AN INTEGRATED SYSTEM THAT UNIFIES MULTIPLE SHAPE FROM TEXTURE ALGORITHMS

Mark L. Moerdler and John R. Kender[1]

Department of Computer Science
Columbia University
New York, N.Y. 10027

ABSTRACT

This paper describes an approach which integrates several conflicting and corroborating shape-from-texture methods in a single system. The system uses a new data structure, the *augmented texel*, which combines multiple constraints on orientation in a compact notation for a single surface patch. The augmented texels initially store weighted orientation constraints that are generated by the system's several independent shape-from-texture components. These texture components, which run autonomously and may run in parallel, derive constraints by any of the currently existing shape-from-texture approaches e.g. shape-from-uniform-texel-spacing. For each surface patch the *augmented texel* then combines the potentially inconsistent orientation data, using a Hough transform-like method on a tesselated gaussian spheres, resulting in an estimate of the most likely orientation for the patch. The system then defines which patches are part of the same surface, simplifing surface reconstruction.

This knowledge fusion approach is illustrated by a system that integrates information from two different shape-from-texture methods, shape-from-uniform-texel-spacing and shape-from-uniform-texel-size. The system is demonstrated on camera images of artificial and natural textures.

1 INTRODUCTION

This paper describes a new approach to the problem of defining and reconstructing surfaces based on *multiple* independent textural cues. The generality of this approach is due to the interaction between textural cues, allowing the methodology to extract shape information from a wider range of textured surfaces than any individual method. The method, as shown in figure 1, consists of three major phases, the calculation of orientation constraints and the generation of *texel patches*[2], the consolidation of constraints into a "most likely" orientation per patch, and finally the reconstruction of the surface.

During the first phase the different shape-from-texture components generate texel patches and *augmented texels*. Each augmented texel consists of the 2-D description of the texel patch and a list of weighted orientation constraints for the patch. The orientation constraints for each patch are potentially inconsistent or incorrect because the shape-from methods are locally based and utilize an unsegmented, noisy image.

In the second phase, all the orientation constraints for each augmented texel are consolidated into a single "most likely" orientation by a Hough-like transformation on a tesselated gaussian sphere. During this phase the system will also merge together all augmented texels that cover the same area of the image. This is necessary because some of the shape-from components define "texel" similarly, and the constraints generated should also be merged.

Finally, the system re-analyzes the orientation constraints to determine which augmented texels are part of the same constraint family and groups them together. In effect, this segments the image into regions of similar orientation. In order to build a complete system one may also want to reconstruct surfaces from these surface patches [Boult 86].

The robustness of this approach is illustrated by a

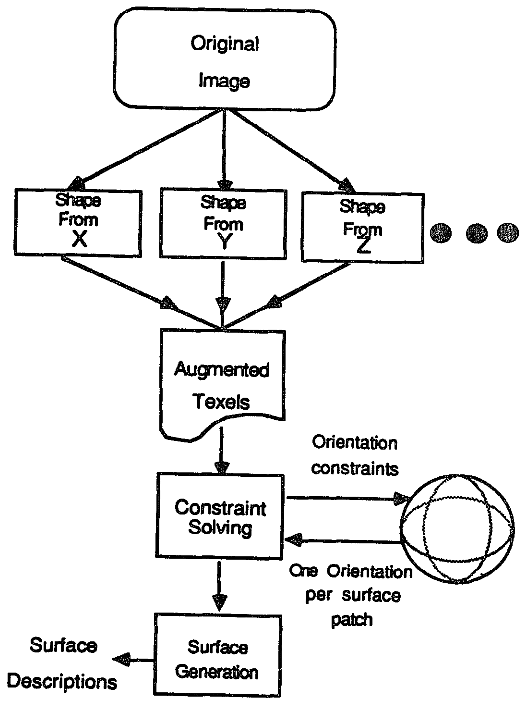

Figure 1: Integrating multiple shape-from methods

---

[1]This research was supported in part by ARPA grant #N00039-84-C-0165, by a NSF Presidential Young Investigator Award, and by Faculty Development Awards from AT&T, Ford Motor Co., and Digital Equipment Corporation.

[2]A *texel patch* is a 2-D description of a subimage that contains one or more textural elements. The number of elements that compose a patch is dependent on the shape-from-texture algorithm.

system that fuses the orientation constraints of two existing shape-from methods: shape-from-uniform-texel-spacing [Moerdler 85], and shape-from-uniform-texel-size [Ohta et. al. 81]. These two methods generate orientation constraints for different overlapping classes of textures.

2 HISTORICAL BACKGROUND

Current methods to derive shape-from-texture are based on measuring a distortion that occurs when a textured surface is viewed under perspective. This perspective distortion is imaged as a change in some aspect of the texture. In order to simplify the recovery of the orientation parameters from this distortion, researchers have imposed limitations on the applicable class of textured surfaces. Some of the limiting assumptions include uniform texel spacing [Kender 80; Kender 83; Moerdler 85], uniform texel size [Ikeuchi 80; Ohta et. al. 81; Aloimonos 85], uniform texel density [Aloimonos 86], and texel isotropy [Witkin 80; Davis et.al. 83]. Each of these are strong limitations causing methods based on them to be appliable to only a limited range of real images.

3 DESIGN METHODOLOGY

The generation of orientation constraints from perspective distortion uses one or more image texels. The orientation constraints can be considered as local, defining the orientation of individual surface patches (called *texel patches*[3]) each of which covers a texel or group of texels. This definition allows a simple extension to the existing shape-from methods beyond their current limitation of planar surfaces or simple non planer surfaces based on a single textural cue. The problem can then be considered as one of intelligently fusing the orientation constraints per patch. Ikeuchi [Ikeuchi 80] and Aloimonos [Aloimonos 85] attempt a similar extension based on constraint propagation and relaxation for planer and non planer surfaces for using only a single shape-from-texture method.

The process of fusing orientation constraints and generating surfaces can be broken down into the following three phases:

1. The creation of texel patches and multiple orientation constraints for each patch.
2. The unification of the orientation constraints per patch into a "most likely" orientation.
3. The formation of surfaces from the texel patches.

3.1 SURFACE PATCH AND ORIENTATION CONSTRAINT GENERATION

The first phase of the system consists of multiple shape-from-texture components which generate augmented texels. Each augmented texel consisting of a texel patch, orientation constraints for the texel patch, and an assurity weighting per constraint. The orientation constraints are stored in the augmented texel as vanishing points which are mathematically equivalent to a class of other orientation notations (e.g. tilt and pan as gradient constraints) [Shafer et.al. 83]. Moreover, they are simple to generate and compact to store.

The assurity weighting is defined separately for each shape-from method and is based upon the intrinsic error of the method. For example, shape-from-uniform-texel-spacing's assurity weighting is a function of the total distance between the texel patches used to generate that constraint. A low assurity value is given when the inter-texel distance is small (1 texel distance) because under these conditions a small digitization error causes a large orientation error. Above this threshold the assurity weighting is set high and then starts to decrease as the inter-texel distance increases. (The optimal shape of this assurity function is under investigation.)

3.2 MOST LIKELY ORIENTATION GENERATION

Once the orientation constraints have been generated for each augmented texel, the next step consists of unifying the constraints into one orientation per augmented texel. The major difficulty in deriving this "most likely" orientation is that the constraints are errorful, inconsistent, and potentially incorrect. A simple and computationally feasible, solution to this is to use a gaussian sphere which maps the orientation constraints to points on the sphere [Shafer et.al. 83]. A single vanishing point circumscribes a great circle on the gaussian sphere; two different constraints generate two great circles that overlap at two points uniquely defining the orientation of both the visible and invisible sides of the surface patch.

The gaussian sphere is approximated, within the system, by the hierarchical by tesselated gaussian sphere based on trixels (triangular shaped faces [Ballard et.al. 82; Fekete et.al. 84; Korn et.al. 86]. See figure 2). The top level of the hierarchy is the icosahedron. At each level, other than the lowest level of the hierarchy, each trixel has four children. This hierarchical methodology allows the user to specify the accuracy to which the orientation should be calculated by defining the number of levels of tesselation that are created.

The system generates the "most likely" orientation for each texel patch by accumulating evidence for all the constraints for the patch. For each constraint, it recursively visits each trixel to check if the constraint's great circle falls on the trixel, and then visiting the children if the result is positive. At each leaf trixel the likelihood value of the trixel is incremented by the constraint's weight. Although this is a

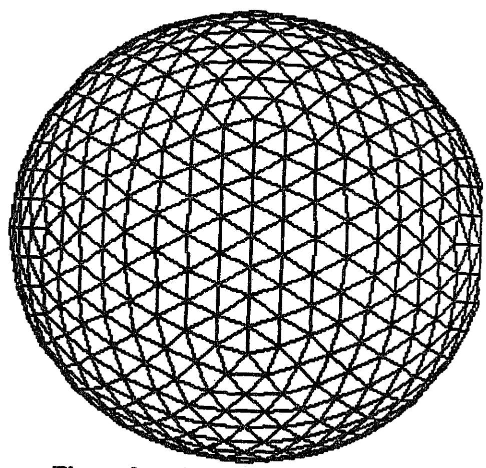

Figure 2: The trixelated gaussian sphere

[3]Texel patches are defined by how each method utilizes the texels. Some methods (e.g. Uniform texel size) use a measured change between two texels; in this case the texels patches are the texels themselves. Other methods (e.g. Uniform texel density) use a change between two areas of the image , in this case the texel patches are these predefined areas.

search process the hierarchical nature of this approach limits the number of trixels that need to be visited.

Once all of the constraints for a texel patch have been considered, a peak finding program smears the likelihood values at the leaves. Currently, this is done heuristically by a rough approximation to a gaussian blur. The "most likely" orientation is defined to be the trixel with the largest smeared value.

3.3 SURFACE GENERATION

The final phase of the system generates surfaces from the individual augmented texels. This is done by re-analyzing the orientation constraints generated by the shape-from methods in order to determine which augmented texels are part of the same surface. In doing this, the surface generation is also performing a first approximation to a surface separation and segmentation.

The re-analysis consists of iterating through each augmented texel, considering all its orientation constraints, and determining which constraints aided in defining the "correct" orientation for the texel patch as described in the previous phase. If an orientation constraint correctly determined the orientation of all the texels that were used in generating the constraint, then these augmented texels are considered as part of the same surface.

4 TEST DOMAIN

The knowledge fusion approach outlined in the previous section has been applied to a test system that contains two shape-from-texture methods, shape-from-uniform-texel-spacing [Moerdler 85], and shape-from-uniform-texel-size [Ohta et. al. 81]. Each of the methods is based on a different, limited type of texture. Shape-from-uniform-texel-spacing derives orientation constraints based on the assumption that the texels on the surface are of arbitrary shape but are equally spaced. Shape-from-uniform-texel-size is based on the unrelated criteria that the spacing between texels can be arbitrary but the size of all of the texels are equivalent but unknown.

In shape-from-uniform-texel-size if the distance from the center of mass of texel T_1 to texel T_2 (see figure 3) is defined as D then the distance from the center of texel T_2 to a point on the vanishing line can be written as :

$$F_2 = D \times S_2^{1/3} / (S_1^{1/3} - S_2^{1/3})$$

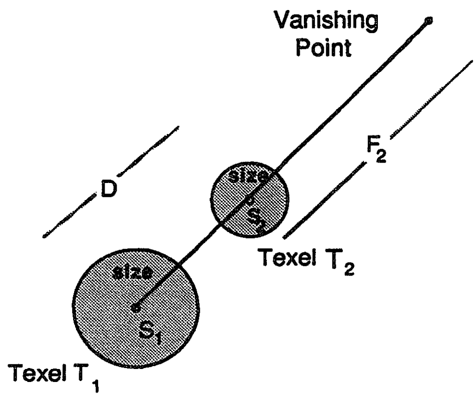

Figure 3: The calculation of shape-from-uniform-texel-size

In shape-from-uniform-texel-spacing the calculations are similar. Given any two texels T_1 and T_2 (see figure 4) whose inter-texel distance is defined as D, if the distance from T_1 to a mid-texel T_3 is equal to L and the distance from T_2 to the same mid-texel T_3 is equal to R, the distance from texel T_1 to a vanishing point is given exactly by :

$$X = [D + (R \times D)] / [L - R]$$

Under certain conditions either method may generate incorrect constraints, which the system will ignored. On textures that are solvable by both methods, they cooperate and correctly define the textured surface or surfaces in the image. Some images are not solvable by either method by itself but can only be correctly segmented and the surfaces defined by the interaction of the cues (i.e. the upper right texel of figure 13).

5 THE EFFECTS OF NOISE

Real images contain noise and shadows which are effectly ignored by the system in many cases. The system treats shadows as potential surface texels (see texels 9 and 13 in figure 5) and uses them to compute orientation constraints. Since many texels are used in generating the orientation for each individual texel the effect of shadow texels is minimized. Even under the conditions where many shadow texels are found they do not effect the computed orientation of surface texels so long as the placement of the shadow texels does not mimic perspective distortion.

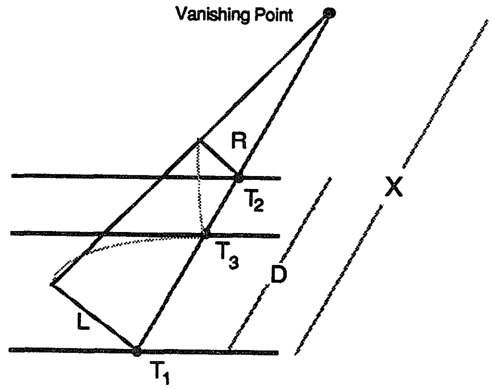

Figure 4: A geometrical representation of back-projecting.

Noise can occur in many ways: it can create texels, and it can change the shape, size, or position of texels. If noise texels are sufficiently small then they are ignored in the texel finding components of the shape-from methods. When they are large, they are treated in much the same way as shadow texels and thus often do not affect the orientation of the surface texel patches. Since many texels are used and more than one shape-from method is employed, noise-created changes in the shape of texels can perturb the orientation results, but the effect appears negligible as shown in the experimental results.

6 EXPERIMENTAL RESULTS

The system has been tested over a range of both synthetic and natural textured surfaces, and appears to show robustness and generality. Three examples are given on real, noisy images that demonstrate the cooperation among the shape-from methods.

Figure 5 shows a real image of a man-made texture consisting of equally spaced, equally sized circles. The system finds fourteen texels: the twelve texels on the surface, plus two noise texels located in the background. It is able to generate the correct gradient space *p* and *q* values for each of the twelve surface texels (see figure 6 for the positions of the texels and figure 8 for the individual *p* and *q* values.) In figure 7 the orientations of the texel patches are displayed as needle-like surface normal vectors.

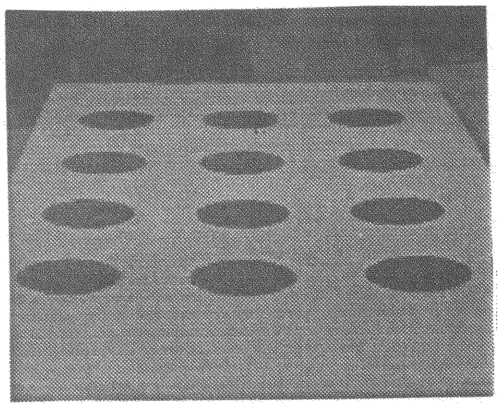

Figure 5: A texture of equal spaced and sized circles

The system is also able to segment the image into three surfaces, one of which contains only the twelve correct surface texels. The noise-generated texels are each individually marked as parts of separate surfaces.

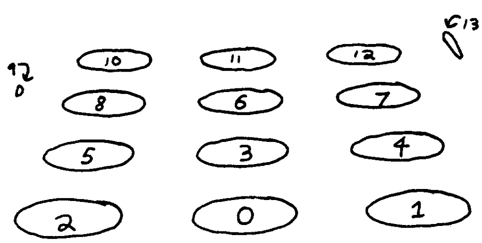

Figure 6: The numbering of texels for the circles texture

The second example consists of a surface textured with twelve uniformly sized coins (see figure 9). The system in this case ignores the incorrect orientation constraints from the shape-from-uniform-texel-spacing algorithm and generates *p* and *q* values that are correct (See figure 11) for ten of the twelve texel patches. The other four texels have an error of eight degrees in their *p* value and a correct *q* value (see figure 12). This error can be traced to a digitization error due to the reflectance of the coins.

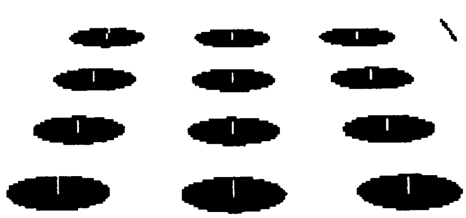

Figure 7: Surface normals for the circles texture

The final example is an image of a box of breakfast buns in which one bun is missing (see figure 13). Eleven texels are found, eight of which are the cream filling on the surface of the buns, two are the result of noisy data, and one is part of the packaging. The texels are approximately evenly spaced and approximately evenly sized.

The system is able to generate approximately correct surface normals (see figure 14) for all but one of the texels on the breakfast buns and an approximate orientation for the outside of the package. The upper left hand texel contains two equally weighted orientations, one correct and one incorrect. Without the shape-from-uniform-texel-spacing method the correct orientation would not have been found.

Texel Numbers	Measured p & q	Actual p & q	Error
0 to 8	p = 3.0 q = 0.0	p = 3.0 q = 0.0	0^o 0^o
9	p=11.0 q=6.7	Shadow Texel	
10 to 12	p = 3.0 q = 0.0	p = 3.0 q = 0.0	0^o 0^o
13	p=11.0 q=6.7	Shadow Texel	

Figure 8: Orientation values for the circles texture

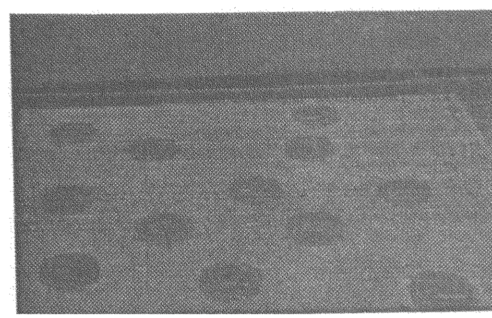

Figure 9: A surface covered with coins

7 CONCLUSION AND FUTURE RESEARCH

In this paper a system has been describes that can fuse the results of a number of shape-from-texture methodologies to generate surface segments and their orientations. The system has been tested using two existing shape-from methods; shape-from-uniform-texel-spacing and shape-from-uniform-texel-size. It has shown the ability to recover the surfaces and their orientation under noisy conditions.

The robustness of the system has been demonstrated

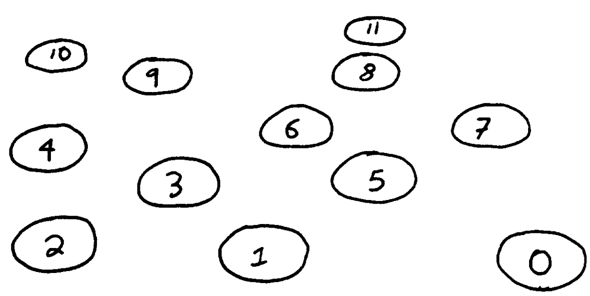

Figure 10: The numbering of the coins found

using images that contain multiple surfaces, surfaces that are solvable by either method alone, and surfaces that are solvable by using only both methods together.

Future enhancements to the system would include addition of other shape-from-texture modules, investigation of other means of fusing information (such as object model approaches), analysis of curved surfaces, studies of error behavior, and optimization of the fusion approach, especially in a parallel processing environment.

Texel Numbers	Measured p & q	Actual p & q	Error
0	p=5.5 q=0.0	p = 3.0 q = 0.0	8° 0°
1 to 4	p = 3.0 q = 0.0	p = 3.0 q = 0.0	0° 0°
5	p=5.5 q=0.0	p = 3.0 q = 0.0	8° 0°
6 to 9	p = 3.0 q = 0.0	p = 3.0 q = 0.0	0° 0°
10 & 11	p=5.5 q=0.0	p = 3.0 q = 0.0	8° 0°

Figure 11: Orientation values for the coins

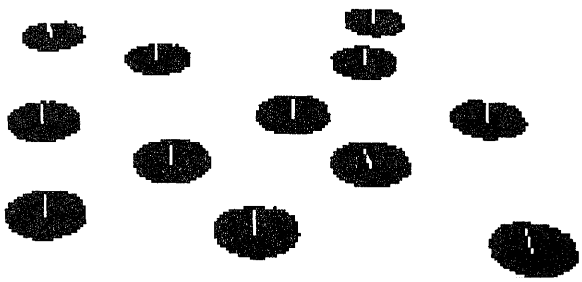

Figure 12: surface normals generated for the coins

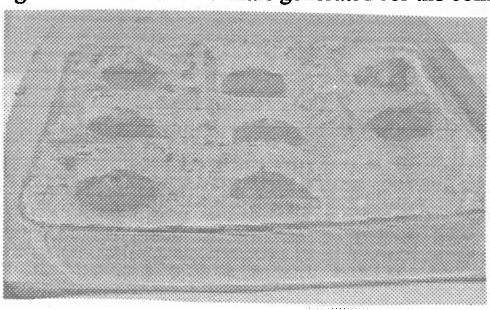

Figure 13: A box of breakfast buns with one bun missing

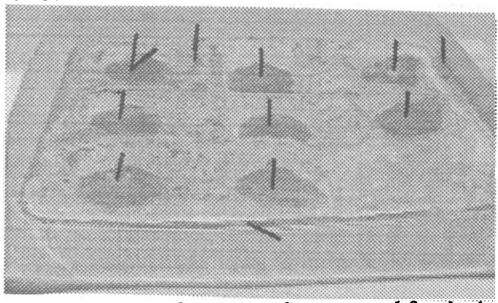

Figure 14: The surface normals generated for the buns

References

[Aloimonos 85] John Aloimonos and Michael J. Swain. Shape from Texture. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence.* IJCAI, 1985.

[Aloimonos 86] John Aloimonos. Detection of Surface Orientation and Motion from Texture: 1. The Case of Planes. *Proceedings of Computer Vision Pattern Recognition Conference* , 1986.

[Ballard et.al. 82] Dana Ballard and Christopher Brown. *Computer Vision.* Prentice-Hall Inc., 1982.

[Boult 86] Terrance E. Boult. *Information Based Complexity in Non-Linear Equations and Computer Vision.* PhD thesis, Department of Computer Science, Columbia University, 1986.

[Davis et.al. 83] L. Davis, L. Janos, and S. Dunn. Efficient Recovery of Shape from Texture. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-5(5), 1983.

[Fekete et.al. 84] Gyorgy Fekete and Larry S. Davis. Property Spheres: A New Representation For 3-D Object Recognition. *Proceedings of the Workshop on Computer Vision Representation and Control* :192 - 201, 1984.

[Gibson 50] James J. Gibson. *Perception of the Visual World.* Riverside Press, 1950.

[Ikeuchi 80] Katsushi Ikeuchi. Shape from Regular Patterns (an Example from Constraint Propagation in Vision). *Proceedings of the Internation Conference on Pattern Recognition* :1032-1039, December 1980.

[Kender 80] John R. Kender. *Shape from Texture.* PhD thesis, C.M.U., 1980.

[Kender 83] John R. Kender. Surface Constraints from Linear Extents. *Proceedings of the National Conference on Artificial Intelligence* , March 1983.

[Korn et.al. 86] M. Korn and C. Dyer. *3-D Multiview Object Representation for Model-Based Object Recognition.* Technical Report RC 11760, IBM T.J. Watson Research Center, 1986.

[Moerdler 85] Mark L. Moerdler and John R. Kender. *Surface Orientation and Segmentation from Perspective Views of Parallel-Line Textures.* Technical Report, Columbia University, 1985.

[Ohta et. al. 81] Y. Ohta, K. Maenobu, and T. Sakai. Obtaining Suface Orientation from Texels under Perspective Projection. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence.* IJCAI, 1981.

[Pentland 86] Alex P. Pentland. Shading into Texture. *Artificial Intelligence* (2):147-170, August 1986.

[Shafer et.al. 83] S. Shafer and T. Kanade and J. Kender. Gradient Space under Orthography and Perspective. *Computer Vision, Graphics and Image Processing* (24), 1983.

[Witkin 80] Andrew P. Witkin. Recovering Surface Shape from Orientation and Texture. In Michael Brady (editors), *Computer Vision,* pages 17-45. North-Holland Publishing Company, 1980.