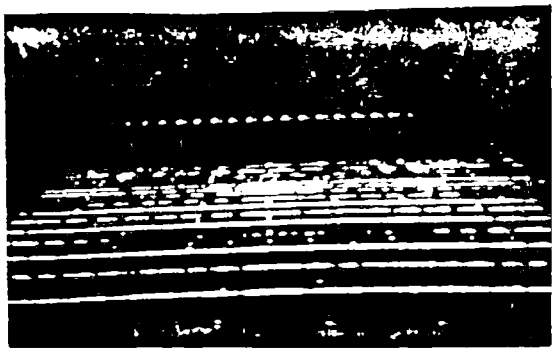


Figure 14:Results of figure 11 minus figure 12

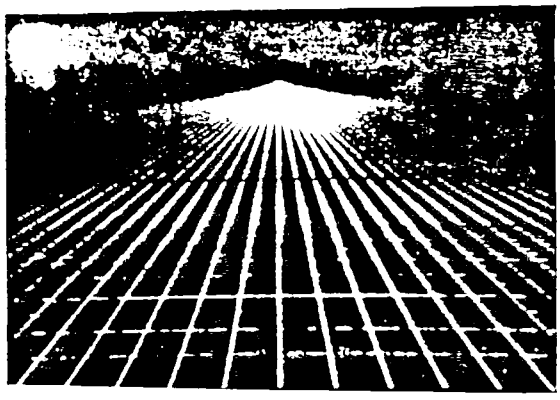
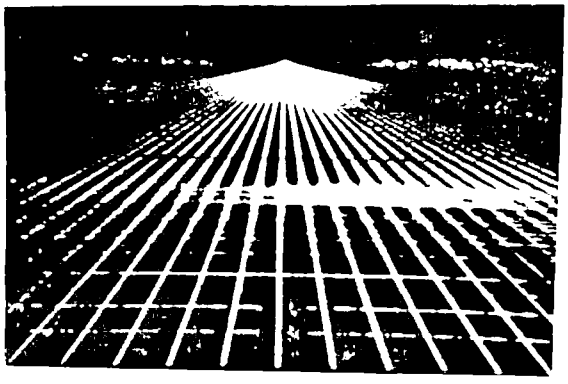


Figure 15:New Transparent image with results



References

[1] Brooks R.A.
Symbolic Reasoning among 3-D Models and 2-D Images.
Artificial Intelligence Journal. , August, 1981.

[2] Kender, J.R.
Shape from Texture.
PhD thesis, Carnegie-Mellon University Computer Science Department, 1980.

[3] Kender J.R.
Surface Constraints from Linear Extents.
Proceedings of the Nation Conference on Artifical Intelligence. , March, 1983.

[4] Sabbah D.
A Connectionist Aprroach to Visual Recognition.
PhD thesis, University of Rochester, 1982.