Progress toward a system that can acquire pallets and clean warehouses

Michael Brady, Stephen Cameron,
Hugh Durrant-Whyte, Margaret Fleck,
David Forsyth, Alison Noble,
and Ian Page
Robotics Research Group,
Department of Engineering Science,
University of Oxford,
Oxford OX1 3PJ,
U.K.

Abstract

Mobile robots are a paradigm of the challenge of systems integration in Robotics. We discuss work at Oxford on the development of an Autonomous Guided Vehicle that can acquire free-standing pallets and can clean warehouses. The work is based on a commercially-available, free-ranging AGV that can plan paths and sense its position using an infrared laser triangulation system. We describe work aimed at providing the AGV with: vision, sonar, and a direct ranging sensor; the ability to integrate these sensors; and 3D geometric reasoning capabilities.

Introduction

The challenge of systems integration is ubiquitous in Robotics, in topics that seem, at first sight, to be completely different: sensor-based assembly; compliant process control; legged locomotion; and, perhaps most comprehensively, path-planning and guidance of mobile robots. Integration is necessary because actuators need to be integrated with sensors and with systems that can reason about geometry and forces. Integration is also necessary because a system that combines the processing of several sensors, or sensor modalities such as stereo and motion, can overcome the inadequacies of each individual sensor. It is partly for this reason that there has recently been a spate of mobile robot research projects, also partly because another round of system building is considered timely by some Robotics researchers and by some funding agents. Outside research laboratories, the use of Automated Guided Vehicles in industry tripled in 1985 alone (Hollier 86), and they constitute one of the fastest growing sectors of the Robotics business. This paper reports work on an Autonomous Guided Vehicle (AGV) at Oxford University. The work springs from a need and an opportunity.

- The *opportunity* arises because the FAST Division of the GEC Company has recently marketed a free-ranging AGV (constructed by Caterpillar) that is designed for the movement of material in factories (Figure 1). The AGV is free-ranging in that it does not require specially prepared roadways or wire guides, it can plan alternative routes to a goal (say, to a pallet or to a machine tool) and it can replan when it is confronted with an unexpected obstacle as detected by the touch bar on the front of the AGV. The GEC AGV plans movements in a 2D representation of the factory floor. It is equipped with a rotating infrared laser scanner that reads bar codes attached to the factory wall. By reading three or more bar codes it can find its position by triangulation. The AGV is accurate to better than half a centimetre over thirty metres. Though odometry is available to the AGV controller, it is rarely used in practice because of wheel slippage on factory floors. The AGV communicates over a radio link to a controlling computer, which maintains a "map" of the factory layout that includes details of processing centres and the "roadways" between such centres. The system software can plan a path between processing centres that avoids obstacles (including other AGVs). The GEC-Caterpillar system is an impressive advance on the technology and controlling software of industrial AGVs. GEC have generously donated an AGV to the Oxford Robotics Research Group.

- The *need* arises because the GEC AGV, like every other mobile robot, is limited in scope. The primary application of the GEC-Caterpillar AGV is in the development of flexible manufacturing systems in which parts and partially manufactured objects are individually routed through a series of processing centres. For many such applications, the sensing and planning capabilities of the GEC-Caterpillar AGV may well be sufficient (though operating machinery has already posed problems for the infrared sensors). For some applications, however, the sensory capabilities, and as a result the planning capabilities, clearly need to be augmented. The GEC AGV can accurately locate itself within its *known* environment as specified by the set of bar charts, whose locations are accurately determined at set-up time. The AGV is cannot sense objects that are not in pre-determined positions other than by colliding with them, and even in that case it simply assumes that the obstacle fills an entire lane of a roadway through the

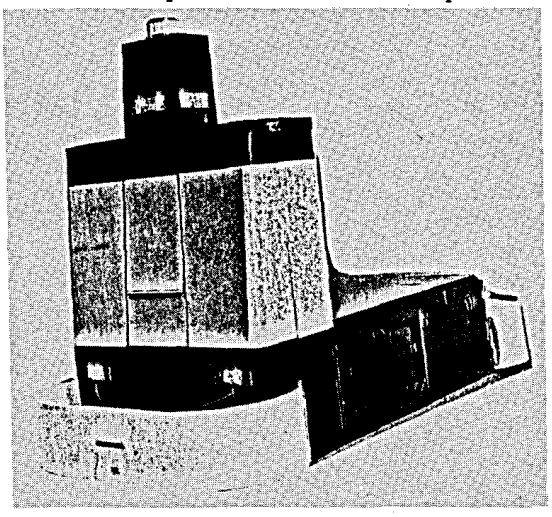


Figure 1: The GEC Autonomous Guided Vehicle. See text for details.

factory. Similarly, the AGV's representation is restricted to a two-dimensional "bird's eye view" of the factory layout. It is incapable of 3D reasoning of the sort required for stacking and palletising.

For these reasons, our work is aimed at extending the GEC AGV's repertoire to the following applications:

1. The GEC-Caterpillar AGV is regularly commanded to move to a place where it can pick up a pallet for delivery to a processing centre. Typically, pallets are placed by fork-lift trucks. Inevitably, they are positioned inaccurately and sometimes the wrong way round. Often, several pallets accumulate at the reception centre, and this complicates the pick-up procedure. We intend to use a mixture of range sensing and vision to determine the position and configuration of a pallet. The information produced by our programs should support spatial reasoning to determine which of several pallets should be picked up, and how.

2. It is intended that the GEC-Caterpillar AGV will be used for applications such as cleaning a large, unattended area. Related applications include warehousing and crop harvesting. So long as the "avenues" to be cleaned are clear, the AGV can perform its job predictably and well. Obstacles, however, pose a problem. Consider, as a typical domain, clearing a baggage handling area at a major airport: if the obstacle is determined to be an item of luggage, it should be pushed gently out of the avenue; but if the obstacle is a trolley, a slight deviation should be planned to take the AGV around it. Note that only a few different types of obstacles may be expected and this information can be utilised in the interpretation programs.

In our current design, the AGV will be equipped with vision, a thirty-two element sonar system (Steer 85), and a novel ranging device (Reid, Rixon, and Messer 84). The vision processes, we are developing include mixed active/passive stereo, motion, and shape from contour. Progress in vision is described in the next section. The responses of the various sensor processes are being integrated using the methods of Durrant--Whyte, and progress is sketched in Section 5. The geometric capabilities of the enhanced GEC AGV are being provided by the S-bounds technique of (Cameron 84) and recent progress, including a computer model of the AGV environment at Oxford and path planning algorithms for the computer model, are described in Section 4. Work on sonar and ranging will be reported elsewhere. Clearly, an AGV of the type we are constructing demands computational power in excess of that provided by serial processors of reasonable cost (M68020 Workstations for example). We are implementing several algorithms on a SIMD/MIMD processor, called the Disputer, designed by (Page 87) and which consists of an array of Transputers coupled to a SIMD engine that implements RasterOp (alias BitBlt) as a primitive. The Disputer is described in Section 3.6

Rich representation of image structure

This section describes some of our recent progress in vision. We begin by noting that representations that capture some the tight constraint available at colour and texture edges and at loci of two-dimensional image change potentially support fast algorithms for the computation of motion, shape from contour, and stereo (see (Brady 87) for more details). We then show that the information we seek can in fact be computed reliably. First, we describe Fleck's *Phantom Edge Finder* (Fleck 87), and we sketch some recent work on the differential geometry of image surfaces by (Noble 87). Then we introduce the work of (Forsyth 87) on a system that can detect and describe colour edges. Finally, we describe a powerful SIMD-MIMD processor which has been constructed at Oxford on which the algorithms described in this Section are being implemented.

Seeds of perception

Not all information is created equal. Different locations in an image (or range image) offer different levels of constraint to visual processes. Generally, a region of an image that is the projection of a portion of a smooth surface imposes less constraint on visual interpretation than do one-dimensional loci of change, which often correspond to visual edges. Informally, we shall refer to loci of one-dimensional intensity change as "edges" despite the fact that the step edges ubiquitously studied in computer vision are only one of a number of important kinds of change (Canny 83) (Ponce and Brady 87) (Asada and Brady 86). Edges offer even more constraint if they can be elaborated by descriptions that relate to important visible surface characteristics such as texture or colour (see below). We call regions that correspond to smooth surface patches *loci of zero-dimensional change*, and they are roughly characterized by two large eigenvalues in the image autocorrelation function as the image "looks" the same in all directions. Similarly, we call edges *loci of one-dimensional change*. By analogy, the autocorrelation function has a large eigenvalue whose eigenvector lies along the edge and a small eigenvalue whose eigenvector lies in the direction of the normal to the edge. Similarly, we refer to corners, points of occlusion (eg T-junctions and X-junctions), various curvature maxima, and configurations of edges that have nearby terminations as *loci of two-dimensional change*.

This observation suggests that the computation of the parameters of a visual process might be computed most effectively by basing that computation first on loci that offer most powerful and then decreasingly powerful constraint. Often this corresponds to working from loci of two-dimensional change, then on loci of one-dimensional change, and finally on loci of smooth change. Indeed, it seems possible that models can be invoked very early in this computation, possibly even on the basis of the determination of the most constraining information available in the image. This suggests in turn that the refinement of model invokation and the determination of rich image descriptions can proceed hand in hand. According to this broad scheme, points of tightest constraint are *seeds of perception* from which are grown more extensive descriptions of change and, eventually, image descriptions. (Brady 87) has explored this theme in a number of visual problems: shape from contour; optic flow and structure from motion; stereo; shape from shading; and model-based recognition of objects. We summarise the main results here.

The computation of *shape from contour* is one of the most powerful passive ranging techniques in human vision. Analysis has concentrated mostly on smooth, planar contours. However, the determination of shape from contour is most effective when there are curvature discontinuities along a curve. Consider a planar curve $\gamma(s)$ that is imaged after undergoing a general affine transform $\mathbf{T}$, that is, a translation, rotation, and scaling. Under affine transforms, as well as under orthographic or per-

spective image projection $\mathbf{P}$, the zero-crossings and curvature peaks of γ appear as zero-crossings and curvature peaks of the image of $\mathbf{P}(\mathbf{T}(\gamma))$ (see also (Marr 77) (Huttenlocher and Ullman 87)). In general, metric quantities such as angles, lengths, and ratios of lengths associated with the shape γ are not preserved in $\mathbf{P}(\mathbf{T}(\gamma))$, and so they are of limited usefulness for determining shape from contour. There are, however, at least two constraints that *can* be used to effect the determination of γ from its oriented projection $\mathbf{P}(\mathbf{T}(\gamma))$:

1. *the order constraint*: the order of curvature changes around the projected planar shape is unaffected by affine transformation. Other shapes may occlude the shape of interest, as for example when an aeroplane is partly occluded by cloud. In such cases, curvature changes not associated with the shape of interest may obtrude into the curvature change sequence; nevertheless, subsequences of curvature changes associated with the shape to be recognised are generally visible.

2. *the type constraint*: there are many different types of curvature change; for example, Asada and Brady consider *corner, crank, end, smooth join, bump* and *dent*, whereas (Hoffman and Richards 82) propose a number of different *codon types* for representing curvature changes. Under a broad range of values for the scaling factor of the affine transformation (determined by the object shape and by the viewing conditions), the type of a curvature change of γ is the same as that of the corresponding curvature change in $\mathbf{P}(\mathbf{T}(\gamma))$. For example, in Asada and Brady's notation, the projection of a transformed crank change is typically a crank.

Asada and Brady implemented an unpublished program that recognised a variety of shapes (aeroplanes) that had undergone affine transformation and partial occlusion. The model was represented as a sequence of curvature changes, each with an associated type. Recognition consisted of a Waltz-like labelling of curvature changes to match subsequences of the curvature changes found in an image to subsequences of the model. Significantly, though not surprisingly, composite types such as cranks and ends were most effective for constraining the subsequence match.

Shape matching and recognition based on local salient features by Asada and Brady and by Turney, Mudge, and Volz may be contrasted with the approach of (Grimson and Lozano-Pérez 86). They argue that matching based on salient features is unreasonable when data is noisy (precluding the accurate computation of salient features) or when objects are overlapped (salient features are more likely to be occluded than gross features). There is a law of excluded middle operating here: salient features cannot be *relied upon* for recognition, for the reasons advanced by Grimson and Lozano-Pérez. However, recognition can be more effective, reliable, and efficient if such salient features are available. Recently, working in our Laboratory, (Stein 87) implemented a recognition program that was a blend of (Turney, Mudge, and Volz 85) and (Grimson and Lozano-Pérez 86). Figure 1 shows a typical recognition result obtained by Stein's program.

Most work on *optic flow* is based on the *motion constraint equation*:

$$\mathbf{N} \cdot \mu = \frac{I_t}{\|\nabla I\|} \qquad (1)$$

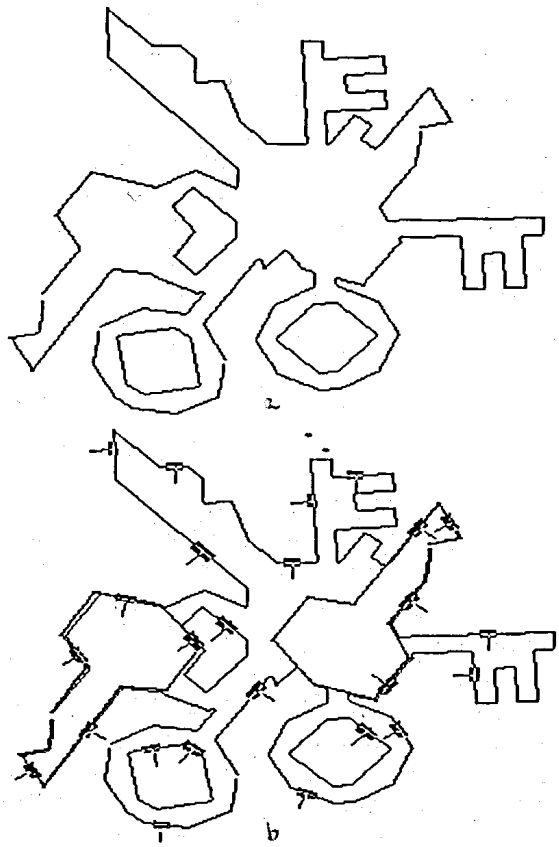


Figure 2: a. Outline of a set of overlapped shapes. b. Key shapes found in the pile shown in a.

which is a first-order Taylor series expansion of $I(\mathbf{x}+\delta\mathbf{x}, t+\delta t)$ (Nagel 87). In this equation, $\mu(\mathbf{x})$ is the optic flow field and $\mathbf{N}$ is the image unit normal $\nabla I/\|\nabla I\|$. Since the image gradient $\|\nabla I\|$ occurs in the denominator of the motion constraint equation, the computation of $\mathbf{N} \cdot \mu(\mathbf{x})$ is poorly conditioned unless $\|\nabla I\|(\mathbf{x})$ is large. Often, points $\mathbf{x}$ at which the gradient is large correspond to edges. This is the basis of Hildreth's (Hildreth 84) scheme for the computation of visual motion. Her algorithm identifies points $\mathbf{x}$ at which the image gradient is large with zero-crossings of a Laplacian of a Gaussian filter (Marr and Hildreth 80). The main novelty of her approach is based on a theorem that states that if $\gamma(s)$ is a closed contour (as zero-crossing contours must be unless they are thresholded or cross the image boundary) then there is a unique optic flow field $\mu(\mathbf{x})$ that minimises the smoothness expression:

$$\oint_\gamma \left(\frac{d\mu}{ds}\right)^2 ds$$

so long as the optic flow is different at at least two different points along the closed curve. Hildreth combines the edge smoothing term with the motion constraint equation using a Lagrange multiplier, and solves the resulting minimisation problem using a conjugate gradient descent algorithm. This minimisation algorithm is inherently sequential (Gong 87), rendering it difficult to implement Hildreth's algorithm in parallel. One way to proceed is to replace the conjugate gradient algorithm by a scheme such as Terzopoulous' multigrid relaxation algorithm (Terzopoulos 83) or graduated non-convexity (Blake and Zisserman 87). We are exploring an alternative approach.

Nagel (Nagel 87) (Dreschler and Nagel 82) has shown that, in practice as well as principle, the *full* optic flow field is com-

putable at what he calls "grey-level corners". Noble (Noble 87) critically analyses Nagel's grey-level corner model. Recently, Nagel (Nagel 86) (Nagel and Enkelmann 86) has shown that the smoothness assumptions underlying the algorithms of Horn and Schunck, of Hildreth, and others amount to special cases of an "oriented smoothness" assumption that is implicit in the use of higher order derivatives. We have proved a generalisation of Nagel's result (see (Brady 87) for the proof.)

Theorem 1. If $\mathbf{H}(\mathbf{x})$ is the image Hessian, and κ is the (planar) curvature of a level contour of the intensity function whose tangent is $\mathbf{T}$ and whose normal is $\mathbf{N}$, then

$$(\mathbf{N}^\top \mathbf{H}\mathbf{T})\mathbf{N} \cdot \mu = \|\nabla I\| \kappa \mathbf{T} \cdot \mu.$$

Since $\mathbf{T} \cdot \mu$ is required to compute the full flow field, it follows that

1. the full optic flow can be computed at any location along a level contour (including zero crossings) at which the curvature is non-zero.

2. the reliability, or numerical conditioning, of the computation of the full flow increases as the curvature increases. Hence it is most reliable at corners.

Nagel (Nagel 87) suggests using the full flow field at corners to determine the optic flow elsewhere. This makes sense since the computation is most well conditioned at those points. Dreschler and Nagel write down the second *spatial* derivative and first *temporal* derivative of the flow field *everywhere* in the image. They note the crucial role played by the image Hessian. This is to be expected from elementary fluid mechanics (Landau and Lifschitz 59). Similarly, in a series of theoretical papers, Koenderinck and van Doorn [1975, 1976, 1981, 1985, 1986] have noted invariant properties of the motion parallax field due to the movement of rigid bodies relative to an observer. Barnard and Thompson (Barnard and Thompson 82) have pursued a similar approach. Similarly, (Davis, Wu, and Sun 83) have studied the motion of image corners in the relatively easy case of images of moving polyhedra.

Returning to Hildreth's computation, we noted above that zero-crossing contours are closed as is required for the application of her main theorem. In general, however, zero-crossing contours of a Laplacian-of-a-Gaussian filter do not necessarily belong to a single object, so that motion smoothing can produce poor results. The Laplacian-of-a-Gaussian does not provide as good an edge map as is produced by a directional operator such as Canny's. In turn, Canny's edge map is not as good, particularly at loci of two-dimensional change, as that produced by Fleck's phantom edge finder (Fleck 87). Although Hildreth's theorem relies upon closed contours, it is straightforward to show that:

Theorem 2: If the full optic flow is specified as $\mu(s_1) = \mu_1$ and $\mu(s_2) = \mu_2$ at two points s_1, s_2 along a level contour, then there is a unique flow field $\mu(s)$ which minimises Hildreth's edge smoothness integral and satisfies the given boundary conditions. □

Theorem 2 can be used to speed the convergence of Hildreth's conjugate gradient scheme by restricting it to portions of contours between loci of two-dimensional change at which the full optic flow field can be computed. Between such loci, one might either:

1. Use the Hessian-based formula developed in Theorem 1 directly. This would be a bad idea because the computation of $\mathbf{T} \cdot \mu$ is poorly conditioned when the curvature is small.

2. or, develop a relaxation formula to interpolate the optic flow between known values μ_1, μ_2 as in Theorem 2.

We are pursuing the latter approach. Note that the situation is trivial for straight edges for which the optic flow linearly interpolates the values at the corners (Murray, Castelow, and Buxton 87). More generally, it is easy to show that:

$$(\mathbf{T} \cdot \mu)(s + \delta s) = (\mathbf{T} \cdot \mu)(s) + \delta s \, \kappa (\mathbf{N} \cdot \mu) \qquad (2)$$

Note that $\delta s \, \kappa = \delta \alpha$, the incremental change in the tangent angle along the curve.

In its usual formulation (Buxton 87), the geometry of *stereo* is a simple case of the geometry of 3-d rigid body motion. This assumes that the stereo cameras have previously been rectified, perhaps mechanically. The term "stereo" vision is also coined for stereo-mapping and aerial reconnaisance in which the aeroplane motion between successive images is well approximated by a translation, rotation, and magnification in the image plane. In such applications, rectification may or may not be a separate pre-process prior to stereo matching (Hannah 80) (Gennery 79). Rectification processes typically determine the global image rotation, magnification, and translation by applying a least squares process to a suitable set of sampled points $\mathbf{x}_j^i$, where $i = 1, 2$ denotes the image from which the sample point is taken. The sample set needs to be sparse (for efficiency) and sufficiently constraining to determine the rectification parameters. Points with low autocorrelation are often chosen as the sample set.

Evidently, rectification can be viewed as a partial stereo match. Indeed, some authors have based stereo algorithms on the sparse point sets that are also suitable for rectification. In perhaps the earliest example of this (Moravec 77) developed an "interest operator" that isolated small image areas with large intensity variation in the four principal directions. His algorithm worked from a coarse to a fine scale choosing the fifty most "interesting" points to match in order to compute a rough range map for his roving vehicle. A variation of Moravec's operator was developed and used by (Hannah 80) in her work on aerial passive navigation and by (Barnard and Thompson 82) in their work on optic flow.

Moravec's stereo matcher restricted attention to a few points for reasons of efficiency. This may not be the only reason to base a stereo matching algorithm upon seeds that are loci of two-dimensional intensity change. Recently, Rogers has noted that the (relative) disparity between two matched scene points varies inversely with the square of their depth difference (see (Brady and Hopkins 87) for a precise statement of this). Similarly, the disparity gradient between those points varies inversely with the difference in depth. Rogers has pointed out that the second derivative of disparity (which, by analogy, is called disparity curvature) of those points does not vary with depth, which makes it a particularly attractive parameter on which to base object recognition. In general, the computation of disparity curvature is ill-conditioned. However, as disparity curvature is related to surface curvature, it is best conditioned at loci of two dimensional image change. The computation of such points is also well suited to computation on SIMD processors. We are currently exploring the use of such points as seeds for stereo.

3.3 The phantom edge finder

Figures 3 and 4 illustrate results with a new edge finding algorithm, named the *Phantom Edge Finder* because of its resemblance to the (Watt and Morgan 87) *MIRAGE* algorithm for one-dimensional boundaries. It also owes a considerable debt to (Pearson and Robinson 85) for the "cartoon" representation of intensity changes. The *Phantom Edge Finder* removes camera noise while preserving fine texture and sharp corners. Its output boundaries are thin, without "feathering" or multiple responses. Results from different scales are combined into a single edge map, and extraneous edges in staircase patterns are removed during scale combination.

The *Phantom Edge Finder* is, in many ways, similar to existing edge finders. Its performance derives from a new view

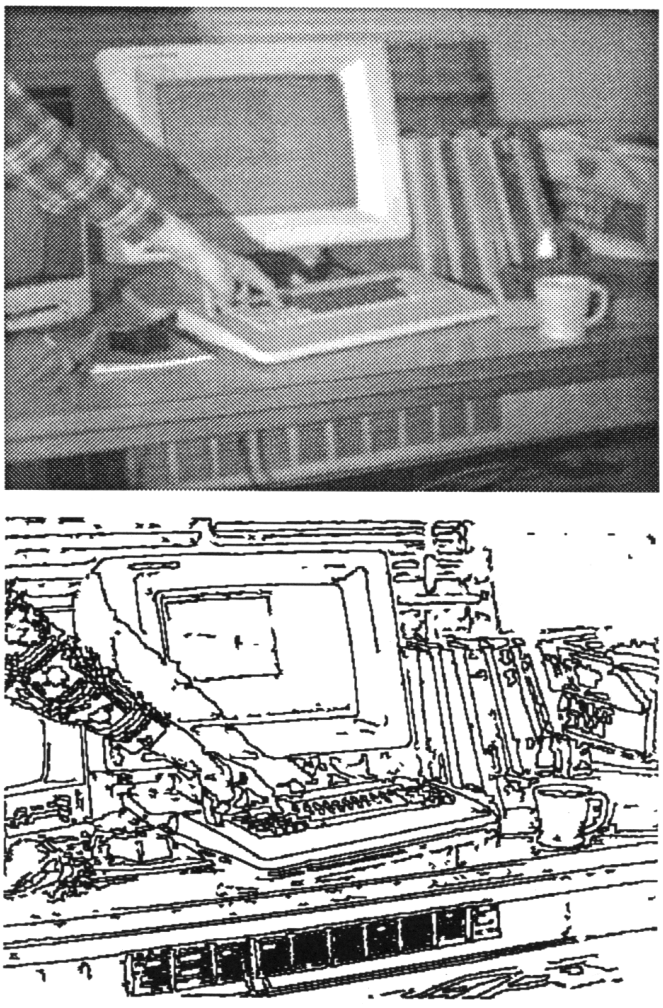
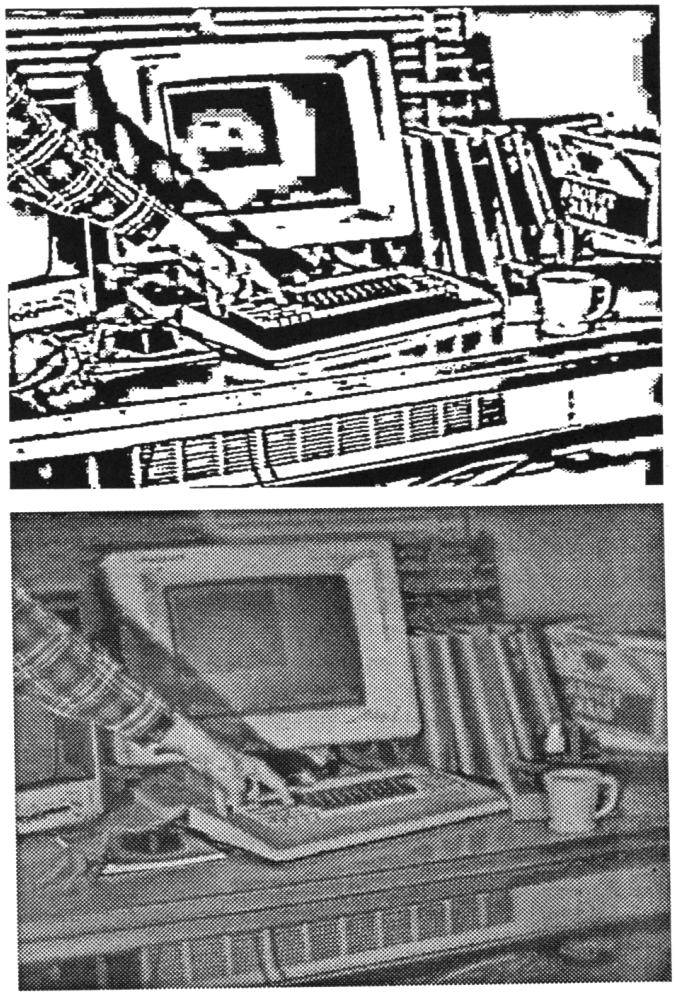


Figure 3: Output of the Phantom Edge Finder on an image of a computer console. Top left: original image. Top right: cartoon, in which black cells are on the dark side of an edge, white cells on the light side, and grey cells both or neither. Bottom left: edge map. Bottom right: reconstructed intensities.

of the relationship between regions and boundaries that is discussed more fully in (Fleck 87). According to this view, *regions* represent subsets of space while *boundaries* represent the topology of space. In vision, regions are represented by sets of cells, such as the receptors in a CCD camera. Boundaries are placed between adjacent cells, that is, between cells sharing an edge or a vertex. In the *Phantom Edge Finder*, most processing takes place at cells. Function values (e.g. intensity) and labels used in computation are stored at cell locations. Boundaries are only represented implicitly, by putting contrasting labels on pairs of adjacent *edge* cells that are separated by boundaries. In fact, the *Phantom Edge Finder* does not compute boundaries directly, but identifies edge cells and deduces the boundary locations from them. Many edge finder computations are done over all cells with significant second difference responses, which form wide bands of cells near boundaries. These algorithms do not fit neatly into the traditional distinction between region--based and boundary-based algorithms.

The *Phantom Edge Finder* detects boundaries using second differences, for three reasons:



1. The sign of a first difference depends on the direction of motion along the corresponding one-dimensional path; that of the second difference does not.

2. A first difference responds at the boundary itself. The second difference, however, gives a high response *near* boundaries. Since processing is done at cell locations and boundaries are located between cells, this is the most useful type of response.

3. Roof edges are signalled by peaks and valleys in the second differences; they are not signalled explicitly in first difference responses.

The *Phantom Edge Finder* detects boundaries using directional second differences. For the rectangular arrays assumed by the current program, second differences are taken in four directions: horizontal, vertical, and two diagonal directions. The operator used is $[1, 0, -2, 0, 1]$. To equalize the responses of step edges and thin bars, the second difference response $a - 2b + c$ is normalized by $\|max(a-b, c-b)\|/\|(a-b)+(c-b)\|$. Because noise is suppressed later in the program, prior smoothing with a Gaussian or related filter is not needed, and a small operator can be used to preserve fine detail. The noise suppression relies on the fact that camera noise is added *after* optical blurring. Noise is identified by the following criterion:

- A cell response due to a real edge has a star-convex neighborhood of responses of the same sign, where the sum of the responses over that neighborhood is high.

The *Phantom Edge Finder* avoids the problem of "feathering" by combining responses from different directions *before* extracting zero crossings. Specifically, it tries to classify each cell with a significant second difference response as either on the light side ("light") or on the dark side ("dark") of a boundary. Cells with no significant response are unlabelled. It is, however, possible for a cell to be on the light side of one boundary and to be on the dark side of another. This happens when the intensity surface has a saddle. Intensity saddles can be images of a saddle on the imaged three-dimensional surface, or they may be caused by smoothing a junction of three or more regions. Such cells are labelled as both light and dark.

The classification of cells as dark or light is based on computing the maximum-amplitude positive and negative responses over all directional differences. For an isolated step edge, the maximum amplitude response is in the direction closest to perpendicular to the boundary. If more than one boundary is involved, the maximum-amplitude response(s) reflect differences perpendicular to the boundaries with the largest intensity changes Since a cell can be on both the light and dark sides of different boundaries, separate positive and negative responses are computed for each cell.

This method of combining directional responses has been designed to give good performance on sharp corners and places where several regions meet at a point. For example, Figure 5 illustrates the performance of the *Phantom Edge Finder* on the tines of a fork. Most edge finders combine responses in ways that are not good indicators of the boundary strengths involved in such cases; examples include: the sum of the directional responses, other non-directional center-surround operators (Marr and Hildreth 80), and the sum of the responses of the correct sign. Such techniques tend to give overly high values to the insides of sharp corners and overly low values to the outsides. In theory (Berzins 84), the zero-crossing of the Laplacian of a Gaussian is closed around a sharp corner but balloons out. In practice, the weak outside response of this type of operator causes the zero-crossing to randomly merge into the background noise. The combination method used in the *Phantom Edge Finder* does not require that the directional responses conform to any particular pattern, e.g. have a unique maximal response, peak responses in some directions, or responses that can be modelled as a linear transformation. Previous work has tended to depend on such assumptions (Canny 83) and (Haralick, Watson, and Laffey 83), although they break down at sharp corners and boundary intersections. Canny's edge finder, for example, tends to leave small gaps at such points.

The maximum-amplitude positive and negative responses

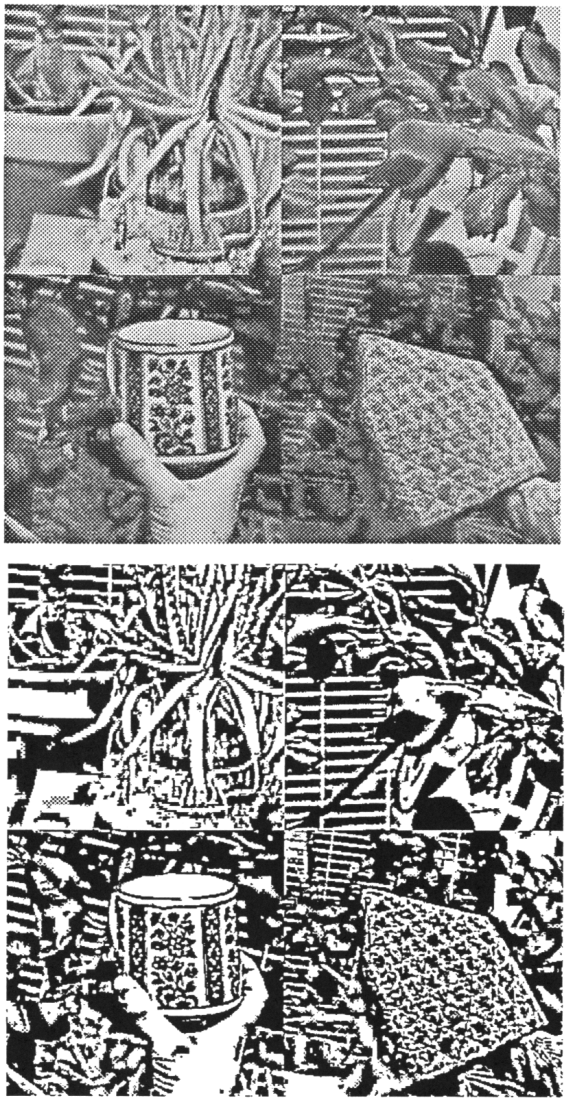


Figure 4: Output of the Phantom Edge Finder on an image of four textured patterns.

are then used to classify each cell as dark or light. Presence of a non-noise response of one type is not sufficient grounds for assigning that label to the cell, since cells near the zero-crossing in a step edge may have both negative and positive responses. In such cases, one response is significantly larger than the other, unless the cell is actually straddling the boundary. Only in cases where the two responses are of similar strength does the program assign both labels. When one response is more than 1.5 times the other response, *Phantom Edge Finder* gives the cell only a label reflecting the larger response.

From the dark/light classification of cells, the algorithm then extracts boundaries. A boundary is considered to exist between a pair of cells when they have opposite dark/light labels. These boundaries correspond to step edges. Thin bars, even as thin as one cell wide, are found as a pair of step edges. Because cells may not be labelled, or may be assigned both labels, boundaries can end abruptly. Roof edges are determined as regions of dark or light response that are not near dark/light transitions. Currently, the algorithm does not detect roof edges which are close to step edges robustly. Further detail, particularly on the combination of information from different scales and the elimination of staircase phenomena, are described in (Fleck 87).

The local geometry of images

The inadequacy of edge finders that are based on detecting large differentials stems from the implicit assumption that edges are loci of one-dimensional change. The simplest example of two-dimensional 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. 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 (Dreschler and Nagel 82), a multitude of parameters are required for a 'T'-junction. 'T'-junctions are relatively simple two-dimensional structures.

With the ultimate goal of defining a two-dimensional image representation, we have investigated some of the differential geometric properties of the intensity image structure. We have shown how the geometry of a simple facet model can characterise idealised instances of features such as intensity junctions and corners. The analysis given here and in (Noble 87) provides the *Phantom Edge Finder* with a theoretical underpinning in differential geometry.

(Haralick, Watson, and Laffey 83) proposed a eigenstructure representation for the Topographic Primal Sketch. Gradients, first and second derivatives, and the Hessian were used to derive ten pixel labels based on surface and edge properties. The calculations of principal curvatures (a crucial part of the scheme) proved 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 Map (defined as the matrix $\mathbf{G}^{-1}\mathbf{D}$ where $\mathbf{D}$ is the Second Fundamental Form and $\mathbf{G}$ is the First Fundamental Form), 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 by the performance of Haralick's *Topographic Primal Sketch*, we 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). We approximate the image surface locally as a linear combination of (the first eight) Chebychev polynomials, to distribute noise evenly through the window. In practice, we work with a (5×5) window; here we derive the simpler case of a (3×3) window centered on the origin, and covering $(x, y) : x = -1, 0, 1; y = -1, 0, 1$.

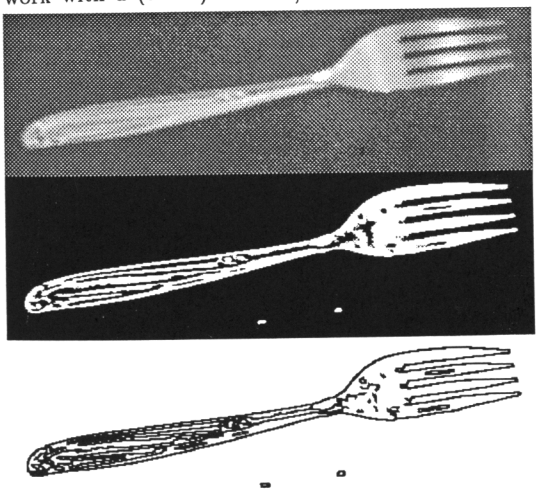


Figure 5: An image of a fork with sharply pointed tines, cartoon output, and edge map.

The intensity function $I(x, y)$ is approximated as:

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

where P_i refers to the *ith* Chebychev polynomial:

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

The coefficients $a_0, a_1, \ldots, a_8$ may be found using the orthogonality of the Chebychev polynomials:

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

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.

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

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

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

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. Now consider the differential geometry of the image surface:

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

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:

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

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

Similarly, the Second Fundamental Form is given by,

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

where $\mathbf{N}$ is the local surface normal. In terms of estimated parameters:

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

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

The matrix of coefficients of the Second Fundamental Form is denoted by $\mathbf{D}$. The determinant of $\mathbf{D}$ can be used to provide a pixel label describing the local surface geometry. As is well-known, a planar point is defined by $L = M = N = 0$, a parabolic point $LN - M^2 = 0$, a hyperbolic point $LN - M^2 < 0$, and an elliptic point $LN - M^2 > 0$.

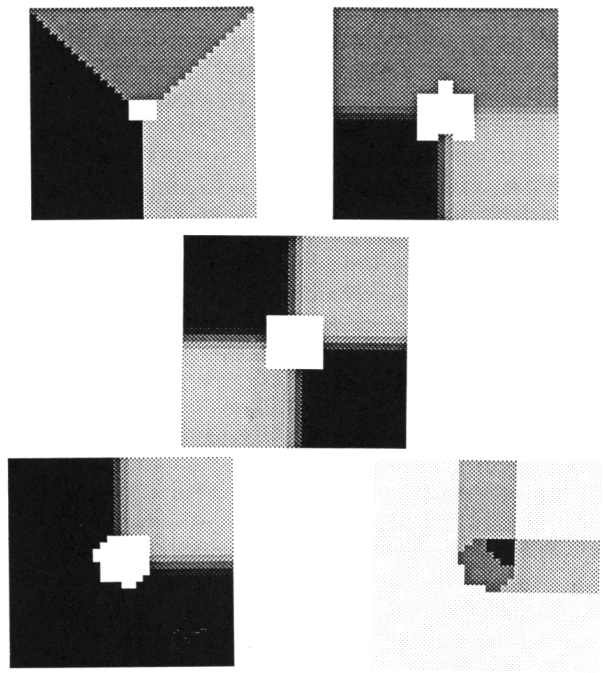


Figure 6: Characterisation of idealised 2D image structures; 2D structures identified by the algorithm are highlighted on the original. (a) 'Y'-junction, (b) 'T'-junction, (c) 'X'-junction,(d) Corner.(e) shows the distribution of elliptics(black),hyperbolics(dark grey) and parabolics(light grey) around the corner in (d).

For noise-free images, this geometric classification is complete. 'Interesting' points are associated with neighbourhoods containing strong evidence of two-dimensional intensity variation (elliptic and hyperbolic points). Results are presented for running the algorithm on synthetic and real data. Figure 6 shows the characterisation of common idealised two-dimensional image structures. Groups of localised two-dimensional (elliptic and hyperbolic) labels correctly identify corners and intersections. The two-dimensional structure identified by the algorithm for an asymmetric chess board is shown in Figure 7a. A Canny operator assumes that a discontinuity has the local structure of a step. Figure 7b illustrates the result of applying the Canny algorithm to the chess board image.

For real images a purely geometric model is inadequate. Figure 8a, show the pixel classification for one image of a cup from a motion sequence. 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 that 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 8b shows the result of thresholding the Cup Image hyperbolic points at a 95 % confidence level on this measure.

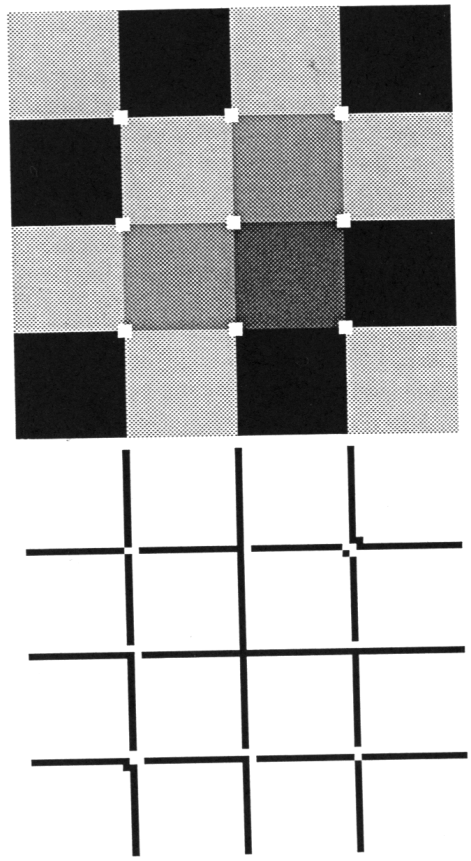


Figure 7: Chess board: (a) 2D structure identified by the algorithm (highlighted white), (b) a Canny operator fails to correctly mark the intersections.

A system that finds changes in colour

Colour provides additional constraint for intermediate vision processes by enriching the descriptions attached to edges with information that concerns properties of the imaged surface. Colour information can be of considerable use in vision applications:

- *Colour makes segmentation more reliable.* The reader might care to look for four impala ram and one lamb in Figure 9. If you cannot, try looking at the colour version of the picture. (Horn 86) has described how the grey level

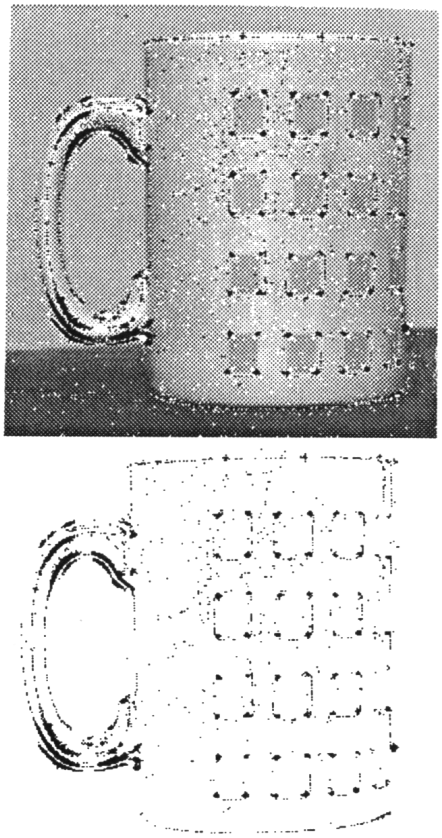


Figure 8: Cup image: (a) Clusters of hyperbolic (black) and elliptic (white) points appear around object corners and along curved edges. Pixel classification of entire image after smoothing with a Gaussian of $\sigma = 5$, (b) Thresholding hyperbolics on a measure of cornerness suppresses false labelling due to noise.

information at pixels depends both on the reflectance and the orientation of a surface. However, the colour of a pixel also depends as well on the shape of the surface reflectance function, which, by making assumptions about the spatial variation of the spectral composition of the illuminant, we are able to recover.

- *Colour constrains matching.* Consider matching edges from a pair of edges, as is usually done in stereo, motion, and in computing shape representations such as *Smoothed Local Symmetries.* Complex images often pose considerable picket-fence problems. Matches may be disambiguated by recording the colour present on each side of an edge in each image.

- *Colour can aid visual monitoring of processes.* For example, the problem of visually monitoring the cooking of meat is most reliably determined from its hue.

The most useful information is concentrated at changes, and so we need to be able to find changes in colour.

To digitize scenes, we use a CCD camera and a number of different gelatine filters. The following differences between our camera setup and the human eye may be noticed:

- The chromatic aberration of a camera lens is so small (at the frequencies of interest) that it can be neglected. As a result, each colour channel has the same spatial resolution.

- There are as many kinds of receptors at each point as we care to put there, within the limitations of the physics of available filters.

These differences are considerable, but the kernel of the problem still remains: *given a set of receptor responses, and knowing the receptor sensitivities, recover the surface reflectance function, up to a constant* (we refer to this rather loosely as the *shape* of the surface reflectance function.) It is clear that this is not possible without additional constraints. Fundamental to all computational work on computing the shape of surface reflectances is the assumption that illumination changes slowly over space. This is equivalent to the assumption that a sharp change in the colour signal is due to a sharp change in the world of reflecting objects, rather than to a sharp change in illumination.

For this reason, the following seems to be a reasonable account of early colour vision:

1. Compute the best approximation to the colour signal, and find the colour changes.

2. From those changes, construct an estimate of the spatial structure of the illuminant.

3. From this, construct a map of the surface reflectance functions that are presumed to have caused the original colour signal.

In our current work on finding colour changes, we avoid the problem of colour constancy by assuming that the spectral content of the illuminant varies slowly over space. Thus, all perceivable spatial changes in colour in images of a given scene are due to changes in the shape of the surface reflectance function. This assumption underlies all colour constancy work, but it is not the same as assuming isochromatic illuminants from scene to scene.

In related work, Nevatia constructed a colour edge detector by applying the Hückel operator to the red, green and blue filtered images with the intention of using the additional information to improve the original edge map, but concluded that most edge information was in the intensity image (Nevatia 77). It turns out that this is essentially correct; but it is an unfortunate conclusion: rather than being a source of *new* edges; the available colour information is a *source of rich descriptors* to attach to existing edges. (Machuca and Philips 83) proposed a colour edge detector operating on the phase of the IQ vector (effectively a representation of hue), but presented few results. (Gershon 85) proposed a colour edge detector based on the presence in the cortex of double opponent cells, but few experimental results.

If we wish to use the spatial bandwidth of our system, the option of inspecting the colour at either side of a brightness change must be rejected. Cameras plus filters are able to spatially localise isoluminant changes in colour, a task on which humans perform poorly. We can exploit this ability. Since we are attempting to find a property of surfaces from images we need to detect changes in the *shape* of the colour signal, and these are assumed to correspond to changes in the shape of the spectral reflectances of the world. An opponent coding of a colour signal determines the shape of the colour signal, if we normalise it by the intensity of the two components involved. For example, $\frac{(B-Y)}{(B+Y)}$, where B is the blue component of the colour signal and Y is yellow, determines the relative size of the "humps" at the long wavelength and short wavelength ends of the spectral energy density. Similarly, $\frac{(R-G)}{(R+G)}$ deter-

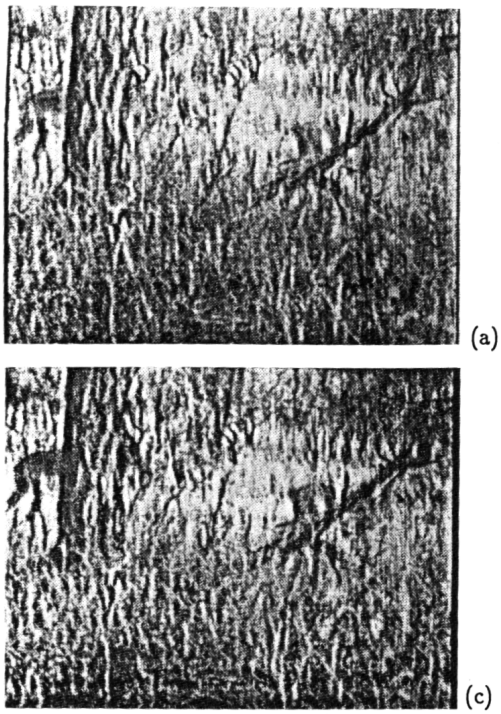


Figure 9: **Impala Rams:** there are four impala rams and one lamb in this image. (a) shows the intensity image of the impala. (b) shows the red component of the colour version of this picture. (c) shows the green component of the colour version. (d) shows the blue component of this picture.

mines the finer scale shape at the long wavelength end of the spectrum. It follows that the opponent cell responses represent a sensible decomposition of the colour signal that the sensors can reconstruct.

Recall that the only signal available to the system is the projection of the colour signal onto three functions. We can represent the colour signal as an infinite sum relative to any given set of basis functions. For the first three basis functions, we may choose any set of functions that span the space spanned by the receptor sensitivity functions. Then, to reconstruct the signal, we simply compute the coefficients of the receptor sensitivity functions in the expansion: in essence, we are blind to the rest of the signal, by construction. This is contrary to the proposal of (Barlow 82), who analysed the response of the receptors to a Fourier decomposition of the spectrum presented. There is a natural basis, namely an orthogonalisation of the three receptor sensitivity functions.

Double opponent cells (see for example (Daw 72), (Gershon 85) or (Livingstone and Hubel 84)) assume a particular significance in this analysis. Consider the receptor sensitivity functions in Figure 10. Notice that there is a clear separation between the hump at the blue end of the spectrum, whereas the receptors at the longwave end of the spectrum are rather close together. Thus, a signal reconstructed as a linear combination of these functions can qualitatively be described by considering two questions:

1. (i)is the hump at the shortwave end of the signal larger or smaller than the composite hump at the longwave end?
2. (ii)considering the finer scale structure of the longwave end of the signal, which hump is larger?

The responses of opponent cells can be viewed as answering these questions. This suggests that colour edges may be detected by applying a conventional edge detector to an opponent encoding of the image. In our implementation, the colour edge finder works as follows:

The opponent signals are computed, and a pyramid of scaled and resampled versions is constructed for the opponent signals, the intensity signal, and the opponent intensity signals. For those regions in the image where the opponent intensity signal is below some threshold, the opponent signals are adjusted to be in balance, as reliable information about the nature of the colour signal and its changes is not available at such points.

The response regions for each opponent at each scale are calculated. Notice that at this stage we do not use the normalised opponent intensity signals since we wish to force the edges in the opponent signals to line up with those in the intensity signal. We have found that using the opponent signals leads to an average error of less than a pixel (or rather, an occasional error of a pixel, and no greater errors), whereas using the normalised opponents leads to a larger error which is difficult to deal with in a principled way. The response regions label the image with areas where the opponents have swung one way or the other. In this way, we can isolate the isoluminant colour changes.

We construct zero crossing maps of the image, where the zero crossings are labelled with (i) the colour in the original image; and (ii) with the nature of the change in colour across

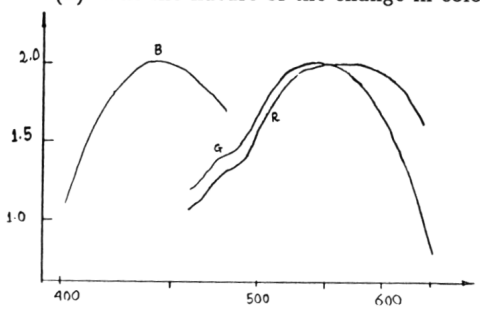


Figure 10: **Receptor sensitivity functions:** The graph shows human cone sensitivity functions plotted as log relative sensitivity against wavelength. The graphs were simplified from Bowmaker and Dartnall, 1980.

the zero crossing. Keeping the images in register is relatively straightforward: the fact that the *Phantom Edge Finder* is so accurate at localising changes, means that the response regions coincide to within at most a single aberrant pixel. Locally, we then require edges in the opponents either to be in register with edges in the brightness image, or to be far from them.

This can easily be enforced. For each pixel marked by the *Phantom Edge Finder* as being on the 'light' side of a zero crossing, we inspect its neighbours that are marked similarly. The neighbours of these pixels which are not marked as being on either side of a zero crossing, are sorted by opponent response, and the majority vote is accepted as a labelling for all pixels visited. The same must be done for all pixels marked as being on the 'dark' side of a zero crossing, *mutatis mutandis*. This technique deals with registration errors in the response images that are one pixel wide and this is the worst case in practice.

As noted in the previous subsection, phantom edges are zero crossings generated from an image by inflexions in the image surface. These occur when, for example, a red region is separated from a green region by a white region: the white region is then redder than the green region and greener than the red region. (Clark 86) has suggested that these may be rejected when one marks zero crossings by considering the actual change over the puported zero crossing: our colour change program is based on an idea due to (Fleck 87), which works by considering the way these zero crossings tend to arise when one combines a map of responses obtained at a coarse scale with a set obtained at a finer scale. A version of Clark's technique is also used, as these spurious edges, although not terribly common in intensity images, are for some reason a real problem in opponent encoded images. Opponent zero-crossings are subjected to this technique as well.

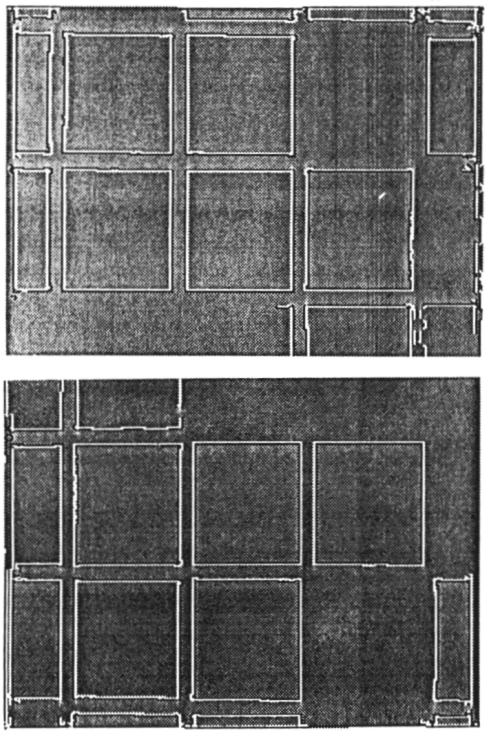


Figure 11: **ColorChecker Edges:** (a) shows the zero crossings of the R-G opponent. (b) shows the zero crossings of the B-Y opponent. The red (blue, resp.) sides of edges are marked white, the green (yellow, resp.) black.

We show results in Figures 11 and 12; the images are presented as a set of three component images, and the response regions and zero crossing maps are shown for a number of examples.

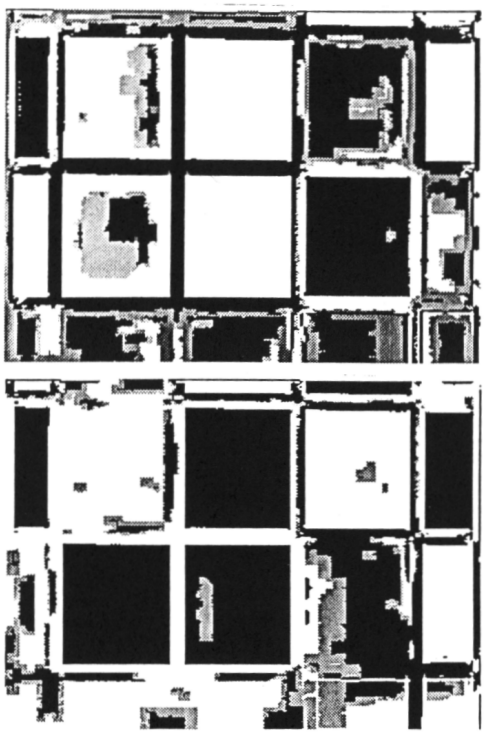


Figure 12: **ColorChecker Edges:** (a) shows the response regions of the R-G opponent. (b) shows the response regions of the B-Y opponent. The red (blue, resp.) responses are marked white, the green (yellow, resp.) black.

The *Disputer*: a dual-paradigm parallel processor for graphics and vision

The *Disputer* is a parallel processor which is a closely coupled arrangement of a 256-processor SIMD machine and a 42-processor MIMD machine. Algorithms at the "back-end" of the graphics pipeline and at the "front-end" of machine vision applications, are often characterised by local-support operations on large arrays of pixels. The data-parallelism of early vision and late graphics computations are well suited to SIMD processing. Later vision, earlier graphics, and many robot control tasks typically exhibit task parallelism that is better matched to the MIMD model of computation. The *Disputer* is the combination of: a SIMD array processor called *DisArray*; a MIMD network of Inmos Transputers; and controller hardware which has been designed to link them together. The two parallel machines are closely coupled and the SIMD machine has a real-time video output channel. The entire system is programmed in Occam 2. This machine is now allowing us to investigate dual-paradigm algorithms very effectively.

Earlier work on a SIMD machine for graphics applications (Page 83) resulted in a system called *DisArray* (Display Array) that significantly reduced one of the major bottlenecks of real-time graphics: rendering the image into a frame buffer. The initial motivation for building *DisArray* was to execute RasterOp (BitBlt), an important operation in bitmap-based graphics, in parallel at very high speed. Though it was primarily intended for exploring parallel algorithms for graphics applications, *DisArray* is a general purpose SIMD machine that is similar to the AMT (previously ICL) *Distributed Array Processor* (S.F.

73). Because it is a general purpose SIMD processor, *DisArray* has proved an effective base for developing a broader class of algorithms than originally envisaged. Parallel algorithms have been developed for many vision and graphics problems, such as polygon rendering.

DisArray is an array of 256 single processing elements (PEs) in a 16×16 arrangement with four nearest-neighbour communications. Row-based and column-based broadcast lines transmit data, addressing, and control information to the PEs. In addition, a video shift register is threaded through the PEs, which supports real-time display of a 512 × 512, 16-colour bitmap from some part of the array memory. A proposed re-design of the video board will upgrade the screen resolution and allow real-time video input as well. Each PE has 256k by 1-bit memory, implemented by a single dynamic RAM. All processors in the array execute the same, globally broadcast, instruction at the same time. The array has a low-level scheduler which arbitrates between requests for such computational cycles and video refresh cycles.

A sequential controller (a Transputer) generates the instructions for the SIMD array. Having given a single instruction to the array, the controller and array operations then proceed in parallel. The array instructions are generally of the form :

Mem [addr] := F (Mem [addr], Register, RowData AND ColumnData)

where F is an arbitrary Boolean function and the operation takes place between two 256-bit square words. The instruction register for the array, together with various address and data registers for communication are mapped into the address space of the controller. The controller itself is a 20MHz, 32-bit, 1Mbyte RISC machine, based on the T414 transputer.

The MIMD array is a six-by-seven array of Transputers, each of which is a 20MHz T414 with no external support chips. The memory for each Transputer is limited to the 2kbytes of on-chip, 50nS static RAM. The nearest-neighbour connections in the array are hard-wired and the 26 edge connections are brought out to a patch panel. The edge connections are also used to communicate with the *DisArray* controller and with the Transputer development system. We have recently augmented the MIMD processor network with a further 40 Transputers each with 256kb of external RAM.

There is a 16-bit DMA link between the *DisArray* controller and a Unix host processor. A running application on the *Disputer* can arbitrarily access the Unix memory. This link is often used to locate and continuously refresh the screen picture from a display file, or other applications-oriented data structure, in the address space of some Unix process. We also use the DMA link to get image data into the *Disputer* from a Datacube frame grabber/processor which is acessible from the Unix host over an Ethernet.

At present, all of the software for the *Disputer* is in Occam 2 and the software development is carried out on a 2Mbyte T414 transputer with an IBM PC/AT acting as terminal and filestore. The complete SIMD/MIMD application can be expressed as a single Occam program, even though it might consist of many hundreds of processes. The development system (transputer and/or IBM PC) can also act as a host to the *Disputer* system, in which case communication is by up to 4 transputer links rather than the DMA link.

Currently, we have a simple low level vision system working on the SIMD hardware and we will soon extend this to incorporate Fleck's *Phantom Edge Finder* described earlier. We then intend to develop some simple intermediate-level vision algorithms running on the MIMD engine performing, for instance, parallel matching against a database of models. It has become increasingly clear that there is a great deal of computational commonality between graphics algorithms and those being developed by the machine vision community, most obviously in the data-parallel algorithms. This is perhaps not too surprising, as graphics and vision are two sides of the same coin, whose currency is computational geometry. Indeed we now feel that the degree of commonality is such that it warrants the development of a *virtual machine* for graphics and vision and we are actively working on its definition. This will provide an interface between applications programs and the highly parallel hardware which implements the kernel algorithms of the application. It is the intention that the virtual machine is not heavily orientated towards any particular hardware model, but should be implementable in a number of different ways.

Geometric reasoning

Geometric reasoning will play a key role within the AGV project. The main geometric reasoning system will perform three major functions:

- Planning collision-free paths for the vehicle, including adjusting the plan to avoid unexpected obstacles (detouring);
- Planning pallet acquisition, using a fork-lift attachment to the vehicle;
- Providing geometric modelling support for the sensor systems, including visibility information.

The last of these functions affects mainly the organisation of the geometric data bases, and will not be discussed in detail here. Providing the first two functions involves the use of a number of techniques/algorithms from different branches of A.I. and robotics, notably different forms of path-planning, path-checking, and search techniques. Although our short-term goal is to provide the three major functions listed above, we are also interested in building a computational framework that will support other geometric reasoning tasks and queries, for example the cleaning application; thus we are putting some effort into the organisation of the geometric reasoning system.

The World Model

The world model is a central database that defines what the AGV believes its surroundings to be. The central role of the world model is illustrated by the system architecture, shown in figure 13. (This diagram is, of course, only a first approximation to the truth!) In this architecture the primary flow of information is from top to bottom, as information from the different sensors is combined, sifted, pruned, and "interesting" features lodged in the world model. In fact, to avoid the AGV sitting and thinking about its surroundings for long periods of time it is necessary for *predictive* information to flow from the world model to the sensors, and so there is a secondary flow of information to be catered for. One important feature of this architecture is that the planning system believes the world model to be accurate; in particular, there is no direct link between the planner and the sensors. Such an approach is not normally recommended in most robotic systems as uncertainty abounds. However we believe that such an approach

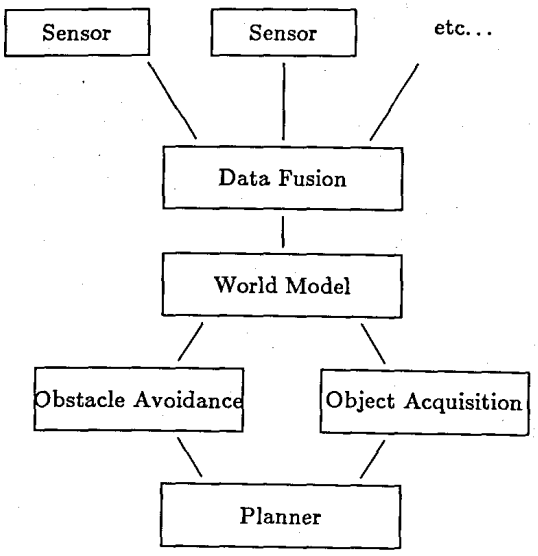


Figure 13: Overall System Architecture

can be tolerated, at least for the work described herein, for three reasons. Firstly, the environment of the vehicle is reasonably friendly; it is not liable to be attacked, and reaction time is not critical. Secondly, the vehicle itself is a relatively inaccurate machine (by normal robotic standards) operating in a fairly coarse environment: there should be little need for the precise, guarded motions of the type required, say, in robotic assembly work. Thirdly, the reliability of the system is not critical; if the vehicle does occasionally fail to find a way of performing its task it can sit and bleep to itself, awaiting human interaction. We see the world model itself as consisting of four components, accessed through a kernel (figure 14). Two of the components are essentially static, namely the Factory Layout Model and the 3D Solid Models. The factory model "looks" like a two-dimensional plan of the factory, on which are marked static items (e.g., machining centres, pillars, doorways), quasi-static items (e.g., waste bins, doors), and nominal roadways. The 3D models are three-dimensional representations of objects that the vehicle senses or (literally) comes into contact with, for which a simplified two-dimensional projection will not suffice. (If there are many instances in the factory of, say, parts bins, only a single instance is stored in this component.) The other components of the world model will change, both due to the discovery of unexpected objects and due to the movement of the vehicle itself. One component is the feature cache—it will store features that are useful for sensing. For example, once a convenient landmark has been identified in a view and its position computed the prominent features can be stored in the cache, as being potentially useful and probably (but not necessarily) invariant. (Of course, such a cache needs a regime for disposing of information that has outlived its usefulness.) The route-planning component has a similar function to the feature cache, namely to save information that was expensive to produce and that could be useful. In this case, the information will consist of local detours around obstacles that can be reused (providing the obstacle does not move). Local C-space maps, which are generated during the obstacle avoidance phases, may also be cached.

Obstacle Avoidance

The fact that the environment of the AGV is reasonably well-

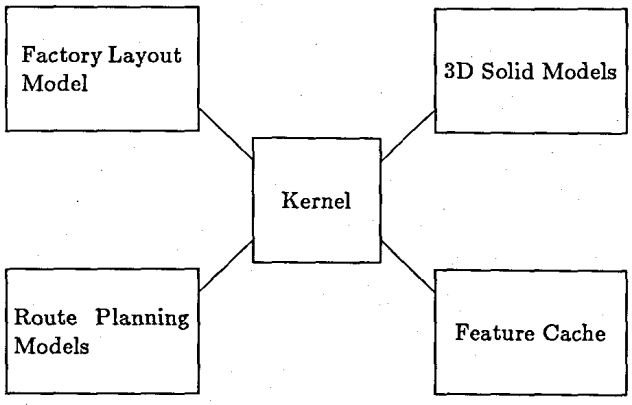


Figure 14: World Model Components

structured means that we can take advantage of very simple path planning algorithms; in particular, much of the time the AGV can use generate-and-test, whereby a path is proposed and then checked for validity. In turn, proposing paths for the AGV is normally quite simple, as unless there are reasons to do otherwise the vehicle can just uses the factory roadways. The only real problem occurs when an unexpected obstacle is encountered, when we expect one of three strategies to be used:

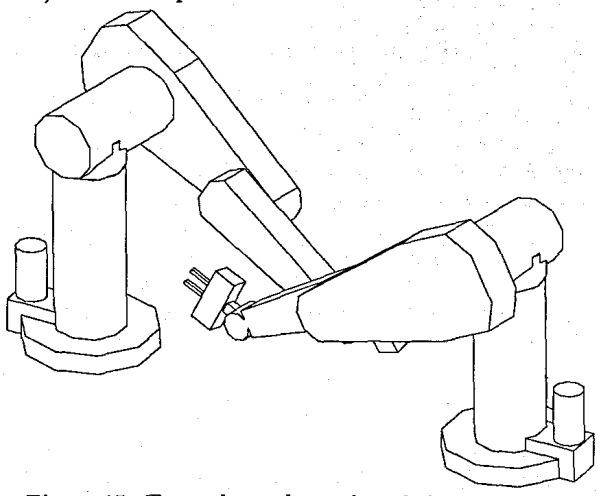


Figure 15: Two robots almost interfering

- If the obstacle is small we will use a potential-field approach to attempt to define a detour motion around it (Khatib 85); this motion is verified by the path-checker before being accepted.
- If the obstacle is larger the system will use a C-space approach, using a number of two-dimensional C-space maps covering a small number of vehicle orientations (Lozano-Pérez 83; Lozano-Pérez 85).
- If the route is blocked the vehicle will try to backtrack to find another route.

To perform collision detection we will use the routines already built into the ROBMOD system (Cameron 85; Cameron 84). These routines have been optimised to perform intersection tests using S-bounds, which is a simple method to reason about the bounding volumes. (Cameron 87). As an example, figure 15 shows two robots that almost interfere. The S-bounds system generates an initial set of spatial bounds for the various parts of the model, and then refines this set to reduce the volume that needs to be searched for interferences (figure 16).

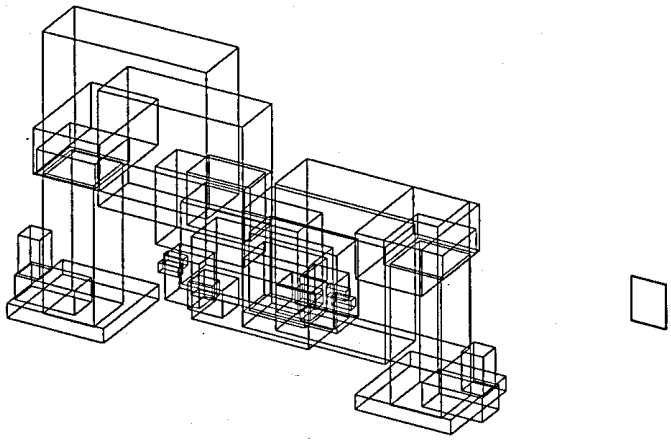


Figure 16: S-bounds in action. The initial spatial bounds for the two robots (left) are reduced to a single, small bound (right).

Object Acquisition

The purpose of the object acquisition experiment is to introduce the AGV into a space into which a number of loaded pallets have been positioned in an irregular manner. The AGV will have a fork-lift attachment, and has to identify the pallets, compute their orientations, and plan how the acquire the pallets using the fork-lift. In doing so it must take into account the positions of other objects and pallets in the area in order to avoid collisions. The path planning required in this case is thus of a different calibre from that required for obstacle avoidance, as it is necessary for the forks of the vehicle to come into close proximity with other objects. However, the class of objects that has to be tackled is restricted—namely, in the first instance, to pallets. Thus our approach is to use simple skeletonised plans to propose paths for the vehicle, which are then tested for validity. This will clearly work in simple cases; the challenge will come in getting the system to work well in relatively cluttered cases.

Three Dimensional Modelling

To test the system we will require realistic, three-dimensional models of the AGV and its environment. Such support will be provided by the ROBMOD modelling system (Cameron and Aylett 87). Developed at Edinburgh University ROBMOD is a Constructive Solid Geometry based system (Requicha and Voelcker 82), which can also produce boundary information and visibility maps for vision work. Figure 17 shows a ROBMOD model of our laboratory. Figure 18 shows another view of the laboratory, but this time from within the room. ROBMOD also provides support for many of the basic geometric functions required by the geometric reasoning system, such as collision detection, and distance computations.

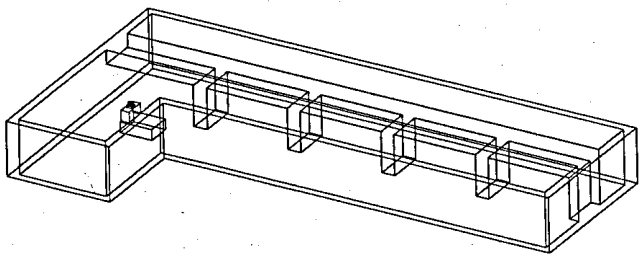


Figure 17: Our AGV Laboratory

Sensor integration

The AGV described will operate in an unknown, but relatively structured environment. A number of different sensors will be used to provide the robot with navigation and part acquisition information. To make effective use of the observations provided by these sensors, it is important that we develop techniques to integrate the available information.

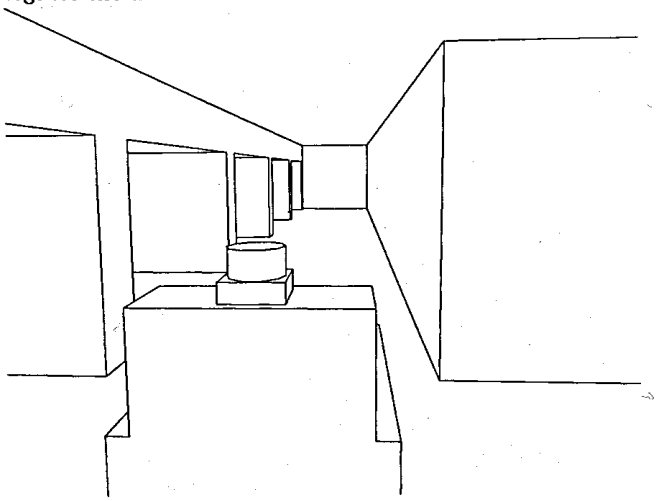


Figure 18: Interior View of the AGV Laboratory

We recall that the AGV will be equipped with stereo-vision, a sonar array, a laser scanner, and proximity switches. These sensors will be used to locate and navigate the robot, provide environment information for path planning, and obtain object information for the acquisition stage. In each of these tasks, different sensors will provide different information, which must be combined to provide a full and complete description of the task domain. The observations made by the different sensors will always be uncertain, usually partial, occasionally spurious or incorrect and often geographically or geometrically incomparable with other sensor views. It is the goal of the integration processes to combine inbformation from all these different sources into a robust and consistent description of the environment

We will consider the environment in terms of geometry and describe all locations and features in terms of parameterized functions. To operate efficiently, the robot system must be able to represent, account for, and reason with the effects of uncertainty in these geometries in a consistent manner. We will describe uncertainty in the environment in terms of a probability distribution defined on the parameter vectors of feature description functions. A description of the environment will be developed in terms of a network of uncertain geometric features. Techniques for manipulating, transforming and combining these stochastic geometric descriptions have been developed (Durrant-Whyte 87c), and will be used as a basis for integrating sensor information.

A general model of sensor characteristics will be used to describe the dependence of sensor observations on the state of the environment, the state of the sensor itself, and other sensor observations or decisions. This sensor model describes a sensor in terms of it's ability to extract uncertain geometric descriptions of the environment (Durrant-Whyte 87b). A constrained Bayesian decision procedure will be used to cluster and integrate sparse, partial, uncertain observations from these diverse sensor systems (Durrant-Whyte 87a).

Acknowledgements

The project described in this paper is a collaborative effort between Oxford University and a number of industrial companies, and is supported by the ACME (Applications of Computers in Manufacturing Engineering) Directorate of the UK Science and Engineering Research Council. We particularly thank our collaborators, Alan Davies, Mike Robbins, and Malcolm Roberts of GEC's FAST Division who designed and developed the GEC AGV. We thank our other industrial collaborators for their continuing contributions to our work: Tony Williams (Thorn--EMI), Roger Hake (IBM), David May (Inmos), Peter Bateman and John Waddington (RARDE), and Jack Betteridge (BP). The Oxford AGV project has benefited substantially from the active involvement of members of the ACME Directorate: Peter Smith, Guy Richards, and Bill Hillier.

References

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

H.B. Barlow, 1982. What causes trichromacy?: a theoretical analysis using comb filtered spectra. *Vision Research*, 22:635–643.

S. T. Barnard and W. B. Thompson, 1982. Disparity analysis of images. *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-2:333–340.

V. Berzins, 1984. Accuracy of Laplacian Edge Detectors. *Comput. Graphics Image Processing*, 27:195–210.

A. Blake and A. Zisserman, 1987. *Visual Reconstruction.* MIT Press, Cambridge, Mass.

J.M. Brady, 1987. Seeds of Perception. In *Proc. Alvey Vis. Conf.*, page , Cambridge, UK.

Michael Brady and P. J. Hopkins, 1987. *Disparity curvature.* Technical Report, Oxford University (in preparation).

B. F. Buxton, 1987. *Notes on computer vision.* Technical Report, GEC Hirst Research Laboratory.

S. A. Cameron, 1984. *Modelling Solids in Motion.* PhD thesis, University of Edinburgh. Available from the Department of Artificial Intelligence.

S. A. Cameron, 1985 (March 25–28). A study of the clash detection problem in robotics. In *Int. Conf. Robotics and Automation*, pages 488–493, St. Louis.

S. A. Cameron, 1987. Efficient intersection tests for objects defined constructively. Submitted to Int. J. Robotics Research.

S. A. Cameron and J. Aylett, 1987. ROBMOD *Users Guide.* Software Report, Department of Artificial Intelligence, University of Edinburgh (U.K.).

J.F. Canny, 1983. *Finding Edges and Lines.* Technical Report Tech.Rep. 720, Massachusetts Inst. Technol.

J.J. Clark, 1986. Authenticating Edges produced by Zero Crossing Algorithms. submitted to PAMI.

L. S. Davis, Z. Wu, and H. Sun, 1983. Contour based motion estimation. *Comput. Graphics Image Processing*, 23:313–326.

N. Daw, 1972. Color-coded cells in goldfish, cat and rhesus monkey. *Investigative Opthalmology*, 11:411–417.

L. Dreschler and H. -H. Nagel, 1982. Volumetric model and 3-d trajectory of a moving car derived from monocular tv frame sequences of a street scene. *Comput. Graphics Image Processing*, 20:199–228.

H. F. Durrant-Whyte, 1987a. Consistent Integration and Propagation of Disparate Sensor Observations. *Int. J. Robotics Research*, 6.

H.F. Durrant-Whyte, 1987b. *Integration, Coordination and Control of Multi-Sensor Robot Systems.* Kluwer, Boston.

H.F. Durrant-Whyte, 1987c. Uncertain Geometry in Robotics. In *Proc. IEEE Int. Conf. Robotics and Automation*, page 851, Raleigh, NC.

Margaret M. Fleck, 1987 (). The phantom edge finder. In *Proceedings of the Alvey Vision Conference*, Cambridge, England.

David Forsyth, 1987. The use of colour in vision. In *Proc. Alvey Vis. Conf.*, Cambridge, UK.

D. B. Gennery, 1979. Stereo camera calibration. In *Proc. Image Understanding Workshop*, page , Science Applications Inc., Silver Springs, Md.

R. Gershon, 1985. Empirical results with a model of color vision. In *Proc. CVPR*, San Francisco.

Shaogang Gong, 1987. *Parallel computation of visual motion.* Technical Report , Oxford University MSc report.

W. Eric L. Grimson and Tomás Lozano-Pérez, 1986 (). Search and sensing strategies for recognition and localization of two- and three- dimensional objects. In *Third Symp. Robotics Research*, pages 73–82, Gouvieux, France.

Marsha Jo Hannah, 1980. Bootstrap stereo. In *Proc. Image Understanding Workshop*, page , Science Applications Inc., Silver Springs, Md.

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

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

R.M. Haralick, L.T. Watson, and T.J. Laffey, 1983. The Topographic Primal Sketch. *International Journal of Robotics Research*, 2(1):50–72.

E. C. Hildreth, 1984. *The Measurement of Visual Motion.* MIT Press, Cambridge, Ma.

D. D. Hoffman and W. A. Richards, 1982. Representing smooth plane curves for recognition: implications for figure-ground reversal. In *Proc. Nat. Conf. Artif. Intell.*, pages 5–8, Pittsburgh, Pa.

R. H. Hollier, 1986. *Automated Guided Vehicle Systems.* IFS, London.

B. K. P. Horn, 1986. *Robot Vision.* MIT Press, Cambridge, Ma.

D. P. Huttenlocher and S. Ullman, 1987 (June). Object recognition using alignment. In *Proceedings of the First International Conf. Comp. Vision.*, pages 102–111, London, England.

L. D. Landau and E. M. Lifschitz, 1959. *Fluid Mechanics.* Pergamon Press, Oxford.

M.S. Livingstone and D.H. Hubel, 1984. Anatomy and Physiology of a color system in the primate visual cortex. *J. Neuroscience*, 4:309–356.

T. Lozano-Pérez, 1983 (February). Spatial planning—a configuration space approach. *IEEE Transactions on Computers*, 108–120.

T. Lozano-Pérez, 1985. Motion planning for simple robot manipulators. In *3rd Int. Sym. Rob. Res.*, Gouvieux.

R. Machuca and K. Philips, 1983. Applications of vector fields to image processing. *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-5.

D. Marr, 1977. Analysis of occluding contour. *Proc. R. Soc. London*, B 197:441–475.

D. Marr and E. C. Hildreth, 1980. Theory of edge detection. *Proc. Roy. Soc. London*, B207:187–217.

H. P. Moravec, 1977. Towards automatic visual obstacle avoidance. In *Proc. Int. Jt. Conf. Artif. Intell.*, page 584, Cambridge, Ma.

D. W. Murray, D. A. Castelow, and B. F. Buxton, 1987. From an image sequence to a recognised polyhedral object. In *Proc. Alvey Vis. Conf.*, Cambridge, UK.

H. -H. Nagel, 1986. Image sequences– ten (octal) years from phenomenology towards a theoretical foundation. In *Proc. 8th Int. Conf. Patt. Recog.*, page , Paris.

H. -H. Nagel, 1987. On the estimation of optical flow. *Artificial Intelligence*, (to appear):.

H. -H. Nagel and W. Enkelmann, 1986. An investigation of smoothness constraints for the estimation of displacement vector fields from image segments. *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-8:565–593.

R. Nevatia, 1977. A Color edge detector and its use in scene segmentation. *IEEE Trans. Sys. Man and Cyb.*, SMC-7:820–826.

J.A. Noble, 1987. The Geometric Structure of Images. M.Sc. report.

I. Page, 1983. DisArray : A Graphics-Oriented Fifth Generation Workstation. In *Nicograph '83*, Tokyo.

Ian Page, 1987. The DisPuter. In *Parallel Architectures for Computer Vision (Page, Ian ed.)*, page , Oxford University Press.

Don E. Pearson and John A. Robinson, 1985. Visual Communication at Very Low Data Rates. *Proc. IEEE*, 73:795–812.

J. Ponce and M. Brady, 1987. Towards a surface primal sketch. In *Three-dimensional vision (Kanade ed.).*

G. T. Reid, R. C. Rixon, and H. I. Messer, 1984. Absolute and comparative measurements of three-dimensional shape by phase measuring Moire topography. *Optics and Laser technology.*

A. A. G. Requicha and H. B. Voelcker, 1982 (March). Solid modeling: a historical summary and contemporary assessment. *IEEE Computer Graphics and Applications*, 2.

Reddaway S.F., 1973. DAP - A Distributed Array Processor. In *First Annual Symposium on Computer Architecture*, Gainsville, Fla.

B. Steer, 1985. *Navigation for the guidance of a mobile robot.* PhD thesis, Warwick University.

Fridtjof Stein, 1987. *Recognition of overlapped objects.* Technical Report OU-RRG-87-11, Oxford University Robotics Research Group.

D. Terzopoulos, 1983. The role of constraints and discontinuities in visible-surface reconstruction. In *Proc. 7th Int. J. Conf. Artif. Intell.*, pages 1073–1077, Karlsrühe.

J. L. Turney, T. N. Mudge, and R. A. Volz, 1985. Recognizing partially occluded parts. *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-7:410–421.

R.J. Watt and M.J. Morgan, 1987. A theory of the primitive spatial code in human vision. *Vision Research.*

Publisher's note: This format is intended to reduce the cost of publishing certain works and to shorten the gap between editorial preparation and publication. The text has been photographed from copy prepared by the authors.

Printed and bound by Halliday Lithograph in the United States of America.

Library of Congress Cataloging-in-Publication Data

Robotics research.

(The MIT Press series in artificial intelligence)
Papers of the Fourth International Symposium on Robotics Research, held at the University of California at Santa Cruz on Aug. 14, 1987; sponsored by the National Science Foundation and the System Development Foundation.
1. Robotics—Research—Congresses. I. Bolles, Robert C. II. Roth, Bernard. III. International Symposium on Robotics Research (4th: 1987: University of California at Santa Cruz) IV. National Science Foundation (U.S.) V. System Development Corporation (Palo Alto, Calif.) VI. Series.
TJ210.3.R636 1988 629.8'92 88-629
ISBN 0-262-02272-9

MEISTER: A Model Enhanced Intelligent and Skillful Teleoperational Robot System 155
Tomomasa Sato and Shigeoki Hirai

Advanced Teleoperation with Configuration Differing Bilateral Master-Slave System 163
Tatsuo Arai, Satoshi Hashino, Eiji Nakano, and Kazuo Tani

IV
KINEMATICS AND FORCES

Introduction 173

Series-Parallel Dualities in Actively Coordinated Mechanisms 175
Kenneth J. Waldron and Kenneth H. Hunt

Whole Arm Manipulation 183
Kenneth Salisbury

Automatic Kinematic Calibration Using a Motion Tracking System 191
John M. Hollerbach and David J. Bennett

Grasping as a Contact Sport 199
Mark Cutkosky, Prasad Akella, Robert Howe, and Imin Kao

Simulation Analysis for Force Control Six-Joint Manipulator 207
T. Kuno, M. Koide, N. Mimura, and H. Miyaguchi

V
RECOGNITION

Introduction 219

Four Steps toward General-Purpose Robot Vision 221
David G. Lowe

Components of a Stereo Vision System 229
S. B. Pollard, J. Porrill, T. P. Pridmore, J. E. W. Mayhew, and J. P. Frisby

Perception via Manipulation 237
Ruzena Bajcsy and Constantinos Tsikos

On the Recognition of Parameterized Objects 245
W. Eric L. Grimson

Modeling Sensor Performance for Model-Based Vision 255
Katsushi Ikeuchi and Takeo Kanade

Generic Surface Interpretation: Observability Model 265
Thomas O. Binford

VI
MOTION

Introduction 275

Self Calibration of Motion and Stereo Vision for Mobile Robots 277
Rodney A. Brooks, Anita M. Flynn, and Thomas Marill

A New Technique for Fully Autonomous and Efficient 3D Robotics Hand-Eye Calibration 287
Roger Y. Tsai and Reimer K. Lenz

Model-Based Motion Analysis—Motion from Motion 299
D. W. Thompson and J. L. Mundy

4 D-Dynamic Scene Analysis with Integral Spatio-Temporal Models 311
E. D. Dickmanns

Location of a Robot using Sparse Visual Information 319
Kokichi Sugihara

The Cycle of Uncertainty and Constraint in Robot Perception 327
Larry Matthies and Takeo Kanade

Maintaining Representations of the Environment of a Mobile Robot 337
Nicholas Ayache and Olivier D. Faugeras

Motor and Spatial Aspects in Artificial Vision 351
G. Sandini, P. Morasso, and M. Tistarelli

Progress toward a System That Can Acquire Pallets and Clean Warehouses 359
Michael Brady, Stephen Cameron, Hugh Durrant-Whyte, Margaret Fleck, David Forsyth, Alison Noble, and Ian Page