

# Computing Contour Closure

James H. Elder<sup>1</sup> and Steven W. Zucker<sup>2</sup>

<sup>1</sup> NEC Research Institute, Princeton, NJ, U.S.A.

<sup>2</sup> Centre for Intelligent Machines, McGill University, Montréal, Canada.

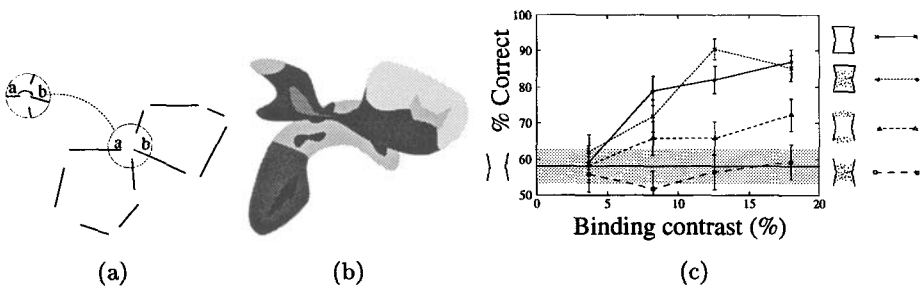
**Abstract.** Existing methods for grouping edges on the basis of local smoothness measures fail to compute complete contours in natural images: it appears that a stronger global constraint is required. Motivated by growing evidence that the human visual system exploits contour closure for the purposes of perceptual grouping [6, 7, 14, 15, 25], we present an algorithm for computing highly closed bounding contours from images. Unlike previous algorithms [11, 18, 26], no restrictions are placed on the type of structure bounded or its shape. Contours are represented locally by tangent vectors, augmented by image intensity estimates. A Bayesian model is developed for the likelihood that two tangent vectors form contiguous components of the same contour. Based on this model, a sparsely-connected graph is constructed, and the problem of computing closed contours is posed as the computation of shortest-path cycles in this graph. We show that simple tangent cycles can be efficiently computed in natural images containing many local ambiguities, and that these cycles generally correspond to bounding contours in the image. These closure computations can potentially complement region-grouping methods by extending the class of structures segmented to include heterogeneous structures.

## 1 Introduction

We address the problem of computing closed bounding contours in real images. The problem is of interest for a number of reasons. A basic task of early vision is to group together parts of an image that project from the same structure in a scene. Studies of perceptual organization have demonstrated that the human visual system exploits a range of regularities in image structure to solve this task [12, 14, 28]. Inspired in part by these studies, algorithms have been developed to apply continuity, cocircularity and smoothness constraints to organize local edges into extended contours [2, 10, 17, 19, 22, 24]. However, despite persuasive psychophysical demonstrations of the role of contour closure in perceptual organization [6, 7, 14, 15, 25], closure constraints for computer vision algorithms have been largely ignored (although see [1, 11]). This is surprising, since existing algorithms, while capable of producing extended chains of edges, are seldom successful in grouping complete contours. Closure is a potentially powerful cue that could complement smoothness cues to allow more complete contours to be computed.

In natural images, contours are fragmented due to occlusions and other effects, making local grouping cues weak and unreliable. While these local cues

may be summed or otherwise integrated along a contour to form global measures of “smoothness”, “likelihood” or “saliency”, such a global measure is as weak as its weakest local constituent. This is illustrated in Fig. 1(a): the most plausible continuation of a contour viewed locally may be clearly incorrect when viewed in global context. This error is not revealed in a summation of local grouping cues over the curve, since both the correct and the incorrect continuations lead to similar measures. A global feature is needed which is far more sensitive to such local errors. Closure is a potentially powerful feature because a single local error will almost certainly lead to a low measure of closure.



**Fig. 1.** (a) Locally, the most plausible continuation of fragment *a* is through fragment *b*. Given global context, fragments instead group to form simple cycles with high measures of closure. (b) A region grouping algorithm would segment this image into 12 disjoint regions, yet human observers see two overlapping objects. Regularities of the object boundaries must be exploited. (c) Psychophysical data for shape identification task. Subjects must discriminate between a fragmented concave shape (shown) and a 1-D equivalent convex shape (not shown), at very low contrasts. Results show that contour closure cues greatly enhance performance. No effect of texture cues is observed. From [8].

The computation of closed contours is also potentially useful for grouping together image *regions* which project from common structures in the scene: objects, parts of objects, shadows and specularities. Existing techniques for region grouping apply homogeneity or smoothness constraints on luminance, colour or texture measures over regions of the image (e.g. 16, 20, 21). These techniques have inherent limitations. While a region-grouping algorithm would segment the image of Fig. 1(b) into 12 disjoint components, human observers perceive two irregularly painted, overlapping objects. Since these are nonsense objects, our inference cannot be based on familiarity. We must be using the geometry of the boundaries to group the objects despite their heterogeneity.

This situation is not artificial. Objects are often highly irregular in their sur-

face reflectance functions, and may be dappled in irregular ways by shadows and specularities. While surface markings, shadows and specularities fragment image regions into multiple components, geometric regularities of bounding contour persist. Contour closure is thus potentially important for segmentation because it broadens the class of structures that may be segmented to include such heterogeneous structures. Interestingly, recent psychophysical experiments [8] suggest that contour grouping cues such as closure may be more important than regional texture cues for the perceptual organization of 2-D form (Fig. 1(c)).

## 2 Previous Work

The problem of contour grouping has been approached in many different ways. Multi-scale smoothness criteria have been used to impose an organization on image curves [4, 17, 22], sequential methods for tracking contours within a Bayesian framework have recently been developed [2] and parallel methods for computing local “saliency” measures based on contour smoothness and total arclength have been studied [1, 24]. In general, these techniques are capable of grouping edge points into extended chains. However, no attempt is made to compute *closed* chains; a necessary condition for computing global 2-D shape properties and for segmenting structures from an image.

A separate branch of research investigates the grouping of *occlusion* edges into complete contours, ordered in depth [18, 26]. While interesting from a theoretical point of view, a large fraction of the edges in real images are not occlusion edges, and a recent study [5] suggests that it is not possible to locally distinguish occlusion edges from other types of edges. It is our view that algorithms for grouping contours must work for all types of structure in an image (e.g. objects, shadows, surface markings).

Jacobs [11] has studied the problem of inferring highly-closed convex cycles of line segments from an image, to be used as input for a part-based object recognition strategy. Given the generality of boundary shape, it is clearly of great interest to determine whether bounding contours can be recovered without such restrictive shape constraints. Most similar to our work is a very recent study by Alter [1] on the application of shortest-path algorithms to the computation of closed image contours. While similar in concept, these two independent studies differ substantially in their implementation.

## 3 Overview of the Algorithm

Our goal here is to recover cycles of edge points which bound two-dimensional structures in an image. The algorithm is to be fully automatic and no restrictions are placed on the type of structure bounded or its shape. Since no constraint of disjointness is imposed, in principle the bounding contours of an entire object, its parts, markings, and shadows can be recovered.

Image contours are represented locally as a set of tangent vectors, augmented by image intensity estimates. A Bayesian model is developed to estimate the

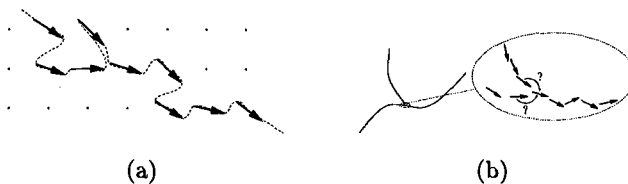
likelihoods that tangent pairs form contiguous components of the same image contour. Applying this model to each tangent in turn allows the possible continuant tangents for each tangent to be sorted in order of their likelihood. By selecting for each tangent the 6 most likely continuant tangents, a sparse (6-connected), weighted graph is constructed, where the weights are the computed pairwise likelihoods.

By assuming independence between tangent pair likelihoods, we show that determining the most probable tangent cycle passing through each tangent can be posed as a shortest path computation over this sparse graph. We can therefore use standard algorithms (e.g. Dijkstra's algorithm [23]) to solve this problem in low-order polynomial time.

## 4 Extended Tangents

Edges are detected by a multi-scale method which automatically adapts estimation scale to the local signal strength and provides reliable estimates of edge position and tangent orientation [9]. In addition to the geometric properties of position and orientation, we make use of local image intensity estimates provided by our edge detector.

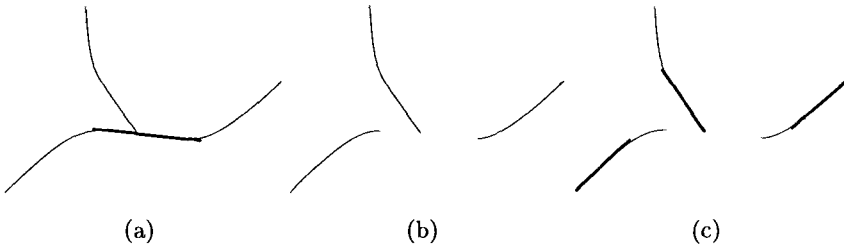
Due to uncertainty induced by discretization and sensor noise, contours generate noisy, laterally-displaced local edges (Fig. 2(a)). Tracing a contour through these local tangents generates a curve corrupted by wiggles due to sensing artifacts. Also, due to blurring of the luminance function at the imaging and estimation stages, edge estimates near corners and junctions are corrupted (Fig. 2(b)).



**Fig. 2.** (a) The set of raw tangent estimates for a contour. Imposing an ordering on these local tangent estimates generates a contour distorted by sampling artifacts. (b) The smoothing of the image at the sensing and estimation stages corrupts tangent estimates near contour junctions and corners.

Achieving a more reliable local representation requires more global constraints. Here, we introduce a method for refining local edge information based on an *extended tangent representation*, which represents a curve as a sequence

of disjoint line segments. Each local edge in the image generates a tangent line passing through the edge pixel in the estimated tangent direction. The subset of tangent estimates which are 8-connected to the local edge and which lie within an  $\epsilon$ -neighbourhood of the local tangent line are identified with the extended tangent model. The algorithm for selecting extended tangents to approximate a contour is illustrated in Fig. 3. Given a connected set of local edges, the longest line segment which faithfully models a subset of these is determined. This subset is then subtracted from the original set. This process is repeated for the connected subsets thus created until all local edges have been modeled.



**Fig. 3.** Computing the extended tangent representation. (a) For each connected set of edge pixels, the subset of pixels underlying the longest extended tangent is selected. (b) The edge points thus modeled are subtracted. (c) The process is repeated for each connected set of edge pixels thus spawned.

Since the extended tangents selected must be consistent with the global geometry of the curves, they provide more accurate estimates of contrast and tangent orientation than do the corrupted local edges near the junction. The extended tangent algorithm is conceptually simpler than most methods for computing polygonal approximations [20], and does not require preprocessing of the edge map to link local edges into ordered lists, as is required for most other methods (e.g. 11, 17, 20).

## 5 A Bayesian Model for Tangent Grouping

The extended tangent representation leads naturally to a representation for global contours as tangent sequences:

**Definition 1.** A *tangent sequence*  $t_1 \rightarrow \dots \rightarrow t_n$  is an injective mapping from a finite set of integers to a set of extended tangents.

The injective property restricts our definition to sequences which do not pass through the same extended tangent twice. The identification of extended

tangents with integers imposes an *ordering* on the tangents which distinguishes a tangent sequence from an arbitrary clutter of tangents. If a contour bounds a 2-D structure in the image, this sequence will come back on itself. Thus bounding contours are represented as *cycles* of extended tangents,  $t_1 \rightarrow \dots \rightarrow t_n \rightarrow t_1 \rightarrow \dots$

By this definition, any ordered set of tangents in an image can form a tangent sequence. In order to compute bounding contours, some measure of the likelihood of a tangent sequence must be established. For this purpose, we develop a Bayesian model for estimating the posterior probability of a tangent sequence given data on the geometric and photometric relations between adjacent tangent tuples of the sequence.

We will begin by assuming that tangent links are independent: i.e.

$$p(t_1 \rightarrow \dots \rightarrow t_n) = p(t_1 \rightarrow t_2)p(t_2 \rightarrow t_3)\dots p(t_{n-1} \rightarrow t_n)$$

This approximation will greatly simplify the computation of tangent sequence likelihoods, reducing likelihood estimation for a tangent sequence to the problem of estimating the likelihoods of its constituent links. The likelihood that two tangents project from the same contour is modeled as the probability of their rectilinear completion (Fig. 4), so that the probability of a link depends on the following observables (see Sections 6 and 7 for details of the model):

1. The lengths  $l_1$  and  $l_2$  of the extended tangents.
2. The length  $r$  of the straight-line interpolant.
3. The 2 orientation changes  $\theta_a$  and  $\theta_b$  induced by the interpolation.
4. The differences in estimated image intensity  $\Delta i_h$ ,  $\Delta i_l$  on the bright side and the dark side of the tangents, respectively.

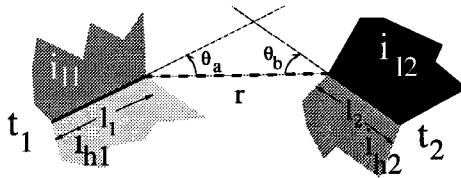


Fig. 4. Rectilinear interpolation model.

Setting  $o = \{l_1, l_2, r, \theta_a, \theta_b, \Delta i_h, \Delta i_l\}$ , Bayes' theorem can be used to express the posterior probability of a link from tangent  $t_1$  to  $t_2$  (called the "link hypothesis") in terms of the likelihoods of the observables:

$$p(t_1 \rightarrow t_2 | o) = \frac{p(o | t_1 \rightarrow t_2)p(t_1 \rightarrow t_2)}{p(o)}$$

Letting  $t_1 \nrightarrow t_2$  represent the hypothesis that  $t_2$  is *not* the continuant of  $t_1$  (the “no-link hypothesis”), the evidence  $p(o)$  can be expanded as

$$p(o) = p(o|t_1 \rightarrow t_2)p(t_1 \rightarrow t_2) + p(o|t_1 \nrightarrow t_2)p(t_1 \nrightarrow t_2).$$

It is convenient to rewrite the posterior probability as

$$p(t_1 \rightarrow t_2|o) = \frac{1}{(1 + LP)}$$

where

$$L = \frac{p(o|t_1 \nrightarrow t_2)}{p(o|t_1 \rightarrow t_2)} \quad P = \frac{p(t_1 \nrightarrow t_2)}{p(t_1 \rightarrow t_2)}$$

The *prior ratio*  $P$  represents the ratio of the probability that a curve ends at  $t_1$ , to the probability that the curve continues. For most images, curves are expected to continue over many tangents. It is therefore appropriate to choose a large value for the prior ratio: in our experiments we use  $P = 50$ .

The *likelihood ratio*  $L$  represents the ratio of the likelihood of the observables given that  $t_2$  is *not* a continuant of  $t_1$  to their likelihood given that  $t_2$  is a continuant of  $t_1$ . Models for these likelihoods are developed in the next two sections.

## 6 Link Hypothesis Likelihoods

In order to model the link hypothesis likelihoods  $p(o|t_1 \rightarrow t_2)$  we must consider the distinct events that can split the image curve into two separate extended tangents  $t_1$  and  $t_2$ . The three possible hypotheses for a *tangent split* considered are termed respectively the *curvature*, *interruption* and *corner* hypotheses:

**Curvature** The contour is curving smoothly: two tangents are needed to model the local edges to  $\epsilon$  accuracy. Relatively small values for  $r$ ,  $\theta_a$  and  $\theta_b$  are expected.

**Interruption** The contour is interrupted, for example by an occlusion, shadow, or loss of contrast. We expect potentially large values for  $r$ , but again relatively small values for  $\theta_a$  and  $\theta_b$ .

**Corner** The contour corners sharply: two tangents are generated on either side of the corner. We expect a relatively small value for  $r$ , but possibly large values for  $\theta_a$  and  $\theta_b$ .

Since each of these hypotheses generates different expectations for the observables, the corresponding link hypothesis likelihoods are decomposed into likelihoods for the 3 disjoint events:

$$\begin{aligned} p(o|t_1 \rightarrow t_2) &= p(o|t_1 \rightarrow t_2, \text{curvature})p(\text{curvature}) \\ &\quad + p(o|t_1 \rightarrow t_2, \text{interruption})p(\text{interruption}) \\ &\quad + p(o|t_1 \rightarrow t_2, \text{corner})p(\text{corner}) \end{aligned}$$

In a natural world of piecewise-smooth objects, the curvature hypothesis is the most likely. For our experiments we assign

$$p(\text{curvature}) = 0.9 \text{ and } p(\text{interruption}) = p(\text{corner}) = 0.05.$$

Combining the observables  $l_1, l_2$  and  $r$  into a normalized gap length  $r'$ ,

$$r' = \frac{r}{\min\{l_1, l_2\}}$$

we write a summarized set of observables as  $o' = \{r', \theta_a, \theta_b, \Delta b, \Delta d\}$ . Approximating these as conditionally independent on the 3 tangent split hypotheses, we use half-Gaussian functions to model the link hypothesis likelihoods for each observables  $o_i$ :

$$p(o_i|t_1 \rightarrow t_2) = \frac{\sqrt{2}}{\sqrt{\pi}\sigma_{o_i}} e^{-\frac{o_i^2}{2\sigma_{o_i}^2}}, \quad o_i > 0.$$

The scale constants  $\sigma_{o_i}$  used in this paper are shown in Table 1.

	$\sigma_r$ (pixels)	$\sigma_{r'}$ (pixels)	$\sigma_{\theta_a} = \sigma_{\theta_b}$ (deg)	$\sigma_{\Delta b} = \sigma_{\Delta d}$ (grey levels)
curvature	2	-	10	20
interruption	-	0.5	10	20
corner	2	-	90	20

**Table 1.** Scale constants for link hypothesis likelihood functions

## 7 No-Link Hypothesis Likelihoods

Modelling the position of a tangent as a uniform distribution over the image domain, for an  $L \times L$  image, and  $r \ll L$  we can approximate the no-link likelihood for  $r$  as  $p(r) \approx \frac{4r}{L^2}$  [5]. No-link likelihood functions for  $\theta_a$  and  $\theta_b$  follow immediately from the assumption of isotropic tangent directions:

$$p(\theta_a|t_1 \nrightarrow t_2) = p(\theta_b|t_1 \nrightarrow t_2) = \frac{1}{\pi}, \quad 0 \leq \theta_a, \theta_b \leq \pi.$$

Modelling image intensity  $i$  as a uniform distribution,  $0 \leq i \leq 255$ , no-link likelihood functions for  $\Delta i_h$  and  $\Delta i_l$ , can be derived [5]:

$$p(\Delta i_h|t_1 \rightarrow t_2) = \frac{2}{255}(1 - \Delta i_h/255) \quad p(\Delta i_l|t_1 \rightarrow t_2) = \frac{2}{255}(1 - \Delta i_l/255)$$

## 8 Constructing a Sparse Graph

Since tangent tuple observations provide only a weak means for discriminating probable from improbable links (Fig. 1(a)), it is essential that the data structure from which tangent cycles are computed represent multiple potential continuations for each tangent. This also allows for the co-occurrence of overlapping cycles, which, as we shall see, occur frequently in natural images.

We construct an appropriate data structure by computing the likelihoods for all tangent pairs, and then selecting for each tangent the  $m$  most likely continuant tangents. In this way, we represent the image as a sparse, weighted graph in which each vertex represents a tangent, directly connected to  $m$  other tangents. Tangent links now become edges in this graph, weighted by the computed posterior probabilities. We set  $m = 6$  for the experiments presented here.<sup>3</sup>

The complexity of building the sparse graph is  $O(n^2)$ , where  $n$  is the number of tangents in the image. In practice, this procedure takes approximately 1 minute of computation on a Sparc 10 for a moderately complex  $300 \times 400$  image.

## 9 Maximum Likelihood Cycles

We set as our goal the computation of the maximum likelihood cycle for each tangent in this tangent graph. By the identification between the vertices and edges of this graph and the extended tangents and tangent links of the contour model, these cycles correspond to highly closed contours in the image. Note that since the graph is sparse, not all tangents lie on a cycle. Also, the same cycle will often be computed by more than one tangent. These duplicate cycles are easily detected and ignored. Finally, observe that any given tangent may lie on many cycles, but each cycle must form the best closure for at least one tangent in the image.

Since tangent links are assumed to be independent, the likelihood of a tangent sequence is simply the product of the probabilities of its constituent links:

$$p(t_1 \rightarrow \dots \rightarrow t_n) = p(t_1 \rightarrow t_2)p(t_2 \rightarrow t_3)\dots p(t_{n-1} \rightarrow t_n)$$

Taking the natural logarithm, we have

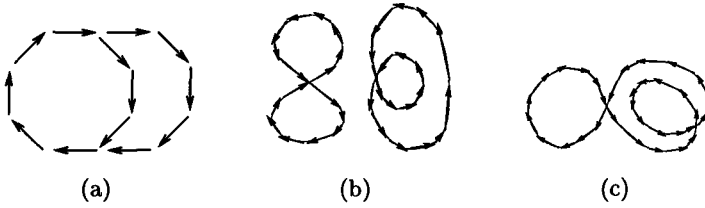
$$\log p(t_1 \rightarrow \dots \rightarrow t_n) = \log p(t_1 \rightarrow t_2) + \log p(t_2 \rightarrow t_3) + \dots + \log p(t_{n-1} \rightarrow t_n)$$

Since all of the terms on the right-hand side are negative, maximizing this sum corresponds to minimizing the sum of the absolute values of the terms. The problem of computing a maximum likelihood sequence can therefore be expressed as a minimization of the absolute sum of the log likelihoods for each link. Thus computing the maximum likelihood cycle for each tangent is a shortest-path problem, and can be solved using standard techniques, such as Dijkstra's algorithm [23], allowing the maximum likelihood cycles for all tangents to be computed in  $O(n^2 \log n)$  operations, where  $n$  is the number of tangents in the image.

<sup>3</sup> Since a tangent may link to another tangent in 2 possible ways (contrast reversals are allowed), the size of the graph is effectively doubled.

## 10 Topological Constraints

Up to this point we have ignored the topology of bounding contours: self-intersecting contours have not been ruled out. Demanding that each extended tangent appear at most once in any tangent sequence eliminates many intersecting cycles (Fig. 5(a)), but there remain those cycles which intersect but which do not overlap the same tangent twice (Fig. 5(b-c)).<sup>4</sup>



**Fig. 5.** Topological constraints on tangent cycles. (a) Each tangent may appear at most once in a tangent sequence. (b) By constraining cycles to have unit rotation index, “figure eights” and “double loops” are eliminated. (c) More complex intersecting contours with unit rotation index are not restricted.

A subclass of these self-intersecting contours can be eliminated by constraining the rotation index [3] of the underlying contour to be

$$\frac{1}{2\pi} \sum_1^n (\theta_{ai} + \theta_{bi}) = \pm 1.$$

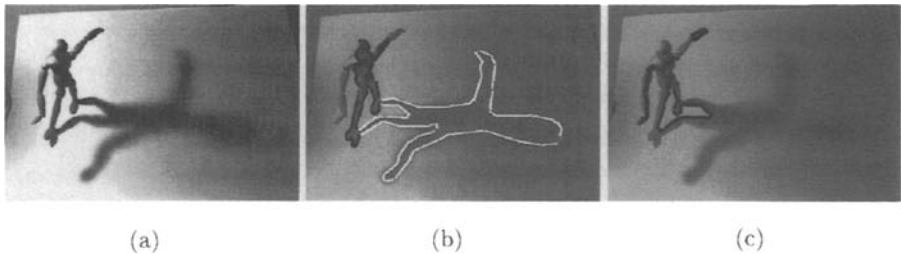
This eliminates the more common types of erroneous cycles, such as “figure eights” (rotation index 0) and “double loops” (rotation index  $\pm 2$ ) (Fig. 5(b)). However, more complex self-intersecting contours with rotation index  $= \pm 1$  are still possible (Fig. 5(c)), and we do not detect these.

These topological constraints cannot be embodied in the weights of the tangent graph, since the rotation index cannot be computed until a complete cycle has been determined. However, by assuming an upper bound on the number of different rotation indices that may be generated from each tangent, Dijkstra’s algorithm can be generalized to incorporate these constraints. This modification does not change the complexity of the algorithm, which requires on the order of 1-2 minutes to complete in our experiments.

<sup>4</sup> The rim of a smooth solid *can* generate a self-intersecting contour in the image [26]. However, the closure algorithm developed here is not restricted to the recovery of occlusion boundaries. By imposing the constraint of non-intersection, we narrow the class of boundaries which may be recovered. The recovery of more complex boundaries will likely require information about the *type* of structure bounded.

## 11 Results

Figs. 6 and 8 show the bounding contours computed for two different images. For clarity, only cycles for extended tangents over 10 pixels in length are shown, and cycles which share one or more extended tangents are shown in separate images. Note that many overlapping cycles are computed, supporting our earlier observation that a constraint of disjointness is too restrictive for natural images.



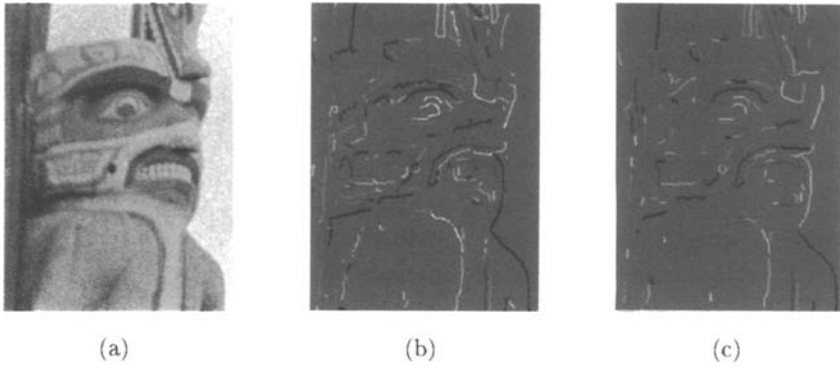
**Fig. 6.** (a) Image of mannequin and shadow. (b-c) Tangent cycles detected in mannequin/shadow image.

Fig. 6 shows the bounding contours computed for the image of the mannequin casting a shadow. The bounding contour of the cast shadow is recovered nearly perfectly. The boundary of the mannequin itself is generally recovered, although the right arm and left forearm are pinched off. The left forearm is recovered as a separate bounding contour in Fig. 6(b), as is the attached shadow on the right arm. Finally, the contour of the “hole” formed by the legs of the mannequin and its shadow is recovered as a bounding contour.

The image of a totem pole shown in Fig. 7(a) (courtesy of David Lowe) poses a greater challenge. Fig. 7(b) shows the edge groupings computed by the multiple hypothesis tracking method of Cox *et al.*, and Fig. 7(c) shows the groupings computed by Lowe’s smoothness criteria [17]. While both methods group edges into extended contours, neither method recovers complete contours which could bound structural units in the image.

The tangent cycles selected as bounding contours by our closure computation are shown in Fig. 8. Major structure boundaries are identified, including the teeth, mouth, eyeball and left shoulder of the human figure and the tongue, lips, eye and eyebrow of the wolf figure, as well as various shadows and decorative markings. An error can be seen in Fig. 8(d), where the eyebrow of the human figure and the lips of the wolf figure have been grouped as a single structure.

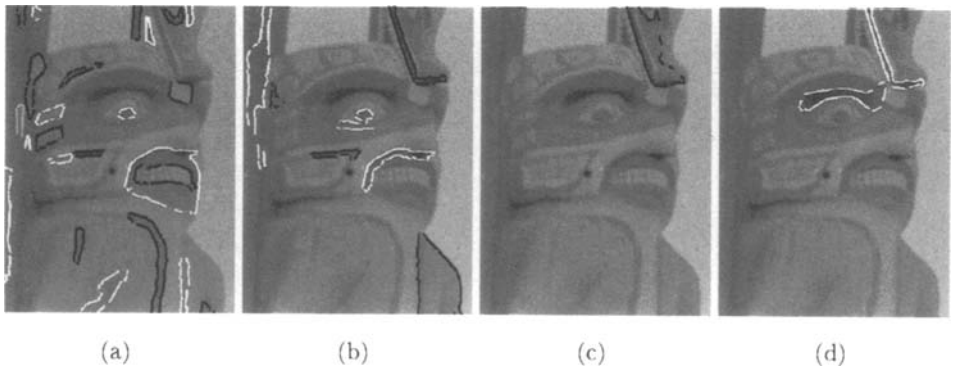
The results of these experiments show that bounding contours can be computed as cycles of tangents in the image. While some errors are made, on the



**Fig. 7.** (a) Noisy image of a totem pole (courtesy of David Lowe). (b) Edges grouped by multiple-hypothesis tracking [2]. (c) Edges grouped by Lowe's smoothness criteria [17].

whole these tangent cycles correspond to the boundaries of two-dimensional structures in the scene: objects, object parts, surface markings and shadows.

Note that closure computations select the tangent cycle bounding the mouth of the human figure in the totem pole (shown in white in Fig. 8(a)), even though the region thus enclosed is highly heterogeneous, encompassing a second closed contour (shown in black). While the many edges within the region of the mouth would cause region-grouping schemes to carve the structure up into smaller com-



**Fig. 8.** Closed contours of totem pole image. The bounding contour of the mouth of the human figure (shown in white in (a)) is recovered despite the heterogeneity of the region it bounds.

ponents, closure computations successfully detect the unitary structure on the basis of regularities in the structure boundary. This may explain in part the recent psychophysical results [8] suggesting a stronger role for boundary cues than regional cues in the perceptual organization of form (Fig. 1(c)).

## 12 Conclusion

Bridging the gap between early visual data structures such as edge maps and higher-level shape representations is a significant challenge for computer vision algorithms. Recent emphasis has been placed on interactive methods which bypass this “weak link” (e.g. [13, 27]), however the development of reliable, fully-automatic methods for contour grouping remains a desirable but elusive goal.

In this paper, we have proposed that the strong global constraint of contour closure may significantly aid in achieving this goal. To support our argument, we have developed an algorithm for computing closed bounding contours as topologically simple cycles of contour tangents. Unlike previous algorithms [11], this closure algorithm does not impose hard constraints on the shape of the image structures to be recovered. Since no constraint of disjointness is imposed, overlapping contours and abutting structures may be computed. Experiments indicate that this closure algorithm generally succeeds in segmenting two-dimensional structures from a variety of real images. While these closure computations do not produce a complete description of the image, they may serve to complement region-grouping methods by extending the class of segmented structures to include heterogeneous structures.

## References

1. T.D. Alter. *The Role of Saliency and Error Propagation in Visual Object Recognition*. PhD thesis, MIT, 1995.
2. I.J. Cox, J.M. Rehg, and S. Hingorani. A Bayesian multiple-hypothesis approach to edge grouping and contour segmentation. *Int. J. Comp. Vision*, 11(1):5–24, 1993.
3. M.P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Englewood Cliffs, NJ, 1976.
4. G. Dudek and J.K. Tsotsos. Recognizing planar curves using curvature-tuned smoothing. In *Proc. 10<sup>th</sup> Int. Conf. on Pattern Recogn.*, Atlantic City, 1990.
5. J. Elder. *The visual computation of bounding contours*. PhD thesis, McGill University, Dept. of Electrical Engineering, 1995.
6. J. Elder and S. W. Zucker. A measure of closure. *Vision Research*, 34(24):3361–3370, 1994.
7. J. Elder and S.W. Zucker. The effect of contour closure on the rapid discrimination of two-dimensional shapes. *Vision Research*, 33(7):981–991, 1993.
8. J. Elder and S.W. Zucker. Boundaries, textures and the perceptual binding of fragmented figures. In *European Conf. on Visual Perception*. Tübingen, Germany, 1995.

9. J. Elder and S.W. Zucker. Local scale control for edge detection and blur estimation. In *Lecture Notes in Computer Science*, New York, 1996. Proc. 4<sup>th</sup> European Conf. on Computer Vision, Springer Verlag.
10. W.T. Freeman. *Steerable Filters and Local Analysis of Image Structure*. PhD thesis, MIT Media Lab, 1992.
11. D.W. Jacobs. Finding salient convex groups. In I.J. Cox, P. Hansen, and B. Julesz, editors, *Partitioning Data Sets*, volume 19 of *DIMACS (Series in Discrete Mathematics and Theoretical Computer Science)*. 1995.
12. G. Kanizsa. *Organization in Vision*. Praeger, New York, 1979.
13. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Proc. 1st Int. Conf. Comp. Vision*, pages 259–268, 1987.
14. K. Koffka. *Principles of Gestalt Psychology*. Harcourt, Brace & World, New York, 1935.
15. I. Kovacs and B. Julesz. A closed curve is much more than an incomplete one: Effect of closure in figure-ground discrimination. *Proc. Natl. Acad. Sci. USA*, 90:7495–7497, 1993.
16. Y.G. Leclerc. Constructing simple stable descriptions for image partitioning. *Int. J. Computer Vision*, 3:73–102, 1989.
17. D.G. Lowe. Organization of smooth image curves at multiple scales. *Int. J. Comp. Vision*, 3:119–130, 1989.
18. M. Nitzberg, D. Mumford, and T. Shiota. Filtering, segmentation and depth. *Lecture Notes in Computer Science*, 662, 1993.
19. P. Parent and S.W. Zucker. Trace inference, curvature consistency, and curve detection. *IEEE Trans. Pattern Anal. Machine Intell.*, 11:823–839, 1989.
20. T. Pavlidis. *Structural Pattern Recognition*. Springer-Verlag, Berlin, 1977.
21. E.M. Riseman and M.A. Arbib. Computational techniques in the visual segmentation of static scenes. *Comp. Graph. Image Proc.*, 6:221–276, 1977.
22. E. Saund. Symbolic construction of a 2-d scale-space image. *IEEE Trans. Pattern Anal. Machine Intell.*, 12(8):817–830, 1990.
23. R. Sedgewick. *Algorithms in C*. Addison-Wesley, Reading, Mass., 1990.
24. A. Sha'ashua and S. Ullman. Structural saliency: The detection of globally salient structures using a locally connected network. In *Proc. 2<sup>nd</sup> Int. Conf. on Computer Vision*, pages 321–327, Tampa, Florida, 1988. IEEE Computer Soc. Press.
25. A. Treisman and S. Gormican. Feature analysis in early vision: Evidence from search asymmetries. *Psychol. Rev.*, 95:15–48, 1988.
26. L.R. Williams. *Perceptual completion of occluded surfaces*. PhD thesis, University of Massachusetts, Amherst, Mass., February 1994.
27. A. Yuille and P. Hallinan. Deformable templates. In A. Blake and A. Yuille, editors, *Active Vision*. MIT Press, Cambridge, Mass., 1992.
28. S.W. Zucker. The diversity of perceptual grouping. In M. Arbib and A. Hanson, editors, *Vision, Brain and Cooperative Computation*, pages 231–261. MIT, Cambridge, Mass., 1986.