SURFACE CONSTRAINTS FROM LINEAR EXTENTS

John R. Kender
Department of Computer Science, Columbia University
New York, NY 10027

Abstract

This paper demonstrates how image features of linear extent (lengths and spacings) generate nearly image-independent constraints on underlying surface orientations. General constraints are derived from the shape-from-texture paradigm; then, certain special cases are shown to be especially useful. Under orthography, the assumption that two extents are equal is shown to be identical to the assumption that an image angle is a right angle (i.e. orthographic extent is a form of slope or skewed symmetry). Under perspective, if image extents are assumed equal and parallel, extent again degenerates into slope. In the general perspective case, the shape constraints are usually complex fourth-order equations, but they often simplify--even to graphic constructions in the image space itself. If image extents are colinear and assumed equal, the constraint equations reduce to second order, with several graphic analogs. If extents are adjacent as well, the equations are first order and the derived construction (the "jack-knife method") is particularly straightforward and general. This method works not only on measures of extent per texel, but also on reciprocal measures: texels per extent. Several examples and discussion indicate that the methods are robust, deriving surface information cheaply, without search, where other methods must fail.*

1 Introduction

In this paper, we show how certain simple aggregate image properties involving spatial extent along one dimension can be used as cues for determining underlying three-dimensional surface orientation. Image-measurable properties such as lengths and spacings are shown to generate constraints on local surface slope in a nearly image-independent way. The derivation of these relationships is identical in analytic method ("shape from texture") and representational structure (the gradient space) to those derived for other imaging phenomena such as skewed symmetry or image slope. Thus, they provide additional surface information in a form (either equation or graph) that is easily integrable with that of other existing algorithms.

Linear extents are measurements along a straight image line of either objects (in which case they are lengths) or virtual objects (in which case they are spacings). The exact form of the input to these analyses can vary. A prior edge-detection and linking step, or a segmentation-like step is assumed. Lengths are then linear measures of image tokens such as elongated blobs, and spacings are linear measures of the virtual lines between image tokens. Spacing behaves the same way as length does; often it is more conveniently available.

*This research was sponsored in part by the Defense Advanced Research Projects Agency under contract N00039-82-C-0427.

In general, this paper follows the image understanding conventions presented in, among other places, [Kender 80a]. That is, the image coordinate system considers the z axis to be positive in the direction of view; the image itself to be plane $z=1$, which has been rotated in front of the lens at the origin; and the unit of length in the system to equal the focal length of the lens. Surfaces in the scene are locally represented by planar patches, and the surface gradient of the patch $z=px+qy+c$ is represented by the point (p,q), its gradient, in the gradient space.

The problem of deriving surface information from textural and regularity assumptions proceeds in two steps. First, the textural element--in this case an image extent--is backprojected onto all surface patches possible. A map (the "normalized texture property map", or NTPM) of the scenic measure of the component vs. the surface's parameters is recorded. The recovered scene extents are usually a function of the image extent's position and the surface's gradient. In the second step, two or more nearby textural elements are assumed to be equal in measure in the scene. Mathematically, this means that the maps can be intersected to find those surface patch parameters that generate for each texel the same measure (that is, the same texture).

2 Extents under Orthography

For the cases of spatial extent under orthography, consider an image with two extents in it. Suppose they arise from parallel extents in in the scene; this parallelism is carried over into the image. Under orthography, either extent can be translated into superimposition on the other. Thus, if they are of equal extent, they (and their NTPMs) exactly coincide and no further information about the scene is obtained. If they are *un*equal in extent, they will superimpose with unequal overlap; there can be no surface that makes them equal. Thus parallelism of equal extents under orthography provides no information about surfaces, except in this weak, negative fashion.

Now suppose the extents arise from *non*-parallel equal extents in the scene. This situation is more interesting: the image extents can be translated so that a pair of their ends will meet and form an angle. The resulting constraint equation is a messy one in terms of their lengths, their joint angle, and second powers of p and q. However, it is not difficult to prove that the constraint on surface orientation that it induces can be graphed as a hyperbola in the gradient space. The following construction shows that the hyperbola is the Kanade hyperbola [Kender 80b], which usually arises under the assumption that a given image angle is caused by a scene right angle.

Consider Figure 1, in which the two image extents forming their angle have been closed off with the addition of a line. It is well known that orthography preserves midpoints of lines; thus the image figure, with yet another line connecting the vertex to this midpoint, can be seen as a scene isosceles triangle in perspective.

Given this, the angle formed by the altitude to the base in the scene must be a right angle: this is the Kanade assumption. The surface constraint then is identically derived. Equal extents in the image under orthography therefore either give trivial results, or reduce to already known cases of image slope and angle.

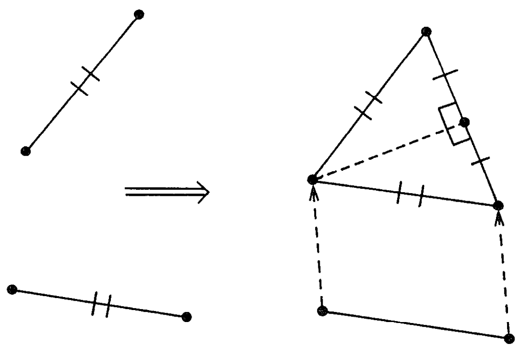

Figure 1: Equal extent is skewed symmetry.

3 Extents under Perspective

The analysis of extents under central perspective is more complex, but it yields more powerful algorithms for image understanding. The image point (x,y,1) is taken onto the surface z=px+qy+c by (-c/(1-px-qy))(x,y,1). This mapping has a non-linear Jacobian; therefore, the mapping of image extents into the scene extents is critically affected by translations in the image. It is not hard to show that there are three free parameters. . The two points (a,y,1) and (b,y,1), where L=b-a, are taken into (-c/(1-pa-qy))(a,y,1) and (-c/(1-pb-qy))(b,y,1), respectively.

The length of the induced surface extent is calculated in the scene by the usual Euclidean metric, yielding a complex NTPM:

L(1/(1-pa-qy))(1/(1-pb-qy))S(p,y), where S(p,y) is independent of both a and b.

Theoretically, this function as usable in its raw form. That is, given two extents in an image under central perspective, it is possible to generate the appropriate NTPMs for both (subject to their position and orientation), and to intersect their graphs, as if they were Hough accumulator arrays. The result would be a small set of surface orientations which would simultaneously normalize the two induced surface extents to equal measure. However, in nearly all cases, this involves the solutions to constraint equations that are of fourth order in p and q. Only a few image configurations generate simpler surface constraints. The ones that do simplify have the added benefit that they appear to be relatively common.

3.1 Equal and Parallel

First assume that image extents arose from scene components that were not only equal in measure, but were parallel on the scene surface. A simple construction will show that once again the image configuration can be handled solely by considerations of image slope. Two equal and parallel scene lines form a parallelogram; in the image, their pairs of sides can be extended to derive two vanishing points. Each vanishing point implies a linear constraint in the gradient space: if an image point (x,y) is a vanishing point of a surface, then the surface must have a gradient (p,q) which satisfies px+qy=1 ([Shafer 83]). Two such linear constraints uniquely define a vanishing line, which in turn uniquely defines the surface orientation.

3.2 Equal and Colinear

Assume now that the image extents did not arise from parallel scene extents. There seems to be only one other simplifying set of cases: those when the scene components are colinear. Interestingly, these cases do *not* reduce the problem to one of image slopes again, as colinear extents have only one slope held in common.

The images of colinear scene components are also colinear. The reverse is not true, though the heuristic positing of that truth often is most useful. It would be yet another preference heuristic, similar to those used in other contexts in image understanding: for example, nearby image pixels arise from actual scene patch neighbors (shape from shading), nearly right angles arise from scene right angles (skewed symmetry), near-parallels arise from parallels (one form of shape from texture), etc.

The image configuration in the most general case reduces to the following. Four points lie on the horizontal image line at height y; they are A=(a,y), with B, C, and D defined similarly). These four points define two image extents, L=(b-a) and R=(d-c), respectively. The assumption of colinearily allows the NTPMs of the extents to be put into correspondence easily: they are already in the proper orientation, due to the one shared image slope. Since they also share identical terms in S(p,y), equating the NTPM yields a surface constraint that reduces to second order in p and q:

$$(1\text{-}pa\text{-}qy)(1\text{-}pb\text{-}qy)/L = (1\text{-}pc\text{-}qy)(1\text{-}pd\text{-}qy)/R$$

Although this equation can be exactly solved, it has a simplifying graphic construction that can be drawn in the image space itself, directly yielding the vanishing point(s). Rewrite it in the following form:

$$(X\text{-}a)(X\text{-}b)/L = (X\text{-}c)(X\text{-}d)/R, \text{ where } X = (1\text{-}qy)/p$$

If X satisfies the constraint equation, then scene extents are equal, as desired. Further, this is a very desirable X: it also satisfies the formal definition that pX+qy=1, that is, the point (X,y) is a vanishing point. Note that (X,y) lies on the line of colinearity; all that must be calculated is the value of X itself. Formally, the equation is of the form of the intersection of two parabolae. The left parabola has value 0 at both a and b, and a minimum value of L/4 midway between them. The right parabola is exactly of the same shape, except for scaling (its midpoint minimum is R/4). Thus, the value of X can be graphically determined by drawing the parabolae on the image, and finding their intersection. (Notice that the mathematics, as well as the construction, finds a vanishing point between b and c, where the image lengths are on *opposite* sides of any vanishing line.)

The parabola method can be refined in the following way. The parabolae are only constrained to pass through the point pairs; their exact shape is not critical, as long as the parabolae are similar (i.e. they can be mutually scaled). Further, since the value of X is a purely formal one, the parabolae can be imagined to be drawn *out of* the image plane: that is, either parabola can be though of as extending into the -z axis direction.

More appropriately, the value of X on either parabola can be considered as an image feature in its own right. The calculation is really a type of local feature assignment, with with each position on the line of colinearity being assigned two simultaneous features. That position where the features are identical is the vanishing point.

Parabolae grow very quickly, however. This can be compensated for formally by taking the square root of this image feature. The assignment of values is now via hyperbolae of similar shape, which grow approximately linearly. They also have the aesthetic advantage of being undefined within the image extents themselves, the interior of which being one place where a vanishing point ought not be. In a pinch, the hyperbolae can also be approximated by their asymptotes, which, being strictly linear, are easier to compute. For example, the left hyperbola is sqrt((X-a)(X-b)/L); its asymptotes originate at the left texel's midpoint, and have slopes of sqrt(L) and -sqrt(L) (see Figure 2). Still other modifications and approximations of this formal equation are possible; they would need to be analyzed for accuracy and computational efficiency.

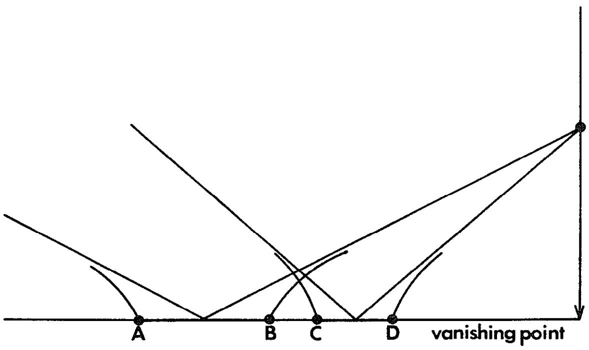

Figure 2: The hyperbola and asymptote methods.

3.3 Equal, Colinear, and Adjacent

The last special case is the simplest, but perhaps the most powerful. Suppose that two colinear and *adjacent* image extents are derived from two colinear, adjacent, and equal scene components. That is, as in Figure 3, the points B and C have merged. Then the constraint given for the general four-point colinear case simplies even further since B=C, to that of a linear constraint in p and q:

$$(1\text{-}pa\text{-}qy)/L = (1\text{-}pd\text{-}qy)/R$$

By the same formal method as above, it can be rewritten as:

$$(X\text{-}a)/L = (X\text{-}d)/R, \text{ where } X = (1\text{-}qy)/p$$

Either side is the equation of a line. With exactly the same flexibilities of the parabola scheme above, these lines can be plotted in the image space (see Figure 3). That is, they can extend out of the image in the -z direction; they can be mutually scaled; X can again be considered an image feature, labeling each position on the line of colinearity with a two-tuple of features. As before, the vanishing point occurs when the features are equal; this occurs at X=(Ld-Ra)/(L-R).

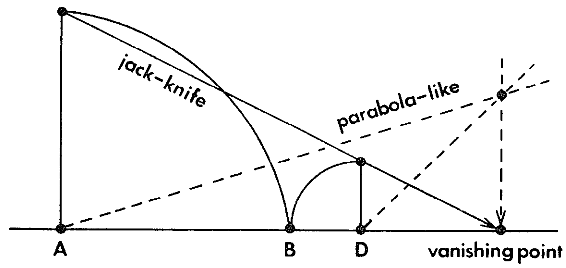

Figure 3: "Jack-knife" method for vanishing points.

Yet another graphic construction is possible. It too has a feature space interpretation, this time very useful. Construct at A a feature of value L; conceptually, this is constructed by a line of length L perpendicular to the line of colinearity. (Alternatively, the line can point in the -z direction.) Similarly construct at D a feature of value R. The resulting figure may resemble a jack-knife, with its two blades opened in parallel, outwards. (As with a jack-knife, the blades do not need to be perpendicular to their base; however, for the method to work, the blades must be parallel. The proof is by similar triangles.) Then under this interpretation, the feature values of all other points on the line of colinearity are determined by linear extrapolation from the two given ones. That is, values are generated from this new X by (R(X-a)-L(X-d))/(L+R). In particular, the vanishing point is where this image feature value is 0, as can be verified by direct substitution. It is not hard to show that this construction really does implement an image feature: it is *scaled inverse depth.*

These methods are formal; as with the parabola method, other modifications of the constraint equation are possible as well. It should be noted that the jack-knife equation can also be derived from the application of methods of projective geometry: either through the cross-ratio, or through the appropriate nine-point geometric construction. The parabola method apparently cannot, however, as it deals with five points at a time.

3.4 A Reciprocal Method

The jack-knife method has an interesting extension. The primary heuristic assumption required for its use requires only that image extents arise from equal surface extents; however, what is meant by extent can be defined in many ways. In particular, a series of N extents laid colinearly end to end on a surface can be considered either as a one extent of length N, or N of length one (or many other combinations). Often, runs of multiple extents can be obtained by looking for repeated distinguishing events along an arbitrary line through the image. (Strong edges of the same polarity, say.)

The jack-knife method, as given, would try to normalize the extent of the entire run. But under the assumption that the events form a texture, the method can be extended to normalize *each event* as well. It does this by simply dividing normalized run extent by event count to get the "average" extent of a unit event. Thus, given a run of events, the extended method divides the run into two sections, each with an image extent and an event count, and solves the modified equation relating their average unit extents.

((1-pa-qy)/L)/l=((1-pd-qy)/R)/r; l and r are event counts

Note that the run can be split in many places, and that the modified equation can be solved by any of the graphic techniques given in the jack-knife method (with L and R appropriately modified to L/l and R/r, respectively.) The optimal ways to split the run would have to be analyzed.

The jack-knife method is based on a measure of extent-per-texel; this reciprocal method uses texels-per-extent. The reciprocal method has many advantages. Detecting the events can be done by detectors of fixed image size and location. Within each detector, event counts can be recovered by simple pattern recognition techniques. The final computation is simple. In effect, the shape constraints under this method come from simple feature detectors.

3.5 Discussion and Implementation Example

The true beauty of the jack-knife methods comes from the fact that they are one-step and robust.

At least two other methods for determining surface orientation rely on an implicit searching for image "regularity"; having found it they postulate the vanishing line to be parallel to it. The remaining surface constraint is determined by algorithmic means [Bajcsy 76; Stevens 79]. Here, the two steps are integrated; "tilt" need not be found before "slant", since any two vanishing points will do.

The jack-knife methods succeed even with difficult textures or orientations. As in the wave texture of figure 4, sometimes the vanishing line direction has *no* measurable regularity; regularity-based tilt-searches must fail. The jack-knife methods will return a proper vanishing point, however, as long as they are *not* aligned with the vanishing line. The jack-knife methods even work without search on frontal ((p,q)=(0,0)) textures, in which *every* direction exhibits image textural regularity. In this case, the jack-knife methods properly return infinite vanishing points.

Consider the synthetic texture in Figure 4, of "waves" and "sand" under a "sunrise". The two surfaces defined in the image both have surface orientations of (p,q)=(0,+infinity). The jack-knife method, in a variant that splits runs into two subruns of equal event count, was run on the "wave" quarter of the image. Sample detected runs of events and their calculated vanishing points are shown. (Detection was in the horizontal, vertical, and principal diagonal directions only, and was done sparsely for purposes of illustration). The calculated vanishing points somewhat undershoot the desired vanishing line, but do lie on the same row of pixels. Further analysis, of course, is necessary to determine how accurate the method can be expected to be (for suitable definitions of "accurate" in this non-linear space).

The jack-knife methods are not difficult to implement; the extraction of features and the computations are comparatively easy. What is more difficult is the issue of applicability: to which regions of an image should they be applied? Implicitly, this is both a segmentation problem and a control vproblem. Segmentation would necessarily involve a textural segmentation of the type well-documented in the literature. Since segmentations customarily return a compressed feature-space description of each segmented region, this description would be helpful in deciding whether the heuristic assumptions behind the methods are

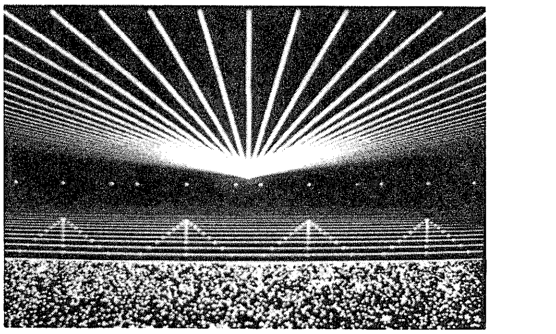

Figure 4: Jack-knife methods on a synthetic texture.

justified there. Control is far more difficult. Except in scenes of isolated, well-defined surfaces, much sophistication is necessary in apportioning belief amongst the results of several such methods' simultaneously calculated results. Integrating the algorithms' computed vanishing points--errorful and often contradictory--into a single surface synthesis appears to be a major challenge.

Acknowledgements

I thank Kerny Calaway for her graphic skills. Mark Moerdler implemented the algorithm shown in Figure 4.

References

[Bajcsy 76] Bajcsy, R., and Lieberman, L. "Texture Gradient as a Depth Cue." *Computer Graphics and Image Processing 5*, 5 (March 1976), 52-67.

[Kender 80a] Kender, J.R. *Shape from Texture.* Ph.D. Thesis, Carnegie-Mellon University Computer Science Department, Nov. 1980.

[Kender 80b] Kender, J.R., and Kanade, T. Mapping Image Properties into Shape Constraints: Skewed Symmetry, Affine-Transformable Patterns, and the Shape-from-Texture Paradigm. Proceedings of the First Annual National Conference on Artificial Intelligence, American Association for Artificial Intelligence, Aug., 1980, pp. 4-6.

[Shafer 83] Shafer, S.A., Kanade, T., and Kender, J.R. "Gradient Space under Orthography and Perspective." *Computer Graphics and Image Processing* (To appear 1983). Also available as CMU Technical Report CMU-CS-82-123, May, 1983

[Stevens 79] Stevens, K.A. *Surface Perception from Local Analysis of Texture and Contour.* Ph.D. Thesis, MIT Artificial Intelligence Lab., Feb. 1979. Available as AI-TR-512