FINDING CORNERS

J. Alison Noble *
Robotics Research Group
Department of Engineering Science
Oxford University
England

Abstract

Many important image cues such as 'T'-,'X'- and 'L'- junctions have a local two-dimensional structure. Conventional edge detectors are designed for one-dimensional 'events'. Even the best edge operators can not reliably detect these two-dimensional features. This contribution proposes a solution to the two-dimensional problem.
In this paper, I address the following:

- 'L'-junction detection. Previous attempts, relying on the second differentials of the image surface have essentially measured image curvature. Recently Harris [Harris 87] implemented a 'corner' detector that is based only on first differentials. I provide a mathematical proof to explain how this algorithm estimates image curvature.
  Although this algorithm will isolate image 'L'-junctions, its performance cannot be predicted for 'T'-junctions and other higher order image structures.
- Instead, an image representation is proposed that exploits the richness of the local differential geometrical 'topography ' of the intensity surface. Theoretical and experimental results are presented which demonstrate how idealised instances of two-dimensional surface features such as junctions can be characterised by the differential geometry of a simple facet model.
- Preliminary results are very encouraging. Current studies are concerned with the extension to real data. I am investigating statistical noise models to provide a measure of 'confidence' in the geometric labelling. The richness and sparseness of a two-dimensional structure can be exploited in many high-level vision processes. I intend to use my representation to explore some of these fields in future work.

Keywords: Image Structure, edge detection, corner detection.

1 Introduction

The purpose of low-level vision is to extract useful information from images. To date, differential edge detectors have proven the best tools for this purpose. However, even the most successful, such as the Canny operator, fail to reliably detect corners and intersections of edge contours (see for example [Canny 83 Fig.6.14b]). Yet it is these image features which have highest information content.

The inadequacy of differential detectors stems from the implicit assumption that all edges are essentially 1-D. That is, the image function depends primarily on only one of the two spatial co-ordinates defining the intensity matrix. However, the projections of important image cues such as surface corners and intersections give rise to 2-D surface structure.

The simplest example of 2-D image structure is provided by the 'L'- junction or gray-level corner, which corresponds to a corner of a polyhedral surface in the real-world. This is the gray-level structure isolated by the Plessey algorithm described in section 2. Another important intensity structure is the 'T'-junction, typically arising where three polyhedra surfaces meet. Whereas it is possible to write down a mathematical definition for an 'L'-junction [Nagel 83], a multitude of parameters are required for a 'T'-junction. 'T'-junctions are relatively simple 2-D structures. Until now, modelling and subsequent use of such gray-level structure has proven beyond the capabilities of machine vision.

Exploiting the high information content implicit within 2-D image structure, defines a richer and sparser image representation than provided by a scheme based on conventional edge points. Such structure has been proposed as the 'interesting' points in the frame-to-frame correspondence, [Barnard-Thompson 80,Ullman 79] structure from motion [Harris 87,Spacek 84], and stereo [Moravec 77,Spacek 84]. Nagel [Nagel 83] has shown how 'gray-level corners' can provide additional constraints in the complete determination of optical flow. Further, points of 'significant curvature' have been successfully used to define a compact representation for 2-D shape description [Asada-Brady 86, Mcdioni-Yasumuto 86,Mokhtarian-Mackworth 86]. However, defining a suitable representation for 3-D shape description remains unsolved. A 2-D image representation provides a means of achieving this goal.

Gray-level corner detection has received some attention in the past [Dreschler-Nagel 81,Zuniga-Haralick 83, Spacek 84, Kitchen-Rosenfeld 82]. A detailed review and discussion of these can be found in [Noble 87]. Essentially, all have used a measure of 'cornerness' C, defined as the product of gradient magnitude (a measure of 'edgeness') and the rate of change of gradient direction with gradient magnitude (a measure of 'cornerness'). That is, declare a corner if the cornerness is above threshold *and* the pixel is an edge point.

*The author acknowledges the support of the Science and Engineering Research Council

AVC 1987 doi:10.5244/C.1.37

$$\text{i.e.} \quad C = (I_{xx}I_y^2 + I_{yy}I_x^2 - 2I_{xy}I_xI_y)/(I_x^2 + I_y^2)^{3/2}$$
$$= 1/|\nabla I|(\kappa_{n\perp})$$

where $\kappa_{n\perp}$ is the curvature in the direction perpendicular to the gradient. Clearly, this measure depends on second differentials of the image function $I(x,y)$. As such, even with noise suppression heuristics, the probability of false corner detection can be expected to be high.

Recently, Harris [Harris 87], implemented a corner detector which is novel in so far as the computations are based entirely on first differential quantities. A description of how and why it works is presented in subsequent sections. However, 'L'-junctions are only a special type of 2-D image structure. The Plessey algorithm is incapable of consistently recognising other 2-D features.

With the ultimate goal of defining a 2-D image representation, I have been investigating some of the differential geometric properties of the intensity image structure. I have shown how the differential geometry of a simple facet model can characterise idealised instances of features such as intensity junctions and corners. Recently, Fleck [Fleck 87] has implemented an edgefinder that performs remarkably well. The analysis given here and in [Noble 87] provides that program with a theoretical underpinning.

2 The Plessey Corner Finder

I first outline the principles underlying the Plessey corner detector.
The algorithm can be divided into the following stages:

Assume a window size $W = (n \times n)$. For each pixel (i,j) in the image,

1. Find I_x and I_y using $(n \times n)$ first difference approximations to the partial derivatives.
2. Calculate the three quantities $I_x^2, I_y^2,$ and I_xI_y.
3. Using a Gaussian smoothing kernel of standard deviation σ, compute the sampled means $\langle I_x^2\rangle, \langle I_y^2\rangle, \langle I_xI_y\rangle$ using the $(n \times n)$ neighbouring point samples found in (2).
4. Evaluate the eigenvalues μ_1, μ_2.

$$\mathbf{A} = \begin{bmatrix} \langle I_x^2\rangle & \langle I_xI_y\rangle \\ \langle I_xI_y\rangle & \langle I_y^2\rangle \end{bmatrix}$$

If both are 'large', declare a corner.

In the implementation, the 'cornerness' $C_{plessey}$ is calculated as the ratio

$$C_{plessey} = Trace\mathbf{A}/Det\mathbf{A}$$
$$= (\langle I_x^2\rangle + \langle I_y^2\rangle)/(\langle I_x^2\rangle\langle I_y^2\rangle - \langle I_xI_y\rangle^2)$$
$$= (\mu_1 + \mu_2)/\mu_1\mu_2 \qquad (1)$$

and a corner marked *iff* $C_{plessey}$ is small. The crucial part in the computations proves to be with the method and assumptions used to evaluate the matrix determinant

$$\langle I_x^2\rangle\langle I_y^2\rangle - \langle I_xI_y\rangle^2$$

3 Image Surface Interpretation

Having described the Plessey algorithm, I now present a theoretical explanation of why the corner finder works on real 'L'-junctions. The analysis to follow is developed using some of the results from differential geometry. The relevant definitions are introduced as they appear in the analysis. For more detailed treatment the reader is referred to textbooks such as [Lipschutz 69,Faux-Pratt 79].

3.1 One Point Sample Means

Equation 1 included the term $(\langle I_x^2\rangle\langle I_y^2\rangle - \langle I_xI_y\rangle^2)$. This section considers the case where this term is zero, which arises when there is perfect imagery so that a point sampled mean $\langle I_x^2\rangle$ is identified with I_x^2 etc.

The image surface $S(x,y)$ is described by the equation

$$\mathbf{S(x,y)} = x\mathbf{i} + y\mathbf{j} + I(x,y)\mathbf{k}$$

where $I(x,y)$ is a gray-level value. Taken together, the gray-levels define a set of discrete samples of a continuous bivariate function, the *image function I(x,y)*. In the discussions that follow $I(x,y)$ is assumed smooth and continuous.
Using appropriate partial derivatives, the surface unit normal $\hat{\mathbf{N}}$ can be shown to be

$$\hat{\mathbf{N}} = \frac{1}{(1 + I_x^2 + I_y^2)^{1/2}}\{-I_x, -I_y, 1\}^T$$

and the orthogonal projection of this vector onto the plane $z = 0$, given by

$$\hat{\mathbf{n}} = \frac{1}{(I_x^2 + I_y^2)^{1/2}}\{-I_x, -I_y, 0\}^T$$

One convenient way to look at the properties of a surface is to consider the First Fundamental Form Φ_1,defined as

$$\Phi_1 = d\mathbf{S}.d\mathbf{S}$$
$$= [dxdy]\begin{bmatrix} E & F \\ F & G \end{bmatrix}\begin{bmatrix} dx \\ dy \end{bmatrix}$$

where the coefficients $E = \mathbf{S}_x.\mathbf{S}_x, F = \mathbf{S}_x.\mathbf{S}_y$, and $G = \mathbf{S}_y.\mathbf{S}_y$, are called the metric coefficients, and their defining matrix $\mathbf{G}$ the *metric* or metric tensor of the surface. The First Fundamental Form provides a measure of the change in the surface representation $|d\mathbf{S}|^2$, for a small change in (dx,dy). For an image surface, $\mathbf{G}$ reduces to

$$\mathbf{G} = \begin{bmatrix} 1 + I_x^2 & I_xI_y \\ I_xI_y & 1 + I_y^2 \end{bmatrix}$$

The eigenvalues of $\mathbf{G}$ are defined by the relation

$$\begin{bmatrix} 1 + I_x^2 - \lambda & I_xI_y \\ I_xI_y & 1 + I_y^2 - \lambda \end{bmatrix} = 0$$

Putting $\lambda - 1 = \mu$, shows the eigenvalues to satisfy the quadratic

$$\mu^2 - (I_x^2 + I_y^2)\mu + I_x^2I_y^2 - (I_xI_y)^2 = 0$$

Further, since $I_x^2I_y^2 - (I_xI_y)^2 = 0$,

$$\mu^2 - (I_x^2 + I_y^2)\mu = 0$$

Hence the eigen-structure of the image surface is described by the two eigenvalues λ_i and associated eigenvectors $\mathbf{u}_i$ where

$\mu_1 = 0$ $\qquad$ $\mathbf{u}_1 = \{-I_y, I_x\}^T$ i.e. one eigenvector lies along the surface orthogonal to the gradient

$\mu_2 = \| \nabla I \|^2 \quad \mathbf{u}_1 = \{I_x, I_y\}^T$ i.e. the other lies along the image surface gradient

This is as expected from intuition. If $\mathbf{W}$ and $\mathbf{V}$ are the eigenvector and dual eigenvector matrices then the eigenmatrix defined by $\mathbf{\Lambda} = \mathbf{VGW}$ is

$$\mathbf{\Lambda} = \begin{bmatrix} \| \nabla I \|^2 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{where,} \quad \mathbf{W} = \| \nabla I \| \, \mathbf{V} = \begin{bmatrix} -I_y & I_x \\ I_x & I_y \end{bmatrix}$$

At an edge, $\| \nabla I \|^2$ is large. (This is not strictly true for all edges as this requirement will ignore weak but well localised edges.) Further, no assumptions have been made on the image structure (except assumed functional continuity), so the results hold true everywhere on the surface, including corners. To demonstrate that the eigenvalues can be used to isolate only corners, we need to show two things;

1. Firstly, equation 1 is approximately zero along an edge or in a low gradient region. If this is true, then the analysis above shows that we get a single high eigenvalue, lying along the intensity gradient. The Plessey algorithm rejects such points (step 4 of their algorithm).

2. Secondly, equation 1 is high at a corner, or at least not negligible. In this case, the analysis above does not apply and both eigenvalues are large.

Consider the first problem:

1. Case 1: Low (or zero) gradient region i.e. a homogeneous patch, well isolated from other edges and image features.

   In this situation, I_x and I_y will be low throughout the window W.
   Consequently $\langle I_x^2 \rangle$, $\langle I_y^2 \rangle$, and $\langle I_x I_y \rangle^2$, are all nearly zero and both eigenvalues are small.

2. Case 2: A step edge, without loss of generality, can be aligned with the $y-$axis.

   For a small enough window kernel, $I_y \simeq 0$, and everywhere along the edge I_x is large. Then, although $\langle I_x^2 \rangle$ is large, $\langle I_y^2 \rangle$ and $\langle I_x I_y \rangle^2$ are both approximately zero. Hence $\langle I_x^2 \rangle \langle I_y^2 \rangle - \langle I_x I_y \rangle^2 = 0$ as required.

We only need to prove that equation 1 is high at a corner. To do this we need to consider the more general case of sampled means.

3.2 The General Case: Sampled Means

Remember that the critical term in the Plessey algorithm is $\langle I_x^2 \rangle \langle I_y^2 \rangle - \langle I_x I_y \rangle^2$. In the previous section where point sample means were assumed, this factor is zero. In general, when using real discrete data, there is no guarantee that this condition is true. Each quantity $\langle I_x^2 \rangle$ (say), is now dependent on the local variations of I_x^2 within the pixel neighbourhood. Therefore $\langle I_x^2 \rangle$ represents a weighted average value of these neighbouring values. In the following analysis, a Gaussian weight matrix $\sigma(\mathbf{x})$, is used for this purpose, where its region of definition is over a window $\mathbf{x} \in W$.
Then

$$\langle I_x^2 \rangle = \frac{\int_W \sigma(\mathbf{x}) I_x^2(\mathbf{x}) d\mathbf{x}}{\int_W \sigma(\mathbf{x}) d\mathbf{x}}$$

The normalising weight matrix will occur several times in the subsequent analysis. Let it be denoted by

$$\int_W \sigma(\mathbf{x}) d\mathbf{x} \equiv \Psi$$

Then

$$\langle I_x^2 \rangle = \frac{\int_W \sigma(\mathbf{x}) I_x^2(\mathbf{x}) d\mathbf{x}}{\Psi} \qquad \text{and,} \qquad \langle I_y^2 \rangle = \frac{\int_W \sigma(\mathbf{x}) I_y^2(\mathbf{x}) d\mathbf{x}}{\Psi}$$

Taking the product

$$\langle I_x^2 \rangle \langle I_y^2 \rangle = \frac{1}{\Psi^2} \left(\int_W \sigma(\mathbf{x}) I_x^2 d\mathbf{x} \right) \left(\int_W \sigma(\mathbf{x}) I_y^2 d\mathbf{x} \right),$$

which, using discrete windows, becomes

$$\langle I_x^2 \rangle \langle I_y^2 \rangle = \frac{1}{\Psi^2} \sum_{i,j} \sigma(i) \sigma(j) I_x^2(i) I_y^2(j)$$

the summation taken over the $(n \times n)$ pixels in the window. Further, since

$$\langle I_x I_y \rangle = \frac{\int_W \sigma(\mathbf{x}) I_x(\mathbf{x}) I_y(\mathbf{x}) d\mathbf{x}}{\Psi}$$

Has discrete equivalent

$$\langle I_x I_y \rangle^2 = \frac{1}{\Psi^2} \left\{ \sum_{i,j} \sigma(i) I_x(i) I_y(j) \right\}^2$$

It follows that

$$\begin{aligned} &\Psi^2 \left(\langle I_x^2 \rangle \langle I_y^2 \rangle - \langle I_x I_y \rangle^2 \right) = \\ &\quad \textstyle\sum_{i \neq j} \sigma(i) \sigma(j) I_x^2(i) I_y^2(j) \\ &\qquad - 2 \textstyle\sum_{i<j} \sigma(i) \sigma(j) I_x(i) I_y(i) I_x(j) I_y(j) \end{aligned} \quad (2a)$$

Here there are $n(n-1)$ different terms of the first sort and $n(n-1)/2$ of the second, where n is the total number of pixels within the window W.
Re-writing equation 2a purely as a single summation (with $n(n-1)$ terms)

$$\begin{aligned} &\Psi^2 \left(\langle I_x^2 \rangle \langle I_y^2 \rangle - \langle I_x I_y \rangle^2 \right) = \\ &\quad \textstyle\sum_{i \neq j} \sigma(i) \sigma(j) I_x(i) I_y(j) \{ I_x(i) I_y(j) - I_x(j) I_y(i) \} \end{aligned} \quad (2b)$$

With equation 2b in mind, consider the cross-product of the two gradients at (x_i, y_i) and (x_j, y_j), where α is the angle between the two gradient vector directions.

$$\begin{aligned} \| \nabla I(i) \times \nabla I(j) \| &= I_x(i) I_y(j) - I_y(i) I_x(j) \\ &= \| \nabla I(i) \| \, . \, \| \nabla I(j) \| \sin \alpha_{ij} \end{aligned}$$

(1) Assuming that the angular separation is small,

$$\sin \alpha \simeq \alpha$$

(2) Further, for unit displacement ds between (x_i, y_i) and (x_j, y_j), the normal curvature orthogonal to the gradient vector κ is given by $\kappa \simeq \sin \alpha$.

Using these observations

$$\begin{aligned} &\| \nabla I(i) \times \nabla I(j) \|^2 \\ &\quad = I_x^2(i) I_y^2(j) + I_x^2(j) I_y^2(i) - 2 I_x(i) I_x(j) I_y(i) I_y(j) \\ &\quad = \{ \| \nabla I(i) \| \, . \, \| \nabla I(j) \| \}^2 \kappa_{ij}^2 \end{aligned} \quad (3)$$

Compare this with equation 2a. By weighting equation 3 with the appropriate (normalised) product of Gaussians and taking the summation over all (i, j), the two are equivalent. Hence, the Plessey operator measures image curvature.

To complete the theoretical analysis, consider the interpretation of the denominator term in equation 1. Again, for the case of discrete sampled means:

$$\begin{aligned} \Psi \{ \langle I_x^2 \rangle + \langle I_y^2 \rangle \} &= \textstyle\sum_i \sigma(i) I_x^2(i) + \sum_i \sigma(i) I_y^2(i) \\ &= \textstyle\sum_i \sigma(i) \| \nabla I(i) \|^2 \end{aligned}$$

Therefore, the denominator provides a measure of the (Gaussian weighted) 'average strength' or gradient magnitude.

4 Image Feature Characterisation Using the Second Fundamental Form

I have shown how the Plessey Corner Finder can be explained in terms of the First Fundamental Form. My approach uses properties of the Second Fundamental Form.

Haralick *et al.* [Haralick-Watson-Laffey 83] proposed an eigen-function representation for the Topographic Primal Sketch (TPS). Gradients, first/second derivatives and the Hessian were used to derive ten pixel labels based on surface and edge properties. However, the calculations of principal curvatures (a crucial part of the scheme) prove complex. Further there is an inherent ambiguity problem with the labelling scheme. An equivalent surface description is provided by using the Gaussian (K) and Mean (H) curvatures. Whereas the principal curvatures are the eigenvalues of the Weingarten Mapping (defined as the matrix $\mathbf{G}^{-1}\mathbf{D}$ where $\mathbf{D}$ is the Second Fundamental Form coefficient matrix defined below, $\mathbf{G}$ is the First Fundamental Form coefficient matrix discussed earlier), H and K correspond to the natural algebraic invariants. However, H and K are scalar quantities. Thus a representation based on their characteristics removes the need to consider directional quantities. Motivated by this and the success of Haralick's TPS the representation I propose to use the characteristics of the Second Fundamental Form.

The foundations of the scheme are derived from the image surface description provided by the facet model [Haralick 80,Haralick 84]. The general principle behind the surface estimation technique is given in the Appendix. Here I derive the relevant differential geometric characteristics of the model.

The Second Fundamental Form is given by,

$$\Phi_2 = -d\mathbf{S}.d\hat{\mathbf{N}} = [dx dy]^T \begin{bmatrix} L & M \\ M & N \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix}$$

where, $d\mathbf{S} = \mathbf{S}_x dx + \mathbf{S}_y dy, \hat{\mathbf{N}} = \mathbf{S}_x \mathrm{x} \mathbf{S}_y / |\mathbf{S}_x \mathrm{x} \mathbf{S}_y|$ the coefficients $L = \mathbf{S}_{xx}.\hat{\mathbf{N}}$, $N = \mathbf{S}_{yy}.\hat{\mathbf{N}}$, $M = \mathbf{S}_{xy}.\hat{\mathbf{N}}$, and their defining matrix usually denoted by $\mathbf{D}$.

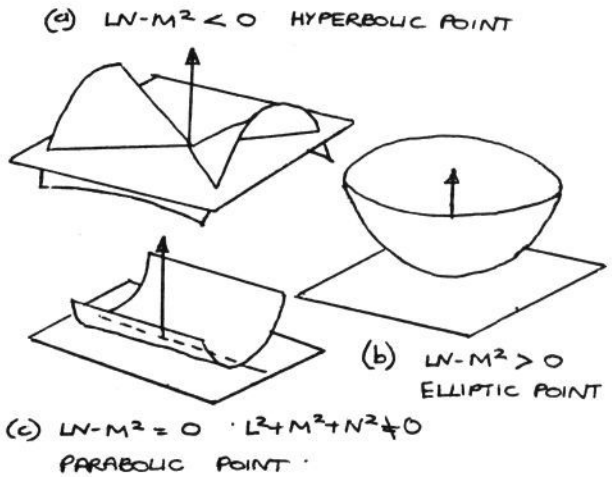

Figure 1: Local Surface Classification in terms of the properties of the Second Fundamental Form

The determinant of $\mathbf{D}$ can be used to provide a pixel label describing the local surface geometry. Referring to Figure 1 for a planar point $L = M = N = 0$, a parabolic point $LN - M^2 = 0$, a hyperbolic point $LN - M^2 < 0$, and an elliptic point $LN - M^2 > 0$.

For the noise-free case, such a geometric classification is complete. 'Interesting' points are associated with neighbourhoods containing strong evidence of two-dimensional intensity variation (elliptic and hyperbolic points). For real data, a purely geometric approach is insufficient. I propose using a statistical analysis of noise to provide 'confidence' in labelling. Preliminary empirical results based on this idea are presented in the next section.

4.1 Preliminary Results

Results are presented for running the algorithm on synthetic and real data.

Figure 2 shows the characterisation of common idealised 2-D image structures. Groups of localised 2-D (elliptic and hyperbolic) labels correctly identify corners and intersections.

The 2-D structure identified by the algorithm for an asymmetric chess board is shown in Figure 3a. A Canny operator assumes that a discontinuity has the local structure of a step. Figure 3b illustrates the result of applying the Canny algorithm to the same chess board as in Figure 3a.

For real images a purely geometric model is inadequate. Figures 4a-e, show the pixel classification for the Cup image. Clusters of hyperbolic and elliptic points appear around object corners and at 'T'- junctions; an observation consistent with Nagel's gray-value corner definition. Namely, a gray-value corner lies between the local maxima of positive Gaussian curvature (elliptic point) and local maxima of negative Gaussian curvature (hyperbolic).

Preliminary empirical investigations suggest that a suitable measure (C) on which to base statistical noise analysis is

$$C = \sqrt{EG - F^2}.\frac{|\kappa_1| + |\kappa_2|}{2}$$

This measure is closely related to that proposed for the Kitchen-Rosenfeld and Zuniga-Haralick corner detectors. Figure 4f shows the result of thresholding the Cup Image hyperbolic points at a 95 % confidence level on this measure.

Conclusions

This paper has looked at the problem of finding 2-D structure in images. First it was shown how a novel corner detector (in the sense it is based on first differentials) estimates image curvature. This detector is only suitable for 'L'-junctions, as its performance is unpredictable on other higher order structures.

The solution I proposed is based on the geometric properties of the image surface. Future work intends to extend the analysis to real data.

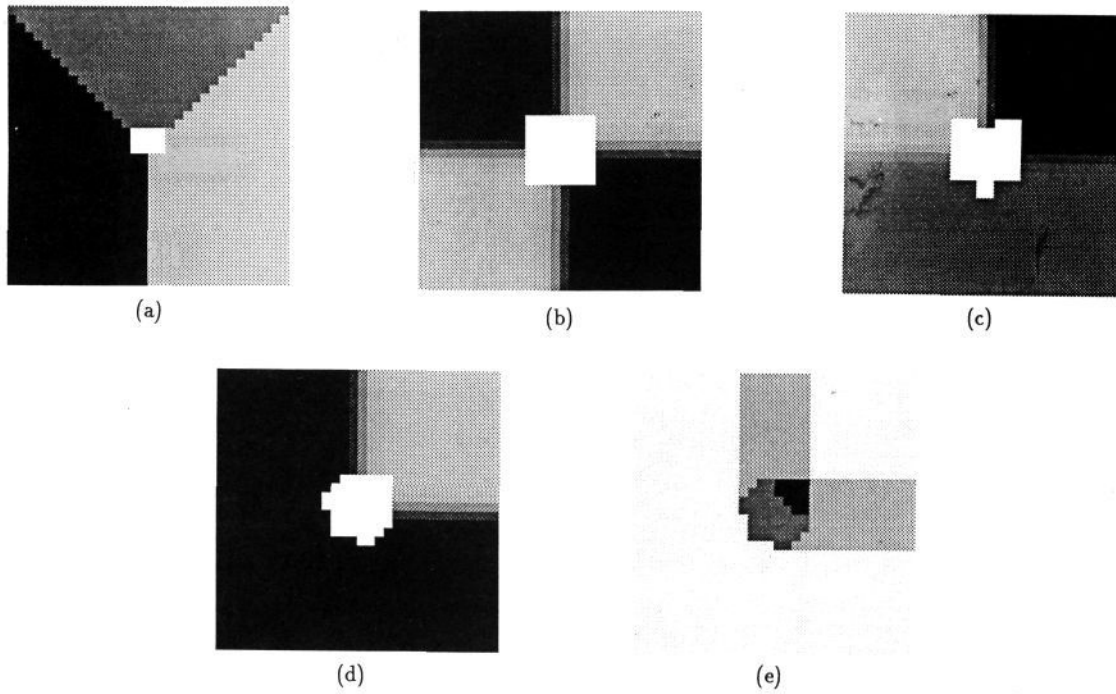

Figure 2: Characterisation of idealised 2-D image structures; 2-D structure identified by the algorithm are highlighted on the original. (a) 'Y'-junction, (b) 'T'-junction, (c) 'X'-junction,(d) Corner.(e) shows the distribution of elliptics(black),hyperbolics(dark grey) and parabolics(light grey) around the corner in (d).

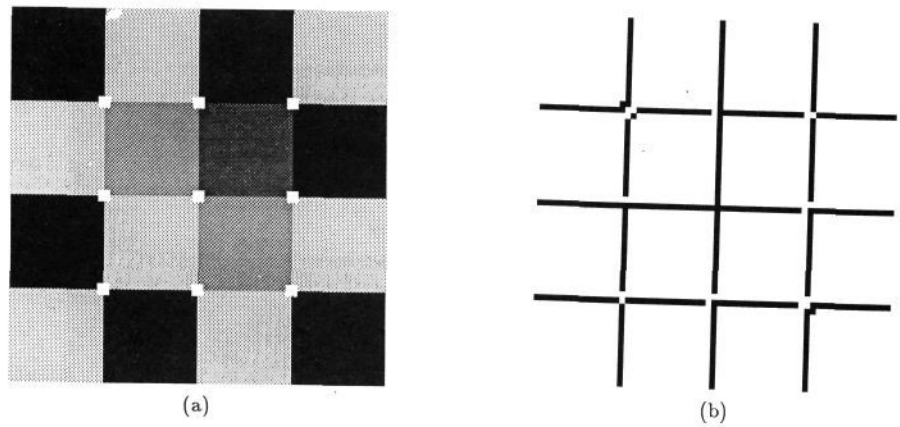

Figure 3: Chess board: (a) 2-D structure identified by the algorithm (highlighted white), (b) a Canny operator fails to correctly mark the intersections

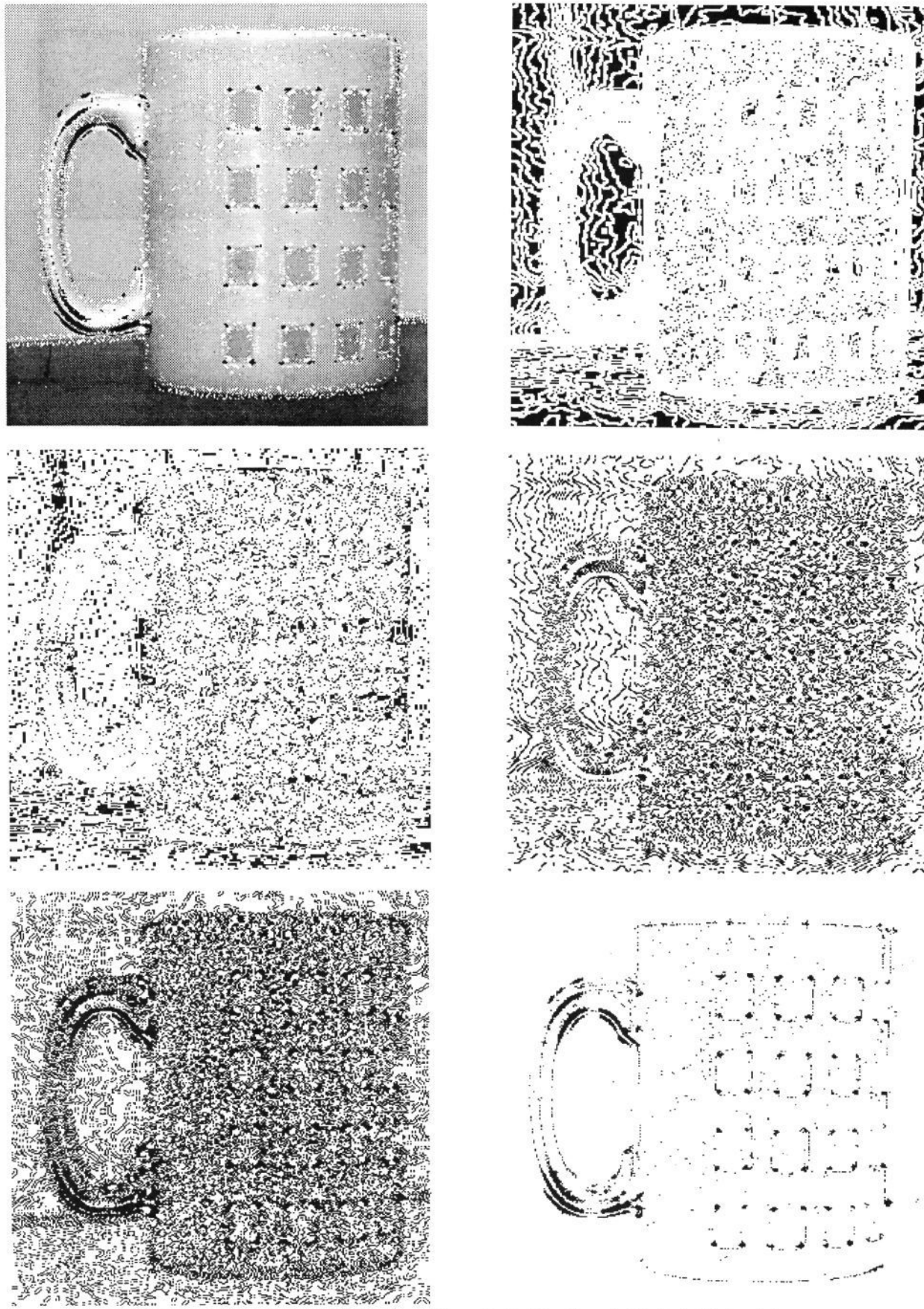

Figure 4: Cup image: (a) Clusters of hyperbolic (black) and elliptic (white) points appear around object corners and along curved edges. Pixel classification of entire image after smoothing with a Gaussian of $\sigma = 5$, (b)flat, (c) parabolic, (d) elliptic, (e) hyperbolic points. (f) Thresholding hyperbolics on a measure of cornerness suppresses false labelling due to noise.

In particular I intend to look at

- Statistical noise models to provide a more robust measure of 'confidence' in pixel labelling.
- The topographical relationship between labels to define symbolic tokens for 'T'-, 'Y'- and other 2-D structures.
- Model sensitive to the choice of surface function order, window size, and polynomial basis set.
- Ultimately combining all these ideas to provide a robust 2-D image representation.

The work is very much on-going research. Progress to date is encouraging and supports past work on 2-D image structure. More significantly I have demonstrated how by treating the gray-levels as a surface it is plausible to reliably find 2-D cues in real images.

Acknowledgements

I am grateful to Mike Brady for providing the initial motivation for this work and for numerous helpful discussions and suggestions. Thanks go to Chris Harris, and Margaret Fleck for discussing their algorithms and Dave Forsyth for some useful comments. The Canny output is from an implementation written by Peter Foulkes.

Appendix — 2-D Chebychev Facet Model

To explain the general principle behind the model, the analysis below considers the simplest case of a (3×3) window. The extension to larger neighbourhoods follows similar reasoning but involves more tedious calculations.

Assume a (3x3) neighbourhood, using coordinates (x, y) centred on the pixel of interest as shown.

$$\begin{array}{ccc} (-1,-1) & (0,-1) & (1,-1) \\ (-1,0) & (0,0) & (1,0) \\ (-1,1) & (0,1) & (1,1) \end{array}$$

Let the intensity function be given by $I(x, y)$ as usual. The digital image pixel values define a sampled version of the real continuous intensity surface in 3-D space. Consider approximating this intensity function by the

$$I(x,y) = \sum_{n=0}^{8} a_n P_n(x,y)$$

Here, P_i refers to the *ith* Chebychev polynomial defined below.

$$\begin{array}{llll} P_0(x,y) = 1 & & P_1(x,y) = x & \\ P_2(x,y) = y & & P_3(x,y) = x^2 - 2/3 & \\ P_4(x,y) = xy & & P_5(x,y) = y^2 - 2/3 & \\ P_6(x,y) = xP_5(x,y) & & P_7(x,y) = yP_3(x,y) & \\ P_8(x,y) = P_3(x,y)P_5(x,y) & & & \end{array}$$

The coefficients $a_0, a_1, \ldots, a_8$ may be found using the orthogonality property

$$a_n = \frac{\sum_x \sum_y P_n(x,y) I(x,y)}{\sum_i \sum_j P_n^2(i,j)}$$

This implies that the fitting coefficients can be computed as a linear combination of the data values in $I(x, y)$ with coefficients

$$\frac{\sum_x \sum_y P_n(x,y)}{\sum_i \sum_j P_n^2(i,j)}$$

Solving for each of the parameters produces the following nine convolution masks.

$$\underset{a}{1/9\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}} \quad \underset{b}{1/6\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}} \quad \underset{c}{1/6\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}}$$

$$\underset{d}{1/6\begin{bmatrix} 1 & -2 & 1 \\ 1 & -2 & 1 \\ 1 & -2 & 1 \end{bmatrix}} \quad \underset{e}{1/4\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}} \quad \underset{f}{1/6\begin{bmatrix} 1 & 1 & 1 \\ -2 & -2 & -2 \\ 1 & 1 & 1 \end{bmatrix}}$$

$$\underset{g}{1/4\begin{bmatrix} -1 & 0 & 1 \\ 2 & 0 & -2 \\ -1 & 0 & 1 \end{bmatrix}} \quad \underset{h}{1/4\begin{bmatrix} -1 & 2 & -1 \\ 0 & 0 & 0 \\ 1 & -2 & 1 \end{bmatrix}} \quad \underset{j}{1/4\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}}$$

Each mask may be applied independently to the image data to determine parameter estimates for all image pixels. In principle, this implies that it is possible to express $I(x, y)$ up to the fourth degree. The interpretation of the surface is sensitive to choice of the degree of the polynomial, and window size. The results presented here use the fourth order 3×3 model.

Consider then the differential geometry of an image (Monge) surface $\mathbf{S}(x, y)$ where

$$\mathbf{S}(x,y) = x\mathbf{i} + y\mathbf{j} + I(x,y)\mathbf{k}$$

The First Fundamental Form is defined by the equation,

$$\Phi_1 = d\mathbf{S}.d\mathbf{S} = Edx^2 + 2Fdxdy + Gdy^2$$

Assuming a fourth order Chebychev model

$$\begin{aligned} I(x,y) &= a + bx + cy + d(x^2 - 2/3) + exy + f(y^2 - 2/3) \\ &\quad + gx(y^2 - 2/3) + hy(x^2 - 2/3) + j(x^2 - 2/3)(y^2 - 2/3) \end{aligned}$$

the First Fundamental Form coefficients can be derived in terms of the parameter estimates:

$$\begin{array}{rcll} E &=& 1 + I_x.I_x &= 1 + (b - 2/3g)^2 \\ G &=& 1 + I_y.I_y &= 1 + (c - 2/3h)^2 \\ F &=& I_x.I_y &= (b - 2/3g)(c - 2/3h) \end{array}$$

$$EG - F^2 = 1 + I_x^2 + I_y^2 = 1 + (b - 2/3g)^2 + (c - 2/3h)^2$$

The Second Fundamental Form is given by,

$$\Phi_2 = -d\mathbf{S}.d\hat{\mathbf{N}} = Ldx^2 + 2Mdxdy + Ndy^2$$

Here $\mathbf{N}$ is the local surface normal.
In terms of estimated parameters (again for the fourth order model):

$$\begin{array}{rcll} L &=& I_{xx} \Big/ \sqrt{1 + I_x^2 + I_y^2} &= 2(d - 2/3j) \Big/ \sqrt{EG - F^2} \\ N &=& I_{yy} \Big/ \sqrt{1 + I_x^2 + I_y^2} &= 2(f - 2/3j) \Big/ \sqrt{EG - F^2} \\ M &=& I_{xy} \Big/ \sqrt{1 + I_x^2 + I_y^2} &= e \Big/ \sqrt{EG - F^2} \end{array}$$

$$LN - M^2 = 4(d - 2/3j)(f - 2/3j) - e^2 \Big/ (EG - F^2)$$

References

[1] H. Asada and M. Brady. The Curvature Primal Sketch. *IEEE Trans. Pattern Anal. Machine Intel.*, PAMI-8(1):2–14, 1986.

[2] S.T. Barnard and W.B. Thompson. Disparity analysis of images. *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-2:333–340, 1980.

[3] L. Dreschler and H.H. Nagel. Volumetric model and 3-d trajectory of a moving car derived from monocular tv-frame sequence of a street scene. In *Proc. IJCAI*, pages 692–697, 1981.

[4] I.D. Faux and M.J. Pratt. *Computational Geometry for Design and Manufacture.* Ellis-Horwood, 1979.

[5] M.M. Fleck. Representing space for practical reasoning. April 1987. Thesis proposal.

[6] R.M. Haralick. Digital Step Edges from Zero-crossings of Second Directional Derivatives. *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-6(1):58–68, 1984.

[7] R.M. Haralick. Edge and Region Analysis for Digital Image Data. *Comput. Graphics Image Processing*, 12:60–73, 1980.

[8] R.M. Haralick, L.T. Watson, and T.J. Laffey. The Topographic Primal Sketch. *Int. J. Robotics Research*, 2(1):50–72, 1983.

[9] C.G. Harris. Determination of Ego-motion from Matched Points. In *Proc. Alvey Vision Conference*, Cambridge,England, 1987.

[10] C.G. Harris and J.M. Pike. 3D Positional Integration from Image Sequences. In *Proc. Alvey Vision Conference*, Cambridge,England, 1987.

[11] L. Kitchen and A. Rosenfeld. Gray-level corner detection. *Pattern Recognition Lett.*, 1:95–102, 1982.

[12] M.M. Lipschutz. *Differential Geometry.* McGraw-Hill, NY, 1969.

[13] G. Medioni and Y. Yasumuto. Corner detection and curve representation using cubic b-splines. In *IEEE Int. Conf. Robotics and Automation*, pages 764–769, , 1986.

[14] F. Mokhtarian and A. Mackworth. Scale-based description and recognition of planar curves and 2d shapes. *IEEE Trans. Pattern Anal. Machine Intel.*, PAMI-8(1):34–43, 1986.

[15] H.P. Moravec. Towards automatic visual obstacle avoidance. In *Proc. Int. Joint Conf. Artificial Intelligence*, page 584, Cambridge, Mass., August 1977.

[16] H.-H. Nagel. Displacement vectors derived from second order intensity variations in image sequences. *Comput. Graphics Image Processing*, 21:85–117, 1983.

[17] J.A. Noble. The Geometric Structure of Images. July 1987. M.Sc. report.

[18] L. Spacek. *The Computation of Visual Motion.* PhD thesis, Univ. Essex, Colchester, 1984.

[19] S. Ullman. *The Interpretation of Visual Motion.* MIT Press, Cambridge, Mass., 1979.

[20] O.A. Zuniga and R.M. Haralick. Corner detection using the facet model. In *Proc. Conf. Pattern Recognition Image Processing*, pages 30–37, 1983.