SRI International

CLASSIFICATION-BASED TRACKING OF OBJECTS AND MATERIALS

Technical Note 443

July 1988

By: Kenneth I. Laws, Computer Scientist
Artificial Intelligence Center
Computer and Information Sciences Division

SRI Projects 2000 and 8388

The work reported herein was supported by the Defense Advanced Research Projects Agency under Contract Nos. MDA903-86-C-0084 and DACA76-85-C-0004.

SRI International®
333 Ravenswood Ave. • Menlo Park, CA 94025
(415) 326-6200 • TWX: 910-373-2046 • Telex: 334-486

Abstract

SRI's KNIFE image analysis system can be used for tracking objects and material classes from one image to another. Variations on this theme are the initial acquisition of target instances from database signatures and the subsequent acquisition of additional instances in an image once a few objects have been labeled. Classification-based tracking is facilitated by improved color and texture-energy transforms. KNIFE's labeling and partitioning methods can be used with complex targets, and are relatively unaffected by occlusions and changes in object appearance during tracking.

Contents

1 Introduction

The KNIFE digital-image analysis system is a software assistant for extracting, labeling, and editing such regions of an image as houses and vegetation in aerial imagery. This report describes tools for tracking scene objects automatically through a sequence of images. The examples emphasize tracking of roads and roadside objects through images acquired from an autonomous land vehicle.

Systems that track objects can exploit their shape, brightness, color, texture, range, or trajectory. I have found the generalized Hough transform [Sloan 80; Ballard 81; Laws 82] to be an effective approach to 2-D-shape tracking, especially when targets may be partially occluded by other objects. Correlative matching, particularly hierarchical or coarse-fine matching [Moravec 80; Hannah 84, 85], is another useful technique for tracking the shape and distinctive markings of an object, especially if combined with a robust method for rejecting false matches [Bolles 81; Fischler 81].

In some applications, shape alone is not sufficiently informative. Road tracking is one since the road orientation and cross section can change significantly between successive images. Shapes of roadside objects are also affected by perspective, occlusion, and shadow effects. New objects or vegetational patches entering the field of view must also be identified; this requires reasoning about material classes as well as regional shape.

A type of material can often be identified by its brightness, color, or texture, although sometimes high-level reasoning about shape and context is necessary. My KNIFE program uses local pixel or neighborhood properties to extract regions that can be passed on to other analytical systems [Fua 85ab, 86, 87ab]. It can either extract homogeneous regions and then label them or label pixels and then extract connected regions. All such partitioning operations are integrated with noise-filtering techniques that prevent textured regions from being split too finely. Segmentation and labeling are usually done autonomously, although the program can be used interactively and may someday be controlled directly by a sophisticated reasoning system [Wesley 83, 84, 86; Fischler 88; Strat 88].

2 Knowledge Representation

The heart of the KNIFE program is its region representation. Most recursive segmenters build a tree to describe how large regions have been divided into smaller ones, as well as to keep track of hypothesized subregions during partitioning. The problem that arises with a tree, or with any technique based on strict spatial hierarchy, is that it does not permit arbitrary merging and attachment. The merging of neighboring regions from different parts of the tree will produce new regions of uncertain parentage. It also destroys the rectilinear alignment used in quadtree representations, forcing abandonment of the homogeneous tree representation if further partitioning is contemplated. Trees pose additional difficulties as cognitive representations supporting the user interface: they sometimes necessitate unintuitive inclusion relationships (such as grouping a mountain's snowcapped peak with sky and cloud regions) and, after merging, do not show how regions were extracted or how they differ from siblings and neighbors.

KNIFE uses a directed acyclic *region graph*, permitting any *region record* to be attached to any other in a parent/child relationship.[1] It also has a *region map* for determining re-

[1]The only exception is the whole-image region, which has no parent, and a pseudoregion representing

gional statistics and spatial relationships. The graph serves as a semantic net, recording spectral or semantic relationships among regions (rather than the partitioning history). Disconnected composite regions—i.e., regions containing nonadjacent subregions—are permitted and are particularly useful for representing material types, occluded objects, and collections of similar or related objects. A road obscured by telephone poles, for instance, could be represented in two ways: as one set of subregions linked to a common "road" composite, and as another set linked to a "telephone pole" composite. A region can belong to more than one such set without there being an imposed spatial or semantic hierarchy.

Region records contain both simple descriptors (e.g., area) and pointers to histograms and other knowledge-base objects. Most fields, including numeric descriptors, can contain a "null" or "unknown" code marking information that is to be computed when needed. Subroutines that access these fields know how to obtain such values from the region map or graph, and will attach the result to the region record so that it will not have to be recomputed if it is needed again. (They also null descriptors that have become incorrect because of editing operations on the region graph.) This "lazy evaluation" saves considerable computation and storage, although it does cause certain analysis-and-display operations to consume varying amounts of time, depending on what has been stored.

From one standpoint, the entire KNIFE system is only a coordinated tool kit of subroutines for manipulating region graphs and maps. Just as a mail-handling program allows a sequence of messages to be read and manipulated, KNIFE offers the capability to display and manipulate partitioned images. Its library of subroutines forms a language for implementing the pseudo-English commands available to the user. A great deal of care has gone into making this a convenient and efficient language, although much remains to be done. The following sections describe some of the algorithms that make KNIFE an effective tool for extracting, identifying, and tracking scene objects.

3 Preprocessing Algorithms

Local properties are any measurements or estimates that can be made about the scene content at a particular pixel location. Intrinsic scene properties, such as surface albedo and orientation, are generally not measurable, so empirical properties such as local image brightness and gradient must be used. Useful properties are those that are characteristic of particular material types or that change very slowly (spatially and temporally) for any particular object. Different local properties may be useful for different objects or parts thereof.

Tracking systems may use radar, range, thermal infrared, or other imagery with special characteristics that dictate particular methods of analysis. I shall assume that the imagery to be analyzed here is typical monochrome or color/multispectral imagery, or can be treated as such. My goal is to extract as much useful information as possible without requiring task-specific or target-specific knowledge. Semantic target-recognition criteria, such as shape, context, and trajectory dynamics, could be added.

Monochrome imagery may be adequate for simple tracking tasks like following a plane against the sky, or for simple object extraction tasks like identifying suburban houses by their bright roofs. Changing imaging conditions across a scene or among scenes make it

the scene area just outside the image.

difficult to distinguish more than a few materials, but distinctive classes, such as cement, asphalt, and soil, can often be separated by brightness alone.

In complex scenes, color or multispectral data may be useful enough to justify the extra cost of collection, transmission, storage, and processing. When color is not available, it can often be replaced by texture bands computed from monochrome data—at least for identifying pixels far from region borders. The KNIFE program currently treats all data bands alike and will perform an analysis with as few or as many data bands as are available. It performs most efficiently, however, when each band encodes information that is not readily available from the others.

3.1 Color Representation

Digital color imagery is typically gathered in red-green-blue (RGB) form, with 256 possible luminance levels per pixel in each band. Often the bands are uncalibrated and individually stretched or scaled, making it difficult to identify material classes by using a database of reference signatures. Fortunately, this has little effect on tracking of object instances from one image to another (or to another portion of the same image). I shall assume that the RGB cube is a vector space with origin at (0,0,0), and shall develop methods of histogram-based analysis that are insensitive to consequent hue instabilities.

RGB bands of natural scenes tend to be highly correlated and hence redundant. I have found intensity-hue-saturation (IHS) representations more useful for partitioning and tracking operations. The exact transform employed makes little difference, but results are best if computed saturation is near zero for all colors near the achromatic axis. (The chromaticity saturation formula most commonly used in computer vision [Tenenbaum 74; Ohlander 75, 78; Kender 76, 77; Price 76],

$$S = S_{\max} \cdot (1 - 3\frac{\min(R,G,B)}{R+G+B}) ,$$

assigns low saturation values to all colors near white and high, unstable values to most colors near black. Saturation thus correlates negatively with brightness.) The National Television Systems Committee YIQ system,

$$\begin{aligned} Y &= 0.509R + 1.000G + 0.194B \\ I &= 1.000R - 0.460G - 0.540B \\ Q &= 0.403R - 1.000G + 0.597B , \end{aligned}$$

is awkward to represent digitally and less useful than IHS for segmentation [Laws 85], perhaps because the Q band contains very little energy or information for natural scenes. Ohta's approximate Karhunen-Loeve color transform [Ohta 80],

$$\begin{aligned} I_1 &= R + G + B \\ I_2 &= R - B \\ I_3 &= 2R - (G + B) , \end{aligned}$$

is easier to encode, but performs much like the YIQ system.

I use the following vivdness-hue-saturation (VHS) system.[2] The bands are sufficiently decoupled and informative, and my methods of analysis sufficiently matched to them, that

[2]No relation to the VHS videotape format.

I have not had to retain additional redundant color bands as other researchers have done [Ohlander 75, 78; Price 76].

Vividness (V) is a measure of intensity or brightness, defined to be the length of a color vector expressed as a fraction of the maximum length it could have within the RGB color cube. (In other words, we divide the distance from the origin to a color point by the distance to the intersection of this ray with a cube surface. An equivalent formula is given below.) Thus, pure spectral colors—magenta, red, yellow, green, cyan, blue, and all of the intermediate colors along the edges of the color cube—have the same vividness as white. Maximally bright desaturated colors, such as shocking pink, are also fully vivid.

Shifted hue (H) is the customary zero-to-2π angular measure, but with the scale's zero rotated from pure red by $\pi/3$ to place the discontinuity at magenta. This permits the circular nature of the hue scale to be ignored, since very few objects in natural imagery are magenta.[3] My classification methods do not require Gaussian histogram peaks, so the effect of splitting a histogram peak between the two ends of the color scale is minimal. (Analysis methods that are suitable for circular features could be incorporated if needed.) A special code is used to mark colors on the achromatic axis, for which hue is meaningless.

Cylindrical saturation (S) is the distance of a color point from the achromatic axis, expressed as a fraction of the maximum possible distance for any vivid color. The nature of the RGB cube constrains all colors near the black or white corners to have low saturation. Primary colors and their complements have the maximum saturation, while other pure spectral colors within the RGB cube are slightly below the maximum.

Formulas for converting RGB sensor coordinates to VHS coordinates are as follows:

$$V = \max(R, G, B)$$

$$H = \begin{cases} \frac{2\pi}{3} + \arctan\frac{\sqrt{3}(G-R)}{G-B+R-B} & R > B, G > B \\ \frac{4\pi}{3} + \arctan\frac{\sqrt{3}(B-G)}{B-R+G-R} & G > R \\ 2\pi + \arctan\frac{\sqrt{3}(R-B)}{R-G+B-G} & B > G \\ \frac{\pi}{3} & R > B \\ \text{ACHROMATIC} & \textit{otherwise} \end{cases}$$

$$S = \sqrt{\frac{(R-B)^2+(R-G)^2+(B-G)^2}{2}}$$
$$= \sqrt{R(R-G-B)+G(G-B)+B^2}$$

where each condition on H is tested in turn. H is also taken modulo 2π, so that amount is subtracted if H exceeds 2π. (This can occur only in the third line of the formula, which computes hue for the bluish-red third of the chromaticity triangle.) V and S can be normalized to be between 0 and 1, although the indicated formulas are more convenient if integer results comparable to the original RGB coordinates are desired. I typically scale H to lie in the range [0,179] with ACHROMATIC = 255.[4]

[3] Exceptions may occur in images of distant mountains.

[4] A code of 0 for the achromatic axis might be preferable for many uses, but is misleading because the achromatic colors should be considered equally close to or far from all other hues.

This color space can be viewed as a cone of height V_{max} having black as the point at the bottom, white as the center of the top end, and the pure spectral colors distributed around the top circle at distance S_{max} from the achromatic axis. The cone may also be envisioned as embedded in a cylinder of radius S_{max}, with black at the bottom center, or (0, ACHROMATIC, 0) position, but colors outside the central cone are also outside the range achievable by most digital (or RGB) sensors and displays.

Whether terms such as "vividness" and "saturation" are psychophysically realistic depends partly on the imaging and display systems employed. I would not expect pure blue and green to be as "vivid" as white and yellow, although they are so treated by my VHS system. Suffice it to say that partitioning and labeling performance is not dependent on precise psychophysical modeling.

3.2 Texture Representation

Local texture properties are more difficult to characterize. Different textures exhibit their distinctive variations at different spatial scales. The texture elements themselves (e.g., gravel, blades of grass, plant stems, tree leaves, roof tiles) are often below the resolving power of our imaging systems, forcing us to characterize texture patterns statistically. One approach is to determine the typical element size or coarseness for any texture patch; another is to determine characteristics that are constant across all scales of measurement [Pentland 84]. My own efforts have concentrated on a class of modified "texture energy" operators [Laws 79, 80ab] that respond to the density and contrast of different spatial patterns in much the same way that simple edge detectors respond to image gradients.

Although I shall explain my approach in some detail, there are many image textures for which brightness and local variance (especially the log of local Gaussian-weighted variance [Laws 88b], as discussed below) are the most useful descriptors. It is rare in natural imagery for adjacent texture patches to have the same brightness and variance (or color and variance), yet differ in directionality or higher-order properties. Sophisticated statistical measures of texture are thus unneccesary for image partitioning, nor are they often of much additional help in material labeling.

I derive a texture measure in two steps. (These are computed during a single pass through the image, but are easier to understand as two separate passes.) First, I convolve the image with a small center-weighted matrix or *mask* such as the 3×3 masks in Figure 1. Next, I apply a Gaussian-weighted local variance operator to the filtered image and report the log of the variance as a local texture descriptor. In other words, I compute log variance of the filtered image in a square window, giving more weight to pixels near the window center.

My original texture energy measures used unweighted windows, which caused blocky artifacts in the outputted image around bright or dark points in the input. Standard deviation within the window was approximated by a sum of absolute values—for zero-mean filter outputs—but variance has better theoretical properties and is almost as easy to compute. Logarithmic output helps match human textural perception and, moreover, can be stretched to make good use of an 8-bit output scale. (Log variance also avoids any square-root operation and differs from log standard deviation only by a factor of two.) I have not conducted rigorous experiments, but the subjective improvement in my texture energy measures seems worth the small computational cost of the Gaussian-

$$\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 2 & -1 \\ -2 & 4 & -2 \\ -1 & 2 & -1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & -2 & 1 \\ 0 & 0 & 0 \\ -1 & 2 & -1 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -2 & -1 \\ 2 & 4 & 2 \\ -1 & -2 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & -1 \\ -2 & 0 & 2 \\ 1 & 0 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

Figure 1: Orthogonal 3 × 3 Texture Masks

weighted variance operation.

A vector of texture energy descriptors can be generated for each pixel by using different texture energy masks or different sizes of variance window; that vector can then be used for material classification or image partitioning. Energy-gathering windows of no more than two or three times the width of the filter mask are recommended; larger sizes give better classification accuracy when applied to large texture patches, but lack the resolution needed for analysis of typical imagery.

In some cases I have used 3 × 3 or even 1 × 1 variance windows (i.e., with no variance computation), since my histogram-based classification and segmentation methods do not depend on the inherent smoothing of larger windows [Laws 88ab]. (This raises interesting questions about the nature of optimal or human texture perception. The "most local" frequency or joint-probability texture measures are typically the most powerful in natural imagery, a fact that was not appreciated when early comparative studies were done [Dyer 76; Weszka 76]. I have yet to investigate the full power of gradient/curvature representations and local histograms or information-theoretic statistics for texture characterization.)

I tend to work with 3 × 3 pattern masks in 3 × 3 or 9 × 9 variance windows (depending on the task), although most other workers have chosen the larger set of 5 × 5 masks in 15 × 15 or 31 × 31 unweighted windows that I originally proposed. (Note that my new measures are even more local than these numbers would imply, since Gaussian weighting of the variance window reduces the effect of all pixels near the window edges.) An alternative to varying the mask and window sizes is to compute identical texture energy measures at each level in a pyramid of image resolutions [Larkin 83]. Small texture operators tend to mimic edge detectors; to exploit texture in natural images, however, local measures must somehow be incorporated.

I have developed a particularly simple class of basis vectors, the lattice aperture waveform sets [Laws 79, 80ab], derived by repeatedly convolving the vectors [1, 1] and [−1, 1]

(or any of the resultant vectors) to obtain basis vectors of any desired length and sequency.[5] Horizontal and vertical pairs (possibly from sets of different order) can be convolved to form separable rectangular masks, such as those in Figure 1, each giving rise to a different texture energy measure. The vectors can also be convolved directly with an image to achieve the same effect as convolving with the rectangular mask, usually with a significant computational saving. Third-order and fifth-order basis-vector sets are

$$\begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$
$$\begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} -1 & -2 & 0 & 2 & 1 \end{bmatrix}$$
$$\begin{bmatrix} -1 & 2 & -1 \end{bmatrix} \qquad \begin{bmatrix} -1 & 0 & 2 & 0 & -1 \end{bmatrix}$$
$$\begin{bmatrix} 1 & -2 & 0 & 2 & -1 \end{bmatrix}$$
$$\begin{bmatrix} 1 & -4 & 6 & -4 & 1 \end{bmatrix}$$

(I have negated—or, equivalently, reversed—the originally published form of the fourth fifth-order vector to maintain consistent sinusoidal phase. Antisymmetric vectors should have a negative term just before the central position.)

Joint classification accuracies of the original texture energy measures are listed in Tables 4–2 and 7–5 of [Laws 80a]. The 3×3 measures in 15×15 windows were about 85% accurate in a difficult eight-class linear-discriminant problem; the 5×5 measures were about 87% accurate; and accuracy of the 7×7 measures was about 88%. Performance remained at 88% when all 83 of these measures were combined. A particular set of 32 co-ocurrence measures was only 71% accurate on the same problem. (I am unaware of any studies using quadratic or nearest-neighbor classification.)

I seldom use more than one or two of the eight zero-mean 3×3 filter masks. While individual texture-energy measures are less powerful than full sets (as detailed in my dissertation), the following quotation [Pietikäinen 82, 83] is reassuring:

> The best [3×3 and 5×5] Laws features correctly classify 25 out of 28 samples, or nearly 90%, a remarkable result for a one-feature, seven-class task.

Corresponding statistics for co-occurrence and "edge per unit area" measures were 71% and 68%. Their own modifications of texture energy achieved 86%, or 93% when a particular nonmaximal-suppression step was added. (The nonmaximal-suppression result is interesting and warrants further study, as does a rank transformation used elsewhere [Harwood 83]. I disagree, however, with Pietikäinen's conclusion that the power of texture energy measures "depends on the general forms of the masks (edge-like, spot-like, etc.), rather than on the specific numerical values ..." While this is true to some degree, Chapter 6 of my dissertation shows that pattern detectors of the same general type can vary greatly in individual classificatory power, and that few approach the power—documented in Figure 7–3—of the texture energy measures.)

For the record, the seventh- through eleventh-order basis-vector sets are shown in Figure 2. I use the zero-sequency vectors for the Gaussian window weighting mentioned previously. Two-dimensional integer convolution with such vectors can be accomplished in 32-bit registers (with minimal truncation error) if care is taken in scaling the intermediate

[5] Sequency is the number of interior zero crossings and is related to frequency. Convolving two basis vectors from any of these sets produces another vector with sequency equal to the sum of the original sequencies.

$$
\begin{bmatrix}
1 & 6 & 15 & 20 & 15 & 6 & 1 \\
-1 & -4 & -5 & 0 & 5 & 4 & 1 \\
-1 & -2 & 1 & 4 & 1 & -2 & -1 \\
1 & 0 & -3 & 0 & 3 & 0 & -1 \\
1 & -2 & -1 & 4 & -1 & -2 & 1 \\
-1 & 4 & -5 & 0 & 5 & -4 & 1 \\
-1 & 6 & -15 & 20 & -15 & 6 & -1
\end{bmatrix}
$$

$$
\begin{bmatrix}
1 & 8 & 28 & 56 & 70 & 56 & 28 & 8 & 1 \\
-1 & -6 & -14 & -14 & 0 & 14 & 14 & 6 & 1 \\
-1 & -4 & -4 & 4 & 10 & 4 & -4 & -4 & -1 \\
1 & 2 & -2 & -6 & 0 & 6 & 2 & -2 & -1 \\
1 & 0 & -4 & 0 & 6 & 0 & -4 & 0 & 1 \\
-1 & 2 & 2 & -6 & 0 & 6 & -2 & -2 & 1 \\
-1 & 4 & -4 & -4 & 10 & -4 & -4 & 4 & -1 \\
1 & -6 & 14 & -14 & 0 & 14 & -14 & 6 & -1 \\
1 & -8 & 28 & -56 & 70 & -56 & 28 & -8 & 1
\end{bmatrix}
$$

$$
\begin{bmatrix}
1 & 10 & 45 & 120 & 210 & 252 & 210 & 120 & 45 & 10 & 1 \\
-1 & -8 & -27 & -48 & -42 & 0 & 42 & 48 & 27 & 8 & 1 \\
-1 & -6 & -13 & -8 & 14 & 28 & 14 & -8 & -13 & -6 & -1 \\
1 & 4 & 3 & -8 & -14 & 0 & 14 & 8 & -3 & -4 & -1 \\
1 & 2 & -3 & -8 & 2 & 12 & 2 & -8 & -3 & 2 & 1 \\
-1 & 0 & 5 & 0 & -10 & 0 & 10 & 0 & -5 & 0 & 1 \\
-1 & 2 & 3 & -8 & -2 & 12 & -2 & -8 & 3 & 2 & -1 \\
1 & -4 & 3 & 8 & -14 & 0 & 14 & -8 & -3 & 4 & -1 \\
1 & -6 & 13 & -8 & -14 & 28 & -14 & -8 & 13 & -6 & 1 \\
1 & -8 & 27 & -48 & 42 & 0 & -42 & 48 & -27 & 8 & -1 \\
-1 & 10 & -45 & 120 & -210 & 252 & -210 & 120 & -45 & 10 & -1
\end{bmatrix}
$$

Figure 2: High-Order Lattice Aperture Waveform Sets

results. These vectors differ only slightly from the "bandpass binomial windows of even order" or the difference-of-binomial vectors investigated by Nicholson [80], a particularly simple basis set to implement as real-time digital filters. Rectangular masks formed from any of these vector sets resemble sampled 2-D Gabor-filter masks [Daugman 84, 85], said to resemble the spatial response of certain neurons in the human visual system. Many other derivations of such filters have been reported [Danielsson 80, 85; Hashimoto 83; Knutsson 83; Tanimoto 78], including an interesting hexagonal form [Nicholson 80].

Gabor filters, used in 1-D detection theory, are Gaussian-weighted sinusoids with sine and cosine phase. Paired waveforms are combined to determine target position and frequency, with minimum joint uncertainty in the two measurements. Each Gabor measurement is essentially a local 2-D Fourier coefficient measured at one pixel position. Ratios of these coefficients for different frequencies can be used to estimate local fractal dimension [David Heeger, unpublished], since true fractals have a $1/f$ relationship between energy and spatial frequency.

The theory of separable 2-D Gabor filters leads to an easy way of constructing rectangular masks with any orientational sensitivity. Rectangular masks that approximate oriented 2-D Gabor filters with an integral number of pixels per cycle along the horizontal and vertical axes can be constructed by multiplying corresponding terms of two vectors with appropriate frequencies:

$$S(x,y) = S(x)C(y) + C(x)S(y)$$

$$C(x,y) = C(x)C(y) - S(x)S(y),$$

where x is the horizontal position, y is the vertical position, $S(x)$ and $C(x)$ are vectors with sine (antisymmetric, or odd) and cosine (symmetric, or even) phase, and $S(x,y)$ and $C(x,y)$ are the two masks for the orientation determined by the arctangent of the relative cycle lengths along the two axes [Heeger, unpublished]. If equal horizontal and vertical cycle lengths are used, the following masks with 45-degree orientation are generated:[6]

$$\begin{bmatrix} 2 & 2 & 0 \\ 2 & 4 & 2 \\ 0 & 2 & 2 \end{bmatrix} \quad \begin{bmatrix} 0 & 2 & 2 \\ -2 & 0 & 2 \\ -2 & -2 & 0 \end{bmatrix} \quad \begin{bmatrix} 2 & -2 & 0 \\ -2 & 4 & -2 \\ 0 & -2 & 2 \end{bmatrix} \quad \begin{bmatrix} 0 & 2 & -2 \\ -2 & 0 & 2 \\ 2 & -2 & 0 \end{bmatrix} .$$

Reversing (or negating) one of the sine vectors rotates the direction of maximal response by 90 degrees. Although these diagonal masks are linear combinations of the masks in Figure 1, and corresponding filtered images are linear combinations of the original filtered images, their local-variance or texture energy measures need not be derivable from the horizontal and vertical texture energies.

The Gabor-filter approach differs from texture-energy filtering in that a sum of squared sine and cosine responses is used. (An arctangent of the ratio may also be computed to obtain a phase or target-position estimate.) Although provably optimal in certain detection problems, this is not suitable as the filtering step in texture energy computation. Gabor filters of a given frequency are invariant to positional (or phase) shift in that frequency, making them insensitive to some rather drastic deformations in spatial target signatures. Gabor filters also tends to "lock onto" to strong image structures as they are scanned across the image, thus blurring the estimate of texture element location. The

[6]These can obviously be scaled down by a factor of 2.

single-mask outputs used in texture energy computation are keyed more to edges and spots than to individual frequencies, and are more closely tied to the mask position.

Although the extra directional freedom of Gabor masks may be of use in a full vision system (as it exists in humans), I have not found the extra texture energy measurements derived from them to be obviously useful in road tracking. I have also not found fractal texture measures (computed as ratios of Gabor-filter outputs) to be useful, primarily because the low-frequency masks are much larger than typical texture elements in this imagery.

My experience is that one should generally use the lowest-sequency zero-sum texture-energy vectors or, equivalently, the smallest filter size for a given cycle length. High-sequency filters gain frequency resolution at the expense of spatial position; therefore, they perform badly in cluttered images, although they may be useful when textures are inherently repetitive (e.g., roof tiles or row crops in high-resolution aerial imagery). In some cases, it is desirable to sum the log variance outputs for matched horizontal and vertical masks (e.g., the two Sobel gradient masks) so as to decrease rotational sensitivity. One might also compute ratios of texture energy outputs to reduce sensitivity to contrast and illumination changes—although sensitivity to certain other texture characteristics may also be reduced. Such normalizations are still a black art.

With the possible exception of fractal measures, none of these operators are stable across different image scales; they are thus difficult to use in low-angle perspective imagery. They also tend to be too coarse, even for 3×3 texture energy masks in 5×5 variance windows, to measure the vegetation textures that are important for autonomous vehicle navigation and aerial cartography. Finally, all such texture operators respond very strongly to object borders, producing meaningless results for small or thin regions. Texture areas found with these discriminant features must therefore be "grown" out to their borders with other regions, whereupon small patches may be missed (or misclassified) entirely. These are among the reasons for my avoiding elaborate texture measures whenever possible.

It seems necessary to segment an image before really meaningful texture measures can be computed. Fortunately, this can usually be done by using pixel brightness and color alone. It is rare (in these applications) to find two adjacent regions that differ only in texture and not in local brightness or color. Segmentation can sometimes be improved even more by including local log variance (of the unfiltered image) as an additional descriptor band, although this measure does respond to region borders.

Unfortunately, prior segmentation is not the whole answer even when it is feasible. No one has yet developed a good way to compute texture energy measures over nonrectangular regions. (It would be necessary to estimate the probable contribution of missing data in the filter window under the assumption that it came from the same unknown texture as the available data.) Other texture measures, such as the less powerful co-occurrence or joint probability measures [Haralick 73, 79], are easier to compute over irregularly shaped regions even though they are less powerful and less plausible as models of human visual processing. An intriguing idea, which I have yet to develop, is to use similarities and differences of local histograms as textures measures. This is related to the classification technique described below (also in [Laws 85, 88b]) for extracting and identifying scene materials by using only single-band histograms.

4 Target Acquisition

The initial step in tracking an object or material type is to find it in the first image. In some applications the position and shape of the prototype are known or can be traced by the user; in others the prototype must be identified by means of spectral signatures or other knowledge. An intermediate method is to let the computer select likely prototypes and have the user indicate which are correct. I shall confine my discussion to autonomous target acquisition, although the KNIFE system does permit specification by tracing.

One approach is to segment the initial image, then classify, group, and further split regions until a target can be identified. My system partitions the image into homogeneous regions, absorbing small fragments while keeping larger regions. Large, distinct regions, such as sky and road, are found first, after which smaller details are extracted. (Identification becomes easier as background regions are labeled and removed from consideration. Spatial and spectral knowledge obtained from other scene objects could also aid recognition.)

This scene parsing approach is a major goal of current research on segmentation and scene-analysis. In limited domains, however, it may be very easy to find a particular target class. In road following, for instance, we may be able to assume that the lower half of the initial image is mostly road surface (possibly shadowed by the vehicle or other objects). All regions that are given the road or shadowed-road label can be merged, and any holes can be absorbed or passed to a reasoning system for analysis. Signatures extracted from these initial regions can be employed in a classification step to recognize additional parts of the road.

Another approach is to label pixels with stored signatures (or histograms), then extract connected regions that have been assigned the target label [Laws 88b]. While this is faster, it is often less reliable than a full parse. The approach works if the stored signatures are representative, which may require that they come from the same imagery or that they be adjusted for variations in illumination or target range. This is essentially the tracking approach; it will be described in more detail in the next section.

A third approach is to use likelihood maps to label initial regions. It may be that we know approximate or hypothesized object positions in the image and that we can estimate an a priori probability of any particular pixel position's belonging to the target. We can then segment the image and look for overlap between the image segments and the hypothesized label positions. This seems to work well even when the "likelihood map" is just a binary mask derived from a synthetic region map—i.e., when we are "tracking" from a hypothesized scene instead of from a previous image. This type of initial labeling is trivial if the synthetic objects overlap their true image regions by at least 50%; otherwise much more elaborate constraint-satisfaction techniques must be employed.

The KNIFE analysis system can be used for any of these approaches, since they all require similar techniques for region representation, threshold analysis, connected-component extraction, and noise cleaning. The acquisition method illustrated in this paper is to start with a reliable seed region and grow it to obtain a larger prototype region with an image-specific signature. The following section documents my method of tracking regions or material types once a partial scene parse is available. I assume that signatures are known or can be extracted for all major scene entities. Just "target" and "background" signatures may be sufficient, but better partitioning is often possible when

all visual classes are independently represented in the training set.

5 Classification Tracking

The tracking problem is to start with one instance of an object (or with an object signature stored in a database) and then find a matching instance in a new image. It may also be desirable to find additional instances of an object or material class in the new image, should they be present. We may even want to use characteristics of the newly identified instances to better extract the original objects—a joint-estimation problem.

The tracking task is complicated by the fact that regions in a second image may correspond poorly to those in the first. The original objects may have moved, of course, or their images may have been broken into smaller regions or left combined with neighboring objects. This is generally an insoluble problem, so I shall assume that the scene is sufficiently stable and the partitioning and labeling system sufficiently good that we can easily determine region correspondences. Relaxation labeling algorithms have been developed for establishing correspondences in more difficult cases [Price 76, 81, 82; Faugeras 81, 82; Kitchen 80, 84]; these are related to search algorithms for labeling region clusters as known objects [Duda 70; Freuder 72, 73, 76, 77; Barrow 76; Tennenbaum 76, 77; Khan 84ab; Wesley 83, 84, 86].

The road-tracking task is simplified because imaging conditions guarantee significant region overlap among images (almost always). In fact, most of the lower portion of the image can be assumed to be road if the vehicle is operating normally. Tracking trees and other roadside objects (to permit map generation, for example) is considerably more difficult. Some of this work must be left to higher-level reasoning processes, but the road tracking itself can almost always be done with low-level classification and partitioning techniques.

The principal output from a road tracker is the boundary of the navigable road in each image. This can be augmented by identifying the center stripe or divider, wheel ruts, potholes, shadows, material changes, and other minor markings. The boundary of distant road sections may be somewhat uncertain as long as the vehicle has time to gather and analyze additional imagery before making navigational decisions. The tracker or associated reasoning system should also detect intersections, forks, dead ends, and such major obstacles as vehicles, gates, shell holes, and missing bridges. It must also distinguish these obstructions from curves, dips, overhanging branches, telephone poles, traffic signs, and other objects that do not impede travel.

Image regions, even when perfectly extracted and labeled, do not correspond directly to such outputs. A telephone pole, for instance, may block our view of a curving road in such a way that the road is split into two pieces. The best that a low-level, task-independent vision system can do is to indicate the visible portions of the road, hypothesize a connection between them, and note the presence and characteristics of the occluding object. Establishing correspondence between the current regions and those in past images may also help to classify them.

A region graph shows how regions have split or joined to form other regions during image partitioning. A similar region graph, which I call a propagation graph, can be built to show how regions in one image have split or joined to form regions in the other.

Pixel classification is at the heart of material identification. Either pixels must be

classified and then grouped into regions or regions must be extracted and then classified. Since materials can exhibit an infinite variety of shapes, region classification is again done mainly by classifying the pixels and selecting a consensus label. When objects are tracked from image to image, such parameters as size, shape, position, and context can be used to facilitate region labeling.

Many target classes are composed of more than one material or surface appearance. The sides and roof of a house, for instance, may appear very different—and even the sides may differ because of detail and illumination. It is often convenient to treat such an object as a unit, rather than model the individual parts. This can reduce classificatory power if only point properties (e.g., brightness and color) are used, but the inclusion of texture measures may prevent a highly textured region from being mistaken for a configuration of homogeneous regions. Because KNIFE uses histograms as signatures, such composite objects can easily be represented. (Representation of mixed material classes by Gaussian parameters—mean, standard deviation, and perhaps higher moments—would be almost meaningless.)

6 Example

The KNIFE program has only recently been adapted to road tracking, and I have yet to develop an adequate control algorithm. The following example hints at the power of KNIFE's classification-based tracking, but it is not yet a fully developed application. Sophisticated monitoring processes will be needed to detect overmerging (or undermerging, loss of track, etc.) and to reacquire a target.

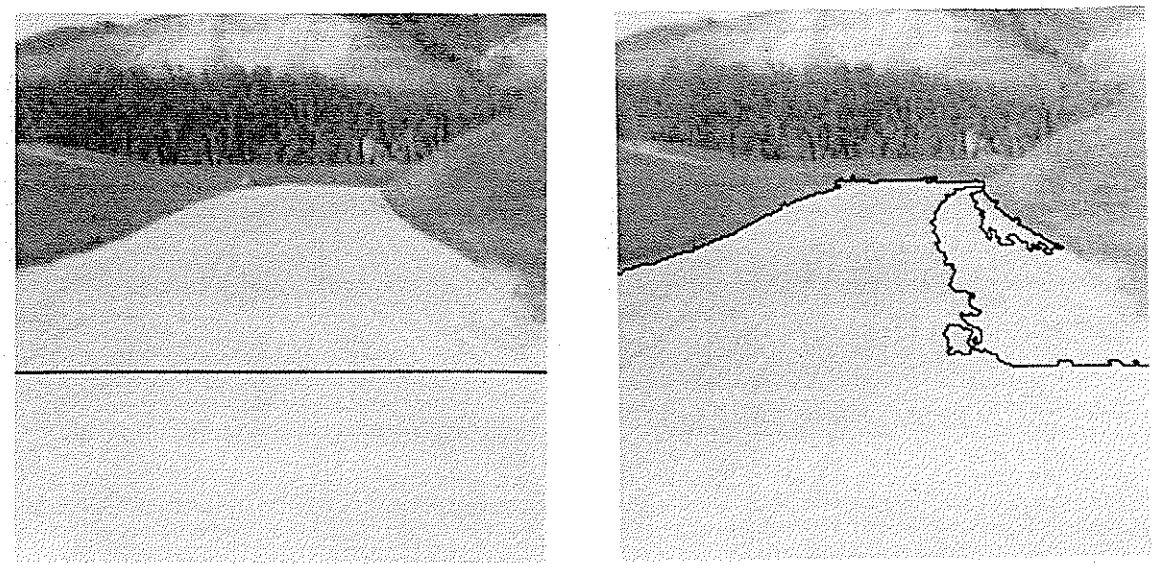
Figure 3(a) shows frame number 500 in a sequence of images taken from FMC's autonomous-vehicle prototype. Only the VHS image's vividness band is shown, although all three bands are used for tracking. The lower portion of the image has been selected (interactively and arbitrarily) as a training region for the road material type. Figure 3(b) shows the result of KNIFE's multiband region growing with seglevel set to 1 [Laws 88a]. The results are not perfect (because of KNIFE's linear model rather than its being curved or polynomial), of multidimensional surfaces, but are adequate and perhaps even typical of road regions extracted without semantic guidance.

I need to track this region into the next available image in the series, Frame 519. First I update my knowledge of road signatures (i.e., multiband histograms), but without forgetting the initial training signature. KNIFE lets me do this by storing both the original and new road signatures as two exemplars of road appearance. I also store the nonroad, or "other" signature so that I can assign pixels in the next image to the more similar of the two semantic categories.

Figure 4(a) shows the result of this classification-based tracking. The central portion of the road is labeled "other" and the mountain face is labeled "road," but, on the whole, the tracking has been successful. The extracted regions are suitable for processing by a higher-level reasoning system.

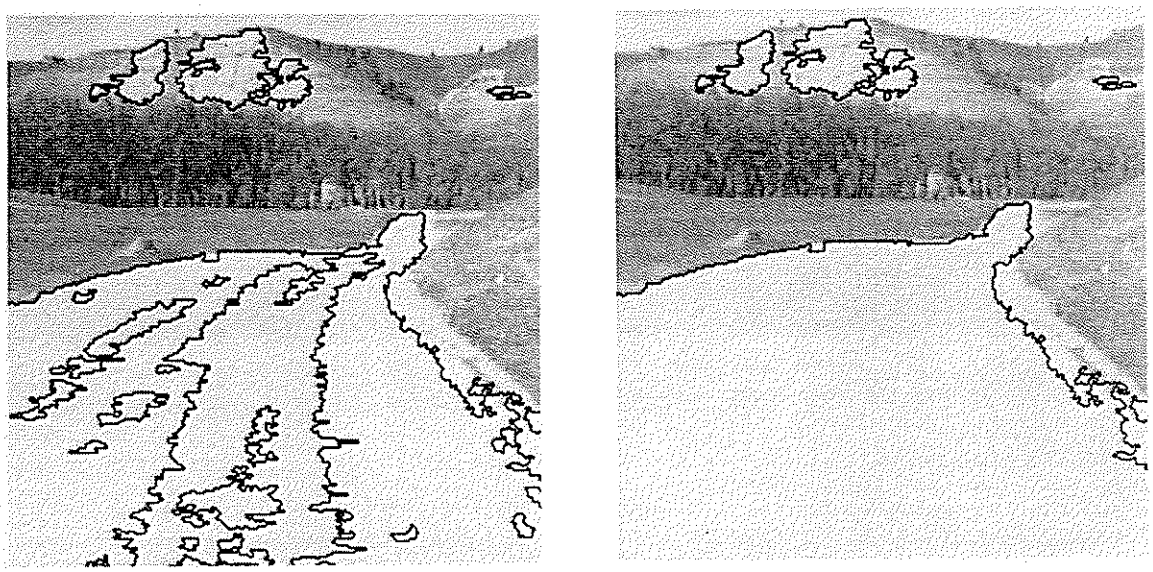
Since KNIFE is domain-independent, I have provided the higher-level control by issuing a region-growing command for the largest road region. This yielded the clean extraction in Figure 4(b), although there are certainly scenes for which simple region growing would be inadequate.

Next, without worrying about the misclassified mountain face, I propagated the new



(a) Training Region (V Band) (b) Grown Region

Figure 3: Track Initiation in the ALV500 Image



(a) Two-Category Classification (b) Grown Road Region

Figure 4: Tracking Results for the ALV519 Image

"road" and "other" signatures (along with the ones gathered previously) into Frame 525. The result is shown in Figure 5(a). The central portion of the road has been misclassified here too, but the bulk is correct. I ordered region growing on the largest road region, obtaining the results depicted in Figure 5(b). It is not obvious whether the program is correct or incorrect in assigning the two distant, dark patches to the road class.

Figure 6 illustrates the same process applied to Frame 533 in the sequence. This time the jump is too large or the two-class model too impoverished. KNIFE does a good job of finding unshadowed road, but lacks knowledge of shadowed road and so assigns it the "other" label. Attempts at growing either of the large road regions result in overmerging and loss of track, as shown in Figure 6(b).

I confess that I had less success with this simple classify/grow cycle on a different sequence of images. I had to use a four-class model—road, sky, trees, and grass—and even then lost track when the road touched a very similar portion of sky. Because of tracking in natural imagery is difficult, knowledge-based control techniques are essential. On the whole, though, KNIFE's histogram-based classification tracking works well and is fast enough to be considered for real-time implementation.

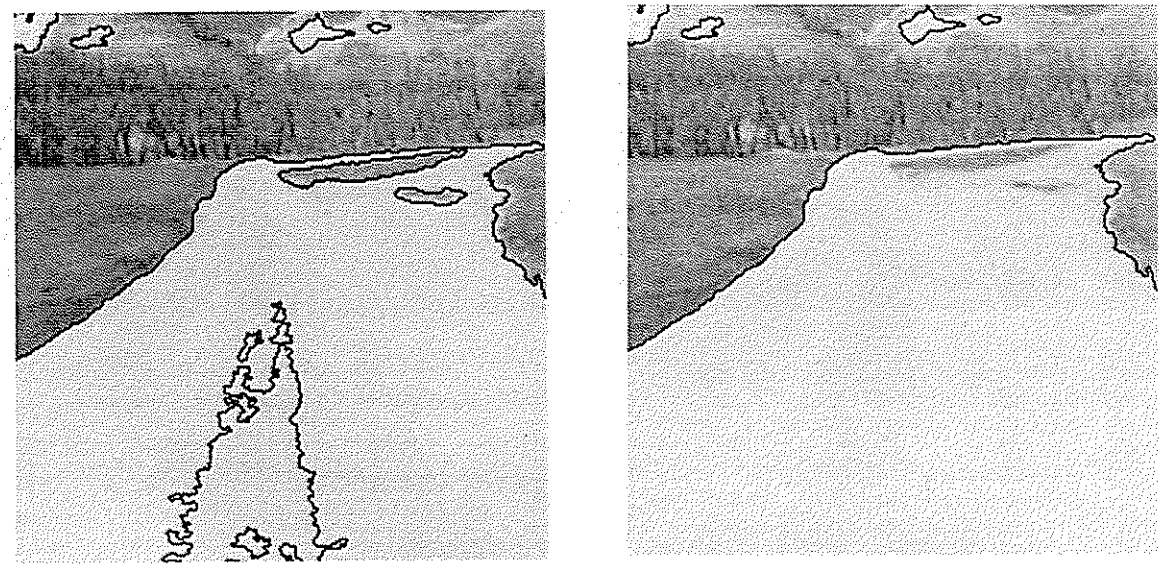
7 Summary

The KNIFE system is a testbed for partitioning and labeling techniques. Just as a mail program permits editing and manipulation of messages, KNIFE permits editing and manipulation of image regions. It can be viewed as a stand-alone program, as part of a larger system, or as an environment in which to build other capabilities.

This paper has described the use of pixel and region classification to track objects or material types from one image to another. Minimal use was made of task-specific or target-specific knowledge so as to facilitate modular construction of more complex systems. Desired targets are described declaratively by their multiband signatures (or histograms), rather than procedurally, thereby simplifying database and interface requirements. These signatures can be extracted automatically from known instances in previous imagery.

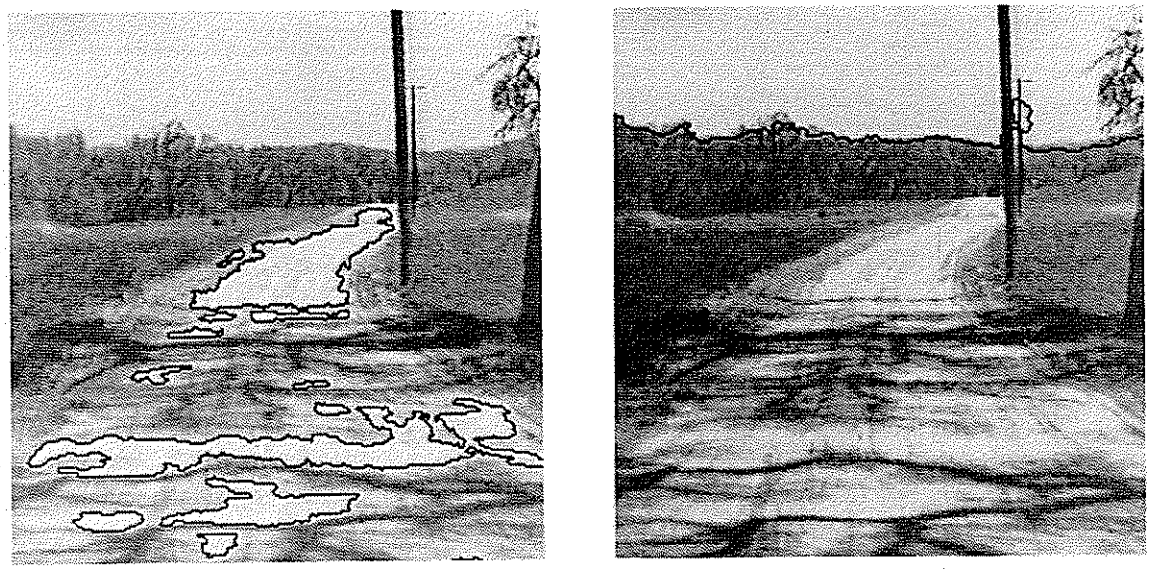
A large portion of this paper was devoted to the subject of color and texture measurement. Although no ultimate solution is available for characterizing such pixel properties, I have specified particular techniques that work well and are easy to compute. These provide a baseline for further advances in texture classification and segmentation.

Once signatures are available, histogram-based techniques can be used to label pixels or regions in new images (or new parts of an initial image). Multiband histograms for these regions can be used to update the stored material-class signatures so that specific objects can be tracked into future images. This approach is fairly invulnerable to occlusions and slow changes in object appearance.



(a) Two-Category Classification (b) Grown Road Region

Figure 5: Tracking Results for the ALV525 Image



(a) Two-Category Classification (b) Grown Road Region

Figure 6: Tracking Results for the ALV533 Image

Acknowledgment

This work was supported by Defense Advanced Research Projects Agency under Contract Nos. MDA903-86-C-0084 and DACA76-85-C-0004.

References

[Ballard 81] D.H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes," *Pattern Recognition,* Vol. 13, No. 2, pp. 111–122 (1981).

[Barrow 76] H.G. Barrow and J.M. Tenenbaum, *MSYS: A System for Reasoning about Scenes,* Technical Note 121, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California (April 1976).

[Bolles 81] R.C. Bolles, "A RANSAC-Based Approach to Model Fitting and its Application to Finding Cylinders in Range Data," *Proc. 7th Int. Jnt. Conf. on Artificial Intelligence,* Vancouver, British Columbia, Canada, pp. 637–643 (24–28 August 1981). Also published as Technical Note 235, Artificial Intelligence Center, SRI International, Menlo Park, California (March 1981).

[Danielsson 80] P.-E. Danielsson, "Rotation-Invariant Linear Operators with Directional Response," *Proc. 5th Int. Jnt. Conf. on Pattern Recognition,* Miami Beach, Florida, pp. 1171–1176 (1–4 December 1980).

[Danielsson 85] P.-E. Danielsson and H. Sauleda, "Rotation-Invariant 2-D Filters Matched to 1-D Features," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* San Francisco, California pp. 155–160 (19–23 June 1985).

[Daugman 84] J.G. Daugman, "Spatial Visual Channels in the Fourier Plane," *Vision Research,* Vol. 24, No. 9, pp. 891–910 (1984).

[Daugman 85] J.G. Daugman, "Uncertainty Relation for Resolution in Space, Spatial Frequency, and Orientation Optimized by Two-Dimensional Visual Cortical Filters," *Journal of the Optical Society of America,* Vol. 2, No. 7, pp. 1160–1169 (July 1985).

[Duda 70] R.O. Duda, *Some Current Techniques for Scene Analysis,* Technical Note 46, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California (October 1970).

[Dyer 76] C.R. Dyer and A. Rosenfeld, "Fourier Texture Features: Suppression of Aperture Effects," *IEEE Trans. on Systems, Man and Cybernetics,* Vol. SMC–6, No. 10, pp. 703–705 (October 1976). Also published as TR–391, Computer Vision Laboratory, Computer Science Center, University of Maryland, College Park, Maryland (July 1975).

[Faugeras 81] O.D. Faugeras and K. Price, "Semantic Description of Aerial Images Using Stochastic Labeling," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* Vol. PAMI–3, No. 6, pp. 633–642 (November 1981). Also in *Proc. DARPA Image Understanding Workshop,* College Park, Maryland, pp. 89–94 (30 April 1980), and in *Proc. 5th Int. Jnt. Conf. on Pattern Recognition,* Miami Beach, Florida, pp. 352–357 (December 1980).

[Faugeras 82] O.D. Faugeras, "Relaxation Labeling and Evidence Gathering," *Proc. IEEE Conf. on Pattern Recognition and Image Processing,* Las Vegas, Nevada, pp. 672–677 (14–17 June 1982).

[Fischler 81] M.A. Fischler and R.C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Communications of the ACM,* Vol. 24, No. 6, pp. 381–395 (June 1981). Also published in *Proc. DARPA Image Understanding Workshop,* College Park, Maryland, pp. 71–88 (30 April 1980), as Technical Note 213, Artificial Intelligence Center, SRI International, Menlo Park, California (March 1980).

[Fischler 84] M.A. Fischler, *Computer Vision Research and its Applications to Automated Cartography,* Second and Third Semiannual Technical Reports, SRI Project 5355, Artificial Intelligence Center, SRI International, Menlo Park, California (September 1984).

[Fischler 88] M.A. Fischler and T.M. Strat, "Recognizing Trees, Bushes, Rocks, and Rivers," *Proc. 1988 Symp. on Physical and Biological Approaches to Computational Vision,* Stanford, California, pp. 62–64 (22–24 March 1988).

[Freuder 72] E.C. Freuder, *Recognition of Real Objects,* Working Paper 33, MIT AI Laboratory, Cambridge, Massachusetts (October 1972).

[Freuder 73] E.C. Freuder, *Active Knowledge,* Working Paper 53, MIT AI Laboratory, Cambridge, Massachusetts (October 1973).

[Freuder 76] E.C. Freuder, *A Computer System for Visual Recognition Using Active Knowledge,* Ph.D. Thesis, AI–TR–345, Massachusetts Institute of Technology, MIT AI Laboratory, Cambridge, Massachusetts (June 1976).

[Freuder 77] E.C. Freuder, "A Computer System for Visual Recognition Using Active Knowledge," *Proc. 5th Int. Jnt. Conf. on Artificial Intelligence,* Cambridge, Massachusetts, pp. 671–677 (1977). Reprinted in Y.-H. Pao and G.W. Ernst (eds.), *Context-Directed Pattern Recognition and Machine Intelligence Techniques for Information Processing,* IEEE Computer Society Press, Silver Spring, Maryland, pp. 511–517 (1982).

[Fua 85a] P. Fua and A.J. Hanson, "Object Labeling Using Generic Knowledge," in M.A. Fischler (ed.), *Computer Vision Research and its Application to Automated Cartography,* Fourth and Fifth Semiannual Technical Reports, SRI Project 5355, Artificial Intelligence Center, SRI International, Menlo Park, California Appendix G, (September 1985).

[Fua 85b] P. Fua and A.J. Hanson, "Locating Cultural Regions in Aerial Imagery Using Geometric Cues," *Proc. DARPA Image Understanding Workshop,* Miami Beach, Florida, pp. 271–278 (9–10 December 1985).

[Fua 86] P. Fua and A.J. Hanson, *Using Generic Geometric Knowledge to Delineate Cultural Objects in Aerial Imagery,* Technical Note 378, Artificial Intelligence Center, SRI International, Menlo Park, California (March 1986).

[Fua 87a] P. Fua and A.J. Hanson, "Resegmentation Using Generic Shape: Locating General Cultural Objects," *Pattern Recognition Letters,* Vol. 5, pp. 243–252 (1987).

[Fua 87b] P. Fua and A.J. Hanson, "Using Generic Geometric Models for Intelligent Shape Extraction," *Proc. DARPA Image Understanding Workshop,* Los Angeles, California, pp. 227–233 (23–25 February 1987). Also in *Proc. 6th National Conf. on Artificial Intelligence,* Seattle, Washington, pp. 706–711 (13–17 July 1987).

[Hannah 84] M.J. Hannah, *Description of SRI's Baseline Stereo System,* Technical Note 342, Artificial Intelligence Center, SRI International, Menlo Park, California (October 1984).

[Hannah 85] M.J. Hannah, "SRI's Baseline Stereo System," *Proc. DARPA Image Understanding Workshop,* Miami Beach, Florida, pp. 149–155 (9–10 December 1985).

[Hanson 83] A.J. Hanson, *Overview of the Image Understanding Testbed,* Technical Note 311, Artificial Intelligence Center, SRI International, Menlo Park, California (October 1983).

[Haralick 73] R.M. Haralick, K. Shanmugam, and I. Dinstein, "Textural Features for Image Classification," *IEEE Trans. on Systems, Man and Cybernetics,* Vol. SMC–3, No. 6, pp. 610–621 (November 1973).

[Haralick 79] R.M. Haralick, "Statistical and Structural Approaches to Texture," *Proc. IEEE,* Vol. 67, No. 5, pp. 786–804 (May 1979). Also published in *Proc. Int. Jnt. Conf. on Pattern Recognition,* pp. 45–60 (1978).

[Harwood 83] D. Harwood, M. Subbarao, and L.S. Davis, *Texture Classification by Local Rank Correlation,* TR–1314, Computer Vision Laboratory, Center for Automation Research, University of Maryland, College Park, Maryland (August 1983).

[Hashimoto 83] M. Hashimoto and J. Sklansky, "Edge Detection by Estimation of Multiple-Order Derivatives," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Washington, D.C., pp. 318–325 (19–23 June 1983).

[Kender 76] J.R. Kender, *Saturation, Hue, and Normalized Color: Calculation, Digitization Effects, and Use*, Dept. of Computer Science, Carnegie-Mellon University, Pittsburgh, Pennsylvania (November 1976).

[Kender 77] J.R. Kender, "Instabilities in Color Transformations," *Proc. IEEE Conf. on Pattern Recognition and Image Processing*, Troy, New York, pp. 266–274 (6–8 June 1977).

[Khan 84a] N.A. Khan, *Distributed Problem Solving for Object Recognition*, Ph.D. Thesis, Dept. of Computer Science, Wayne State University, Detroit, Michigan (January 1984).

[Khan 84b] N.A. Khan and R. Jain, "Matching an Imprecise Object Description with Models in a Knowledge Base," *Proc. 7th Int. Jnt. Conf. on Pattern Recognition*, Montreal, Canada, pp. 1131–1134 (30 July – 2 August 1984).

[Kitchen 80] L.J. Kitchen, "Relaxation Applied to Matching Quantitative Relational Structures," *IEEE Trans. on Systems, Man and Cybernetics*, Vol. SMC–10, pp. 96–101 (1980).

[Kitchen 84] L.J. Kitchen and A. Rosenfeld, "Scene Analysis Using Region-Based Constraint Filtering," *Pattern Recognition*, Vol. 17, No. 2, pp. 189–203 (1984).

[Knutsson 83] H. Knutsson and G.H. Granlund, "Texture Analysis Using Two-Dimensional Quadrature Filters," *Proc. IEEE Workshop on Computer Architecture for Pattern Analysis and Image Database Management*, Pasadena, California, pp. 206–213 (12–14 October 1983).

[Larkin 83] L.I. Larkin and P.J. Burt, "Multi-Resolution Texture Energy Measures," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Washington, D.C., pp. 519–520 (19–23 June 1983).

[Laws 79] K.I. Laws, "Texture Energy Measures," *Proc. DARPA Image Understanding Workshop*, Los Angeles, California, pp. 47–51 (7–8 November 1979).

[Laws 80a] K.I. Laws, *Textured Image Segmentation*, Ph.D. Thesis, Report USCIPI 940, Image Processing Institute, University of Southern California, Los Angeles, California (January 1980).

[Laws 80b] K.I. Laws, "Rapid Texture Identification," *Proc. SPIE Conf. on Image Processing for Missile Guidance*, San Diego, California, Vol. 238, pp. 376–380 (29 July – 1 August 1980).

[Laws 82] K.I. Laws, *The GHOUGH Generalized Hough Transform Package: Description and Evaluation*, Technical Note 288, Artificial Intelligence Center, SRI International, Menlo Park, California (December 1982).

[Laws 84] K.I. Laws (ed.), *The DARPA/DMA Image Understanding Testbed Programmer's Manual*, Technical Note 298, Artificial Intelligence Center, SRI International, Menlo Park, California (January 1984).

[Laws 85] K.I. Laws, "Goal-Directed Textured-Image Segmentation," *Proc. SPIE Conf. on Applications of Artificial Intelligence II*, Vol. 548, Arlington, Virginia, (9–11 April 1985). Summarizes Technical Note 334, Artificial Intelligence Center, SRI International, Menlo Park, California (September 1984).

[Laws 88a] K.I. Laws, *Integrated Split/Merge Image Segmentation*, Technical Note 441, Artificial Intelligence Center, SRI International, Menlo Park, California (July 1988).

[Laws 88b] K.I. Laws, *Coarse Coding for Material and Object Identification*, Technical Note 442, Artificial Intelligence Center, SRI International, Menlo Park, California (July 1988).

[Moravec 80] H.P. Moravec, *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover*, Ph.D. Thesis, Report AIM-304a, Artificial Intelligence Laboratory, Stanford University, Stanford, California (1980).

[Nicholson 80] W.Q. Nicholson and K.S. Davis, "The Binomial Window," *Proc. SPIE Conf. on Image Processing for Missile Guidance*, San Diego, California, Vol. 238, pp. 467–479 (29 July – 1 August 1980).

[Ohlander 75] R.B. Ohlander, *Analysis of Natural Scenes*, PhD. Thesis, Dept. of Computer Science, Carnegie-Mellon University, Pittsburg, Pennsylvania (April 1975).

[Ohlander 78] R. Ohlander, K. Price, and D.R. Reddy, "Picture Segmentation Using a Recursive Region Splitting Method," *Computer Graphics and Image Processing*, Vol. 8, No. 3, pp. 313–333 (December 1978).

[Ohta 80] Y. Ohta, T. Kanade, and T. Sakai, "Color Information for Region Segmentation," *Computer Graphics and Image Processing*, Vol. 13, pp. 222–241 (1980).

[Pentland 84] A.P. Pentland, "Fractal-Based Description of Natural Scenes," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-6, No. 6, pp. 661–674 (November 1984), reprinted in R. Chellappa and A..A. Sawchuk (eds.), *Digital Image Processing and Analysis: Volume 2: Digital Image Analysis*, pp. 368–381 (1985), and in A.P. Pentland (ed.), *From Pixels to Predicates: Recent Advances in Computational and Robotic Vision*, Ablex, Norwood, New Jersey pp. 227–252 (1985). Also published in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Washington, D.C., pp. 201–209 (19–23 June 1983), in *Proc. DARPA Image Understanding Workshop*, Arlington, Virginia, pp. 184–192 (23 June 1983).

[Pietikäinen 82] M. Pietikäinen, A. Rosenfeld, and L.S. Davis, "Texture Classification Using Averages of Local Pattern Matches," *Proc. 6th Int. Jnt. Conf. on Pattern Recognition*, Munich, Germany, pp. 301–303 (19–22 October 1982).

[Pietikäinen 83] M. Pietikäinen, A. Rosenfeld, and L.S. Davis, "Experiments with Texture Classification Using Averages of Local Pattern Matches," *IEEE Trans. on Systems, Man and Cybernetics*, Vol. SMC–13, No. 3, pp. 421–426 (May/June 1983).

[Price 76] K.E. Price, *Change Detection and Analysis in Multi-Spectral Images*, Ph.D. Thesis, Dept. of Computer Science, Carnegie-Mellon University, Pittsburgh, Pennsylvania (December 1976).

[Price 81] K.E. Price, "Relaxation Matching applied to Aerial Images," *Proc. DARPA Image Understanding Workshop*, Washington, D.C., pp. 22–25 (23 April 1981).

[Price 82] K.E. Price, "Symbolic Matching of Images and Scene Models," *Proc. Workshop on Computer Vision: Representation and Control*, Rindge, New Hampshire, pp. 105–112 (23–25 August 1982). Also published in *Proc. DARPA Image Understanding Workshop*, Palo Alto, California, pp. 299–308 (15–16 September 1982).

[Sloan 80] K.R. Sloan and D.H. Ballard, "Experience with the Generalized Hough Transform," *Image Understanding Workshop*, College Park, Maryland, pp. 150–156 (30 April 1980).

[Strat 88] T.M. Strat and G.B. Smith, "Core Knowledge System: Storage and Retrieval of Inconsistent Information," *Image Understanding Workshop*, Cambridge, Massachusetts, pp. 660–665 (6–8 April 1988).

[Tanimoto 78] S.L. Tanimoto, "An Optimal Algorithm for Computing Fourier Texture Descriptors," *IEEE Trans. on Computers*, Vol. C–27, No. 1, pp. 81–84 (January 1978).

[Tenenbaum 74] J.M. Tenenbaum, T.D. Garvey, S.A. Weyl, and H.C. Wolf, "An Interactive Facility for Scene Analysis," Technical Note 87, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California (January 1974).

[Tenenbaum 76] J.M. Tenenbaum and H.G. Barrow, "IGS: A Paradigm for Integrating Image Segmentation and Interpretation," *Proc. 3rd Int. Jnt. Conf. on Pattern Recognition,* Coronado, California, pp. 504–513 (8–11 November 1976).

[Tenenbaum 77] J.M. Tenenbaum and H.G. Barrow, "Experiments in Interpretation Guided Segmentation," *Artificial Intelligence,* Vol. 8, No. 3, pp. 241–274 (June 1977). Also published as Technical Note 123, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California (March 1976).

[Wesley 83] L. Wesley, "Reasoning About Control: The Investigation of an Evidential Approach," *Proc. 8th Int. Jnt. Conf. on Artificial Intelligence,* Karlsruhe, West Germany, pp. 203–210 (8–12 August 1983).

[Wesley 84] L.P. Wesley, J.D. Lowrance, and T.D. Garvey, *Reasoning About Control: An Evidential Approach,* Technical Note 324, Artificial Intelligence Center, SRI International, Menlo Park, California (July 1984).

[Wesley 86] L.P. Wesley, "Evidential Knowledge-Based Computer Vision," *Optical Engineering,* Vol. 25, No. 3, pp. 363–379 (March 1986). Also published as Technical Note 374, Artificial Intelligence Center, SRI International, Menlo Park, California (January 1986).

[Weszka 76] J.S. Weszka, C.R. Dyer, and A. Rosenfeld, "A Comparative Study of Texture Measures for Terrain Classification," *IEEE Trans. on Systems, Man and Cybernetics,* Vol. SMC-6, No. 4, pp. 269–285 (April 1976).