International Journal of Computer Vision 27(1), 71–100 (1998)

Hierarchical Image Segmentation—Part I: Detection of Regular Curves in a Vector Graph

STEFANO CASADEI AND SANJOY MITTER
Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139
casadei@lids.mit.edu

Received June 4, 1996; Accepted January 23, 1997

Abstract. The problem of edge detection is viewed as a hierarchy of detection problems where the geometric objects to be detected (e.g., edge points, curves, regions) have increasing complexity and spatial extent. An early stage of the proposed hierarchy consists in detecting the regular portions of the visible edges. The input to this stage is given by a graph whose vertices are tangent vectors representing local and uncertain information about the edges. A model relating the input vector graph to the curves to be detected is proposed. An algorithm with linear time complexity is described which solves the corresponding detection problem in a worst-case scenario. The stability of curve reconstruction in the presence of uncertain information and multiple responses to the same edge is analyzed and addressed explicitly by the proposed algorithm.

1. Introduction

1.1. The Role of Global Information

The problem of curve inference from a brightness image is of fundamental importance for image analysis. Curve detection and reconstruction is a difficult task since brightness data provides only uncertain and ambiguous information about curve location. A source of uncertainty is for instance the presence of "invisible curves", namely curves across which there is no brightness change (e.g., the sides of the white triangle in Fig. 1). Local information is not sufficient to resolve these uncertainties reliably and "global" information has to be used somehow. Zucker et al. (1988) have pointed out the need for a multistage approach which exploits both local and global computation. Methods based on optimization of a cost functional derived according to Bayesian, minimum description length, or energy-based principles (Geman and Geman, 1984; Marroquin et al., 1987; Mumford and Shah, 1989; Nitzberg and Mumford, 1990; Nitzberg et al., 1993; Zhu et al., 1995) introduce global information by adding an appropriate term to the cost functional. These formulations are simple and compact but may lead to computationally intractable problems. Moreover, it is often difficult or impossible to guarantee that the optimal solution of these cost functionals represents correctly all the desired features such as junctions and invisible curves (Richardson and Mitter, 1994). Active contour methods ("snakes") (Kass et al., 1988; Cohen and Kimmel, 1996; Shah, 1996) are able to use global information more efficiently but may require external initialization. More recent active contour approaches (Caselles et al., 1995; Kichenassamy et al., 1995) have somewhat overcome the initialization problem but detect only closed contours. Iterative procedures, such as relaxation labeling, can produce good results but only at a high computational cost (Parent and Zucker, 1989; Hancock and Kittler, 1990).

To exploit global constraints, interactions between data from distant locations are necessary. Therefore, one has to use large "contextual neighborhoods". This typically causes a combinatorial explosion of the search space since the number of possible configurations in each neighborhood grows exponentially with its size.

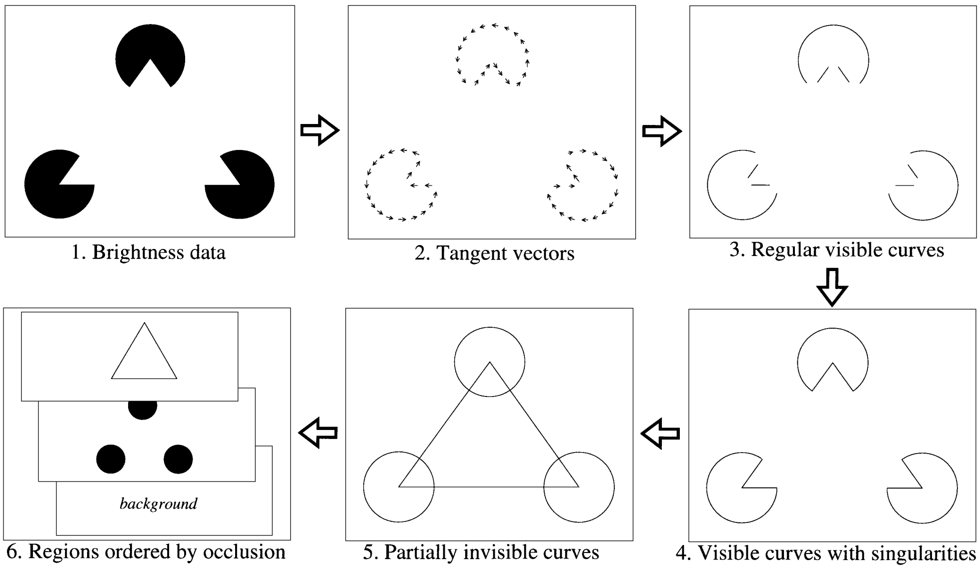


Figure 1. Hierarchy of edge representations for image segmentation. At the highest level the data is represented as a white triangle floating on top of three black disks (Kanizsa, 1979).

A strategy to avoid this combinatorial complexity is to *gradually* increase the maximum allowed interaction distance between descriptors by decomposing the whole detection process into several stages with increasing scales of interaction. At every stage, the process selects a small number of configurations which describe succinctly all possible data interpretations permitted by the maximum interaction distance of that stage. This leads to a hierarchy of representations where the spatial extent and complexity of descriptors increases by moving up in the hierarchy whereas the number of descriptors decreases. The scale increase and the difference in expressive power between any two adjacent levels of the hierarchy should be small enough so that computation is always local and efficient.

1.2. A Hierarchy of Representations for Image Segmentation

Figure 1 shows a hierarchy of contour representations whose highest level is a *2.1D sketch*, that is, a set of planar regions ordered by depth (Nitzberg and Mumford, 1990). At the bottom of the hierarchy we have the raw brightness data. At the second level edges are represented by tangent vectors whose magnitudes are proportional to the likelihood or strength of each edge hypothesis. Efficient methods exist to estimate these tangent vectors where the brightness discontinuity is large enough compared to the noise (Canny, 1986; Haralick, 1984; Perona and Malik, 1990). Notice that at this stage the number of edge hypotheses is roughly proportional to the size (area) of the image.

The next stages (3, 4 and 5) represent edges by means of curves. These curves can be obtained from sequences of tangent vectors by means of some linking or fitting procedure. Due to false negatives, false positives and other kinds of uncertainties in the tangent vector representation, a very large number of such sequences need to be considered as possible curve hypotheses. Clearly, the number of curve hypotheses depends exponentially on the number of bifurcations (locations with multiple tangents) and on the number of multiple responses to the same edge. Also, due to invisible edges, one should consider curve hypotheses connecting distant tangent vectors, which causes another combinatorial explosion of the number of possible hypotheses.

The basic assumption underlying the hierarchical approach is that the *order* in which uncertainties are resolved is the most important factor affecting computational costs. Thus, in estimating curves, it is important to determine what uncertainties can be resolved immediately and what should be instead deferred to a later stage.

This paper shows that uncertainties in the orientation and magnitude of tangent vectors and the problem of multiple responses can be tackled effectively at an early stage. Curve singularities such as corners and junctions are left for the next stage of curve estimation (level 4 in Fig. 1). This should be contrasted with methods which estimate corners and junctions directly from brightness (Deriche and Blaszka, 1993; Rosenthaler et al., 1992; Rohr, 1992; Perona, 1992). Other hierarchical algorithms (Parent and Zucker, 1989) estimate multiple tangents *before* curve reconstruction by using local curvature information to select a set of tangents at every point. Another approach to deal with multiple tangents is presented in (Iverson and Zucker, 1995) where curves with singularities are detected by using *logical/linear operators* obtained by composing nonlinearly elementary edge properties.

The reason why we believe that curve singularities should be estimated *after* the recovery of regular curves is that estimation of multiple junctions directly from brightness can be computationally expensive. In fact, the simplest edge models which yield efficient edge detection algorithms (constancy of brightness along the edge and sharp variations in the orthogonal direction) break down near curve singularities. On the other hand, more general edge models lead to nonlinear optimization problems which require computationally expensive iterative procedures or convolutions with many filters with large spatial support. (However, Nayar et al. (1996) propose a quite efficient approach for solving these nonlinear problems.) Thus, since regular curves can be estimated reliably and efficiently (as shown in this paper) they should be used to constrain the search for multiple tangents. A way for doing this is presented in (Casadei and Mitter, 1996a).

Invisible curves are dealt with at the last stage of curve estimation (level 5). Finally, two dimensional descriptors (regions) and occlusion information are introduced at the last stage of image segmentation. This paper is devoted to the computation of regular visible curves (level 3) from tangent vectors (level 2). For some results on the relationship between levels 1, 2 and 3 see (Casadei, 1995). Work on the remaining parts of the hierarchy is in progress. Some experimental results for level 4 are reported in (Casadei and Mitter, 1996a).

1.3. Compositional vs. "Wavelet" Hierarchies

The hierarchical approach adopted here is similar to multiscale schemes such as wavelet analysis in that representations at a large scale are constructed efficiently from local data by using the appropriate number of intermediate levels. The main difference is that the dictionary of primitive elements used in wavelet-like multiscale approaches does not change across the levels, except for a dilation transformation. As a result, information is lost or simplified and representations become coarser at larger scales. On the contrary, we are interested in hierarchies where the high levels contain *more* information than the lower levels. Thus the expressive power of the underlying dictionary has to increase when moving up in the hierarchy.

Whereas the basic transformation underlying wavelet and similar multiscale representations is a dilation applied to the domain of the raw data, the hierarchies considered here are based on a *compositional* transformation (Bienenstock and Geman, 1995). That is, the models to be detected are decomposed recursively into simpler units, leading to a *hierarchy of models*. Computation is mostly a bottom-up process which detects and reconstructs these models by means of composition. Top-down feedback, whose laws are derived from the model hierarchy, is used to select groupings of primitive units which are consistent with high level models. Pruning the exponentially large search space of all possible groupings is necessary to keep computation efficient and can be done by using top-down feedback. A *hierarchy of descriptions* obtained by recursive composition of descriptors is generated from the input data as a result of this process.

To achieve robustness and computational efficiency, one needs to design a "smooth" hierarchy, that is a hierarchy where consecutive levels are sufficiently "close" to each other so that every level of the hierarchy contains all the information necessary to reconstruct efficiently the objects at the following level. Thus, to design the next level (in a bottom-up design approach), one has to understand what composite objects can be formed efficiently and robustly from the parts present in the current level (for instance, regular visible curves can be formed efficiently and robustly from tangent vectors according to the model of Section 2.3). This "smoothness" constraint limits the amount of expressive power which can be added from one level to the next and therefore determines how many levels in the hierarchy are needed to bridge the gap between the input data and the desired final representation. For this reason (as argued in Section 1.2), we believe that in edge detection curve singularities as well as invisible curves should be left out of the lowest level dictionary and included only at a higher level.

By using a hierarchical approach, it is possible to resolve uncertainties and ambiguities when the necessary contextual information is available. Typically, it is impossible to eliminate efficiently all uncertainties in a single step. Then, one should eliminate part of the uncertainties first and then use the new, more informative representation to resolve more uncertainties. Those uncertainties which can not be resolved should be propagated to the higher levels, rather than being resolved arbitrarily. Therefore, in general, intermediate representations are quite uncertain and ambiguous and may contain mutually inconsistent hypotheses.

1.4. Perceptual Organization as a Detection Problem: Hierarchical Coverings and Completeness

Many authors have suggested that hierarchical algorithms can be used to infer global structures efficiently (Dolan and Riseman, 1992). Computation of global descriptors from simpler ones is related to the process of *perceptual organization* (Sarkar and Boyer, 1993; Mohan and Nevatia, 1992; Lowe, 1985). Perceptual organization, which can be repeated recursively and hierarchically, consists in grouping descriptors according to properties such as proximity, continuity, similarity, closure and symmetry. The role of these properties in human visual perception is well documented in (Kanizsa, 1979). Every grouping of descriptors is then composed into a higher level and more global descriptor. To assess the significance of groupings in a task independent fashion, principles such as non-accidentalness (Lowe, 1985) and minimum description length (Bienenstock and Geman, 1995) have been proposed. These principles provide a general framework to design all the grouping procedures in the hierarchy. However, they do not guarantee per se that particular classes of objects are detected and reconstructed correctly by these procedures. The ultimate goal of perceptual organization is to detect and represent explicitly all the relevant structures present in the data. Thus, perceptual organization can be viewed as a *detection* problem where the dictionary of objects to be detected consists of all the high level descriptors which represent compactly the groupings of low level descriptors. A detection algorithm is successful if it makes explicit all the object in the dictionary which are consistent with the set of low level descriptors in the input representation. For instance, the problem of grouping edge-points or tangent vectors into curvilinear structures can be cast as a curve detection problem by defining a class of curves to be detected and their relationship with their constituent tangent vectors. A model for this relationship is proposed in Section 2.3. Equivalently, a detection problem can be formulated as a *covering* problem where the goal is to construct a small subset of the high level dictionary which approximates every possible high level object consistent with the low level data according to the given model (compare with Theorem 5).

In a hierarchical approach, perceptual organization is really a *hierarchy* of detection problems. Each problem consists in computing explicitly all the objects in the dictionary of that level which are consistent with the data at the previous levels. Detection can be achieved by composition: object parts are detected first and then composed to reconstruct the object. This aggregation method is repeated at every level up to the top level which contains the desired global description.

To guarantee robust performance, the representation computed at each level must be *complete* with respect to the dictionary of that level. In other words, the computed representation must be a covering of the set of objects in the dictionary which are consistent with the data. That is, each consistent object must be approximated by at least one object in the computed representation. For instance, the set of regular curves computed by the proposed algorithm contains an approximation to every regular curve which is consistent with the input set of tangent vectors. This completeness property guarantees that all possible hypotheses will be explored and that uncertainties are propagated upwards in the hierarchy instead of being resolved arbitrarily.

1.5. Outline

The problem of detecting regular visible curves from a set of tangent vectors is considered. It is assumed that these tangent vectors represent all the possible hypotheses about edges which can be derived by local estimates of the brightness variations. Since this paper deals only with visible contours with sufficiently high brightness change, we can assume that only nearby tangent vectors can be consecutive points of a path. Thus, all curve hypotheses can be represented as paths in a graph with local connectivity. This graph is called the *vector graph* of the image. In Section 2 a model is defined which relates the curves to be detected to the vector graph. The problem of approximating all these curves efficiently is non trivial because the set of all possible curve hypotheses is exponentially large due to multiple responses and uncertainties in the magnitude and orientation of the tangent vectors.

In Section 3 the concept of stability in edge linking and the notion of stable graphs are introduced. It will be argued that some of the errors incurred by conventional linking algorithms are due to the instability of the underlying search graph.

The algorithm proposed to detect regular curves is composed of two parts (an earlier version of this algorithm is described in (Casadei and Mitter, 1996b). First, a stable graph is computed from the initial vector graph (Sections 4 and 5). It is proved that this stable graph preserves the relevant information about the regular curves to be detected. At the same time, the uncertainties which may cause instability are removed so that an approximation of every curve can be computed efficiently. The price for eliminating instabilities is that information about multiple tangents at the same point is lost. This information can be recovered more reliably at a later stage (level 4 of Fig. 1. See also (Casadei and Mitter, 1996a)). The second part of the algorithm resolves the remaining ambiguities by selecting the paths with minimum turn (Section 6). Section 7 discusses how to deal with cycles in the graph. The results of experiments are presented in Section 8. Section 10 contains the proofs of the theorems.

1.6. Notation List

- $P \subset \mathbb{R}^2$: set of given candidate edge-points
- $V = \{v_p : p \in P\}$: set of given tangent vectors representing edge-point hypotheses
- $\phi(p) = |v_p|$: strength of edge hypothesis v_p
- $\theta(p) \in [0, 2\pi]$ orientation of v_p (estimate of edge orientation)
- $A \subset P \times P$: arcs of the vector graph
- $l^{\max}(A)$: maximum arc length in A
- π: path in vector graph (P, V, A)
- $\text{arcs}(\pi)$: the arcs in the path π
- $\sigma(p_1, p_2)$: straight line segm. between p_1 and p_2
- $\sigma(a)$: str. line segm. between end-points of $a \in A$
- $\sigma(\pi)$: polygonal line associated with path π
- $\mathbf{B}_w(p)$: ball centered at p with radius w
- $N_w(\pi)$ neighborhood of $\sigma(\pi)$ with radius w
- $d(S_1; S_2)$ asymmetric Hausdorff distance of S_1 from S_2 (Eq. (1))
- $\Gamma = \{\gamma\}$: set of curves to be detected
- κ: maximum curvature of a curve in Γ
- Θ_1 maximum error on estimated orientation
- δ_0: maximum deviation of approximating path in (P, V, A) from curve γ to be detected
- δ_1: distance from γ at which $\phi(p)$ decays
- δ_2: max. dist. at which γ affects vector field
- D^0_γ, D^1_γ, D^2_γ: neighborhoods of γ with radii δ_0, δ_1, δ_2 (Fig. 5)
- $\sigma^{\perp}_w(p)$: segment centered at p with length $2w$ and perpendicular to tangent vector v_p
- $u_\perp(p)$: unit vector perp. to tangent vector v_p
- $U_w(\pi)$: attraction basin of π (Fig. 9(a))
- $\beta_w(\pi)$: boundary of $U_w(\pi)$
- $\beta^-_w(\pi)$, $\beta^+_w(\pi)$: lateral components of $\beta_w(\pi)$
- $\beta^-_w(a)$, $\beta^+_w(a)$: lateral segments of arc a (Fig. 9(b))
- $A - P'$: subgraph of A obtained by removing the vertices in P' from A
- $E(a_1, a_2)$: The four end-points of arcs a_1, a_2
- $\tilde{P}_w(V, A)$: vertices suppressed by stabilization procedure (Eq. (11))
- $S^w(V, A)$: arcs obtained by stabilization proc.
- $\bar{A}$: graph obtained by sigma-connecting A (Fig. 13)

2. Vector Graphs

This paper addresses the problem of detecting and reconstructing a set of regular curves Γ from local and noisy information about these curves. This information is represented by a *vector graph*, namely a triple (P, V, A) where

- $P \subset \mathbb{R}^2$ is a finite set of candidate curve points;
- $V = \{v_p : p \in P\}$ is a discrete vector field with vertices P;
- $A \subset P \times P$ represents a set of directed arcs.

A directed arc between p_1 and p_2 is represented by a pair $(p_1, p_2) \in P \times P$. The pair (P, A) is a *directed graph* with arcs A and vertices P. Figure 2 shows an example of a vector graph and introduces some notation. The orientations $\theta(p)$ of the tangent vectors $v_p \in V$ represent estimates of the local orientation of

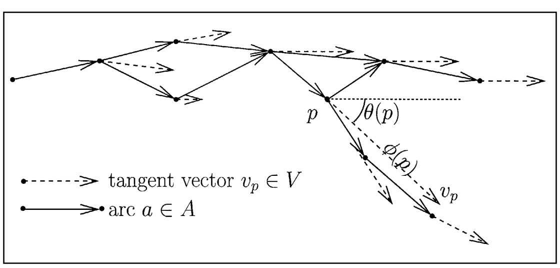


Figure 2. Example of a vector graph. Solid segments represent arcs of the directed graph and dashed segment represent tangent vectors.

the curves in Γ and the magnitude $\phi(p)$ of these vectors measures the likelihood that each candidate point $p \in P$ belongs to some curve. The vector graph (P, V, A) can be viewed as a noisy local projection of the curves in Γ onto small neighborhoods of the image. A model of the relationship between Γ and the vector graph (P, V, A) is proposed in Section 2.3.

2.1. Computation of the Vector Graph

To guarantee that the proposed algorithm generates a meaningful set of curves, the procedure which computes the vector graph from the brightness image must satisfy the three assumptions described in Section 2.3. These assumptions can be reformulated in terms of a noisy brightness model of the edge, as explained in (Casadei, 1995). By means of this model, it is possible to write all the parameters in the three assumptions of Section 2.3 as a function of the contrast and scale of the edge and the noise amplitude in the image.

In our implementation, a fitting method similar to (Haralick, 1984) has been used to compute the vector graph. This method assumes that brightness changes significantly across the curves to be detected. Furthermore, it is assumed that the scale at which this change occurs is known. To compute a set of tangent vectors, the image is tiled with a set of overlapping regions. One tangent vector v_p is computed from each region R by means of the following steps (see Fig. 3):

- Estimate the gradient of the brightness data in R by fitting a linear polynomial. Let $\theta(R)$ be the direction orthogonal to the estimated gradient.
- Fit a third order polynomial constant along $\theta(R)$ to the brightness data in R.
- By using the fitted third order polynomial, locate to sub-pixel accuracy the point where the estimated brightness gradient is locally maximum in the direction of the gradient. Let p be this point.
- Let $\theta(p) = \theta(R)$. This is an estimate of the orientation of the curve passing through p.
- Let $\phi(p)$ be the gradient magnitude at p. In general, $\phi(p)$ is some positive quantity depending on the gradient magnitude (and maybe also on the fitting error) which represents some sort of feedback from the brightness image. This feedback evaluates the likelihood that there exists indeed a curve with orientation $\theta(p)$ passing through p.
- Let v_p be the tangent vector with foot p, orientation $\theta(p)$, and magnitude $\phi(p)$.

Let P be the set of all the estimated points p and let V be the set of all the tangent vectors v_p, $p \in P$. V is a *discrete vector field*. The set of arcs A is then given by the set of all pairs $(p_1, p_2) \in P \times P$ estimated from adjacent regions and aligned with v_{p_1} (namely such that $(p_2 - p_1) \cdot v_{p_1} \geq 0$). A *path* in this graph is a sequence $\pi = (p_1, \ldots, p_n)$ such that $(p_i, p_{i+1}) \in A$, $i = 1, \ldots, n-1$. The corresponding sequence of vectors $(v_{p_1}, \ldots, v_{p_n})$ will also be called a path.

2.2. Relationship Between the Vector Graph (P, V, A) and the Curves Γ

Γ denotes the set of regular curves to be detected from the vector graph (P, V, A). What assumptions are appropriate to model the relationship between Γ and (P, V, A)? In the ideal case, when no noise is present, the following assumptions are quite natural:

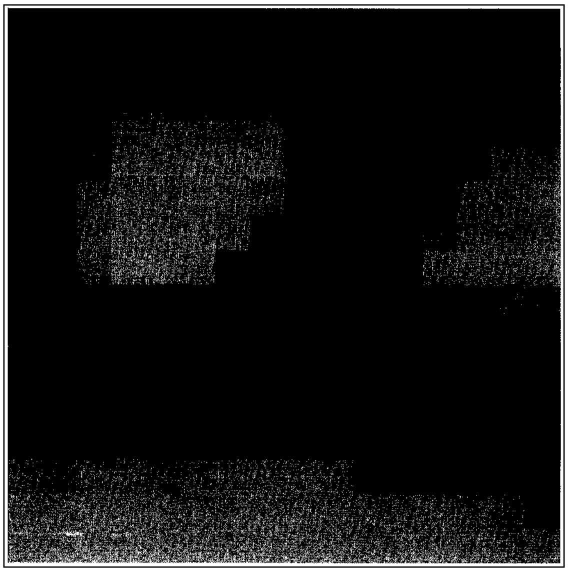
(C1) The vector graph contains a connected sampling of the tangent bundle of each curve $\gamma \in \Gamma$.
(C2) The vector field V is locally maximum on every curve $\gamma \in \Gamma$.

Recall that the tangent bundle of a curve is the set of all its tangents. Thus, the first condition requires that for every curve $\gamma \in \Gamma$, the vector graph (P, V, A) contains a path whose vertices are tangents to γ. This is called the approximating path of γ.

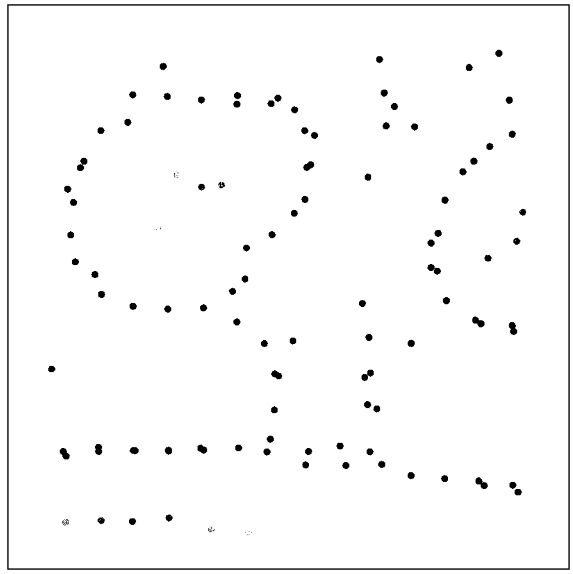
The second condition (C2) is necessary because the graph may contain tangent vectors other than those belonging to curve-approximating paths. Ideally, the magnitude of these spurious vectors is always smaller than nearby vectors belonging to a true curve.

When noise is present, the following distortions may be present:

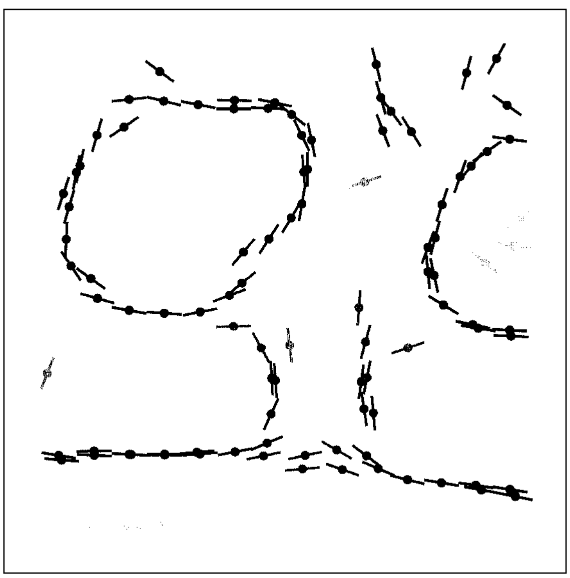
- The vertices of the approximating paths are not exactly on the curve $\gamma \in \Gamma$. Let δ_0 be an upper bound on the distance of these vertices from γ.
- The tangent vectors v_p of the approximating path are not perfectly aligned with the curve tangents of the approximated curve γ. Let Θ_1 be an upper bound on this deviation.
- The magnitude of the vector field V may achieve the local maximum at some distance away from γ. Let δ_1 be an upper bound on this distance.



(a) Brightness



(b) Points P



(c) Vectors V

(d) Arcs A

Figure 3. The vector graph (P, V, A) computed from the brightness image in (a) by using the method described in Section 2.1. (b) The estimated curve points P. (c) The tangent vectors V. The magnitude of the tangent vectors is coded by the gray level of the segments. (d) The set of arcs A (the direction of the arcs is not shown).

2.2. *Formal Assumptions*

We now proceed to state the three assumptions which define the curve model in terms of the vector graph (P, V, A). These assumptions define rigorously the distortion parameters δ_0, δ_1 and Θ_1. These three parameters can be related to the parameters of a noisy brightness model of the edge (sampling rate, contrast/noise ratio and scale) as explained in (Casadei, 1995).

For simplicity, the term "curve," which usually means a continuous mapping from an interval to $\mathbb{R}^2$, will be used to denote the image of the curve, which is a subset of $\mathbb{R}^2$. Thus, if γ is a curve, then $\gamma \subset \mathbb{R}^2$.

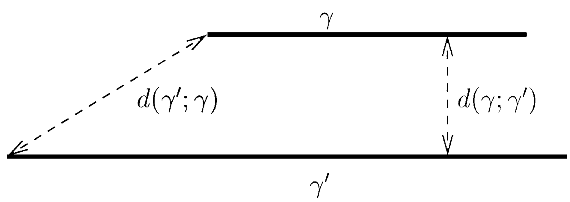


Figure 4. The asymmetric Hausdorff distance of γ from γ', denoted $d(\gamma; \gamma')$, is the maximum distance of a point in γ from the set γ'. Notice that $d(\gamma; \gamma') \neq d(\gamma'; \gamma)$.

The polygonal curve defined by the path π is denoted $\sigma(\pi)$. It is given by the union of all the straight line segments $\sigma(a)$ on the path. Here $\sigma(a)$, $a = (p_1, p_2)$, denotes the set of points lying on the straight line segment with end-points p_1 and p_2.

Let S_1, S_2 be subsets of $\mathbb{R}^2$. The *asymmetric Hausdorff distance* of S_1 from S_2 is defined as

$$d(S_1; S_2) = \max_{p_1 \in S_1} d(p_1; S_2) = \max_{p_1 \in S_1} \min_{p_2 \in S_2} \|p_1 - p_2\| \tag{1}$$

where $d(p_1; S_2)$ is the distance of the point p_1 from the set S_2. See Fig. 4.

The first assumption requires that every curve $\gamma \in \Gamma$ has an approximating path in (P, A) with error bounded by some constant δ_0.

Covering Condition. *The graph (P, A) covers Γ with distance δ_0. That is, for every $\gamma \in \Gamma$ there exists a path π in (P, A) such that $d(\gamma; \sigma(\pi)) < \delta_0$.*

Notice that since the asymmetric Hausdorff distance is used, the approximating path may be longer than the path itself. Had we used the symmetric Hausdorff distance instead, we would have assumed that the graph (P, A) contains information about curve end-points, which is too strong an assumption.

The other two conditions involve only the vector field V and the curves Γ. For simplicity, these constraints are formulated only for unbounded curves with zero curvature (namely infinite straight lines).

Let γ be a curve in Γ. The decay condition requires that the vector field V decays at a distance δ_1 from γ. More precisely (see Fig. 5),

Decay Condition.

$$p_1 \in D^0_\gamma;\ p_2 \in D^2_\gamma \setminus D^1_\gamma \Rightarrow \phi(p_1) > \phi(p_2) \tag{2}$$

where D^i_γ, $i = 0, 1, 2$, are the neighborhoods of γ with radius δ_i, $0 < \delta_0 \leq \delta_1 < \delta_2$:

$$D^i_\gamma = \{p \in \mathbb{R}^2 : d(p; \gamma) < \delta_i\} \quad i = 0, 1, 2\,;$$

and δ_0 is the parameter used in the covering condition. The parameter δ_2 is the distance up to which γ constrains the vector field V. Notice that the magnitude

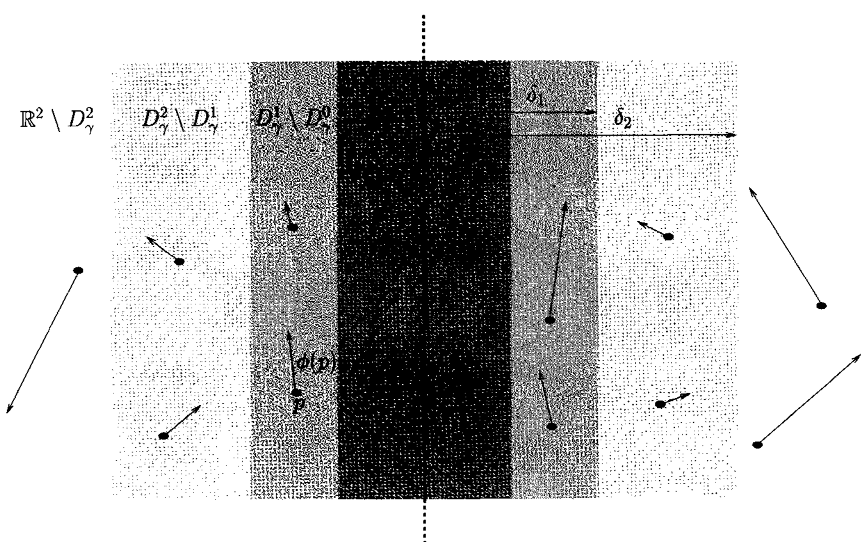


Figure 5. Constraints on the vector field in the vicinity of a curve. The magnitude $\phi(p)$ of the tangent vectors (length of arrows) is larger in D^0_γ than in $D^2_\gamma \setminus D^1_\gamma$. The magnitude $\phi(p)$ is arbitrary in $D^1_\gamma \setminus D^0_\gamma$. The angle formed by a tangent vector in D^1_γ with respect to the orientation of the curve is less than Θ_1.

of V in $D^1_\gamma \backslash D^0_\gamma$ is arbitrary. This is to model the fact that, due to noise, the local maxima of $\phi(p)$ may be displaced from the inner neighborhood D^0_γ.

Finally, the alignment condition requires that the orientation of the vector field V at a distance from γ less than δ_1 does not deviate by more that Θ_1 from the orientation of γ. That is, if θ_γ denotes the orientation of γ,

Alignment Condition.

$$p \in D^1_\gamma \Rightarrow \|\theta(p) - \theta_\gamma\| < \Theta_1 \tag{3}$$

Definition 1. Let Γ be a set of curve. The vector graph (P, V, A) is said to be a *projection* of Γ with admissible deviations δ_0, δ_1, δ_2 and Θ_1 if it satisfies the covering, decay and alignment conditions on Γ.

3. Stability

The goal of the algorithm is to compute a set of disjoint curves $\hat{\Gamma}$ which approximate every curve in Γ. Attaining robust performance in the presence of the uncertainties implicit in the model described in Section 2.3 is potentially an intractable problem. In fact, interference due to nearby curves and uncertainty in curve location and orientation generate ambiguities as to how candidate points should be linked together to form a curve. These ambiguities result in bifurcations in the vector graph. The number of possible paths can be exponentially large and it might be impossible to explore efficiently all of them. On the other hand, to obtain a complete representation every plausible path must be somehow taken into account.

3.1. *Inadequacy of Greedy Linking Methods*

Figure 6 illustrates the inadequacy of straightforward linking methods in the presence of uncertainties. Let's assume that there is just one curve γ to be detected, namely $\Gamma = \{\gamma\}$, and that γ satisfies the decay condition. Also, let's assume that $\delta_2 = \infty$. Thus the vector field in the inner neighborhood D^0_γ (dark shaded area) is larger than the vector field in the outer region $\mathbb{R}^2 \backslash D^1_\gamma$

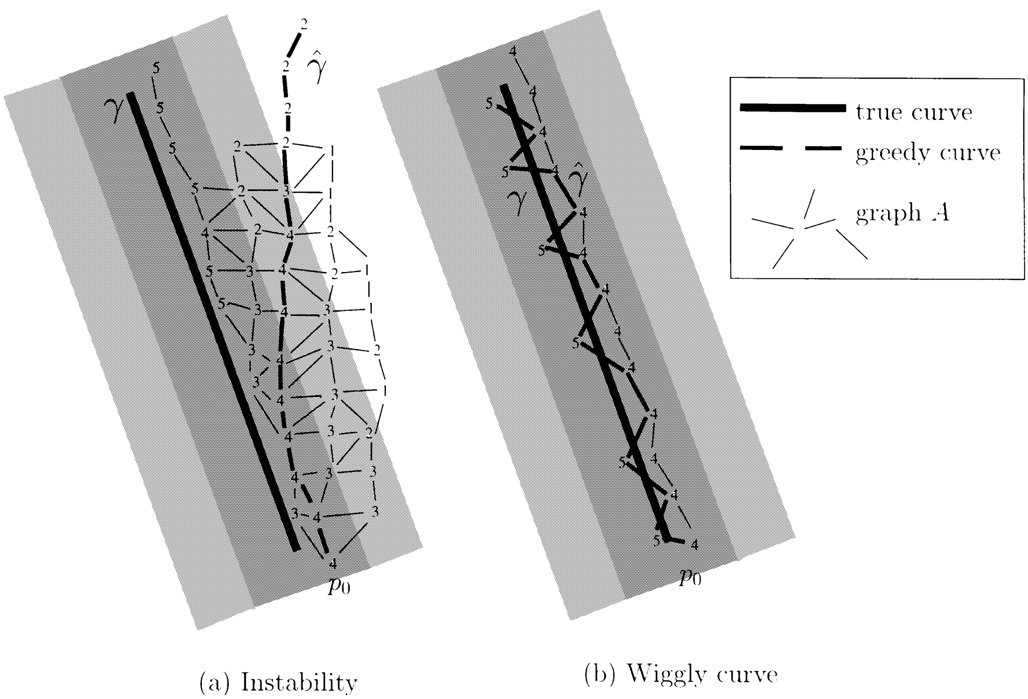
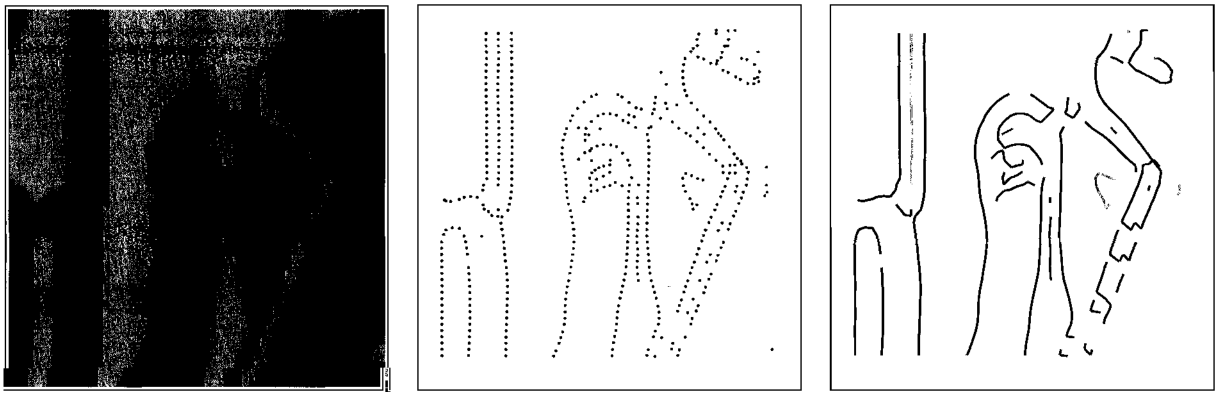


Figure 6. Noise can cause instability and wiggly curves in greedy tracking algorithms. (a) The model assumes that the vector field in the inner neighborhood D^0_γ (dark area) is larger than in the outer region $\mathbb{R}^2 \backslash D^1_\gamma$ (white area). Instability in curve tracking occurs because the maxima of $\phi(p)$ leak from D^0_γ into $D^1_\gamma \backslash D^0_\gamma$ (light gray areas). (b) Notice that noise can cause greedy linking to follow a wiggling path rather than the smoother path on the right.



(a) Brightness image (b) Edge-points (c) Greedy linking

Figure 7. Result of Canny's edge detection with greedy linking on the image shown in (a). The set of points found by Canny's algorithm is shown in (b). The polygonal curves obtained by linking each point to one of its neighbors are shown in (c). When ambiguities are present, a "best" neighbor is determined by minimizing a cost function which depends on the brightness gradient and on the orientation similarity between the linked points. Notice that these ambiguities, which are usually caused by multiple responses to the same edge, can disrupt the tracking process and lead to instability. That is, the reconstructed path can diverge significantly from the true edge. This is particularly evident for the two parallel edges of the bright thin long line on the right of the image.

(white area) but it is not necessarily larger than the vector field in the intermediate areas $D^1_\gamma \backslash D^0_\gamma$ (the light shaded areas).

Now suppose that a polygonal curve $\hat{\gamma}$ is constructed from the point p_0 by a greedy linking procedure. That is, the current point is linked to its strongest neighbor and then the procedure is restarted from the new point. Notice that the tracked path exits first the inner neighborhood, then the outer neighborhood and then it diverges arbitrarily from γ. Thus, this simple "greedy" procedure is *unstable* because a small deviation of the tracked path from the path closest to the curve can lead to an arbitrarily large deviation between the two paths. This type of error occurs frequently in real images if greedy linking is used. Several instances of this error are shown in Fig. 7. Another problem with greedy linking is that it might generate wiggly curves (see Fig. 6(b)).

3.2. Definition of Stability

An important definition in this paper is that of a stable graph. Roughly speaking, a graph is stable if every path in the graph "attracts" nearby paths. A weak definition of stability is given below and a stronger one will be given in Section 5. Both definitions depend on a positive constant w, which is the scale parameter used by the stabilization algorithm described in Section 4.

For $p \in \mathbb{R}^2$, $w > 0$, let $\mathbf{B}_w(p)$ be the ball centered at p with radius w:

$$\mathbf{B}_w(p) = \{p' \in \mathbb{R}^2 : \|p - p'\| < w\}.$$

The w-neighborhood of a path π is the set of points in $\mathbb{R}^2$ with distance from $\sigma(\pi)$ less than w. It is given by:

$$N_w(\pi) = \bigcup_{p \in \sigma(\pi)} \mathbf{B}_w(p).$$

Let $\pi = (p_1, \dots, p_n)$ and $\pi' = (p'_1, \dots, p'_{n'})$ be two paths and let $q \in \sigma(\pi')$. Let $\sigma_q(\pi')$ be the largest connected subcurve of $\sigma(\pi')$ which contains q and is separated by at least w from the end-points of π (see Fig. 8):

$$\sigma_q(\pi') \cap \mathbf{B}_w(p_1) = \sigma_q(\pi') \cap \mathbf{B}_w(p_n) = \emptyset.$$

In Fig. 8, $\sigma_q(\pi')$ is the curve between q^- and q^+.

Definition 2. A path π in A is a w-*attractor* if there is a neighborhood U of $\sigma(\pi)$, $U \subset N_w(\pi)$, such that $\sigma_q(\pi') \subset U$ for every path π' in A and every $q \in \sigma(\pi') \cap U$. The set U is called an *attraction basin* for π. A graph (P, A) is w-*stable* if every path in it is a w-attractor.

A different way to construct attraction basins from tangent vectors is described in (Parent and Zucker, 1989; David and Zucker, 1990). Their method is

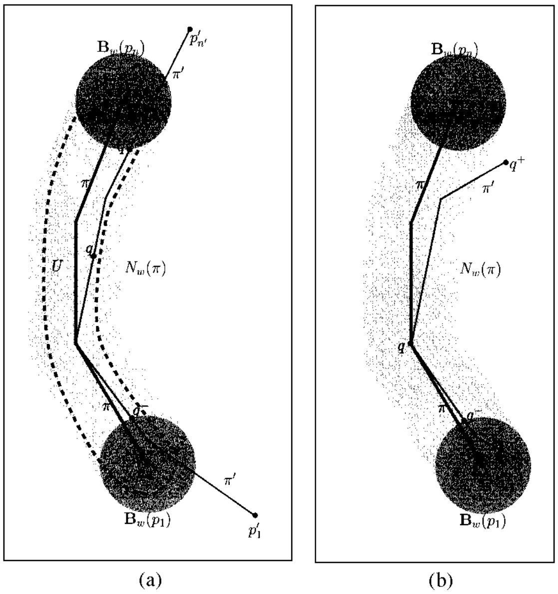


Figure 8. The path π in (a) is an attractor because the curve $\sigma_q(\pi')$, namely the curve between q^- and q^+, lies in some neighborhood U of $\sigma(\pi)$ contained in $N_w(\pi)$. The path π in (b) is not an attractor because there is no such neighborhood U of π. In fact, $\sigma_q(\pi')$ diverges (laterally) from π, that is, the subcurve $q \to q^+$ exits the neighborhood $N_w(\pi)$ without intersecting $\mathbf{B}_w(p_n)$.

based on a potential function obtained by summing the weighted contributions of several tangent vectors. The desired curves are then defined as the valleys of this potential. Each estimated curve is represented by a covering consisting of many smooth curve pieces. Each piece is computed by using a "snake" evolving according to the potential function and initialized near a tangent vector. Instead, the algorithm proposed here constructs attraction basins in a more geometric fashion (see Fig. 9) and reconstructs each curve as a whole entity by means of a linear-time dynamic programming procedure.

4. Stabilization of a Vector Graph

This section describes an algorithm which computes a stable graph from an arbitrary vector graph (P, V, A). Moreover, if (P, V, A) is a projection of Γ then the result is also a projection of Γ. The set of arcs in the computed graph is denoted $S^w(V, A)$ where w is the scale parameter. This parameter is related to the constants of the curve model by means of the bounds in Theorem 2 (see below). Ultimately, w should depend on the amount of noise in the brightness image and on the sharpness of the brightness discontinuity across edges. In the current implementation of the algorithm w has to be provided externally.

To obtain a stable graph, the algorithm ensures that every path $\pi = (p_1, \ldots, p_n)$ in $S^w(V, A)$ has an attraction basin $U_w(\pi)$ contained in $N_w(\pi)$. The boundary of $U_w(\pi)$, denoted $\beta_w(\pi)$, is a polygonal curve constructed as follows. For every $p \in P$ let p^+ and p^- be the points lying w away from p in the direction orthogonal to the vector field at p. That is,

$$p^+ = p + w u_\perp(p) \tag{4}$$

$$p^- = p - w u_\perp(p) \tag{5}$$

where $u_\perp(p)$ is the unit vector perpendicular to the direction of the vector field at p, $u_\perp(p) = (\sin\theta(p), -\cos\theta(p))$. The boundary of $U_w(\pi)$ is then the polygonal curve with vertices:

$$p_1^+, \ldots, p_n^+, p_n^-, \ldots, p_1^-, p_1^+$$

(see Fig. 9(a)). The algorithm can be described as follows. Let $\beta_w^-(a)$, $\beta_w^+(a)$ be the *lateral segments* of the arc $a = (p_1, p_2) \in A$ defined by (see Fig. 9(b)):

$$\beta_w^-(a) = \sigma(p_1^-, p_2^-) \tag{6}$$

$$\beta_w^+(a) = \sigma(p_1^+, p_2^+) \tag{7}$$

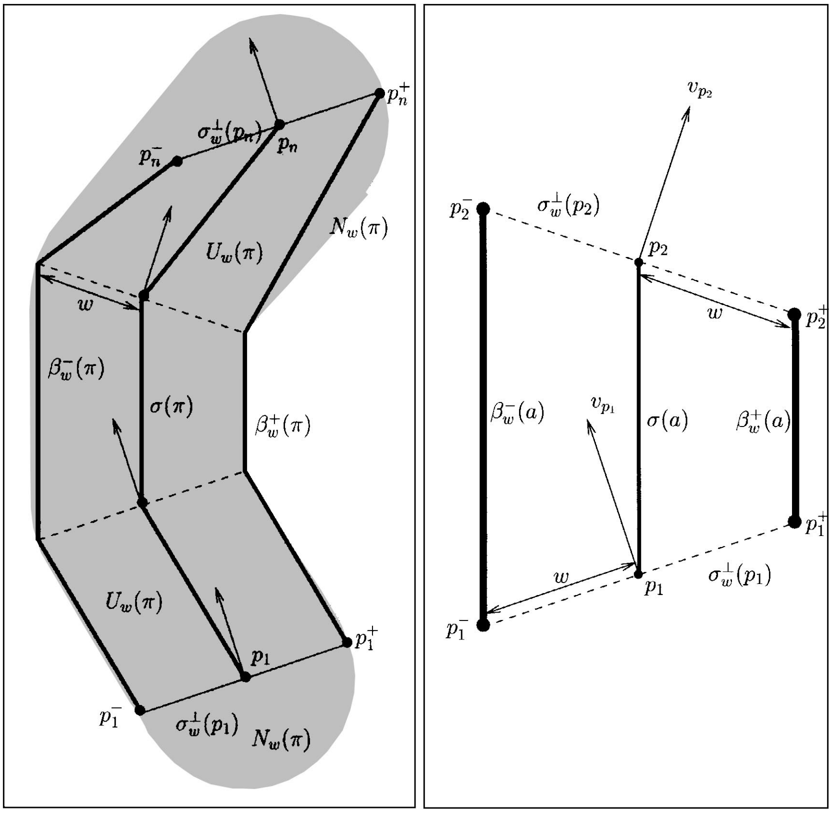
Let a_1, a_2 be arcs in A. If a_1 intersects one of the two lateral segments of a_2 or vice-versa then we say that (a_1, a_2) is an *incompatible* pair. Let's define a boolean function ψ_w: $A \times A \to \{\texttt{false}, \texttt{true}\}$ such that $\psi_w(a_1, a_2) = \texttt{true}$ if (a_1, a_2) is incompatible and $\psi_w(a_1, a_2) = \texttt{false}$ otherwise. Thus we have

$$\begin{aligned}\psi_w(a_1, a_2) = \; & \sigma(a_1) \cap \beta_w^-(a_2) \neq \emptyset \\ & \vee \sigma(a_1) \cap \beta_w^+(a_2) \neq \emptyset \\ & \vee \sigma(a_2) \cap \beta_w^-(a_1) \neq \emptyset \\ & \vee \sigma(a_2) \cap \beta_w^+(a_1) \neq \emptyset \end{aligned} \tag{8}$$

where $\vee$ denotes the logical "or" operator. Let I_w be the set of incompatible pairs of arcs in A. For every $(a_1, a_2) \in I_w$ let

- $E(a_1, a_2)$ be the four end-points of a_1 and a_2:

$$E(a_1, a_2) = \{p_1(a_1), p_2(a_1), p_1(a_2), p_2(a_2)\}$$



(a) Lateral boundaries of path π (b) Lateral segments of arc a

Figure 9. (a): The attraction basin $U_w(\pi)$ is the polygon with vertices $p_1^+, \ldots, p_n^+, p_n^-, \ldots, p_1^-, p_1^+$. Its perimeter $\beta_w(\pi)$ is composed of four parts: $\beta_w(\pi) = \beta_w^-(\pi) \cup \sigma_w^\perp(p_n) \cup \beta_w^+(\pi) \cup \sigma_w^\perp(p_1)$ where $\sigma_w^\perp(p_i) = \sigma(p_i^-, p_i^+)$; $\beta_w^\pm(\pi) = \bigcup_{a \in \mathrm{arcs}(\pi)} \beta_w^\pm(a)$; arcs$(\pi)$ are the arcs of π; and $\beta_w^\pm(a)$ are the lateral segments of a shown in (b). Notice that each point in $U_w(\pi)$ has distance from $\sigma(\pi)$ less than w, that is, $U_w(\pi) \subset N_w(\pi)$.

- $P_{\min}(a_1, a_2)$ be the set of points minimizing ϕ in $E(a_1, a_2)$:

$$\phi_0 = \min_{p \in E(a_1, a_2)} \phi(p) \tag{9}$$

$$P_{\min}(a_1, a_2) = \{p \in E(a_1, a_2) : \phi(p) = \phi_0\} \tag{10}$$

If ϕ takes different values on the elements of $E(a_1, a_2)$, then $P_{\min}(a_1, a_2)$ is a singleton. Let $\tilde{P}_w(V, A)$ be the union of all these minimum-achieving points over all pairs of incompatible arcs:

$$\tilde{P}_w(V, A) = \bigcup_{(a_1, a_2) \in I_w} P_{\min}(a_1, a_2) \tag{11}$$

The set of vertices of the computed graph is $P \backslash \tilde{P}_w(V, A)$ and

$$S^w(V, A) = \{(p_1, p_2) \in A : p_1, p_2 \notin \tilde{P}_w(V, A)\}. \tag{12}$$

By using the following notation

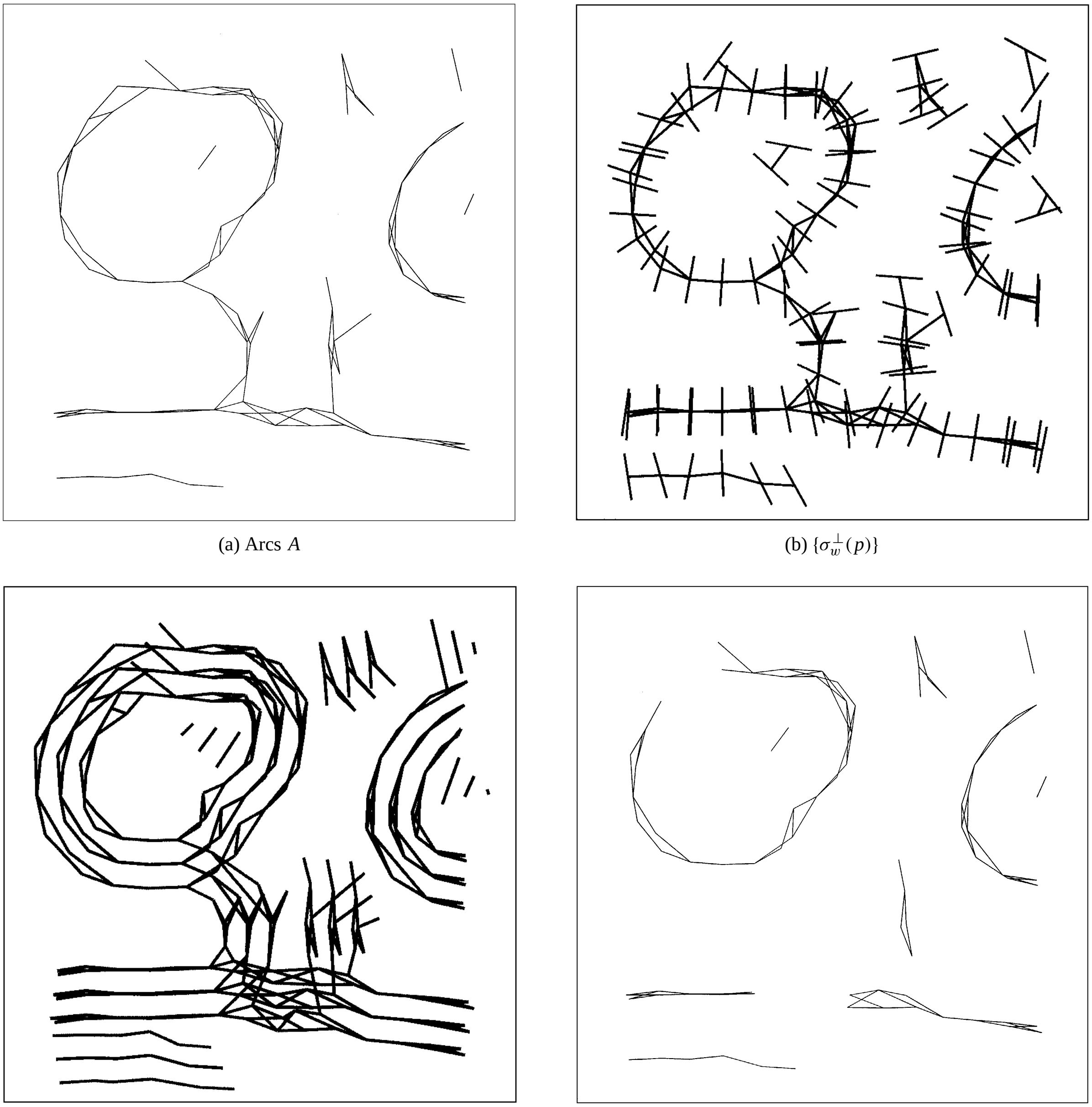
$$A - P' := \{(p_1, p_2) \in A : p_1, p_2 \notin P'\} \tag{13}$$

for any $P' \subset P$, Eq. (12) can be rewritten as $S^w(V, A) = A - \tilde{P}_w(V, A)$. The proofs of the following theorems are in Section 10. The result of the stabilization algorithm on the graph of Fig. 10(a) is shown in Fig. 10(d). Let $U_w(\pi)$ be as in Fig. 9.

Theorem 1. *For any vector graph (P, V, A) and $w > 0$, the graph with arcs $S^w(V, A)$ generated by the stabilization algorithm is w-stable with attraction basins $U_w(\pi)$.*

Let $l^{\max}(A)$ be the maximum arc length of the graph (P, A),

$$l^{\max}(A) = \max\{\|p_1 - p_2\| : (p_1, p_2) \in A\}$$



(a) Arcs A

(b) $\{\sigma_w^{\perp}(p)\}$

(c) $\{\beta_w^{\pm}(a)\}$

(d) $S^w(V, A)$

Figure 10. Stabilization of the graph in (a). (b) The segments $\sigma_w^{\perp}(p),\ p \in P$. (c) The lateral boundaries. (d) The result of the stabilization algorithm.

Theorem 2. *Let Γ be a set of curves with bounded curvature and let (P, V, A) be a projection of Γ with admissible deviations δ_0, δ_1, δ_2 and Θ_1. If*

$$\frac{2\delta_1}{\cos\Theta_1} < w \cdot (1 - \epsilon_1(\kappa)), \tag{14}$$

$$\delta_2 - \delta_1 > \max\{w, l^{\max}(A)\} \cdot (1 + \epsilon_2(\kappa)) \tag{15}$$

then the graph $S^w(V, A)$ covers Γ with distance δ_0. Namely, for every $\gamma \in \Gamma$, $S^w(V, A)$ contains a path π such that $d(\gamma; \sigma(\pi)) < \delta_0$.

In Theorem 2, κ denotes the maximum curvature of the curves in Γ and $\epsilon_1(\kappa)$, $\epsilon_2(\kappa)$ are positive functions such that $\epsilon_1(0) = \epsilon_2(0) = 0$. As a corollary of Theorems 1 and 2, notice that the vector graph with arcs $S^w(V, A)$ is a stable projection of Γ.

Since the asymmetric Hausdorff distance $d(\gamma; \sigma(\pi))$ is used, the estimated curve $\sigma(\pi)$ may be longer than the curve γ. This is a reasonable result since also the covering condition was based on the asymmetric Hausdorff distance.

5. Invalid End-Points and Strong Stability

In a stable graph, the tracked path is guaranteed to remain close to the true curve if this path contains a point sufficiently close to the curve. Thus, errors such as those in Fig. 6(a) can not occur in a stable graph. However, it is not guaranteed that all the paths near the curve are long enough to cover the curve completely. Therefore, the tracked path might terminate before reaching the end of the curve (see Fig. 11).

To avoid this problem, invalid end-points such as p_3 in Fig. 11 are connected to some collateral path by adding an arc (e.g., (p_3, p_5)) to the graph. To describe how this is done, a few definitions are needed. First of all, let us assume for simplicity that the relationship between the points in P and the other geometric entities is "generic", that is,

$$\sigma(p_1, p_2) \cap P = \{p_1, p_2\} \tag{16}$$

Table 1. The stabilization algorithm. Steps 2 and 4 need not be carried out over all pairs of arcs. In fact, for each arc, it is enough to consider all the arcs in a fixed neighborhood around its midpoint. If we further assume that the density of arcs in the image is upper bounded, then the complexity of the algorithm is linear in the number of arcs.

```
1  For every (a1, a2) ∈ A × A
2     compute ψ_w(a1, a2), as given by Eq. (8)
3  For every a1, a2 ∈ A such that ψ_w(a1, a2) = true
4     compute P_min(a1, a2) as given by Eq. (10)
5  P̃_w(V, A) = ⋃_{(a1,a2)∈I_w} P_min(a1, a2)
6  Return A − P̃_w(V, A)
```

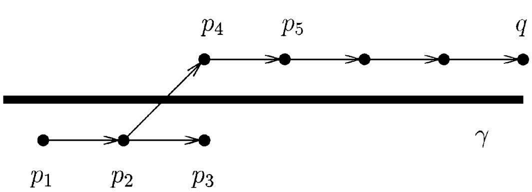
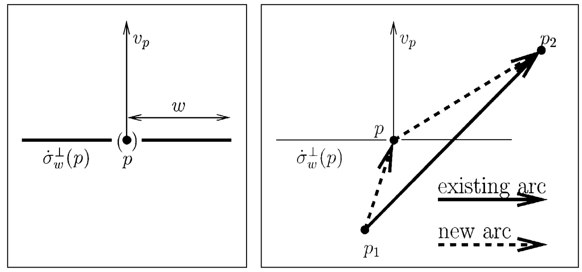


Figure 11. The path $(p_1, p_2, p_4, \ldots, q)$ covers γ whereas (p_1, p_2, p_3) does not (both paths are maximal). p_3 is said to be an "invalid" end-point. If during curve tracking p_3 is chosen instead of p_4 at the bifurcation point p_2 (this occurs if the most collinear arc with (p_1, p_2) is chosen), then the resulting path does not cover γ completely.



(a) (b)

Figure 12. (a) Definition of $\dot\sigma_w^\perp(p)$. (b) To deal with invalid end-points, p_1 is connected to p and p to p_2 whenever $\sigma(p_1, p_2)$ intersects $\dot\sigma_w^\perp(p)$.

$$\sigma_w^\perp(p) \cap P = \{p\} \tag{17}$$

$$\beta_w(\pi) \cap P = \{p_1, p_n\} \tag{18}$$

where $\pi = (p_1, \ldots, p_n)$. Also, without loss of generality, let us assume that (P, A) does not contain *isolated* vertices. Namely, every vertex in the graph (P, A) is connected to at least one other vertex. Let A_p^{in} and A_p^{out} be the set of in-arcs and out-arcs from p, that is $A_p^{\text{in}} = \{a \in A : p_2(a) = p\}$, $A_p^{\text{out}} = \{a \in A : p_1(a) = p\}$. A path $\pi = (p_1, \ldots, p_n)$ is said to be *maximal in A* if $A_{p_1}^{\text{in}} = A_{p_n}^{\text{out}} = \emptyset$. Let (see Fig. 12(a))

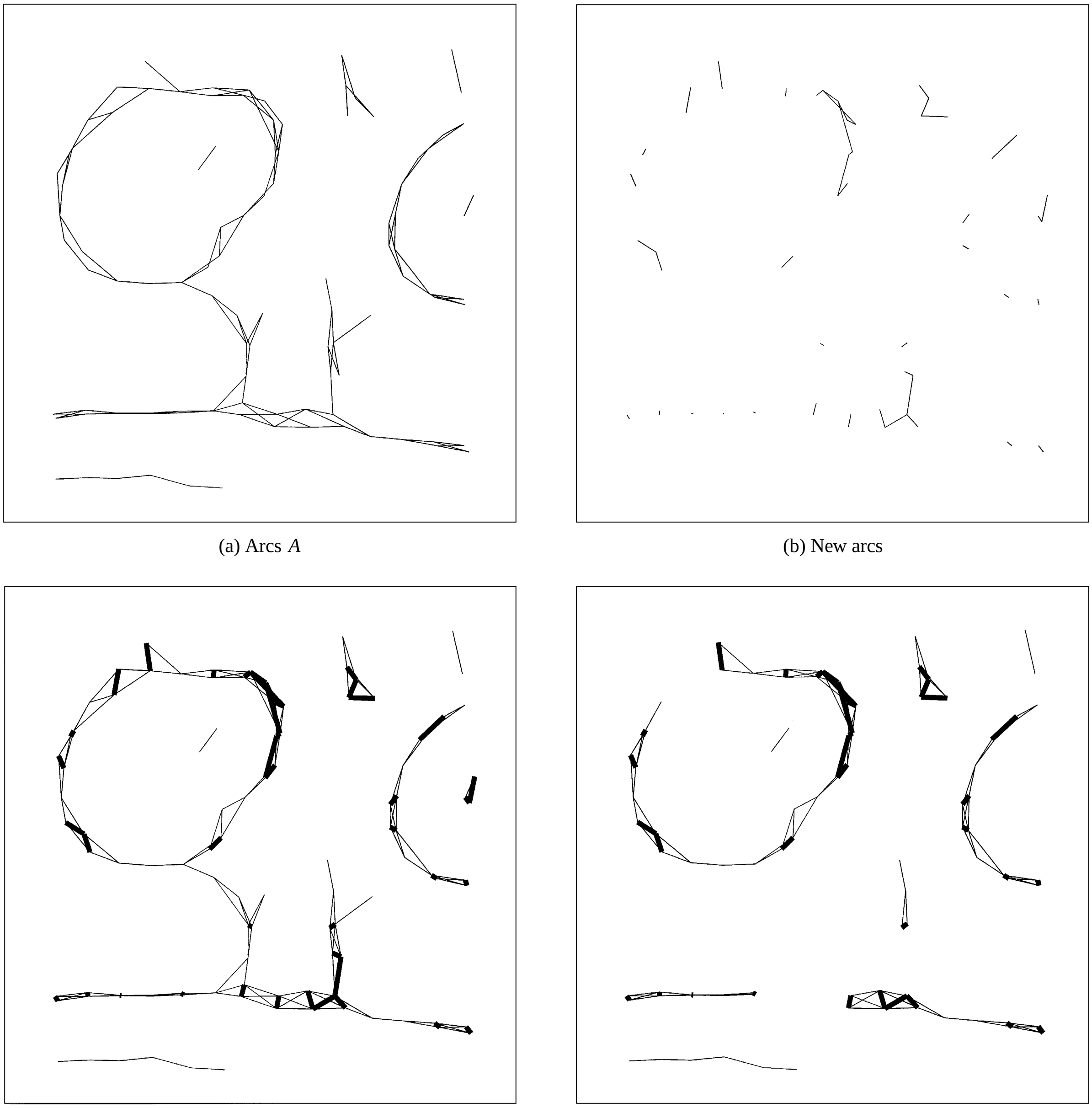
$$\dot\sigma_w^\perp(p) = \sigma_w^\perp(p) \backslash \{p\}$$

Definition 3. A point $p \in P$ is an *end-point* if $A_p^{\text{out}} = \emptyset$ or $A_p^{\text{in}} = \emptyset$. An end-point p is *invalid* if $\sigma(p_1, p_2) \cap \dot\sigma_w^\perp(p) \neq \emptyset$ for some $(p_1, p_2) \in A$.

Notice that if π is a path terminating at an invalid end-point p, then there might be a collateral path of π which "prolongs" it. This longer path contains the arc (p_1, p_2) such that $\sigma(p_1, p_2) \cap \dot\sigma_w^\perp(p) \neq \emptyset$.

To ensure that tracking does not get stuck at an invalid-end point, a new graph with arcs $\bar{A} \supset A$ is constructed as follows. Initially, let $\bar{A} = A$. Then, for any p such that $\sigma(p_1, p_2) \cap \dot\sigma_w^\perp(p) \neq \emptyset$, $(p_1, p_2) \in A$, add the arcs (p_1, p) and (p, p_2) to $\bar{A}$.[1] The new graph $\bar{A}$ obtained from the graph in Fig. 13(a) is shown in Fig. 13(d). By stabilizing the graph with arcs $\bar{A}$ one obtains a graph which possesses the following strong stability property. Let A, $\hat{A}$ be sets of arcs.

Definition 4. A path π in $\hat{A}$ is a *strong attractor with respect to A* if there is a neighborhood U of $\sigma(\pi)$, $U \subset N_w(\pi)$, such that $\sigma(\pi') \cap U \neq \emptyset$ implies $\sigma(\pi') \subset \bar{U}$ for every path π' in $\hat{A} \cap A$. $\hat{A}$ is *strongly w-stable*



(a) Arcs A (b) New arcs

(c) $\bar{A}$ (d) $S^w(V, \bar{A})$

Figure 13. Computation and stabilization of $\bar{A}$. (a) Initial graph. (b) The arcs needed to connect invalid end-points to a collateral path. (c) The graph $\bar{A}$ with the new arcs shown in bold. (d) The graph obtained by stabilizing $\bar{A}$. This graph is strongly stable.

with respect to A if every maximal path in $\hat{A}$ is a strong attractor with respect to A.

Here, $\bar{U}$ denotes the topological closure of U.

Theorem 3. *For any vector graph (P, V, A) and $w > 0$, the graph with arcs $S^w(V, \bar{A})$ is strongly w-stable with respect to A with attraction basins $U_w(\pi)$.*

A result similar to Theorem 2 holds also for $S^w(V, \bar{A})$. The only difference is that the lower bound on $\delta_2 - \delta_1$ is now proportional to $w + l^{\max}(A)$ rather than w.

Theorem 4. *Let Γ be a set of curves with bounded curvature and (P, V, A) a projection of Γ with* *admissible deviations* δ_0, δ_1, δ_2 *and* Θ_1. *If*

$$\frac{2\delta_1}{\cos\Theta_1} < w \cdot (1 - \epsilon_1(\kappa)), \tag{19}$$

$$\delta_2 - \delta_1 > (w + l^{\max}(A)) \cdot (1 + \epsilon_2(\kappa)) \tag{20}$$

then, *the graph* $S^w(V, \bar{A}) \cap A$ *covers* Γ. *Namely*, *for every* $\gamma \in \Gamma$, $S^w(V, \bar{A}) \cap A$ *contains a path* π *such that* $d(\gamma; \sigma(\pi)) < \delta_0$.

Notice that, as a corollary of Theorems 3 and 4, $S^w(V, \bar{A}) \cap A$ is a strongly stable projection of Γ.

6. Computing a Path Covering

In the previous sections a strongly stable projection of Γ with arcs $S^w(V, \bar{A})$ was constructed. This section addresses the problem of computing a family of disjoint paths $\{\pi_1, \ldots, \pi_N\}$ in $S^w(V, \bar{A})$ which covers Γ. It is assumed for now that $S^w(V, \bar{A})$ does not contain any cycles. The more general case where cycle can be present is treated in the next section.

Recall that the length of an arc in A is less or equal to $l^{\max}(A)$ whereas arcs in $S^w(V, \bar{A})$ have lengths less or equal to $l^{\max}(A) + w$. Thus, elements in $S^w(V, \bar{A}) \cap A$ will be called *short arcs* and the other elements of $S^w(V, \bar{A})$ will be called *long arcs*. Notice that the subgraph of short arcs is sufficient to cover Γ. Long arcs have been added to ensure that all maximal paths are sufficiently extended to cover the tracked curve.

The paths $\pi_1, \ldots, \pi_N$ are computed one at a time by an iterative procedure. This procedure extracts a maximal path π_j from the current search graph A_j and then defines the new search graph $A_{j+1} \subset A_j$ by removing from A_j all arcs with a vertex in the neighborhood $U_w(\pi_j)$. This is continued until the search graph is empty. Existence of a maximal path in A is guaranteed by the assumption that the graph does not contain cycles. The algorithm is shown in Table 2.

Table 2. The algorithm to compute regular curves from $S^w(V, \bar{A})$. A *source* in A_j is a vertex with no in-arcs. In line 6, $\mathrm{ver}(A_j)$ denotes the set of vertices of the graph A_j. The procedure *maximalPath*(s_j, A_j) returns a maximal path in A_j with starting point s_j (see Section 6.1).

1	$A_1 = S^w(V, \bar{A})$
2	$j = 1$
3	Do until $A_j = \emptyset$
4	pick a source s_j in A_j
5	$\pi_j =$ *maximalPath*(s_j, A_j)
6	$Q_j = \mathrm{ver}(A_j) \cap U_w(\pi_j)$
7	$A_{j+1} = A_j - Q_j$
8	$j = j + 1$
9	end do

Since $U_w(\pi_j)$ is a neighborhood of $\sigma(\pi_j)$, all the vertices of π_j belong to $U_w(\pi_j)$, except for the two end-points (which belong to the boundary of $U_w(\pi_j)$). Thus, if π_j has at least three vertices, then no arc of π_j belongs to A_{j+1} because every arc of π_j has at least one vertex in $U_w(\pi_j)$ (see lines 6 and 7 of Table 2). That is, we have

$$\mathrm{arcs}(\pi_j) \cap A_{j+1} = \emptyset \tag{21}$$

where $\mathrm{arcs}(\pi_j)$ denotes the arcs belonging the the path π_j. If π_j has just two vertices, then one needs to modify slightly the algorithm shown in Table 2 so that (21) is still true. We omit these details for the sake of simplicity. From (21) (and $\mathrm{arcs}(\pi_j) \subset A_j$) we have that A_{j+1} is a proper subset of A_j. Since A_1 is finite, this implies that the algorithm terminates after a finite number of steps. Moreover, it implies that the paths π_j are arc-disjoint, that is,

$$\mathrm{arcs}(\pi_j) \cap \mathrm{arcs}(\pi_k) = \emptyset, \quad j \neq k \tag{22}$$

Let $\hat{\Gamma} = \{\sigma(\pi_j) : 1 \leq j \leq N\}$.

Theorem 5. *Let* Γ *be a set of curves with bounded curvature and let* (P, V, A) *be projection of* Γ *with admissible deviations* δ_0, δ_1, δ_2 *and* Θ_1. *If*

$$\frac{2\delta_1}{\cos\Theta_1} < w \cdot (1 - \epsilon_1(\kappa)), \tag{23}$$

$$\delta_2 - \delta_1 > (w + l^{\max}(A)) \cdot (1 + \epsilon_2(\kappa)). \tag{24}$$

then for every $\gamma \in \Gamma$ *there exists* $\hat{\gamma} \in \hat{\Gamma}$ *such that* $d(\gamma; \hat{\gamma}) < w + \delta_0$.

Notice that, in the zero curvature case, and if w is set to the lower bound given by Eq. (23), then the localization error is

$$\delta_0 + 2\frac{\delta_1}{\cos\Theta_1}.$$

The parameters δ_0 and δ_1 represent two different types of localization uncertainty in the local data whereas Θ_1 is an upper bound on orientation uncertainty. It is not clear how tight this bound is. Probably, the factor 2

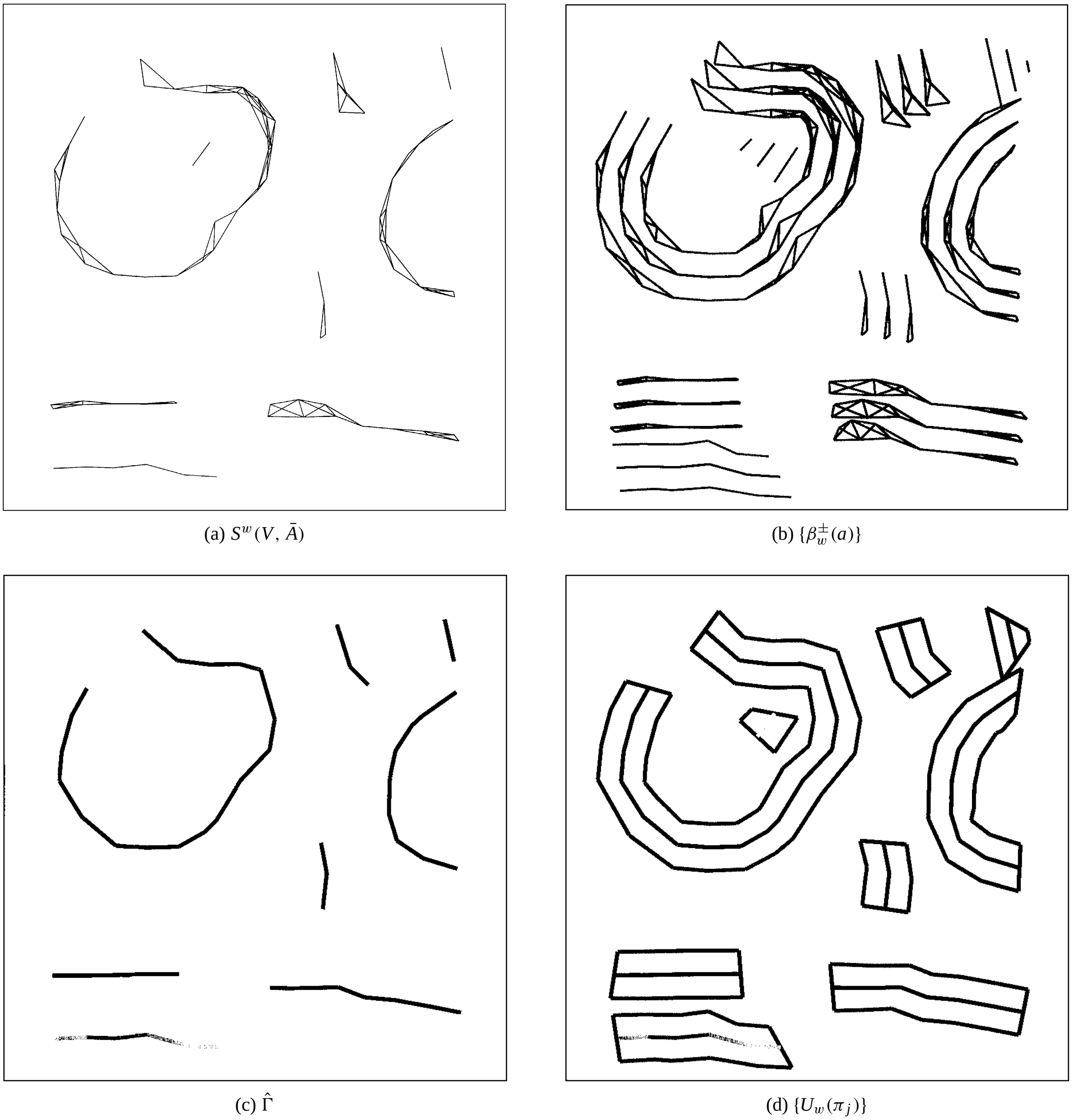


Figure 14. Computation of regular curves from the strongly stable graph $S^w(V, \bar{A})$ in (a). (b) The lateral boundaries formed by the lateral segments $\{\beta_w^{\pm}(a)\}$. (c) The regular curves computed by the algorithm. (d) The neighborhoods $\{U_w(\pi_j)\}$.

can be reduced but it is unlikely that it can be made smaller than 1.

The parameter δ_2, which represents the minimum separation distance between two curves, has to be at least $3\delta_1 + l^{\max}(A)$ (by letting $\kappa = \Theta_1 = 0$). The $l^{\max}(A)$ part can be made arbitrarily small by breaking down the arcs of the vector graph into smaller pieces. This entails a linear increase of the computational complexity. As for the other term, $3\delta_1$, we do not know how tight it is.

6.1. Computing the Optimal Path

The algorithm in Table 2 uses the procedure *maximal-Path*(s_j, A_j) which returns a path $\pi_j \in M_j(s_j)$, where $M_j(s_j)$ denotes the set of maximal paths in A_j with first element s_j. Notice that Theorem 5 holds for any choice of a path in $M_j(s_j)$. Thus, this path can be determined according to an arbitrarily chosen cost function $c(\pi)$. If this cost is additive,

$$c(\pi) = \sum_{i=1}^{n} c(p_i), \quad \pi = (p_1, \ldots, p_n)$$

then the minimizing path can be computed by using an efficient dynamic programming algorithm which runs in linear time in the number of arcs in the graph. In fact, let $c^\star(p)$, $p \in \text{ver}(A_j)$, be the optimal cost of a maximal path starting at p. Then, the following Bellman equation holds

$$c^\star(p) = c(p) + \min_{p' \in F(p)} c^\star(p'), \quad p \in \text{ver}(A_j)$$

where $F(p)$ denotes the set of vertices p' such that $(p, p') \in A_j$. The recursive algorithm in Table 3 can be used to compute $c^\star(p)$ for all $p \in \text{ver}(A_j)$ (assuming the graph does not contain cycles). The optimal path starting from p is then

$$\begin{aligned} &\mathit{maximalPath}(s_j, A_j) \\ &\quad = (s_j, \nu(s_j), \nu(\nu(s_j)), \ldots, \nu^n(s_j)) \end{aligned} \tag{25}$$

where $\nu(p)$ is the minimizer of $c^\star(p')$, $p' \in F(p)$.

Notice that this method can be generalized to cost functions of the form

$$c(\pi) = \sum_{i=1}^{n-k} c(p_i, p_{i+1}, \ldots, p_{i+k}).$$

To do this, one needs to construct a graph whose nodes are all possible $(k+1)$-paths $(q_1, \ldots, q_k)$ in the original graph and whose arcs are all the pairs $((q_1, \ldots, q_k), (q'_1, \ldots, q'_k))$ where $q'_i = q_{i+1}$, $i = 2, \ldots, k-1$ (see Casadei, 1995).

Table 3. The recursive algorithm to compute optimal maximal paths.

```
optimize(p)
    if F(p) = Ø
        c*(p) = c(p)
    else
        for every p' ∈ F(p)
            optimize(p')
        c*(p) = c(p) + min{c*(p') : p' ∈ F(p)}
return
```

This approach can be used to compute paths with minimum total turn. In fact, the total turn of a path $\pi = (p_1, \ldots, p_n)$ is given by

$$c(\pi) = \sum_{i=2}^{n-1} c(p_{i-1}, p_i, p_{i+1})$$

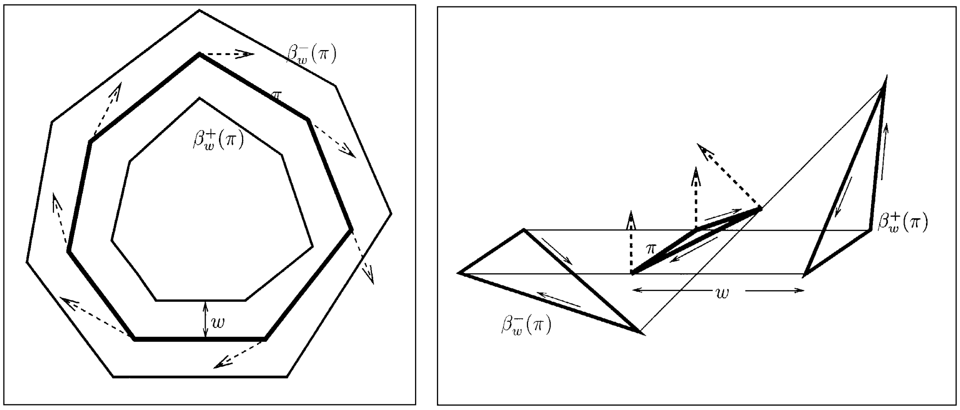
where $c(p_{i-1}, p_i, p_{i+1})$ is the absolute value of the orientation differences between the arcs (p_{i-1}, p_i) and (p_i, p_{i+1}). Similarly one could use this method to incorporate other types of constraints (e.g., energy-based) depending also on brightness.

A dynamic programming approach for optimal curve detection was also proposed in (Subirana-Vilanova and Sung, 1992; Sha'ashua and Ullman, 1988). Their method minimizes a cost function which penalizes curvature and favors the total length of the curve. Our approach is simpler in that only curvature appears in the cost function. This simplification is possible because all the curves which can be constructed from a given point cover each other, i.e., they have the same "length". This is a consequence of the strong stability of the graph.

Due to the above simplification, the dynamic programming algorithm needs to pass through each point only once and therefore has linear time complexity. Also, since the stabilization algorithm runs in linear time (see comments in Table 1) and since the procedure which computes $\bar{A}$ from A runs also in linear time, it follows that the whole algorithm has linear time complexity.

7. Classification and Detection of Cycles

The curve extraction procedure of Section 6 needs some adjustment to deal with cycles in the vector graph. Recall that a cycle is a path $\pi = (p_1, \ldots, p_n)$ such that $p_1 = p_n$. The structure of the stabilized graph $S^w(V, \bar{A})$ makes it possible to classify cycles into two classes, *regular* cycles and *irregular* cycles (or *looplets*). To do this, notice first of all that the two lateral boundaries $\beta_w^+(\pi)$ and $\beta_w^-(\pi)$ of a cycle are closed curves (see Fig. 15). Since $S^w(V, \bar{A})$ is the output of the stabilization algorithm, the closed polygonal curve $\sigma(\pi)$ is disjoint from $\beta_w^+(\pi)$ and $\beta_w^-(\pi)$ (see Proposition 2 in Section 10). Thus, two cases are possible. In the first case, $\sigma(\pi)$ encloses either $\beta_w^+(\pi)$ or $\beta_w^-(\pi)$. Then, the other lateral boundary encloses the other two closed curves (see Fig. 15(a)). A cycle of this type is said to be *regular*. In the second case,



(a) Regular cycle (b) Irregular cycle (looplet)

Figure 15. (a) The cycle π is regular because $R(\beta_w^+(\pi)) \subset R(\pi) \subset R(\beta_w^-(\pi))$. Here, $R(\cdot)$ denotes the region enclosed by a closed curve. (b) A looplet π satisfies $R(\beta_w^+(\pi)) \cap R(\pi) = R(\beta_w^-(\pi)) \cap R(\pi) = \emptyset$.

neither $\beta_w^+(\pi)$ nor $\beta_w^-(\pi)$ encloses $\sigma(\pi)$ and the interiors of the three cycles are all disjoint (Fig. 15(b)). A cycle of this kind will be called a *looplet*, or irregular cycle.

The procedure *optimize*(p) in Table 3 can be modified slightly so that cycles are detected and handled appropriately. The modified procedure *optimize*$'$ works as follows. Every point p is assigned a state variable which can take one of three possible values: "new" (which is the initial state), "active" and "done". When a point is opened for the first time, namely *optimize*$'$ is called with argument p, then the state of p is changed from "new" to "active". When the procedure *optimize*$'(p)$ returns, the state of p is changed from "active" to "done". A cycle occurs whenever *optimize*$'$ opens a point which is already "active".

To check whether the cycle is regular it is sufficient to pick one of its vertices and verify whether it is enclosed by either $\beta_w^+(\pi)$ or $\beta_w^-(\pi)$. If the cycle is regular then the path *maximalPath*(s_j, A_j) in (25) is set equal to this cycle and lines 7, 8 of the procedure in Table 2 are applied to it. If the cycle is a looplet then all its nodes are coalesced into a "super-node" and the procedure *optimize*$'$ continues from this new super-node.

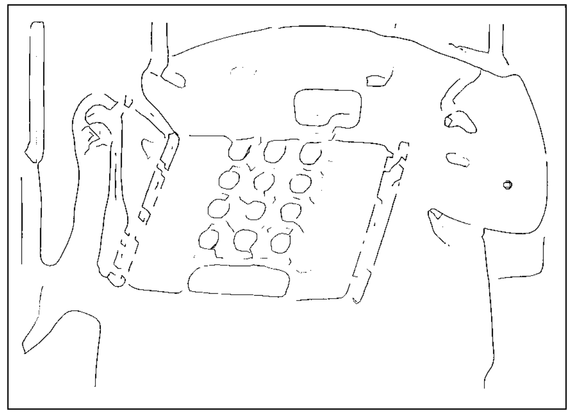
8. Experiments

The results of the proposed algorithm on two images are shown in Figs. 16 and 17. The vector graph was computed by using the method described in Section 2.1. Rectangular regions with width 4 and height 3 were used to compute the cubic polynomial fit and the tangent vectors. Some thresholds were used to eliminate some of these tangent vectors. The parameter w was set to 0.75 for all the experiments.

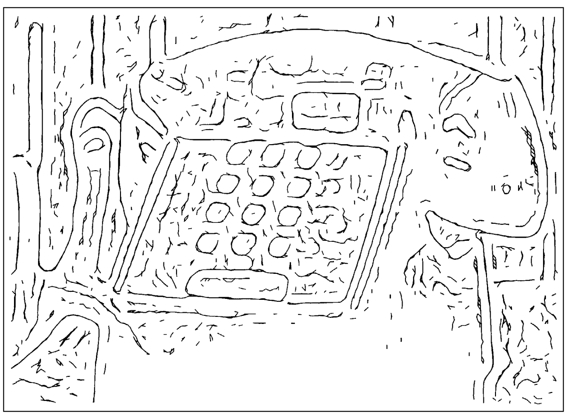
The results are compared to Canny's algorithm with sub-pixel accuracy followed by greedy edge linking (Figs. 16(b) and 17(b)). The performance of the two algorithms is comparable on edges which are isolated, with low curvature and with significant brightness change. However, when the topology of the edges is more complex, the advantages of a more robust approach become evident. In fact, greedy linking connects points with very little knowledge about the semi-local curvilinear structure. If an unstable bifurcation is present, then greedy linking makes a blind choice which may lead to a curve diverging from the true edge. For instance, the two parallel edges on the left of telephone keyboard are disconnected and merged together in Fig. 16(b). The contour of the telephone keys and of the flower petals (Fig. 17(b)) are often confused with nearby edges. One could decrease the chances of greedy linking getting confused by raising the thresholds in Canny's edge detector, so that fewer multiple responses are generated. However this would cause many edges to be missed (see the two parallel edges on the right of the telephone keyboard in Fig. 16(b)). Instead, the contours obtained by the proposed algorithm are more complete (even though sometimes disconnected) and do not diverge from the true edges (however, notice the errors near the top keys).

The limits of the proposed algorithm lie in the assumptions which are necessary for a curve to be recovered without disconnections. If a curve in the

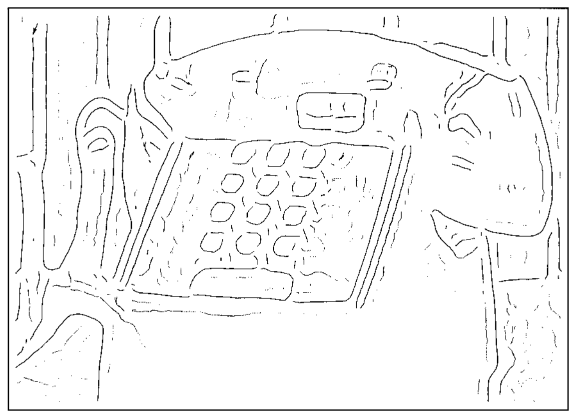
(a) Brightness image



(b) Canny's algorithm (for comparison)



(c) Vector graph



(d) Regular curves

Figure 16. Telephone image. The final result is shown in (d).

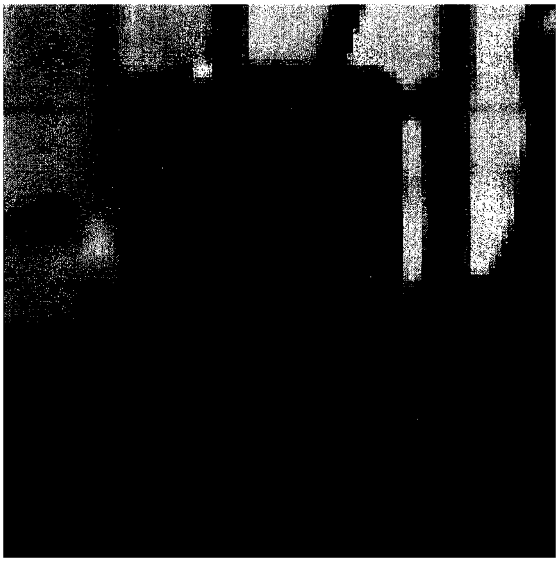
image violates one of these assumptions, then the curve might be disconnected at the point where the violation occurs. For instance, notice that edges are broken at points of high curvature and at curve singularities such as T-junctions. Also, since the asymmetric Hausdorff distance is used to evaluate the match between the true curve and the reconstructed one, curve end-points may not be recovered correctly. Sometimes, curves are extended a little beyond the ideal end-point.

These limitations can be dealt with at higher levels of the edge detection hierarchy by using more general edge models. The regular curves extracted at the current level can be used to construct hypotheses about the missing portions of the edges. More global information can then be applied to prune the set of these hypotheses so that the process can be repeated recursively and efficiently. For instance, Casadei and Mitter (1996a) showed how this method can be used to bridge small gaps in the curves due to high curvature and T-junctions (see Fig. 18).

Finally, Fig. 19 shows the results of the algorithm (including the edge continuation step) on two MRI images (courtesy of A. Tannenbaum). Since the brightness discontinuities occur at a higher scale in these images, the tangent vectors V have been generated by using rectangular regions 6 pixel wide and 7 pixel tall (the parameter w was kept the same as before, 0.75).

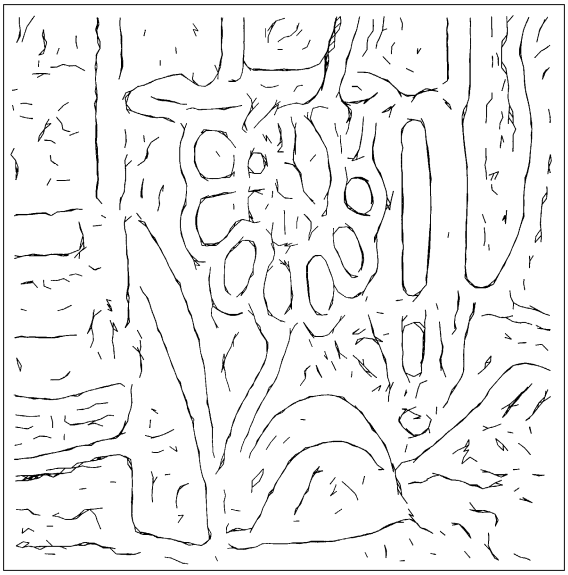
9. Conclusions

To estimate contours in an image reliably it is necessary to apply different types of local and global information. For the sake of computational efficiency, this information should be introduced in several stages so that uncertainties are resolved in the right context. The

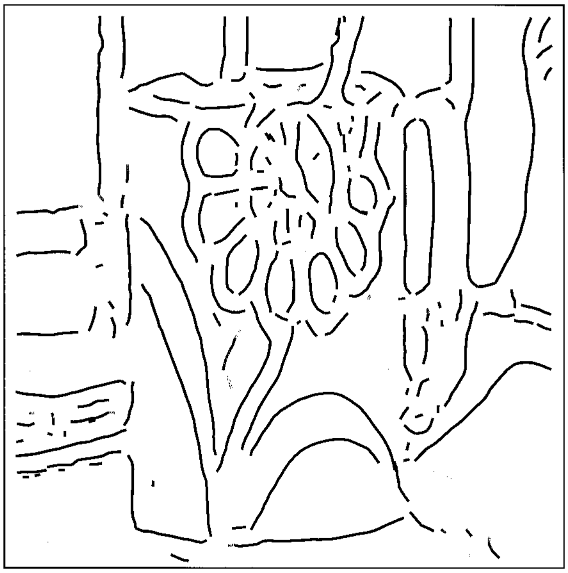


(a) Brightness image

(b) Canny's algorithm (for comparison)



(c) Vector graph



(d) Regular curves

Figure 17. Flower image. The final result is shown in (d).

state of the computational process is given by a set of curve hypotheses represented by geometric descriptors whose complexity and spatial support increase as computation proceeds. When building a new layer in the hierarchy, care has to be taken so that deriving the new representation from the previous ones can be done efficiently without resolving uncertainties arbitrarily. This remark has two consequences.

First, a theoretical framework is needed to prove that, at every stage, the computed representation approximates *all* the possible curve hypotheses which can be expressed by the dictionary of that stage. Indeed, in this paper it has been proved that the proposed algorithm approximates efficiently all the regular curve hypotheses which are consistent with the given set of tangent vectors.

Figure 18. Result after selecting locally-best edge continuations in Figs. 16(d) and 17(d).

Second, constraints exist on the *order* in which uncertainties should be resolved. In fact, only certain types of hypotheses can be formulated efficiently from the information available at the current stage. For the problem of edge detection, this led us to believe that curve singularities (corners and junctions) and invisible contours should be recovered only after representing the regular portions of the contours by means of maximally long curves.

A model for these regular curves has been proposed. The most important assumption of this model is that the local edge-strength decays away from the edge. The precise formulation of this assumption allows for the presence of noise in the model and makes the algorithm robust to noise. Correct detection of the regular curves described by this model entails the resolution of ambiguities due to multiple responses to the same edge and uncertainties in the orientation and strength of the point-like edge estimates. These issues are closely related to the problem of stability in edge tracking.

9.1. Future Work

An important generalization of the algorithm presented here is to include automatic adaptation to the scale of brightness discontinuities. If this scale can be estimated directly from local brightness data (Elder and Zucker, 1996a) then each tangent vector in the input can be labeled with this scale estimate. Then, the parameter w, which is now constant throughout the image, becomes a function of the local scale and noise estimates.

The information about multiple tangents (corners and junctions) present in the given set of tangent vectors V and removed during the stabilization procedure needs to be reintegrated into the representation at the next stage. This was partly done in (Casadei and Mitter, 1996a) but needs to be done in a more rigorous way. Also, some corners and junctions in the image might not be represented at all in V, especially if V was computed by an algorithm which assumes a single tangent at every point. Thus one perhaps needs to create new edge hypotheses by extrapolating the computed regular curves.

The problem of detecting large gaps between visible curves requires different steps. First of all, one needs to generate all possible invisible-curve hypotheses connecting visible curves. Some diffusion process emanating from the existing curves can be used for this purpose (see for instance Williams and Jacobs, 1995; Geiger and Kumaran, 1996). Alternatively, one could use some localized versions of the Hough transform to cluster curves in parameter space.

The set of all invisible-curve hypotheses together with the visible curves generates a graph where paths corresponds to partially invisible curves. This graph might contain many bifurcations (nodes with many arcs) so that finding "optimal" paths is in general

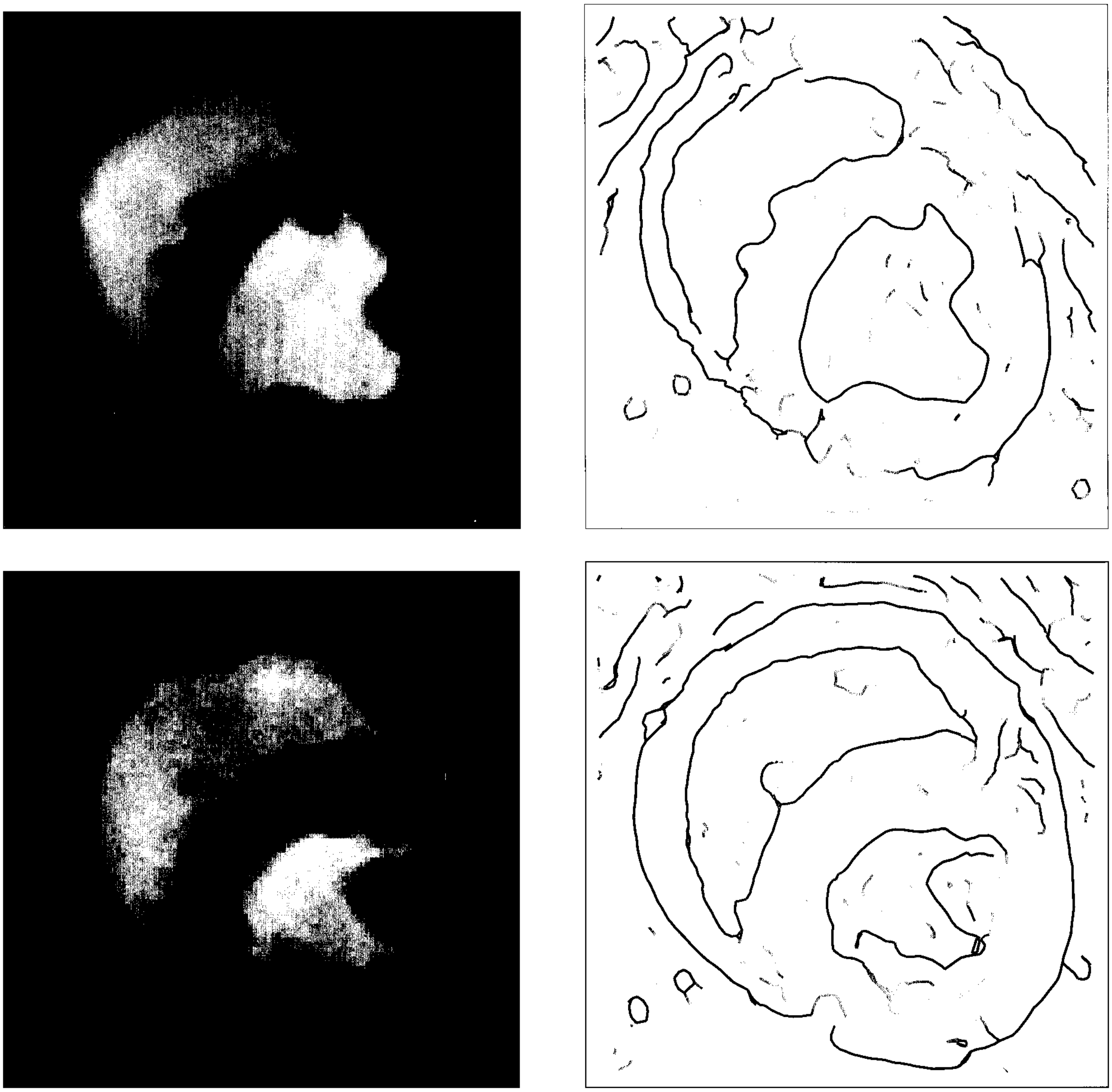


Figure 19. Left: Two MRI of an heart. Right: Result of proposed algorithm (followed by edge continuation processing).

computationally hard. As noted by Geiger et al. (1996), the major problem is not hypothesizing possible curve continuations or their shape but selecting the globally-best arrangements of continuations.

To do this efficiently, one needs to identify top-down feedback mechanisms to guide the search in the curve graph. Closure (Elder and Zucker, 1996b) can provide an important clue. However, for closure information to be useful one needs a continuous measure of "being closed" which applies to both closed and non-closed curves. This is necessary to generate feedback signals to curve hypotheses which are not yet completely formed into a closed contour.

At some stage, information about relative depth of the curves has to be included to facilitate continuation behind occluding contours, which should be the first ones to be detected and "lifted" from the image (Geiger et al., 1996). Eventually, at the highest level, a 2.1 sketch of the whole image should emerge from the computational process.

10. Proofs of the Results

10.1. Proof of Theorem 1

The proofs of the main results will be preceded by some useful propositions.

Definition 5. A vector graph (P, V, A) is said to be *arc-compatible* if $\psi_w(a_1, a_2) = \texttt{false}$ for every pair of arcs $(a_1, a_2) \in A \times A$. The set A will also be said to be arc-compatible.

Proposition 1. *For any vector graph (P, V, A) and $w > 0$, $S^w(V, A)$ is arc-compatible.*

Proof: Let $\psi_w(a_1, a_2) = \texttt{true}$ for some $a_1, a_2 \in A$. Then, $\tilde{P}_w(V, A)$ contains at least one of the points in $E(a_1, a_2)$. Hence either $a_1 \notin A - \tilde{P}_w(V, A)$ or $a_2 \notin A - \tilde{P}_w(V, A)$ so that $S^w(V, A)$ can not contain both a_1 and a_2. □

Proposition 2. *Let A be arc-compatible. Then for any paths π, π' in A:*

$$\sigma(\pi) \cap \beta_w^-(\pi') = \sigma(\pi) \cap \beta_w^+(\pi') = \emptyset \tag{26}$$

Proof: Notice that

$$\sigma(\pi) = \bigcup_{a \in \mathrm{arcs}(\pi)} \sigma(a)$$
$$\beta_w^+(\pi') = \bigcup_{a' \in \mathrm{arcs}(\pi')} \beta_w^+(a')$$
$$\beta_w^-(\pi') = \bigcup_{a' \in \mathrm{arcs}(\pi')} \beta_w^-(a')$$

The result follows then from Proposition 1. □

Proposition 3. *Let (P, V, A) be a vector graph and $w > 0$. Then $U_w(\pi) \subset N_w(\pi)$ for any path π in the graph (P, A).*

Proof: The result follows directly from the construction of the boundary of $U_w(\pi)$ (see Fig. 9(a)). □

Proof of Theorem 1: Let $\pi = (p_1, \ldots, p_n)$ be a path in $S^w(V, A)$. From Proposition 3 we have $U_w(\pi) \subset N_w(\pi)$. We have to prove that $U_w(\pi)$ is an attraction basin for π (see Definition 2). Recall that the boundary of $U_w(\pi)$ is

$$\begin{aligned}\partial U_w(\pi) &= \beta_w(\pi) \\ &= \beta_w^-(\pi) \cup \sigma_w^\perp(p_n) \cup \beta_w^+(\pi) \cup \sigma_w^\perp(p_1)\end{aligned}$$

Let $q \in U_w(\pi)$ and let π' be a path such that $\sigma(\pi')$ contains q. From Proposition 2 it follows that

$$\sigma(\pi') \cap \beta_w^-(\pi) = \sigma(\pi') \cap \beta_w^+(\pi) = \emptyset$$

so that $\sigma(\pi')$ can intersect $\beta_w(\pi)$ only at its extremities, $\sigma_w^\perp(p_1)$ and $\sigma_w^\perp(p_n)$, which are contained in $\mathbf{B}_w(p_1) \cup \mathbf{B}_w(p_n)$:

$$\begin{aligned}(\sigma(\pi') \cap \beta_w(\pi)) &\subset (\sigma_w^\perp(p_1) \cup \sigma_w^\perp(p_n)) \\ &\subset (\mathbf{B}_w(p_1) \cup \mathbf{B}_w(p_n))\end{aligned}$$

Therefore, $\sigma(\pi')$ can intersect $\partial U_w(\pi)$ only inside $\mathbf{B}_w(p_1) \cup \mathbf{B}_w(p_n)$. Hence $\sigma_q(\pi')$—namely the largest connected subcurve of $\sigma(\pi')$ disjoint from $\mathbf{B}_w(p_1) \cup \mathbf{B}_w(p_n)$—is contained in $U_w(\pi)$. Thus, $U_w(\pi)$ is an attraction basin for π. □

10.2. Proof of Theorem 2

Lemma 1. *Let γ be an unbounded straight line and let (P, V, A) be a vector graph satisfying the alignment condition on γ. Let $a \in A$. If $\sigma(a) \subset D_\gamma^1$ and*

$$\frac{2\delta_1}{\cos\Theta_1} < w, \tag{27}$$

$$\delta_2 - \delta_1 > w \tag{28}$$

then $\beta_w^+(a) \subset D_\gamma^2 \backslash D_\gamma^1$ and $\beta_w^-(a) \subset D_\gamma^2 \backslash D_\gamma^1$.

Proof: Let us assume without loss of generality that γ coincides with the y-axis. For any $p \in \mathbb{R}^2$ let $x(p)$ be the x-coordinate of p so that $d(p; \gamma) = |x(p)|$ (see Fig. 20). Let p_1^-, p_2^- be the end-points of $\beta_w^-(a)$ and p_1^+, p_2^+ the end-points of $\beta_w^+(a)$. Recall that

$$p_i^\pm = p_i \pm w u_\perp(p_i), \quad i = 1, 2$$

where $u_\perp(p_i)$ is the unit vector perpendicular to v_{p_i}. Then,

$$x(p_i^\pm) = x(p_i) \pm w \cos\theta_i, \quad i = 1, 2$$

where θ_i is the angle between v_{p_i} and γ. The alignment condition requires $\theta_i < \Theta_1$ so that

$$|x(p_i^\pm) - x(p_i)| > w \cos\Theta_1, \quad i = 1, 2$$

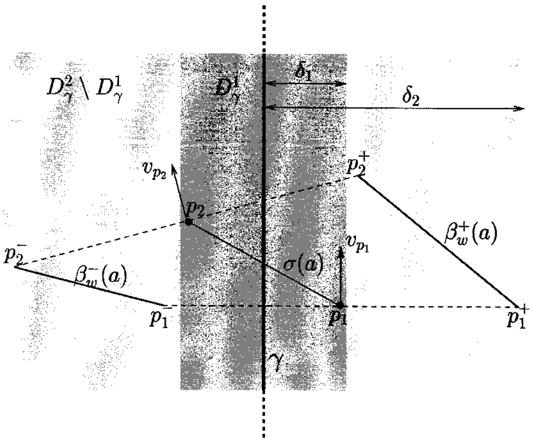


Figure 20. Proof of Lemma 1.

Therefore, since $|x(p_i)| < \delta_1$ and by using (27),

$$|x(p_i^{\pm})| > w\cos\Theta_1 - |x(p_i)| > w\cos\Theta_1 - \delta_1 > \delta_1$$

Also, by using (28),

$$|x(p_i^{\pm})| \le |x(p_i)| + w < \delta_1 + w < \delta_2$$

Thus $p_i^{\pm} \in D^2_\gamma \setminus D^1_\gamma$ and the result follows from the convexity of $D^2_\gamma \setminus D^1_\gamma$. □

Proposition 4. *Let γ be an unbounded straight line. Let (P, V, A) satisfy the decay and alignment conditions on γ. Suppose Eqs.* (27) *and* (28) *hold and $\delta_2 - \delta_1 > l^{\max}(A)$. Let $p \in P$. Then*

$$p \in D^0_\gamma \Rightarrow p \notin \tilde{P}_w(V, A)$$

Proof: Let A_p be the set of all arcs in A which have p as one of its two end-points. Let $a = (p, \bar{p}) \in A_p$ and $a' = (q_1, q_2) \in A$. From Eqs. (8)–(11), we must prove that either a and a' are compatible or $\phi(p) > \phi_0$, where

$$\phi_0 = \min\{p, \bar{p}, q_1, q_2\}$$

Since $\|p - \bar{p}\| \le l^{\max}(A) < \delta_2 - \delta_1$ and $d(p; \gamma) < \delta_1$ we have

$$d(\bar{p}; \gamma) < d(p; \gamma) + \|p - \bar{p}\| < \delta_1 + (\delta_2 - \delta_1) < \delta_2,$$

that is $\bar{p} \in D^2_\gamma$. If $\bar{p} \in D^2_\gamma \setminus D^1_\gamma$ then $\phi(p) > \phi_0$, because $p \in D^0_\gamma$ and $\phi(p) > \phi(\bar{p})$ from the decay condition (2). Let us assume then that $\bar{p} \in D^1_\gamma$. This implies that $\sigma(a) \subset D^1_\gamma$ and, from Lemma 1,

$$\beta_w^+(a) \cup \beta_w^-(a) \subset D^2_\gamma \setminus D^1_\gamma \tag{29}$$

The geometric relationship between $a' = (q_1, q_2)$ and γ can fall into one of three possible cases (see Fig. 21):

(i) At least one of q_1, q_2 belongs to $D^2_\gamma \setminus D^1_\gamma$.
(ii) $q_1, q_2 \in D^1_\gamma$.
(iii) $q_1, q_2 \in \mathbb{R}^2 \setminus D^2_\gamma$.

The case where one of q_1, q_2 is in D^1_γ and the other is in $\mathbb{R}^2 \setminus D^2_\gamma$ cannot occur because $\|q_1 - q_2\| \le l^{\max}(A) < \delta_2 - \delta_1$.

Case (i) From the decay condition it follows $\phi(p) > \phi(q_1)$ or $\phi(p) > \phi(q_2)$ and therefore $\phi(p) > \phi_0$.

Case (ii) Notice that $\sigma(a') = \sigma(q_1, q_2) \subset D^1_\gamma$. From this and from Eq. (29) we have

$$\sigma(a') \cap ((\beta_w^-(a) \cup \beta_w^+(a)) = \emptyset.$$

Similarly, from $\sigma(a) \subset D^1_\gamma$ and from Lemma 1 applied to a',

$$\sigma(a) \cap ((\beta_w^-(a') \cup \beta_w^+(a')) = \emptyset.$$

Hence $\psi_w(a, a') = \texttt{false}$ and (a, a') is a compatible pair.

Case (iii) Since $\delta_2 - \delta_1 > w$, it follows that

$$D^1_\gamma \cap ((\beta_w^-(a') \cup \beta_w^+(a')) = \emptyset$$

and therefore, from $\sigma(a) \subset D^1_\gamma$,

$$\sigma(a) \cap ((\beta_w^-(a') \cup \beta_w^+(a')) = \emptyset.$$

Since $\sigma(a') = \sigma(q_1, q_2) \subset \mathbb{R}^2 \setminus D^2_\gamma$, from (29) we have

$$\sigma(a') \cap ((\beta_w^-(a) \cup \beta_w^+(a)) = \emptyset$$

and therefore $\psi_w(a, a') = \texttt{false}$. □

Proof of Theorem 2: Let $\gamma \in \Gamma$ and let us assume that γ is an unbounded straight line (The proof is given for this case only).[2] Since (P, V, A) is a projection of Γ, $\gamma \in \Gamma$ satisfies the covering, decay and alignment

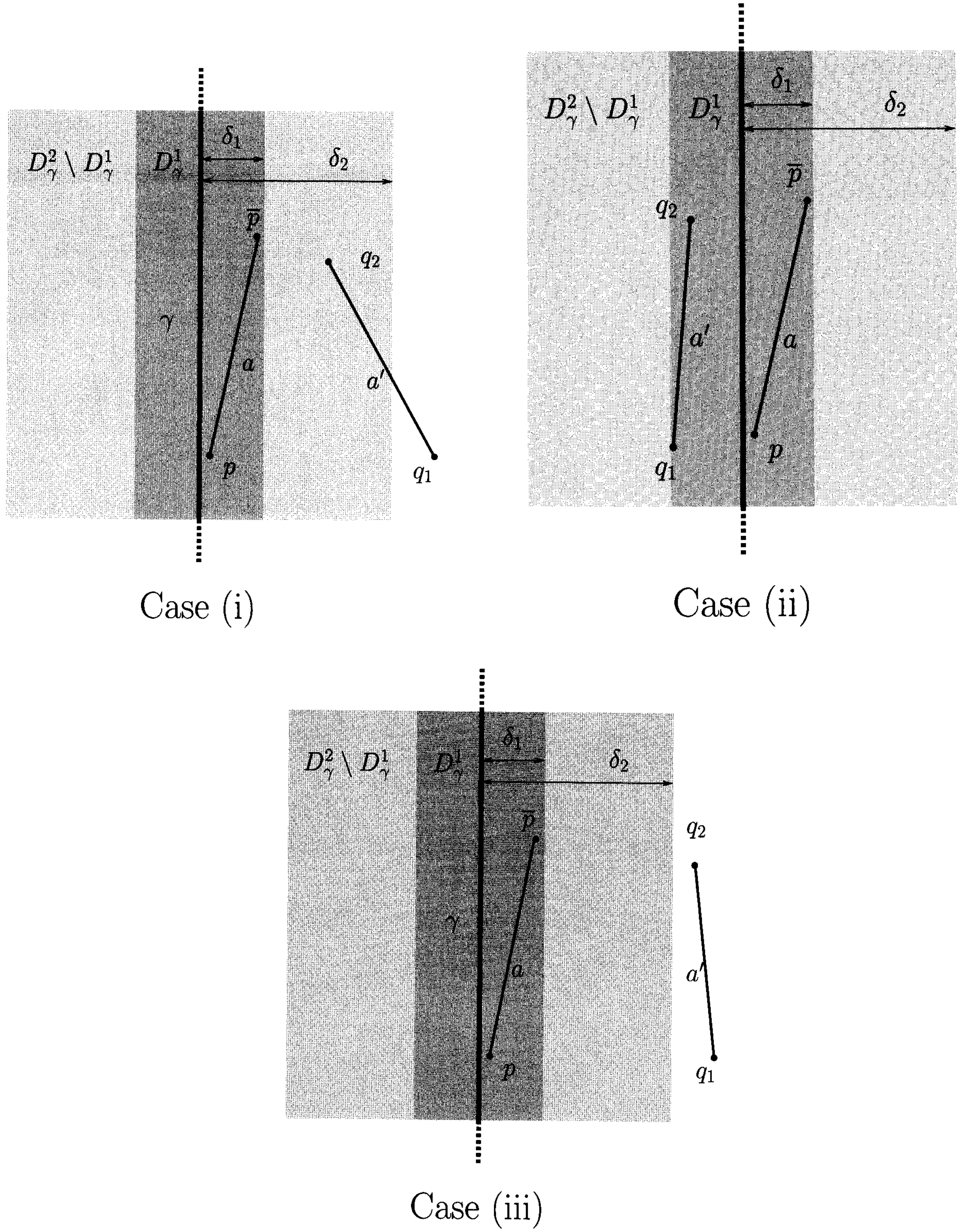


Figure 21. Proof of Proposition 21. The three possible cases for a'.

conditions. From the covering condition, there exists a path π in A such that $d(\gamma; \sigma(\pi)) < \delta_0$. Notice that the vertices of π belong to D^0_γ so that, from Proposition 4, they do not belong to $\tilde{P}_w(V, A)$. Therefore, π is also a path in $S^w(V, A) = A - \tilde{P}_w(V, A)$. □

10.3. Proofs of Theorems 3 and 4

Definition 6. Let A, $\hat{A}$ be sets of arcs and let $\hat{P}$ be the set of vertices of $\hat{A}$. The set $\hat{A}$ is said to be $\sigma_w^\perp$-*connected with respect to* A if

$$\sigma(p_1, p_2) \cap \dot{\sigma}_w^\perp(p) \neq \emptyset \Rightarrow (p_1, p) \in \hat{A}\ (p, p_2) \in \hat{A} \tag{30}$$

for every $(p_1, p_2) \in \hat{A} \cap A$ and every $p \in \hat{P}$.

Proposition 5. *The set $\bar{A}$ constructed in Section 5 is $\sigma_w^\perp$-connected with respect to A.*

Proof: The result follows directly from the definition of $\bar{A}$. □

The following Lemma ensures that a graph remains $\sigma_w^\perp$-connected if an arbitrary set of vertices is suppressed.

Lemma 2. *Let A, $\hat{A}$ be sets of arcs whose vertices belong to P. Let $P' \subset P$. If $\hat{A}$ is $\sigma_w^\perp$-connected with respect to A then $\hat{A} - P'$ is also $\sigma_w^\perp$-connected with respect to A.*

Proof: Let p be a vertex in $\hat{A} - P'$ and (p_1, p_2) an arc in $(\hat{A} - P') \cap A$ such that $\sigma(p_1, p_2) \cap \dot{\sigma}_w^\perp(p) \neq \emptyset$. We have to prove that $(p_1, p) \in \hat{A} - P'$ and $(p, p_2) \in \hat{A} - P'$. Notice that since p is a vertex in $\hat{A} - P'$ it is also a vertex in $\hat{A}$. Also, from $(p_1, p_2) \in (\hat{A} - P') \cap A$ we have $(p_1, p_2) \in A$. Therefore, since $\hat{A}$ is $\sigma_w^\perp$-connected w.r.t. A,

$$(p_1, p) \in \hat{A}, \quad (p, p_2) \in \hat{A} \tag{31}$$

Since p is a vertex in $\hat{A} - P'$ we have $p \notin P'$. Also, from $(p_1, p_2) \in (\hat{A} - P') \cap A$ we have $p_1, p_2 \notin P'$. Hence, from (31) it follows that $(p_1, p) \in \hat{A} - P'$ and $(p, p_2) \in \hat{A} - P'$. □

Proposition 6. *For any vector graph (P, V, A), $S^w(V, \bar{A})$ is $\sigma_w^\perp$-connected with respect to A.*

Proof: Notice that $S^w(V, \bar{A}) = \bar{A} - \tilde{P}_w(V, \bar{A})$. Therefore, since $\bar{A}$ is $\sigma_w^\perp$-connected w.r.t. A, the result follows from Lemma 2. □

Proposition 7. *Let $\hat{A}$ be $\sigma_w^\perp$-connected with respect to A, and let $\pi = (p_1, \ldots, p_n)$ be a maximal path in $\hat{A}$. Then, for every $(q_1, q_2) \in \hat{A} \cap A$,*

$$\sigma(q_1, q_2) \cap \dot{\sigma}_w^\perp(p_1) = \sigma(q_1, q_2) \cap \dot{\sigma}_w^\perp(p_n) = \emptyset$$

Proof: For the purpose of contradiction, let $(q_1, q_2) \in \hat{A} \cap A$ be such that

$$\sigma(q_1, q_2) \cap \dot{\sigma}_w^\perp(q) \neq \emptyset \tag{32}$$

where q is either p_1 or p_n. Since $\hat{A}$ is $\sigma_w^\perp$-connected, we have that (q, q_2) and (q_1, q) belong to $\hat{A}$ and therefore q has at least one out-arc and one in-arc. This contradicts the fact that π is maximal in $\hat{A}$. □

Proposition 8. *Let $\hat{A}$ be $\sigma_w^\perp$-connected with respect to A and arc-compatible.*

- *For every maximal path $\pi = (p_1, \ldots, p_n)$ in $\hat{A}$ and every path π' in $\hat{A} \cap A$,*

$$\beta_w(\pi) \cap \sigma(\pi') \subset \{p_1, p_n\} \tag{33}$$

- *$\hat{A}$ is strongly w-stable with respect to A with attraction basins $U_w(\pi)$.*

Proof: Let $\pi = (p_1, \ldots, p_n)$ be a maximal path in $\hat{A}$ and π' a path in $\hat{A} \cap A$. Since $\hat{A}$ is arc-compatible we have from Proposition 2,

$$\sigma(\pi') \cap \beta_w^-(\pi) = \sigma(\pi') \cap \beta_w^+(\pi) = \emptyset$$

Since $\hat{A}$ is $\sigma_w^\perp$-connected and π is maximal in $\hat{A}$ we have from Proposition 7

$$\sigma(\pi') \cap \dot{\sigma}_w^\perp(p_1) = \sigma(\pi') \cap \dot{\sigma}_w^\perp(p_n) = \emptyset$$

Thus, since $\sigma_w^\perp(p) = \dot{\sigma}_w^\perp(p) \cup \{p\}$,

$$\begin{aligned}\sigma(\pi') \cap \beta_w(\pi) = \sigma(\pi') \cap (\beta_w^-(\pi) \cup \sigma_w^\perp(p_n) \\ \cup \beta_w^+(\pi) \cup \sigma_w^\perp(p_1)) \subset \{p_1, p_n\}\end{aligned} \tag{34}$$

which proves the first part. Let now π' be a path in $\hat{A} \cap A$ such that $\sigma(\pi') \cap U_w(\pi) \neq \emptyset$. To prove that $\hat{A}$ is strongly w-stable, one has to show that $\sigma(\pi') \subset \bar{U}_w(\pi)$. From the assumptions (16)–(18) and from $A^{\text{in}}_{p_1} = A^{\text{out}}_{p_n} = \emptyset$ we have that $\sigma(\pi')$ can not exit $\bar{U}_w(\pi)$ through the points p_1, p_n. This, together with (34), yields the result. □

Proof of Theorem 3: From Proposition 1, $S^w(V, \bar{A})$ is arc-compatible because it is the output of the stabilization algorithm on the vector graph $(P, V, \bar{A})$. Moreover, $S^w(V, \bar{A})$ is $\sigma_w^{\perp}$-connected from Proposition 6. The result then follows from Proposition 8. □

Proof of Theorem 4: The proof is similar the that of Theorem 2. Let $\gamma \in \Gamma$ and, as in the proof of Theorem 2, let us assume that γ is an unbounded straight line. Since (P, V, A) is a projection of Γ, γ satisfies the covering, decay and alignment conditions. From the covering condition, there exists a path π in A such that $d(\gamma; \sigma(\pi)) < \delta_0$. Notice that the length of the segments added to A to construct $\bar{A}$ is at most $w + l^{\max}(A)$. Therefore, $l^{\max}(\bar{A}) \leq w + l^{\max}(A)$. Since the vertices of π belong to D^0_γ, by using Proposition 4 with A replaced by $\bar{A}$, we have that none of the vertices of π belongs to $\tilde{P}_w(V, \bar{A})$. Hence, since $S^w(V, \bar{A}) = \bar{A} - \tilde{P}_w(V, A)$ and $A \subset \bar{A}$, π is also a path in $S^w(V, \bar{A}) \cap A$. □

10.4. Proof of Theorem 5

Proposition 9. *A_j is strongly w-stable with attraction basins $U_w(\pi)$.*

Proof: Notice that

$$A_j = S^w(V, \bar{A}) - \left(\bigcup_{k<j} Q_k \right)$$

Thus, from Lemma 2 and Proposition 6, we have that A_j is $\sigma_w^{\perp}$-connected. Also, A_j is arc-compatible because it is a subset of $S^w(V, \bar{A})$ which is arc-compatible. The result then follows from Proposition 8. □

Proposition 10. *Let π be a path in $S^w(V, \bar{A}) \cap A$. Then there exists a π_j such that $d(\sigma(\pi); \sigma(\pi_j)) \leq w$.*

Proof: As a first step we construct a partition of $A_1 = S^w(V, \bar{A})$ into N sets B_j, $j = 1, \ldots, N$. These sets are given by:

$$B_j = \{(p_1, p_2) \in A_j : p_1 \in U_j \vee p_2 \in U_j\} \tag{35}$$

where $U_j = U_w(\pi_j)$. Notice that $B_j \subset A_j$. The following must be proven:

$$\bigcup_{j=1}^{N} B_j = S^w(V, \bar{A}) = A_1 \tag{36}$$

$$B_j \cap B_k = \emptyset, \quad k > j \tag{37}$$

From the recursive definition of A_j (lines 6 and 7 of Table 2) we have

$$\begin{aligned} A_{j+1} &= A_j - Q_j \\ &= A_j \backslash \{(p_1, p_2) \in A_j : p_1 \in U_j \vee p_2 \in U_j\} \end{aligned}$$

and therefore

$$A_{j+1} = A_j \backslash B_j \tag{38}$$

From this it follows that $B_j \cap A_{j+1} = \emptyset$. Thus, if $k > j$ we have $B_j \cap B_k = \emptyset$ because $B_k \subset A_k \subset A_{j+1}$. This proves (37). Notice that by iterating (38) one obtains:

$$A_j = A_1 \Bigg\backslash \bigcup_{k<j} B_k \tag{39}$$

Let N be the number of steps done by the procedure before terminating. That is, from line 3 of Table 2,

$$A_{N+1} = \emptyset$$

By using (39) one gets

$$A_{N+1} = A_1 \Bigg\backslash \bigcup_{k \leq N} B_k = \emptyset$$

from which (36) follows. Notice that from (39) and (36) we have

$$A_j = A_1 \Bigg\backslash \bigcup_{k<j} B_k = \bigcup_{k \geq j} B_k \tag{40}$$

Now let k be the smallest j such that B_j contains at least one arc of the path π:

$$\text{arcs}(\pi) \cap B_j = \emptyset, \quad j < k \tag{41}$$

$$\text{arcs}(\pi) \cap B_k \neq \emptyset. \tag{42}$$

From (36), (41) and (40) we have

$$\begin{aligned}\mathrm{arcs}(\pi) &= \bigcup_{1\le j\le N} (\mathrm{arcs}(\pi)\cap B_j)\\ &= \bigcup_{j\ge k}(\mathrm{arcs}(\pi)\cap B_j) \subset \bigcup_{j\ge k} B_j = A_k\end{aligned}$$

Thus, since π is path in $A_1 \cap A$ by assumption, π is a path in $A_k \cap A$. From $\mathrm{arcs}(\pi) \cap B_k \neq \emptyset$ and from (35) we have $\sigma(\pi) \cap U_k \neq \emptyset$. Then, since (from Proposition 9) A_k is strongly w-stable, we have $\sigma(\pi) \subset \bar{U}_k \subset \bar{N}_w(\pi_k)$. From this it follows that $d(\sigma(\pi); \sigma(\pi_k)) \le w$. □

Proof of Theorem 5: Let $\gamma \in \Gamma$. From Theorem 4 we have that there exists a path π in $S^w(V, \bar{A}) \cap A$ such that $d(\gamma; \sigma(\pi)) < \delta_0$. From Proposition 10 there exists a path π_j such that $d(\sigma(\pi); \sigma(\pi_j)) \le w$. Then, by the triangular inequality of the asymmetric Hausdorff distance one gets $d(\gamma; \sigma(\pi_j)) < w + \delta_0$. □

Acknowledgments

Research supported by US Army grant DAAL03-92-G-0115, Center for Intelligent Control Systems and US Army grant DAAH04-95-1-0494, Center for Imaging Science (Washington University).

Notes

1. This has to be done for *all* the points p, not just the end-points of A. In fact, new end-points can be created by the stabilization procedure, which will be applied to $\bar{A}$.
2. The proof for finite curves requires a somewhat complicated generalization of the decay and alignment conditions to constrain the vector field in the vicinity of the curve end-points. The generalization to curves with bounded curvature is trivial, given the arbitrariness of the functions $\epsilon_1(\kappa)$ and $\epsilon_2(\kappa)$.

References

Bienenstock, E. and Geman, S. 1995. Compositionality in neural systems. In *The Handbook of Brain Theory and Neural Networks*, M.A. Arbib (Ed.), MIT Press.

Canny, J. 1986. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–698.

Casadei, S. 1995. Model based detection of curves in images. Technical Report LIDS-TH-2306, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology.

Casadei, S. and Mitter, S.K. 1996a. A hierarchical approach to high resolution edge contour reconstruction. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*.

Casadei, S. and Mitter, S.K. 1996b. Hierarchical curve reconstruction. Part 1: Bifurcation analysis and recovery of smooth curves. In *Proc. of the Fourth European Conf. on Computer Vision*.

Caselles, V., Kimmel, R., and Sapiro, G. 1995. Geodesic active contours. In *Proc. of the Int. Conf. of Computer Vision*, pp. 694–699.

Cohen, L. and Kimmel, R. 1996. Global minimum for active contour models: A minimum path approach. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 666–673.

David, C. and Zucker, S. 1990. Potentials, valleys, and dynamic global coverings. *International Journal of Computer Vision*, 5(3):219–238.

Deriche, R. and Blaszka, T. 1993. Recovering and characterizing image features using an efficient model based approach. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*.

Dolan, J. and Riseman, E. 1992. Computing curvilinear structure by token-based grouping. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*.

Elder, J. and Zucker, S. 1996a. Local scale control for edge detection and blur estimation. In *Proc. of the Fourth European Conf. on Computer Vision*, pp. II:57–69.

Elder, J.H. and Zucker, S.W. 1996b. Computing contour closure. In *Proc. of the Fourth European Conf. on Computer Vision*, pp. 399–411.

Geiger, D. and Kumaran, K. 1996. Visual organization of illusory surfaces. In *Proc. of the Fourth European Conf. on Computer Vision*, pp. I:413–424.

Geiger, D., Kumaran, K., and Parida, L. 1996. Visual organization for figure/ground separation. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 155–160.

Geman, S. and Geman, D. 1984. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741.

Hancock, E. and Kittler, J. 1990. Edge-labeling using dictionary-based relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:165–181.

Haralick, R. 1984. Digital step edges from zero crossing of second directional derivatives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:58–68.

Iverson, L. and Zucker, S. 1995. Logical/linear operators for image curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(10):982–996.

Kanizsa, G. 1979. *Organization in Vision: Essays on Gestalt Perception*. Praeger: NY.

Kass, M., Witkin, A., and Terzopoulos, D. 1988. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331.

Kichenassamy, S., Kumar, A., Olver, P., Tannenbaum, A., and Yezzi, A. 1995. Gradient flows and geometric active contour models. In *Proc. of the Int. Conf. of Computer Vision*, pp. 810–815.

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

Marroquin, J., Mitter, S.K., and Poggio, T. 1987. Probabilistic solution of ill-posed problems in computational vision. *Journal of American Statistical Ass.*, 82(397):76–89.

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

Mumford, D. and Shah, J. 1989. Optimal approximations of piecewise smooth functions and associated variational problems. *Comm. in Pure and Appl. Math.*, 42:577–685.

Nayar, S., Baker, S., and Murase, H. 1996. Parametric feature detection. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 471–477.

Nitzberg, M. and Mumford, D. 1990. The 2.1-d sketch. In *Proc. of the Third Int. Conf. of Computer Vision*, pp. 138–144.

Nitzberg, M., Mumford, D., and Shiota, T. 1993. *Filtering, segmentation, and depth*, vol. 662 of *Lecture Notes in Computer Science*. Springer-Verlag.

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

Perona, P. 1992. Steerable-scalable kernels for edge detection and junction analysis. In *Proc. of the Second European Conf. on Computer Vision*, pp. 3–18.

Perona, P. and Malik, J. 1990. Detecting and localizing edges composed of steps, peaks and roofs. In *Proc. of the Third Int. Conf. of Computer Vision*, Osaka, pp. 52–57.

Richardson, T.J. and Mitter, S.K. 1994. Approximation, computation, and distorsion in the variational formulation. In *Geometry-Driven Diffusion in Computer Vision*. B. Ter Haar Romeny (Ed.), Kluwer.

Rohr, K. 1992. Recognizing corners by fitting parametric models. *International Journal of Computer Vision*, 9(3).

Rosenthaler, L., Heitger, F., Kubler, O., and von der Heydt, R. 1992. Detection of general edges and keypoints. In *Proc. of the Second European Conf. on Computer Vision*.

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

Sha'ashua, A. and Ullman, S. 1988. Structural saliency: The detection of globally salient structures using a locally connected network. In *Proc. of the Second Int. Conf. of Computer Vision*, pp. 321–327.

Shah, J. 1996. A common framework for curve evolution, segmentation and anisotropic diffusion. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 136–142.

Subirana-Vilanova, J.B. and Sung, K.K. 1992. Perceptual organization without edges. In *Image Understanding Workshop*.

Williams, L. and Jacobs, D. 1995. Stochastic completion fields: A neural model of illusory contour shape and salience. In *Proc. of the Int. Conf. of Computer Vision*, pp. 408–415.

Zhu, S., Lee, T., and Yuille, A. 1995. Region competition: Unifying snakes, region growing, and bayes/mdl for multi-band image segmentation. Technical Report CICS-P-454, Center for Intelligent Control Systems.

Zucker, S.W., David, C., Dobbins, A., and Iverson, L. 1988. The organization of curve detection: Coarse tangent fields and fine spline coverings. In *Proc. of the Second Int. Conf. of Computer Vision*.