Driving Vision by Topology

Charlie Rothwell, Joe Mundy, Bill Hoffman, Van-Duc Nguyen

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Driving Vision by Topology

Charlie ROTHWELL

Joe MUNDY

Bill HOFFMAN

Van-Duc NGUYEN

N° 2444

Decembre 1994

PROGRAMME 4

apport de recherche

ISSN 0249-6399

INRIA
SOPHIA ANTIPOLIS

Driving Vision by Topology

Charlie ROTHWELL*
Joe MUNDY**
Bill HOFFMAN**
Van-Duc NGUYEN**

Programme 4 — Robotique, image et vision
Projet Robotvis

Rapport de recherche n ° 2444 — Decembre 1994 — 29 pages

Abstract: Recently, vision research has centred on both the extraction and organization of geometric features, and on geometric relations. It is largely assumed that topological structure, that is linked edgel chains and junctions, cannot be extracted reliably from image intensity data. In this paper we demonstrate that this view is overly pessimistic and that visual tasks, such as perceptual grouping, can be carried out much more efficiently and reliably if well-formed topological structures are available.

The widespread assumption that edge detectors produce incomplete and erroneous topological relations, such as the image projection of polyhedral face-edge-vertex structures, is shown to be false by analyzing the causes for failure in traditional edge detectors. These deficiencies can largely be overcome, and we show that a good compromise between topological completeness and geometric accuracy can be achieved.

Furthermore, edge detection should not be carried out in isolation. The resulting topological and geometric descriptions should be motivated by the subsequent representations to be built on such image features. Here, algorithm development is driven by general object recognition, which provides an objective evaluation of the resulting edgel structure. In effect, we test the design by examining the results of the entire recognition process. Although unconventional, we feel that visual algorithms can be developed more successfully when considered as part of a complete system.

Key-words: Low level processing, Segmentation and grouping, Systems and applications

(Résumé : tsvp)

*INRIA, Sophia Antipolis.
**GE CRD, Schenectady, NY.

Unité de recherche INRIA Sophia-Antipolis
2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex (France)
Téléphone : (33) 93 65 77 77 – Télécopie : (33) 93 65 77 65

Vision Fondée sur la Topologie

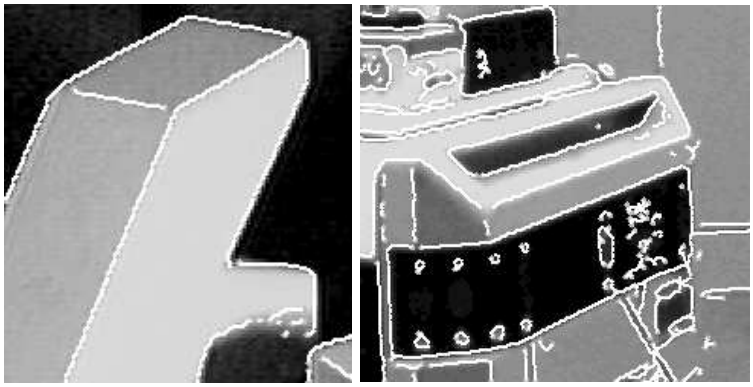
Résumé : Jusqu'à présent, la recherche en vision a été principalement axée sur l'extraction de primitives géométriques, ces primitives étant regroupées par la suite à l'aide de relations géométriques. Il est généralement admis que la structure topologique des images, basée sur les chaînes de contours et les jonctions, est difficile à extraire de manière fiable à partir d'images de niveaux de gris . Dans ce rapport, nous démontrons que ce point de vue est trop pessimiste et qu'il est en fait possible d'obtenir cette information. De plus, nous montrons comment certaines tâches visuelles, telles que le groupement perceptuel, peuvent être exécutées avec beaucoup plus d'efficacité et de fiabilité dès lors que ces structures topologiques correctes sont connues.

Nous montrons que l'hypothèse courante selon laquelle les détecteurs de contours produisent des relations topologiques incomplètes est erronée et fausse. Par exemple, nous montrons qu'il est possible de récuperer la structure topologique de l'image d'un polyèdre (relations entre sommets, arêtes et faces). Pour cela nous analysons les raisons pour lesquelles les détecteurs de contours classiques peuvent échouer. Dans une large mesure, ces problèmes peuvent être résolus, et nous montrons qu'un bon compromis entre la complétude topologique et la précision géométrique peut être atteint.

Qui plus est, la détection de contours ne doit pas être effectuée indépendamment du but recherché. Les descriptions topologiques et géométriques produites doivent être motivées par les représentations qu'elles vont permettre de construire. L'algorithme présenté ici est orienté vers la réalisation d'un système général de reconnaissance d'objets. Cela nous fournit un critère objectif d'évaluation des résultats obtenus: nous testons l'algorithme en examinant les résultats du processus de reconnaissance complet. Nous pensons, en effet, que les résultats des algorithmes de vision peuvent être mieux interprétés lorsque les algorithmes sont considérés comme faisant partie d'un système complet (cela n'est malheureusement pas une pratique usuelle en vision par ordinateur).

Mots-clé : traitement de bas niveau, segmentation et groupement, systèmes et applications

Contents



a b

Figure 1: *Two simple scenes in which a Canny edgel detector fails to extract the image topology correctly. Higher level processing (which is time consuming) would have to be used to join-up each junction and corner that has been broken.*

1 Introduction

It is generally assumed that the boundaries of objects in intensity images are such that each point on a boundary corresponds to a step discontinuity in the projected intensity. These points or *edgels*[1] are located both where the local intensity surface changes significantly over a short distance, and where the change is directed along a prominent direction. In the vicinity of junctions, where a number of discontinuity edges meet, the intensity behavior is more complex and is not well-modeled by a simple edgel event.

Figure 1 shows very clearly the failure of a traditional edgel detector such as the Canny operator [8]. For a polyhedral object we find that trihedral junctions are frequently missed and sometimes even 'L' junctions are not recovered (such as the one lower left of the centre in Fig. 1a). Any higher level program that uses the edgel detector output shown in the figure would have to patch-up the gaps in the edgel chains using top-down processing; this can be expensive and a burden on processing time as discussed by Binford [5] and Lowe [23]. Similarly, Fig. 1b shows the Canny edgel-map of the front end of a robot. Again note that the edgel detector is unable to provide the complete boundary of single object faces. In this paper we describe an edgel detector that is designed to recover sound topological descriptions. This goal is in contrast to the motivations for earlier edge detectors, such as those of Canny [8] or Deriche [9], which emphasize localization and noise immunity. The design of the edgel detector is guided by these premises:

[1] We use the term *edgel* to represent a single high contrast gradient point to avoid confusion with the topological object called an *edge*. This terminology is standard, and is analogous to the use of the word *pixel*.

1. Geometric information, that is edgel location, can be accurately recovered since the actual noise level in modern image acquisition devices is negligible. Smoothing is required only to account for image quantization.

2. Curve fitting is mature and provides reliable geometric descriptions of edgel chains. Edgel chains provide adequate descriptions of curved boundaries which do not admit a functional description.

3. The recovery of scene topology is essential to recognition and other visual tasks. Incomplete topology prevents reliable extraction of object boundaries and subsequent failure of recognition and other visual tasks. The issue is not just computational efficiency but intractability of grouping on fragmented boundary structures, even if these structures are geometrically accurate.

It is clear that accurate geometric information can be reliably extracted from intensity images as demonstrated by the assortment of working vision systems that have appeared over recent years in the literature [1, 2, 14, 15, 25, 31, 39]. These systems generally assume that only fragmentary topological structure is available. For example, two adjacent boundaries meeting at a vertex is considered to be a major feature discovery. By carrying out exhaustive grouping it is possible to align specific 3D models with set of corresponding image features and achieve reliable recognition.

However, the computational cost of grouping and indexing on fragmentary geometry becomes intractable for the recognition of generic curved objects in complex scenes. As demonstrated very clearly in the recent book by Grimson [15], geometry is not a sufficiently complete description of the scene for a task such as recognition. Sets of points or line segments such as those represented in Fig. 1 do not provide enough structure to support multiple object recognition in complex scenes. On the other hand, the use of simple topological cues such as nearest-neighbor connectivity for grouping (Huttenlocher [22]), can lead to much more viable and efficient solutions.

A great deal of prior literature has centred on the problem of constructing optimal edgel, corner, and other types of feature detectors. In these works, the definition of optimal is usually framed from a signal processing point of view, where signal to noise ratio performance and localization are key issues. Indeed great advances have been made since the days of the Roberts [33], Hueckel [19, 20], and Beaudet [3] operators. The progress has occurred in the reliability of detection and in the accuracy of edgel localization. Although the list is far from exhaustive, we feel that significant advances in edgel detection have been made through the works of [7, 9, 16, 24, 26, 29, 41], and have brought us to the current standard of edgel extraction.

In our view, this line of development is now mature. The central issue for future development in intensity event detection is completeness of description rather than noise or localization performance. Illustrative of this new emphasis is the growing use of snakes and deformable templates which strive to fit a complete geometric and topological description directly to the intensity data [4, 38]. However, it remains to be seen that these global ap-

proaches can overcome the combinatorial complexity of structural configurations and local minima in the snake optimization cost function.

Another category of intensity analysis, those based on morphological methods, are designed to produce correct grey-level intensity-image topology which is motivated by the belief that connected components over binary image regions represent a valid topology. Such approaches are reported by Haralick [17] and Noble [28]. However, what we really need is correct *object* topology. There is no reason to believe that such methods will be superior to linear edge detectors with improved topological performance.

The goal of this paper is thus to raise the importance of topological constraints up to the same level previously held by geometry, and then to employ these constraints as additional optimality criteria for edgel detection. In doing this, we demonstrate that significant visual tasks can be solved before considering the geometry underlying the image observations. However, we shall endeavour (and in fact ensure) that no geometric information is lost which could have been made available through the use of more traditional approaches to edgel detection.

In Section 2 we discuss why many previous approaches to edgel detection have failed to extract topology correctly. From these observations we have been able to design an improved edgel detector that is described in detail in Section 3. In Section 4 we give some examples comparing a traditional implementation (the Canny operator [8]) with the new algorithm. Finally in Section 5 we demonstrate how the new edgel detector can simplify a task such as the detection of polyhedra in images.

2 Why is topology incomplete?

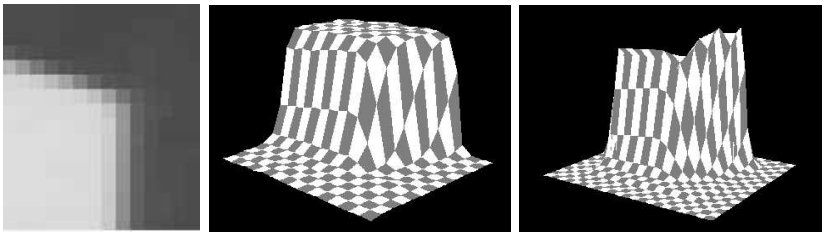
Given that edgel detection has been such an active area of research, and thus there exists a wide variety of methods available for detecting edgels, we should question why the majority fail to extract scene topology accurately. It has been noted that the edgel detectors reliant on the computation of directional first derivatives (such as the Canny [8] or Deriche [9] operators) fail most often at junctions and in areas of high contour[2] curvature.

2.1 The nature of the gradient surface

In conventional edge detectors, the intensity image I is filtered with a two dimensional Gaussian kernel (or some similar approximation [8, 9]) to produce a smoothed continuous image S. We then take direction derivatives using first differences to produce:

$$S_x = \frac{\partial S_\sigma}{\partial x} \quad \text{and} \quad S_y = \frac{\partial S_\sigma}{\partial y}.$$

[2] We use the term *contour* to mean the locus of edgels in the image.



a **b** **c**

Figure 2: *For a simple image corner we find that the boundary topology implied by $|\nabla S|$ is complete. (b) is a plot of the intensity surface I for (a). (c) shows that there is still a slight reduction in $|\nabla S|$ at the corner, though the effect is small.*

We could derive a richer description of I by employing steerable filters as suggested by Freeman and Adelson [13], though for the purposes of this paper we restrict our attention to derivatives along the x and y axes. These are sufficient to illustrate the major source of edgel detection failure at junctions.

From S_x and S_y we compute the information used to classify edgels:

$$|\nabla S| = \sqrt{S_x^2 + S_y^2} \qquad \text{and} \qquad \theta = \arctan\left(\frac{S_y}{S_x}\right). \tag{1}$$

A real image of a binary corner is shown in Fig. 2. We see quite clearly that the surface describing $|\nabla S|$ is generally well-formed, having only minor degradations in edgel strength around junctions and corners. Of course there are ridges and spikes produced by what may be called *noise*, though in practice we find that these are usually commensurate to observable scene features such as specularities, texture or other markings.

Although camera noise does exist in the form of an additive (or multiplicative) error on the intensity values, or even as a systematic interference pattern, for most modern cameras we find that its effects are negligible compared to scene features. Thus, the computed gradient surface corresponds very closely to the contrast changes of the original intensity surface. Consequently, all *significant* ridges in $|\nabla S|$ should be interpreted as being geometrically interesting. We now show why boundaries in the neighbourhood of junctions are not extracted reliably.

2.2 Edgel orientation

Because of spatial quantization in the image, the normal to the edgel direction θ is often unreliable. Ideally this normal is equal to the direction of maximal change of gradient of I. Usually we can measure the orientation at an edgel to within only ± 15 degrees of the correct

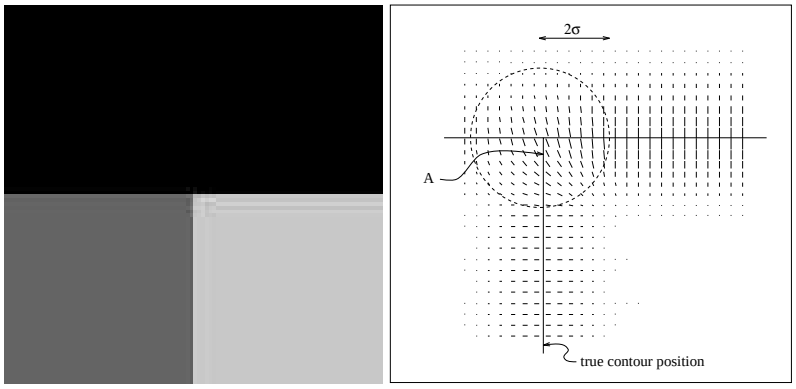


Figure 3: *(b) shows a needle plot of the strengths and orientation response of a first order edgel operator at each pixel applied to the central part of image (a) for a smoothing kernel of* $\sigma = 3$*. Note that the orientations are unreliable within a circle of radius* 2σ *of the junction, and at 'A' are nearly 70 degrees off.*

value.[3] This bound on the accuracy applies only along straight or low curvature portions of object boundaries. The situation is far worse near corners and junctions where the computed orientation bears little relation to the actual orientation.

Figure 3 demonstrates the problem for a synthetic grey level image representing a trihedral junction; to emphasize the range of the effect we have used a smoothing kernel with $\sigma = 3$. At each pixel the edgel strengths are represented as needles of length proportional to $|\nabla S|$ and oriented along the estimated edge normals. We find that the values of θ are noticeably unreliable within a radius of 2σ of the corner. Most importantly, observe that around region 'A' in the image the orientation is out by almost 70 degrees.

The unreliability in the measurement of θ is a real concern because it has traditionally played a significant role in the classification of edgels. Ignoring the hysteresis process (which further filters out weak edgels [8]), pixels have previously been classified as coinciding with edgel locations if:

1. $|\nabla S|$ is above some threshold, say t_{low};
2. $|\nabla S|$ is greater than local estimates of the gradient in the direction $\pm\theta$. These values are called $|\nabla S_{+\theta}|$ and $|\nabla S_{-\theta}|$.

The second of these processes in called *non-maximum suppression* and attempts to enforce that all of the edgel points are ridge points of the surface $|\nabla S|$ (up to the digitization).

[3] Obviously if we integrate the orientation information over a number of proximal pixels we can do much better than this.

This step is excessively reliant on the estimates of θ, and so often fails to mark edgels at junctions, corners, and even on some smooth curve portions where the contrast changes are too poorly defined.

We emphasize the fact that edgel detectors do not have *significantly diminished* responses near corners. Certainly the value of the norm of the gradient is reduced locally, but as is clear from the Fig. 2 that all of the points that we would label as edgels are locally maximal in some direction. Our key claim is that the information which is required to identify object contours is largely present in the norm of the gradient of the intensity $|\nabla S|$. Algorithms based on a step edge intensity model fail because they do not exploit the available information.

2.3 Effects of quantization and smoothing

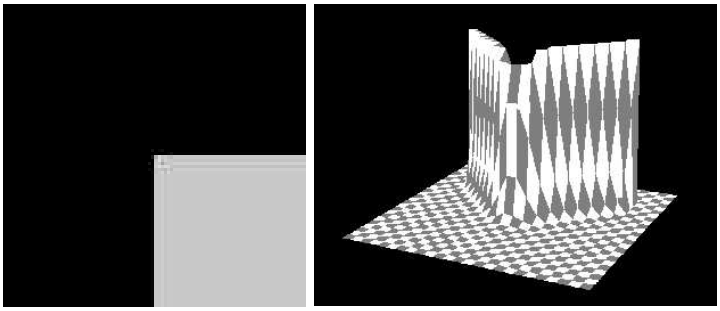
Additional topological disturbances arise from the smoothing process used to suppress higher frequency noise components in the intensity process. The smoothing means that interference arises between proximal but distinct intensity events such as two contours coming into a junction; the smoothing usually causes a reduction in the strength of the directional derivatives we compute. If we are not careful we can interpret the weakening of the edgel strengths along the contours as marking the termination of the feature.

The range of the effect varies with the size of the smoothing kernel used (which we consider to be an isotropic Gaussian of standard deviation σ). Even without smoothing we are unable to make differential measurements at a single point, but must consider collections of pixels. This is because we based our computations on the evaluation of finite differences that spread over a finite range; for a simple first difference operator [-1,1] we consider pairs of neighbouring pixels, or for the central difference operator [-1,0,1] we use triples of points. As soon as we introduce smoothing the range of the diminution effect will extend up to a circle of diameter 3σ.

The effects of edgel strength weakening has been studied in significant detail for the case of *continuous* intensity surfaces [9]. In Fig. 4 we demonstrate the presence of the effect for a synthetic *discrete* intensity surfaces which is based around an ideal two dimensional corner (a binary image). The effect is shown for a real image of a binary corner in Fig. 2. From the two figures we see that the effect is present, but that it is not all that significant. We emphasize that analyses for continuous surfaces are not always totally representative of those for the discrete domain, even though the quantization process may be considered simply to be a sampling of a continuous analytic surface. As an example of the effects of the quantization process (that is one of the features that distinguishes the continuous and discrete domains), we demonstrate in Fig. 5 for a synthetic case how the norm of the gradient response changes with the contour orientation when we use first order directional derivatives aligned with the x and y axes.

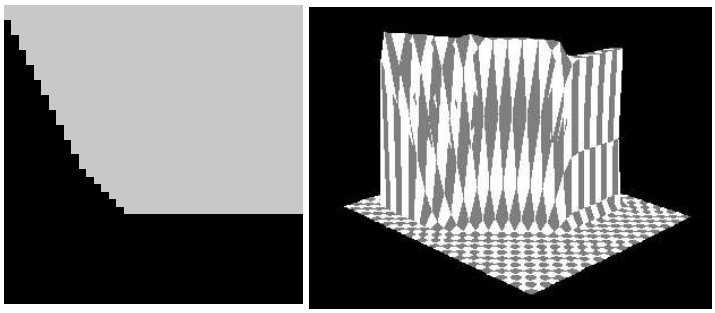
2.4 Summary

In conclusion, we infer that the failure of the Canny and similar algorithms actually result from unreliable orientation data making non-maximal suppression results erroneous. This



a **b**

Figure 4: *For a discrete synthetic image, composed of just two grey levels, we find a reduction in the edgel magnitude in the vicinity of the corner. In (b) each square represents a pixel. The smoothing kernel is based on a Gaussian of* $\sigma = 1$*, and the norm of the gradient* $|\nabla S|$ *is plotted.*



a **b**

Figure 5: *The quantization present in digital images causes a variation in edgel strength for even for cases in which we would really expect equal gradients.*

observation is not new, though surprisingly seems to have been ignored in most recent approaches which attempt to improve edgel detector response. Du Li, *et al.* [11] made exactly the same observation and demonstrated how the orientation information in the neighbourhood of junctions actually becomes mis-aligned. Although we use a different process to recover the lost edgels, we are in precise agreement with their philosophy in building up information from images. The following quotation is drawn from [11], though with the emphasis added by us:

> The Canny operator has previously attracted much critical evaluation, e.g. [27, 28, 30, 32, 36]. A few studies have considered the failure of connectivity at junctions in detail. Noble [28] criticized this behaviour of the Canny operator at junctions and proposed a completely different approach, based on mathematical morphology to overcome the problem. *It is more common to assume that such failures can be corrected at a later stage of analysis, by use of higher level knowledge [32]. The latter approach inevitably makes the search at the line grouping stages of analysis more ambiguous. This uncertainty can be avoided if the junctions are detected properly.*

This desire is very much at odds with one of the current streams of thought regarding the extraction of information from images. Principally, it is believed that edgel detectors do not provide useful information around corners and junctions, and so we should use more finely tuned methods such as corner detectors to locate them instead [34, 10]. Although we certainly agree that a trade-off is required between top-down and bottom-up processing,[4] this approach is less efficient as we already have seen that the edgel information is satisfactorily contained within the $|\nabla S|$ image.

3 A topology-based algorithm

In this section we describe in detail the edgel detection algorithm that provides the major contribution of this paper. We base our design constraints around two distinct goals:

1. **Topology:** the topology of the contour description we derive should be as close as possible to the projection of the scene topology. We assume that the 3D scene objects are described by a general curved surface topology and the corresponding image topology is defined by:

   (a) *Vertex:* vertices in the image correspond either to 3D vertices in the scene or occlusion "T" junctions;

[4] We also realise that corner detectors are potentially *far* more accurate at detecting corners than edge detectors and so should be used later as a top-down process once we have a notion of where the corners are. We are currently evaluating this approach.

(b) *Edge:* an edge in the image is either a complete or partially occluded 3D object edge;

(c) *Face:* an enclosed image region, or face, is either a complete or partially occluded projection of a 3D object face.

In addition to these topological features, the existence of surface markings and surface textures introduce additional intensity events which are best described as geometric patterns. However for generic object recognition, it is crucial to derive as complete a topological description of an object's boundary as possible.

2. **Geometry:** the edgels we extract should correspond as closely as possible (in a geometric sense) to the projection of the scene features. Loosely speaking, the geometric constraints cover the traditional optimality criteria of previous investigations [7]:

   (a) *Low error rate:* it is important that edgels which occur in the image should not be missed and that there be no spurious responses;

   (b) *Localization:* the distance between the points marked by the detector and the 'centre' of the true edgel should be minimized.

   (c) *Single response:* each edgel should provide only a single output at a given point in the image.

We now formulate an algorithm which strives to satisfy these two goals in concert. The algorithm consists of the following major steps:

1. Smooth the image I using a two dimensional Gaussian kernel;
2. Compute the directional derivatives S_x and S_y;
3. Derive of the norm of the gradient image $|\nabla S|$ and the orientation estimates at each pixel;
4. Where possible estimate the maximum of $|\nabla S|$ to sub-pixel accuracy;
5. Compute the local edgel contrast thresholds which are used to classify pixels as being either edgels or otherwise merely contained within image regions;
6. A topologically correct thinning of the edgel data sets which re-assigns pixels previously marked as edgels as simply parts of an image region;
7. The extraction of topological primitives and the description of the scene as a topology network.

We describe each step in more detail throughout the following paragraphs.

3.0.1 Image smoothing

The intensity I image is smoothed with a discretely sampled form of the two dimensional isotropic Gaussian distribution:

$$G(x,y) = \frac{1}{\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}. \tag{2}$$

We make use of the separability of the Gaussian kernel and smooth using successive one dimensional kernels in the x and y directions. The size of the discrete convolution kernel is chosen so that the limits of the function are $\delta = 0.015$ of the central value.[5] This means that the dimension of the convolution kernel is:

$$w = 2 \times \left\lceil \sigma\sqrt{-2\ln\delta} \right\rceil + 1.$$

Typically we which to smooth the image with a kernel of $\sigma = 1$, and so $w = 7$ (i.e. we use a pair of 7×1 discrete kernels).

3.0.2 Computation of the direction derivatives

S_x and S_y are measured using central finite difference operators of the form $[-1, 0, 1]$ and $[-1, 0, 1]^{\mathrm{T}}$. According to Fleck [12], this provides better localization than the non-central operators $[-1, 1]$ and $[-1, 1]^{\mathrm{T}}$.

3.0.3 Measurement of $|\nabla S|$ and θ

$|\nabla S|$ and θ are computed for every point quite using the expressions in Eqn (1). For convenience we denote $|\nabla S|$ by the discrete function N. A quantized value of θ is also computed so that we know in which octant the maximal gradient direction lies:

$$\theta_q = \left\lfloor \frac{4\theta}{\pi} \right\rfloor,$$

that is $\theta_q \in \{0, \ldots, 7\}$.

[5] Canny [8] uses $\delta = 0.001$. We find that using a limit an order of magnitude larger does not detract from the quality of the edgel image. There also is little practical difference between using $\delta = 0.015$ and $\delta = 0.01$ in the resultant edge image, but by the definition that follows it means we can use a kernel of size 7 rather than 9. This makes the convolution faster.

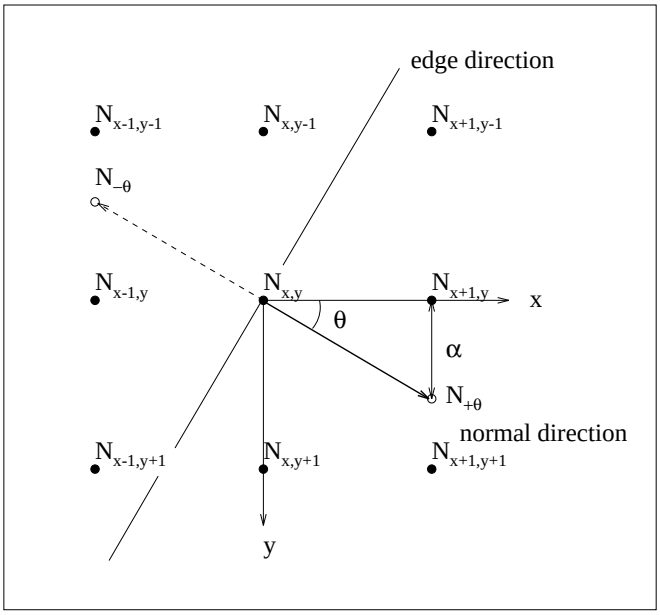


Figure 6: *Sub-pixel localization of edgels in achieved using a quadratic interpolation process based on the estimate of the normal to the edgel direction (this is the direction of maximal gradient in the smoothed intensity surface $N_{x,y}$).*

3.0.4 Sub-pixel localization

Although pixels have not not yet been classified as edgels, we proceed by estimating where a hypothetical edgel would lie for each pixel having $N > t_{low}$ using the sub-pixel interpolation technique suggested by Canny [8]. We can afford to perform this computation for such a large number of image points because a by-product of the process is used for classification.

Figure 6 demonstrates how the sub-pixel interpolation process works around a pixel $N_{x,y}$. The process itself involves both a linear and a quadratic interpolation step. First we compute estimates of the surface gradients $N_{+\theta}$ and $N_{-\theta}$ which lie between known gradient values. The value of $N_{+\theta}$ depends on two values of $N_{i,j}$ where the i and j depend on θ_q; similarly $N_{-\theta}$ depends on two values of the gradient surface. We examine the case for $\theta_q = 0$. The other cases follow quite simply.

Using θ defined by the direction of maximal intensity gradient we set the value $\alpha = \tan\theta$. $N_{+\theta}$ and $N_{-\theta}$ are then defined by linear interpolation between pairs of pixels:

$$N_{+\theta} = \alpha N_{x+1,y+1} + (1-\alpha)N_{x+1,y} \quad \text{and} \quad N_{-\theta} = \alpha N_{x-1,y-1} + (1-\alpha)N_{x-1,y}.$$

It is at this stage of the computation we make use of the current information for the initial classification of pixels as edgels. Generally it is accepted [8, 9] that if $N_{x,y} > N_{+\theta}$ and $N_{x,y} > N_{-\theta}$, then the pixel at (x, y) represents a well defined ridge-point in the function N and is thus an edgel. Previously these points have represented all of the edgels extracted from the image, we consider them to be only a base set Σ_0 on which to build up the topological description of the image.

For points in Σ_0 we define an edgel sub-pixel location by fitting a parabola along the normal direction (as shown in Fig. 6) to the set $\{N_{-\theta}, N_{x,y}, N_{+\theta}\}$. The maximum of the parabola is assumed to coincide with correct location of the edgel.[6] The edgel associated with the point (x, y) is therefore marked at the maximum along the line defined by $\{N_{-\theta}, N_{+\theta}\}$.

We also compute the quadratic interpolation for all points not within Σ_0, and if the maximum of the parabola lies within one pixel of (x, y) we define the edgel position in the same way as done for the good points. If the distance is greater than a pixel we declare that the sub-pixel interpolation process has failed, but still maintain the possibility that the point (x, y) is an actual edgel point. In this case we have to be satisfied with the non-interpolated coordinates should the pixel later be classified as an edgel.

Although we are still fixed to a quantized grid that represents the structure of the image, after this stage in processing the vast majority of edgels have associated with them a geometrically accurate location. This location has been computed using traditional methods that have been shown to have good geometric properties, and so we have ensured that our resultant edgel map will also have a reliable geometry.

3.0.5 Thresholding of N

Once the values of N and θ have been computed, we are principally faced with a classification problem of deciding whether a pixel represents an edgel, that is a point on a topological boundary, or is rather some other less precisely defined image feature. This classification is conventionally carried out by thresholding. Typically, if $N_{x,y}$ is greater than a threshold t_{low} the point (x, y) is marked as a candidate edgel. In effect, we assume that some image discontinuities are too small to be considered either because they correspond to a boundary contrast too weak to be described reliably, or because an intensity fluctuation is present due to image sensor noise. Thresholding usually returns far too many candidate edgels, since we want to be conservative in our choice of threshold. The poorer potential edgel locations in the candidate set are filtered out (if we use the Canny operator) using two other processes:

1. If an edgel point fails to be marked as a ridge during non-maximal suppression it should be rejected. The principal behind this process is that the rejected edgel will *almost certainly* be represented by another proximal one, and so data is not lost. Unfortunately this *thinning* process often removes too many edgels.

[6] We demonstrate in appendix A that the maximum of the parabola coincides with the location of the zero-crossing of the second derivative of the intensity surface. The position of the zero-crossing is often taken to be the correct location of the edgel [24].

2. 8-way or 4-way connectivity are used to define sets of connected edgels. Any set that does not have at least one element with a strength greater than a second threshold t_{high} are removed from consideration (obviously $t_{high} > t_{low}$). This is called *hysteresis*.

The hysteresis filtering process requires further comment. The threshold t_{low} is meant to represent a measure of spurious edgels that are due to either camera noise, or perhaps weak world features in which we have no interest. Therefore, we first designate all edgels stronger than t_{low} as *geometrically interesting features*. We then introduce t_{high} heuristically to change the classification because we are unable to handle the large number of features that get passed on to our higher level visual algorithms.

We agree that there is a basic noise level present in images below which we should disregard responses. Therefore our algorithm still employs a t_{low}. However, we do not make use of a threshold t_{high}. There is no reason to believe that the strength of an event has any particular relevance to its value for a higher level process such as object recognition. It is quite likely that illumination effects and shadows can reduce the contrast of an object boundary to t_{low}, even though the boundary is very significant to further processing. The generally low-contrast interior boundaries of planar polyhedra offer a good example of this problem. On the other hand, many of the other stronger responses in N are due to features in which (from the geometric standpoint) we are not interested; these are due to specularities and from surface textures that do no provide sufficient geometric constraint to be of immediate of use.[7]

We use a novel approach to remove these features given that *a priori* we have no knowledge of which threshold value should be used. The filtering algorithm makes use of dynamic thresholding, that is the threshold is defined by a spatial distribution that varies across the image. We compute a threshold surface (a function called $T(x, y)$ on the discrete image domain) and use it to classify the image pixels by declaring an edgel whenever $N(x, y) > \alpha T(x, y)$. The use of the constant α, for which $0 < \alpha \leq 1$, is described below.

At this stage we revert to the information provided by non-maximum suppression. The edgel strengths of the pixels in the set Σ_0 provide a very reliable indication of how strong edgels should be to be classified correctly (as around all of the members of Σ_0 we were able to measure the local surface geometry accurately). We therefore use these strengths to set up local threshold values within the function T. More formally, we could define the value of T(x,y) to be:

$$\begin{aligned} T(\mathbf{x}) \quad &= \quad N(\mathbf{p}) \ \text{ where } \ \mathbf{p} \in \Sigma_0, \\ &\text{and} \quad \forall \mathbf{q} \in (\Sigma_0 - \{\mathbf{p}\}),\ d(\mathbf{x}, \mathbf{p}) \leq d(\mathbf{x}, \mathbf{q}), \end{aligned}$$

where $d(\mathbf{x}, \mathbf{p})$ represents the Euclidean distance between the pixel location of $\mathbf{x}$ and $\mathbf{p}$. Essentially, this definition sets the local threshold to the strength of the nearest edgel in Σ_0.

[7] It is the case that stronger features produce more accurate geometric descriptions than weaker features. We can however quite simply order our subsequent investigations by a measure of the edgel strengths.

The justification for this definition is that the actual image contrast is predicted suitably well by nearby surface boundaries. Edgels close to junctions are not initially included in Σ_0 purely because of incoherent direction values.

This definition yields a surface T that is not continuous (in a loose discrete sense) as it is composed only of a series of flat plateaux. Within single object regions one would at least hope that the threshold function should change smoothly. Therefore, we would like to define a function T whose continuous equivalent has the following form: take each member $\mathbf{x}$ from Σ_0 and place a point in 3-space corresponding to $\{\mathbf{x}, N(\mathbf{x})\}$. Construct a surface that interpolates between the points in a linear fashion as smoothly as possible (conceptually by dropping a sheet over the points). We then sample the surface to yield $T(\mathbf{x})$.

In fact, we change the surface slightly by remembering that the smoothing kernel tends to cause effects that have a larger influence than at a point. This disrupts the computed gradients unfavourably. We therefore replace each point in Σ_0 by a disk of radius 2σ at a height of $N(\mathbf{x})$, and ensure than the threshold surface contains these disks. We use the lower threshold value wherever the projections onto the xy plane of the disks of two edge points overlap.

We can compute a very good approximation to this function using a morphology based distance transform [18]. Typically, such algorithms return the distance of a point to the nearest edgel location. We in fact use chamfer masks to compute the distances based on an algorithm given by Borgefors [6], though we have made substantial modifications to provide both an estimate of the distance to the nearest *two* pixels within the disks around the edgels, and also to return the value of $N(\mathbf{x})$ at each pixel.[8] Given the distance to two proximal edgels, and their values of $N(\mathbf{x})$, we can then linearly interpolate to define the local value of $T(\mathbf{x})$. The complete algorithm is given in Appendix B.

Thresholding of the image then proceeds by comparing the value of $N(\mathbf{x})$ against the threshold function $T(\mathbf{x})$ and classifying the point $\mathbf{x}$ as an edgel if the former is at least 90% of the latter. We introduce this scaling parameter α to account for the fact that edgel strengths do become slightly diminished close to junctions, and so this enhances our chances of correct classification. All pixels that pass the threshold test are included in the set Σ; obviously $\Sigma_0 \subset \Sigma$.

3.0.6 Edgel thinning

The thresholding process yields an edgel image Σ that has connected subsets. Two elements of Σ are *immediately connected* if they are 8-way neighbours of each other. The widths of the sets are frequently equal to two or three pixels and so do not represent the topology of a digital curve. We thin the edgel sets so that the chains are of unit thickness (in an 8-way connected sense) and in a way which ensures that the topology of the original thick set is preserved.

[8] We ensure that the two nearest pixels are themselves not close to each other, but must come from approximately diametrically distinct regions of the image.

The thinning process is based on the Tsao-Fu thinning algorithm [40] which eliminates elements from a set based on *genus*. Points that would cause a *change in genus* equal to zero on deletion can be removed from the data set without effecting the set topology. The process works in a way that does not shorten edgels chains which have a free end (i.e. edgels that are connected only to one other edgel are not moved during the thinning process).

Tsao-Fu thinning treats all members of the set equally, and so a strong edgel may be deleted in preference to a weaker one. This means that edgel chains can be displaced slightly (often by a full pixel) from their optimal positions. The way to overcome this problem is to order all of the elements in Σ by their edgel strengths (the value of $N(x, y)$). We then examine the genus of weakest edgels first and delete them as necessary. Such an algorithm relies on sorting the edgel set and so can become slow for large numbers of edgels (as sorting is $O(n \ln n)$). Although we have not been hindered by the reduction in speed due to the sorting, one could in principle recover the efficiency loss by tiling the image and processing each tile independently.

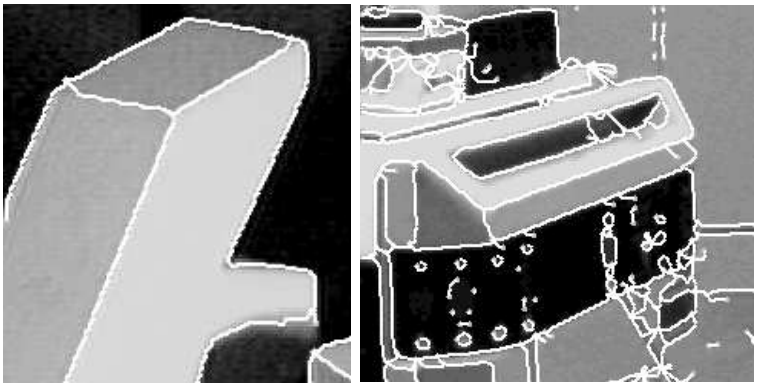
3.0.7 Computation of the topological description

We have now come to the easiest part of the whole process. The phases described above have demonstrated how to derive a topologically reliable edgel description of a scene (when represented as a single grey scale image) where we have also made an effort to extract the correct geometry of the edgels. We now simply have to extract the topological description from the discrete edgel image and associate it with the sub-pixel edgel coordinates which have been have computed.

First, we search the image for all of the vertices that will make up the topological description (described as a vertex-edge-face network). Vertices are placed at edgels that have either only a single neighbour (in which case they represent the end of a dangling edgel chain), or are edgels that have more than two edgels connected to them (these are junctions). Topologically, a corner point defined by the meeting of two edgel chains is contained within a single edge. We do not attempt to *segment* the edges at this point, though the implementation of the topology network does not prevent us from later breaking edges into 1-chains (chains of edges) and hence having vertices lying on only two edges.

Once the vertices have been extracted, we follow the edgel chains between them in the edgel image using an 8-way following algorithm. As each edgel is extracted, its sub-pixel location is written to a list managed by the topological edge of which it is part. This last part means that we are able to maintain the same level of geometric accuracy achieved by a Canny [8] or Deriche [9] type of edgel operator.

It is worth commenting on the implementation of the topological environment. Through object oriented programming (written in C++), we have an object hierarchy allowing not only vertices, edges and faces (and blocks for 3D descriptions), but also n-chains where $n \in \{1, \ldots\}$ for the association of groups of features. Given any one type of element in the hierarchy, we can request information about its *inferiors* and *superiors*, and so discover any topologically pertinent information. For example, we may determine on which face a given vertex lies. We may also change parts of the description without undue effort; for instance,



a **b**

Figure 7: *A comparison of the output of the new algorithm. These results should be compared to those in Fig.1. Note how we have been able to recover a far superior topological description of the edgel image.*

a single edge may be broken into two connected edges and a new vertex created should we feel that such a description is more fitting to the scene in question. The reverse type of process is also feasible (merging of features). Therefore, we can continually update the scene description without any difficulty depending on our current understanding of the data.

4 Examples

Here we show some sample outputs of the algorithm as a comparison to the Canny results given in Fig. 1. The Canny output was tuned to give the best topological results possible, and so we believe that the comparison is fair. Figure 7 shows the improvement in topology around contour junctions. In (a), the topology of all of the junctions are recovered correctly except in the bottom left corner where shadow has obscured the edges. Note that the geometry of the measurements around the junctions is far from perfect, though at least we have recovered the topology correctly. The topological edgel description indicates the presence (and complexity) of junctions, and so we could correct the geometry using parametric models such as those suggested by Rohr [34] or Deriche and Giraudon [10] (we are currently evaluating the efficacy of such an approach). All of the edgel points away from the junctions and corners have been detected using a scheme that recovers almost identical sub-pixel geometry to that of the Canny detector, and so for general points we have made no sacrifice to geometry.

Figure 7b shows the quality of the topological description on the robot. Although the representation is complex around the fine-detailed parts of the scene, it is in general correct. We also see how reliably the method returns edges in the original intensity surface: towards the right of the vent just above the centre of the image a set of edgels have been recovered.

a **b** **c**

Figure 8: *A comparison of (b) the Canny detector and (c) the new algorithm for an image of a building. The main difficulty for the Canny operator is posed by the weak boundary contrast on interior edges which produces poor definition of edgel orientations. The new algorithm recovers a much better topological description, even in this case of minimal gradient strengths.*

These contours are present in the intensity surface and are actually stronger than some of those of the right-hand boundary of the object (they result from mutual illumination effects). The Canny operator (Fig. 1) was able to pick out some of the points on the vent, but was unable to recover the complete structure.

Another comparison between the Canny operator and the new algorithm is given in Fig. 8. This is small part of a scene which might be used for a navigation or stereo reconstruction task. Note that the Canny output in (b) has many gaps along the edges of the building (again we choose the thresholds to give the maximum possible contour continuity without allowing too many spurious features to be produced). Although the edgels corresponding to the building edges are generally stronger both than t_{low} and t_{high}, the estimate of the orientation which is crucial for non-maximal suppression is poor. Subsequently, many edgels are filtered out. However, Fig. 8c shows the result for the new algorithm which uses conventional non-maximal suppression only to assign the classification thresholds, and then uses topologically correct thinning to derive the final description. As shown we recover a significant proportion of the image topology.

5 Using topology

Although the results given above are subjective (simply because on real edgel data one can not judge the results of an edgel detector fairly just by looking at its output superimposed on an image), we find that the new algorithm substantially eases the processing stages

that we do subsequently. For instance, a good knowledge of image topology reduces the amount of work that has to be done during edge grouping. Grouping is the process used to associate image features together than are themselves too weak to be useful recognition or navigation cues, but once combined provide strong visual constraints. Binford [5] and Lowe [23] demonstrated early examples of grouping mechanisms; a more recent review is contained in the paper by Sarkar and Boyer [35]. Typically grouping uses geometric measures such as proximity or orientation to associate features rather than employing topology.

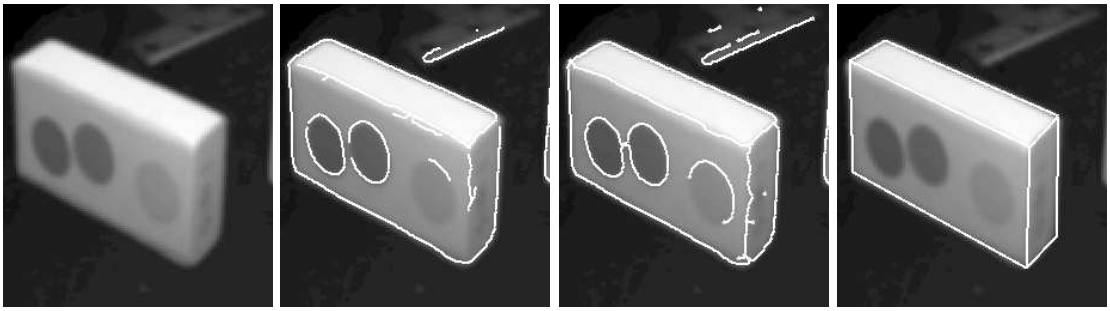
One clear situation in which the benefits of using topological cues for grouping overcome the brute force methods occurs in the alignment algorithm of Huttenlocher and Ullman [21]. In the traditional algorithm, which is dependent on the computation of a plane affine transformation between a model and the image, three model points are paired with each conceivable triple of image points. If we consider only the image points, of which there are n, a total of ${}^nP_3 = O(n^3)$ configurations exist (as the ordering of the points is important). However, as suggested by Huttenlocher in [22], we can make use of the connectivity along an image curve to associate configurations points together. In practice we choose sets of three consecutive feature points on the curve (such points might be points of inflection). Such an algorithm has has a complexity equal to the number of image points, that is $O(n)$. Consequently, the use of topological cues dramatically enhances the efficiency of the algorithm.

It is this philosophy that has driven us to provide better edgel topology; searching for the relationships later on is just far too expensive. Here we demonstrate how the topology extracted from the edgel detector simplifies a task which could be considered to be one of the early processing phases in the recognition of simple polyhedral objects (such as described by Sugihara [37]). Although we do not go into the details of the design of an object recognition system (nor in fact demonstrate that we are actually recognizing objects), we show how we can derive templates of model instances, which are topologically correct line drawings, in scenes that could later be used for recognition.

In Figs 9 and 10 we show how knowledge of the image topology makes the extraction of polyhedral models from images a relatively simple task. No searching for image associations was required for either of the examples; the edge detector provided all of the connectivity and junction information. This means that higher-level processes do not have to expend valuable time organizing the input data. Briefly, the order of processing is: first to detect edgels using the topological algorithm; the representation of straight edgel chains by lines; the extraction of topological structures commensurate with the polyhedral that we wish to recognize (topological faces); and finally a process that ensures the geometric consistency of the edgel and line descriptions (using a polyhedral snake composed of straight lines and junctions). In practice we would then proceed to match the geometric description to models in a library, though this process is beyond the scope of this paper.

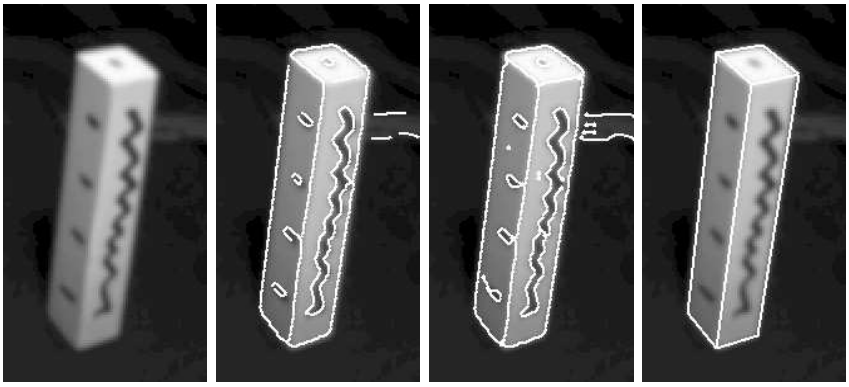
6 Discussion

In this paper we have proposed that topological image descriptions should be derived in a bottom-up manner rather than relying on higher-level processing to find them later. Al-



a **b** **c** **d**

Figure 9: *For a polyhedron we are able to extract topologically correct edgel chains in (c) and then use a snake-like process to derive a geometrically reasonable description in (d). This process succeeds even though the image features are very small and so are easily confounded (the smaller faces have widths of about eight pixels). For comparison, Canny output is shown in (b).*



a **b** **c** **d**

Figure 10: *A second example showing the extraction of a polyhedral description from the intensity image via a topologically reliable edgel description. Again (b) shows Canny output, and (c) the new algorithm.*

though we are not *always* able to find such descriptions, we have demonstrated that far more information can be extracted from the intensity image via edge detection that found previously. This information has also been recovered with very little loss in processing time compared to more conventional edgel detectors. However, the time saved in not having to group the features later on is significant. Furthermore, given that we have examined the intensity surface before constructing any hypotheses associating features together, we can be more sure to find the correct descriptions than if we had used only geometric relationships in the planar image (such as the proximity relationship exploited too readily in traditional grouping schemes).

The efficacy of the approach has been demonstrated on real images. We have not only demonstrated improved performance again a conventional edgel detector (the Canny operator), but have also shown how the topological descriptions can be used to derive useful input data for higher level algorithms. In effect, we have evaluated the performance of the algorithm by examining its usefulness to higher-level processes.

The descriptions are not always perfect. We are now investigating ways to use bottom-down processing to verify the hypotheses made by the edgel detector. For instance, we are evaluating the effectiveness of parametric corner and junction detectors at confirming of rejecting the existence of each of the rich feature descriptions that we have extracted from the intensity image.

Acknowledgments: *We would like to thank acknowledge a variety of people whose different ideas have contributed to the algorithm presented in this paper. Principally we wish to thank Mike Brady, Rachid Deriche, Margaret Fleck, David Forsyth, Luc Robert, Théo Papadopoulo and Andrew Zisserman.*

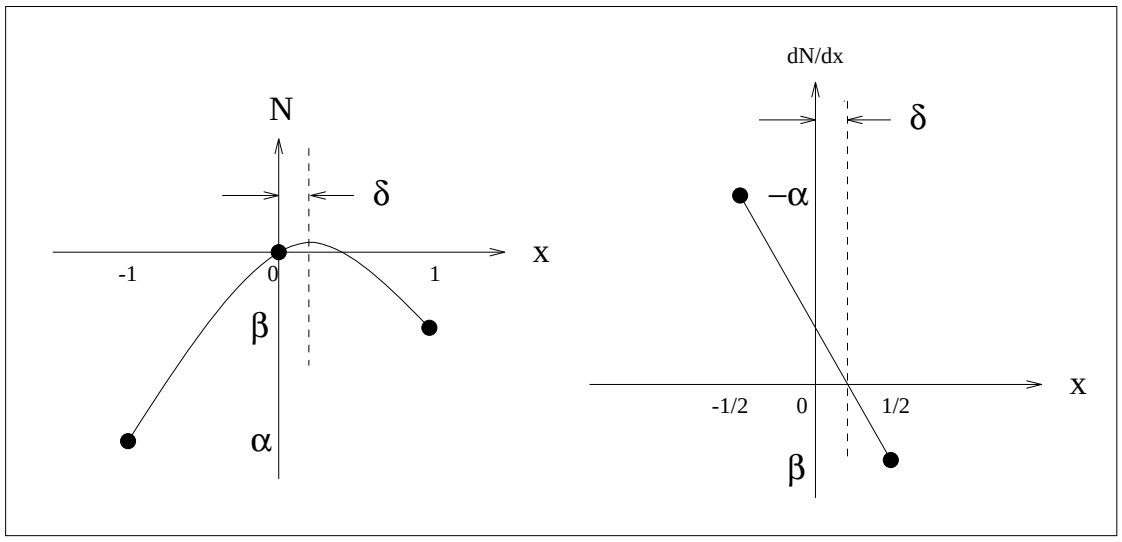


Figure 11: *A cross section through the surface N in the direction of maximal gradient yields the three points shown in the left figure; the coordinates of the points have been normalised to simplify the calculations. We demonstrate that the maximum of a parabolic fit through these three points is equivalent to the localisation of the zero crossing in the second derivative of the image intensities dN/dx, when computed using first differences. This is represented by the right-hand figure.*

A Proof of the equivalence of the parabolic fit and the zero-crossing

In this appendix we demonstrate that the maximum of a local parabolic fit to the norm of the gradient surface N is equivalent to the computation and localisation of the zero crossing of the first derivative of N when computed using finite differences. Therefore, the peak of the parabola is placed exactly at the zero crossing of the second derivative of the intensity surface I.

We consider fitting to three data points that represent a cross-section through the smoothed gradient surface $N = |\nabla S|$ in the direction of maximal gradient in the intensity surface. These points are therefore $\{N_{-\theta}, N_{x,y}, N_{+\theta}\}$, and are shown in Fig. 11. Due to the invariant properties of the fitting and first difference operations we are able to normalise the data so that:

$$N_{-\theta} = (-1, \alpha), \qquad N_{x,y} = (0, 0), \qquad N_{+\theta} = (1, \beta).$$

A parabolic fit of the form $N(x) = a_2x^2 + a_1xa_0$ to these three points is exact, and takes the form:

$$N(x) = \frac{\alpha+\beta}{2}x^2 + \frac{\beta-\alpha}{2}x.$$

The function $N(x)$ represents a continuous approximation to the norm of the gradient surface. The maximum of this function (or equivalently the minimum for certain values of α and β) satisfies the constraint $dN(x)/dx = 0$ and lies at:

$$x = \frac{1}{2}\ \frac{\alpha-\beta}{\alpha+\beta}.$$

Now, computing an estimate of dN/dx using first differences provides us with two constraint points: the differences are computed between $\{N_{-\theta}, N_{x,y}\}$ and $\{N_{x,y}, N_{\theta}\}$. As in Fig. 11 we place these points at $x = \pm 1/2$. Linearly interpolating between these points provides a continuous estimate of $dN(x)/dx$, or equivalently d^2I/dx^2. The zero crossing of the function is also located at:

$$x = \frac{1}{2}\ \frac{\alpha-\beta}{\alpha+\beta}.$$

Thus the parabolic fitting approach to edgel localisation, and that based on the estimation of the zero crossing, are precisely equivalent.

B Chamfer filtering

The masks used to compute the distance of a pixel to the two nearest edgel points are based on the 34-distance transform of Borgefors [6]. We define four sets of masks which evolve from the four corners of the image. The masks work in pairs (diametrically opposed, but in fact partially overlapping with the alternate pair of masks). The output of each mask gives the distance of the closest edgel to a point in the area covered by the mask up to the time that the given point was the focus of attention. There are two image-sized arrays associated with each mask: the first records the distance d_i to the nearest edgel (up to a constant scale factor); and the second n_i, the strength of that edgel.

Initially, we set the four p_i images to a large constant value for all pixels further than a distance 2σ from each edgel. This value must be greater than the maximum possible distance of any one point to any other (i.e. the diagonal dimension of the image). All pixels corresponding to edgel points, and those within the disks, are set to zero (as the distance to the nearest edgel point, when at one, must be zero). The n_i images are initialised to have correct edgel

strengths at the edgel locations, are equal to the central edgel strength for all points within 2σ the disk, and arbitrary values elsewhere. Wherever any value of n_i is multiply defined (due to overlap of the disks), the lower value is taken. We represent the discrete image location of a point by the coordinates $[i, j]$.

The four masks are evolved in the pairs A_mask_i and B_mask_i, $i \in \{1, 2\}$, with the A_masks forming one pair and the B_masks the other:

A_mask_1:
for $i = 1$ **to** $width - 2$ **step 1 do**
  for $j = 1$ **to** $height - 2$ **step 1 do**
    switch min$(d_1[i-1, j-1] + 4, d_1[i-1, j] + 3, d_1[i-1, j+1] + 4, d_1[i, j-1] + 3, d_1[i, j])$;
      case $d_1[i-1, j-1] + 4$:
        $d_1[i, j] = d_1[i-1, j-1] + 4$; $n_1[i, j] = n_1[i-1, j-1]$;
      case $d_1[i-1, j] + 3$:
        $d_1[i, j] = d_1[i-1, j] + 3$; $n_1[i, j] = n_1[i-1, j]$;
      case $d_1[i-1, j+1] + 4$:
        $d_1[i, j] = d_1[i-1, j+1] + 4$; $n_1[i, j] = n_1[i-1, j+1]$;
      case $d_1[i, j-1] + 3$:
        $d_1[i, j] = d_1[i, j-1] + 3$; $n_1[i, j] = n_1[i, j-1]$;
      case $d_1[i, j]$:
        nop;

A_mask_2:
for $i = width - 2$ **to 1 step -1 do**
  for $j = height - 2$ **to 1 step -1 do**
    switch = **min**$(d_2[i, j], d_2[i, j+1] + 3, d_2[i+1, j-1] + 4, d_2[i+1, j] + 3, d_2[i+1, j+1] + 4)$;
      case $d_2[i, j]$:
        nop;
      case $d_2[i, j+1] + 3$:
        $d_2[i, j] = d_2[i, j+1] + 3$; $n_2[i, j] = n_2[i, j+1]$;
      case $d_2[i+1, j-1] + 4$:
        $d_2[i, j] = d_2[i+1, j-1] + 4$; $n_2[i, j] = n_2[i+1, j-1]$;
      case $d_2[i+1, j] + 3$:
        $d_2[i, j] = d_2[i+1, j] + 3$; $n_2[i, j] = n_2[i+1, j]$;
      case $d_2[i+1, j+1] + 4$:
        $d_2[i, j] = d_2[i+1, j+1] + 4$; $n_2[i, j] = n_2 A[i+1, j+1]$;

B_mask_1:
for $i = 1$ **to** $width - 2$ **step 1 do**
  for $j = height - 2$ **to** $j = 1$ **step -1 do**
    switch min$(d_3[i-1, j+1] + 4, d_3[i-1, j] + 3, d_3[i-1, j-1] + 4, d_3[i, j+1] + 3, d_3[i, j])$;
      case $d_3[i-1, j+1] + 4$:

$d_3[i,j] = d_3[i-1,j+1] + 4; \quad n_3[i,j] = n_3[i-1,j+1];$
case $d_3[i-1,j] + 3$:
$d_3[i,j] = d_3[i-1,j] + 3; \quad n_3[i,j] = n_3[i-1,j];$
case $d_3[i-1,j-1] + 4$:
$d_3[i,j] = d_3[i-1,j-1] + 4; \quad n_3[i,j] = n_3[i-1,j-1];$
case $d_3[i,j+1] + 3$:
$d_3[i,j] = d_3[i,j+1] + 3; \quad n_3[i,j] = n_3[i,j+1];$
case $d_3[i,j]$:
nop;

B_mask_2:
for $i = width - 2$ **to** $i = 1$ **step -1 do**
for $j = 1$ **to** $j = height - 2$ **step 1 do**
val = min$(d_4[i,j], d_4[i,j+1] + 3, d_4[i+1,j-1] + 4, d_4[i+1,j] + 3, d_4[i+1,j+1] + 4)$;
case $d_4[i,j]$:
nop;
case $d_4[i,j+1] + 3$:
$d_4[i,j] = d_4[i,j+1] + 3; \quad n_4[i,j] = n_4[i,j+1];$
case $d_4[i+1,j-1] + 4$:
$d_4[i,j] = d_4[i+1,j-1] + 4; \quad n_4[i,j] = n_4[i+1,j-1];$
case $d_4[i+1,j] + 3$:
$d_4[i,j] = d_4[i+1,j] + 3; \quad n_4[i,j] = n_4[i+1,j];$
case $d_4[i+1,j+1] + 4$:
$d_4[i,j] = d_4[i+1,j+1] + 4; \quad n_4[i,j] = n_4[i+1,j+1];$

Here we have made use of C-style indexing for arrays, that is the first element of an array is index in a form equivalent to $a[0]$. The term 'nop' is the null operation.

Once the masks have been passed over the images we test for each point x that is not an edgel. Depending on whether the nearest edgel position (the smallest value of $d_i[\mathbf{x}]$, $i \in \{1,\ldots,4\}$) is in the A set or the B set of masks, we compute the value of the threshold function $T(\mathbf{x})$ by:

switch min$(d_1[\mathbf{x}], d_2[\mathbf{x}], d_3[\mathbf{x}], d_4[\mathbf{x}])$
case $d_1[\mathbf{x}]$ **or** $d_2[\mathbf{x}]$:
$t = \frac{d_2[\mathbf{x}] n_1[\mathbf{x}] + d_1[\mathbf{x}] n_2[\mathbf{x}]}{d_1[\mathbf{x}] + d_2[\mathbf{x}]}$;
case $d_3[\mathbf{x}]$ **or** $d_4[\mathbf{x}]$:
$t = \frac{d_4[\mathbf{x}] n_3[\mathbf{x}] + d_3[\mathbf{x}] n_4[\mathbf{x}]}{d_3[\mathbf{x}] + d_4[\mathbf{x}]}$;
If $(\mathbf{x})$ **is a known edgel**
$T[\mathbf{x}] = N[\mathbf{x}]$
else if $t > t_{low}$

$T[\mathbf{x}] = t$
else
$T[\mathbf{x}] = t_{low}$

Note any value of d_i equal to the initial large value means that no edgel has been found within the scope of the mask. This means that there is only one closest edgel defined, and so $T[\mathbf{x}]$ should be set equal to the strength of that single edgel. Such situations occur frequently around the borders of the image.

References

[1] **Ayache, N. and Faugeras, O.D. "HYPER: A New Approach for the Recognition and Positioning of Two-Dimensional Objects,"** *IEEE Trans. PAMI,* **Vol. 8, No. 1, p.44-54, January 1986.**

[2] **Ayache, N. and Faugeras, O.D. "Building a Consistent 3D Representation of a Mobile Robot Environment by Combining Multiple Stereo Views," Proceedings IJCAI, p.808-810, 1987.**

[3] **Beaudet, P.R. "Rotationally Invariant Image Operators," Proceedings ICPR, p.579-583, 1979.**

[4] **Blake, A. and Cipolla, R. "Robust Estimation of Surface Curvature from Deformation of Apparent Contours," Proceedings ECCV1, p.465-474, 1990.**

[5] **Binford, T.O. "Inferring Surfaces from Images,"** *Artificial Intelligence,* **Vol. 17, p.205-244, 1981.**

[6] **Borgefors, G. "Hierarchical Chamfer Matching: A Parametric Edge Matching Algorithm,"** *IEEE Trans. PAMI,* **Vol. 10, No. 6, p.849-865, November 1988.**

[7] **Canny J.F. "A Computational Approach to Edge Detection,"** *IEEE Trans. PAMI,* **Vol. 8, No. 6, p.679-698, 1986.**

[8] **Canny, J.F. "Finding Edges and Lines in Images," M.Sc. thesis, MIT, 1983.**

[9] **Deriche, R. "Using Canny's Criteria to Derive a Recursively Implemented Optimal Edge Detector,"** *IJCV,* **Vol. 1, p.167-187, 1987.**

[10] **Deriche, R. and Giraudon, G. "A Computational Approach for Corner and Vertex Detection,"** *IJCV,* **Vol. 10, No. 2, p.101-124, 1993.**

[11] **Du Li, Sullivan, G.D. and Baker, K.D. "Edge Detection at Junctions," Proceedings 5th Alvey Vision Conference, p.121-125, 1989.**

[12] **Fleck, M.M. "Boundaries and Topological Algorithms," Ph.D. Thesis, MIT AI Lab, 1988.**

[13] **Freeman, W.T. and Adelson, E.H. "The Design and Use of Steerable Filters,"** *PAMI,* **Vol. 13, No. 9, p.891-906, 1991.**

[14] **Grimson, W.E.L. and Lozano-Pérez, T. "Localizing Overlapping Parts by Searching the Interpretation Tree,"** *IEEE Trans. PAMI,* **Vol. 9, No. 4, p.469-482, July 1987.**

[15] **Grimson, W.E.L.** *Object Recognition by Computer, The Role of Geometric Constraints,* **MIT Press, 1990.**

[16] Haralick, R,M. "Digital Step Edges from Zero Crossing of Second Directional Derivatives," *PAMI,* **Vol. 6, p.58-68, 1984.**

[17] Haralick, R,M. "Model-Based Morphology: Simple and Complex Shapes," *Visual Form,* **p.275-285, 1992.**

[18] Haralick, R.M. and Shapiro, L.G. *Computer and Robot Vision,* **Addison-Wesley, 1993.**

[19] Hueckel, M.H. "An Operator which Locates Edges in Digitized Pictures," *Journal of the Association for Computing Machinery,* **Vol. 18, No. 1, p.113-125, 1971.**

[20] Hueckel, M.H. "A Local Visual Operator which Recognizes Edges and Lines," *Journal of the Association for Computing Machinery,* **Vol. 20, No. 4 p.634-647, 1971.**

[21] Huttenlocher, D.P. and Ullman, S. "Object Recognition Using Alignment," Proceedings ICCV1, p.102-111, 1987.

[22] Huttenlocher, D.P. "Three-Dimensional Recognition of Solid Objects from a Two-Dimensional Image," Ph.D. Thesis, Department of Electrical Engineering and Computer Science, MIT, 1988.

[23] Lowe, D.G. *Perceptual Organization and Visual Recognition,* **Kluwer Academic Publishers, 1985.**

[24] Marr, D. and Hildreth, E. "Theory of Edge Detection," Proceedings of the Royal Society of London, Series B, Vol. 207, p.187-217, 1980.

[25] Mundy, J.L. and Heller, A.J. "The Evolution and Testing of a Model-Based Object Recognition System," Proceedings ICCV3, p.268-282, 1990.

[26] Nalwa, V.S. and Binford, T.O. "On Detecting Edges," Proceedings IUW, p.450-465, 1985.

[27] Noble, J.A. "Finding Corners," Proceedings 3rd Alvey Vision Conference, p.267-274, 1987.

[28] Noble, J.A. "Morphological Feature Detection," Proceedings 4th Alvey Vision Conference, p.203-210, 1988.

[29] Perona, P. and Malik, J. "Scale Space and Edge Detection using Anisotropic Diffusion," Proceedings Workshop on Computer vision, p.16-22, 1987.

[30] Petrou, M. "On the Optimal Edge Detector," Proceedings 4th Alvey Vision Conference, p.191-196, 1988.

[31] Pollard, S.B, Pridmore, T.P, Porrill, J., Mayhew, J.E.W. and Frisby, J.P. "Geometrical Modelling from Multiple Stereo Views," *International Journal of Robotics Research,* **Vol. 8, No. 4, p.132-138, 1989.**

[32] Rake, S. "Finding Curved Ribbons," *TR IBM UKSC 200*, **1988.**

[33] Roberts, L.G. "Machine Perception of Three-Dimensional Solids," *Optical and Electro-optical Information Processing,* **Tippett,** *et al.* **editors, MIT Press, p.159-197, 1965,**

[34] Rohr, K. "Recognizing Corners by Fitting Parametric Models," *IJCV,* **Vol. 9, No. 3, p.213-230, 1992.**

[35] Sarkar, S. and Boyer, K.L. "Perceptual Organization in Computer Vision: A Review and a Proposal for a Classificatory Structure," *IEEE Systems, Man, and Cybernetics,* **Vol. 23, p.382-399, 1993.**

[36] Spacek, L. "Edge Detection and Motion Detection," *Image and Vision Computing,* **Vol. 4, No. 1, 1986.**

[37] Sugihara, K. *Machine interpretation of Line Drawings,* **MIT Press, 1986.**

[38] Terzopoulos, D., Witkin, A. and Kass. "Constraints on Deformable Models: Recovering Shape and non-Rigid Motion," *Artificial Intelligence,* **Vol. 36, No. 1, p.91-123, 1988.**

[39] Thompson, D.W. and Mundy, J.L. "Three-Dimensional Model Matching from an Unconstrained Viewpoint," Proceedings ICRA, p.208-220, 1987.

[40] Tsao, Y.F. and Fu, K.S. "Parallel Thinning Operations for Digital Binary Images," Proceedings CVPR, p.150-155, 1981.

[41] Witkin, A.P. "Scale-Space Filtering," *Readings in Computer Vision - Issues, Problems, Principles and Paradigms,* **Morgan Kaufmann, p.329-332, 1987.**

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399