April, 1998

LIDS- P 2415

Research Supported By:

MURI grant DAAHO4-96-1-0445

A Perceptual Organization Approach to Contour Estimation via Composition, Compression and Pruning of Contour Hypotheses

Casadei, S.

Mitter, S.

A perceptual organization approach to contour estimation via composition, compression and pruning of contour hypotheses *

Stefano Casadei, Sanjoy Mitter

Laboratory for Information and Decision Systems

Massachusetts Institute of Technology

Cambridge, Massachusetts 02139

e-mail: casadei@lids.mit.edu

April 1, 1998

Abstract

The problem of estimating scene contours by means of perceptual organization is formulated in a probabilistic fashion. The goal of estimation is to compute a set of contour descriptors which approximate every scene contour with high probability. A hierarchy of contour descriptors designed for this purpose is proposed. Computation at each level of the hierarchy consists of three basic ingredient: hypothesis generation by means of grouping; hypothesis evaluation and pruning; and, where necessary, compression of equivalent multiple responses. A specific algorithm to enumerate, prune and

*Research supported by MURI grant DAAH04-96-1-0445, Foundations of Performance Metrics for Object Recognition.

compress contour-cycles in a graph is proposed. This algorithm can easily incorporate feedback loops to prune the search which exploit global information.

1 Introduction

1.1 Previous work

Estimation of the contours of the objects in a visual image, based only on object-independent (generic) information, is an important ingredient of many image analysis applications. For certain classes of images, an important source of information to perform this task is given by the sharp variations of brightness perpendicular to the edge which often occur in the vicinity of the boundaries between objects. However, estimators based solely on this type of information are known to produce fragmented and incomplete contour representations. To overcome this limitation, the human visual system exploits certain geometrical relationships which typically exist between contour fragments belonging to the same object. By detecting these relationships, human perception is able to infer information about the scene not obtainable from brightness variations alone. The Gestalt psychologists [16] have documented several of these geometrical properties: proximity, continuity, similarity, closure, symmetry. In the last fifteen years, computer vision algorithms have been proposed to detect these geometrical relationships in order to aggregate contour fragments into more complete and more global contour representations. This grouping process, often referred to as perceptual organization, is based on the statistical assumption that these geometric relationships between contour fragments are much more likely to occur when these fragments indeed belong to the same contour (non-accidentalness principle) [33, 18]. A good review and classification scheme for perceptual organization work in computer vision can be found in [25].

The difficulty of detecting geometric relationships for perceptual organization and contour estimation varies greatly according to the scale at which these relationships manifest themselves. Proximity (by definition) occurs at a small scale and is quite easy to incorporate into contour estimation. Smoothness is also a quite local property. Co-linearity and co-circularity can occur both at local and a global scale depending on the size of the gap between contour fragments. Many contour estimation algorithms exploit this type of relationship [23, 21, 22, 13, 12, 17]. Convexity and closure are global properties of contours [15, 9], as is symmetry [5, 6]. Still at a higher level, one has properties involving the occlusion relationship between sets of contours [21, 22, 32, 10, 31]. Since the geometric relationships used for perceptual organization provide a context to reduce contour uncertainty, the scale at which these relationship can be detected will be called *context scale.*

The descriptors computed by perceptual organization can be organized into a hierarchy according to context scale. At the bottom level sits the brightness image whose measurements span one single pixel. Then, one has point-like contour descriptors obtained by detecting brightness variations in small neighborhoods. These point-like descriptors can be composed into extended contours by using proximity, co-linearity, etc. At the next stage, these contours can be composed into ribbons by detecting symmetry or parallelism, and so on. Several contour hierarchies of this type, generated by means of recursive grouping of contour fragments, have been proposed [19, 26, 8, 20, 3]. Hierarchical representations provide a means to resolve contour uncertainties with the necessary context information by propagating these uncertainties up to the level where the context scale is sufficiently large to resolve them reliably [8, 3].

An aspect shared by most perceptual organization algorithms is that hard decision, if any, are taken only after a substantial amount of distributed "soft" information has been collected and processed. For instance, the Hough transform and similar methods collect votes in a suitable contour parameter space and then detect the most voted contours [12]. Voting methods can also be used to detect sets of contour fragments with similar attributes [26]. Some approaches estimate explicitly the probabilities of contour hypotheses and use these estimates to focus the search [24, 7, 5, 11, 9]. Dynamic programming can be used to detect contours which minimize a cost function given by a sum of local contributions [27, 29, 2]. Relaxation labeling is a powerful iterative technique to propagate and accumulate soft information [23]. Convolution with suitable kernels is a biologically motivated approach for accumulating evidence of contours [13, 17]. Recently, a method based on spectral graph theory has been proposed whereby the similarity between image regions is encoded in a soft fashion into the weights of a graph before computing a segmentation of the image [28, 17]. Approaches motivated by statistical physics encode the geometric relationship between contour fragments into coupling constants between interacting entities and then formulate perceptual organization as a combinatorial optimization problem which can be solved by using techniques borrowed from statistical mechanics [14, 10]. A combinatorial formulation by means of integer programming has also been proposed [30, 31].

1.2 Contributions of this paper

Ideally, a contour estimation algorithm should compute a set of contour descriptors which contains an accurate approximation to every contour in the scene. Also, it would be desirable to have probability estimates associated to these contour descriptors. A difficulty with perceptual organization is that these requirements can easily result in problems of combinatorial complexity, unless appropriate assumptions or restrictions are made. To our knowledge, very few algorithms have been designed by requiring that all the contours in a certain class are estimated correctly with high probability [15, 5, 4, 3].

A perceptual organization algorithm whose goal is to compute a *complete* representation, namely one which approximates all contours in a certain class with high probability, has to somehow search exhaustively the combinatorial space given by all possible groupings of contour fragments. Thus, to control computational complexity, it is necessary to design ways to focus the search without sacrificing completeness.

The basic method to control the complexity of the search, which is widely used, is to prune hypotheses with low probability. The way this is done depends on the structure of the specific algorithm. For instance, in a tree search one can prune nodes with low probability.

A cause of combinatorial explosion of the search which is often overlooked is given by the presence of multiple responses to the same contour. In fact, it is usually the case that in order to ensure completeness of the final representation one has to allow for the possibility that certain portions of the contours are represented locally by multiple contour fragments. Pruning out some of these fragments based only on local information can result in false negatives. Therefore, one has to carry these redundancies through the grouping process up to a point where it becomes possible to *compress* the set of multiple responses without disrupting the construction of a global approximating descriptor of the contour [2, 3].

In this paper we describe the contour hierarchy which we have been studying in the last few years and which consists of two graphs. At the lower level we have an edgel-graph which is used to compute polygonal approximations of regular visible contours (Section 3). These contours are then used to construct a contour-graph (Section 4) whose arcs can represent "invisible" contour fragments, namely fragments across which brightness variations are small. We propose an algorithm to search efficiently all closed and maximal contours in this graph (Section 5). This algorithm allows to incorporate rather arbitrary probabilistic information at any context scale. Any function defined on the collection of contour-paths can be used to guide the search (compare for instance with dynamic programming algorithms which assume that the function is a sum of local terms). Multiple responses are detected and compressed without causing false negatives. Under appropriate assumptions, this algorithm computes a complete representation of the contours with high probability.

2 The problem of contour estimation

Let us consider the problem of generating a set of contour descriptors of minimal complexity such that, with high probability, every scene contour is approximated by at least one contour descriptor in the computed representation. The set of scene contours, namely the set of "true" planar contours between objects in the scene, is a random variable denoted Γ. This random variable takes values in the collection of subsets of some class of contours $\mathbf{\Gamma}$, that is, $\Gamma \subset \mathbf{\Gamma}$. Section 2.3 describes some contour classes $\mathbf{\Gamma}$. The set of *contour fragments*, denoted $\overline{\Gamma}$, is given by the set of all sub-contours of Γ. The observed image I is a random variable jointly distributed with Γ. The contour estimation algorithm computes from I a set of contour descriptors $\alpha(I) \subset \hat{\mathbf{\Gamma}}$. Here, $\hat{\mathbf{\Gamma}}$ is some class of contour descriptors dense in $\mathbf{\Gamma}$. That is, for every $\gamma \in \mathbf{\Gamma}$ and $\epsilon > 0$ there exists $\hat{\gamma} \in \hat{\mathbf{\Gamma}}$ such that $d(\gamma, \hat{\gamma}) < \epsilon$, where $d : \mathbf{\Gamma} \times \hat{\mathbf{\Gamma}} \to [0, \infty]$ denotes a distance function. Also, let $\hat{d} : \hat{\mathbf{\Gamma}} \times \hat{\mathbf{\Gamma}} \to [0, \infty]$ be a distance function defined on the class of contour descriptors.

Let $P_\epsilon(\hat{\gamma}|I)$ be the conditional probability that there exists $\gamma \in \overline{\Gamma}$ such that $d(\gamma, \hat{\gamma}) < \epsilon$ given that the image I is observed. That is, $P_\epsilon(\hat{\gamma}|I)$ is the conditional probability that the scene contains a contour fragment ϵ-near to $\hat{\gamma}$. The map $(\hat{\gamma}, I) \mapsto P_\epsilon(\hat{\gamma}|I)$ is a characterization of the estimation system under consideration, namely it describes the relationship between the three components of the system: the set of scene contours, the observed image and the contour approximators.

2.1 A brute-force algorithm for contour estimation

If the map $(\hat{\gamma}, I) \mapsto P_\epsilon(\hat{\gamma}|I)$ is known then the following *brute-force* algorithm can be used to compute a set of contour descripors which approximates every scene contour with guaranteed confidence.

1. Compute an ϵ-covering $\hat{\mathbf{\Gamma}}_\epsilon \subset \hat{\mathbf{\Gamma}}$ of $\mathbf{\Gamma}$. That is, for every $\gamma \in \mathbf{\Gamma}$ there exists $\hat{\gamma} \in \hat{\mathbf{\Gamma}}_\epsilon$ such that $d(\gamma, \hat{\gamma}) < \epsilon$ (this step of the algorithm is independent of I).

2. For each $\hat{\gamma} \in \hat{\mathbf{\Gamma}}_\epsilon$ evaluate the probability $P_\epsilon(\hat{\gamma}|I)$. Let $\hat{\Gamma}$ be the set of all $\hat{\gamma} \in \hat{\mathbf{\Gamma}}$ such that $P_\epsilon(\hat{\gamma}|I)$ is greater than a threshold δ.

3. Compute a minimal ϵ-sampling $\hat{\Gamma}^*$ of $\hat{\Gamma}$. That is, $\hat{\Gamma}^*$ is the smallest complexity subset of $\hat{\Gamma}$ with the following property: for every $\hat{\gamma} \in \hat{\Gamma}$, there exists $\hat{\gamma}^* \in \hat{\Gamma}^*$ such that $\hat{d}(\hat{\gamma}, \hat{\gamma}^*) < \epsilon$.

Let $|\hat{\mathbf{\Gamma}}_\epsilon|$ denote the cardinality of $\hat{\mathbf{\Gamma}}_\epsilon$.

Lemma 1 *The set of descriptors $\hat{\Gamma}$ computed by the second step of the brute-force estimation algorithm is an ϵ-covering of Γ with probability at least $1 - |\hat{\mathbf{\Gamma}}_\epsilon|\delta$.*

Proof. Let $B_\epsilon(\hat{\gamma})$ be the set of $\gamma \in \mathbf{\Gamma}$ such that $d(\gamma, \hat{\gamma}) < \epsilon$. If $\hat{\Gamma}$ is not an ϵ-covering of Γ then there exists $\gamma \in \Gamma$ such that

$$\gamma \notin \bigcup_{\hat{\gamma} \in \hat{\Gamma}} B_\epsilon(\hat{\gamma}).$$

That is, since

$$\gamma \in \bigcup_{\hat{\gamma} \in \hat{\mathbf{\Gamma}}_\epsilon} B_\epsilon(\hat{\gamma}) = \mathbf{\Gamma},$$

if $\hat{\Gamma}$ is not an ϵ-covering of Γ, then there exists $\hat{\gamma} \in \hat{\mathbf{\Gamma}}_\epsilon \setminus \hat{\Gamma}$ such that $\gamma \in B_\epsilon(\hat{\gamma})$ and therefore $\Gamma \cap B_\epsilon(\hat{\gamma}) \neq \emptyset$. Thus, the event "$\hat{\Gamma}$ is not an ϵ-covering of Γ" is contained in the union over all $\hat{\gamma} \in \hat{\mathbf{\Gamma}}_\epsilon$ of the events "$\Gamma \cap B_\epsilon(\hat{\gamma}) \neq \emptyset$ and $\hat{\gamma} \notin \hat{\Gamma}$". The conditional probability given I of each of these events is at most δ. Thus, by using the union bound, the probability given I that $\hat{\Gamma}$ is not an ϵ-covering of Γ is at most $|\hat{\mathbf{\Gamma}}_\epsilon|\delta$. □

Theorem 1 *If for any $\gamma \in \mathbf{\Gamma}$ and $\hat{\gamma}_1, \hat{\gamma}_2 \in \hat{\mathbf{\Gamma}}$,*

$$d(\gamma, \hat{\gamma}_2) \leq d(\gamma, \hat{\gamma}_1) + \hat{d}(\hat{\gamma}_1, \hat{\gamma}_2), \tag{1}$$

then the set of descriptors $\hat{\Gamma}^*$ computed by the brute-force estimation algorithm is a 2ϵ-covering of Γ with probability at least $1 - |\hat{\mathbf{\Gamma}}_\epsilon|\delta$.

Proof. Lemma 1 ensures that, with probability at least $1-|\hat{\mathbf{\Gamma}}_\epsilon|\delta$, for every $\gamma \in \Gamma$ there exists $\hat{\gamma} \in \hat{\Gamma}$ such that $d(\gamma, \hat{\gamma}) < \epsilon$. Since $\hat{\Gamma}^*$ is an ϵ-sampling of $\hat{\Gamma}$, there exists $\hat{\gamma}^* \in \hat{\Gamma}^*$ such that $\hat{d}(\hat{\gamma}, \hat{\gamma}^*) < \epsilon$. Then, from assumption (1), $d(\gamma, \hat{\gamma}^*) < 2\epsilon$. □

It should be noted that Lemma 1 and Theorem 1 still hold if the set of scene contours Γ is replaced with $\overline{\Gamma}$. That is, the brute force algorithm computes a 2ϵ-covering of the set of all contour fragments in the scene with probability at least $1-|\hat{\mathbf{\Gamma}}_\epsilon|\delta$. Notice also that the distance function $\hat{d}$ is not required to be symmetric. For instance, one can define $\hat{d}$ in such a way that whenever $\hat{\gamma}_1$ is a sub-contour of $\hat{\gamma}_2$, then $\hat{d}(\hat{\gamma}_1, \hat{\gamma}_2) = 0$. For instance, one can let $\hat{d}(\hat{\gamma}_1, \hat{\gamma}_2)$ be the directed Hausdorff distance from $\hat{\gamma}_1$ to $\hat{\gamma}_2$ (see Section 3.3). This property, together with assumption (1), guarantees that a contour descriptor $\hat{\gamma}$ which is an accurate approximation of a scene contour, is also a good approximation of any fragment of this scene contour.

The three steps of the brute-force algorithm represent three important components of estimation, namely hypothesis generation, hypothesis evaluation and compression. These are the basic ingredients of the contour estimation algorithms described in the rest of the paper.

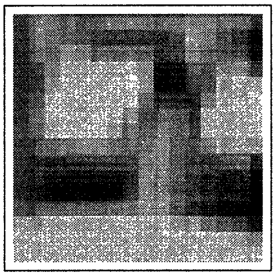
2.2 The hierarchical approach

Thanks to certain properties of the estimation system, there exist more efficient methods to solve the problem than the brute force algorithm just described. Hierarchical representations and perceptual organization provide one such methodology. Roughly speaking, the basic idea is to construct a nested sequence of contour descriptor classes of increasing complexity: $\hat{\mathbf{\Gamma}}_1 \subset \hat{\mathbf{\Gamma}}_2 \subset \cdots \subset \hat{\mathbf{\Gamma}}_n$. The hierarchy which we have been developing during the last few years consists of the following contour descriptors: [brightness data] $\rightarrow$ gradient vectors $\rightarrow$ edgels $\rightarrow$ edgel-arcs $\rightarrow$ edgel-paths $\rightarrow$ contour primitives $\rightarrow$ contour-arcs $\rightarrow$ contour-paths. Fig. 1 illustrates the layers of this hierarchy computed from a 16×16 image fragment of a telephone keyboard. A coarser characterization of this hierarchy is obtained by viewing these layers as different stages of the computation of two graphs, the *edgel-graph* (gradient vectors, edgels, edgel-arcs, edgel-paths), and the *contour-graph* (contour primitives, contour-arcs, contour-paths). The first graph feeds into the second, since contour primitives, which are the vertices of the contour-graph, are polygonal contours obtained by splitting the edgel-paths computed in the edgel-graph.

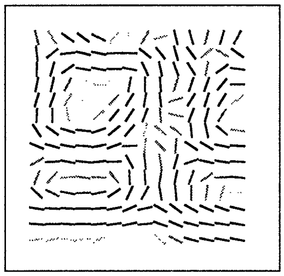
Each "dictionary" $\hat{\mathbf{\Gamma}}_i$ yields a contour estimation problem and this hierarchy of problems is solved in a bottom-up fashion. At each level, the same basic ingredients are used to solve the estimation problem. These ingredients correspond to the three steps of the brute force algorithm described earlier: hypothesis generation, hypothesis evaluation and, if applicable, "metric" compression. In the proposed contour hierarchy, metric compression is applied to the edgel-path and contour-path levels due to the arbitrary cardinality of these composite descriptors which leads to a combinatorial explosion of the number of hypotheses due to the presence of multiple responses to the same contour. In most layers, hypothesis generation (enumeration) is performed by composing/grouping the descriptors of the previous levels. Contour primitives are an exception since they are obtained by means of splitting. The descriptors of the bottom level $\hat{\mathbf{\Gamma}}_1$ (gradient vectors) are simple enough that they can be explicitly enumerated efficiently. As for hypothesis evaluation, since the map $(\hat{\gamma}, I) \mapsto P_\epsilon(\hat{\gamma}|I)$ is usually not available in practice, these probabilities are estimated by constructing probabilistic models of the composed descriptors in terms of the geometric and photometric relationships of their constituents. The reason for having so many layers in the hierarchy is to have as many entry points as possible for complexity-controlling feedback loops, such as hypothesis evaluation via probability estimates and "safe" elimination of redundant hypothesis (metric compression).

2.3 Classes of contours

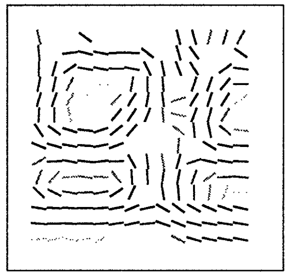
The projection on the image plane of a scene contour γ can be represented by a planar curve. The image of this curve is a subset of the real plane and will be called the *trace*



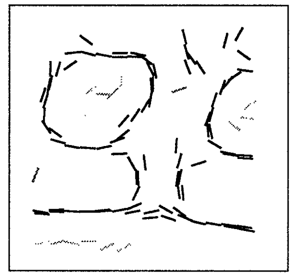
(a) Brightness data



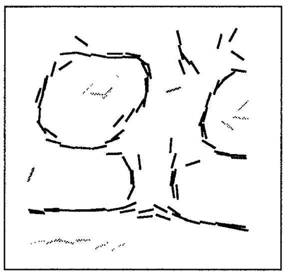
(b) Gradient vectors



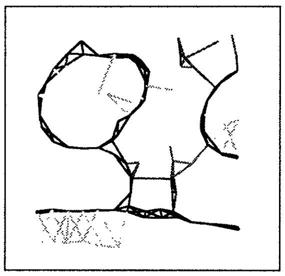
(c) Gradient vectors (pruned)



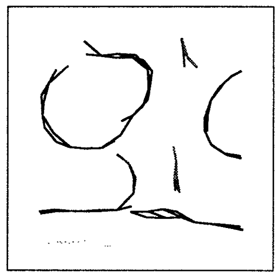
(d) Edgels



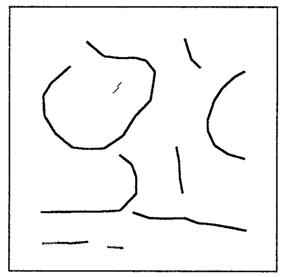
(e) Edgels (pruned)



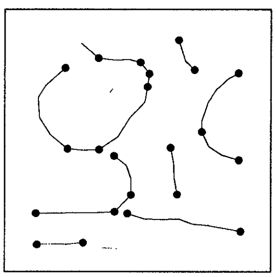
(f) Edgel-arcs



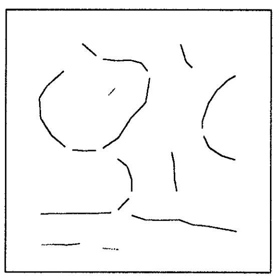
(g) Edgel-arcs (pruned)



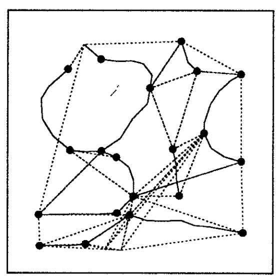
(h) Edgel-paths



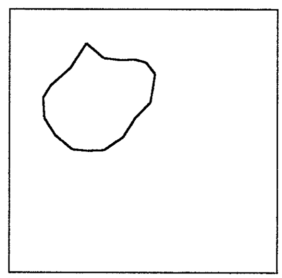
(i) Emission points



(j) Contour primitives



(k) Contour-arcs



(l) Contour-cycles

Figure 1: Hierarchy of contour descriptors. The edgel-graph from which edgel-paths are computed is shown in Fig. 1(f). The contour-graph from which contour-cycles are computed is shown in Fig. 1(k).

of γ and denoted $T(\gamma)$. The simplest model for a scene contour, called *flat* contour, is given by a contour with zero curvature, infinite length and brightness model constant along the contour. The set of flat contours is denoted $\mathbf{\Gamma}_0$.

For simplicity, we focus on contour brightness models given by a blurred step discontinuity. Such contours are characterized by the following attributes: the orientation of the contour with respect to the x-axis; the distance of the contour from the origin; the brightness intensities on the left and right sides of the contour, denoted $\gamma.b_1$ and $\gamma.b_2$; the blur scale of the one dimensional discontinuity profile across the contour, $\gamma.s$; the intensity of the noise (e.g. its variance), denoted $\gamma.\nu$. The blurred step brightness model for a flat contour is given by:

$$I(i,j) = \beta(i,j|\gamma) + \eta(i,j), \qquad (i,j) \in D(\gamma) \tag{2}$$

where

$$\beta(i,j|\gamma) = \gamma.b_1 + (\gamma.b_2 - \gamma.b_1) \cdot \mathrm{erf}\left(\frac{\xi(i,j)}{\gamma.s}\right); \tag{3}$$

$\xi(i,j)$ is the signed distance from $T(\gamma)$ to the the pixel (i,j); η is a noise field assumed to be i.i.d. and gaussian with variance $\gamma.\nu$; $\mathrm{erf}(\cdot)$ is the error function given by

$$\mathrm{erf}(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{1}{2}u^2} du;$$

$D(\gamma) \subset \mathbb{Z} \times \mathbb{Z}$ is the *domain* of γ, namely the set of image measurement affected by the presence of the contour γ in the scene. The signal-to-noise ratio of γ is given by $|\gamma.b_1 - \gamma.b_2|/\gamma.\nu$. If this is large enough then the contour is said to be *visible*.

A more general class of contours, denoted $\mathbf{\Gamma}_1$ and called *quasi-flat* contours, is given by contours with bounded curvature and with b_1, b_2, s, ν slowly varying along the contour. Notice that the precise definition of $\mathbf{\Gamma}_1$ depends on the upper bound on curvature and the upper bounds on the rate of variation of b_1, b_2, s, ν. Finally we denote by $\mathbf{\Gamma}_2$ the contours obtained by composing a sequence of quasi-flat visible contours. Contours in $\mathbf{\Gamma}_2$, which are called *composite*, can contain discontinuities such as corners and invisible portions (see Fig. 2).

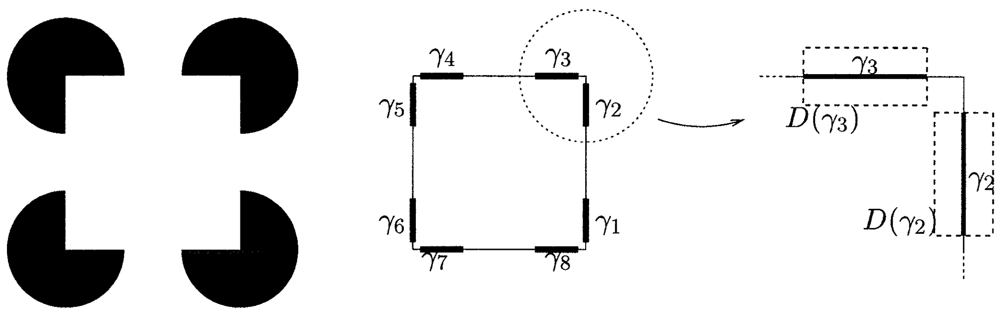


Figure 2: The contour of the illusory square floating on top of the four black disks is composed of eight flat visible contours. Notice that the domains of these contours are pairwise disjoint.

3 The edgel-graph

This section briefly describes the computation involving the edgel-graph which results in a representation of the quasi-flat visible contours in the scene by means of polygonal lines. This includes the lowest four levels of the hierarchy: gradient vectors, edgels, edgel-arcs and edgel-paths. A more complete description of this work can be found in [2, 3, 4].

3.1 Gradient vectors

Gradient vectors are obtained by estimating the gradient of the brightness intensity in square blocks of pixels. A gradient vector g is characterized by six parameters (attributes): $g = \langle i, j, n, \theta, \rho, \nu \rangle$. The integers i, j, n specify the square block of pixels used to estimate the gradient vector. The pair (i, j) specifies the location of the center of the block and n its size. θ denotes the perpendicular direction to the estimated brightness gradient and ρ its magnitude. Finally, ν is the L_2-norm of the residual of the linear fit. The attributes of a gradient vector g will also be denoted $g.i$, $g.j$, etc.

A collection of gradient vectors is computed by least-square fitting a linear function to the brightness data in each block of pixels of size n, where n varies in some set. In the current implementation only one size, $n = 3$, is used. Fig. 1(c) shows the gradient vectors computed from the data in Fig. 1(a). If the center i, j of the block is sufficiently close to a visible scene contour and if the blurred step model (2), (3) is assumed to be valid, then, by expanding (3) into its Taylor expansion, the data inside the block can be modeled as a linear function plus gaussian noise. If the signal-to-noise ratio (SNR) of the scene contour is sufficiently high then, with high probability, the SNR of the estimated gradient vector g, given by $\frac{g.\rho}{g.\nu}$, will also be high. By using the above model for the data, one can convert the SNR of g into an estimate of the error of the edge orientation. A threshold is then applied to the estimated orientation error obtained in this way. In Fig. 1(c) the gradient vectors with estimated orientation error less than 30° are shown.

3.2 Edgels

For each gradient vector g, a set of edgels is estimated by fitting a cubic polynomial constant in the direction of g to the brightness data in a rectangular region of size $l_{||}$ in the direction of g and $2l_{\perp}$ in the direction perpendicular to g. In general, several edgels can be obtained from the same gradient vector by using different values of $l_{\perp}$ and $l_{||}$. In the current implementation, only one value is used, $l_{\perp} = 2.0$, $l_{||} = 3.0$. As in the estimation of gradient vectors, it is assumed that within the fitting region the blurred step model (3) can be approximated by its Taylor expansion of the appropriate order (third order for edgels).

From the four estimated parameters of the least-square fitted cubic polynomial one can compute estimates of the position p of the contour; the blur scale s and the brightness values on the two sides of the contour, b_1 and b_2. Thus, an edgel e is specified by the following attributes: $e = \langle i, j, n, l_{\perp}, l_{||}, p, \theta, s, b_1, b_2, \nu \rangle$. The parameter $p \in \mathbb{R}^2$ is the estimated projection of (i, j) onto the contour and ν is the L_2-norm of the residual of the cubic fit. Similarly to gradient vectors, the set of edgels is pruned by computing the expected error of some of its attributes and by thresholding out edgels with large errors. This guarantees that the edgels computed from regions near to a scene contour which is quasi-flat and visible are retained with high probability.

3.3 Edgel-arcs and edgel-paths

An edgel-arc is a pair of edgels (e_1, e_2) such that $(e_1.i, e_1.j)$ and $(e_2.i, e_2.j)$ are nearest neighbors. For any edgel-arc $a = (e_1, e_2)$, let $\sigma(a)$ denote the straight line segment with end-points $e_1.p, e_2, p$. The set of all segments $\sigma(a)$ computed from Fig. 1(a) is shown in Fig. 1(f).

Pruning is more complicated for edgel-arcs than it is for the previous levels. In fact, the purpose of pruning edgel-arcs is not just to reduce the number of hypotheses but also to ensure that a complete set of paths can be computed efficiently in the associated edgel-graph. To see why computing a complete set of edgel-paths is not an easy task, notice that an edgel-path is a composite descriptor containing an arbitrary number of edgel-arcs. Thus, the total number of edgel-paths in the graph is combinatorially large (see Fig. 1(f)).

An approach to deal with this difficulty is to remove some edgel-arcs so that the remaining subgraph does not contain *divergent bifurcations* [2] or, similarly, so that it is *stable* [3] or *compressible* [4]. Roughly speaking, an edgel-graph is said to be compressible if any two paths between any two vertices are close to each other. In a compressible graph it is possible to compute efficiently a small set of paths which approximate every other path. Fig. 1(g) shows the compressible graph computed from Fig. 1(f).

Clearly, the hard part of this approach is to prune the edgel-graph into a compressible graph in such a way that there remains at least one approximating path for every scene contour satisfying certain sufficient conditions. A class of scene contours which satisfies these sufficient conditions are quasi-flat visible contours, that is, contours with no singularities (corners or junctions) and with sufficiently large signal to noise ratio.

It can be proved that a set of edgel-paths $\hat{\Gamma}^*_{\text{ed.p.}}$ can be computed which approximates each quasi-flat visible scene contour with high probability [4]. More precisely, if γ is

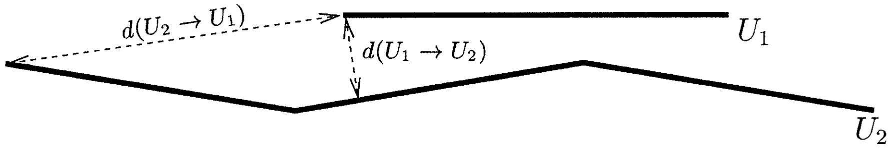


Figure 3: The directed Hausdorff distance from U_1 to U_2, denoted $d(U_1 \to U_2)$, is the maximum distance from a point in the set U_1 to the set U_2. If U_1 and U_2 are two polygonal lines and $d(U_1 \to U_2)$ is small but $d(U_2 \to U_1)$ is large, then U_2 is a *covering* approximation of U_1 which is in general "longer" than U_1.

a scene contour, then with high probability (which depends on the total length of the contours and their signal-to-noise ratio) there exists $\hat{\gamma} \in \hat{\Gamma}^*_{\text{ed.p.}}$ such that

$$d(T(\gamma) \to T(\hat{\gamma})) < \epsilon,$$

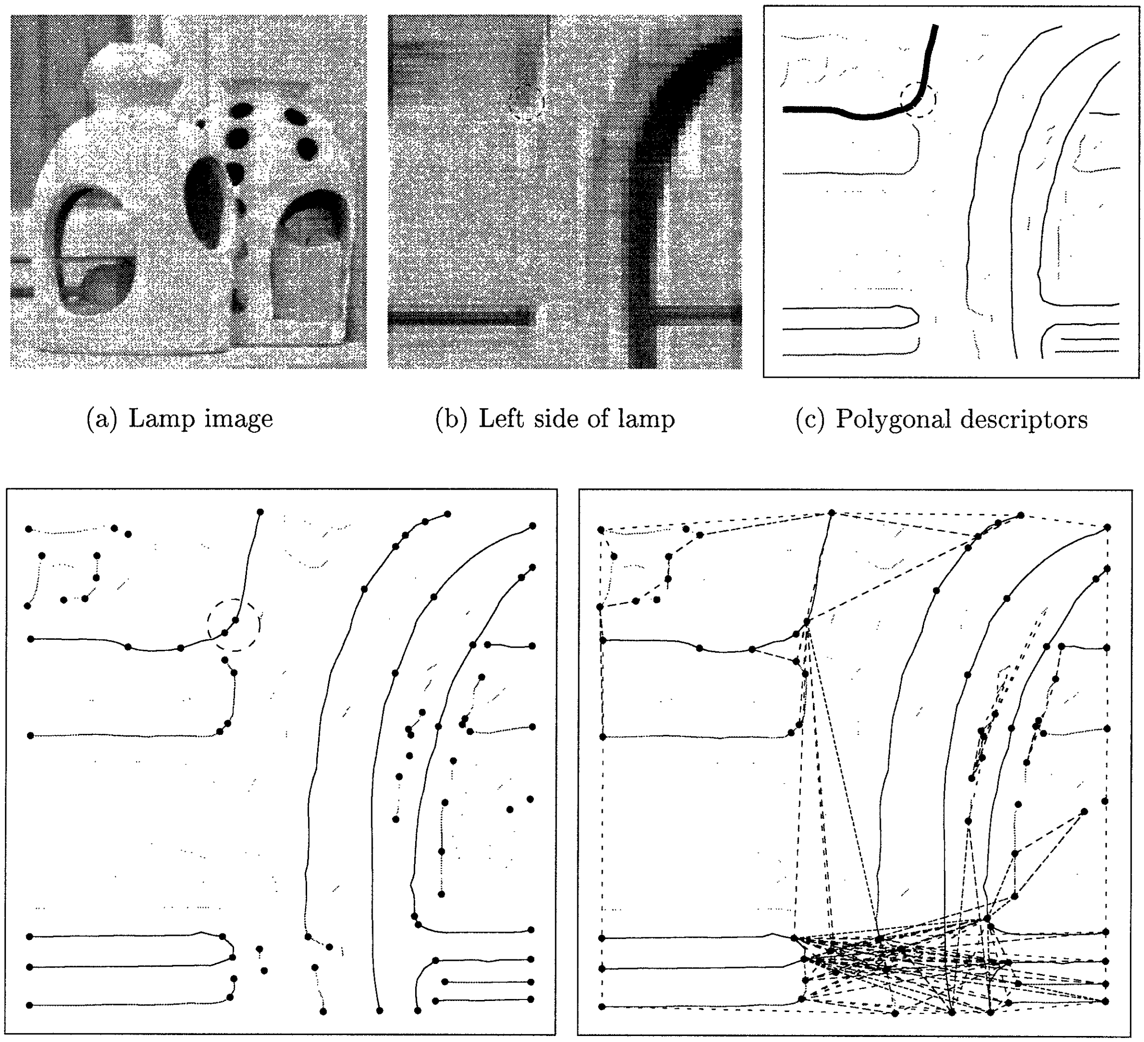
where ϵ is a small constant and $d(T(\gamma) \to T(\hat{\gamma}))$ is the directed (asymmetric) Hausdorff distance from $T(\gamma)$ to $T(\hat{\gamma})$ defined by (see Fig. 3):

$$d(U_1 \to U_2) = \max_{p_1 \in U_1} d(p_1 \to U_2) = \max_{p_1 \in U_1} \min_{p_2 \in U_2} ||p_1 - p_2||. \tag{4}$$

It should be noted that the distance in the other direction, $d(T(\hat{\gamma}) \to T(\gamma))$ is arbitrary, which implies that, roughly speaking, the computed contour descriptor can continue beyond the end-points of the scene contour.

4 The contour-graph

The set of edgel-paths $\hat{\Gamma}^*_{\text{ed.p.}}$ computed from the edgel-graph is used to construct a contour-graph whose vertices are given by contour primitives obtained by splitting edgel-graph at points where local characteristics of the contour (such as orientation and brightness) vary rapidly. Ultimately, scene contours will be represented by a set of loops computed from the contour-graph.



(a) Lamp image (b) Left side of lamp (c) Polygonal descriptors

(d) Emission points (e) Contour-arcs

Figure 4: The highlighted polygonal line in (c) needs to be split and composed with other contour primitives in order to reconstruct the whole contour of the lamp.

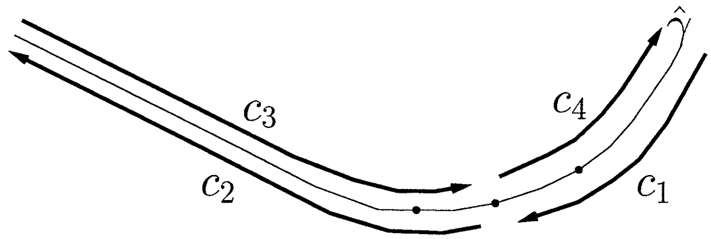


Figure 5: An edgel-path polygonal descriptor $\hat{\gamma} \in \hat{\Gamma}^*_{\text{ed.p.}}$ which generates four contour primitives. Three adjacent emission points are present on $\hat{\gamma}$ and $\hat{\gamma}$ is cut in correspondence of the middle one.

4.1 Contour primitives

Quasi-flat visible contours can be used to compose more general contours models containing corners, junctions and invisible portions [1] (see Fig. 2). However, since the polygonal approximator $\hat{\gamma} \in \hat{\Gamma}^*_{\text{ed.p.}}$ of a regular visible contour γ computed from the edgel-graph might be "longer" than γ (see Fig. 3), it is necessary to consider each point on $\hat{\gamma}$ as a candidate end-point of γ. For instance, consider the marked T-junction in Fig. 4(b). This junction sits at the intersection between the left side of the lamp and the top edge of the white strip in the background (which is partly occluded by the lamp). Notice that a portion of lamp contour and a portion of the strip contour have been merged into the same polygonal contour descriptor (highlighted in Fig. 4(c)) and that the junction corresponds to a high curvature section of this polygonal line. Clearly, the portion which approximates the lamp contour needs to be detached from the other contour fragment and composed with pieces of other polygonal descriptor in order to reconstruct the whole contour of the lamp. In order to do this it is necessary to hypothesize an end-point in the neighborhood of the marked corner of the polygonal descriptor.

Fig. 4(d) shows all the points which have been hypothesized as candidate end-points of regular visible contours by the current implementation of the algorithm. These points are called *emission* points since the following stage of the algorithm searches for good contour continuations by "emitting" lines from these points. Notice that two emission

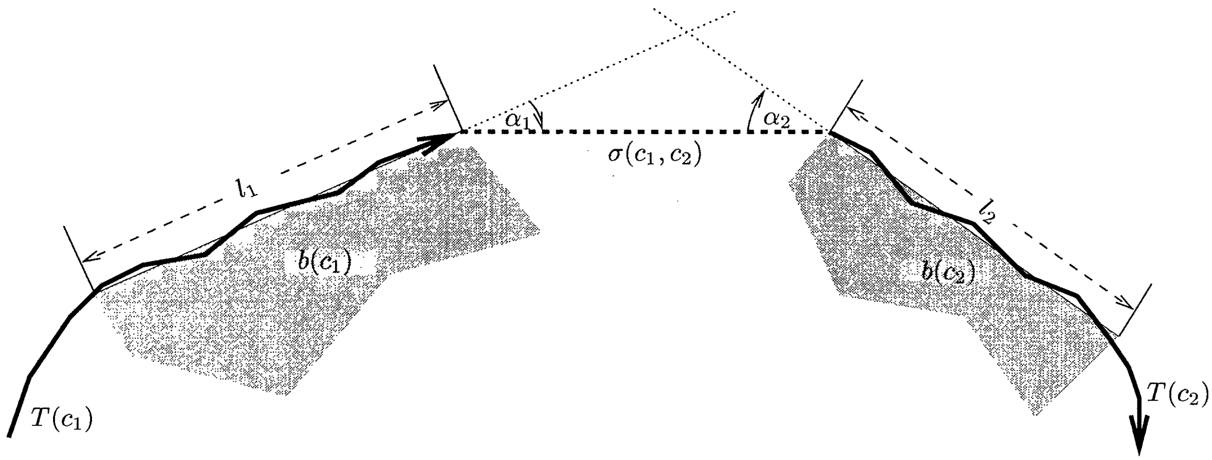


Figure 6: A contour-arc $a = (c_1, c_2)$. Its trace $T(a)$ is given by $T(c_1) \cup \sigma(c_1, c_2) \cup T(c_2)$. The estimated probability $\hat{P}_\epsilon(a)$ depends on l_1, l_2, α_1, α_2, and the brightness values $b(c_1)$ and $b(c_2)$.

points exist in correspondence of the marked T-junction. In order to be considered as an emission point, a point on a polygonal descriptor must exhibit rapid change in some local attributes of the contour such as orientation or brightness values. Most of the emission points in Fig. 4(d) correspond to high curvature points of the contour. Emission points are also hypothesized where the brightness contrast of the contour descriptor varies rapidly. In fact, this contrast change is an indication that the descriptor might continue beyond a scene contour and into a homogeneous region (because of noise) or that the brightness on one of the two sides of the contour has changed.

Small sets of adjacent emission points on a polygonal descriptor are clustered together and the descriptor is cut at the center of the cluster (see Fig. 5). Two *contour primitives* are created for every fragment of a polygonal descriptor, one primitive for each side of the descriptor. Thus a contour which separates two regions in the image is represented by two contour primitives, one for each region. The set of all contour primitives obtained in this way is denoted C.

4.2 Contour-arcs and the contour-graph

A *contour-arc* is a pair $a = (c_1, c_2) \in C \times C$ representing the hypothesis that the scene contains a contour near the descriptor obtained by concatenating c_1 with c_2. If, for the sake of simplicity, only linear interpolation between c_1 and c_2 is considered, then the polygonal line associated with the contour-arc a is given by $T(c_1) \cup \sigma(c_1, c_2) \cup T(c_2)$ where $T(c_1)$, $T(c_2)$ are the polygonal lines associated with c_1 and c_2 and $\sigma(c_1, c_2)$ is the straight-line segment connecting the head point of $T(c_1)$ to the tail point of $T(c_2)$ (see Fig. 6). Similarly, the polygonal line $T(\pi)$ associated with a *contour-path* π, that is a path in the *contour-graph* $(C, C \times C)$, is given by $T(c_1) \cup \sigma(c_1, c_2) \cup T(c_2) \cdots \cup T(c_n)$, where $c_1, \ldots, c_n$ are the vertices of π.

For any $\gamma \in \mathbf{\Gamma}$ and contour-arc a, let $d(\gamma, a)$ be the distance between γ and a given by:

$$d(\gamma, a) = \min_{\mu \in H(\gamma, a)} \max_{p \in T(\gamma)} ||p - \mu(p)||,$$

where $H(\gamma, a)$ is the set of all homeomorphisms between $T(\gamma)$ and $T(a)$. Notice that $d(\gamma, a)$ is greater or equal to the Hausdorff distance between $T(\gamma)$ and $T(a)$. Similarly, for any path π, let

$$d(\gamma, \pi) = \min_{\mu \in H(\gamma, \pi)} \max_{p \in T(\gamma)} ||p - \mu(p)||, \tag{5}$$

where $H(\gamma, \pi)$ is the set of all homeomorphisms between $T(\gamma)$ and $T(\pi)$.

For every contour-arc $a \in C \times C$, let $P_\epsilon(a) := P_\epsilon(a|I)$ be the probability that there exists a contour fragment in the scene, $\gamma \in \overline{\Gamma}$, such that $d(\gamma, a) < \epsilon$. Similarly, $P_\epsilon(\pi) := P_\epsilon(\pi|I)$ is the probability that there exists $\gamma \in \overline{\Gamma}$ such that $d(\gamma, \pi) < \epsilon$.

Lemma 2 *Let $\gamma \in \overline{\Gamma}$, and let π be a path such that $d(\gamma, \pi) < \epsilon$. If a is an arc of π, then there exists $\gamma' \in \overline{\Gamma}$ such that $d(\gamma', a) < \epsilon$. Similarly, if π' is a sub-path of π, then there exists $\gamma' \in \overline{\Gamma}$ such that $d(\gamma', \pi') < \epsilon$.*

Proof. Let $\mu \in H(\gamma, \pi)$ be the homeomorphism which acheives the distance $d(\gamma, \pi)$ and let γ' be the sub-contour of γ defined by $T(a) = \mu(T(\gamma'))$. Clearly, $d(\gamma', a) \leq d(\gamma, \pi) < \epsilon$. Similarly for the other part. □

A crucial component of the proposed approach to contour estimation consists in finding approximations to the probabilities $P_\epsilon(a)$ and $P_\epsilon(\pi)$. In principle, these probabilities depend on the whole image I but, presumably, one can construct an approximation $\hat{P}_\epsilon(\pi)$ to $P_\epsilon(\pi)$ which depends on a small set of localized "features". These features might include geometric relationships between the vertices of π; (recall that each such vertex is a contour primitive); brightness measurement in the neighborhood of $T(\pi)$; other contour paths having a vertex in common with π and their geometric relationship with π itself; etc. Clearly, the accuracy by which $\hat{P}_\epsilon(\pi)$ approximates $P_\epsilon(\pi)$ increases if more features are included in the model. Note that the problem of constructing approximations of $P_\epsilon(\pi)$ can be subdivided into two parts: selecting an appropriate set of features $\{f_1, \ldots, f_n\}$ and estimate the function $\hat{P}_\epsilon(f_1, \ldots, f_n)$ which "best" approximates $P_\epsilon(\pi)$. It is conceivable that the latter task might be carried out by means of training.

It should be noted that the contour descriptors computed at the various level of the hierarchy serve a dual purpose. In fact, besides the obvious role of approximators of scene contours, they also provide a means to parameterize the probabilities $P_\epsilon(\pi)$. Thus, descriptors with large spatial extent can effectively be used to incorporate global information efficiently. For instance, two co-linear contour fragments can effectively instantiate an hypothesis of an invisible contour fragment interpolating between them. The probability of this hypothesis can be expressed in terms of the geometric relationship between the two visible fragments.

We now describe a simple model to approximate $P_\epsilon(a)$, which is used in the current implementation of the algorithm. This model is similar to the one proposed in [9], except that in our model the sum of the probabilities of the arcs incident from a vertex is not normalized to one. The reason for this is that these probability estimates will be used for pruning rather than detection of maximum likelihood cycles. Therefore, as explained later, it will be assumed that our probability estimates $\hat{P}_\epsilon(a)$ are upper bounds to the true probabilities $P_\epsilon(a)$.

The probability estimate $\hat{P}_\epsilon(a)$ depends on the following features (see Fig. 6):

- The lengths l_1 and l_2 of the longest straight line segments which can be fitted to the polygonal lines $T(c_1)$ and $T(c_2)$ with a given upper bound on of the fitting error (see Fig. 6).
- The length of the straight-line interpolant $\sigma(c_1, c_2)$.
- The two orientation changes α_1 and α_2 induced by the interpolation.
- The difference $|b(c_1) - b(c_2)|$ in the estimated image brightness on the region side of the two polygonal descriptors c_1 and c_2.

The estimated probability $\hat{P}_\epsilon(a)$ is computed by considering three possible rules for composing contour primitives into a composite contour. These rules correspond to three possible reasons for which a contour is not represented by a single contour primitive (compare again with the model proposed in [9]).

- The contour was originally connected at the edgel-path stage and was split during the computation of the contour primitives. In this case the length of $\sigma(c_1, c_2)$ is zero. The probability estimate $\hat{P}_\epsilon(a)$ is set to a very high value (1 in the current implementation).
- The contour is split because of a sharp orientation change (corner). We expect potentially large values of α_1 and α_2 but small values of $|b(c_1) - b(c_2)|$ and of the length of $\sigma(c_1, c_2)$.
- The contour is split because of a loss of contrast. The length of $\sigma(c_1, c_2)$ can be large but the angles α_1 and α_2 should be small.

A probability estimate is computed for each of the hypotheses and $\hat{P}_\epsilon(a)$ is set to the largest of these values.

Every contour-arc whose probability estimate is less than a threshold $\delta_{\text{c.a.}}$ is pruned out. Let A be the set of remaining contour-arcs. The main property of the contour-graph (C, A) is described by the theorem below. It is assumed that the probability estimates $\hat{P}_\epsilon(a)$ are upper bounds to the true probabilities $P_\epsilon(a)$. Recall that a contour-graph is said to be an ϵ-covering of $\overline{\Gamma}$ if for every $\gamma \in \overline{\Gamma}$ there exists a path π in the contour-graph such that $d(\gamma, \pi) < \epsilon$. Let $1 - \delta_\epsilon(C)$ be the probability that $(C, C \times C)$ is an ϵ-covering of $\overline{\Gamma}$.

Theorem 2 *Let us assume that $\hat{P}_\epsilon(a) \geq P_\epsilon(a)$ for all $a \in C \times C$. Then, with probability at least*

$$1 - \delta_\epsilon(C) - |C|^2 \delta_{\text{c.a.}},$$

(C, A) is an ϵ-covering of $\overline{\Gamma}$.

Proof. Let $P(C, C \times C)$ denote the set of paths in $(C, C \times C)$. For any $\pi \in P(C, C \times C)$, let $B_\epsilon(\pi)$ be the set of $\gamma \in \mathbf{\Gamma}$ such that $d(\gamma, \pi) < \epsilon$. Let $\overline{\Gamma}_0 \subset \overline{\Gamma}$ be the set of contour fragments in the scene which are ϵ-covered by the graph $(C, C \times C)$:

$$\overline{\Gamma}_0 = \overline{\Gamma} \cap \bigcup_{\pi \in P(C, C \times C)} B_\epsilon(\pi).$$

First, let us prove that the probability that (C, A) is not an ϵ-covering of $\overline{\Gamma}_0$ is at most $|C|^2 \delta_{\text{c.a.}}$. The proof is similar to the proof of Lemma 1. If (C, A) is not an ϵ-covering of $\overline{\Gamma}_0$, then there exists $\gamma \in \overline{\Gamma}_0$ such that

$$\gamma \notin \bigcup_{\pi \in P(C, A)} \overline{\Gamma}_0 \cap B_\epsilon(\pi),$$

where $P(C, A) \subset P(C, C \times C)$ denotes the set of paths in (C, A). Therefore, since

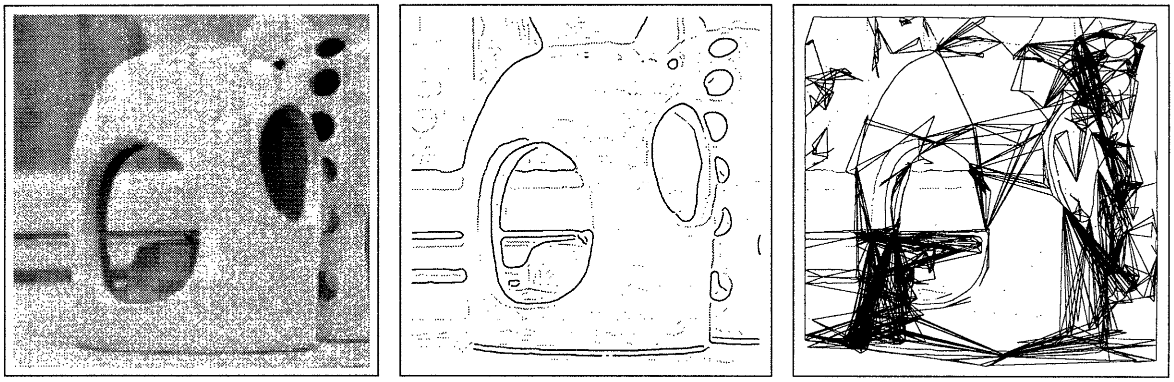
$$\gamma \in \overline{\Gamma}_0 = \bigcup_{\pi \in P(C, C \times C)} \overline{\Gamma}_0 \cap B_\epsilon(\pi),$$

if (C, A) is not an ϵ-covering of $\overline{\Gamma}_0$, then there exists a path $\pi \in P(C, C \times C) \setminus P(C, A)$ for which $\overline{\Gamma}_0 \cap \gamma \in B_\epsilon(\pi)$ and therefore $\overline{\Gamma}_0 \cap B_\epsilon(\pi) \neq \emptyset$. Since $\pi \notin P(C, A)$, there exists an arc a of π such that $a \notin A$. From $\overline{\Gamma}_0 \cap B_\epsilon(\pi) \neq \emptyset$ and Lemma 2, it follows that $\overline{\Gamma}_0 \cap B_\epsilon(a) \neq \emptyset$. Thus, the event "$P(C, A)$ is not an ϵ-covering of $\overline{\Gamma}_0$" is contained in the union over all $a \in C \times C$ of the events "$\overline{\Gamma}_0 \cap B_\epsilon(a) \neq \emptyset$ and $a \notin A$". Since $\hat{P}_\epsilon(a) \geq P_\epsilon(a)$, the conditional probability given I of each of these events is at most $\delta_{\text{c.a.}}$. Thus, by using the union bound, the probability given I that (C, A) is not an ϵ-covering of $\overline{\Gamma}_0$ is at most $|C|^2 \delta_{\text{c.a.}}$. The probability that $\overline{\Gamma}_0 \neq \overline{\Gamma}$ is at most $\delta_\epsilon(C)$ so that, by using the union bound again, the probability that (C, A) is not an ϵ-covering of $\overline{\Gamma}$ is at most $\delta_\epsilon(C) + |C|^2 \delta_{\text{c.a.}}$. □

Fig. 7 illustrates the covering properties of the contour-graph (C, A) guaranteed by Theorem 2. Fig. 7(c) shows the contour-graph (C, A) obtained by setting $\delta_{\text{c.a.}}$ equal to 0.005. The bottom part of the figure shows a cycle (selected by hand) in (C, A) which approximates the contour of the lamp in Fig. 7(a). All the contour-arcs incident from the vertices of this loop are also shown to illustrate the local ambiguity of this contour representation (i.e., notice the high degree of the vertices). Observe that large portions of this contour are invisible and that the local ambiguity of the contour representation is mostly caused by the presence of nearby parallel contours belonging to the two holes in the lamp. Notice also the presence of multiple nearby sub-paths along the lamp contour which illustrates the redundancy of the local contour-arc representation. Many more multiple responses to the lamp contour are present in the whole graph (C, A) (Fig. 7(c)).

5 Computing the cycles of a contour-graph

The last level of the hierarchy approximates scene contours by means of cycles in the contour-graph (C, A). A cycle represents a complete contour of an object or of a surface patch. It should be noted that by adding one special vertex c_∞ to the graph and connecting it to every other vertex in both directions, it is possible to represent every path as a cycle. An arbitrary path $\langle c_0, \ldots, c_l \rangle$ in the original contour-graph A is represented in the extended contour-graph by the cycle $\langle c_\infty, c_0, \ldots, c_l, c_\infty \rangle$. Notice that



(a) Brightness data (b) Edgel-paths (c) Contour-graph

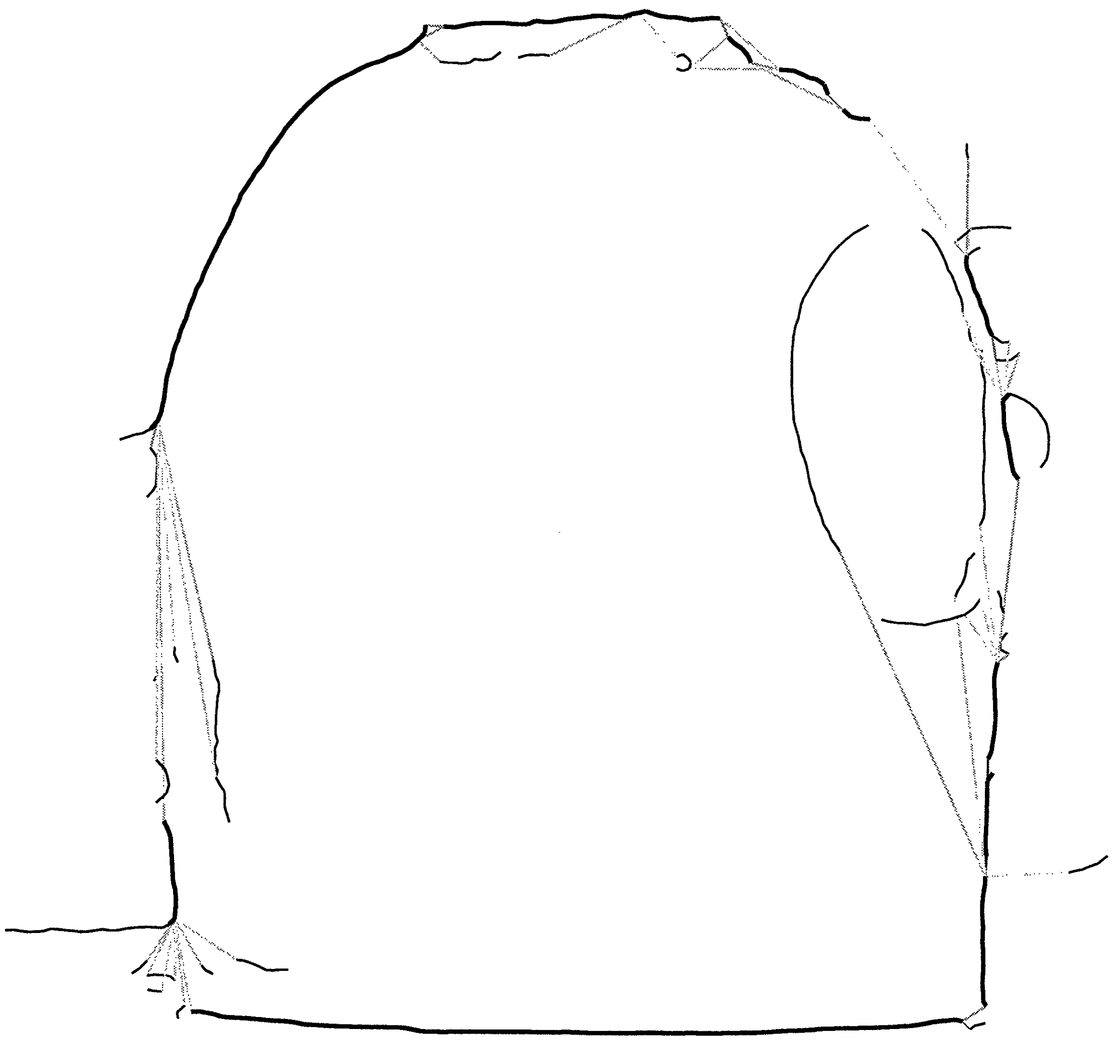


Figure 7: Contour-arcs near the lamp contour (see the text). The gray level of each contour-arc is proportional to the estimated probability $\hat{P}_\epsilon(a)$.

in the extended contour-graph a subtle difference exists between the paths $\langle c_0, \ldots, c_l \rangle$ and $\langle c_\infty, c_0, \ldots, c_l, c_\infty \rangle$. In fact, the former represents the assertion that the scene contains a contour fragment which is close to the polygonal line associated with the path, whereas the second one represents the assertion that the scene contains a *complete* contour close to the same polygonal line. A contour is complete if it is *maximal*, that is, if it is not a fragment of a larger contour. An open contour is complete if, for instance, it belongs to an object whose projection on the image plane is not wholly contained inside the image. The probabilities $P_\epsilon(\cdot)$ in the extended graph are defined accordingly. Thus, for instance, if $a = (c, c_\infty)$, then $P_\epsilon(a)$ is the probability that c is the last contour primitive of a maximal contour-path.

The proposed approach to computing the cycles in a graph is motivated by two reasons. Firstly, the difficulty encountered by local approaches to contour estimation suggests that, in order to construct good approximations of the probabilities $P_\epsilon(\pi)$, it is necessary to include global features of the path π in the parameterization of $P_\epsilon(\pi)$. In other words, it does not seem the case that the contour estimation problem can be modeled accurately by decomposing $P_\epsilon(\pi)$ into the product of local independent contributions, such as the probabilities $P_\epsilon(a)$ of the arcs of π. Accurate estimation of $P_\epsilon(\pi)$ is important to prune out a sufficiently large number of hypotheses and at the same time have high confidence that all scene contours are detected. If hypothesis evaluation has to be based on global features of the path, then the hypothesized path must be represented explicitly in their entirety, which rules out standard methods such as dynamic programming.

Secondly, our experimental work indicates that contour-graphs computed by using the method explained in Section 4 exhibit the same multiple-response phenomenon which occurs in the edgel-graph representation. That is, there exist bifurcations in the graph which do not lead to substantially different contour-paths. For long paths, this causes a combinatorial explosion of the number of similar paths which approximate the same scene contour. Thus, the cycle detection algorithm must allow for the compression of similar equivalent paths in order to avoid this unnecessary complexity. Again, this calls for an explicit representation of each path which is being hypothesized.

5.1 Regular paths

Let (C, A) be a contour-graph. A path $\pi = \langle c_0, \ldots, c_l \rangle$ in this graph is said be a *walk* if $c_0 \neq c_l$ and a *loop* if $c_0 = c_l$. A walk is said to be *simple* if all its vertices are distinct; a loop $\langle c_0, \ldots, c_{l-1}, c_0 \rangle$ is simple if the path $\langle c_0, \ldots, c_{l-1} \rangle$ is a simple walk. The set of all walks and of all loops are denoted W and L respectively. The set of all paths is $P = W \cup L$. Simple walks and loops are denoted W_s and L_s and the set of simple paths is $P_s = W_s \cup L_s$.

Two loops $\pi = \langle c_0, \ldots, c_l \rangle$, $\pi' = \langle c'_0, \ldots, c'_l \rangle$, are equivalent, $\pi \equiv \pi'$, if they have the same set of arcs or, equivalently, if there exists an integer j such that $c_i = c'_{(i+j) mod\ l}$. A *cycle* is an equivalence class of loops. Thus, the set of all cycles is given by the quotient of L by the equivalence relation, $L/\equiv$.

Notice that all the loops in the same equivalence class represent the same scene contour. Thus, to avoid having the same contour being represented more than once, we need to select a representative loop in each equivalence class. To do this, let us assign an arbitrary linear order to the set of contour-primitives C and let $\rho : C \rightarrow \{1, \ldots, N\}$ be the bijective map which assigns to every $c \in C$ its order rank.

The end-points c_0 and c_l of a non-zero length path $\pi = \langle c_0, \ldots, c_l \rangle$ are called the *external* vertices of π. The other ones are called *internal* vertices.

Definition 1 *A path $\pi = \langle c_0, \ldots, c_l \rangle$ is said to be ρ-*regular *or simply* regular, *if it is simple and if the ranks of its external vertices are both larger than the ranks of all its internal vertices,*

$$\rho(c_0) > \rho(c_i), \quad \rho(c_l) > \rho(c_i), \qquad i = 1, \ldots, l-1.$$

Note that all paths of length one are regular. Let $\rho_{\text{fi}}(\pi)$ and $\rho_{\text{la}}(\pi)$ denote the rank of the first and the last vertices of a path $\pi = \langle c_0, \ldots, c_l \rangle$ and let $\rho_{\text{int}}(\pi)$ be the maximum rank of its internal vertices:

$$\rho_{\text{fi}}(\pi) = \rho(c_0), \tag{6}$$

$$\rho_{\text{la}}(\pi) = \rho(c_l), \tag{7}$$

$$\rho_{\text{int}}(\pi) = \begin{cases} \max\limits_{1\leq j\leq l-1} \rho(c_j) & \text{if } l \geq 2 \\ 0 & \text{if } l = 1. \end{cases} \tag{8}$$

The set of all regular paths, denoted Ω, is then

$$\Omega = \{\pi \in P_{\text{s}} \ : \ \rho_{\text{fi}}(\pi) > \rho_{\text{int}}(\pi), \rho_{\text{la}}(\pi) > \rho_{\text{int}}(\pi)\}.$$

Two paths π, π' such that the last vertex of one path coincides with the first vertex of the other can be concatenated (composed). The composition of π with π' is denoted $\pi \circ \pi'$. Thus, if $c_l = c'_0$ then we have:

$$\langle c_0, \dots, c_l\rangle \circ \langle c'_0, \dots, c'_{l'}\rangle = \langle c_0, \dots, c_l, c'_1, \dots, c'_{l'}\rangle.$$

A fundamental property of regular paths is that they can be uniquely decomposed into regular paths. In fact, let $\pi = \langle c_0, \dots, c_l\rangle$, $l > 1$ and let c_{i^*} be the internal vertex of π with highest rank: $\rho(c_{i^*}) = \rho_{\text{int}}(\pi)$. Then it is easy to prove that the unique pair of regular paths π_1, π_2 such that $\pi_1 \circ \pi_2 = \pi$ is $\pi_1 = \langle c_0, \dots, c_{i^*}\rangle$, $\pi_2 = \langle c_{i^*}, \dots, c_l\rangle$.

5.2 Enumeration of regular paths

The algorithm described below enumerates all regular paths uniquely. That is, the algorithm generates a list of regular paths in which each regular path occurs exactly once. As a byproduct, all regular loops are uniquely enumerated and therefore every cycle is also uniquely enumerated.

Let us introduce the following operators defined on the collection of all sets of paths and which project a set of paths S onto a subset of it.

$$\begin{aligned} \xi S &= S \cap P_{\text{s}}, \\ \sigma_n S &= \{\pi \in S \ : \ \rho_{\text{fi}}(\pi) = n < \rho_{\text{la}}(\pi)\}, \\ \tau_n S &= \{\pi \in S \ : \ \rho_{\text{la}}(\pi) = n < \rho_{\text{fi}}(\pi)\}, \\ \lambda S &= \{\pi \in S \ : \ \rho_{\text{fi}}(\pi) = \rho_{\text{la}}(\pi)\}. \end{aligned} \tag{9}$$

Notice the paths in $\sigma_n S$ start from the vertex with rank n and that the paths in $\tau_n S$ terminate at the vertex with rank n. Thus, for any set of paths S, the last vertex of any path in $\tau_n S$ is the same as the first vertex of any path in $\sigma_n S$ and therefore any path in $\tau_n S$ can be composed with any path in $\sigma_n S$. The resulting set of paths is denoted $\tau_n S \circ \sigma_n S$.

Consider the following iterative equations:

$$X[0] = A; \tag{10}$$

$$Z[k] = \xi\left(\tau_{k+1} X[k] \circ \sigma_{k+1} X[k]\right), \quad k \geq 0; \tag{11}$$

$$X[k+1] = X[k] \cup Z[k], \quad k \geq 0. \tag{12}$$

Theorem 3 *Let $X[k]$, $Z[k]$, $k \geq 0$ be given by* (10)-(12). *Then,*

$$X[k] = \{\pi \in \Omega \ : \ \rho_{\text{int}}(\pi) \leq k\}, \tag{13}$$

$$Z[k] = \{\pi \in \Omega \ : \ \rho_{\text{int}}(\pi) = k+1\}, \tag{14}$$

$$X[N-2] = \Omega. \tag{15}$$

Proof. Equations (13) and (14) can be proved by induction. For $k = 0$, (13) holds because of (10). Let (13) be true for some $k \geq 0$. Then (14) is obtained by substituting (13) in (11). Let now (13) and (14) be both true for $k = k_0 \geq 0$. Then, (13) at $k+1$ follows by using (12). Finally, (15) is obtained by substituting $k = N-2$ in (13). □

As a corollary it follows that the set of all regular loops is given by $\lambda X[N-2]$. Moreover, from (14) it follows that $Z[k] \cap Z[k'] = \emptyset$ if $k \neq k'$. Therefore, each regular path is composed only once by the algorithm.

5.3 Compression

Because of the redundancy in the contour-primitive and contour-arc representations, needed to ensure that at least one contour-path exists for every scene contour with high probability, there might be multiple paths near to the same scene contour. These paths are close to each other so that one can eliminate redundant paths by computing

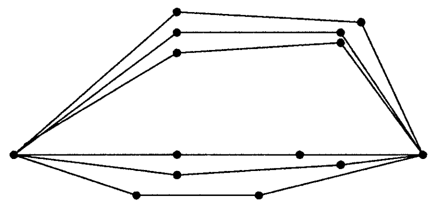


Figure 8: Six paths organized into two clusters.

the distance between them. To maximize computational efficiency, this *compression* operation should be carried out as early as possible during the enumeration of regular paths, rather than waiting until all paths have been enumerated. A way to do this is to consider the collections of paths between two vertices and then eliminate redundant paths in these collections at every stage k of the iterative algorithm given by (10)-(12). This guarantees that compression is carried out only within a class of equivalent paths.

Let $\beta_{n,n'}$ be the operator defined by:

$$\beta_{n,n'}S = \{\pi \in S \ : \ \rho_{\rm fi}(\pi) = n, \rho_{\rm la}(\pi) = n'\} \,. \tag{16}$$

For any two sets of paths S, T, let $S \cup=_\epsilon T$ denote the set of paths obtained by adding incrementally to S elements of T whose distance from every path in S is at least ϵ. Thus, $S \cup=_\epsilon T$ is an ϵ-separated ϵ-covering of $S \cup T$ if S is ϵ-separated (a set is ϵ-separated if the distance between any two elements is greater than ϵ). The modified algorithm which includes incremental compression is described by the following iterative equations:

$$X[0] = A; \tag{17}$$

$$Z[k] = \xi\left(\tau_{k+1}X[k] \circ \sigma_{k+1}X[k]\right), \quad k \geq 0; \tag{18}$$

$$X[k+1] = \bigcup_{n,n'} \left(\beta_{n,n'}X[k] \ \cup=_\epsilon \ \beta_{n,n'}Z[k]\right), \quad k \geq 0. \tag{19}$$

This algorithm produces the desired result only if the following condition is satisfied.

Definition 2 *The graph (C, A) is* ϵ-quantized *if for any three paths π_1, π_2, π_3 with the same initial vertex and final vertex, $d(\pi_1, \pi_2) \leq \epsilon$ and $d(\pi_2, \pi_3) \leq \epsilon$ imply $d(\pi_1, \pi_3) \leq \epsilon$.*

This condition requires that multiple responses to the same contour be organized into clusters. It is related to the ϵ-compressibility condition mentioned in Section 3 and defined in [4]. Fig. 8 shows an example of a set of paths between two vertices organized into clusters.

Theorem 4 *Let (C, A) be an ϵ-quantized graph; let $X[k]$, $Z[k]$, $k \geq 0$ be the solution of the non-compressed iterative system* (10)-(12)*; and let $X^c[k]$, $Z^c[k]$, $k \geq 0$ be the solution of the compressed iterative system* (17)-(19)*. Let us assume that in* (18) *we have*

$$\xi\left(\tau_{k+1}X[k] \circ \sigma_{k+1}X[k]\right) = \tau_{k+1}X[k] \circ \sigma_{k+1}X[k]. \tag{20}$$

Then $Z^c[k]$ is an ϵ-covering of $Z[k]$ and $X^c[k]$ is an ϵ-covering of $X[k]$.

Proof. Let us prove by induction on k that for every n, n', $\beta_{n,n'}Z^c[k]$ is an ϵ-covering of $\beta_{n,n'}Z[k]$ and $\beta_{n,n'}X^c[k]$ is an ϵ-covering of $\beta_{n,n'}X[k]$. For $k = 0$, we have that $\beta_{n,n'}X^c[0]$ is an ϵ-covering of $\beta_{n,n'}X[0]$ because $X^c[0] = X[0] = A$ and therefore $\beta_{n,n'}X^c[0] = \beta_{n,n'}X[0]$. Now, let us assume that $\beta_{n,n'}X^c[k]$ is an ϵ-covering of $\beta_{n,n'}X[k]$ and let us prove that $\beta_{n,n'}Z^c[k]$ is an ϵ-covering of $\beta_{n,n'}Z[k]$. From (11) and assumption (20) we have

$$\beta_{n,n'}Z[k] = \beta_{n,k+1}X[k] \circ \beta_{k+1,n'}X[k].$$

Thus, for every $\pi \in \beta_{n,n'}Z[k]$ there exist $\pi_1 \in \beta_{n,k+1}X[k]$ and $\pi_2 \in \beta_{k+1,n'}X[k]$ such that $\pi = \pi_1 \circ \pi_2$. Since $\beta_{n,k+1}X^c[k]$ and $\beta_{k+1,n'}X^c[k]$ are ϵ-coverings of $\beta_{n,k+1}X[k]$ and $\beta_{k+1,n'}X[k]$ respectively, there exist $\pi_1^c \in \beta_{n,k+1}X^c[k]$ and $\pi_2^c \in \beta_{k+1,n'}X^c[k]$ such that $d(\pi_1, \pi_1^c) \leq \epsilon$ and $d(\pi_2, \pi_2^c) \leq \epsilon$. Then, from the definition (5) of the distance function d, it follows that $d(\pi, \pi^c) = d(\pi_1 \circ \pi_2, \pi_1^c \circ \pi_2^c) \leq \epsilon$ and therefore $\beta_{n,n'}Z^c[k]$ is an ϵ-covering of $\beta_{n,n'}Z[k]$ because $\pi_1^c \circ \pi_2^c \in \beta_{n,n'}Z^c[k]$. It remains to be proved that $\beta_{n,n'}X^c[k+1]$ is an ϵ-covering of $\beta_{n,n'}X[k+1]$ if $\beta_{n,n'}X^c[k]$ ϵ-covers $\beta_{n,n'}X[k]$ and $\beta_{n,n'}Z^c[k]$ ϵ-covers $\beta_{n,n'}Z^c[k]$. From (12) and (19) we have

$$\beta_{n,n'}X[k+1] = \beta_{n,n'}X[k] \cup \beta_{n,n'}Z[k], \tag{21}$$

$$\beta_{n,n'}X^c[k+1] = \left(\beta_{n,n'}X^c[k] \cup\!\!=_\epsilon \beta_{n,n'}Z^c[k]\right). \tag{22}$$

From (21), since $\beta_{n,n'}X^{\mathrm{c}}[k] \cup \beta_{n,n'}Z^{\mathrm{c}}[k]$ ϵ-covers $\beta_{n,n'}X[k] \cup \beta_{n,n'}Z[k]$ by inductive assumption, it follows that for every $\pi \in \beta_{n,n'}X[k+1]$ there exists $\pi_1^{\mathrm{c}} \in \beta_{n,n'}X^{\mathrm{c}}[k] \cup \beta_{n,n'}Z^{\mathrm{c}}[k]$ such that $d(\pi, \pi_1^{\mathrm{c}}) \leq \epsilon$. From the definition of $\cup\!=_\epsilon$, we have that $\beta_{n,n'}X^{\mathrm{c}}[k] \cup\!=_\epsilon \beta_{n,n'}Z^{\mathrm{c}}[k] = \beta_{n,n'}X^{\mathrm{c}}[k+1]$ ϵ-covers $\beta_{n,n'}X^{\mathrm{c}}[k] \cup \beta_{n,n'}Z^{\mathrm{c}}[k] = \beta_{n,n'}X^{\mathrm{c}}[k+1]$ so that, there exists $\pi_2^{\mathrm{c}} \in \beta_{n,n'}X^{\mathrm{c}}[k+1]$ such that $d(\pi_1^{\mathrm{c}}, \pi_2^{\mathrm{c}}) \leq \epsilon$. Since the graph is ϵ-quantized, $d(\pi, \pi_2^{\mathrm{c}}) \leq \epsilon$. □

The assumption (20) in this theorem is needed because paths which are simple but are ϵ-close to a non-simple path are missed by the algorithm. This suggests the following definition of a strongly simple path.

Definition 3 *A simple path π in a contour-graph is* strongly simple *if there exists no polygonal line $U \subset T(\pi)$ and loop π' such that $d(U, T(\pi')) \leq \epsilon$.*

We conjecture that the algorithm (17)-(19) computes an approximation to all strongly simple paths. As a consequence, it follows that for every strongly simple loop π in (C, A) there exists a loop $\hat{\pi} \in \lambda X^{\mathrm{c}}[N-2]$ such that $d(\pi, \hat{\pi}) \leq \epsilon$.

5.4 Pruning paths with low probability

The last component of the algorithm consists in pruning paths for which the probability of representing a fragment of a scene contour is below a threshold. Recall that $\hat{P}_\epsilon(\pi)$ denotes an estimate of the conditional probability that the scene contains a contour fragment in the ϵ-neighborhood of $T(\pi)$. Let us assume that such an estimate is available and that it is an upper bound of the true conditional probability $P_\epsilon(\pi)$. For any $\epsilon, \delta > 0$, let $\vartheta_{\epsilon,\delta}$ denote the thresholding operator which filters out paths whose probability estimate $\hat{P}_\epsilon(\pi)$ is less than the threshold δ. That is, for any set of paths S,

$$\vartheta_{\epsilon,\delta} S = \left\{ \pi \in S \ : \ \hat{P}_\epsilon(\pi) \geq \delta \right\}.$$

The thresholding operator is applied at every iteration of the algorithm with a given probability threshold $\delta_{\mathrm{c.p.}}$ and with distance tolerance 2ϵ. Thus, the iterative equations are:

$$X[0] = A; \tag{23}$$

$$Z[k] = \vartheta_{2\epsilon,\delta_{\text{c.p.}}}\xi\left(\tau_{k+1}X[k] \circ \sigma_{k+1}X[k]\right), \quad k \geq 0; \tag{24}$$

$$X[k+1] = \bigcup_{n,n'} \left(\beta_{n,n'}X[k] \cup_{=_\epsilon} \beta_{n,n'}Z[k]\right), \quad k \geq 0. \tag{25}$$

Theorem 5 *Let (C, A) be an ϵ-quantized graph and let $X[k]$, $Z[k]$, $k \geq 0$ be given by (23)-(25). Let us assume that $\hat{P}_\epsilon(\pi) \geq P_\epsilon(\pi)$ for all paths π in $C \times C$. Let π be a strongly simple loop in (C, A) and let $|\pi|$ be its length. Then, with probability at least $1 - |\pi| \cdot \delta_{\text{c.p.}}$, if there exists a scene contour $\gamma \in \Gamma$ such that $d(\gamma, \pi) \leq \epsilon$ then there exists a loop $\hat{\pi} \in \lambda X[N-2]$ such that $d(\gamma, \hat{\pi}) \leq 2\epsilon$.*

Proof. It will be assumed that the conjecture postulated in 5.3 holds, namely, that the algorithm described by equations (17)-(19) computes an approximation to all strongly simple paths. Without loss of generality, let us assume that π is a regular strongly simple loop and let $c_0, \ldots, c_l$, $l = |\pi|$, be its vertices. Let $r_i^0 = \rho(c_i)$, $i = 0, \ldots, l$. For any $k \geq 0$, let $r_0^k, \ldots r_{l_k}^k$ be the sub-sequence of $r_0^0, \ldots, r_l^0$ obtained by removing entries less or equal than k. Notice that in the original algorithm (10)-(12), the set $X[k]$ contains a representation of π consisting of l_k sub-paths $c_1^k, \ldots, c_{l_k}^k$, where c_i^k, $1 \leq i \leq l_k$, is the sub-path of π containing the vertices between the vertex with rank r_{i-1}^k and the vertex with rank r_i^k. Let us sort the integers $r_0^0, \ldots, r_l^0$ and let $k_1 < \cdots < k_l$ be the resulting sorted sequence. Notice that for $k = k_i - 1$, equation (11) executes the composition $c_{i_k}^k \circ c_{i_k+1}^k = c_{i_k}^{k+1}$, where i_k is defined by $r_{i_k+1}^k = k_i$. Thus a total of l compositions are necessary to reconstruct the path π.

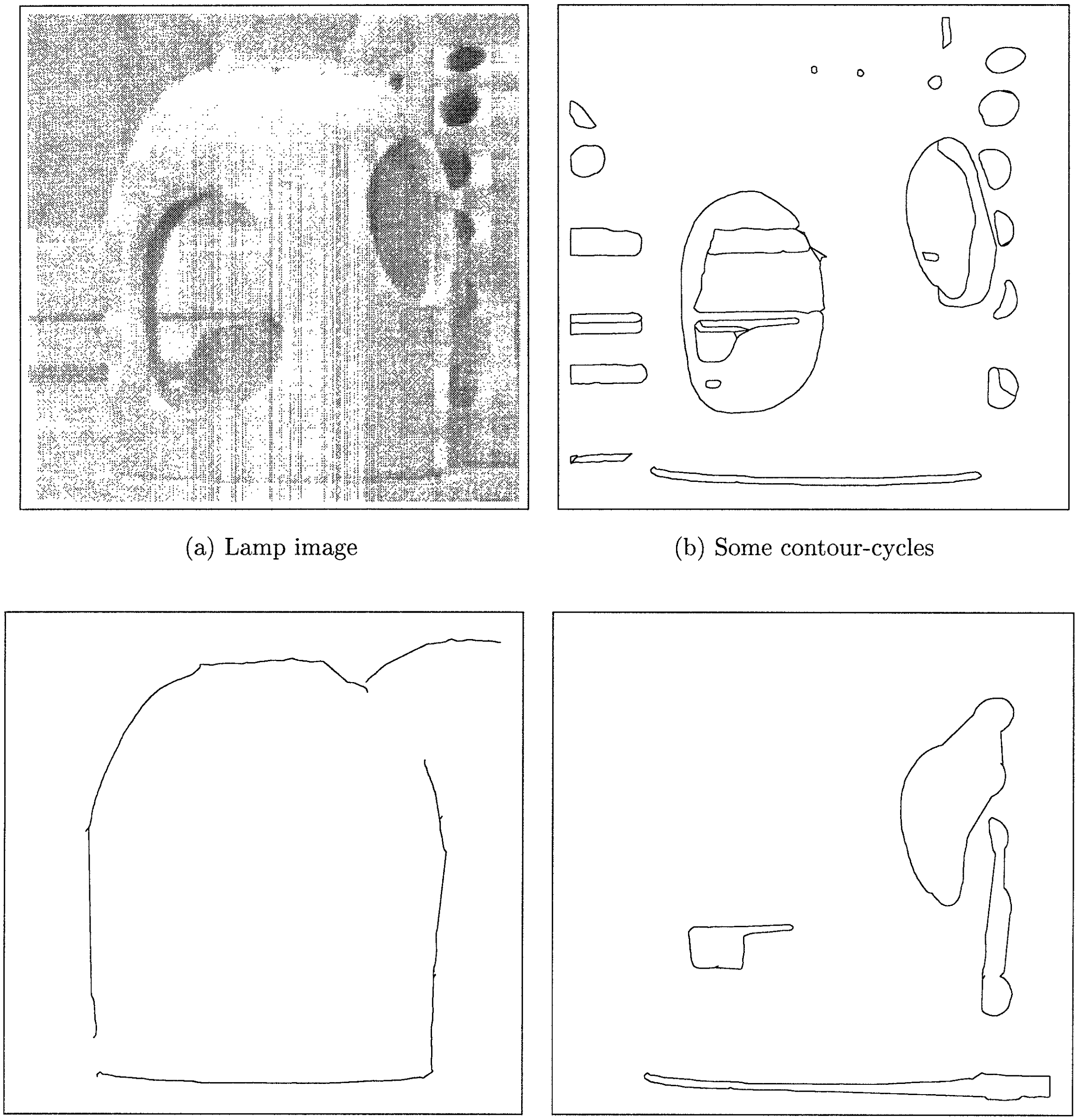
When compression is introduced (equations (17)-(19)) each sub-path c_i^k, $i = 1, \ldots, l_k$, is in general replaced by a sub-path $\hat{c}_i^k$ such that $d(c_i^k, \hat{c}_i^k) \leq \epsilon$. If there exists a scene contour $\gamma \in \Gamma$ such that $d(\gamma, \pi) \leq \epsilon$ then, from Lemma 2 for each $k \geq 0$ and $i = 1, \ldots, l_k$ there exists a scene contour fragment $\overline{\gamma}_i^k \in \overline{\Gamma}$ such that $d(\overline{\gamma}_i^k, c_i^k) \leq \epsilon$ and therefore, $d(\overline{\gamma}_i^k, \hat{c}_i^k) \leq 2\epsilon$. At the iterations $k = k_i - 1$, $i = 1, \ldots, l$ of equation (24), the composed path $\hat{c}_{i_k}^k \circ \hat{c}_{i_k+1}^k = \hat{c}_{i_k}^{k+1}$ is subjected to the pruning operator $\vartheta_{2\epsilon,\delta_{\text{c.p.}}}$. If the the composed path $\hat{c}_{i_k}^{k+1}$ is pruned out, then the probability that there exists a contour fragment $\overline{\gamma} \in \overline{\Gamma}$ such that $d(\overline{\gamma}, \hat{c}_{i_k}^{k+1}) \leq 2\epsilon$ is less than $\delta_{\text{c.p.}}$. Therefore, the probability that there exists a scene contour $\gamma \in \Gamma$ such that $d(\gamma, \pi) \leq \epsilon$ and that $\hat{c}_{i_k}^{k+1}$ is pruned out, is less than $\delta_{\text{c.p.}}$. Hence, by using the union bound, the probability that at least one of the $\hat{c}_{i_k}^{k+1}$, $k = 1, \ldots, l$ is prune out is at most $l\delta_{\text{c.p.}} = |\pi|\delta_{\text{c.p.}}$. Thus, with probability at least $1 - |\pi| \cdot \delta_{\text{c.p.}}$, none of the paths $\hat{c}_{i_k}^{k+1}$, $k = 1, \ldots, l$ is prune out so that the final set $X[N-2]$ contains a loop $\hat{\pi}$ such that $d(\gamma, \hat{\pi}) \leq 2\epsilon$. □

5.5 Experimental results

The proposed algorithm for cycle estimation has been tested on the lamp image shown in Fig. 9(a). The probability estimator $\hat{P}_\epsilon(\pi)$ used in these experiment is simply given by the product of the probability estimates of the arcs of π. Some of the computed cycles are shown in Fig. 9(b) and Fig. 10. Fig. 9(c) shows three open contours computed by the algorithm. The algorithm failed to compute explicitly a cycle corresponding to the whole contour of the lamp because the probability estimates $\hat{P}_\epsilon(\pi)$ of these cycles are smaller than the threshold $\delta_{\text{c.p.}}$. Clearly, one has to improve the probability estimator in order to recover the lamp contour in its entirety. To do this one needs to construct a more global model for $\hat{P}_\epsilon(\pi)$ which includes dependences on global features of the path.

6 Conclusions and future work

A probabilistic formulation of contour estimation via perceptual organization has been proposed where the goal is to compute efficiently a set of contour descriptors which approximates every scene contour with high probability. A hierarchy of contour descriptors has been proposed which allows to use information at several context scales to control the combinatorial explosion of the number of hypotheses. For the same purpose, the importance of compression of redundant equivalent hypotheses has been motivated to counteract the problem of multiple responses.



(a) Lamp image

(b) Some contour-cycles

(c) Three open contour-paths

(d) Some spurious contour-cycles

Figure 9: Contours computed by the current implementation of the algorithm

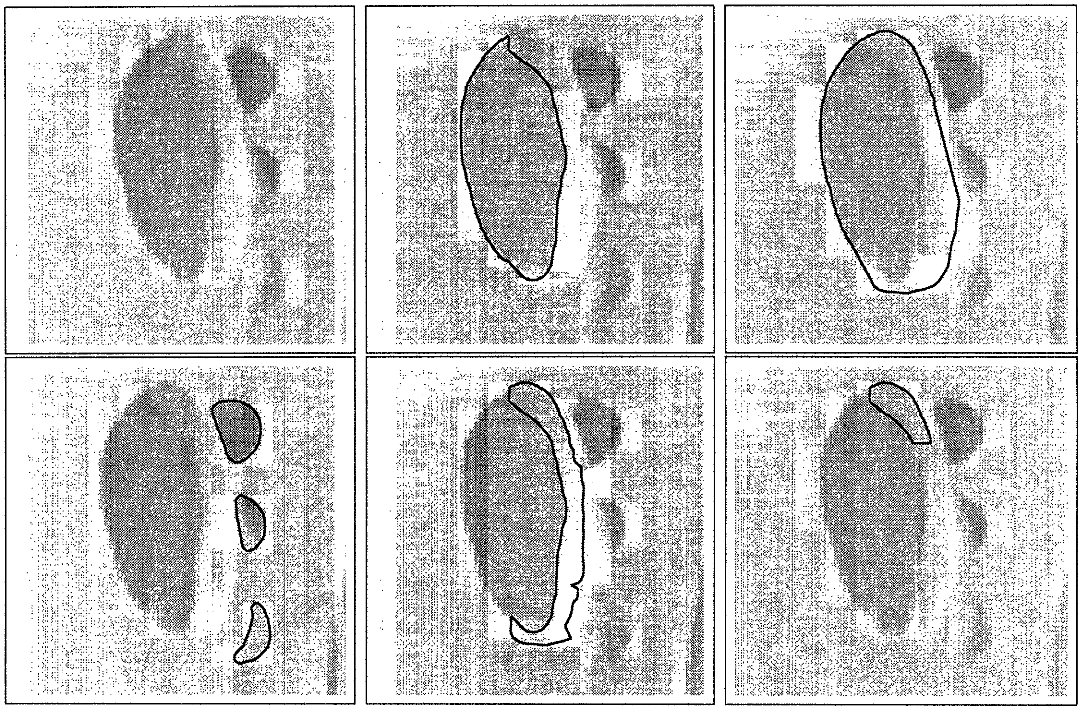


Figure 10: Some computed cycles

A specific algorithm to enumerate, prune and compress the cycles in a contour-graph has been proposed. This algorithm is able to incorporate global information during hypotheses evaluation and pruning. The current implementation of the algorithm, which does not yet take full advantage of global information, illustrates the intrinsic limitations of contour models based solely on local information.

The major issue which needs to be addressed is the development of better contour models to construct estimators of the probabilities $P_\epsilon(\pi)$, which are indeed the backbone of the whole approach. Global contour features such as convexity, symmetry, closure must be measured and included in the estimation of $P_\epsilon(\pi)$. Another strategy which we are exploring to refine these estimates is to detect groups of contour hypotheses which are likely to be mutually exclusive (e.g. contours passing through the same vertex).

References

[1] S. Casadei and S. K. Mitter. A hierarchical approach to high resolution edge contour reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1996.

[2] S. Casadei and S. K. Mitter. Hierarchical curve reconstruction. part 1: bifurcation analysis and recovery of smooth curves. In *European Conference on Computer Vision*, 1996.

[3] S. Casadei and S. K. Mitter. Hierarchical image segmentation – part i: Detection of regular curves in a vector graph. *International Journal of Computer Vision*, 27(3), March 1998.

[4] S. Casadei and S. K. Mitter. Multiscale estimation of regular image contours via a graph of local edgel hypotheses. Technical Report LIDS-P-2414, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, March 1998.

[5] R.L. Castano and S. Hutchinson. A probabilistic approach to perceptual grouping. *Computer Vision and Image Understanding*, 64(3):399–419, November 1996.

[6] T.J. Cham and R. Cipolla. Geometric saliency of curve correspondences and grouping of symmetric contours. In *European Conference on Computer Vision*, pages I:385–398, 1996.

[7] I.J. Cox, J.M. Rehg, and S. Hingorani. A bayesian multiple-hypothesis approach to edge grouping and contour segmentation. *International Journal of Computer Vision*, 11(1):5–24, August 1993.

[8] J. Dolan and E. Riseman. Computing curvilinear structure by token-based grouping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1992.

[9] J. H. Elder and S. W. Zucker. Computing contour closure. *European Conference on Computer Vision*, pages 399–411, 1996.

[10] D. Geiger and K. Kumaran. Visual organization of illusory surfaces. In *European Conference on Computer Vision*, pages I:413–424, 1996.

[11] D. Geman and B. Jedynak. An active testing model for tracking roads in satellite images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(1):1–14, January 1996.

[12] G. Guy and G.G. Medioni. Inferring global perceptual contours from local features. *International Journal of Computer Vision*, 20(1/2):113–133, 1996.

[13] F. Heitger and R. von der Heydt. A computational model of neural contour processing: Figure-ground segregation and illusory contours. In *International Conference on Computer Vision*, pages 32–40, 1993.

[14] L. Herault and R. Horaud. Figure-ground discrimination: A combinatorial approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):899–914, September 1993.

[15] D.W. Jacobs. Robust and efficient detection of salient convex groups. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(1):23–37, January 1996.

[16] G. Kanizsa. *Organization in vision : essays on gestalt perception.* Praeger, 1979.

[17] Thomas Leung and Jitendra Malik. Contour continuity in region based image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1998.

[18] D. Lowe. *Perceptual Organization and Visual Recognition.* Kluwer, 1985.

[19] D. Marr. *Vision.* W.H.Freeman & Co., 1982.

[20] R. Mohan and R. Nevatia. Perceptual organization for scene segmentation and description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14, June 1992.

[21] M. Nitzberg and D. Mumford. The 2.1-d sketch. In *International Conference on Computer Vision*, pages 138–144, 1990.

[22] M. Nitzberg, D. Mumford, and T. Shiota. *Filtering, segmentation, and depth*, volume 662 of *Lecture notes in computer science.* Springer-Verlag, 1993.

[23] P. Parent and S.W. Zucker. Trace inference, curvature consistency, and curve detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11, August 1989.

[24] S. Sarkar and K.L. Boyer. Perceptual organization using bayesian networks. In *IEEE Computer Vision and Pattern Recognition or CVPR.*, pages 251–256, 1992.

[25] S. Sarkar and K.L. Boyer. Perceptual organization in computer vision: A review and a proposal for a classificatory structure. *IEEE Trans. Systems, Man, and Cybernetics*, 23:382–399, 1993.

[26] S. Sarkar and K.L. Boyer. A computational structure for preattentive perceptual organization: Graphical enumeration and voting methods. *IEEE Trans. Systems, Man, and Cybernetics*, 24:246–267, 1994.

[27] A. Sha'ashua and S. Ullman. Structural saliency: the detection of globally salient structures using a locally connected network. In *International Conference on Computer Vision*, pages 321–327, 1988.

[28] J. Shi and J. Malik. Normalized cuts and image sementation. In *IEEE Computer Vision and Pattern Recognition*, pages 731–737, 1997.

[29] J.B. Subirana-Vilanova. Curved inertia frames and the skeleton sketch: Finding salient frames of reference. In *International Conference on Computer Vision*, pages 702–708, 1990.

[30] L.R. Williams and A.R. Hanson. Perceptual completion of occluded surfaces. In *IEEE Computer Vision and Pattern Recognition*, pages 104–112, 1994.

[31] L.R. Williams and A.R. Hanson. Perceptual completion of occluded surfaces. *Computer Vision and Image Understanding*, 64(1):1–20, July 1996.

[32] L.R. Williams and D. Jacobs. Stochastic completion fields: A neural model of illusory contour shape and salience. In *International Conference on Computer Vision*, pages 408–415, 1995.

[33] A.P. Witkin and J.M. Tenenbaum. On the role of structure in vision. In *Human and Machine Vision*, pages 481–543. Academic, 1983.