June 1995

LIDS-TH-2306

Research Supported By:

US Army Research Office
ARO DAAL03-92-G-0115
ARO DAAL03-86-K-0171

Robust Detection of Curves In Images

Stefano Casadei

June 1995

LIDS-TH-2306

Sponsor Acknowledgments

US Army Research Office
ARO DAAL03-92-G-0115
ARO DAAL03-86-K-0171

Robust Detection of Curves In Images

Stefano Casadei

This report is based on the unaltered thesis of Stefano Casadei submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy at the Massachusetts Institute of Technology in May 1995.

This research was conducted at the M.I.T. Laboratory for Information and Decision Systems with research support gratefully acknowledged by the above mentioned sponsor(s).

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA 02139, USA

Robust Detection of Curves In Images

by

Stefano Casadei

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1995

Author ....
Department of Electrical Engineering and Computer Science
May 26, 1995

Certified by ....
Sanjoy K. Mitter
Professor of Electrical Engineering
Thesis Supervisor

Accepted by ....
Frederic R. Morgenthaler
Chairman, Departmental Committee on Graduate Students

Robust Detection of Curves In Images

by

Stefano Casadei

Submitted to the Department of Electrical Engineering and Computer Science
on May 26, 1995, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

The problem of representing the edges in an image by means of a set of curves is considered. An important point of view here is that certain types of edges should not be recovered in the first stage of computation. More precisely, singular points such as junctions and corners, and edges where brigthness change is small should be recovered only at a later stage. Thus, we study the problem of inferring smooth curves to describe edges that are far from singularities and that contain brightness changes. A model of curvës with these properties is defined, both for the ideal and noisy case. An algorithm is proposed which employs an intermediate representation consisting of a set of curve fragment hypotheses, obtained by estimating the position and orientation of the edge at every point in the image. This algorithm is proven to detect all the curves which satisfy the worst-case noise model. Furthermore, the reconstruction error of each curve, measured by the Hausdorff distance, is guaranteed to vanish linearly when the upper bound on the deviation from the ideal model goes to zero. Some experimental results are presented.

Thesis Supervisor: Sanjoy K. Mitter
Title: Professor of Electrical Engineering

Acknowledgments

I'd like to express my deep gratitude to Sanjoy Mitter for his constant support and guidance during all these years. His way of solving problems, rigorous and open-minded at the same time, has had a great influence on my education as a researcher. Sanjoy's insight into problems is so profound that his remarks sometimes reveal their importance only after a great deal of meditation and research. My work has benefited a lot from his ideas and suggestions.

Many thanks to the other member of the thesis committee, Robert Gallager, Alan Willsky and John Tsitsiklis for their comments and suggestions which improved significantly the quality of the final manuscript. Special thanks to Robert Gallager, Irvin Schick and Sanjoy Mitter for carefully reading the draft and suggesting several improvements of the text.

I would also like to thank Luigi Radicati di Brozolo, and the other faculty of the Scuola Normale Superiore, for their role in my education in the exact sciences.

Many other people have contributed to this thesis with their ideas, comments, suggestions and criticism. Among them, Pietro Perona, Brian Subirana, Tom Richardson, Fabio Fagnani, Guillermo Sapiro and Stephan Mallat.

I thank Mitch Trott for all his help with the computer system. His wizardry with computers is really impressive.

Many thanks to all the administrative staff, especially Kathleen O' Sullivan and Sheila Hegarty, for their precious help in all aspects of the lab's daily life.

If some of the talks I gave recently were not a complete disaster, the credit goes to an anonymous person with a special gift for laying out great slides and staying awake until late at night.

Being at MIT gave me the opportunity to meet great people whose friendship means a lot to me. The nerdy duo of River Street, Mike and Seema, took part in all the wildest events (not so many) of my life in Boston and tried hard, but in vain, to correct my English pronunciation. Nicola provided a great deal of wise advice on all matters, as well as many funny stories and anecdotes. Together with Maruska, they offered their help when it was needed. The office neighbors, Ted, Mike, Venkatesh and Mitch shared with me many leisurely moments. Finally, it was really a fortune for me to meet Francesca, who soon became my dearest friend.

Thanks to my family, Natalia, Giampiero, and Franco, for their unconditional love and their contributions to my education.

This was work was in part supported by the US Army Research Office under contracts DAAL03-92-G-0115 and DAAL03-86-K-0171. This funding was greatly appreciated.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The edge detection problem

1.1.1 Importance of global information

Detecting and representing boundaries between objects in images is a fundamental problem in computer vision. A good description of object boundaries can be useful for other tasks such as object recognition and motion estimation.

The most important source of information for edge detection is that objects of the real world often appear in the image as uniform regions with different brightnesses. Thus, edges between objects usually correspond to points in the image where brightness changes abruptly.

Then, a simple way to detect edges is to estimate brightness variations at every point in the image and declare an edge-point where brightness variation is high or maximum. This procedure is *local*, meaning that each small area of the image is processed independently from the rest of the image and the decision taken at each point — whether or not the point belongs to an edge — depends only on the data in a small neighborhood around it.

It is well known that this purely local method gives rise to many false positive and false negative errors. In fact, it often occurs that an edge does not generate a change of brightness. This can be due to occlusion by another object, a shadow, or simply because the two objects have very similar colors. Conversely, brightness change can occur away from edges due to noise, texture or small surface imperfections.

To overcome these problems it is necessary to make the edge detection process more global. That is, to decide whether or not an edge exists at some point, information from other areas of the image has to be taken into account. The distance that this information has to travel might be very large, as the Kanisza triangle example illustrates (figure 1-1).

1.1.2 Edge representations

An important issue which affects the capability of exploiting global information depends on the representation of edges. The simplest representation is given by a

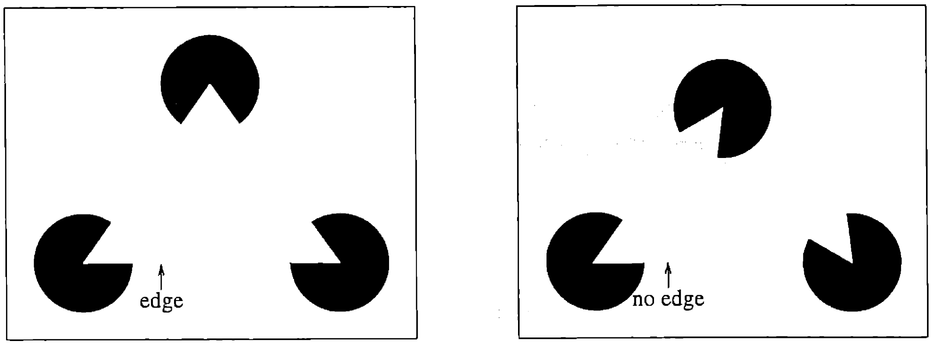


Figure 1-1: Kanisza triangle. Left: The human visual system interprets this image as a white triangle on top of three black circles. Edges are perceived even where there is no local clue to their presence. Right: The edge in the left figure indicated by the arrow "disappears" when the image is modified far away from this point. This shows that edge detection in the human visual system combines information from distant areas of the image.

collection of points. A slightly more sophisticated one is obtained by associating the estimated orientation of the edge to each edge-point. One has then a collection of vectors, or tiny curve fragments. Most local edge detectors produce a representation of this kind. Unfortunately, due to noise in the image, these curve fragments are usually disconnected from each other.

A more powerful representation is obtained by describing edges by means of a collection of curves. Clearly, a collection of curves rather than a collection of points gives also information about which edge-points belong to the same edge. Typically, a representation based on curves contains fewer primitive elements than one given by a collection of vectors. In fact, one single curve can embody the information contained in many vectors. A yet more high-level representation is given by a collection of regions, possibly ordered by occlusion. A representation of this type not only identifies the edges in the image — which are simply the boundaries of these regions — but also contains information about the surfaces of the imaged objects.

As one moves up the hierarchy of these geometric descriptors, from points to curves and regions, both the complexity and the spatial extent of these descriptors increases. Since high level descriptors cover large portions of the image they provide a suitable vehicle to carry and propagate global information. For instance, region descriptors are capable of representing occluded (and therefore "invisible") edges. The Kanisza triangle example illustrates this point. The brightness of this image can be encoded by a set of four overlapping regions — a white triangle on top of three black circles. This is a strong indication that the edges of the triangle exist even where there is no local evidence of their presence. This type of representation, referred to as *2.1 sketch*, was proposed in (Nitzberg and Mumford, 1990).

The fact that global information can be carried by these high level geometric descriptors suggests that edge detection can be done more effectively by using at some stage this kind of descriptors to represent edges. Unfortunately, moving up the hierarchy of descriptors is accompanied by a combinatorial explosion of the number of possible descriptors. In fact, whereas the number of points or pixel grows polynomially (quadratically) with the size of the image, the number of possible curves or regions grows exponentially. Thus, while exhaustive search in the space of all possible descriptors is feasible for detecting edge-points, it is not practicable for detecting curves or regions. For example, it is not possible to match all possible curves against the image until a good one is found. Other strategies are necessary to overcome the intrinsic combinatorial complexity brought about by including global information into the problem.

Many approaches and algorithms have been developed which try to incorporate global information into the edge detection process. They can be roughly classified into two classes: hierarchical methods and methods based on global cost functionals.

1.1.3 Hierarchical approaches

One way to exploit global information without causing combinatorial explosion of complexity is to introduce it *gradually.* Namely, the overall problem is broken into a hierarchy of subproblems which mirrors somehow the hierarchy of geometric descriptors discussed earlier. Each subproblem uses a little bit more global information than the ones at the next lower level of the hierarchy. Hopefully, the knowledge about the image gathered at any stage will help to guide the process in the following steps so that combinatorial explosion can be avoided.

A typical example of a hierarchical strategy is the following. First, a local edge detector is applied to the image. This step generates a collection of edge-points (or vectors). In the second step, these points are linked together to form sequences of points and a curve is interpolated through each of these sequences. At this stage, to limit the number of possible combinations of point groupings, only nearby points are linked together. In the third step, curves obtained in the previous step are grouped together so that occluded boundaries can be assembled. At this step, information about the structure of each curve can be used to guide grouping so that only a fraction of all the possible combinations are considered. For instance, grouping can be restricted initially to curves which are collinear or CO-circular. This makes possible the interaction of curve descriptors which are far away from each other. Finally, region descriptors can be associated with closed curves constructed in this way. The brightness inside these regions can be checked to verify that they are indeed uniform regions.

Clearly, information is not constrained to flow bottom-up in this hierarchy. It is also possible to have a top-down feedback which refines computation at a lower level by taking advantage of information gathered at higher level. For instance, one might apply the local edge detector again on the hypothetical line which connects two collinear segments found by the linking step. Probably, the better estimate of the edge position and orientation provided by the two collinear segments can enable the local edge detector to find edge-points that could not be detected beforehand.

More global information is progressively introduced into the process as bigger and more complex descriptors interact with each other. For computational reasons, each token can interact with only a small number of other tokens at every step. In other words, only a small number of all the possible groupings can be considered. This is why several steps are necessary to compute very global descriptors.

The architecture of procedures necessary to compute global descriptors can be very complex, due to the variety of phenomena which have to be taken into account. Brightness variations across edges can occur at different scales and can have many different shapes. Edges can exist with different degrees of smoothness and curvature. Critical points such as corners, junctions of several curves, can complicate the problem. Parts of the edges can be occluded or invisible. Noise with different amplitudes and characteristics can be present. Thus, a variety of procedures have to be integrated together to cope appropriately with each of these phenomena.

Designing and implementing this architecture is a daunting task and there is very little theory to help. It is not clear what this architecture should be, namely which aspects of the problem should be considered first, and which ones deferred until more information has been extracted from the image. Furthermore, to keep computational complexity under control, it is necessary to take irreversible decisions at some point and the timing of these decisions can be crucial.

As a consequence, research in the area has been quite fragmented. Many different architectures and algorithms have been developed independently of each other, and it quite impossible to integrate them together. Most of these approaches have a very heuristic flavor and are tailored to specific applications.

1.1.4 Global optimization methods

A second class of approaches is based on cost functionals which incorporate global information right from the beginning of the problem formulation. Different constraints and sources of information are combined into a unique model and a functional is defined which measures how good each possible image description is. Then, the problem is to find the description which optimizes this functional.

In a probabilistic setting, global information can be modeled by assigning high a-priori probabilities to those image descriptions which have certain global properties. The functional to be optimized is the conditional probability of the description given the observed image. Another popular approach models edges by elastic deformable curves ("snakes") embedded in a force field which pushes them toward areas where the gradient of brightness is high (Kass et al., 1988). In (Mumford and Shah, 1989) global information is represented by a term in the cost functional which penalizes the total length of the edges. A similar approach uses region primitives to model occlusion and is able to continue edges behind occluded regions (Nitzberg and Mumford, 1990).

A nice feature of these methods is that they offer a concise and mathematically precise formulation of the whole problem with a quite simple descriptive language. Also, many algorithms developed in these frameworks exploit global information explicitly from the beginning, whereas in the hierarchical approach global information comes into the picture only later. However, it is difficult to incorporate all the possible constraints and sources of information into a unique and unstructured model without making oversimplifying assumptions. As with all systems which have to operate in a complex and diverse environment, it is unlikely that a good system can be designed without decomposing the problem into subproblems.

Another difficulty is that including global information right from the beginning causes the optimization problem to be computationally very hard. In most cases one has to be satisfied with a local minimum. Or, it might be necessary to specify a good initial condition from outside (as in the methods based on deformable curves). Or, information about the initial condition can be provided by the hierarchical approach.

Moreover, in many cases it is not even clear that the optimal solution of the functional is a good description of the image. For instance, the formulations in (Mumford and Shah, 1989) can not represent junctions directly, but only in an asymptotic limit (Richardson, 1990).

In conclusion, global optimization approaches can be used successfully to solve parts of the edge detection problem but are not flexible enough to cope by themselves with the complexity and diversity of the overall problem.

1.2 Motivation and main ideas

The main points which motivate this thesis will be discussed briefly now. Then they will be explained in more details in the following sections.

Due to the complexity of the problem and the diversity of situations that can occur, a robust edge detector should rely on many models and procedures organized into some sort of network. Thus a framework is necessary to describe the role of each procedure and specify how the different components interact and communicate with each other. This framework has to provide a common language to characterize the input-output properties and performance of each procedure. Understanding what each component can do and can not do is important to figure out how procedures should be used together.

Unfortunately, many of the existing algorithms for edge detection, especially those which deal with linking and grouping, are based on quite heuristic rules and their input-output behaviors and performances are not very well understood. Moreover, when some theoretical analysis is done, this is based most of the time on probabilistic models. We claim that probabilistic evaluation of performance is not appropriate for constructing networks of procedures. In fact, knowing the probability of error does not say very much about what disturbances cause an error. Therefore, it is not clear how to design other procedures to compensate for these errors.

On the other hand, worst-case design and analysis *guarantee* certain levels of performance for entire collections of bounded disturbances. This knowledge allows the designer to focus on those disturbances which cause bad performance and develop compensating procedures. Thus, this type of framework is more suitable to deal with a problem which requires the integration of many procedures.

In this thesis, a new method is proposed to describe the input-output properties of procedures and assess their performance in the worst-case scenario. This method is based on the idea of *covering*. The output of a procedure is represented by a set of descriptors belonging to some *dictionary* which can be a family of points, curves, regions or other geometric objects. A procedure is then characterized by proving that its output is a covering of the set of all the possible descriptors of the input under the assumption that the norm of the disturbance is less than a threshold. In other words, the procedure has to produce an approximation for every descriptor in the dictionary which is "true" up to a disturbance smaller than the threshold. Note that this is a worst-case characterization of performance. This framework will be used to design a linking procedure for curve detection and to analyze its performance.

Because of the multi-process nature of edge detection, ambiguity in the computed representation is not necessarily bad. This is particularly true for those procedures which lie inside the network and whose output can be interpreted as a set of different hypothesis or possible interpretations of the input. Instead, most existing approaches are based on the search for the *optimal* description, disregarding the fact that there might not be enough information at a given stage to take an unambiguous decision. By resolving ambiguity at the right stage, edge detection can be done more reliably.

This point is illustrated by the relationship between local edge detection and linking. Most local edge detector approaches formulate the problem so that exactly *one* optimal estimate of edge-point position is generated in every small neighborhood containing an edge. Efforts are made to minimize multiple responses to the edge thus sacrificing other performance criteria. This is probably the right approach if local edge detection is regarded as a stand-alone process. However, if local edge detection has to be followed by edge linking then there is no need to determine edge-point position unambiguously at this stage. Indeed, we shall argue that some ambiguity in the edge-point representation can be useful and can make linking more robust. Most of this ambiguity is then eliminated by the linking procedure, which is in a better position to take hard decisions since it relies on more contextual information.

1.2.1 A network of procedures

The work on edge detection of the last thirty years shows how complex and difficult the problem is and how many different approaches can be followed to tackle it. Many algorithms have been developed which are successful with specific aspects of the problem and perform optimally on different classes of images. Many different types of geometric descriptors have been used to represent edges: points, vectors, curves and regions.

It is quite clear that there is no single approach or procedure which performs satisfactorily in every circumstance. It appears then that a robust edge detector has to rely on many different procedures designed to solve optimally each part of the problem in each possible situation that might occur. Also, different types of representations should be used. To use a popular metaphor, the edge detection process should look like a "society of procedures".

How do procedures cooperate and interact to achieve the goal ? Two procedures can be "connected" in two ways. One way (parallel connection) is when the two procedures deal with the same subproblem, for instance detection of T-junctions. These procedures might be specialized to detect different types of junctions or they might operate optimally with different types of noise. Thus they cooperate to achieve a more robust performance. For the connection to be successful one has to show that whenever one procedure fails then the other does not. Also, there has to be a way to interpret their combined output.

The other way (serial connection) is obtained when one procedure feeds its output to the other. In this case one has to show that the data received by the second routine is rich and precise enough for it to achieve its goal. For instance, a local edge detector can feed a collection of curve fragments to a linking routine. If for any reason a curve fragment is missing from the output of the local edge detector then the linking stage will generate a disconnected curve at that point.

To facilitate the task of integrating many procedures, it is essential to have an abstract description of the input-output characteristic of each routine in terms of a common language. To make an analogy, the design of very complex circuits is greatly simplified by modeling each component by means of a transfer function. This interface separates the details of the internal implementation of the procedure from its external functionality. The same idea is at the heart of object oriented programming.

1.2.2 Worst-case assessment of performance

How should a procedure be described ? There are basically two levels of description. One consists in specifying all the processing steps that lead from the input to the output (implementation level). The other way is to describe the goals of the procedure and evaluate how well they are achieved (functional level).

As argued in the previous section, a functional characterization of procedures which describes their input-output behaviors and their performance is very important for integrating many procedures together. A method to get this characterization is to construct a model of the input. This model assumes that the input consists of an ideal "signal" corrupted by some noise or disturbance. The goal of the procedure is then to isolate the signal from the noise and represent it in a convenient format. How well the signal is reconstructed determines how well the goal has been attained. For instance, in edge detection one can model the input image as a piecewise smooth map. The goal is then to detect these smooth components or the edges between them from a noisy image. To evaluate how well the goal is achieved one can compare the "true" edges in the model (which represent the "signal") with the edges generated by the procedure.

A common approach is to seek an optimal solution which minimizes the error or some other cost functional. However, knowing that the solution is optimal is not sufficient in many cases. In fact, it would be useful to have some information about the nature of the errors and what disturbances cause them. One approach would be to estimate the errors based on some probabilistic criterion. Another approach is to guarantee that for a given family of disturbances the errors are small (worst-case analysis). To carry out this type of analysis one has to identify all those situations which cause the procedure to generate large errors. For this reason, the information provided about the functionality of the procedure is potentially more powerful than estimating errors based on a probabilistic criterion. In fact, worst-case analysis also specifies when failures occur, namely what disturbances cause large errors. With this information available it is easier to find out ways to overcome these failures and define the role of the procedure in the overall architecture.

A disadvantage of worst-case analysis is that error bounds can be quite conservative, namely they can be larger than what they appear to be in practice. Also, the models proposed in this thesis do not make any distinction between model uncertainty and measurement errors. Both types of uncertainty are modeled as bounded deviations from an ideal model.

1.2.3 Divide-and-conquer strategy

A way to tackle difficult problems is to use a divide-and-conquer strategy to decompose the problem into more manageable problems. Thus the problem is solved by a network of procedures, one associated with each subproblem.

This strategy can be used to tackle some difficult issues in edge detection, namely curve singularities (corners and junctions) and gaps caused by low signal to noise ratio along some parts of the edge. To do this, we consider three subproblems. The first one consists in computing curves which do not contain any singularity and which have a large enough signal to noise ratio. Then, the other two problems are 1) recovering curve singularities and 2) bridging the gaps. Information obtained by the solution of the first problem can simplify significantly the two other problems. A large part of this thesis is devoted to solving the first of these subproblems.

1.2.4 Improving local edge detection

A weakness of many existing edge detection methods is that the decision concerning the position of edges is done on the base of very local information. The algorithms proposed in (Canny, 1986; Haralick, 1984; Perona and Malik, 1990) determine edge position by maximizing some local measure of "edginess" such as the magnitude of the brightness gradient.

Due to fluctuations of the gradient magnitude and orientation, maximization of local edginess can lead to curve disconnections and wiggly curves, as shown in figures 1-2 and 1-3. Furthermore, to limit multiple responses it is necessary to compromise other performance criteria. For instance, Canny showed that the mean localization error might have to increase by a factor of 3 to have a sufficiently low probability of multiple responses.

These shortcomings are due to considering local edge detection as a stand-alone process which has to make an unambiguous decision about edge localization. On the contrary, this is only the first step of the whole process. If some ambiguity is allowed in the representation produced by local edge detection then the following steps (e.g. linking) can be done more reliably. The algorithm proposed in this thesis performs maximization in a more sophisticated way to minimize disconnections and wiggling curves and achieve at the same time good localization of the edge. Roughly speaking, the method consists in suppressing those edge-points whose distance from the local maximum is in the appropriate range of values. Edge points very close to the local

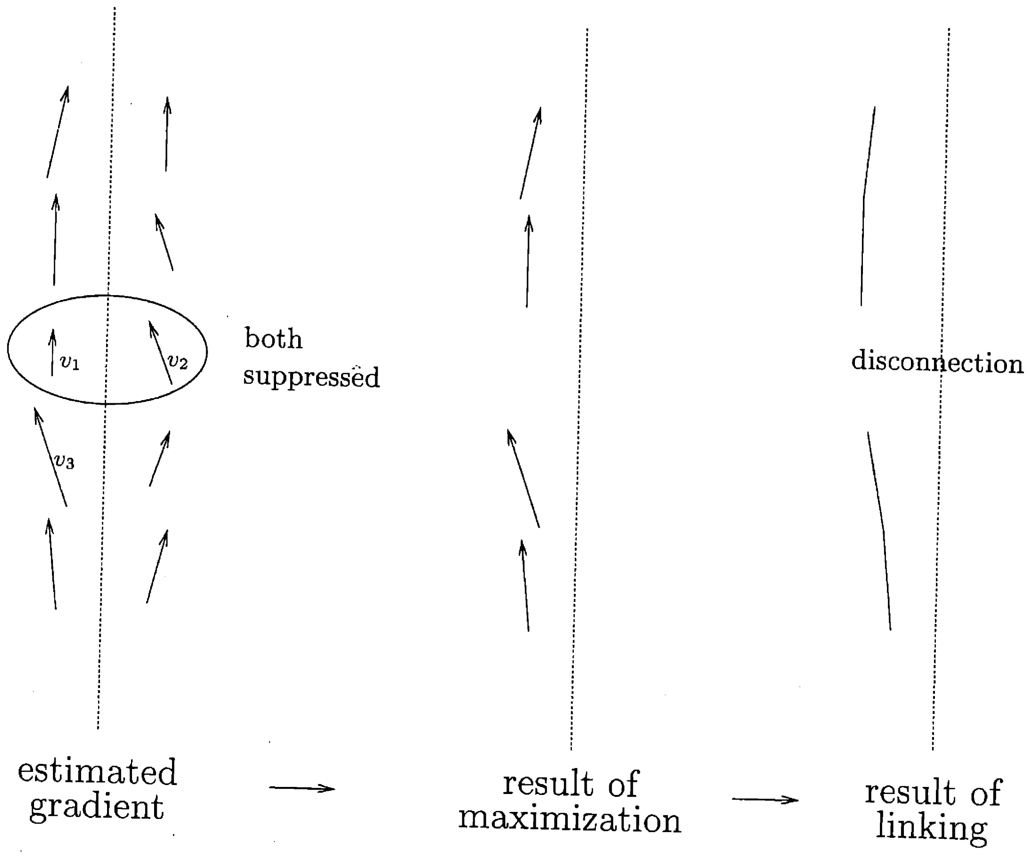


Figure 1-2: Straightforward maximization of edge strength in the direction perpendicular to the estimated orientation, as is usually done in most existing algorithms, can cause disconnections because of fluctuations of the vector field magnitude and orientation. In this example, the vector v_1 is suppressed by v_2 which is bigger. In turn, v_2 is suppressed by v_3. Thus both v_1 and v_2 are suppressed and a disconnection occurs when the remaining vectors are linked together.

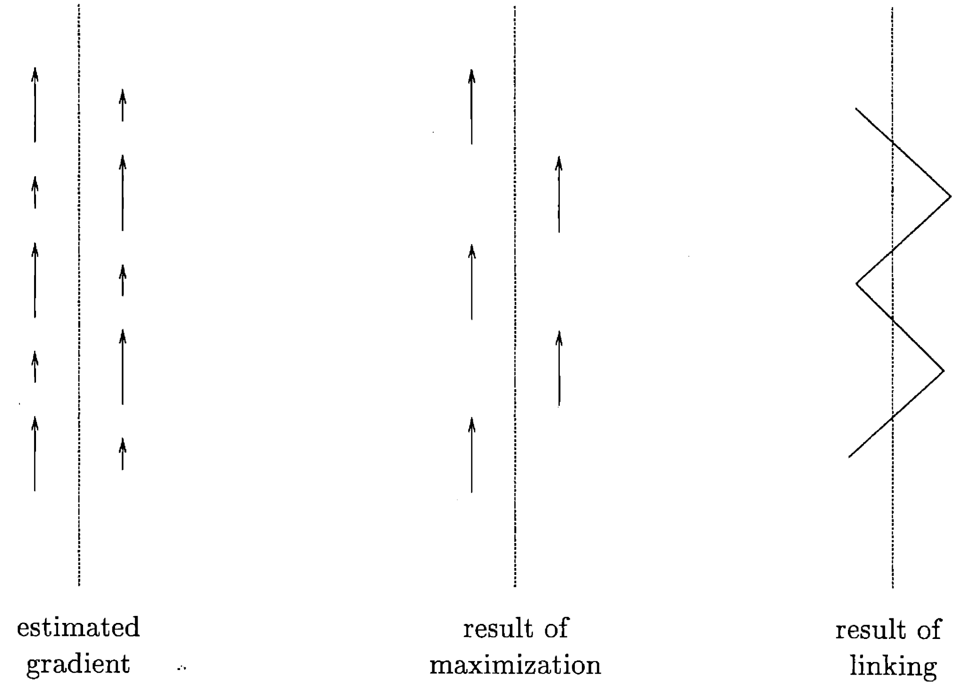


Figure 1-3: Similarly to figure 1-2, straightforward maximization can cause wiggly curves.

maximum are not suppressed, thus introducing some ambiguïty in edge localization. Linking then resolves this ambiguity using semi-local, more contextual information.

Similar arguments have been made in (Zucker et al., 1988; Parent and Zucker, 1989). These authors also argued that edge position should be determined only coarsely initially, when just local information is used. The method of non-maximum suppression they propose has some features in common with ours.

1.3 Characterization of procedures by means of dictionary coverings

1.3.1 Descriptors

What is the nature of the output of a procedure ? A possible point of view is that a procedure makes explicit some properties of the input. Thus its output is a list of properties. Another interpretation is that a procedure detects or isolates objects — e.g. geometric objects — which are implicitly contained in the input. In this case the output would be a set of objects.

This distinction between properties and objects is somewhat artificial and the output of a procedure can be regarded as a set of *descriptors* where a descriptor is some object which describes some property of the input. For instance, an edge produced by Canny's edge detector is a geometric object which describes the gradient of the input image inside some small region. This descriptor asserts that the magnitude of the gradient has a maximum at some particular point. An example of a more complex descriptor is the partially ordered set of regions (the 2.1 sketch) from which the brightness of the image in figure 1-1 can be completely reconstructed.

These two examples show that descriptors can be associated with information of very different kinds and at very different scales, from very local to very global. A local descriptor characterizes a very small portion of the input while a global one can code the whole input, as in the second example. Also, whereas a local descriptor is identified by few parameters a global one usually contains several parameters. Typically, an image can be represented either by many local descriptors (e.g. a set of edges) or by few global descriptors (a set of curves or the 2.1 sketch).

An image representation consisting of few global descriptors is preferable in many cases. To compute such a representation it might be necessary to use several intermediate representations which cover the whole spectrum from local to global. This requires several procedures which utilize different "dictionaries" of descriptors.

1.3.2 η-true descriptors

The output of a procedure is a set of descriptors. The set of all possible descriptors which can be generated by a procedure π is the *dictionary* of the procedure and is denoted by $\mathcal{D}_\pi$. Thus, if I denotes the input to the procedure then its output $\pi(I)$ is a subset of $\mathcal{D}_\pi$.

A basic requirement that the descriptors $\pi(I)$ should satisfy is that they represent "true" properties of I. A descriptor can be viewed as a statement about the input and, of course, it is desirable that only true statements be made about the input.

However, the clear-cut distinction between "true" and "false" descriptors is not suitable when noise is present and it is useful to allow a continuous spectrum of trueness. To do this, a confidence parameter η is introduced which measures how much a descriptor is "true" for a particular input. If $\eta = 0$ then the descriptor is "crisp true" or, in other words, it describes the input perfectly. The set of descriptors which have confidence parameter less than η for a given input I is denoted by $D_\eta(I)$.

As an example, let $\mathcal{D}_\pi$ be the set of all unit vectors in the plane and let $0 \leq \eta \leq 180^\circ$ represent the confidence in the orientation of the vectors[1]. Consider now the input image in figure 1-4. The set of all true descriptors $D_0(I)$ contains the vectors tangent to the circle. The η-true descriptors $D_\eta(I)$ are those vectors whose orientation differs less than η from the tangent. Notice that $\eta_1 \leq \eta_2 \implies D_{\eta_1}(I) \subseteq D_{\eta_2}(I)$. Note also that the set $D_\eta(I)$, $\eta > 0$ is typically very redundant. In our example, for every point on the boundary of the circle, $D_\eta(I)$ contains an infinite number of vectors.

[1] To be more rigorous, two confidence parameters should be used, one for vector orientation and one for the position of the vector in the plane.

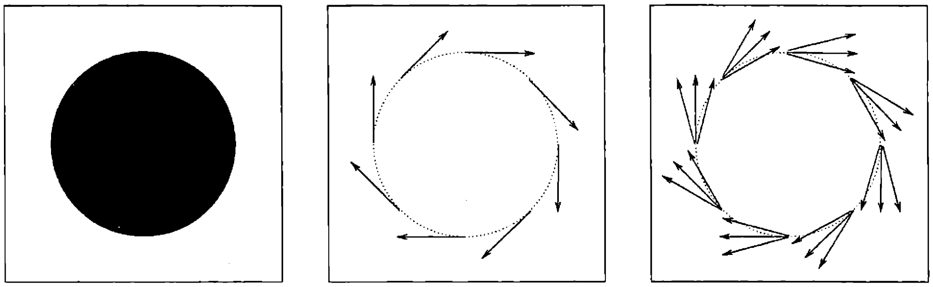


Figure 1-4: Left: An image. Center: some of the descriptors in $D_{0^\circ}(I)$. Right: some of the descriptors in $D_{15^\circ}(I)$.

1.3.3 Complete descriptions and coverings

The other fundamental property that a good description should have is completeness. In an ideal situation, such as in the example just described (figure 1-4), a complete description is one which contains all the true descriptors, namely all the elements in the dictionary which describe the data exactly. In the example, this would be the set $D_0(I)$ of all unit vectors tangent to the circle.

In a realistic situation where noise is present, there might be no descriptors which match the data exactly. In this case, a complete description or, more precisely, an approximately complete description, is one which contains an approximation to every η-true descriptor, where η is fixed a priori. That is, every η-true descriptor d should be represented in the output by some $d' \approx d$. Namely, a complete description $\pi(I)$ is one for which

$$\forall d \in D_\eta(I) \quad \exists d' \in \pi(I) \text{ such that } d' \approx d \tag{1.1}$$

Notice that, assuming that true-ness is a "smooth" property of descriptors, it follows from $d' \approx d$ and $d \in D_\eta(I)$ that d' is η'-true for some $\eta' \approx \eta$.

The notation $d' \approx d$ means that some distance function on the set of descriptors has been defined and that the distance between d' and d is less than some ρ, where ρ is sufficiently small. Thus, another way to rephrase (1.1) is to say that $\pi(I)$ is a covering of $D_\eta(I)$ with balls of radius ρ. Namely, $D_\eta(I)$ can be written as the union of the balls centered at the elements in $\pi(I)$ with radius ρ.

The definition of the distance function on the dictionary is a very critical issue. In fact this distance specifies how errors are measured and performance assessed. The constant ρ represents the approximation error of the computed description, that is, the distance between the η-true descriptors $D_\eta(I)$ and the computed ones $\pi(I)$.

In this framework, two parameters, η and ρ, are needed to measure performance. The former, η, is used to identify *which* objects (descriptors) are detected by the algorithm. The latter, ρ, specifies how accurately these objects are reconstructed. Namely, elements of $D_\eta(I)$ are always detected and reconstructed with an upper bound ρ on the recostruction error. For objects which do not belong to $D_\eta(I)$ nothing can be said.

1.3.4 Steps for characterizing a procedure

To formulate a description problem, or characterize a procedure π and assess its performance, the following steps are necessary.

- Specify the dictionary $\mathcal{D}_\pi$
- Identify the confidence parameter(s) η
- Define the sets $D_\eta(I) \subset \mathcal{D}_\pi$
- Give a distance function on $\mathcal{D}_\pi$
- Show that the computed description $\pi(I)$ covers $D_\eta(I)$

1.3.5 Connections between procedures

As discussed earlier, procedures can cooperate either by means of a serial or a parallel connection. A parallel connection occurs when two procedures π_1, π_2 address the same problem independently of each other. That is, the two procedures act on the same input and use the same output dictionary. The connection is implemented by unioning the outputs of the two procedures so that the output of the parallel connection is $\pi_1(I) \cup \pi_2(I)$. The joint system does a better job than either π_1 or π_2 alone if $\pi_1(I) \cup \pi_2(I)$ is a better covering of $D_\eta(I)$ than $\pi_1(I)$ or $\pi_2(I)$.

Two procedures are connected serially when the input to the second procedure is the output of the first one, $I_2 = \pi(I_1)$. A series of procedure can be useful to insert an intermediate representation between the input dictionary and the output dictionary when these two are too different to be "bridged" by a single procedure. For instance, consider the problem of computing a set of curves to represent the edges in an image. The input is the set of brightness values at every pixel and the output is a set of curves. An intermediate representation is given by a collection of vectors, or small curve fragments. To use this intermediate representation two procedures are necessary. The first computes a sampled vector field from the brightness image and the second one computes a set of curves from the sampled vector field.

1.4 Outline of thesis

The problem addressed in this thesis is that of computing curves from a brightness image by using a vector field as an intermediate representation. Chapter 2 describes an algorithm which generates a set of curves from a discrete vector field, namely from a finite set of vectors.

Chapter 3 analyzes the properties of this algorithm and specifies how to derive the algorithm parameters from the model parameters. The main result guarantees that all the curves which satisfy the noisy model are reconstructed without disconnections and gives an upper bound to the localization error. This bound vanishes linearly in the noise-free limit. The curves which satisfy the model, and therefore can be detected accurately, must have the following properties:

(i) The curve does not contain any singularity (such as sharp corners or T-junctions)

(ii) Curvature is low

(iii) The magnitude of the vector field must decay at some distance away from the curve

(iv) The vector field must be approximately aligned with the curve

If any of these assumptions is not satisfied at some point of the curve, then a discontinuity (i.e. a disconnection) occurs at that point, namely the curve is broken into two parts.

In chapter 4 the noisy model of a curve is defined in terms of the input brightness (gray-level) image. This model assumes that the brightness across the curve is given by a profile (e.g. a step profile) whose gradient is maximum on the curve. Noise is additive and bounded within each block of pixels of a given size. The main result of this chapter ensures that if noise is small compared to the brightness variation at each point of the curve and the conditions (i), (ii) above hold, then the curve is reconstructed without disconnections and with bounded localization error. If the assumptions are not satisfied at some point of the curve, then a discontinuity occurs at that point. The algorithm computes a vector for each block of pixels by fitting a local model to the brightness data. The resulting vector field is then fed to the algorithm described in chapter 2.

Chapter 2

Curve reconstruction from vector fields

2.1 Problem formulation

One way to describe the edges in a brightness (gray level) image is by means of a set of curves. These curves can be computed in two steps. First, a discrete[1] vector field Φ is obtained from the input image and then curves are computed out of Φ. This chapter describes an algorithm to carry out the second step. To formulate the problem and characterize the performance of the algorithm the approach discussed in section 1.3 will be used.

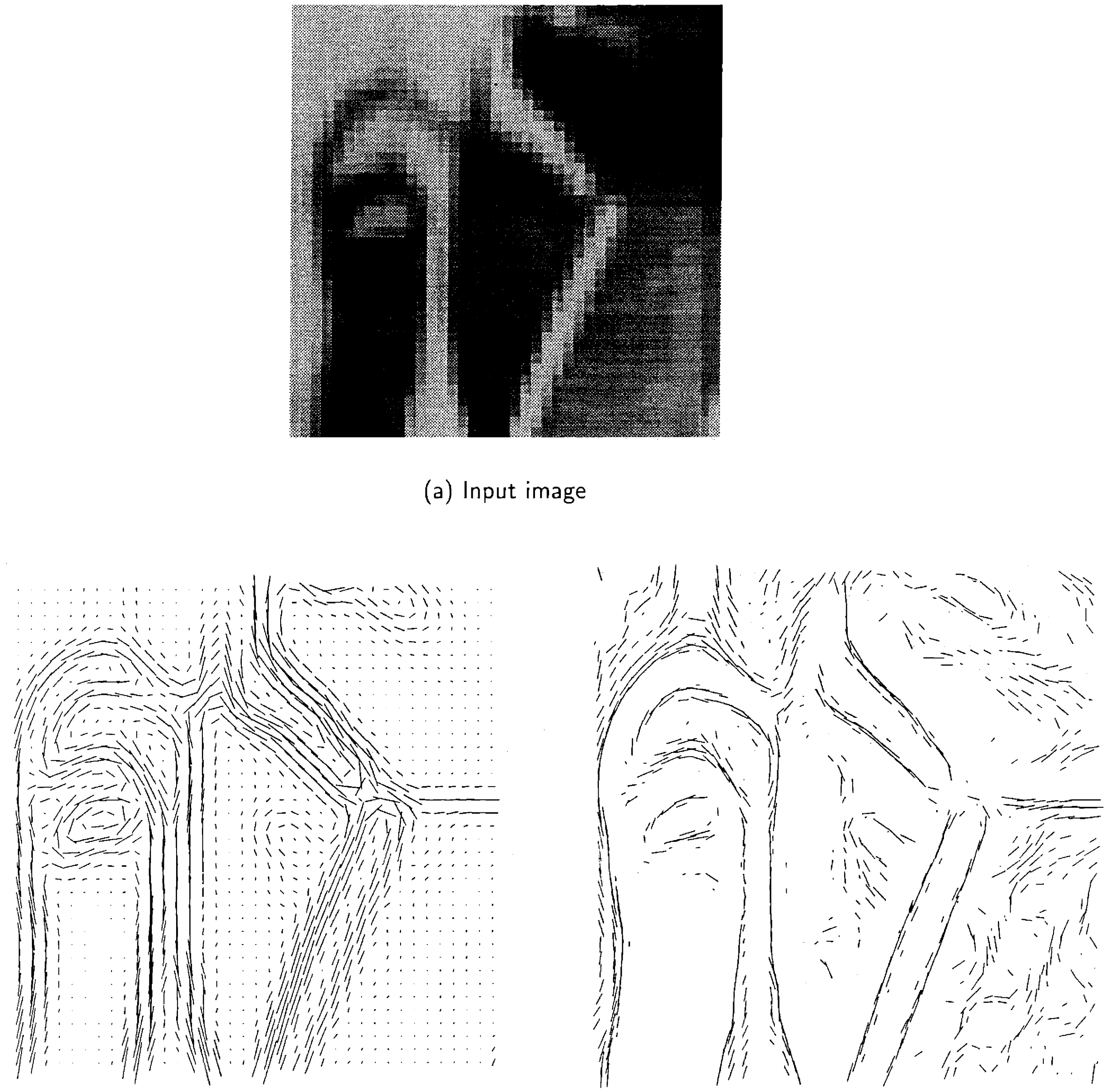
The input to the algorithm is a discrete vector field, namely a finite collection of vectors $\Phi = \{\phi(p) \ : \ p \in G\}$ where $G \subset \mathbb{R}^2$ is finite and $\phi(p)$ is a vector with vertex (base point) p. A simple way to obtain a vector field is to estimate the brightness gradient at every point p of a discrete grid G and then let $\phi(p)$ be the gradient at p rotated by $\pi/2$ (figure 2-1(b)). A more sophisticated way to construct the vector field is described in chapter 4 (figure 2-1(c)). In general, the magnitude or length of $\phi(p)$, denoted $|\phi(p)|$, measures the amplitude of the brightness variation at the point p. The orientation of $\phi(p)$, denoted $\theta(p)$, is a local estimate of the orientation of the edge.

2.1.1 Curve model

What curves should be computed from the given vector field Φ ? In other words, what kind of information must the field possess about a curve for the curve to be detectable ? First, let us consider the ideal case when there is no noise. Also, for simplicity let us assume for now that the vector field $\phi(p)$ is defined on the whole plane $\mathbb{R}^2$. Then, a natural characterization of a curve is given by two properties (see figure 2-2(a)):

(C1) The vector field on the curve is aligned with the curve tangent

[1] "Discrete" means here "containing a finite number of elements".

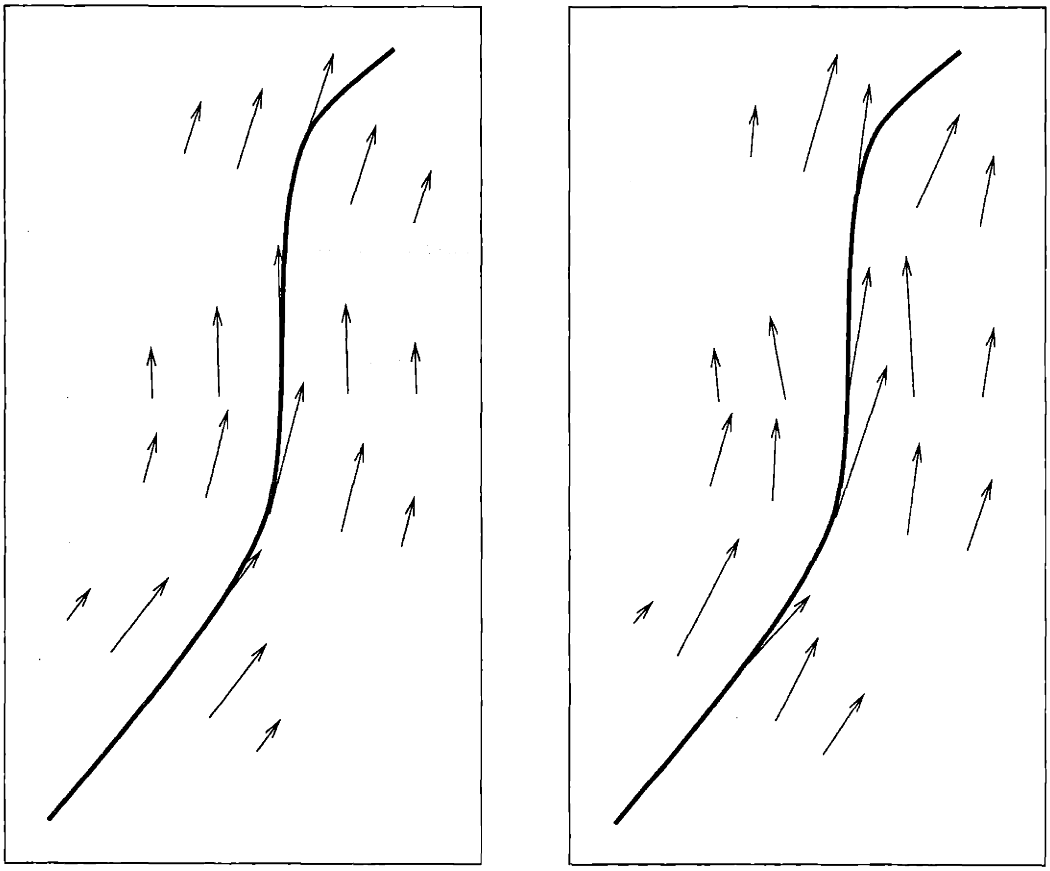


(a) Input image

(b) Uniformly sampled vector field obtained by rotating the estimated gradient by $\pi/2$.

(c) Vector field obtained with template fitting (see chapter 4)

Figure 2-1: Examples of vector fields.



(a) No noise

(b) With noise

Figure 2-2: The vector field in the neighborhood of a curve.

(C2) The magnitude of the vector field is locally maximum on the curve

More precisely, the second condition says that the restriction of ϕ to a segment perpendicular to the curve tangent has a local maximum on the intersection with the curve.

In the terminology of section 1.3, a curve with these properties is a "true" descriptor for the input vector field ϕ. The set of all these curves is denoted by $C_{0,0}(\phi)$.

With noise present, conditions (C1) and (C2) have to be relaxed. In fact, noise has two effects on the vector field ϕ. First, it displaces maxima away from the curve. Second, it distorts the orientation of the field so that it is no longer aligned with the curve tangent (see figure 2-2(b)). The deviation from the ideal case caused by the noise can be characterized by two parameters w, Θ. The parameter w is the distance from the curve at which the magnitude of the vector field is guaranteed to be less then the value attained on the curve (both values in the noisy case). Θ is an upper bound on the difference between the orientation of the curve tangent and the orientation of the field close to the curve.

To understand better the meaning of the parameter w, it should be noted that for typical boundaries in real images, the brightness variation occurs in a neighborhood of the boundary and then decays as one moves away from it.[2] Consequently, $|\phi(p)|$ has a high value in a neighborhood of the boundary and then decreases at some distance from it. Thus one can assume that at some distance w from the curve the strength $|\phi(p)|$ is lower than its value on the curve (or, more generally, the value at a point very close to the curve).

The set of all curves for which the vector field ϕ satisfies the above conditions is denoted by $C_{w,\Theta}(\phi)$ (see figure 2-3). Notice that the pair w, Θ corresponds to the parameter η described in section 1.3. A precise definition of the sets $C_{w,\Theta}(\phi)$ for the case when the vector field is discrete is given in section 3.1.

2.1.2 Goal of the algorithm

Given the discrete vector field Φ and the parameters w, Θ the algorithm computes a set of curves[3] $\hat{C}(\Phi) = \{\hat{\gamma}_1, \dots, \hat{\gamma}_n\}$ (n is not fixed a priori, nor is the length of each computed curve). These curves have properties very similar to the elements of $C_{w,\Theta}(\phi)$. Namely, they pass through points where the field has higher magnitude than regions away from the curve and is approximately aligned with the curve tangent.

Also (see section 3.2), the set of computed curves generates a covering of $C_{w,\Theta}(\phi)$ with balls of radius $\epsilon(w, \Theta) \simeq w/\cos\Theta$. That is, for every $\gamma \in C_{w,\Theta}(\phi)$ there exists $\hat{\gamma} \in \hat{C}(\Phi)$ such that

$$d(\gamma, \hat{\gamma}) \leq \epsilon(w, \Theta) \tag{2.1}$$

The parameter $\epsilon(w, \Theta)$ represents the localization error and vanishes when w and Θ both go to zero. The distance function d for which (2.1) holds is the asymmetric Hausdorff distance given by (see figure 2-4):

$$d(\gamma, \hat{\gamma}) = \max_{p\in\gamma} \min_{\hat{p}\in\hat{\gamma}} ||p - \hat{p}||$$

where γ denotes the set of points lying on the curve γ. In other words, $d(\gamma, \hat{\gamma})$ is the maximum distance of a point in γ from the set $\hat{\gamma}$.

Notice that $d(\gamma, \hat{\gamma})$ can be small even when $\hat{\gamma}$ is much longer than γ. Indeed, $d(\gamma, \hat{\gamma})$ is zero whenever γ is a subset of $\hat{\gamma}$. A distance function with this property makes sense here because the algorithm computes "maximal" curves and the set $C_{w,\Theta}(\phi)$ contains also non-maximal curves, namely sub-curves of the elements of $C_{w,\Theta}(\phi)$ (see figure 2-3). That the computed curves are not too long is guaranteed by the fact that the vector field has high magnitude on each point of these curves (relative to lateral regions away from the curve).

[2] The distance at which this decay takes place varies and depends mainly on the sharpness of brightness discontinuity (that is, how much this discontinuity is "blurred"); the amplitude of this discontinuity; and the amplitude of the noise.

[3] By "curve", in this thesis, we mean a subset of $\mathbb{R}^2$ which is the image of some smooth map $g : [0, T] \to \mathbb{R}^2$.

A curve γ which does not belong to $C_{w,\Theta}(\phi)$ because condition (C1) or (C2) on page 27 fails at some point $p \in \gamma$ will not be recovered as a unique, continuous curve. Instead, two curves will be generated which approximate each connected component of $\gamma \setminus \{p\}$. To recover γ in its entirety one can interpolate between all neighboring pairs of curve-ends and verify that the resulting curve satisfies (C1) and (C2) almost everywhere.

2.2 Brief description of the algorithm

The algorithm contains six steps. Step 1 consists of a non-maximum suppression procedure which computes the *dominant* points $P \subset G$ of the vector field $\Phi = \{\phi(p) \ : \ p \in G\}$. A point $p \in G$ is said to be dominant if

- the magnitude of the vector field at p is larger than its value in two lateral regions to be described.
- the orientation of the vector field in a neighborhood of p is approximately constant

Step 2 computes the connected components $P_1, \ldots, P_K$ of a graph with vertices in P. The arcs of these graph are the pairs $(p_1, p_2) \in P \times P$ such that $||p_1 - p_2||$ is less than a threshold a. If the parameters used in step 1 are chosen appropriately then each connected component has a thin elongated shape. The remaining steps of the algorithm are applied independently to each connected component P_k. A generic component P_k will be denoted henceforth as Q.

Step 3 assigns a longitudinal coordinate, or arc length parameter, to every every point in P_k. An important property of this coordinate map is that the distance between two points having the same coordinate is guaranteed to be small.

Step 4 computes a set T of triples of points in Q, namely $V_k \subset Q \times Q \times Q$. This set contains all the "good" discrete curve fragments (paths) of three elements. It has a natural directed-graph structure whose arcs are the pairs of triples which have two consecutive elements in common.

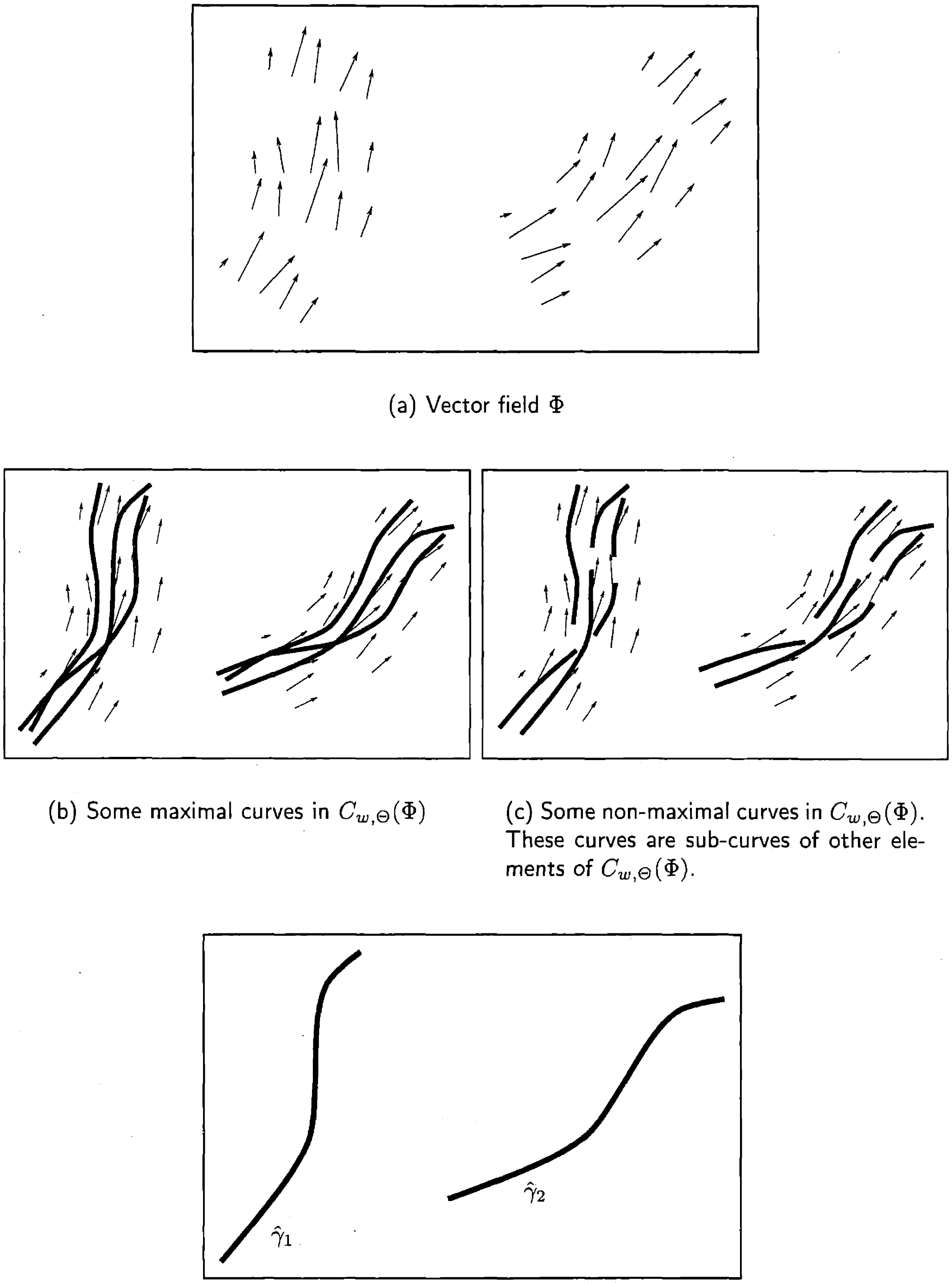
It should be noted that this step eliminates isolated vectors which survived non-maximum suppression. In fact, only vectors which belong to some triple of neighboring vectors are considered as possible curve fragments.

Step 5 decomposes T into maximal subgraphs containing a covering path, namely a path which connects the point with lowest longitudinal coordinate with the point with highest coordinate.

Finally, step 6 computes the path with the smallest turn in every subgraph. A polygonal continuous curve is associated with this path.

2.3 Non-maximum suppression

Since brightness variations are typically larger in the neighborhood of boundaries, only those points where brightness variation is high have to be considered to compute



(a) Vector field Φ

(b) Some maximal curves in $C_{w,\ominus}(\Phi)$

(c) Some non-maximal curves in $C_{w,\ominus}(\Phi)$. These curves are sub-curves of other elements of $C_{w,\ominus}(\Phi)$.

(d) Two-curve covering of $C_{w,\ominus}(\Phi)$

Figure 2-3: The set $C_{w,\ominus}(\Phi)$ and a covering of it.

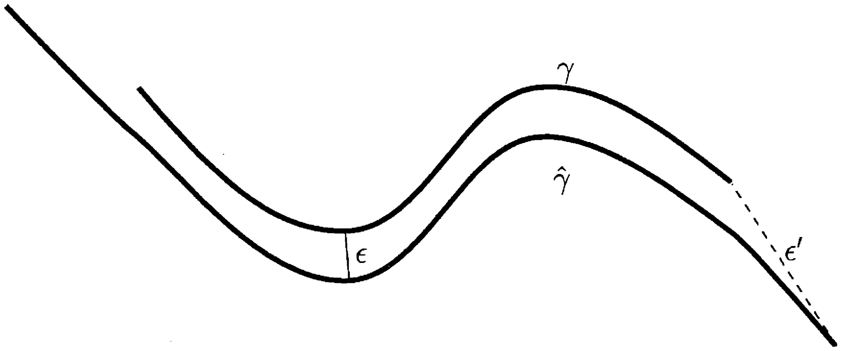


Figure 2-4: Two curves whose asymmetric Hausdorff distance $d(\gamma, \hat{\gamma})$ is ϵ. Notice that $d(\hat{\gamma}, \gamma) = \epsilon'$.

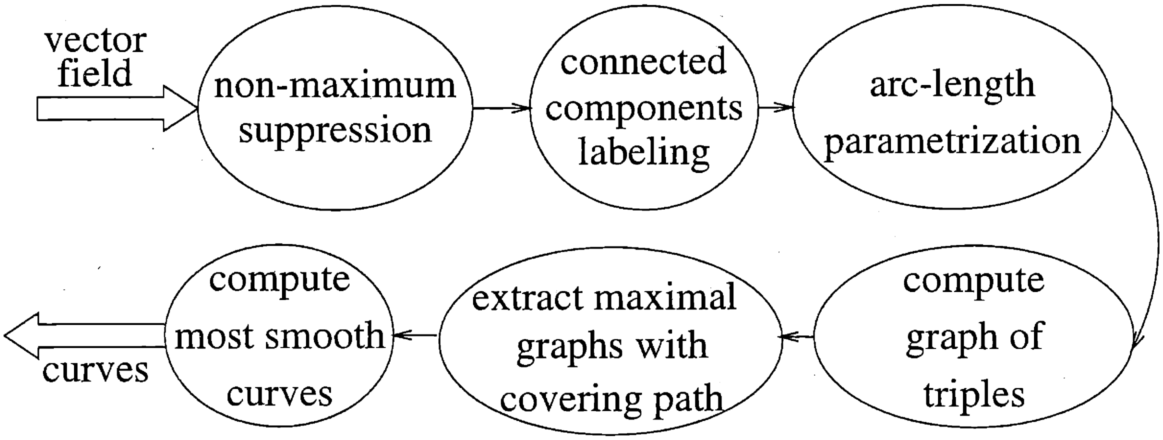

vector field
non-maximum suppression
connected components labeling
arc-length parametrization
compute graph of triples
extract maximal graphs with covering path
compute most smooth curves
curves

Figure 2-5: Flow chart of the algorithm to compute curves from a vector field.

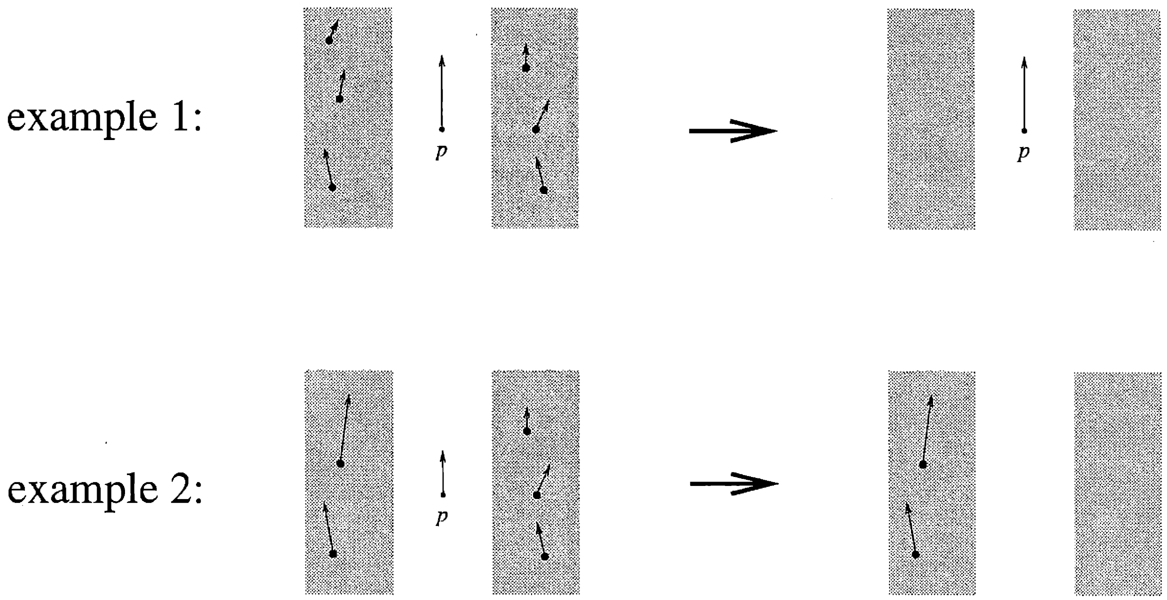


Figure 2-6: First step of non-maximum suppression. The figure illustrates what happens when the suppression routine is applied to the point p. In example 1 the point p "survives" while in example 2 it is suppressed.

curves which represent these boundaries. Many edge detectors select these points by searching for locations where brightness change is maximum in the direction perpendicular to the estimated boundary orientation. A threshold can also be applied to reject maxima created by noise.

This maximization (or non-maximum suppression) step has to be done in the right way since undesired effects such as curve disconnections and wiggling can occur if some points are erroneously suppressed (see section 1.2.4 and figures 1-2, 1-3 on page 21). On the other hand, enough points have to be suppressed so that curve interpolation can be carried out efficiently on the remaining set of points.

The non-maximum suppression technique proposed here addresses these issues. It will be proved that with the right choice of parameters this method allows the reconstruction of curves with bounded curvature without disconnections if the vector field which represents brightness variations has the properties explained in section 2.1.1 in the neighborhood of every curve.

The suppression procedure consists of two independent steps. Step one (figure 2-6) suppresses those points $p \in G$ where the field magnitude is not larger than the magnitude in the two lateral regions $F_1(p)$ and $F_2(p)$ shown in figure 2-7. Namely, if there exists $p' \in G \cap F(p)$, where $F(p) = F_1(p) \cup F_2(p)$, such that $|\phi(p')| \geq |\phi(p)|$ then p is suppressed. Also, if $p' \in G \cap F(p)$ and, instead, $|\phi(p)| \geq |\phi(p')|$ then p' is suppressed. Let $\sigma(p)$, $p \in G$ be a boolean variable which is initially set to zero and is set to one whenever p is marked for suppression. Then the first part of non-maximum suppression consists of the following procedure.

```
For every $p \in G$
        For every $p' \in G \cap F(p)$
```

$$\text{If } |\phi(p')| \geq |\phi(p)| \text{ then } \sigma(p) := 1$$
$$\text{If } |\phi(p)| \geq |\phi(p')| \text{ then } \sigma(p') := 1$$

Notice that points close enough to p where the field magnitude is smaller than $|\phi(p)|$ are not marked for suppression. This reduces the chances that a point be erroneously eliminated because of field fluctuations caused by noise.

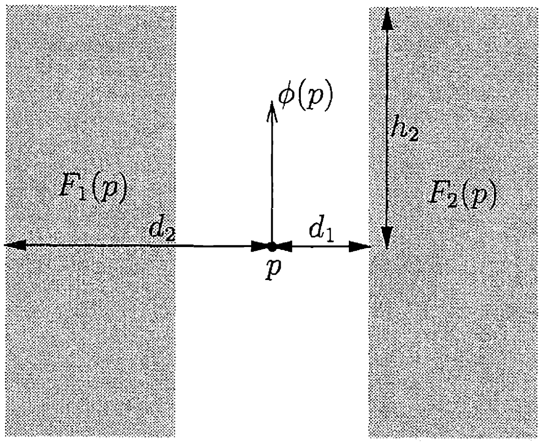


Figure 2-7: The lateral regions $F_1(p)$ and $F_2(p)$ used during the first step of non-maximum suppression.

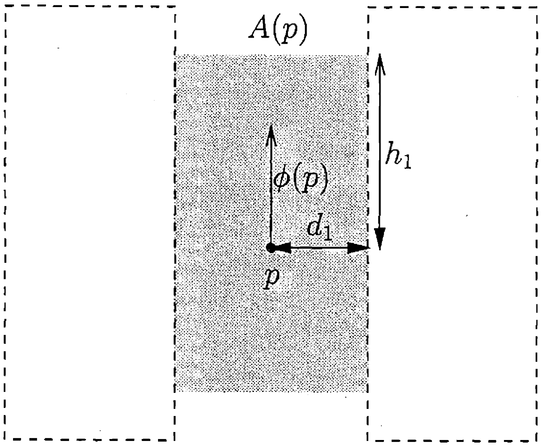


Figure 2-8: The neighborhood $A(p)$ used during the second step of non-maximum suppression.

Step two (figure 2-9) suppresses a point $p \in G$ if its neighborhood $A(p)$ (shown in figure 2-8) contains a point p' with higher field magnitude *and* field orientation which differs more than a threshold $\overline{\Theta}$ from the orientation at p. If instead the field magnitude is smaller at p', then p' is suppressed. That is,

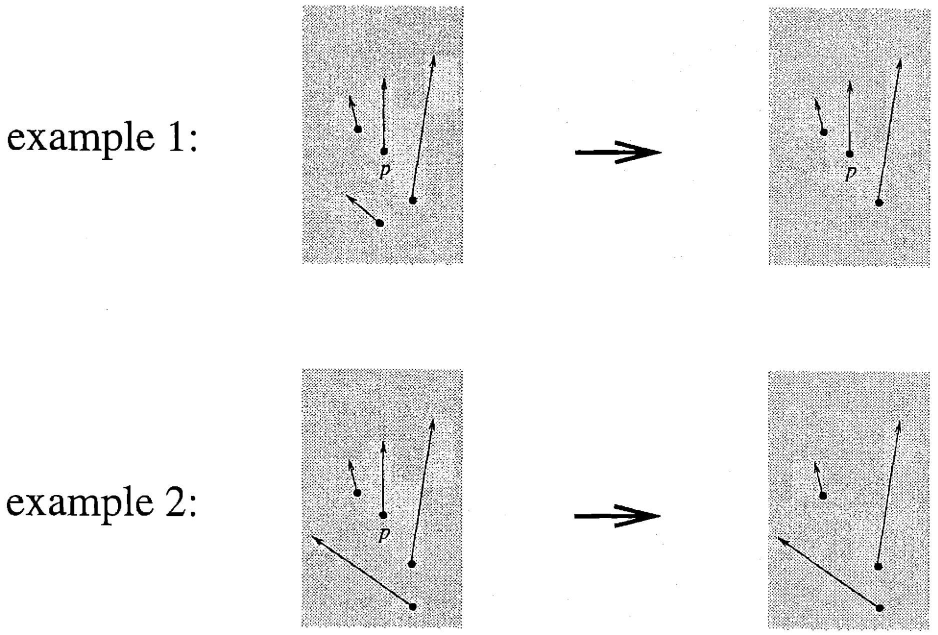


Figure 2-9: Second step of non-maximum suppression. The figure illustrates what happens when the suppression routine is applied to the point p. In example 1 the point p "survives" while in example 2 it is suppressed.

```
For every $p \in G$
    For every $p' \in G \cap A(p)$
        If $||\theta(p) - \theta(p')|| \geq \overline{\Theta}$
            If $|\phi(p')| \geq |\phi(p)|$ then $\sigma(p) := 1$
            If $|\phi(p)| \geq |\phi(p')|$ then $\sigma(p') := 1$
```

Notice that if the field orientation in $A(p)$ does not vary more than $\overline{\Theta}$ then no point is suppressed inside $A(p)$ by step two. The set of selected or *dominant* points will be denoted by P:

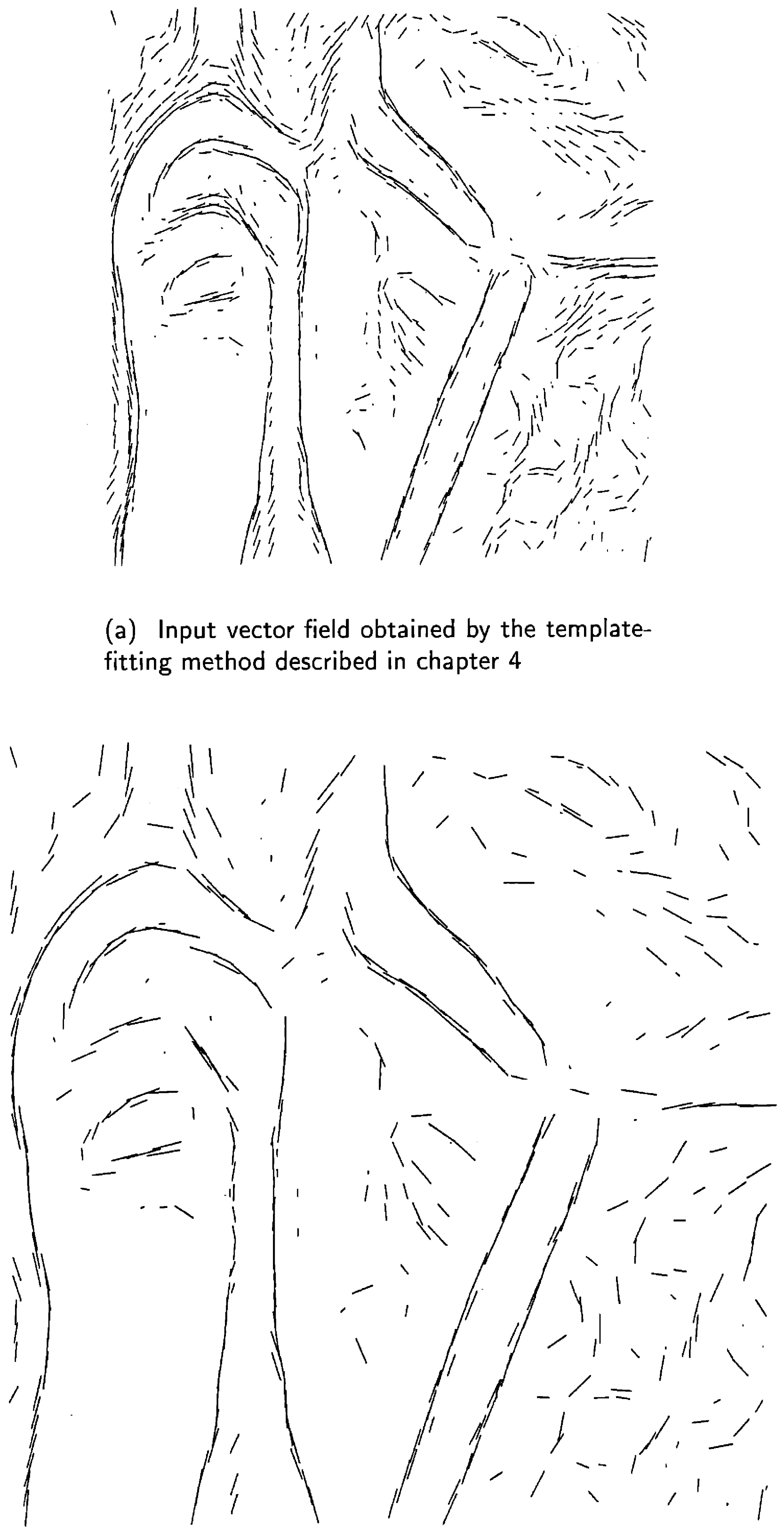
$$P = \{p \in G \ : \ \sigma(p) = 0\}$$

Notice that the points are *not* suppressed while they are scanned but just "marked" for suppression (that is, the variable $\sigma(p)$ is set to one). Suppression occurs only at the end. Therefore, the set P does not depend on the order by which the points in G are scanned.

For every point $p \in P$, the lateral regions $F_1(p)$ and $F_2(p)$ do not contain any points which belong to P (proposition 3 on page 57). Also, the orientation of the field at the points in $A(p) \cap P$ differs less than $\overline{\Theta}$ from the orientation at p (proposition 4 on page 57).

2.4 Connected components of P

Let $\mathcal{G}$ be the graph whose nodes are the points in P and whose arcs are the pairs $(p_1, p_2) \in P \times P$ such that $||p_1 - p_2||$ is less than a threshold a. This threshold is



(a) Input vector field obtained by the template-fitting method described in chapter 4

(b) Vector field after non-maximum suppression

Figure 2-10: Result of non-maximum suppression.

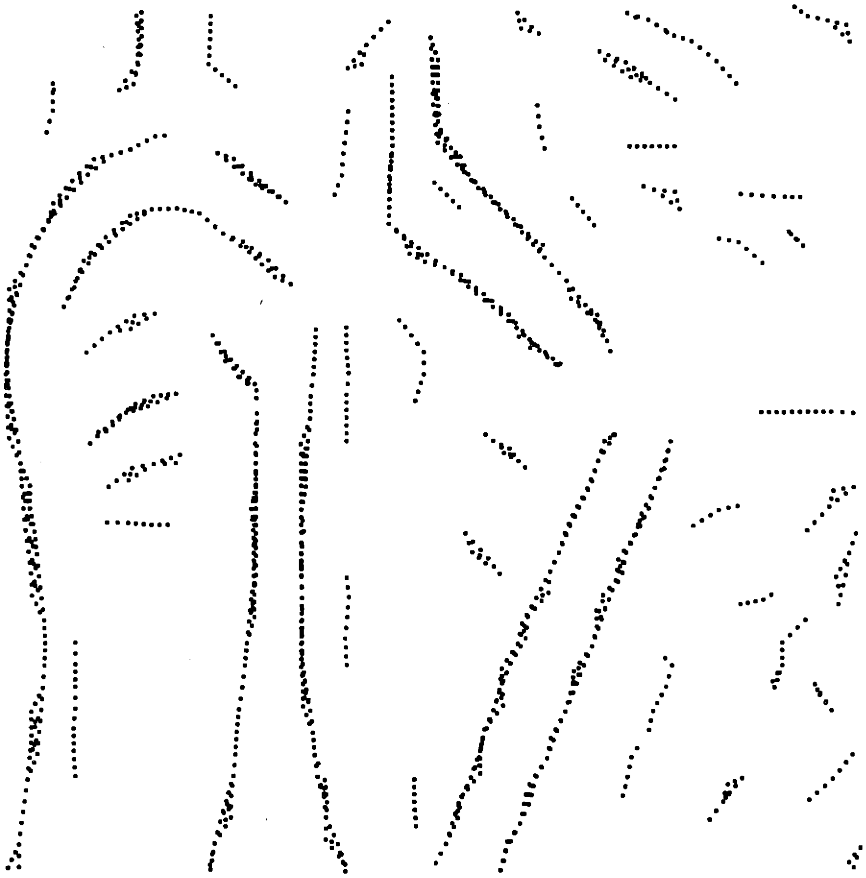


Figure 2-11: The set of dominant points P. This set is decomposed into connected components. Here, only the components with at least 5 points are shown. Note that each component has a thin elongated shape.

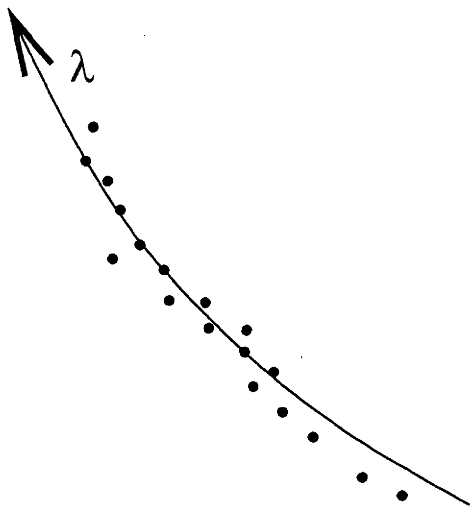


Figure 2-12: A component Q and its longitudinal coordinate λ.

essentially the sampling resolution of the vector field and can be made arbitrarily small by sampling the field densely enough.

A *path* in $\mathcal{G}$ is a sequence of points $p_1, \ldots, p_n$ such that $||p_i - p_{i+1}|| \leq a$ for $i = 1, \ldots, n-1$. A subset of P is said to be connected if there exists a path which connects any two points in the subset. The *connected components* $P_1, \ldots, P_K$ of the graph are the maximal connected subsets of P. The following properties hold:

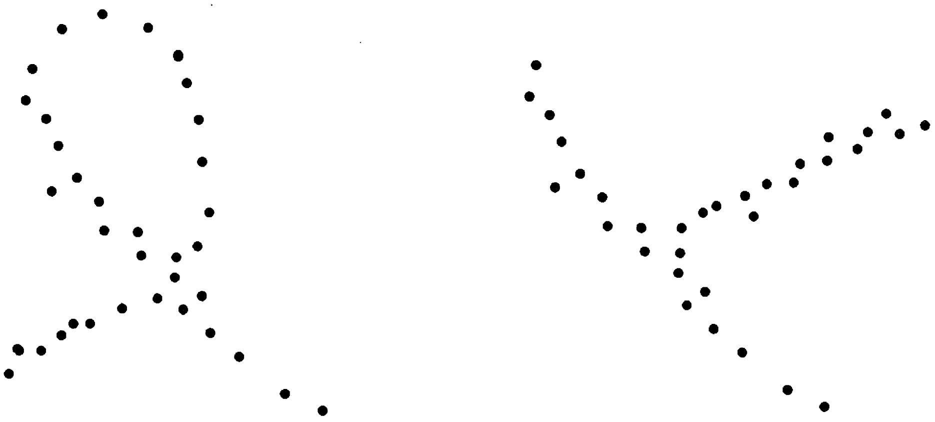
- $P = \bigcup_{k=1}^{K} P_k$
- $P_k \cap P_{k'} = \emptyset$ for $k \neq k'$
- A path exists between two points if and only if they belong to the same connected component

Step 2 of the algorithm computes the connected components $P_1, \ldots, P_K$ of $\mathcal{G}$. A unique integer label $k(p)$ is assigned to every point $p \in P$ which identifies the connected component containing p. The remaining steps of the algorithm are applied independently to each component P_k, denoted by Q from now on.

2.5 Arc-length parameterization

Each component Q can have either a thin elongated shape (see figure 2-12), or the shape of a closed ring. For simplicity, we consider here only the first case. The second case can be dealt with by introducing a few adjustments, discussed at the end of this section.

More complex shapes, such as those shown in figure 2-13 do not occur because of the way non-maximum suppression is done. This will be proven (implicitly) in section 3.5.



self-intersecting curve

T-junction

Figure 2-13: Some of the shapes that the component Q *can not* have. Notice that these two sets do not allow a global longitudinal coordinate.

Step 3 assigns a longitudinal coordinate, or arc length parameter $\lambda(p)$, to every point $p \in Q$. A coarse description of the procedure which defines the longitudinal coordinates is as follows:

- A point s_0 is chosen randomly from Q
- Two sequences $s_0, s_1^+, \ldots, s_{L^+}^+$ and $s_0, s_1^-, \ldots, s_{L^-}^-$ are constructed. Each sequence converges toward one of the two ends of Q.
- Coordinates are assigned to these two sequences of points such that

$$\lambda(s_{L^-}^-) < \cdots < \lambda(s_{-1}^-) < \lambda(s_0) = 0 < \lambda(s_1^+) < \cdots < \lambda(s_{L^+}^+)$$

- The coordinates are interpolated to all the remaining points in Q

An important property of the coordinate map λ is that the distance between two points with the same coordinates is small[4]. A rough upper bound to it is about $2d_1/\cos\overline{\Theta}$.

Extraction of the two sequences The points $s_1^+, \ldots, s_{L^+}^+$ are computed recursively starting from s_0 by using the search region $S^+(s_i^+)$ shown in figure 2-14. For every $i \geq 0$, the point s_{i+1}^+ is chosen randomly in $S^+(s_i^+) \cap Q$ until $S^+(s_i^+) \cap Q = \emptyset$. Thus we have:

- $s_{i+1}^+ \in S^+(s_i^+) \cap Q$

[4] Notice that since this coordinate map is defined from a two-dimensional domain to a one-dimensional domain two points with the same coordinate are not necessarily identical.

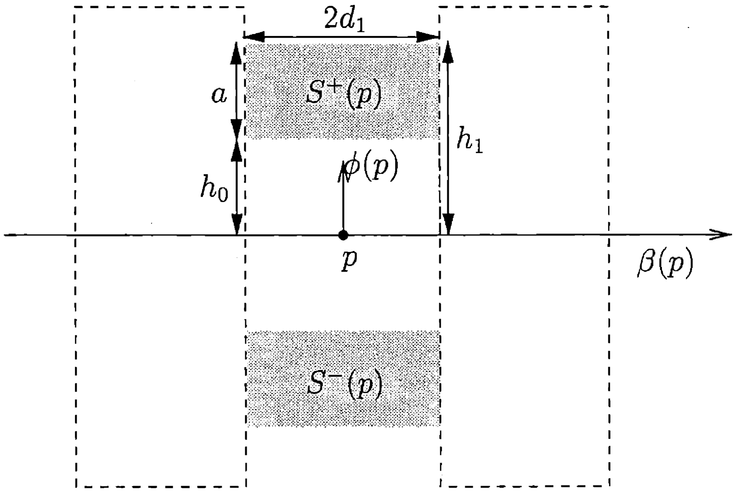


Figure 2-14: The search regions $S^+(p)$ and $S^-(p)$. For each point s_i^+, the next point s_{i+1}^+ is chosen from the region $S^+(s_i^+)$. Similarly, s_{i+1}^- is chosen from the region $S^-(s_i^-)$. Notice that $A(p)$ is the area between the lateral regions delimited by (and including) $S^+(p)$, $S^-(p)$ (compare with figure 2-8).

- $S^+(s_{L^+}^+) \cap Q = \emptyset$

The width of $S^+(s_i^+)$ is $2d_1$, its height a, and its distance from the baseline $\beta(s_i^+)$ is (see figure 2-14)

$$h_0 = 2d_1 \tan \overline{\Theta} \tag{2.2}$$

The height of the regions $A(p)$ used during non-maximum suppression is

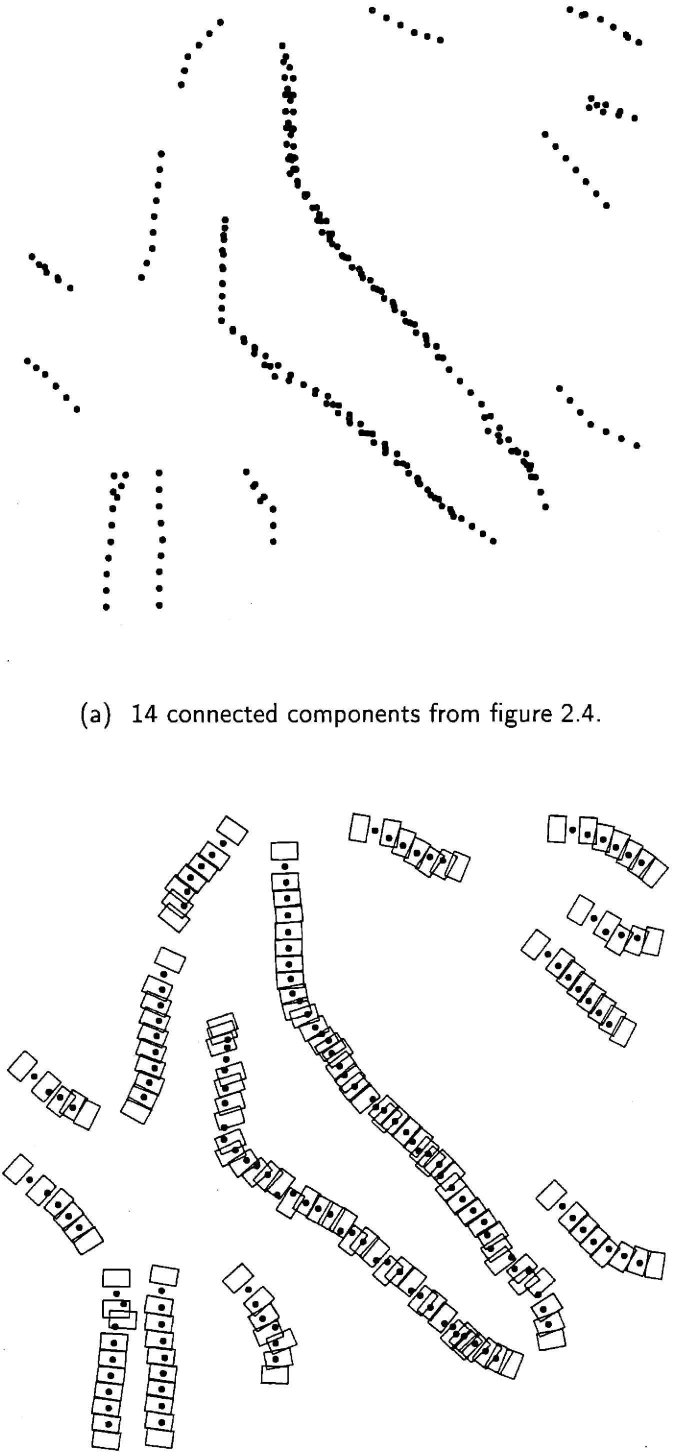
$$h_1 = h_0 + a$$

so that the region $S^+(s_i^+)$ is the top portion of $A(s_i^+)$ (see figure 2-14). Thus $s_{i+1}^+ \in A(s_i^+)$ and therefore, by proposition 4 on page 57,

$$||\theta(s_{i+1}^+) - \theta(s_i^+)|| \leq \overline{\Theta} \tag{2.3}$$

The sequence $s_0, s_1^-, \ldots, s_{L^-}^-$ is computed in a similar way by using the search regions $S^-(s_i^-)$. Figure 2.5 shows the sequences $s_{L^-}^-, \ldots, s_{L^+}^+$ for several components Q. We describe the remaining steps only for the sequence $s_1^+, \ldots, s_{L^+}^+$. They are done similarly for $s_1^-, \ldots, s_{L^-}^-$.

Coordinate assignment The coordinates $\lambda(s_i^+)$ are assigned as follows. First let $\lambda(s_0) = 0$. Then, for $i > 0$, $\lambda(s_{i+1}^+)$ is computed recursively by adding to $\lambda(s_i^+)$ an estimate of the distance between the baselines $\beta(s_{i+1}^+)$ and $\beta(s_i^+)$ in the neighborhood of s_{i+1}^+, s_i^+. If $\theta(s_{i+1}^+) = \theta(s_i^+)$ then the baselines $\beta(s_{i+1}^+)$ and $\beta(s_i^+)$ are parallel and one can set $\lambda(s_{i+1}^+) - \lambda(s_i^+)$ equal to their distance. For the general case when



(a) 14 connected components from figure 2.4.

(b) The sequences $s_{L^-}^-, \ldots, s_{L^+}^+$.

Figure 2-15: The sequences extracted during arc-length parameterization for 14 connected components. Note the initial point which is the only one not belonging to a search region. It is chosen according to some scanning order.

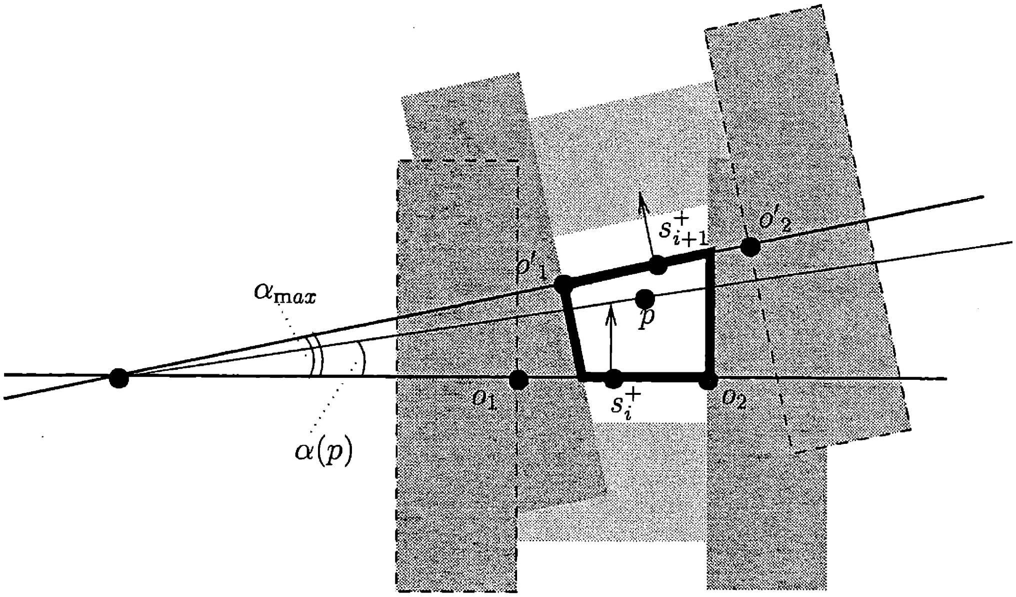


Figure 2-16: Notation for the interpolation formula (2.4).

$\theta(s^+_{i+1}) \neq \theta(s^+_i)$ let

$$\lambda(s^+_{i+1}) = \lambda(s^+_i) + \frac{1}{2}d(s^+_i, \beta(s^+_{i+1})) + \frac{1}{2}d(s^+_{i+1}, \beta(s^+_i))$$

where $d(p, \beta)$ denotes the distance of the point p from the straight line β.

Coordinate interpolation The coordinate map λ is interpolated to other points in such a way that the baselines $\beta(s^+_i)$ are level curves for λ. The other level curves are the straight lines passing through the intersection point of $\beta(s^+_{i+1})$ with $\beta(s^+_i)$ and lying between them (see figure 2-16). More precisely,

$$\lambda(p) = \left(1 - \frac{\alpha(p)}{\alpha_{max}}\right)\lambda(s^+_i) + \frac{\alpha(p)}{\alpha_{max}}\lambda(s^+_{i+1}) \tag{2.4}$$

By using (2.2) and (2.3) it can be proved (proposition 5 on page 58) that the intersection point of $\beta(s^+_{i+1})$ with $\beta(s^+_i)$ lies outside $A(s^+_i)$ and $A(s^+_{i+1})$ and therefore the interpolation (2.4) is well defined for all the points between the empty regions $F_1(s^+_i) \cup F_1(s^+_{i+1})$ and $F_2(s^+_i) \cup F_2(s^+_{i+1})$.

For the points lying above the most positive baseline $\beta(s^+_{N+})$, the interpolation formula (2.4) can not be used. The coordinate of these points is defined as

$$\lambda(p) = \lambda(s^+_{N+}) + d(p, \beta(s^+_{N+})) \tag{2.5}$$

Notice that the interpolation formulae (2.4) and (2.5) guarantee that λ is continuous.

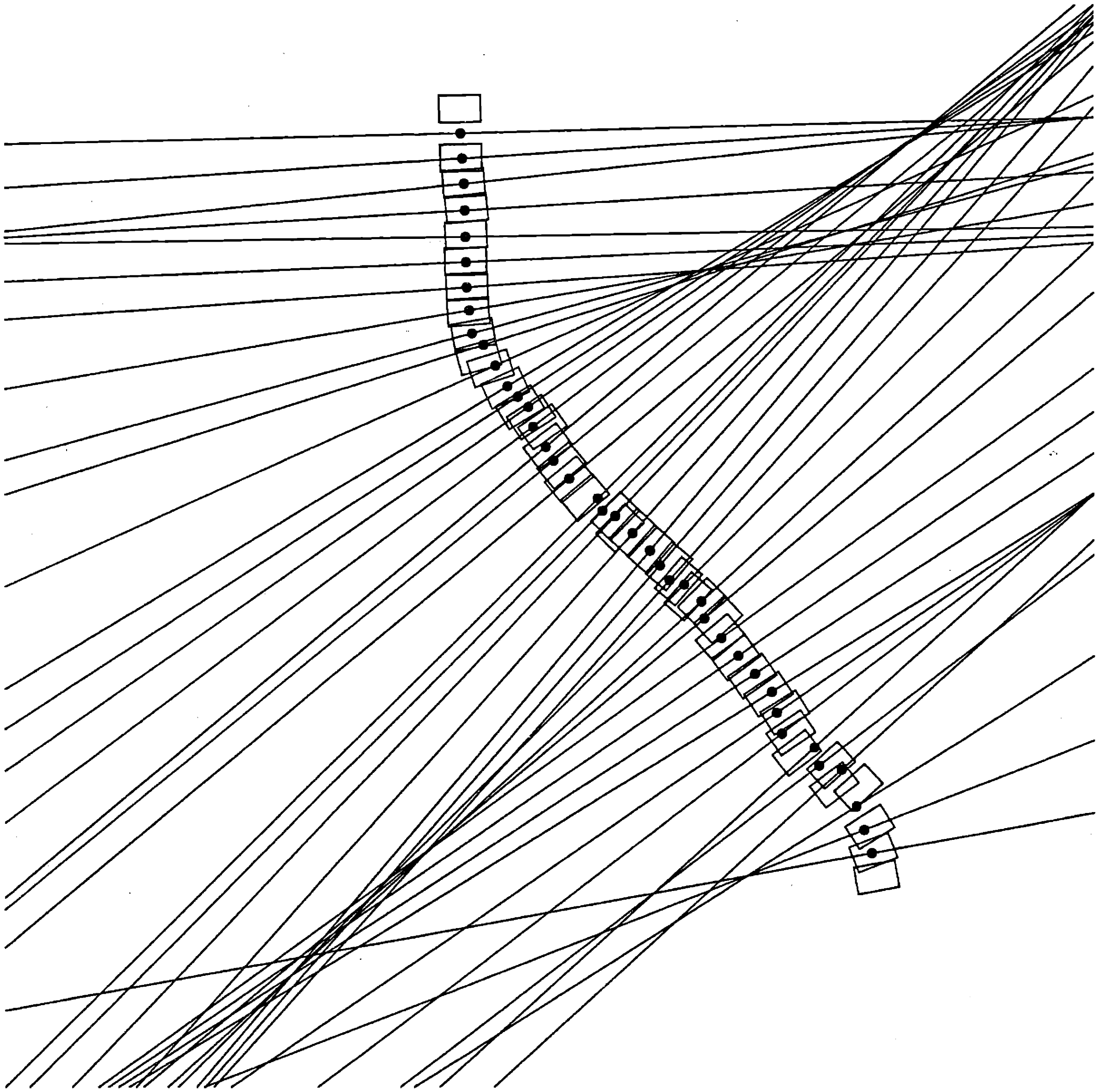


Figure 2-17: The level curves of the longitudinal coordinate λ, namely the curves on which λ is constant. They are the straight lines $\beta(p_i)$ perpendicular to the field $\phi(p_i)$ for every point p_i in the extracted sequence $s^-_{L^-}, \ldots, s^+_{L^+}$.

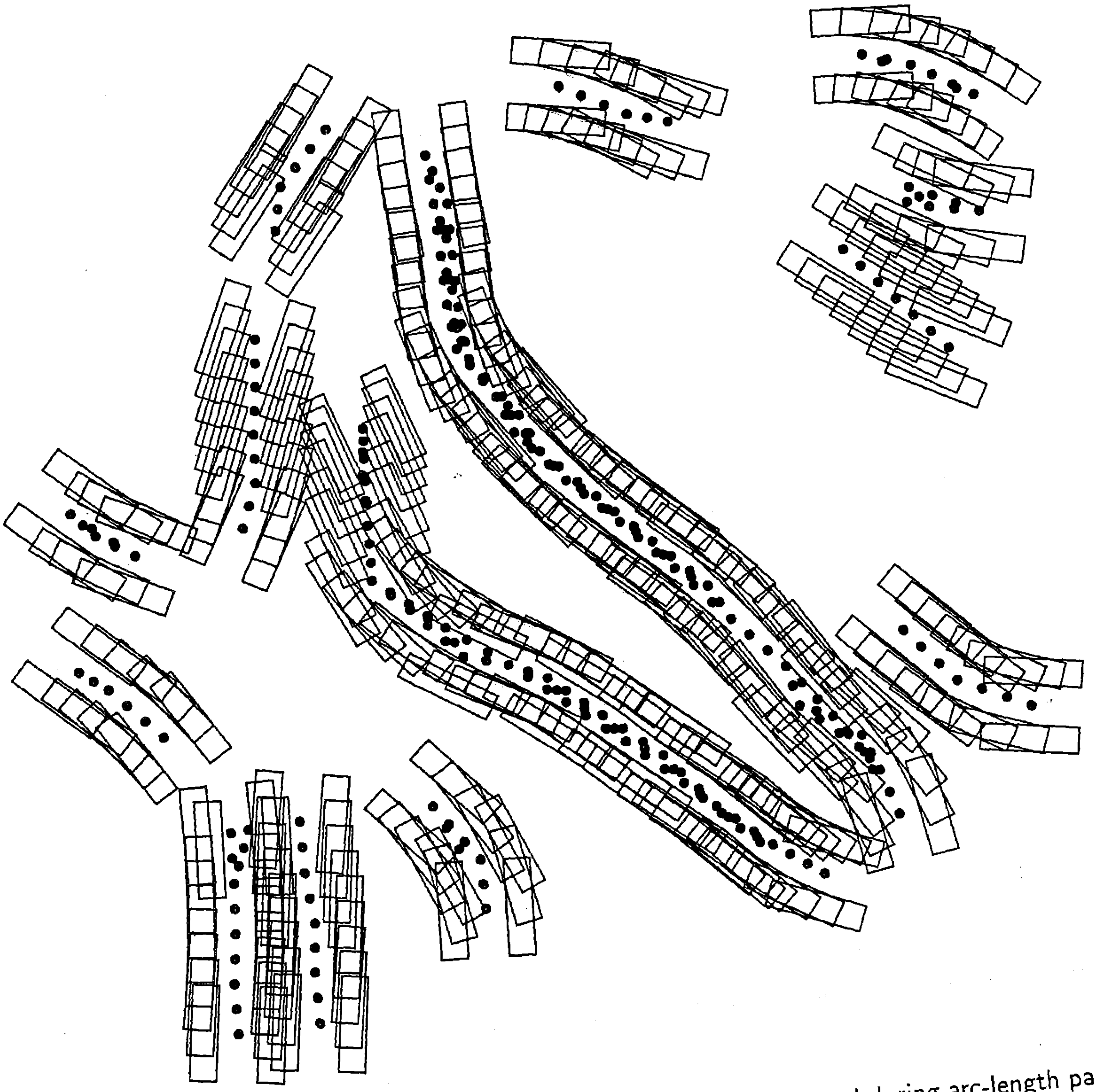


Figure 2-18: Lateral regions of the sequences $s_{L^-}^-, \ldots, s_{L^+}^+$ extracted during arc-length parameterization. Note that these regions form two lateral "channels" which do not contain any point in P.

2.5.1 Domain of the map λ.

For this procedure to work, it is necessary that d_2 and h_2 be large enough. Section 3.9 deals with the problem of determining suitable values for these parameters and how they depend on d_1, $\overline{\Theta}$ and a. In section 3.5 it is proved that with the right choice of the parameters the map λ constructed by the procedure is well defined on the connected subset of $\mathbb{R}^2$ given by the union of all segments between points $p_1, p_2 \in Q$ such that $||p_1 - p_2|| \leq a$.

2.5.2 Detecting components with no ends.

To make the procedure work in the more general situation, namely when Q might have the shape of a ring, it is necessary to introduce a few modifications. If Q is ring-shaped the recursion which extracts $s_1^+, \ldots, s_{L+}^+$ returns eventually to a neighborhood of the starting point s_0. This event can be detected by labeling the points as they are encountered. As soon as a point is visited a second time the recursion interrupts. Clearly, there is no need to extract the sequence in the other direction. Also, it is impossible to construct a continuous coordinate in this case. However, the remaining part of the algorithm can be adjusted to work also with this single point of discontinuity. But, for simplicity, it will be still assumed that no ring-like components are present.

2.6 Graph of curve fragments

To compute curves, steps 4 through 6 of the algorithm extract one or more sequences of points $p_1, \ldots, p_N$ from every Q and then interpolate a continuous curve through this sequence of points. The simplest way to do this interpolation is by means of a polygonal curve, which is obtained by joining with a straight-line segment every pair of consecutive points p_i, p_{i+1}. Higher order splines could also be used. However, to keep the analysis simple, only polygonal curves are considered here.

What criteria should be used to extract the sequence $p_1, \ldots, p_N$? That is, what are the desired properties of the polygonal curve defined by $p_1, \ldots, p_N$? One requirement is that the curve be as long as possible. Ideally, a single curve from one end of Q to the other would be desirable. Also, the curve should be "smooth" or "simple" according to some definition. For instance, the total turn — which for a polygonal curve is just the sum of all the angles at its vertices — should be small.

To evaluate the smoothness or regularity of the curve at some point one has to look at a neighborhood of this point. If the curve is polygonal then it is sufficient to consider a small number of consecutive vertices. For instance, to determine the total turn of the curve one has to consider all triples of consecutive vertices. In fact, the angle at every vertex depends also on the preceding and following vertex.

This suggests that a curve should not be viewed as a sequence of points but, instead, as a sequence of neighborhoods or small curve fragments. If the curve is polygonal then these fragments consist of n-tuples of consecutive vertices.

Step 4 of the algorithm computes a set of triples of points to be used as fragments of a polygonal curve. A number of constraints can be imposed on these triples (p_1, p_2, p_3):

- Constraint on length of segments:

$$r \leq ||p_1 - p_2|| \leq r + a, \qquad r \leq ||p_2 - p_3|| \leq r + a \tag{2.6}$$

  The parameter r represents a lower bound on the length of a curve segment. The constraint says that the the length of these segments should be as constant as possible. However, it is necessary to leave some uncertainty in these lengths to avoid disconnections. Recall that a is the sampling resolution of the input discrete vector field Φ.

- Alignment of the points (p_1, p_2, p_3) with the longitudinal coordinate:

$$\lambda(p_1) < \lambda(p_2) < \lambda(p_3) \tag{2.7}$$

- Alignment of the segments $\overline{p_1p_2}$, $\overline{p_2p_3}$ with the vector field:

$$||\alpha_{12} - \theta(p_1)|| \leq \Theta_2, \qquad ||\alpha_{12} - \theta(p_2)|| \leq \Theta_2 \tag{2.8}$$

$$||\alpha_{23} - \theta(p_2)|| \leq \Theta_2, \qquad ||\alpha_{23} - \theta(p_3)|| \leq \Theta_2 \tag{2.9}$$

  where α_{12}, α_{23} are the orientation of the segments $\overline{p_1p_2}$, $\overline{p_2p_3}$. These conditions ensure that the orientation of computed curve is similar to the orientation $\theta(p)$ of the vector field.

- Upper bound on the angle at vertex p_2:

$$\widehat{p_1p_2p_3} \leq \Theta_3 \tag{2.10}$$

Let T be the set of all the triples satisfying these constraints. A directed graph $\mathcal{T}$ with nodes T can be defined by putting an arc between any two triples (p_1, p_2, p_3), $({p'}_1, {p'}_2, {p'}_3) \in T$ if $p_2 = {p'}_1$ and $p_3 = {p'}_2$. Namely, all the pairs of triples of the form (p_1, p_2, p_3), (p_2, p_3, p_4) are connected by an arc. A path in this graph with n nodes specifies a polygonal curve with $n + 2$ vertices. Conversely, a polygonal curve with at least three vertices and whose consecutive triples of vertices satisfy the constraints yield a path in this graph.

2.7 Maximal coverable subgraphs

In general, there might be no polygonal curve satisfying the constraints formulated in section 2.6 which covers the whole component Q, namely such that

$$\lambda(p_1) = \min_{p \in Q} \lambda(p), \quad \lambda(p_N) = \max_{p \in Q} \lambda(p)$$

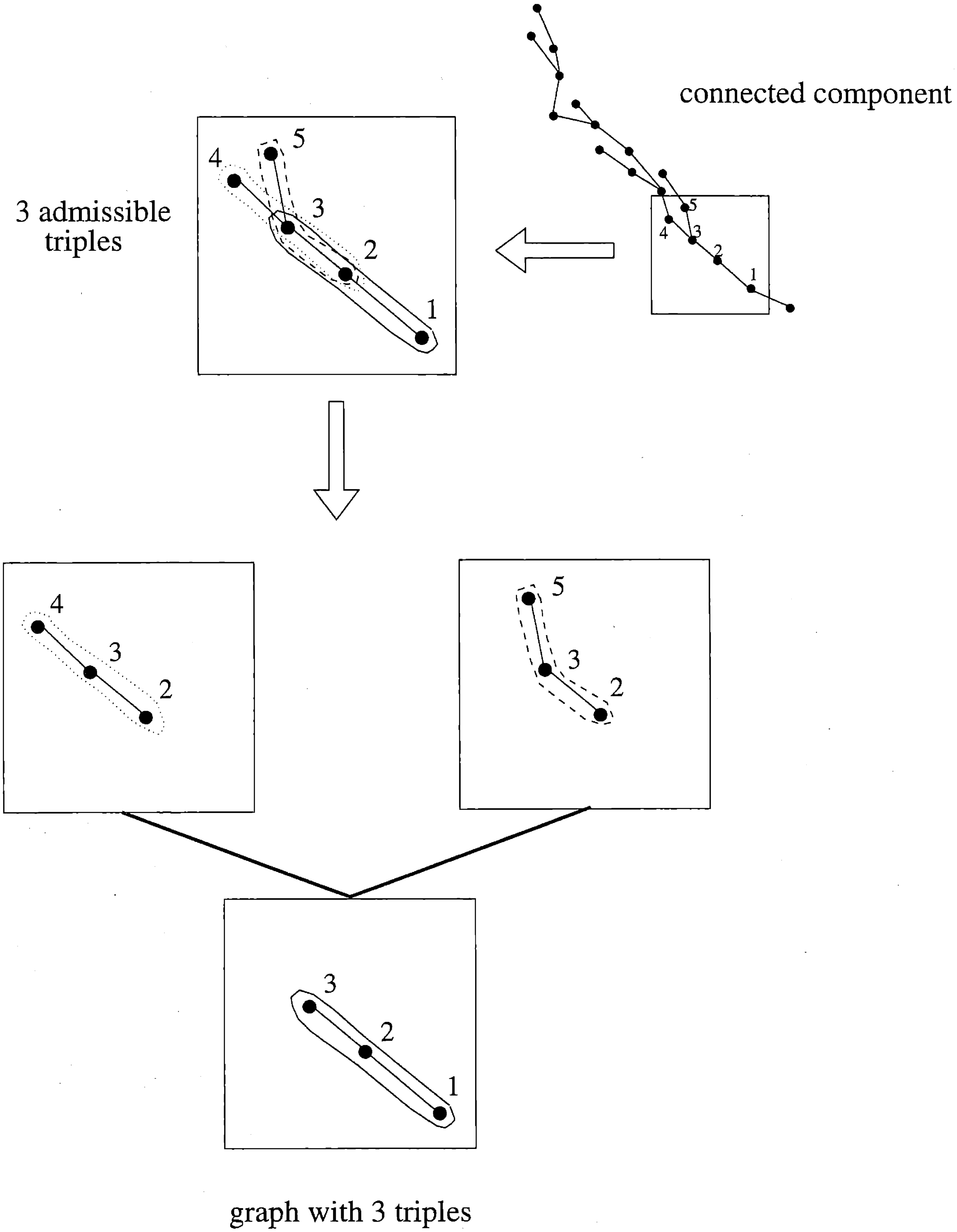


Figure 2-19: Graph of triples.

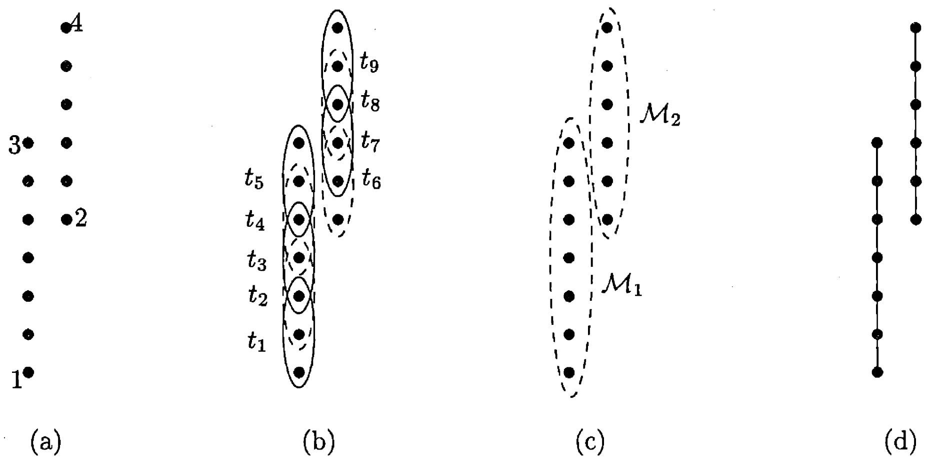


Figure 2-20: Example where a single curve is not sufficient to cover the whole component Q. Here, triples are allowed only if they lie on a straight line. (a): the connected component Q. (b): the admissible triples. Notice that there is no triple connecting the left column of points to the right one. Thus $\mathcal{T}$ does not contain any path which connects 1 to 4. (c): The two maximal coverable graphs extracted by step 5 of the algorithm. (d): The polygonal curves computed by step 6.

where p_1 and p_N are the vertices at the ends of the curve. Step 5 of the algorithm extracts the maximal subgraphs of $\mathcal{T}$ for which there exists a covering path, namely a path connecting one end of the subgraph to the other (see figure 2-20).

For any $t = (p_1, p_2, p_3) \in T$, let

$$\lambda^-(t) = \lambda(p_1), \quad \lambda^+(t) = \lambda(p_3)$$

If $\mathcal{S}$ is a subgraph of $\mathcal{T}$ with nodes $S \subset T$, then we say that a path $t_1, \ldots, t_n$ in $\mathcal{S}$ covers $\mathcal{S}$ if

$$\lambda^-(t_1) = \min_{t\in S} \lambda^-(t), \quad \lambda^+(t_n) = \max_{t\in S} \lambda^+(t)$$

$\mathcal{S}$ is *coverable* if it contains a path which covers it. The goal of step 5 is to compute all the maximal coverable subgraphs of $\mathcal{T}$, denoted $\mathcal{M}_1, \ldots, \mathcal{M}_J$. A description of the procedure follows.

Step 5.1 For every $t \in T$, let $R(t)$ be the set of all t' reachable from t, that is all t' for which a path from t to t' exists. Then let

$$\lambda^*(t) = \max_{t' \in R(t)} \lambda^+(t') \tag{2.11}$$

$\lambda^*(t)$ is computed by means of a dynamic programming procedure. In fact, let $N(t)$ be the set of elements t' such that an arc from t to t' exists. Then, consider the equation

$$\lambda^*_{i+1}(t) = \max\{\lambda^*_i(t') \ : \ t' \in N(t) \cup \{t\}\} \tag{2.12}$$

with initial condition

$$\lambda_0(t) = \lambda^+(t) \tag{2.13}$$

Notice that, for any i, $\lambda_i^*(t) \leq \lambda^*(t)$ because the max in (2.12) involves only nodes which can be reached from t. Furthermore, if $n(t)$ is the number of nodes in a path from t to a node t' which achieves the max in (2.11), then $\lambda_i^*(t) = \lambda^*(t)$ for $i \geq n(t)-1$. Therefore, for i larger than the number of nodes in $\mathcal{T}$, the equation (2.12) converges and $\lambda^*(t) = \lambda_i^*(t)$.

Step 5.2 Let $\mathcal{I}$ be the family of all intervals $[\lambda^-(t), \lambda^*(t)] \subset \mathbb{R}$ for all $t \in T$. An interval $I \in \mathcal{I}$ is maximal if there is no $I' \in \mathcal{I}$ such that $I \subset I'$. This step computes the set I^* of all the maximal intervals in $\mathcal{I}$.

Step 5.3 Each element $I_j = [l_1^j, l_2^j]$ of I^* defines a maximal coverable subgraph $\mathcal{M}_j$ whose nodes are

$$M_j = \left\{t \in T \;:\; l_1^j \leq \lambda^-(t),\; \lambda^*(t) \leq l_2^j\right\}$$

Note that each M_j contains at least one path $t_1, \ldots, t_n$ such that $\lambda^-(t_1) = l_1^j$, $\lambda^+(t_n) = l_2^j$. Let $t_1^j, t_2^j \in M_j$ be such that $\lambda^-(t_1^j) = l_1^j$, $\lambda^+(t_2^j) = l_2^j$.

2.8 Curves with minimum turn

The last step of the algorithm computes the curve with minimum turn which covers each subgraph $\mathcal{M}_j$. The total turn of a polygonal curve is defined as

$$E = \sum_{i=1}^{n} \alpha(t_i)$$

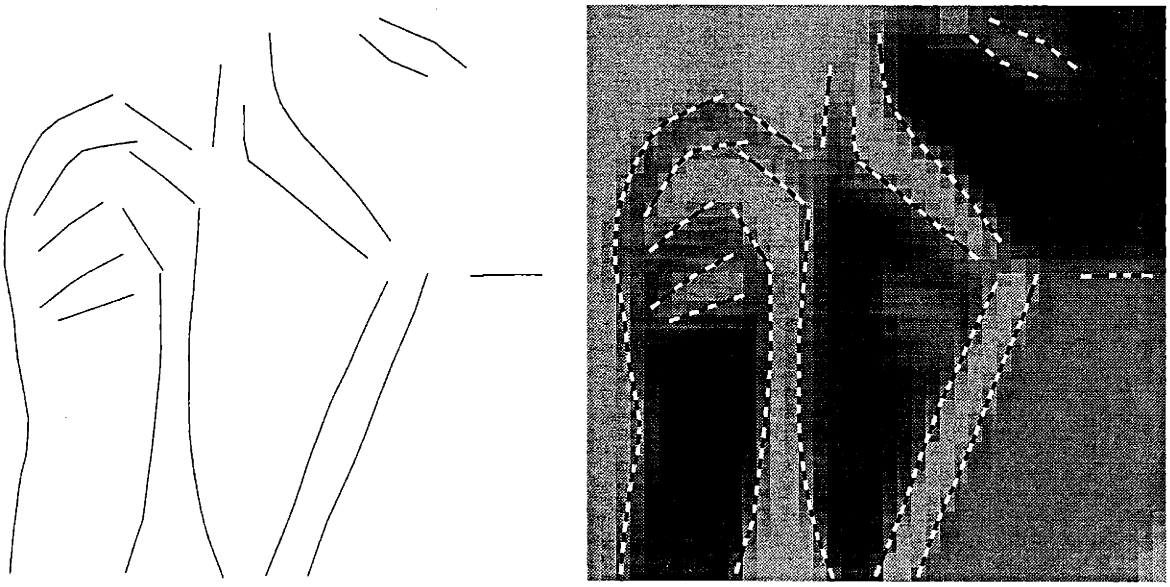
where $t_1, \ldots, t_n$ is the sequence of triples of the curve and $\alpha(t_i)$ is the angle of i-th triple. The following equation

$$E_{i+1}(t) = \alpha(t) + \min_{t' \in N(t)} E_i(t), \quad i > 0 \tag{2.14}$$

with initial condition

$$E_0(t) = \begin{cases} \alpha(t) & \text{if } \lambda^+(t_n) = l_2^j \\ \infty & \text{otherwise} \end{cases}$$

can be used to find the curve which minimizes the total turn. To prove this, let $n(t)$ be the number of nodes in a path from t to t_2^j which minimizes the total turn. If there is no such path then $n(t) = \infty$. Then, by induction, one can verify that for $i \geq n(t) - 1$, $E_i(t)$ is the minimum turn of a path from t to t_2^j. On the other hand, if there is no path from t to t_2^j, then $E_i(t) = \infty$ for all i. Therefore, for i larger than



(a) Computed curves

(b) Computed curves on brightness image

Figure 2-21: The final result of the algorithm on the vector field shown in figure 2-1(c).

the number of nodes in $\mathcal{T}$, the equation (2.14) converges. Since there is a path from t_1^j to t_2^j by assumption, then $E_i(t_1^j) \leq \infty$. The node which follows t in a minimizing path from t_1^j to t_2^j is $\text{argmin}_{t' \in N(t)} E_i(t)$.

Chapter 3

Bound on localization error and parameter setting

In this chapter the main result is stated and proved. This result says that all curves with bounded curvature which satisfy the model defined in section 3.1 are detected by the algorithm described in chapter 2. An upper bound to the localization error is given which vanishes linearly when the "noise parameter" w and the sampling distance a go to zero. Moreover, as part of this theoretical analysis, it will be explained how to set all the parameters of the algorithm in terms of the a priori parameters w, Θ, a.

For simplicity, it will be assumed that the domain G of the vector field $\Phi = \{\phi(p) \ : \ p \in G\}$ is a regular grid with sampling distance a (see figure 3-1).

3.1 The curve model and the sets $C^{\kappa}_{w,\Theta}(\Phi)$

An informal description of the curve model was given in section 2.1.1. A more precise definition is given here.

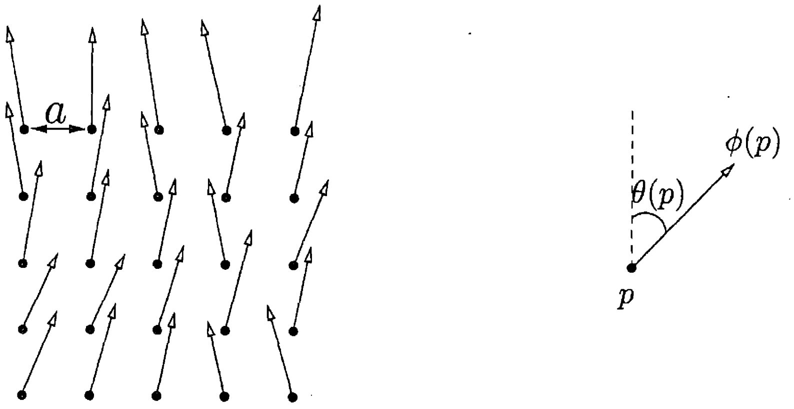


Figure 3-1: The vector field ϕ is defined on a regular grid G with sampling distance a. The orientation of $\phi(p)$ is denoted $\theta(p)$.

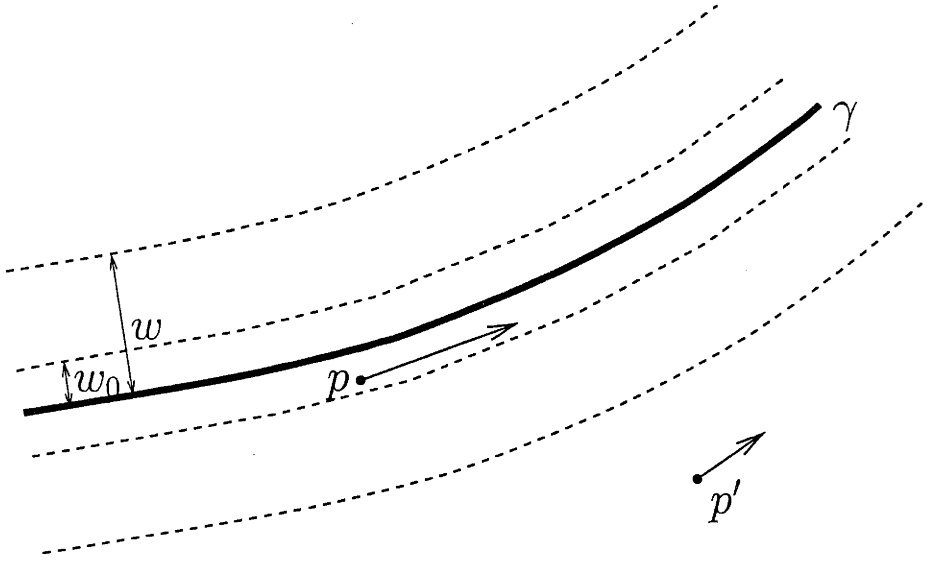


Figure 3-2: Condition (C1). If $\gamma \in C^{\kappa}_{w,\Theta}(\Phi)$ and $||p-p'||^2 \leq d_2^2 + h_2^2$ then $|\phi(p)| > |\phi(p')|$.

Definition 1 *$C^{\kappa}_{w,\Theta}(\Phi)$ is the set of all curves γ with curvature less than κ which satisfy the following two conditions:*

(C1) The field magnitude at a point p close to the curve is larger than the field magnitude at a point p' whose distance from the curve is at least w. More precisely (see figure 3-2),

$$\left.\begin{array}{rcl} d(p,\gamma) & \leq & w_0 \\ d(p',\gamma) & \geq & w \\ ||p-p'||^2 & \leq & d_2^2+h_2^2 \end{array}\right\} \quad \Rightarrow \quad |\phi(p)| > |\phi(p')| \tag{3.1}$$

where

- $w_0 = a/\sqrt{2}$
- *d_2, h_2 are the parameters of the lateral regions used for non-maximum suppression (see figure 2-7).*

(C2) The field orientation $\theta(p)$ at a distance less than w from γ is similar to the orientation of the curve tangent at p^*, where p^* is the point of γ closest to p. That is (see figure 3-3),

$$d(p,\gamma) \leq w \quad \Rightarrow \quad ||\theta(p) - \theta_\gamma(p^*)|| < \Theta \tag{3.2}$$

where $\theta_\gamma(p^*)$ denotes the orientation of the curve tangent at p^*.

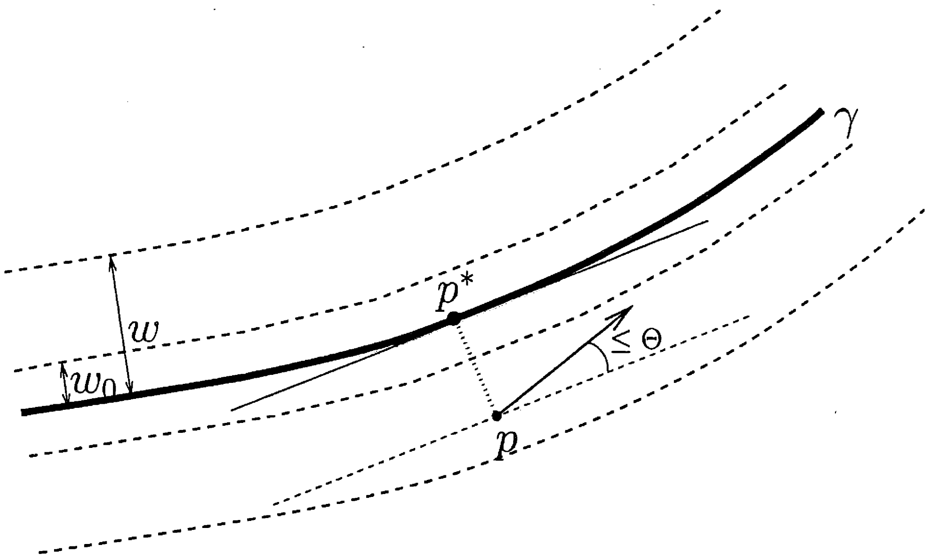


Figure 3-3: Condition (C2). If $\gamma \in C^{\kappa}_{w,\Theta}(\Phi)$ then the orientation of $\phi(p)$ with respect to to the tangent of γ at p^* is less than Θ.

3.2 Worst-case error bound: main result

Recall that d_1, d_2, h_1, h_2 are the parameters which characterize the regions used for non-maximum suppression (figures 2-7 and 2-8 on page 35). Also, $\overline{\Theta}$ is the threshold for the orientation difference used for non-maximum suppression.

The parameters $d_1, \overline{\Theta}$ are tightly related to the model parameters w, Θ. These parameters, $d_1, \overline{\Theta}$, should be set so that the points in the discrete approximation[1] of any curve $\gamma \in C^{\kappa}_{w,\Theta}(\Phi)$ must not be suppressed by non-maximum suppression. If $\kappa = 0$ and $\Theta = 0$ then one can verify that $d_1 = w + w_0$ and $\overline{\Theta} = 0$ are appropriate choices. In the more general case when $\kappa = 0$ and $\Theta > 0$, in section 3.7 it is argued that appropriate values are given by:

$$\overline{\Theta} = 2\Theta \tag{3.3}$$

$$d_1 = \frac{w + w_0}{\cos\Theta} + h_2 \tan\Theta \tag{3.4}$$

If $\kappa > 0$ then $d_1, \overline{\Theta}$ have to be larger. Let us define $\epsilon_1(\kappa)$, $\epsilon_2(\kappa)$ so that $d_1, \overline{\Theta}$ can be written as:

$$\overline{\Theta} = 2\Theta(1 + \epsilon_2(\kappa)) \tag{3.5}$$

$$d_1 = \left(\frac{w + w_0}{\cos\Theta} + h_2 \tan\Theta\right)(1 + \epsilon_1(\kappa)) \tag{3.6}$$

Notice that $\epsilon_1(0) = \epsilon_2(0) = 0$.

As it will explained in section 3.4, the parameter h_1 must be chosen so that the

[1] The discrete approximation of a curve is defined in section 3.6.

interpolation formula (2.4) for the longitudinal coordinate map λ is well defined. We get then:

$$h_1 = 2d_1 \tan \overline{\Theta} + a \tag{3.7}$$

The parameters d_2, h_2 have to be chosen so that the lateral regions of each component Q form two connected regions (see figure 2.5 on page 45). In the sections 3.5 and 3.9 it will be proved that the constraints on d_2, h_2 can be written as

$$d_2 - d_1 > \max\{d_1, a\} \tag{3.8}$$

$$h_2 = d_1 \, \overline{h}_2 \left(\overline{\Theta}, \frac{a}{d_1} \right) \tag{3.9}$$

Theorem 1 *Let Φ be a set of vectors of the form $\Phi = \{\phi(p) \, : \, p \in G\}$, where G is a regular grid with sampling distance a. Let $\hat{C}(\Phi)$ be the set of polygonal curves generated by the algorithm described in chapter 2 on input Φ. Let the parameters $\overline{\Theta}$, d_1, d_2, h_1, h_2 be chosen so that (3.5)-(3.9) hold. Then, for every $\gamma \in C^{\kappa}_{w,\Theta}(\Phi)$ there exists $\hat{\gamma} \in \hat{C}(\Phi)$ such that*

$$d(\gamma, \hat{\gamma}) \leq d_1 + w_0 \tag{3.10}$$

The proof will be given in section 3.8.

Proposition 1 *If $a \to ta$, $w \to tw$, $t > 0$, then the parameters of the algorithm can be chosen so that $(d_1 + w_0) \to t(d_1 + w_0)$. Thus in the limit $a \to 0$, $w \to 0$, the upper bound on $d(\gamma, \hat{\gamma})$ vanishes linearly with a, w.*

Proof. Let w,Θ,$\overline{\Theta}$,a,d_1,d_2,h_1,h_2 be such that (3.5)-(3.9) are satisfied. Then, it is easy to verify that for any $t > 0$, the parameters tw,Θ,$\overline{\Theta}$,ta,td_1,td_2,th_1,th_2 also satisfy (3.5)-(3.9). Thus, if $a \to ta$, $w \to tw$, and the other parameters are scaled accordingly, then one gets (recall that $w_0 = a/\sqrt{2}$): $(d_1 + w_0) \to t(d_1 + w_0)$. □

3.3 Properties of P

The following proposition follows immediately from the definition of the non-maximum suppression procedure. Recall that P denotes the set of points in the vector field after non-maximum suppression. Let $F(p) = F_1(p) \cup F_2(p)$, where $F_1(p)$, $F_2(p)$ are the lateral regions used for non-maximum suppression.

Proposition 2

$$P = P_F \cap P'_F \cap P_A \cap P'_A \tag{3.11}$$

where

$$\begin{aligned}
P_F &= \{p \in G \ : \ p' \in F(p) \implies |\phi(p)| > |\phi(p')|, \forall p' \in G\} \\
P'_F &= \{p' \in G \ : \ p' \in F(p) \implies |\phi(p')| > |\phi(p)|, \forall p \in G\} \\
P_A &= \left\{p \in G \ : \ p' \in A(p), ||\theta(p) - \theta(p')|| \geq \overline{\Theta} \implies |\phi(p)| > |\phi(p')|, \forall p' \in G\right\} \\
P'_A &= \left\{p' \in G \ : \ p' \in A(p), ||\theta(p) - \theta(p')|| \geq \overline{\Theta} \implies |\phi(p')| > |\phi(p)|, \forall p \in G\right\}
\end{aligned}$$

Notice that

- P_F is the set of points p whose intensity exceeds the intensity at lateral points $p' \in F(p)$
- P'_F is the set of points p' whose intensity exceeds the intensity at any point p such that $p' \in F(p)$
- P_A is the set of points p whose intensity always exceeds the intensity of points $p' \in A(p)$ with different orientation
- P'_A is the set of points p' whose intensity exceeds the intensity of any point p with different orientation and such that $p' \in A(p)$

Proposition 3 *After non-maximum suppression, the lateral regions of each point in the vector field are empty. That is, for any* $p \in P$

$$P \cap F(p) = \emptyset \tag{3.12}$$

Proof. By contradiction, let $p, p' \in P$, $p' \in F(p)$. By proposition 2 we have $p \in P_F$ and $p' \in P'_F$. From $p \in P_F$ it follows $|\phi(p)| > |\phi(p')|$. Similarly, from $p' \in P'_F$ it follows $|\phi(p')| > |\phi(p)|$. □

Proposition 4 *After non-maximum suppression, the variation of the field orientation inside* $A(p)$ *is less than* $\overline{\Theta}$. *That is, for any* $p, p' \in P$

$$p' \in A(p) \implies |\theta(p) - \theta(p')| < \overline{\Theta} \tag{3.13}$$

Proof. Let $p, p' \in P$, $p' \in A(p)$ and, by contradiction, $|\theta(p) - \theta(p')| \geq \overline{\Theta}$. By proposition 2 we have $p \in P_A$ and $p' \in P'_A$. Then, from $p \in P_A$ it follows $|\phi(p)| > |\phi(p')|$. Similarly, from $p' \in P'_A$ it follows $|\phi(p')| > |\phi(p)|$. □

3.4 The parameter h_1

Figure 2-16 on page 43 shows that a condition which should be imposed to guarantee that the interpolation formula (2.4) for $\lambda(p)$ is well defined is that the segments $\overline{o_1 o_2}$ and $\overline{o'_1 o'_2}$ do not intersect. For this to be the case, the distance h_0 between the region

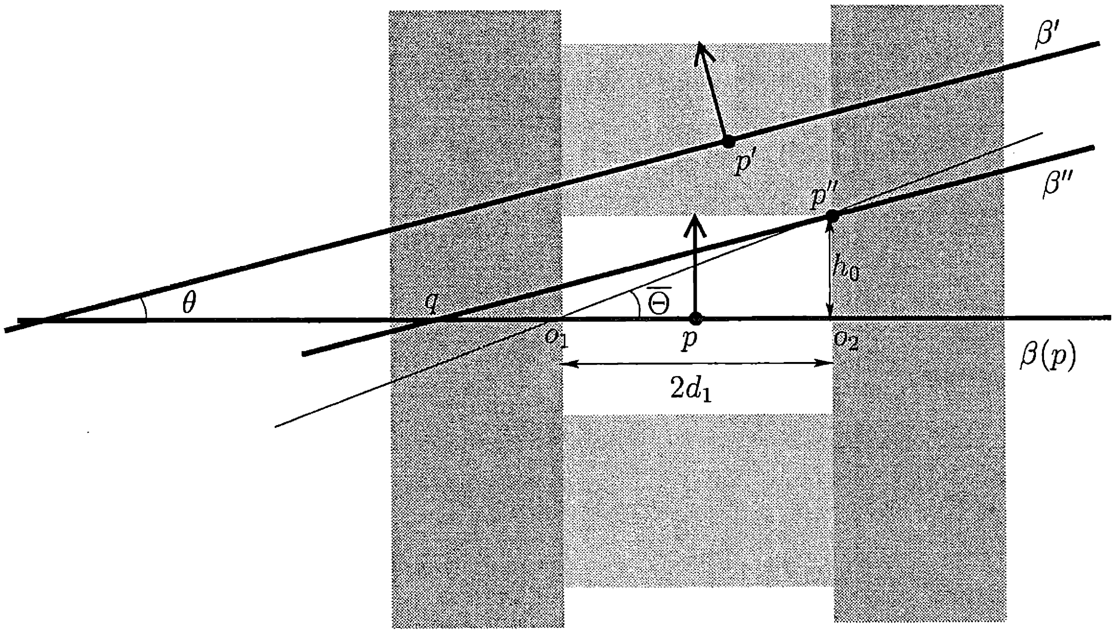


Figure 3-4: Proof of proposition 5

$S^+(p_i)$, to which p_{i+1}^+ belongs, and the line $\beta(p_i)$ has to be large enough. Proposition 5 guarantees that the value given in 2.2 works fine. Thus, since the height of $S^+(p_i)$ is a (see figure 2-14),

$$h_1 = h_0 + a = 2d_1 \tan \overline{\Theta} + a \tag{3.14}$$

Proposition 5 *Let $p, p' \in P$, $p' \in S^+(p)$ and $h_0 = 2d_1 \tan \overline{\Theta}$. Then,*

$$\overline{o_1 o_2} \cap \overline{o'_1 o'_2} = \emptyset$$

Proof. Refer to figure 3-4. The segment $\overline{o'_1 o'_2}$ lies on the straight line β'. Thus it is enough to prove that $\overline{o_1 o_2} \cap \beta' = \emptyset$ or that $\overline{o_1 o_2} \cap \beta'' = \emptyset$ where β'' is parallel to β' and contains the lower right corner of $S^+(p)$, p''. From proposition 4 it follows $\theta < \overline{\Theta}$ so that $||q - o_2|| > ||o_1 - o_2||$ and therefore β' does not intersect $\overline{o_1 o_2}$. □

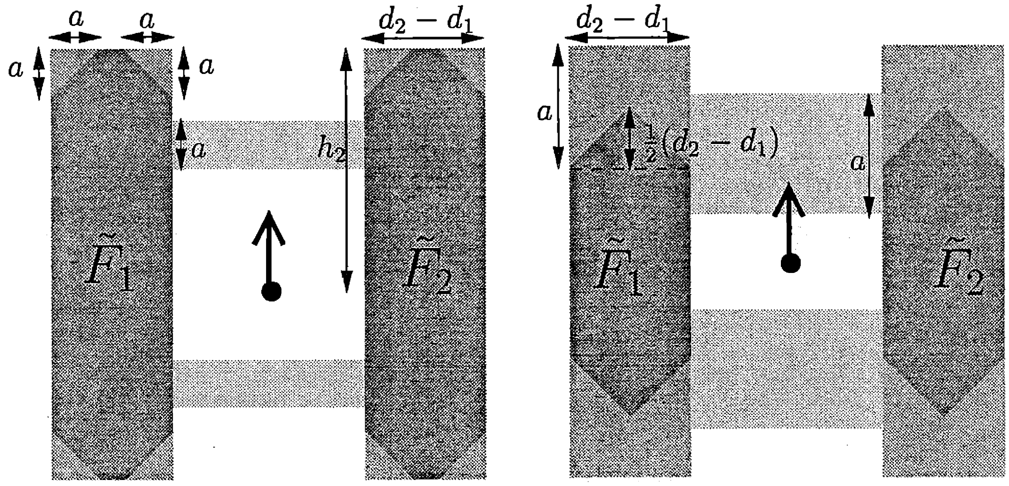
3.5 Proof that the domain of λ contains Q

3.5.1 The lateral regions $\tilde{F}_1, \tilde{F}_2$

Let $h_2 > a$ and $d_2 - d_1 > a$. Then define $\tilde{F}_1, \tilde{F}_2$ as in figure 3-5.

Proposition 6 *Let $h_2 > a$ and $d_2 - d_1 > a$. Fix $i \in \{1, 2\}$. Let p_1, p_2 be two points outside $F_i(p)$: $p_1 \notin F_i(p)$, $p_2 \notin F_i(p)$ such that $||p_2 - p_1|| \leq a$. Then the segment $\overline{p_1 p_2}$ does not intersect $\tilde{F}_i(p)$:*

$$\overline{p_1 p_2} \cap \tilde{F}_i(p) = \emptyset$$



(a) Case 1: $2a \leq d_2 - d_1$

(b) Case 2: $a \leq d_2 - d_1 \leq 2a$

Figure 3-5: The region $\tilde{F}_i, i = 1, 2$ is a subset of F_i which does not intersect any segment of length less than a with end-points outside F_i.

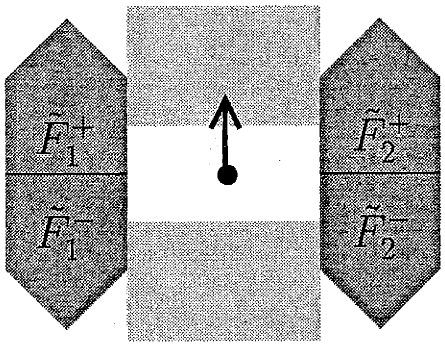


Figure 3-6: The top and bottom parts of the regions $\tilde{F}_1$, $\tilde{F}_2$.

3.5.2 Set of admissible parameters $\overline{\Theta}$, d_1, d_2, h_2, a

For any $p \in \mathbb{R}^2$, $\theta \in [0, 2\pi]$ let $S^+(p, \theta)$ be the search region of a vector with vertex p and orientation θ. Similarly for the other regions $S^-(p, \theta)$, $F_1(p, \theta)$, $F_2(p, \theta)$, $\tilde{F}_1(p, \theta)$, $\tilde{F}_2(p, \theta)$. As before, if the orientation θ does not appear in the notation, then the orientation of the vector field is assumed. That is, for instance, $S^+(p) = S^+(p, \theta(p))$ for any $p \in P$. Let $\tilde{F}_1^+$,$\tilde{F}_1^-$ denote the top and bottom parts of $\tilde{F}_1$. Similarly for $\tilde{F}_2^+$,$\tilde{F}_2^-$ (see figure 3-6).

Definition 2 *Let $\mathcal{A}$ be the set of 5-tuples $(\overline{\Theta}, d_1, d_2, h_2, a)$ such that the following hold:*

- $h_2 \geq h_1 = 2d_1 \tan \overline{\Theta} + a$
- $d_2 - d_1 > \max\{d_1, a\}$

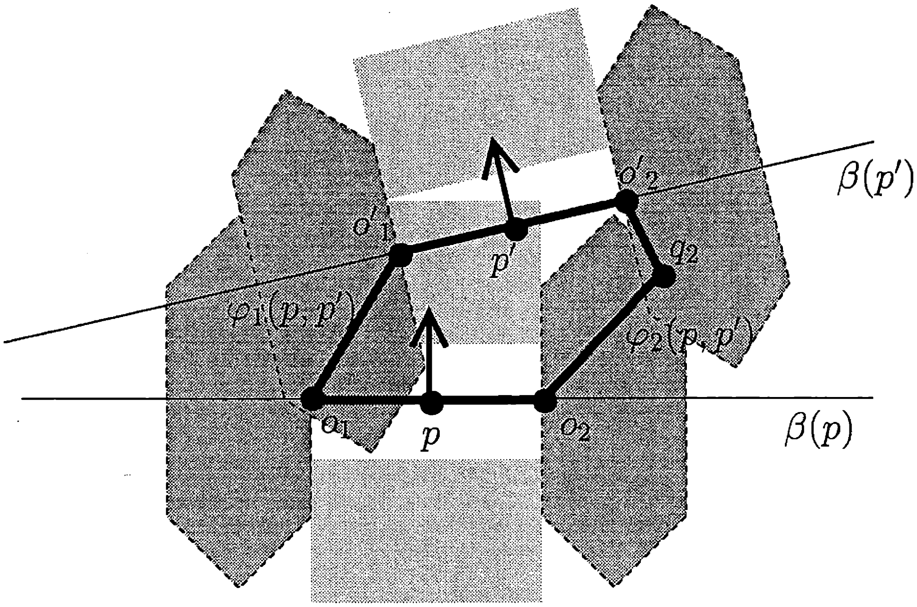


Figure 3-7: The closed curve $\Omega(p,p') = \overline{o_1 o_2} \cup \varphi_2(p,p') \cup \overline{o'_2 o'_1} \cup \varphi_1(p,p')$. In this example, $\varphi_1(p,p') = \overline{o_1 o'_1}$ (since $q_1 = o_1$) and $\varphi_2(p,p') = \overline{o_2 q_2 o'_2}$. The coordinate map λ is well defined inside $\Omega(p,p')$.

- *For any $p' \in S^+(p,\theta)$, and for any θ' such that $||\theta - \theta'|| \leq \overline{\Theta}$:*

$$\tilde{F}_1^+(p,\theta) \cap \tilde{F}_1^-(p',\theta') \neq \emptyset \tag{3.15}$$

$$\tilde{F}_1^+(p,\theta) \cap \tilde{F}_2^-(p',\theta') = \emptyset \tag{3.16}$$

Notice that (3.15) and (3.16) are equivalent to

$$\tilde{F}_2^+(p,\theta) \cap \tilde{F}_2^-(p',\theta') \neq \emptyset \tag{3.17}$$

$$\tilde{F}_2^+(p,\theta) \cap \tilde{F}_1^-(p',\theta') = \emptyset \tag{3.18}$$

Also notice that if (3.15) and (3.16) hold for a particular value of p, θ then they hold for all values of p, θ.

Proposition 7 *Let $(\overline{\Theta}, d_1, d_2, h_2, a) \in \mathcal{A}$, $p, p' \in P$, $p' \in S^+(p)$. Then, there exist two polygonal curves $\varphi_1(p,p')$, $\varphi_2(p,p')$ such that (see figure 3-7):*

(i) $\overline{o_1 o_2} \cup \varphi_2(p,p') \cup \overline{o'_2 o'_1} \cup \varphi_1(p,p')$ *is a closed curve which does not self-intersect. This curve will be denoted $\Omega(p,p')$.*

(ii) $\varphi_1(p,p') \subset \left(\tilde{F}_1^+(p) \cup \tilde{F}_1^-(p')\right)$

(iii) $\varphi_2(p,p') \subset \left(\tilde{F}_2^+(p) \cup \tilde{F}_2^-(p')\right)$

(iv) The longitudinal coordinate map λ is well defined and continuous in the interior of $\Omega(p,p')$.

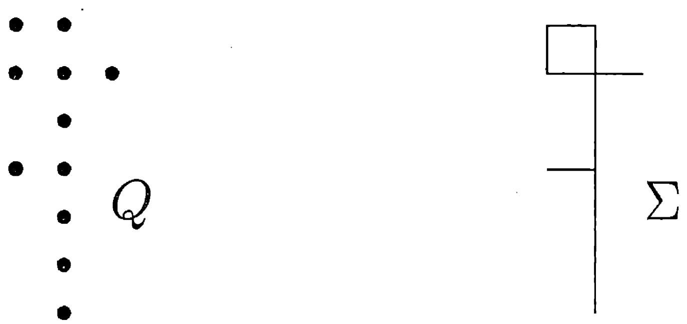


Figure 3-8: A connected component Q and the set Σ.

Proof. From proposition 4 and $S^+(p) \subset A(p)$ it follows $||\theta(p) - \theta(p')|| < \Theta$. Then, from (3.15) there exists a point $q_1 \in (\tilde{F}_1^+(p) \cap \tilde{F}_1^-(p'))$. Let then $\varphi_1(p,p') = \overline{o_1 q_1 o'_1}$, namely $\varphi_1(p,p')$ is the polygonal curve with vertices o_1, q_1, o'_1. Similarly, $\varphi_2(p,p') = \overline{o_2 q_2 o'_2}$ where $q_2 \in (\tilde{F}_2^+(p) \cap \tilde{F}_2^-(p'))$. Since the regions $\tilde{F}_1^+$,$\tilde{F}_2^+$,$\tilde{F}_1^-$,$\tilde{F}_2^1$ are convex, (ii) and (iii) follow.

Let us now prove (i). From proposition 5 it follows that $\overline{o_1 o_2} \cap \overline{o'_2 o'_1} = \emptyset$. From (3.16) and (3.18) we have $\varphi_1(p,p') \cap \varphi_2(p,p') = \emptyset$. Finally, $\varphi_i(p,p') \cap \overline{o_1 o_2} = \{o_i\}$ and $\varphi_i(p,p') \cap \overline{o'_1 o'_2} = \{o'_i\}$, $i = 1,2$ are true because the points q_1, q_2 lie between the two baselines $\beta(p)$, $\beta(p')$. Finally, point (*iv*) is clear from the interpolation formula (2.4) and (2.5). □

Let Σ be the union of all points which lie on a segment $\overline{p_1 p_2}$ where $p_1, p_2 \in Q$ and $||p_1 - p_2|| \leq a$ (see figure 3-8). Notice that $Q \subset \Sigma$. Then, the following proposition follows immediately from proposition 6 and parts (ii),(iii) of proposition 7.

Proposition 8 *Let* $(\overline{\Theta}, d_1, d_2, h_2, a) \in \mathcal{A}$, $p, p' \in Q$, $p' \in S^+(p)$. *Then the two polygonal curves* $\varphi_1(p,p')$, $\varphi_2(p,p')$ *do not intersect* Σ*:*

$$\varphi_1(p,p') \cap \Sigma = \emptyset$$
$$\varphi_2(p,p') \cap \Sigma = \emptyset$$

Let $s_1, \ldots, s_L$ be the sequence extracted by step 3 of the algorithm (previously denoted $s^-_{L^-}, \ldots, s^+_{L^+}$). Define Ω to be the closed, non self-intersecting curve shown in figure 3-9. Let D be the domain with boundary Ω. Clearly, λ is well defined on D.

Proposition 9 *Let* $(\overline{\Theta}, d_1, d_2, h_2, a) \in \mathcal{A}$. *Then* $\Sigma \subset D$. *Thus* λ *is well defined on* Σ *and* Q.

Proof. Since D contains some points of Σ (that is the sequence $s_1, \ldots, s_N$), it is enough to prove that Σ does not intersect the boundary of D, namely that $\Sigma \cap \Omega = \emptyset$.

That Σ does not intersect $\varphi_1(s_i, s_{i+1})$ nor $\varphi_2(s_i, s_{i+1})$ for any i is guaranteed by proposition 8. Also, Σ does not intersect $\varphi(s_N)$ nor $\varphi(s_1)$. In fact, $S^+(s_N) \cap Q = F_1(s_N) \cap Q = F_2(s_N) \cap Q = \emptyset$; the width of the regions $F_1(s_N)$, $F_2(s_N)$ and the height of $S^+(s_N)$ is larger than a. Therefore no segment $\overline{p_1 p_2}$ where $p_1, p_2 \in Q$ and $||p_1 - p_2|| = a$ can intersect $\varphi(s_N)$. □

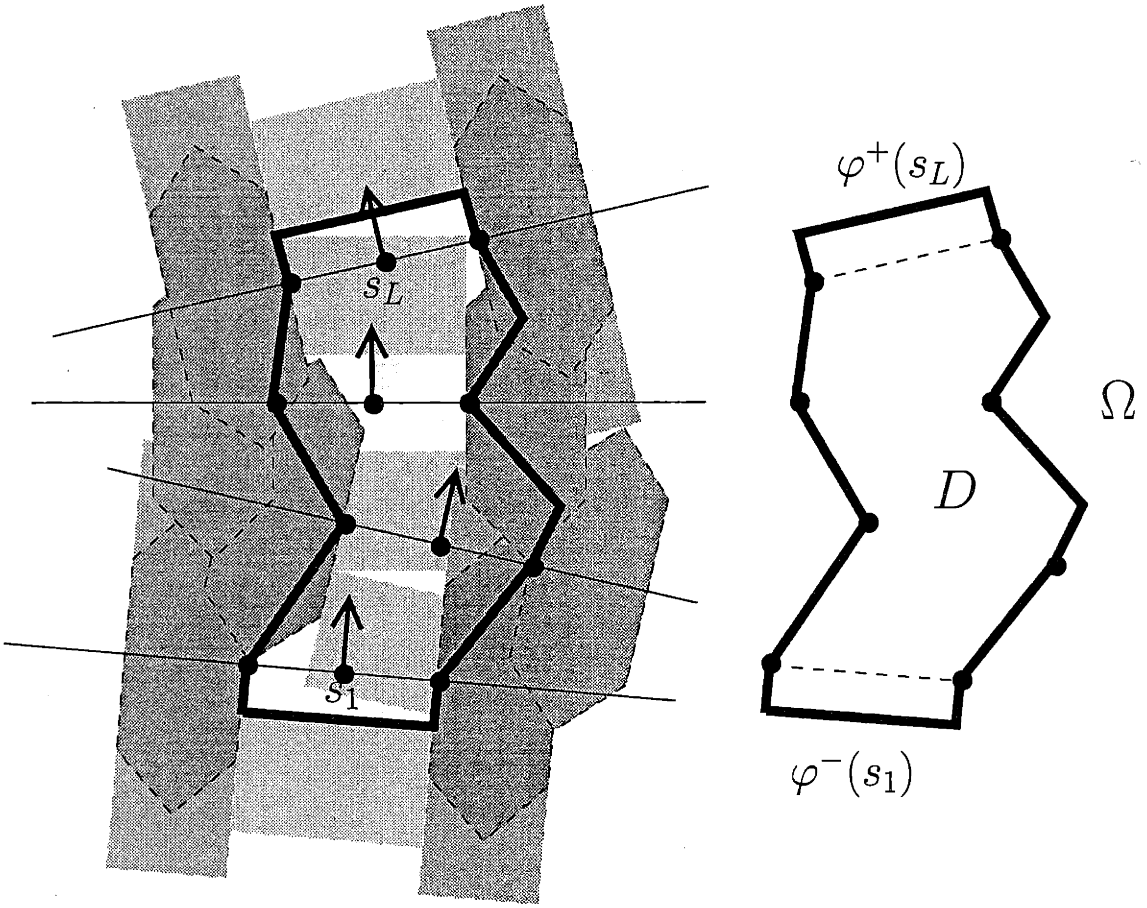


Figure 3-9: The domain D, on which λ is well defined, and its boundary Ω. The boundary of D, Ω is obtained by concatenating $\varphi_1(s_i, s_{i+1}), 1 \le i \le L$; $\varphi^+(s_L)$; $\varphi_2(s_i, s_{i+1})$; and $\varphi^-(s_1)$. The polygonal curves $\varphi^-(s_1)$ and $\varphi^+(s_L)$ contain 3 straight line segments.

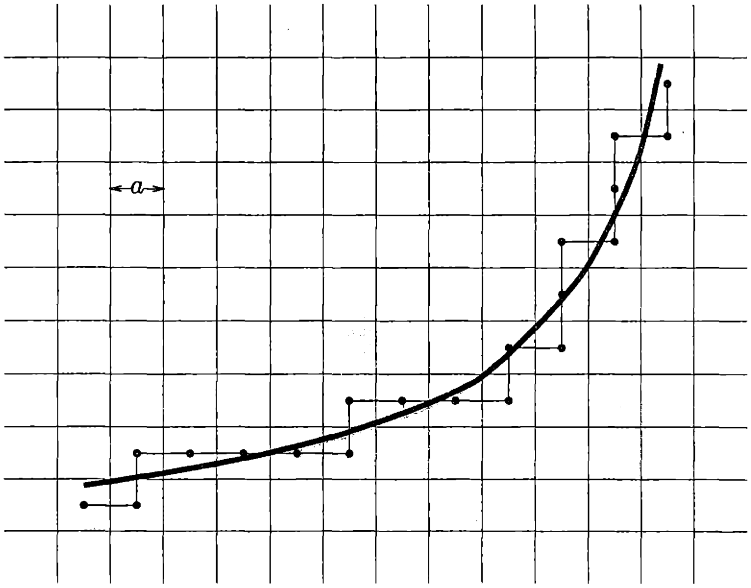


Figure 3-10: The discrete approximation of γ, Δ_γ.

3.6 Discrete approximation of a curve

Recall that the set of points G on which ϕ is defined forms a regular grid with sampling distance a. The discrete approximation of a curve γ, Δ_γ, is the maximal sequence of points $(p_1, \ldots, p_n)$ such that

$$||p_i - p_{i+1}|| = a$$

and

$$\gamma \cap \boldsymbol{S}_{p_i,a} \neq \emptyset$$

where $\boldsymbol{S}_{p_i,a}$ is the square region centered at p with side of size a (see figure 3-10).

Proposition 10 *The Hausdorff distance of γ from its discrete approximation $\Delta_\gamma = (p_1, \ldots, p_N)$ is no larger than $w_0 = a/\sqrt{2}$. That is, for every p_i in Δ_γ there is $p \in \gamma$ such that $||p_i - p|| \leq w_0$ and for every $p \in \gamma$ there is p_i in Δ_γ such that $||p_i - p|| \leq w_0$.*

3.7 Setting the parameters d_1, $\overline{\Theta}$

The parameter d_1 has to be large enough so that the points in the discrete approximation Δ_γ of a curve $\gamma \in C^\kappa_{w,\Theta}(\Phi)$ are never suppressed by part 1 of non-maximum suppression. Similarly, $\overline{\Theta}$ has to be large enough so that the points in Δ_γ are never suppressed by part 2 of non-maximum suppression. First, let us consider the case when γ is a straight line, that is when $\kappa = 0$.

Proposition 11 *Let $\gamma \in C^0_{w,\Theta}(\Phi)$ and let*

$$d_1 \cos\Theta \geq w + h_2 \sin\Theta + w_0 \tag{3.19}$$

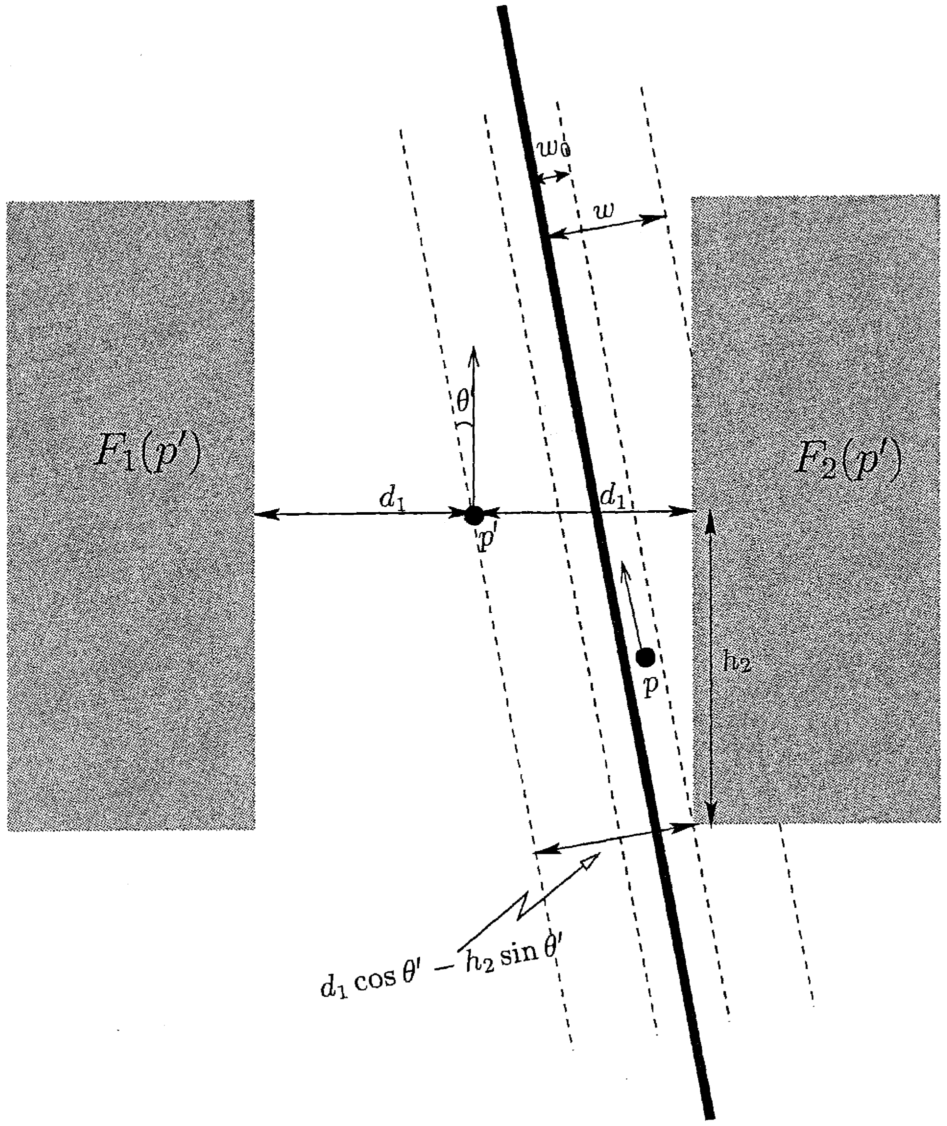


Figure 3-11: The only points p' where the field magnitude can be larger than the magnitude at p are within a distance w from γ. To make sure that p' does not suppress p, the parameter d_1 has to be large enough so that $p \notin F(p') = F_1(p') \cup F_2(p')$. Since θ' is at most Θ, then $d_1 \cos\Theta - h_2 \sin\Theta > w + w_0$ implies $p \notin F(p')$.

Then, a point p such that $d(p, \gamma) \leq w_0$ is not suppressed by part 1 of non-maximum suppression.

Proof. See figure 3-11. Let $p' \in G$. First, let us assume $d(p', \gamma) \leq w$. Then, from (3.2) it follows $\theta' < \Theta$ and, as shown in figure 3-11, $p \notin F(p')$. By a similar argument, $p' \notin F(p)$. Thus, if $d(p', \gamma) \leq w$, then p' does not suppress p.

Then, consider the case $d(p', \gamma) \geq w$. If $||p - p'||^2 > h_2^2 + d_2^2$ then, again, $p \notin F(p')$ and $p' \notin F(p)$. If, instead, $||p - p'||^2 \leq h_2^2 + d_2^2$ then from (3.1) it follows $|\phi(p)| > |\phi(p')|$ and therefore p' does not suppress p. □

Proposition 12 *Let $\gamma \in C^0_{w,\Theta}(\Phi)$ and let*

$$\overline{\Theta} \geq 2\Theta \tag{3.20}$$

Then, a point p such that $d(p, \gamma) \leq w_0$ is not suppressed by part 2 of non-maximum suppression.

Proof. [2] See figure 3-12. Let p be a point in Δ_γ and $p' \in G$. From proposition 10 we know $d(p, \gamma) < w_0 = a/\sqrt{2}$. From (3.2) it follows $|\theta| < \Theta$. First, let us assume $d(p', \gamma) \leq w$. Then, from (3.2) it follows $|\theta'| < \Theta$. Therefore $||\theta - \theta'|| < 2\Theta \leq \overline{\Theta}$, and therefore p is not suppressed by p'.

Let now $d(p', \gamma) > w$. Two cases are possible. If $||p - p'||^2 \geq h_1^2 + d_1^2$ then $p' \notin A(p)$, $p \notin A(p')$ and therefore p is not suppressed by p'. If instead $||p - p'||^2 \leq h_1^2 + d_1^2 < h_2^2 + d_2^2$ then from (3.1) it follows $|\phi(p)| > |\phi(p')|$ and therefore p' does not suppress p. □

3.8 Proof of theorem 1

To prove theorem 1 in the easiest way, we assume that, of the conditions in section 2.6, only (2.6) is used and the parameter r is set to zero. Thus, since the points in P belong to a regular grid with sampling distance a, any triple (p_1, p_2, p_3) such that $||p_1 - p_2|| = ||p_2 - p_3|| = a$ is allowed. This ensures immediately that every connected component Q contains a covering path, namely a sequence $p_1, \ldots, p_N$ such that $||p_i - p_{i+1}|| = a$ and

$$\lambda(p_1) = \min_{p \in Q} \lambda(p), \quad \lambda(p_N) = \max_{p \in Q} \lambda(p) \tag{3.21}$$

Let $\hat{\gamma}$ be the polygonal curve with vertices $p_1, \ldots, p_N$ where $p_1, \ldots, p_N$ is the path which minimizes the total curvature among all those which satisfy (3.21). Recall that this curve is computed by the last step of the algorithm.

[2] It should be clear from this proof that if we let Θ_0 be an upper bound to the orientation uncertainty within a distance w_0 from γ — namely $d(p, \gamma) \leq w_0 \Rightarrow ||\theta(p) - \theta_\gamma(p^*)|| < \Theta_0$; compare with (3.2) — then (3.20) can be replaced with the weaker condition $\overline{\Theta} \geq \Theta + \Theta_0$. Also, 3.3 and 3.5 can be changed accordingly.

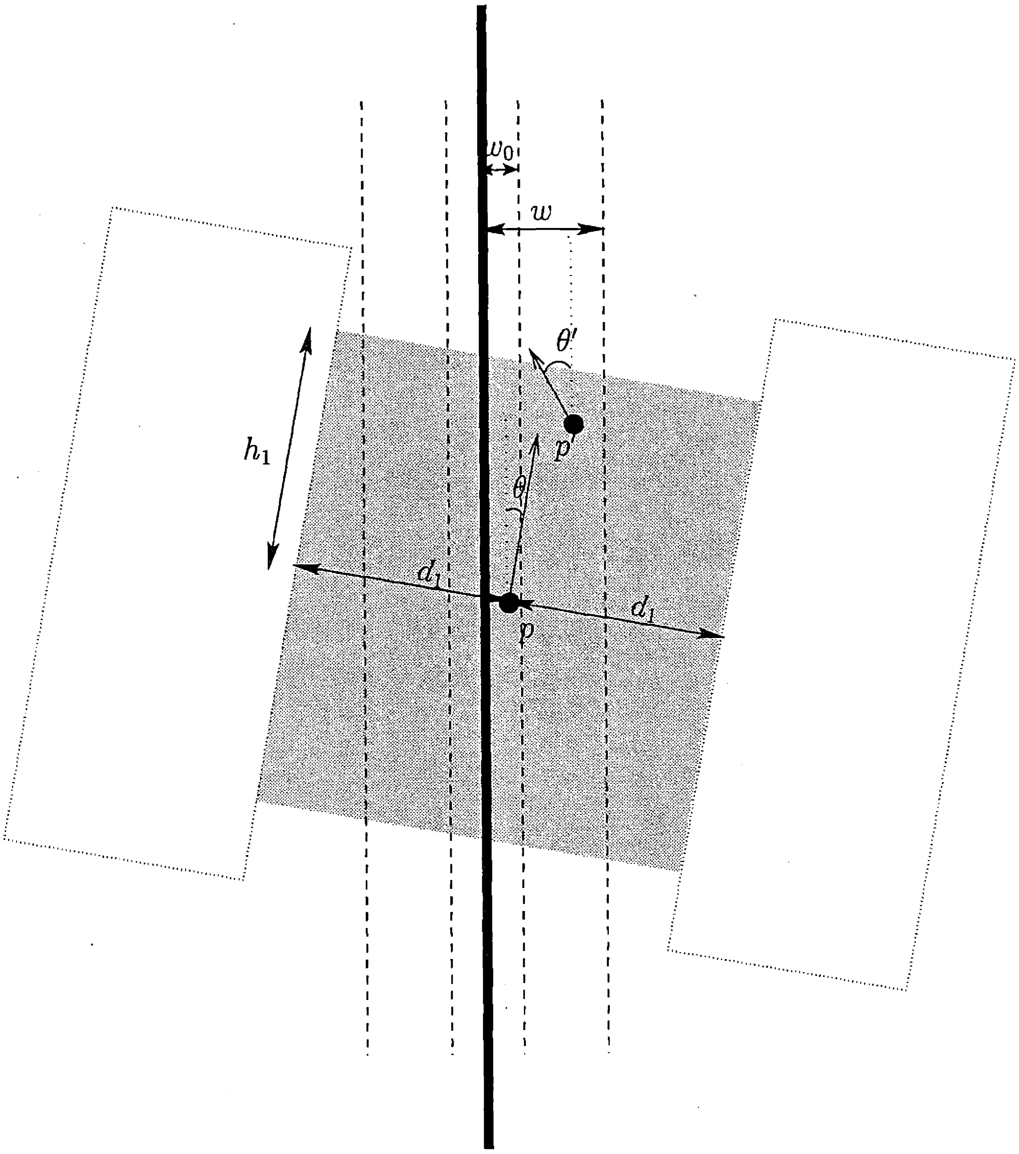


Figure 3-12: Proposition 12. To ensure that the point p, which belongs to the discrete approximation of γ, is not suppressed by $p' \in A(p)$ it is sufficient to prove that $|\phi(p)| > |\phi(p')|$ whenever $||\theta(p) - \theta(p')|| \geq \overline{\Theta}$.

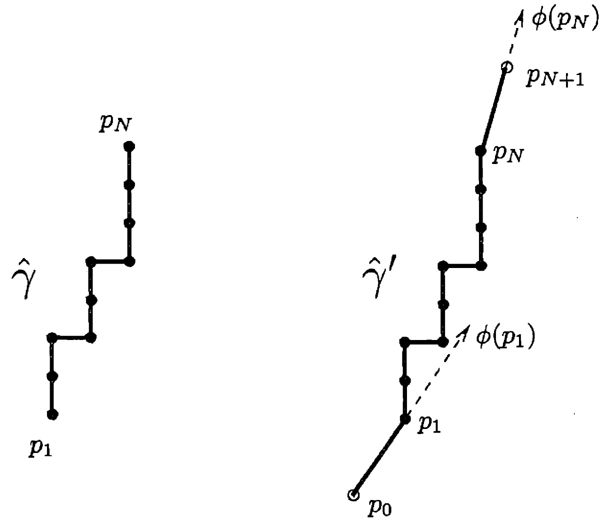


Figure 3-13: Left: The polygonal curve $\hat{\gamma}$ with vertices $p_1, \ldots, p_N$. Right: The polygonal curve $\hat{\gamma}'$ obtained by extending $\hat{\gamma}$ in the direction of the vector field. The length of $\overline{p_0p_1}$ and $\overline{p_Np_{N+1}}$ is d_1

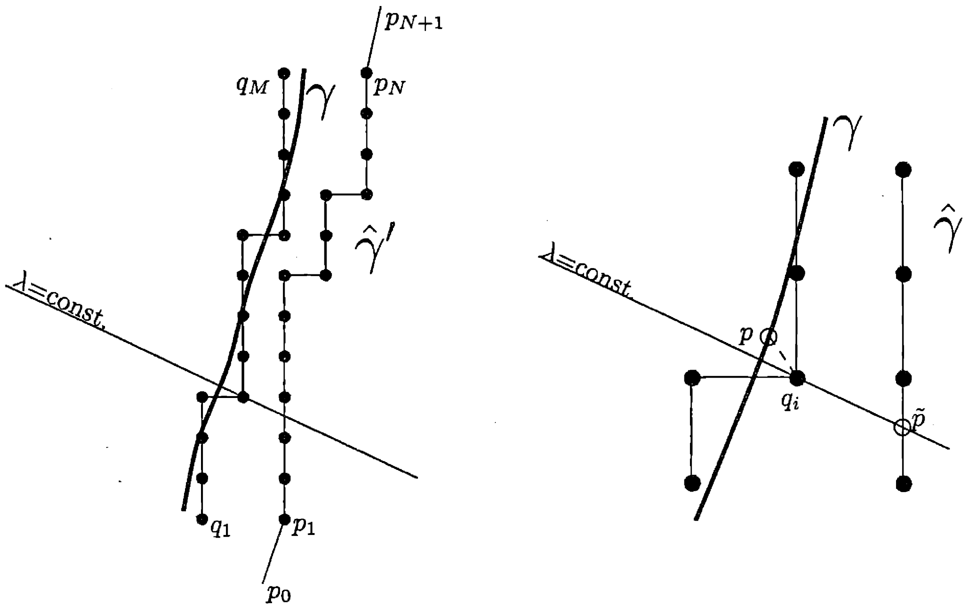


Figure 3-14: Left: A curve $\gamma \in C^{\kappa}_{w,\Theta}(\Phi)$ and its discrete approximation $q_1, \ldots, q_M$. Also shown is the computed curve $\hat{\gamma}'$. Right: For every $q_i \in \Delta_\gamma$ there exists $\tilde{p}$ such that $\lambda(q_i) = \lambda(\tilde{p})$.

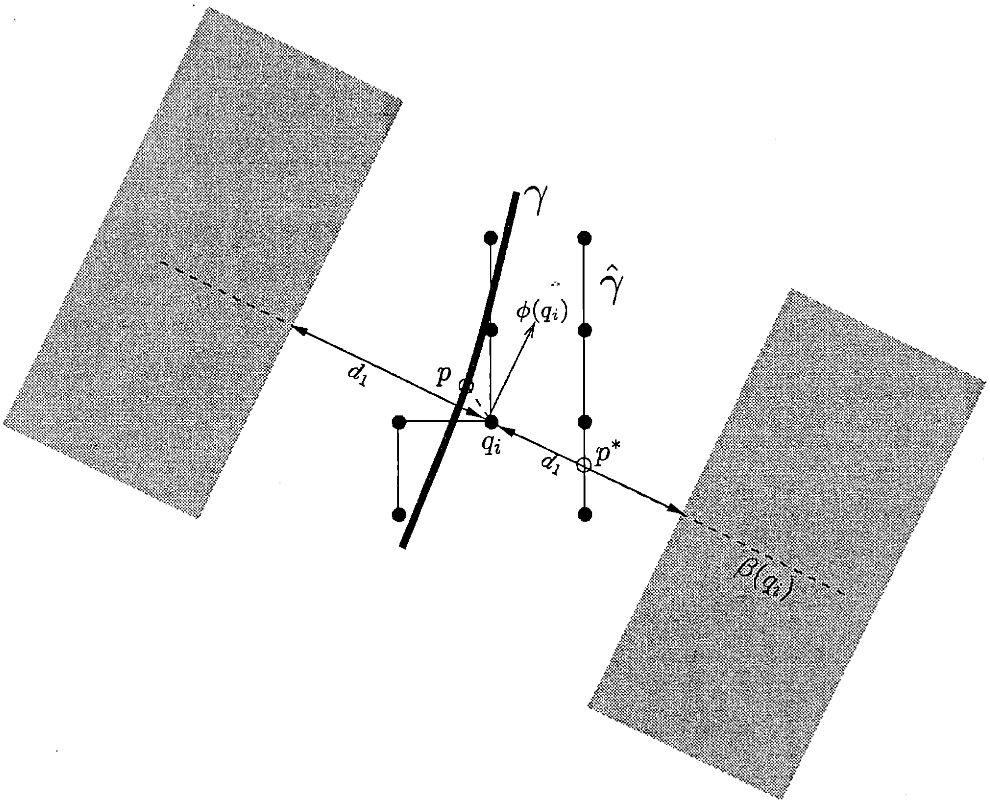


Figure 3-15: An upper bound for $d(q_i, \hat{\gamma})$. The baseline $\beta(q_i)$, perpendicular to the vector $\phi(q_i)$ intersects $\hat{\gamma}$ in p^*. Thus $||q_i - p^*|| \leq d_1$ and $d(q_i, \hat{\gamma}) \leq d_1$.

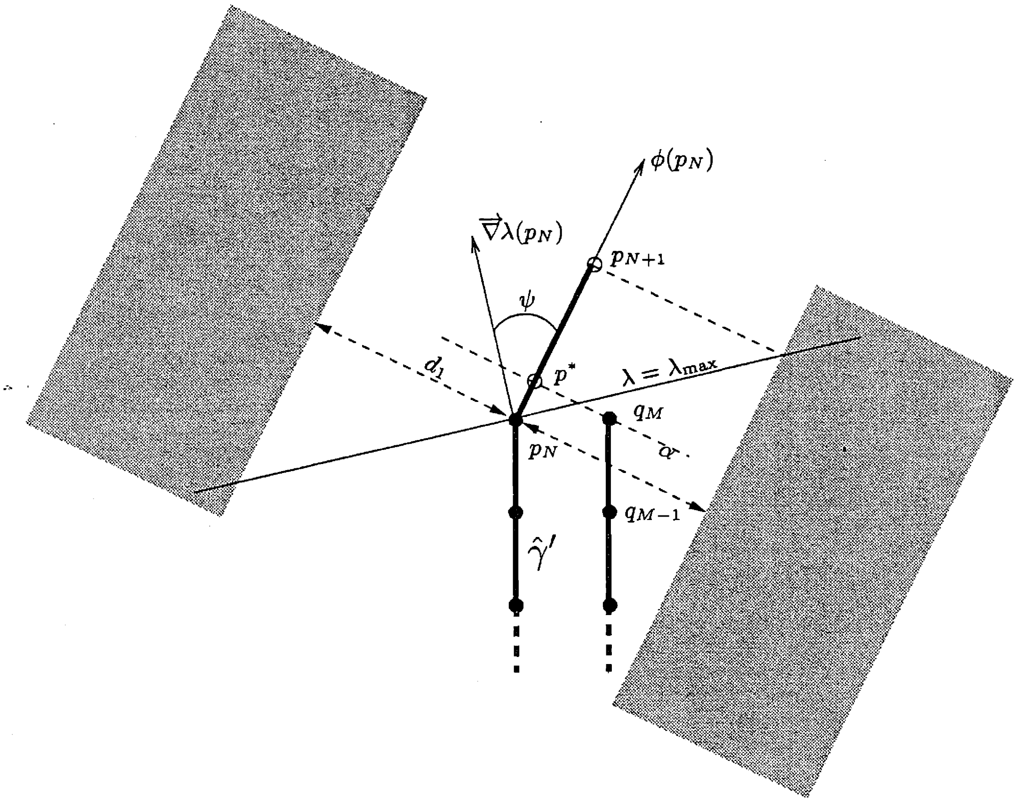


Figure 3-16: How long does the segment $\overline{p_N p_{N+1}}$ have to be to guarantee that the straight line α intersects it ? From (3.21) it follows that $\lambda(q_M) \leq \lambda_{\max}$ and therefore the point q_M lies below the level curve where λ is equal to $\lambda_{\max}$. Thus, from the geometry of the picture, $||p^* - p_N|| \leq ||p^* - q_M|| \tan\psi$. Let Θ_λ be an upper bound to the angle between the gradient of λ and the orientation of the vector field. Then $\psi \leq \Theta_\lambda$. Also, notice that $||p^* - q_M|| \leq d_1$ so that $||p^* - p_N|| \leq d_1 \tan\Theta_\lambda$. If we assume $\Theta_\lambda \leq \pi/4$ then $||p^* - p_N|| \leq d_1$ and $||p_{N+1} - p_N|| = d_1$ makes $\overline{p_N p_{N+1}}$ sufficiently long.

To ensure that the upper bound (3.10) on page 56 in theorem 1 holds also in the neighborhood of the two ends of the computed curve, it is necessary to add two short segments, $\overline{p_0 p_1}$ and $\overline{p_N p_{N+1}}$, to each end of $\hat{\gamma}$. The length of these segments is d_1. The resulting curve, $\hat{\gamma}'$ is shown in figure 3-13. The set of computed curves $\hat{\gamma}'$, one for each connected component, is denoted $\hat{C}(\Phi)$.

Proof of theorem 1 Let $\gamma \in C^{\kappa}_{w,\Theta}(\Phi)$ and let $\Delta_\gamma = (q_1, \dots, q_M)$ be its discrete approximation (see figure 3-14, left). From propositions 11 and 12, it follows that no point in Δ_γ is suppressed, thus $q_i \in P$, $i = 1, \dots, M$. Furthermore, since $||q_i - q_{i+1}|| = a$, there is a connected component Q which contains all the points q_i.

Let $p_1, \dots, p_N$ be the discrete curve produced by the algorithm from the component Q and let $\hat{\gamma}$ be the corresponding polygonal curve. Since $\hat{\gamma} \subset \Sigma$, it follows from proposition 9 that λ is defined on every point of $\hat{\gamma}$. Then, from (3.21) and the continuity of λ it follows that for every point q_i in Δ_γ there exists $\tilde{p} \in \hat{\gamma}$ such that $\lambda(q_i) = \lambda(\tilde{p})$ (see figure 3-14). If W denotes the *width* of the domain of λ, namely the maximum distance between two points having the same coordinate λ, then $||q_i - \tilde{p}|| \leq W$ and therefore

$$d(q_i, \hat{\gamma}) = \min_{p \in \hat{\gamma}} ||q_i - p|| \leq W, \quad i = 1, \dots, M$$

Let now p be a point in γ. Then, from proposition 10, there is a point q_i in Δ_γ such that

$$||q_i - p|| \leq w_0$$

where $w_0 = a/\sqrt{2}$ Thus, by the triangular inequality,

$$d(p, \hat{\gamma}) \leq W + w_0$$

and

$$d(\gamma, \hat{\gamma}) = \max_{p \in \gamma} d(p, \hat{\gamma}) \leq W + w_0$$

where $d(\gamma, \hat{\gamma})$ is the one-sided Hausdorff distance.

To obtain the stronger and more explicit bound (3.10), consider figure 3-15. If q_i is far enough away from q_1 and q_M then the straight line $\beta(q_i)$ intersects $\hat{\gamma}$ in p^* and

$$||q_i - p^*|| \leq d_1$$

On the other hand, if q_i is close to one of the two tips, then the straight line $\beta(q_i)$ is not guaranteed to intersect $\hat{\gamma}$. However, if the added segments $\overline{p_0 p_1}$, $\overline{p_N p_{N+1}}$ are long enough (see figure 3-16), then $\beta(q_i)$ intersects one of them. Thus,

$$d(q_i, \hat{\gamma}') = \min_{p \in \hat{\gamma}'} ||q_i - p|| \leq d_1, \quad i = 1, \dots, M$$

and, by the triangular inequality,

$$d(p, \hat{\gamma}') \leq d_1 + w_0$$

so that

$$d(\gamma, \hat{\gamma}) \leq d_1 + w_0$$

□

3.9 Numerical computation of $\overline{h}_2(\overline{\Theta}, a/d_1)$

To ensure that the coordinate map λ is defined on every point in Q and Σ, the parameters d_2 and h_2 have to be chosen so that (3.15) and (3.16) hold for every $p' \in S^+(p, \theta)$, and for any θ' such that $||\theta - \theta'|| \leq \overline{\Theta}$.

First of all let us rewrite (3.15) and (3.16) in a more convenient way. Let $\tilde{\mathbf{F}}_1^+ = \tilde{F}_1^+(\mathbf{0}, 0)$ where $\mathbf{0}$ denotes the origin of $\mathbb{R}^2$. Similarly for the other regions $\tilde{\mathbf{F}}_1^-$,$\tilde{\mathbf{F}}_2^+$,$\tilde{\mathbf{F}}_2^-$. Let T_p be the translation operator on subsets of $\mathbb{R}^2$:

$$T_p F = \{p' + p \;:\; p' \in F\}$$

Similarly, R_θ is the rotation operator:

$$R_\theta F = \{(x \cos\theta + y \sin\theta, y \cos\theta - x \sin\theta) \;:\; (x, y) \in F\}$$

Then, (3.15) and (3.16) can be rewritten as,

$$T_p R_\theta \tilde{\mathbf{F}}_1^- \cap \tilde{\mathbf{F}}_1^+ \neq \emptyset, \quad p \in \mathbf{S}^+, \; |\theta| \leq \overline{\Theta} \tag{3.22}$$

$$T_p R_\theta \tilde{\mathbf{F}}_1^- \cap \tilde{\mathbf{F}}_2^+ = \emptyset, \quad p \in \mathbf{S}^+, \; |\theta| \leq \overline{\Theta} \tag{3.23}$$

Since all regions involved are convex, it is enough to check (3.22) and (3.23) only for $p \in \mathbf{V}^+$, where $\mathbf{V}^+ = \{u_1, u_2, l_1, l_2\}$ is the set of vertices of $\mathbf{S}^+$ (see figure 3-17).

Let us start by considering the case when $\overline{\Theta} = 0$. Let $\mathcal{A}_0$ be the set of parameters (d_1, d_2, h_1, h_2, a) for which (3.22) and (3.23) hold with $\overline{\Theta} = 0$, that is

$$T_p \tilde{\mathbf{F}}_1^- \cap \tilde{\mathbf{F}}_1^+ \neq \emptyset, \quad T_p \tilde{\mathbf{F}}_1^- \cap \tilde{\mathbf{F}}_2^+ = \emptyset, \qquad p \in \mathbf{V}^+$$

It's easy to verify that two sufficient conditions for (d_1, d_2, h_1, h_2, a) to be in $\mathcal{A}_0$ are:

$$d_2 \geq 2d_1, \tag{3.24}$$

$$2(h_2 - a) \geq h_1 \tag{3.25}$$

For $(d_1, d_2, , h_1 h_2, a) \in \mathcal{A}_0$, let $\overline{\Theta}_{\max}(d_1, d_2, h_1, h_2, a)$ be the maximum value for which (3.22) and (3.23) hold:

$$\overline{\Theta}_{\max}(d_1, d_2, h_1, h_2, a) =$$
$$\max\left\{\theta \geq 0 \;:\; T_p R_\theta \tilde{\mathbf{F}}_1^- \cap \tilde{\mathbf{F}}_1^+ \neq \emptyset, \;\; T_p R_\theta \tilde{\mathbf{F}}_1^- \cap \tilde{\mathbf{F}}_2^+ = \emptyset; \;\; p \in \mathbf{V}^+\right\} \tag{3.26}$$

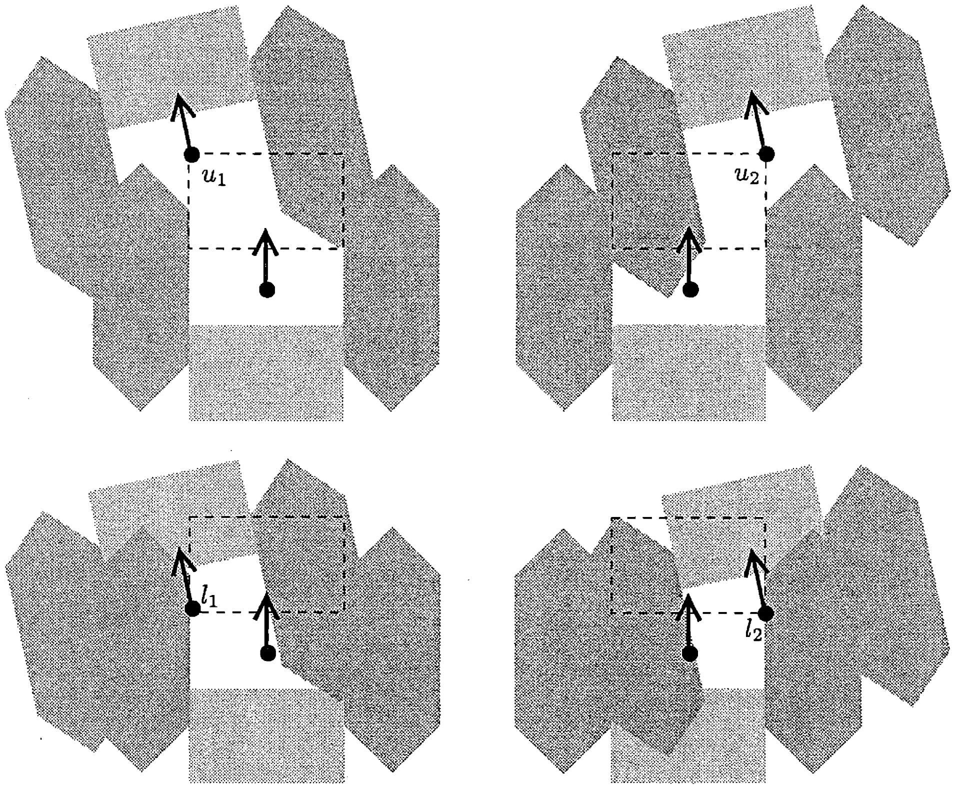


Figure 3-17: The parameters d_2, h_2 have to be chosen so that each lateral region intersects the corresponding lateral region for every possible vector in $\mathbf{S}^+$ (see (3.22) and (3.23)).

A property of this function is that it is invariant under scale transformations:

$$\overline{\Theta}_{\max}(\alpha d_1, \alpha d_2, \alpha h_1, \alpha h_2, \alpha a) = \overline{\Theta}_{\max}(d_1, d_2, h_1, h_2, a) \tag{3.27}$$

The function $\overline{\Theta}_{\max}(d_1, d_2, h_1, h_2, a)$ can be computed numerically. Figures 3-18 and 3-19 show some sections of this function.

Then let

$$h_2^{\min}(d_1, d_2, h_1, \overline{\Theta}, a) = \min\left\{h_2 \ : \ \overline{\Theta}(d_1, d_2, h_1, h_2, a) \geq \overline{\Theta}\right\} \tag{3.28}$$

and

$$\tilde{h}_2^{\min}(d_1, d_2, \overline{\Theta}, a) = h_2^{\min}(d_1, d_2, 2d_1 \tan\overline{\Theta}, \overline{\Theta}, a) \tag{3.29}$$

Because of (3.27), $\tilde{h}_2^{\min}(d_1, d_2, \overline{\Theta}, a)$ can be rewritten as

$$\tilde{h}_2^{\min}(d_1, d_2, \overline{\Theta}, a) = d_1 \ \overline{h}_2^{\min}\left(\overline{\Theta}, \frac{a}{d_1}, \frac{d_2}{d_1}\right) \tag{3.30}$$

Figure 3-20 shows several graphs of h_2 as a function of $\overline{\Theta}$. The parameter d_2 is chosen so that the area of the lateral regions $F_i(p)$, $2h_2(d_2 - d_1)$, is minimum. Thus, only two free parameters, $\overline{\Theta}$, a remain and h_2 can be written as:

$$h_2 = d_1 \ \overline{h}_2\left(\overline{\Theta}, \frac{a}{d_1}\right) \tag{3.31}$$

Notice that (3.5)-(3.9) on page 55 do not express explicitely the algorithm parameters d_1, d_2, h_2 in terms of the model parameters w, Θ, a. In fact, the parameter h_2 appears on the right hand side of (3.6) and d_1 appears on the right hand side of (3.9). However, $w' \equiv w + w_0, d_2, h_2$ can be explicitely written in terms of d_1, Θ, a. In fact, from (3.9) it follows that h_2 can be obtained from d_1, a and from (3.6) we have (in the case $\kappa = 0$):

$$w' = d_1 \cos\Theta - h_2 \sin\Theta \tag{3.32}$$

Table 3.1 gives the values of $\overline{\Theta}, d_2, h_2, w'$ for $d_1 = 1$ and $a = 0.2, 0.5, 1.0$. Then, by using this table and substituting $w' = w + w_0 = w + a/\sqrt{2}$, it is possible to derive d_1, d_2, h_2 explicitely from w, Θ, a (table 3.2(a)). Finally, table 3.2(b) shows the parameter of table 3.2(a) normalized by w. Recall that multiplying all the length parameters (namely w, a, d_1, d_2, h_2) by a constant yields another valid combination of parameters.

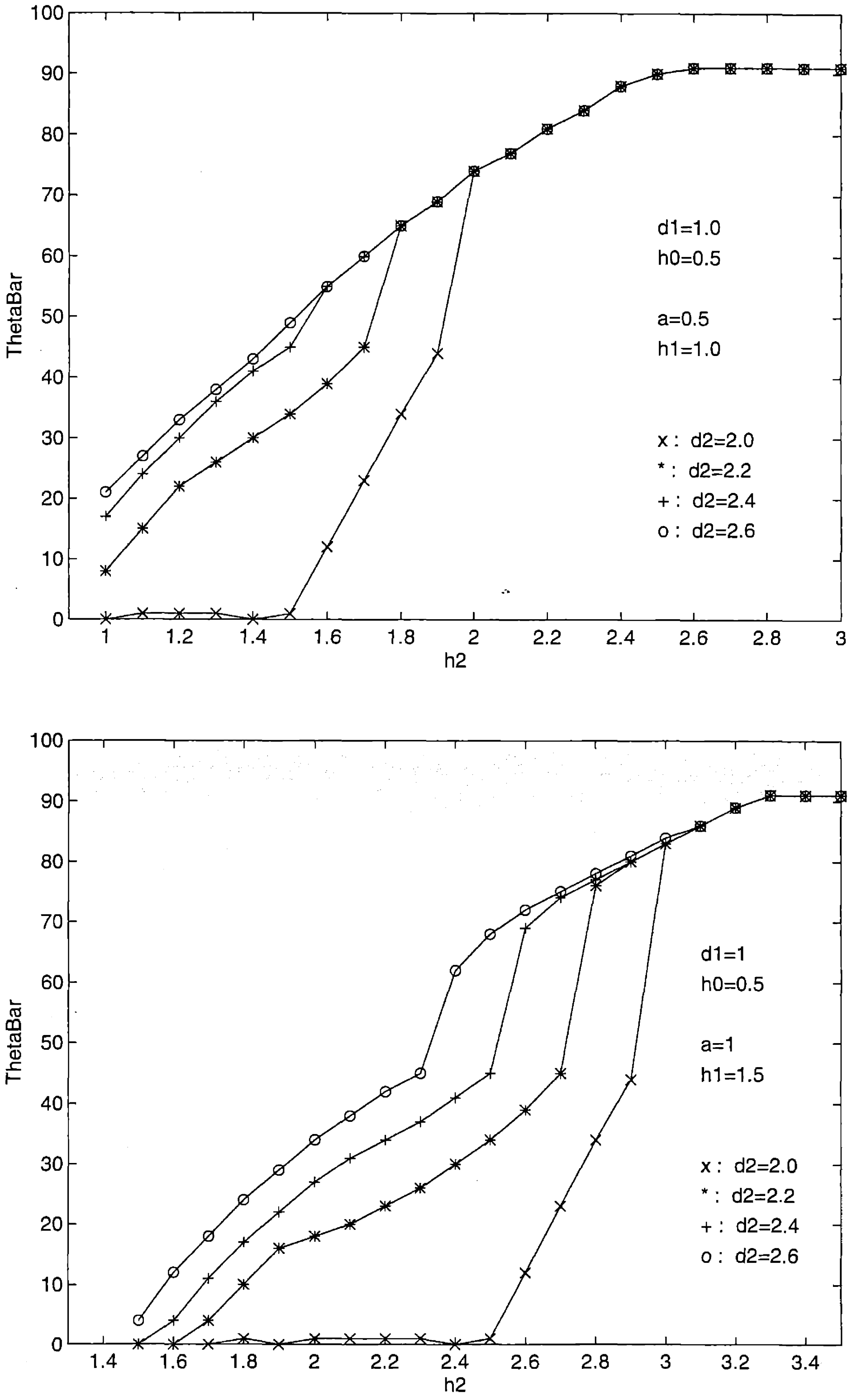


Figure 3-18: $\overline{\Theta}$ versus h_2 for four values of d_2. For every graph: $d_1 = 1.0$, $h_0 = 0.5$. Top: $a = 0.5$. Bottom $a = 1.0$.

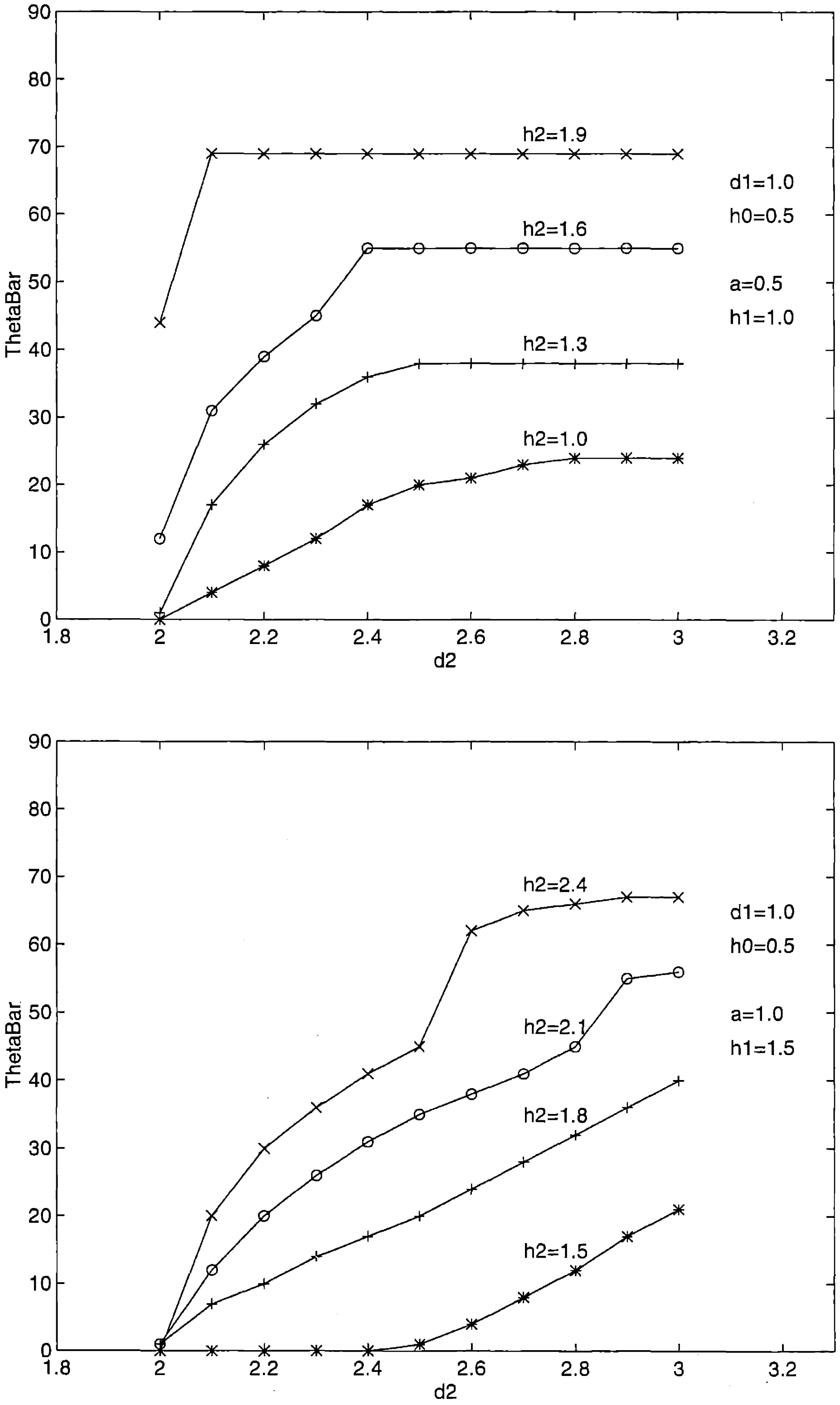
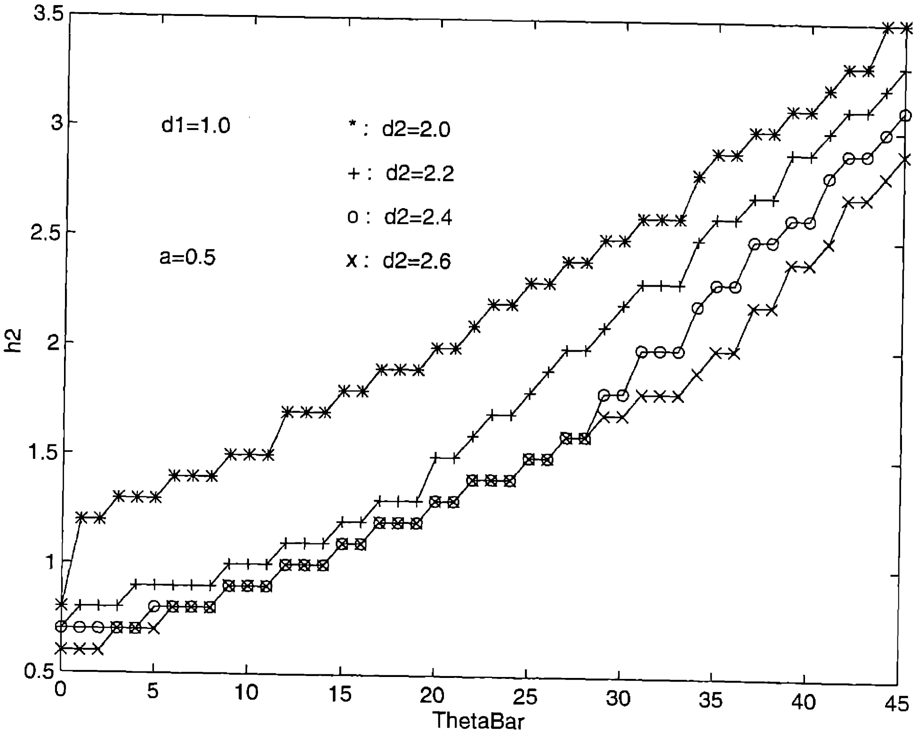


Figure 3-19: $\overline{\Theta}$ versus d_2 for four values of h_2. For every graph: $d_1 = 1.0$, $h_0 = 0.5$. Top: $a = 0.5$. Bottom $a = 1.0$.



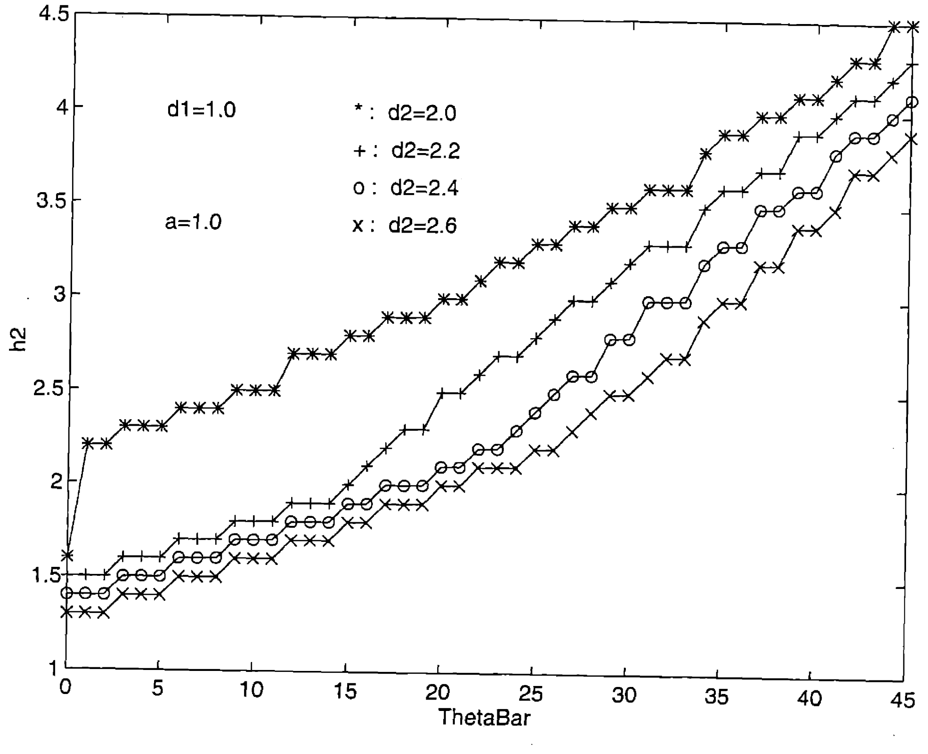


Figure 3-20: The function $\tilde{h}_2^{\min}(d_1; d_2, \overline{\Theta}, a)$.

		a=0.2			a=0.5			a=1.0		
$\overline{\Theta}$	Θ	d_2	h_2	w'	d_2	h_2	w'	d_2	h_2	w'
0.0	0.0	2.0	.4	1.0	2.0	.8	1.0	2.0	1.6	1.0
5.0	2.5	2.2	.4	.98	2.2	.9	.96	2.1	1.7	.92
10.0	5.0	2.2	.6	.94	2.2	.9	.92	2.2	1.8	.84
15.0	7.5	2.2	.8	.89	2.2	1.2	.83	2.3	1.9	.74
20.0	10.0	2.2	1.0	.81	2.4	1.3	.76	2.4	2.1	.62
25.0	12.5	2.2	1.2	.71	2.4	1.5	.65	2.5	2.3	.48
30.0	15.0	2.3	1.5	.58	2.4	1.8	.50	2.6	2.5	.32
35.0	17.5	2.4	1.7	.44	2.6	2.0	.35	2.8	2.7	.14
40.0	20.0	2.4	2.0	.25	2.6	2.4	.12	3.0	2.9	-.05

Table 3.1: The parameters $d_2, h_2, w' = w + w_0$ as a function of Θ for $d_1 = 1$, $a = 0.2, 0.5, 1.0$. Notice that $\overline{\Theta} = 2\Theta$. d_2, h_2 have been obtained from the graphs in figure 3-20 by choosing the value of d_2 for which $h_2(d_2 - d_1)$ is minimum. w' is given by (3.6): $w' = d_1 \cos\Theta - h_2 \sin\Theta$.

Θ	w	d_1	d_2	h_2
0	.86	1	2.0	.4
5	.80	1	2.2	.6
10	.67	1	2.2	1.0
15	.44	1	2.3	1.5

(a) This table is obtained from table 3.1, $a = 0.2$, with $w_0 = 2^{-\frac{1}{2}}a = 0.14$.

Θ	a/w	d_1/w	d_2/w	h_2/w
0	.23	1.16	2.32	.46
5	.25	1.25	2.75	.75
10	.30	1.49	3.28	1.49
15	.45	2.27	5.22	3.40

(b) Admissible values of the algorithm parameters normalized to w.

Table 3.2: Some admissible values for the algorithm parameters.

3.10 Generalization to arbitrarily sampled vector fields

So far, the domain G of the input vector field was assumed to be a regular grid with sampling distance a. The constant w_0 was then given by $a/\sqrt{2}$. Theorem 1 on page 56 can be generalized to the case where G is an arbitrary discrete set, provided that the constants a, w_0 and the family of curves $C^{\kappa}_{w,\Theta}(\Phi)$ are appropriately redefined.

First, let us introduce some notation. Let γ be a smooth curve with length L and let p_0 be one of its two end points. Then, for any $p \in \gamma$, let $l(p)$ be the length of the curve from p_0 to p, also called *arc-length parameter* (see figure 3-21). For any $l \in [0, L]$ let S_l be the segment with length $2w_0$ perpendicular to γ which intersects it symmetrically at the point with arc-length parameter l (see figure 3-21).

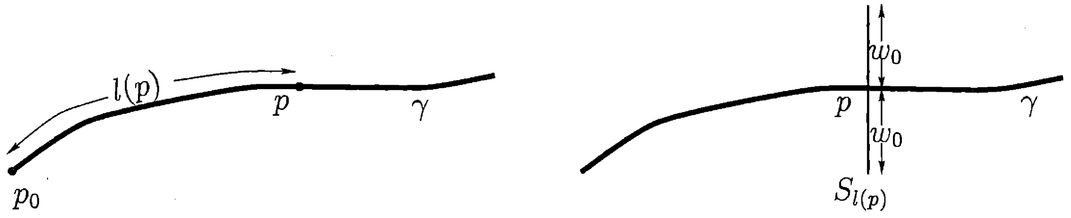


Figure 3-21: Left: the arc-length parameter $l(p)$ is the length of the curve from p_0 to p. Right: the segment S_l has length $2w_0$ and intersects perpendicularly the curve at the point p with arc-length parameter l.

Definition 3 *Let $C^{\kappa}(\Phi; w, \Theta, w_0, a)$ be the set of finite curves with curvature less than κ such that:*

(C1) Condition (C1) on page 54 holds. Namely, for any $p, p' \in G$ (see figure 3-2 on page 54):

$$\left.\begin{array}{rcl} d(p,\gamma) & \leq & w_0 \\ d(p',\gamma) & \geq & w \\ ||p-p'||^2 & \leq & d_2^2+h_2^2 \end{array}\right\} \quad \Rightarrow \quad |\phi(p)| > |\phi(p')| \tag{3.33}$$

(C2) Condition (C2) on page 54 holds. Namely, for any $p \in G$ (see figure 3-3 on page 55):

$$d(p,\gamma) \leq w \quad \Rightarrow \quad ||\theta(p) - \theta_{\gamma}(p^*)|| < \Theta \tag{3.34}$$

(C3) The segments S_l, $0 \leq l \leq L$, are all pairwise disjoint:

$$l_1 \neq l_2 \implies S_{l_1} \cap S_{l_2} = \emptyset \tag{3.35}$$

(C4) The segments S_0, S_L contain a point in G:

$$S_0 \cap G \neq \emptyset, \qquad S_L \cap G \neq \emptyset \tag{3.36}$$

(C5) Let $p \in S_l \cap G$, $l \neq L$. Then there exists $p' \in G$ such that:

$$||p - p'|| \leq a \tag{3.37}$$
$$p' \in S_{l'}, \qquad l' > l \tag{3.38}$$

Condition (C3) is locally equivalent to requiring that the curvature of γ be less than $1/w_0$. Condition (C4) and (C5) replace the assumption that G is a uniform grid with sampling distance a. (C3) through (C5) are used only to prove proposition 13. Thus they could be replaced by the condition that elements in $\gamma \in C^\kappa(\Phi; w, \Theta, w_0, a)$ have the property described in proposition 13.

Proposition 13 *If $\gamma \in C^\kappa(\Phi; w, \Theta, w_0, a)$ then there exists a sequence of points $\Delta_\gamma = \{q_0, \dots, q_M\} \subset G$ such that*

$$||q_i - q_{i+1}|| \leq a \tag{3.39}$$
$$d(q_i, \gamma) \leq w_0 \tag{3.40}$$
$$d(\gamma, \Delta_\gamma) \leq \delta(\kappa, w_0, a) \tag{3.41}$$

where

$$d(\gamma, \Delta_\gamma) = \max_{p \in \gamma} \min_{0 \leq i \leq M} ||p - q_i||$$

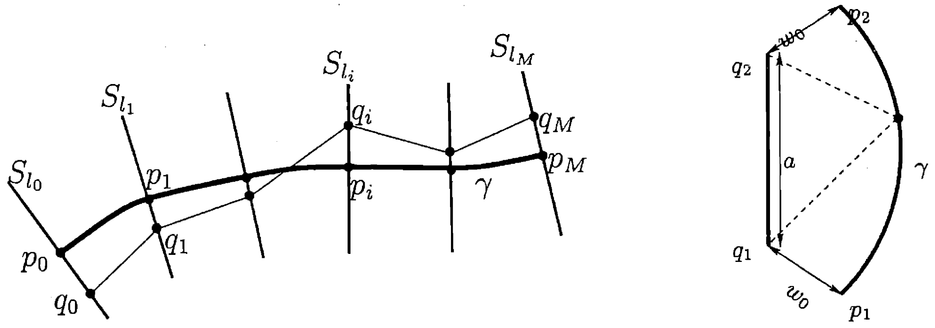
and $\delta(\kappa, w_0, a)$ is given by definition 4.

Definition 4 *Let $q_1, q_2 \in \mathbb{R}^2$ be such that $||q_1 - q_2|| \leq a$ (see figure 3-22(b)). For $i = 1, 2$, let $B(q_i)$ be the ball in $\mathbb{R}^2$ with radius w_0 and center q_i. Then $\delta(\kappa, w_0, a)$ is defined as:*

$$\delta(\kappa, w_0, a) = \sup_{p_1 \in B(q_1), p_2 \in B(q_2)} \; \sup_{\gamma \in C^\kappa(p_1, p_2)} \; \max_{p \in \gamma} \min \{||p - q_1||, ||p - q_2||\} \tag{3.42}$$

where $C^\kappa(p_1, p_2)$ is the set of all curves with end-points p_1, p_2; curvature less than κ; and such that the segments $\overline{p_1 q_1}$, and $\overline{p_2 q_2}$ are perpendicular to the curve tangents at its end-points.

Notice that $\delta(0, w_0, a) \leq w_0 + a/2$.
Proof of proposition 13. From (C4) it follows that there is $q_0 \in S_0 \cap G$. Then, by using (C5) iteratively starting from q_0, a sequence $q_0, q_1, \dots$ can be constructed such that (3.39) holds and $q_i \in S_{l_i}$ for some l_i. From (C3) it follows that the l_i for which $q_i \in S_{l_i}$ is unique. Thus (3.38) implies $l_{i+1} > l_i$ for every i and since G is finite, the sequence can be prolonged until $l_M = L$ for some $M > 0$.



(a) Existence of sequence $q_0, \dots, q_M$.

(b) Definition of $\delta(\kappa, w_0, a)$

Figure 3-22: Proof of proposition 13.

Let p_i be the intersection point between S_{l_i} and γ. Since $q_i \in S_{l_i}$, we have $||q_i - p_i|| \leq w_0$ and therefore $d(q_i, \gamma) \leq w_0$, which proves (3.40).

Finally, let us decompose γ into M pieces γ_i with end-points p_i, p_{i+1}.Then, from definition 4,

$$\min\{d(\gamma_i, q_i), d(\gamma_i, q_{i+1})\} \leq \delta(\kappa, w_0, a)$$

and therefore $d(\gamma, \Delta_\gamma) \leq \delta(\kappa, w_0, a)$. □

Let us assume as in section 3.8 that only the constraint (2.6) is used in step 4 of the algorithm (section 2.6) and that $r = 0$. Then, the following result can be proved in a similar way as theorem 1 (compare with section 3.8).

Theorem 2 *Let Φ be an arbitrary set of vectors. Let $\hat{C}(\Phi)$ be the set of polygonal curves generated by the algorithm described in chapter 2 on input Φ. Let the parameters $\overline{\Theta}$, d_1, d_2, h_1, h_2 satisfy (3.3)–(3.9) on page 55. Then, for every $\gamma \in C^\kappa(\Phi; w, \Theta, w_0, a)$ there exists $\hat{\gamma}' \in \hat{C}(\Phi)$ such that*

$$d(\gamma, \hat{\gamma}') \leq d_1 + \delta(\kappa, w_0, a) \tag{3.43}$$

Proof. From proposition 13, there exists a sequence of points $\Delta_\gamma = \{q_0, \dots, q_M\} \subset G$ such that (3.39)–(3.41) hold. From (3.40), conditions (C1), (C2) and propositions 11 and 12, it follows that the points $q_0, \dots, q_M$ are not suppressed and therefore belong to P. Furthermore, (3.39) ensures that they all belong to the same connected components Q.

Let $\hat{\gamma}$ be the polygonal curve computed by the algorithm from Q. As explained in section 3.8, its vertices satisfy (3.21). By an argument similar to the one in section 3.8, we have that $d(q_i, \hat{\gamma}') \leq d_1$ (see figures 3-15, 3-16) and therefore $d(\Delta_\gamma, \hat{\gamma}') \leq d_1$.

Θ	w	d_1	d_2	h_2
0	.9	1	2.0	.8
5	.82	1	2.2	.9
10	.66	1	2.4	1.3
15	.40	1	2.4	1.8

(a) The parameters d_2, h_2 as a function of w, Θ obtained with $d_1 = 1$, $w_0 = 0.1$, $a = 0.5$.

Θ	a/w	w_0/w	d_1/w	d_2/w	h_2/w
0	.56	.11	1.11	2.22	1.89
5	.61	.12	1.22	2.68	1.1
10	.75	.15	1.51	3.62	1.96
15	1.25	.25	2.5	6.0	4.5

(b) The parameters w_0, a, d_1, d_2, h_2 normalized by w as a function of Θ obtained from the left table.

Table 3.3: Some admissible settings of the parameters. $\overline{\Theta} = 2\Theta$.

Then, by using (3.41) and the triangular inequality:

$$\begin{aligned} d(\gamma, \hat{\gamma}') &\leq d(\gamma, \Delta_\gamma) + d(\Delta_\gamma, \hat{\gamma}') && (3.44) \\ &\leq d_1 + \delta(\kappa, w_0, a) && (3.45) \end{aligned}$$

□

The tables 3.3 show some admissible parameters for the algorithm for $w_0 = 0.1$ and $a = 0.5$. Normalization is with respect to d_1 in the left table and with respect to w in the right one.

Chapter 4

Curve reconstruction from the brightness image

In the previous two chapters, an algorithm to compute curves from a discrete *vector field* has been described and analyzed. This chapter considers the problem of computing curves from a *brightness image* by using a discrete vector field as an intermediate representation. That is, a vector field is derived from the brightness image and then fed to the algorithm described in chapter 2 to compute curves. A model of the brightness distribution in the neighborhood of edges is used to compute the vector field. A result similar to theorems 1 and 2 will be proved which ensures that all curves which satisfy the noisy model are recovered with an upper bound on localization error. This noisy model is defined with respect to the brightness image, whereas the model of the previous chapters was defined with respect to the vector field. The relation between these two models is established in the proof of theorem 3 on page 93.

4.1 Curve model and brightness templates

What does the brightness distribution look like in the neighborhood of an edge ? A simple model is to assume that brightness is constant along the edge and changes rapidly across the edge[1]. Let's consider the case of a straight-line edge γ and let's fix a Cartesian coordinate system (x', y') such that γ lies on the y'-axis (see figure 4-2) . Then the brightness model is given by:

$$I(x', y') = f_u(x') \tag{4.1}$$

where $f_u : \mathbb{R} \to \mathbb{R}$ is a function, called *profile*, which represents the brightness change across the edge. The multi-dimensional parameter $u \in U$ identifies a specific profile from a family $\{f_u \ : \ u \in U\}$ of possible profiles. For example, a family of blurred step profiles is indexed by a three dimensional parameter $u = (c_1, c_2, \sigma)$ and is shown in figure 4-1. The amplitude of the brightness change in the profile f_u will be denoted by $A(u)$. For the blurred step profile, it is given by : $A(c_1, c_2, \sigma) = |c_1 - c_2|$.

[1] The assumption that brightness is constant along the edge can probably be removed.

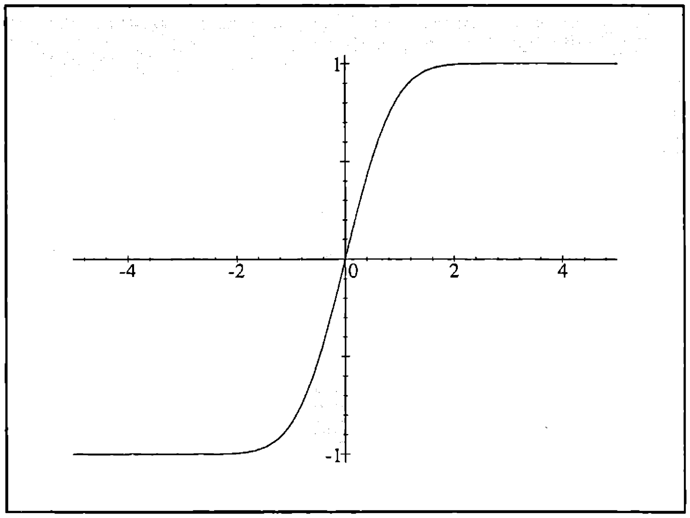


Figure 4-1: Blurred step profile with $c_1 = -1$, $c_2 = 1$, $\sigma = 1$. The one dimensional brightness distribution of this profile is obtained by convolving a step function $c_1 + (c_2 - c_1)\delta^{(-1)}(x)$ with a Gaussian smoothing filter of variance σ. One gets then: $f_{c_1,c_2,\sigma}(x) = c_1 + (c_2 - c_1)\sqrt{1/2\pi\sigma^2}\int_{-\infty}^{x}\exp(-t^2/2\sigma^2)\mathrm{d}t$.

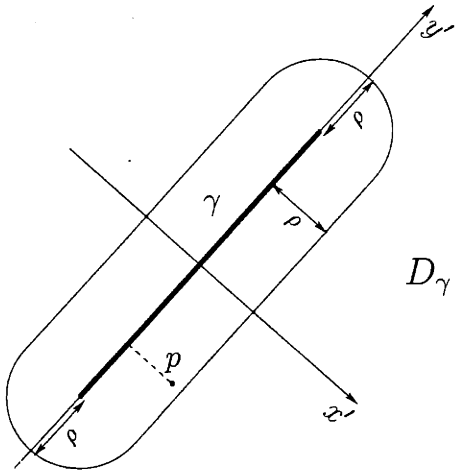


Figure 4-2: The domain D_γ of the brightness distribution $I_{\gamma,u}$. For any $p \in D_\gamma, I_{\gamma,u} = f_u(x'_p)$

For any straight line γ and profile parameter $u \in U$ let $I_{\gamma,u}$ denote the brightness distribution, called *template*, defined by (4.1). Its domain, denoted D_γ, is the set of points in the plane with distance from γ less than ρ (see figure 4-2). The constant ρ is given by

$$\rho = \frac{\delta}{2} + \sqrt{d_2^2 + h_2^2} + w_0 + \frac{1}{2}$$

where δ is the diameter of the regions used to compute the tangent vectors and d_2, h_2 are the parameters of the lateral regions used for non-maximum suppression.

The input image I is defined on the set of integer pairs $\mathbb{Z}^2$. For any $(i,j) \in \mathbb{Z}^2$, let $N^{i,j} \subset \mathbb{Z}^2$ denote the neighborhood of (i,j) used to compute a tangent vector (section 4.2 describes an approach to compute tangent vectors). Let $I|N^{i,j}$, $I_{\gamma,u}|N^{i,j}$ be the restrictions of I, $I_{\gamma,u}$ to the neighborhood $N^{i,j}$. Let $||\cdot||$ be a norm on brightness distributions and let $||\cdot||_{N^{i,j}}$ be its restriction to $N^{i,j}$. Namely, for instance,

$$||I||_{N^{i,j}} = ||I|N^{i,j}||$$

Definition 5 *For any input image I and $\eta \geq 0$, $C_\eta^0(I)$ is the set of straight line segments γ for which there exists $u \in U$ such that*

$$\frac{||I - I_{\gamma,u}||_{N^{i,j}}}{A(u)} \leq \eta \tag{4.2}$$

for every $N^{i,j} \subset D_\gamma$.

If the l_2-norm is used, then (4.2) can be rewritten as

$$\frac{1}{|N^{i,j}|} \sum_{(i',j') \in N^{i,j}} [I(i',j') - I_{\gamma,u}(i',j')]^2 \leq \eta A(u)$$

where $|N^{i,j}|$ is the number of elements in $N^{i,j}$.

4.1.1 Generalization to non-zero curvature

Formula (4.1), which defines brightness templates, can be easily generalized to curves with non-zero curvature. To do this, let's rewrite it in the following form:

$$I_{\gamma,u}(p) = f_u(\text{sign}(p,\overline{\gamma})d(p,\overline{\gamma})), \qquad p \in D_\gamma \tag{4.3}$$

where

- $\overline{\gamma}$ is the curve obtained by adding two straight line segments of length ρ to each end of γ
- $d(p,\overline{\gamma})$ is the distance between the point p and the curve $\overline{\gamma}$
- D_γ is again the set of points with distance from γ less than ρ.
- $\text{sign}(p,\overline{\gamma})$ is ± 1 depending on which side of $\overline{\gamma}$ p lies on.

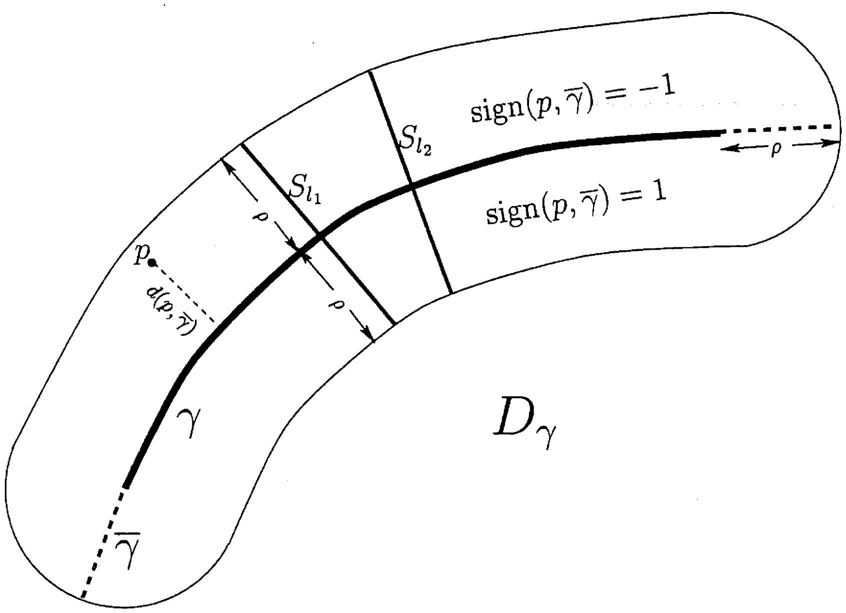


Figure 4-3: The curve $\overline{\gamma}$ is obtained by adding two straight line segments of length ρ to each end of γ. The domain D_γ is split into two parts by $\overline{\gamma}$.

Clearly, for $\text{sign}(p, \overline{\gamma})$ to be well-defined, it is necessary to assume that D_γ is split into two disjoint parts by $\overline{\gamma}$. This is guaranteed by requiring that γ satisfy a condition similar to (C3) on page 78. Namely, let $\overline{L}$ be the length of $\overline{\gamma}$ and for any $l \in [0, \overline{L}]$ let S_l be the segment with length 2ρ perpendicular to γ which intersects it symmetrically at the point with arc-length parameter l. Then a condition which ensures that (4.3) is well-defined is:

$$l_1 \neq l_2 \implies S_{l_1} \cap S_{l_2} = \emptyset \tag{4.4}$$

The family of curves for which (4.4) holds is denoted $\overline{C}_\rho$.

Definition 6 *For any input image I, $\kappa \geq 0$, and $\eta \geq 0$, $C^\kappa_\eta(I)$ is the set of curves γ with curvature less than κ and belonging to $\overline{C}_\rho$ for which there exists $u \in U$ such that*

$$\frac{||I - I_{\gamma,u}||_{N^{i,j}}}{A(u)} \leq \eta \tag{4.5}$$

for every $N^{i,j} \subset D_\gamma$.

4.2 Computing a vector field by template fitting

To detect and reconstruct the curves satisfying the noisy model given by definition 6, a set of tangent vectors $\Phi = \{\phi(\hat{p}_{i,j})\}$ is computed and then fed to the algorithm described in chapter 2. Each point $\hat{p}_{i,j}$ and vector $\phi(\hat{p}_{i,j})$ are obtained by fitting a local brightness template to the block of data $I|N^{i,j}$ for every $(i, j) \in \mathbb{Z}^2$. The point

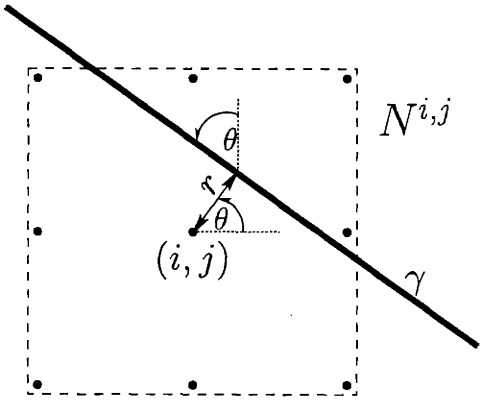


Figure 4-4: The polar coordinates r, θ used to parameterize straight lines with respect to a fixed points (i,j). $N^{i,j}$ is a neighborhood of discrete points around (i,j).

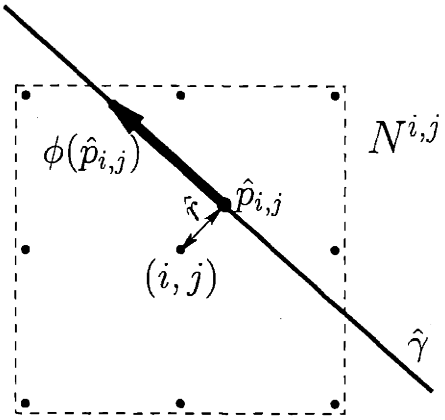


Figure 4-5: The vector $\phi(\hat{p}_{i,j})$ estimated from the neighborhood $N^{i,j}$.

$\hat{p}_{i,j}$ is constrained to be close to (i,j):

$$||\hat{p}_{i,j} - (i,j)|| \leq \frac{1}{2} \tag{4.6}$$

To describe this local fitting operator, let's fix some $(i,j) \in \mathbb{Z}^2$ and let's parameterize straight lines by means of polar coordinates r, θ with respect to (i,j), as shown in figure 4-4. Then let $I_{r,\theta,u}$ denote the brightness template $I_{\gamma,u}$ whose curve γ is given by the straight line with polar parameters r, θ. The local fitting operator, denoted **F**, estimates the parameters r, θ, u by fitting $I_{r,\theta,u}$ to $I|N^{i,j}$ so that the error

$$\mathcal{E}(r,\theta,u) \equiv ||I - I_{r,\theta,u}||_{N^{i,j}} \tag{4.7}$$

is minimized. Thus, the following optimization problem is obtained

$$\min_{0\leq r \leq \frac{1}{2}} \min_{0 \leq \theta \leq 2\pi} \min_{u \in U} ||I - I_{r,\theta,u}||_{N^{i,j}} \tag{4.8}$$

If the l_2-norm is used then the problem is:

$$\min_{r,\theta,u} \sum_{(i',j') \in N^{i,j}} [I(i',j') - I_{r,\theta,u}(i',j')]^2$$

The estimated parameters will be denoted $\hat{r}, \hat{\theta}, \hat{u}$, so that the fitting operator **F** is really a map: $\mathbf{F} : I|N^{i,j} \mapsto (\hat{r}, \hat{\theta}, \hat{u})$. Notice that we are not assuming that the triple $(\hat{r}, \hat{\theta}, \hat{u})$ is *exactly* the minimizer of (4.8). The performance of **F**, which implicitly describes how close $(\hat{r}, \hat{\theta}, \hat{u})$ is to the optimal solution, is characterized by several error measures defined in section 4.3.

The point $\hat{p}_{i,j}$ is given by the intersection between the estimated line $\hat{\gamma}$ and the line perpendicular to $\hat{\gamma}$ passing through (i,j) (see figure 4-5). The vector $\phi(\hat{p}_{i,j})$ is parallel to $\hat{\gamma}$ and its magnitude is:

$$|\phi(\hat{p}_{i,j})| = \frac{A(\hat{u})}{\hat{E}^{i,j}} = \frac{A(\hat{u})}{||I - I_{\hat{r},\hat{\theta},\hat{u}}||_{N^{i,j}}} \tag{4.9}$$

where $\hat{E}^{i,j}$ denotes the fitting error:

$$\hat{E}^{i,j} = ||I - I_{\hat{r},\hat{\theta},\hat{u}}||_{N^{i,j}} \tag{4.10}$$

4.3 Performance of the local fitting operator

4.3.1 Fitting error and accuracy of $\hat{p}_{i,j}$

To define the performance of the fitting operator $\mathbf{F} : I|N^{i,j} \mapsto (\hat{r}, \hat{\theta}, \hat{u})$, let's assume that the data in the neighborhood $N^{i,j}$, $I|N^{i,j}$, is close to a brightness template $I_{\overline{r},\overline{\theta},\overline{u}}$ such that $\overline{r}$, namely the distance of the template's curve from the point (i,j), is less than 1/2:

$$||I - I_{\overline{r},\overline{\theta},\overline{u}}||_{N^{i,j}} \leq \eta A(\overline{u}), \qquad \overline{r} \leq 1/2 \tag{4.11}$$

Notice that $\eta \geq 0$ is an upper bound to the distance between the data $I|N^{i,j}$ and the ideal template $I_{\overline{r},\overline{\theta},\overline{u}}$ normalized by the amplitude of the brightness change $A(\overline{u})$. If $\eta = 0$ then the data is exactly equal to the ideal template and therefore (4.8) vanishes for $r = \overline{r}, \theta = \overline{\theta}, u = \overline{u}$, which is then, in this case, the optimal solution of the optimization problem (4.8). In general, for $\eta > 0$, (4.8) is positive but no greater than $\eta A(\overline{u})$.

The performance of the fitting operator depends on how close to $\overline{r}, \overline{\theta}, \overline{u}$ the estimated parameters $\hat{r}, \hat{\theta}, \hat{u}$ are. Rather than measuring performance by means of upper bounds on the distance between these parameters, it is more convenient to consider other parameters, namely the residual fitting error $\hat{E}^{i,j}$ normalized by $A(\hat{u})$ and the displacement of the point $\hat{p}_{i,j}$ from its true position $\overline{p}_{i,j}$. Let then $\hat{\eta}(\eta)$ be the maximum normalized error over all observations satisfying (4.11):

$$\hat{\eta}(\eta) = \sup_{\overline{r},\overline{\theta},\overline{u},I} \left\{ \frac{\hat{E}^{i,j}}{A(\hat{u})} \ : \ ||I - I_{\overline{r},\overline{\theta},\overline{u}}||_{N^{i,j}} \leq \eta A(\overline{u}), \ \overline{r} \leq \frac{1}{2} \right\} \tag{4.12}$$

where $\hat{E}^{i,j} = ||I - I_{\hat{r},\hat{\theta},\hat{u}}||_{N^{i,j}}$. Notice that from (4.9) it follows that (4.12) implies

$$|\phi(\hat{p}_{i,j})| \geq \frac{1}{\hat{\eta}(\eta)} \tag{4.13}$$

For uniformity of notation let now $\hat{p} = \hat{p}_{i,j}$, $\overline{p} = \overline{p}_{i,j}$. The error in $\hat{p}$ is measured by the parallel and orthogonal projections of $||\hat{p} - \overline{p}||$ onto the straight line $\overline{\gamma}$. These error measures are denoted y_0 and w_0 respectively. From figure 4-6,

$$||\hat{p} - \overline{p}||^2 = \overline{r}^2 + \hat{r}^2 - 2\overline{r}\hat{r}\cos(\overline{\theta} - \hat{\theta}) \tag{4.14}$$

$$y_0 = \hat{r} \sin|\overline{\theta} - \hat{\theta}| \tag{4.15}$$

$$w_0 = |\overline{r} - \hat{r}\cos(\overline{\theta} - \hat{\theta})| \tag{4.16}$$

Let $\hat{y}_0(\eta)$, $\hat{w}_0(\eta)$ be the suprema of y_0, w_0,

$$\hat{y}_0(\eta) = \sup_{\overline{r},\overline{\theta},\overline{u},I} \left\{ y_0 \ : \ ||I - I_{\overline{r},\overline{\theta},\overline{u}}||_{N^{i,j}} \leq \eta A(\overline{u}), \ \overline{r} \leq \frac{1}{2} \right\} \tag{4.17}$$

$$\hat{w}_0(\eta) = \sup_{\overline{r},\overline{\theta},\overline{u},I} \left\{ w_0 \ : \ ||I - I_{\overline{r},\overline{\theta},\overline{u}}||_{N^{i,j}} \leq \eta A(\overline{u}), \ \overline{r} \leq \frac{1}{2} \right\} \tag{4.18}$$

In the noise-free case, $\eta = 0$, if the local fitting operator returns the parameters $(\overline{r}, \overline{\theta}, \overline{u})$ exactly, namely $\hat{r} = \overline{r}$, $\hat{\theta} = \overline{\theta}$, $\hat{u} = \overline{u}$, then

$$\hat{y}_0(0) = \hat{w}_0(0) = \hat{\eta}(0) = 0$$

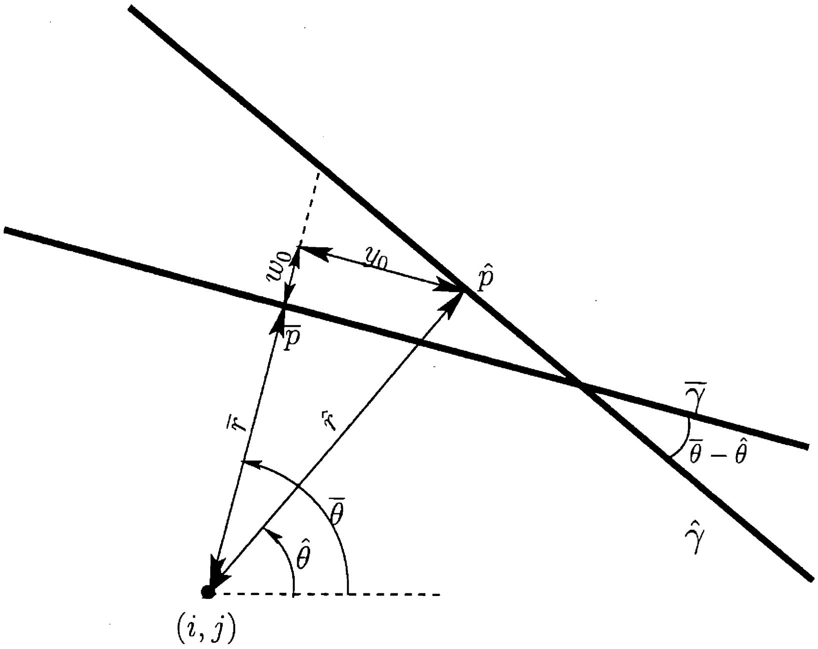


Figure 4-6: The parameters y_0 and w_0, are the parallel and perpendicular displacements of $\hat{p}$ from $\overline{p}$ relative to the model curve $\overline{\gamma}$.

4.3.2 Decay width $w(\eta)$ and orientation uncertainty $\Theta(\eta)$

An important issue is to determine at what distance from a curve $\gamma \in C^0_\eta(I)$ the computed vector field is guaranteed to decay below the value achieved close to the curve. To this purpose, we seek now an upper bound to the field magnitude at a point $\hat{p}$ as a function of its distance w from γ (see figure 4-7).

This upper bound can be derived because $\gamma \in C^0_\eta(I)$ puts a constraint on I in the neighborhood of $\hat{p}$, provided that $\hat{p}$ is not too far from γ.

Let's assume that the distance of $\hat{p}$ from $(i,j) \in \mathbb{Z}^2$ is less than 1/2 (see figure 4-7). Let $\hat{\gamma}$ be a straight line passing through $\hat{p}$ with orientation $\hat{\theta}$ with respect to the x-axis. For some $\hat{u} \in U$ let $I_{\hat{p},\hat{\theta},\hat{u}} = I_{\hat{\gamma},\hat{u}}$ be the brightness template with curve $\hat{\gamma}$ and profile parameter $\hat{u}$. If $\hat{p}$, $\hat{\theta}$, $\hat{u}$ are the parameters returned by the local fitting operator on the neighborhood $N^{i,j}$ then

$$|\phi(\hat{p})| = \frac{A(\hat{u})}{||I - I_{\hat{p},\hat{\theta},\hat{u}}||_{N^{i,j}}} \tag{4.19}$$

Let θ be the orientation of γ with respect to $\hat{\gamma}$ and let w be the distance of $\hat{p}$ from γ. Since $\gamma \in C^0_\eta(I)$, definition 5 guarantees that there exists a brightness template $I_{w,\theta,u}$ such that

$$||I - I_{w,\theta,u}||_{N^{i,j}} \leq \eta A(u)$$

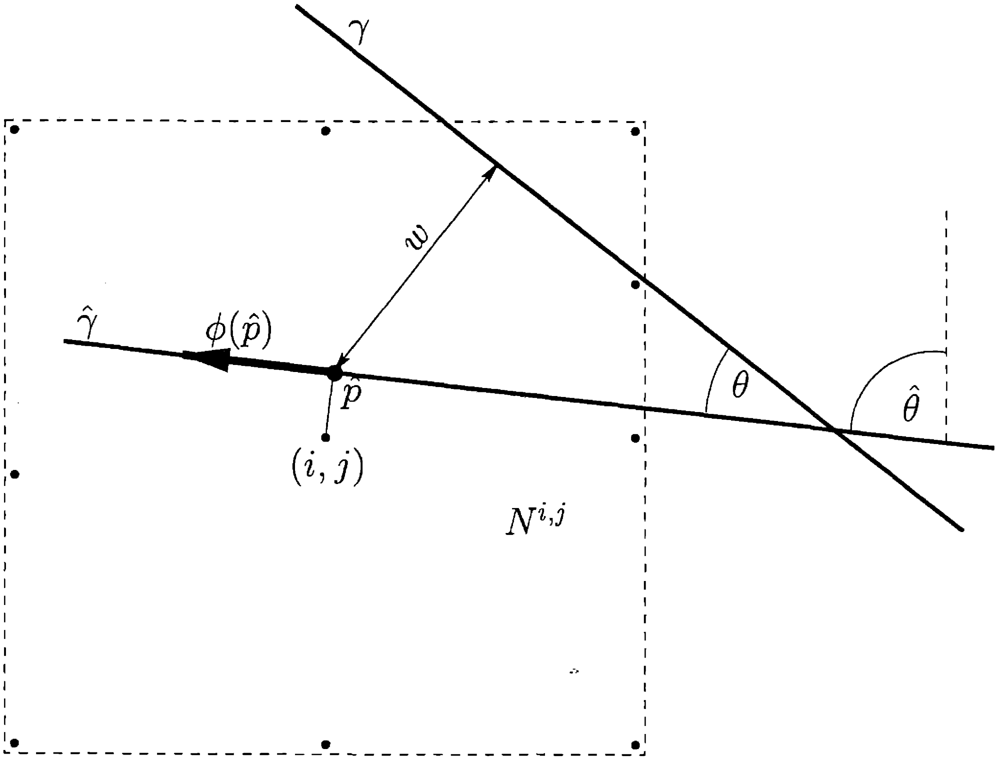


Figure 4-7: Definition of $w(\eta)$.

Then, by using the triangular inequality:

$$\begin{aligned}||I_{\hat{p},\hat{\theta},\hat{u}} - I||_{N^{i,j}} &\geq ||I_{\hat{p},\hat{\theta},\hat{u}} - I_{w,\theta,u}||_{N^{i,j}} - ||I_{w,\theta,u} - I||_{N^{i,j}} \\ &\geq ||I_{\hat{p},\hat{\theta},\hat{u}} - I_{w,\theta,u}||_{N^{i,j}} - \eta A(u)\end{aligned}$$

so that, by using (4.19),

$$|\phi(\hat{p})| \leq \frac{A(\hat{u})}{||I_{\hat{p},\hat{\theta},\hat{u}} - I_{w,\theta,u}||_{N^{i,j}} - \eta A(u)} \tag{4.20}$$

Now, by defining $F(u,w,\theta,\eta)$ as follows

$$F(u,w,\theta,\eta) = \max_{\hat{p},\hat{\theta},\hat{u}} \frac{A(\hat{u})}{||I_{\hat{p},\hat{\theta},\hat{u}} - I_{w,\theta,u}||_{N^{i,j}} - \eta A(u)}$$

one gets

$$|\phi(\hat{p})| \leq F(u,w,\theta,\eta) \tag{4.21}$$

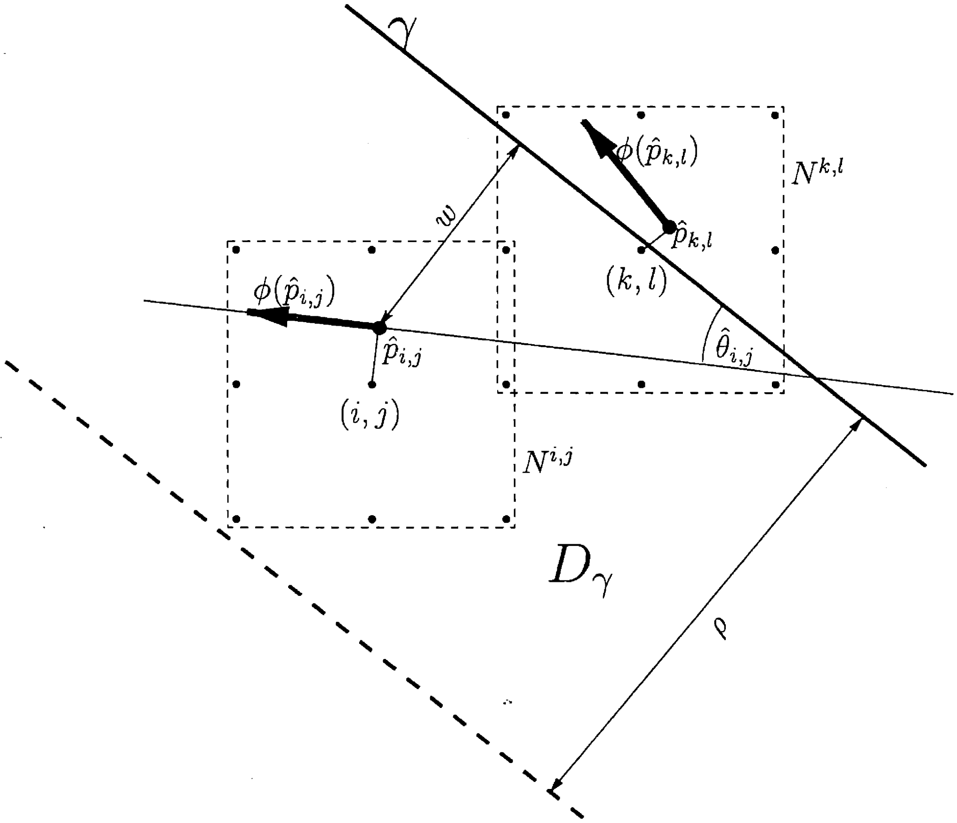


Figure 4-8: Proposition 14

Finally, let

$$w(u,\eta) = \max\left\{w \;:\; \exists\theta,\; F(u,w,\theta,\eta) \geq \frac{1}{\hat{\eta}(\eta)}\right\} \tag{4.22}$$

$$w(\eta) = \max_{u\in U} w(u,\eta) \tag{4.23}$$

$$\Theta(u,\eta) = \max\left\{\theta \;:\; \exists w,\; F(u,w,\theta,\eta) \geq \frac{1}{\hat{\eta}(\eta)}\right\} \tag{4.24}$$

$$\Theta(\eta) = \max_{u\in U} \Theta(u,\eta) \tag{4.25}$$

Proposition 14 *Let $\gamma \in C^0_\eta(I)$ and $N^{i,j} \subset D_\gamma$, where D_γ is the domain of the template $I_{\gamma,u}$ (figure 4-8). Let $\phi(\hat{p}_{i,j})$ be the tangent vector computed from $I|N^{i,j}$ and let $\hat{\theta}_{i,j}$ be its orientation with respect to γ. Similarly, let $\phi(\hat{p}_{k,l})$ be the tangent vector computed from $I|N^{k,l}$ where (k,l) is such that $d((k,l),\gamma) \leq 1/2$. Then*

$$d(\hat{p}_{i,j},\gamma) > w(\eta) \implies |\phi(\hat{p}_{i,j})| < |\phi(\hat{p}_{k,l})| \tag{4.26}$$

$$|\phi(\hat{p}_{i,j})| \geq |\phi(\hat{p}_{k,l})| \implies ||\hat{\theta}_{i,j}|| \leq \Theta(\eta) \tag{4.27}$$

$$||\hat{\theta}_{k,l}|| \leq \Theta(\eta) \tag{4.28}$$

Proof. Since $\gamma \in C^0_\eta(I)$, $N^{i,j} \subset D_\gamma$, $N^{k,l} \subset D_\gamma$, there is $u \in U$ such that

$$||I - I_{\gamma,u}||_{N^{i,j}} \leq \eta A(u) \tag{4.29}$$

$$||I - I_{\gamma,u}||_{N^{k,l}} \leq \eta A(u) \tag{4.30}$$

The upper bound (4.30) and $d((k,l),\gamma) \leq 1/2$ imply that the two conditions inside the brackets of the sup in (4.12) are satisfied so that (4.13) holds:

$$\frac{1}{\hat{\eta}(\eta)} \leq |\phi(\hat{p}_{k,l})| \tag{4.31}$$

Let $w = d(\hat{p}_{i,j},\gamma)$, and let $\theta \equiv \hat{\theta}_{i,j}$ be the orientation of $\phi(\hat{p}_{i,j})$ with respect to γ. Then, from (4.21) one gets

$$|\phi(\hat{p}_{i,j})| \leq F(u,w,\theta,\eta) \tag{4.32}$$

To prove (4.26), let's assume that $w > w(\eta)$. From (4.23) we get $w \geq w(u,\eta)$. Then, (4.22) implies

$$F(u,w,\theta,\eta) < \frac{1}{\hat{\eta}(\eta)} \tag{4.33}$$

The three inequalities (4.32),(4.33), (4.31), imply $|\phi(\hat{p}_{i,j})| < |\phi(\hat{p}_{k,l})|$ so that (4.26) is proved. To prove (4.27), let's assume that $|\phi(\hat{p}_{i,j})| \geq |\phi(\hat{p}_{k,l})|$. Then, by (4.31),

$$\frac{1}{\hat{\eta}(\eta)} \leq |\phi(\hat{p}_{i,j})| \tag{4.34}$$

and from (4.32)

$$\frac{1}{\hat{\eta}(\eta)} \leq F(u,w,\theta,\eta) \tag{4.35}$$

Thus, (4.24), (4.25) imply $\theta \equiv \hat{\theta}_{i,j} \leq \Theta(\eta)$, which proves (4.27). Finally, (4.28) follows by letting $(i,j) = (k,l)$ in (4.27). □

4.4 Result on worst-case error

Theorem 3 *Let $\gamma \in C^0_\eta(I)$. Let Φ be the vector field obtained by the local fitting operator described in sections 4.2,4.3. Let $\hat{C}(\Phi)$ be the set of polygonal curves generated by the algorithm described in chapter 2 with the parameter constraints (3.5)–(3.9) on*

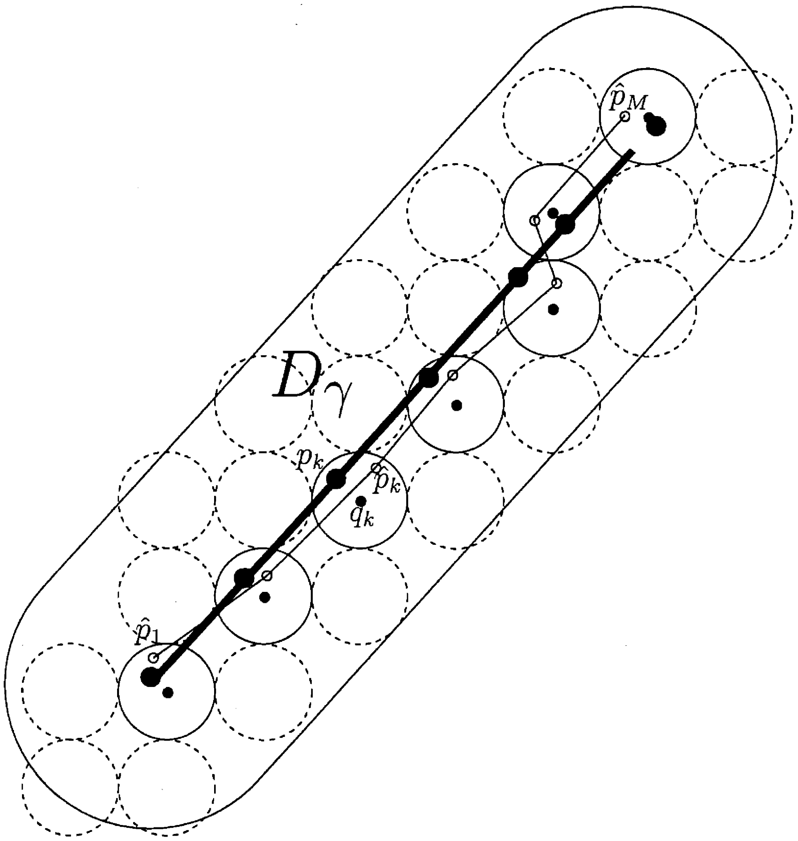


Figure 4-9: Theorem 3. The points $q_1, \dots, q_M$ belong to the regular grid $\mathbb{Z}^2$. They are the points in $\mathbb{Z}^2$ such that the unit ball $B_1(q_k)$ intersects γ. The points p_k (big dots) are the projections of q_k onto γ. The points $\hat{p}_k$ are the estimates of p_k. Notice that $p_k \in B_1(q_k)$, $\hat{p}_k \in B_1(q_k)$. Moreover, $||q_k - q_{k+1}|| \leq \sqrt{2}$ and $||p_k - p_{k+1}|| \leq \sqrt{2}$.

page 55 and a, Θ, w, w_0 set as follows:

$$a^2 = \left(\sqrt{2} + 2\hat{y}_0(\eta)\right)^2 + 4\hat{w}_0^2(\eta) \tag{4.36}$$

$$w = w(\eta) \tag{4.37}$$

$$w_0 = \hat{w}_0(\eta) \tag{4.38}$$

$$\Theta = \Theta(\eta) \tag{4.39}$$

Then, there exists $\hat{\gamma} \in \hat{C}(\Phi)$ such that

$$d(\gamma, \hat{\gamma}) \leq d_1 + \frac{a}{2} \tag{4.40}$$

Proof. The most natural proof would be to show that

$$C^0_\eta(I) \subset C^0(\Phi; w, \Theta, \hat{w}_0, \hat{a}) \tag{4.41}$$

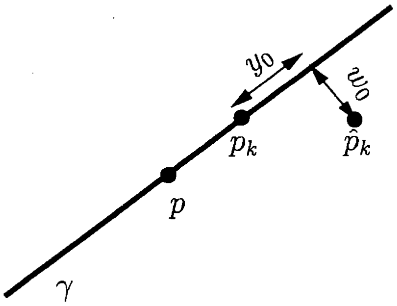


Figure 4-10: For every $p \in \gamma$ there exists $\hat{p}_k$ such that $||p - \hat{p}_k||^2 \leq \left(\sqrt{1/2} + \hat{y}_0(\eta)\right)^2 + \hat{w}_0^2(\eta)$.

where $C^0(\Phi; w, \Theta, \hat{w}_0, \hat{a})$ is given by definition 3 on page 79 and then use theorem 2 on page 80. However, since this approach would require some minor modifications of definition 3, a direct proof is given instead.

Let $\gamma \in C^0_\eta(I)$ and let $\Delta_\gamma = \{q_1, \dots, p_M\}$, $q_k = (i_k, j_k)$ be the sequence of points in $\mathbb{Z}^2$ shown in figure 4-9. Clearly,

$$d(q_k, \gamma) \leq \frac{1}{2}, \qquad k = 1, \dots, M \tag{4.42}$$

$$||q_k - q_{k+1}|| \leq \sqrt{2}, \qquad k = 1, \dots, M-1 \tag{4.43}$$

Let p_k be the projection of q_k onto γ. Let $\hat{p}_k \equiv \hat{p}_{i_k, j_k}$ be the estimated vertex from $I|N^k$, where $N^k \equiv N^{i_k, j_k}$. Since $\gamma \in C^0_\eta(I)$, there exists $u \in U$ such that, for each k,

$$||I - I_{\gamma,u}||_{N^k} \leq \eta A(u)$$

Then, by (4.18) on page 89 and $w_0 = \hat{w}_0(\eta)$ it follows that

$$d(\hat{p}_k, \gamma) \leq w_0$$

Similarly, from (4.17) we have that the longitudinal displacement of $\hat{p}_k$ from p_k is upper bounded by $\hat{y}_0(\eta)$. Then,

$$\begin{aligned} ||\hat{p}_k - \hat{p}_{k+1}||^2 \leq (||p_k - p_{k+1}|| + 2\hat{y}_0(\eta))^2 + 4\hat{w}_0^2(\eta) \leq \\ \left(\sqrt{2} + 2\hat{y}_0(\eta)\right)^2 + 4\hat{w}_0^2(\eta) = a^2 \end{aligned} \tag{4.44}$$

For every $p \in \gamma$ there exists $\hat{p}_k$ such that (see figure 4-10)

$$||p - \hat{p}_k||^2 \leq \left(\sqrt{1/2} + \hat{y}_0(\eta)\right)^2 + \hat{w}_0^2(\eta) = \frac{a^2}{4} \tag{4.45}$$

Now, we want to prove that the points $\hat{p}_k$ belong to P. For this, it is sufficient to prove that $\gamma \in C^0_{w,\Theta}(\Phi)$, namely that conditions (C1),(C2) on page 54 hold and then use propositions 11 and 12 on page 65. From proposition 14 on page 92 and

$w = w(\eta)$ we get

$$d(\hat{p}_{i,j}, \gamma) > w \implies |\phi(\hat{p}_{i,j})| < |\phi(\hat{p}_k)|$$

so that condition (C1) is satisfied. Again, from proposition 14 we get

$$|\phi(\hat{p}_{i,j})| \geq |\phi(\hat{p}_k)| \implies ||\hat{\theta}_{i,j}|| \leq \Theta(\eta) = \Theta$$

which is slightly different from (C2) but still sufficient to guarantee that part two of non-maximum suppression does not suppress any $\hat{p}_k$. In fact, from (4.28) one has $||\hat{\theta}_{k,l}|| \leq \Theta(\eta)$. Therefore,

$$|\phi(\hat{p}_{i,j})| \geq |\phi(\hat{p}_k)| \implies ||\hat{\theta}_{i,j} - \hat{\theta}_{k,l}|| \leq 2\Theta = \overline{\Theta}$$

that is,

$$||\hat{\theta}_{i,j} - \hat{\theta}_{k,l}|| > \overline{\Theta} \implies |\phi(\hat{p}_{i,j})| < |\phi(\hat{p}_k)|$$

The last part of the proof is the same as the proof of theorem 2. The upper bound (4.44) ensures that $\hat{p}_1, \ldots, \hat{p}_M$ all belong to the same connected component Q.

Let $\hat{\gamma}$ be the polygonal curve computed by the algorithm from Q. As explained in section 3.8, its vertices satisfy (3.21). By an argument similar to the one in section 3.8, we have that $d(\hat{p}_k, \hat{\gamma}') \leq d_1$ (see figures 3-15, 3-16)[2] and therefore $d(\Delta_\gamma, \hat{\gamma}') \leq d_1$, where $\Delta_\gamma = \hat{p}_1, \ldots, \hat{p}_M$. Then, by using (4.45) and the triangular inequality:

$$d(\gamma, \hat{\gamma}') \leq d(\gamma, \Delta_\gamma) + d(\Delta_\gamma, \hat{\gamma}') \leq d_1 + \frac{a}{2} \qquad (4.46)$$

□

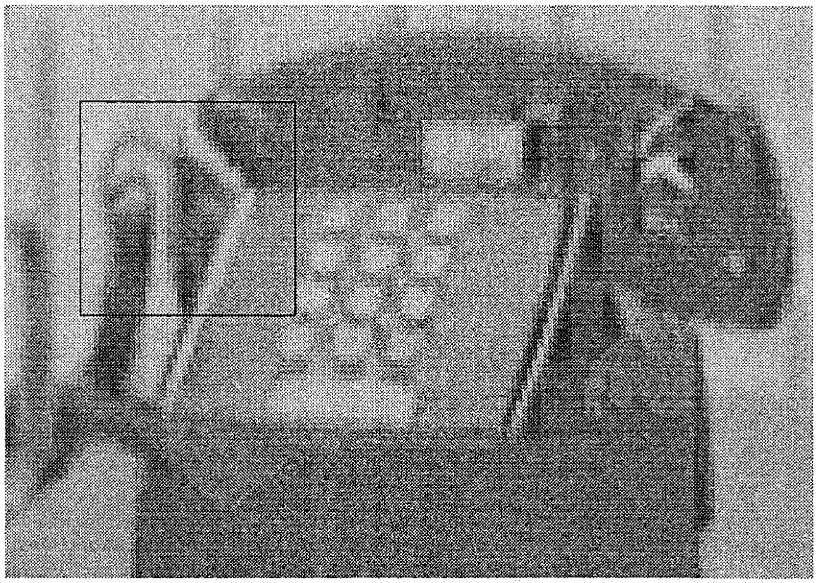
[2] The points $\hat{p}_k$ are denoted q_i in these figures.

Chapter 5

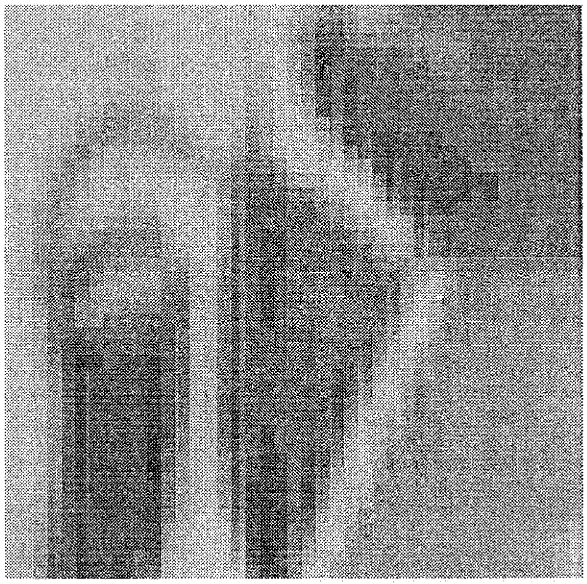
Experiments

Chapters 2 and 3 described an algorithm to compute curves from a brigthness image by using a finite vector field as an intermediate representation. This chapter reports the results of some experiments carried out with some real images. To compute the vector field, the step-profile model described in section 4.1 and figure 4-1 has been used with the parameter σ set to 0.5. The input brightness image I has the form $I : \Omega \to \{0, 1, \ldots, 255\}$ where $\Omega \subset \mathbb{Z}^2$. The regions $\{N^{i,j} : (i,j) \in \mathbb{Z}^2\}$ used to estimate the vectors (see section 4.2) are square blocks with 3×3 pixels. A vector is computed for every such block contained in the domain of the image. The constraints (2.6)-(2.9) on admissible triples have been used with following thresholds: $r = 1$, $\Theta_2 = 30°$. The constraint (2.10) has not been used. The other parameters have been set as follows: $\overline{\Theta} = 15$, $a = 0.5$ (except exp. 2 where $a = 0.2$), $d_1 = 0.4, 0.5$, $d_2 = 0.92, 1.15$, $h_1 = 0.76$, $h_2 = 0.96$. Only curves for which the vector field magnitude and brightness variation are above a threshold are shown.

5.1 Experiment 1: long edges with low curvature

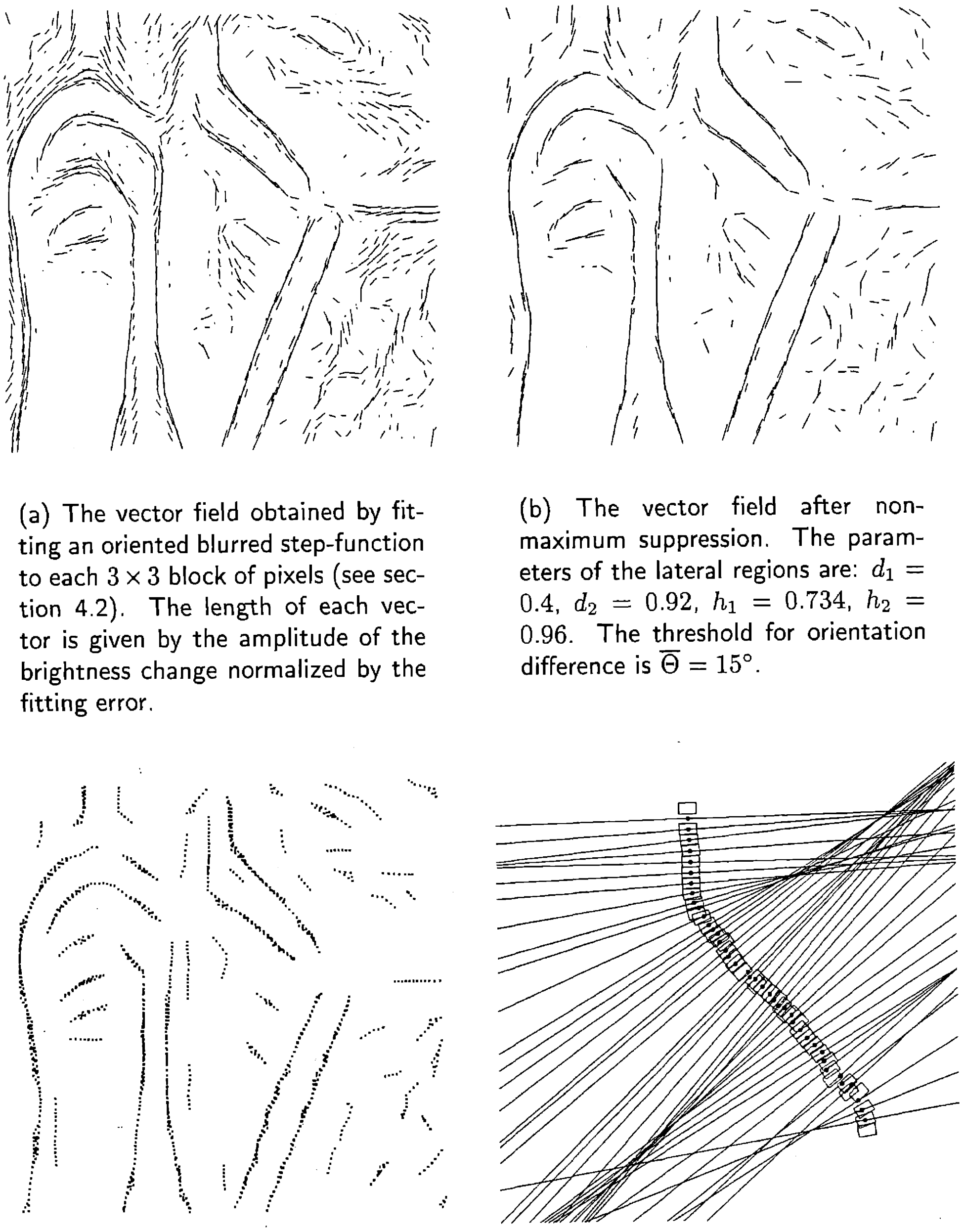


(a) Image of a telephone.



(b) The input image for experiment 1.

Figure 5-1: The input image (b) for experiment 1 is the portion of the image shown in (a). Continues in figure 5-2.



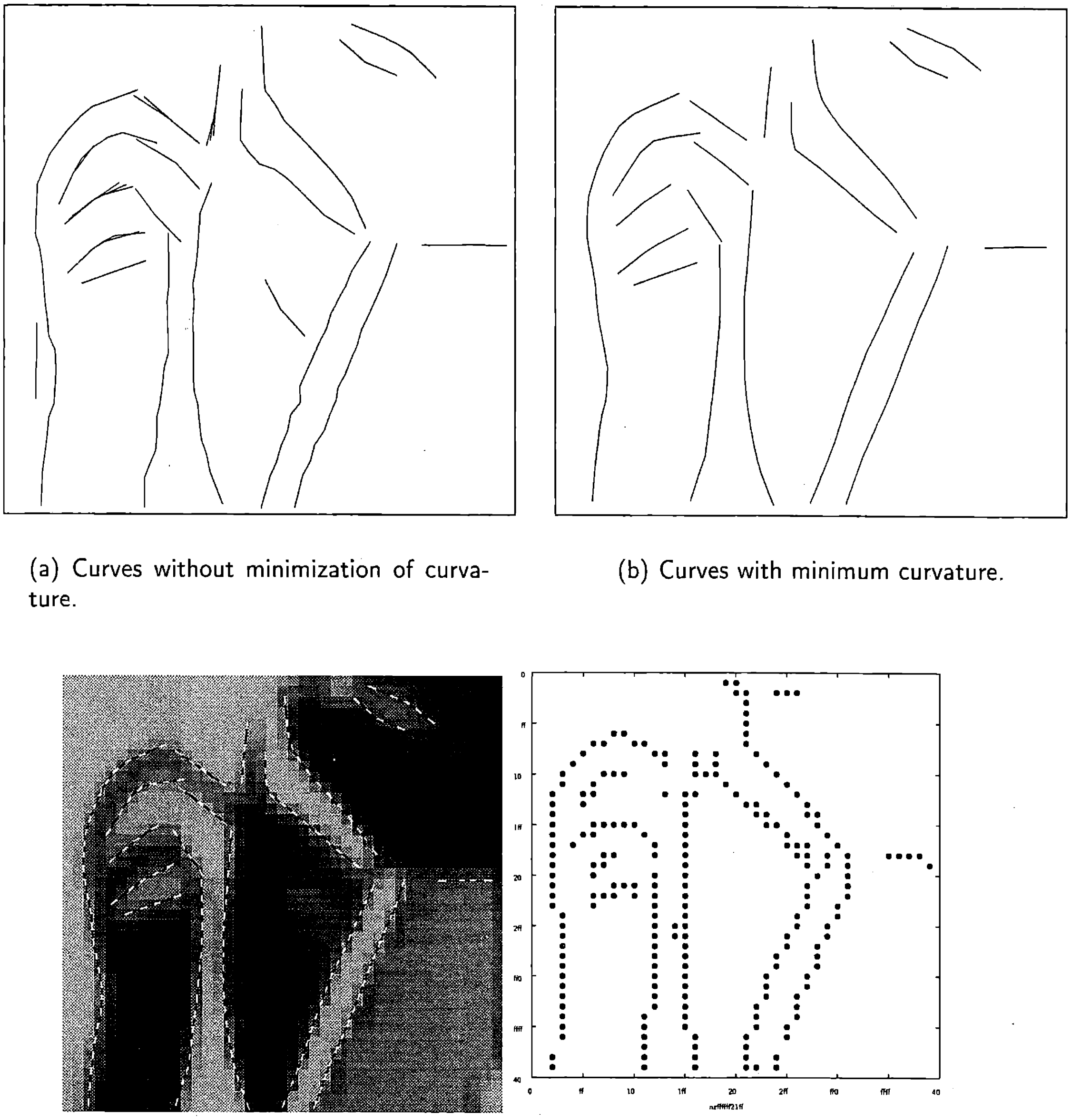
(a) The vector field obtained by fitting an oriented blurred step-function to each 3×3 block of pixels (see section 4.2). The length of each vector is given by the amplitude of the brightness change normalized by the fitting error.

(b) The vector field after non-maximum suppression. The parameters of the lateral regions are: $d_1 = 0.4$, $d_2 = 0.92$, $h_1 = 0.734$, $h_2 = 0.96$. The threshold for orientation difference is $\overline{\Theta} = 15°$.

(c) The connected components with at least 10 points obtained from the vector field in (b). Points have been added between any two neibhouring points so that $a = 0.5$.

(d) The longitudinal coorinate map λ of one connected component from (c). Each straight line represents a level curve for λ.

Figure 5-2: Steps of the algorithm on the image in 5-1(b). Continues in figure 5-3.



(a) Curves without minimization of curvature.

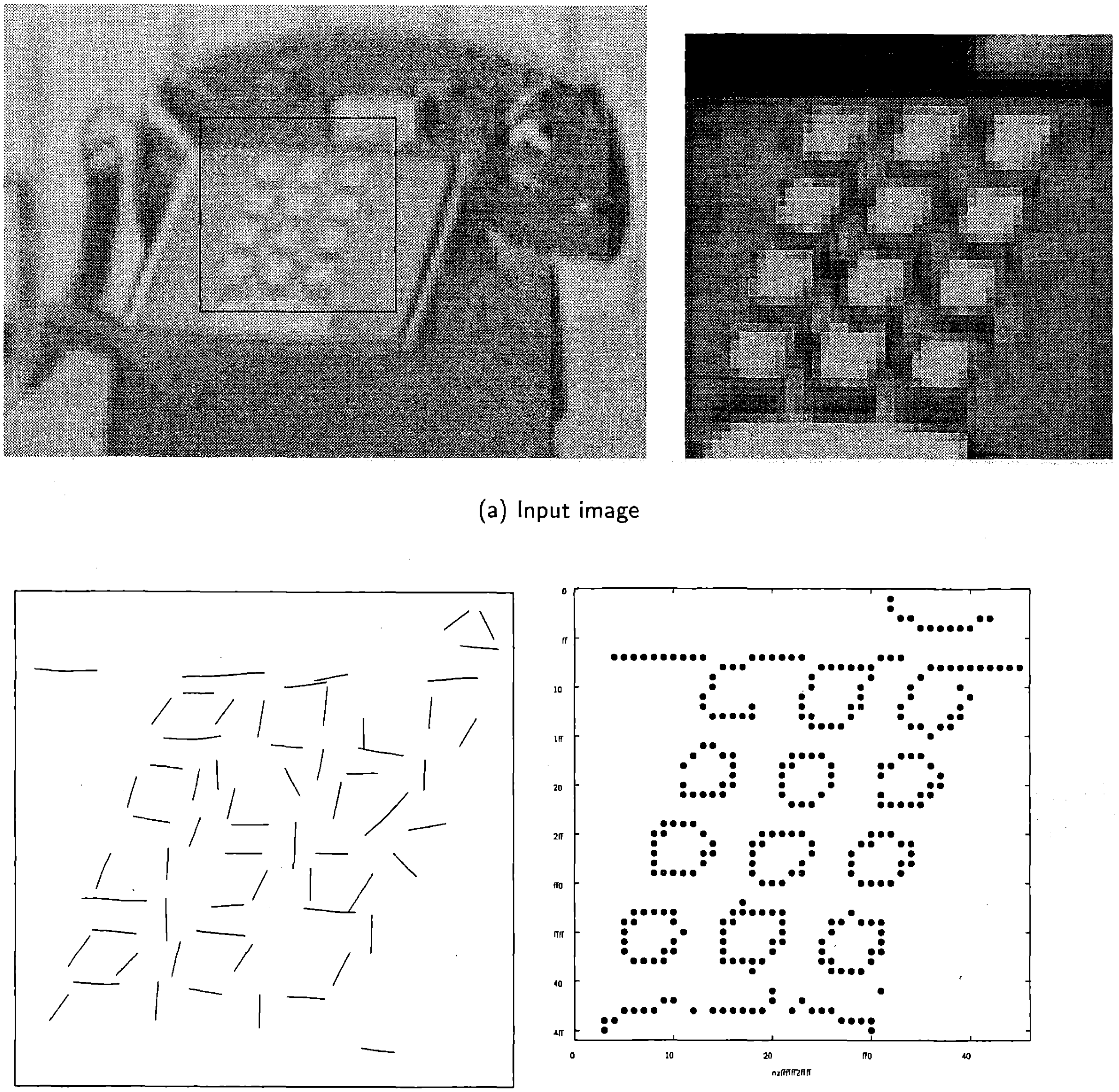
(b) Curves with minimum curvature.

(c) Computed curves on brightness image

(d) Edge points computed by Canny's algorithm

Figure 5-3: Continuation of figure 5-2. Only curves for which the vector field magnitude is above a threshold (0.3) and the brightness change is above a threshold (5.0) are shown. (d) The output of the Canny edge detector on the same image (figure 5-1(b)).

5.2 Experiment 2: short edges with high curvature



(a) Input image

(b) Curves computed by the algorithm with $\overline{\Theta} = 15°$, $a = 0.2$, $d_1 = 0.5$, $d_2 = 1.15$, $h_1 = 0.468$, $h_2 = 0.5$.

(c) Edge points computed by Canny's algorithm

Figure 5-4: Result of experiment 2. In this case, Canny's algorithm seems to perform better than the proposed algorithm. This is because the curves in the image are too short and their curvature is too high.

5.3 Experiment 3: image of a telephone

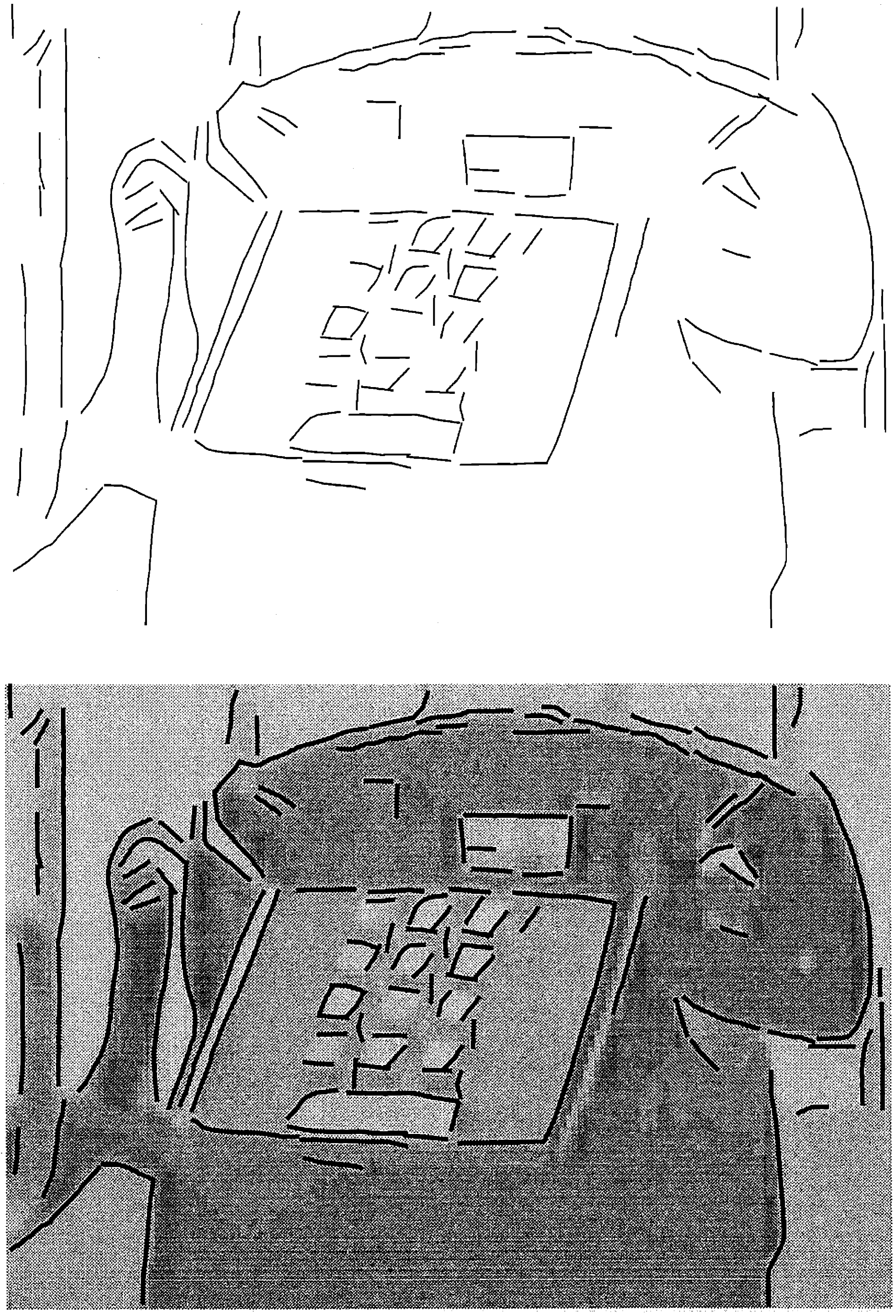
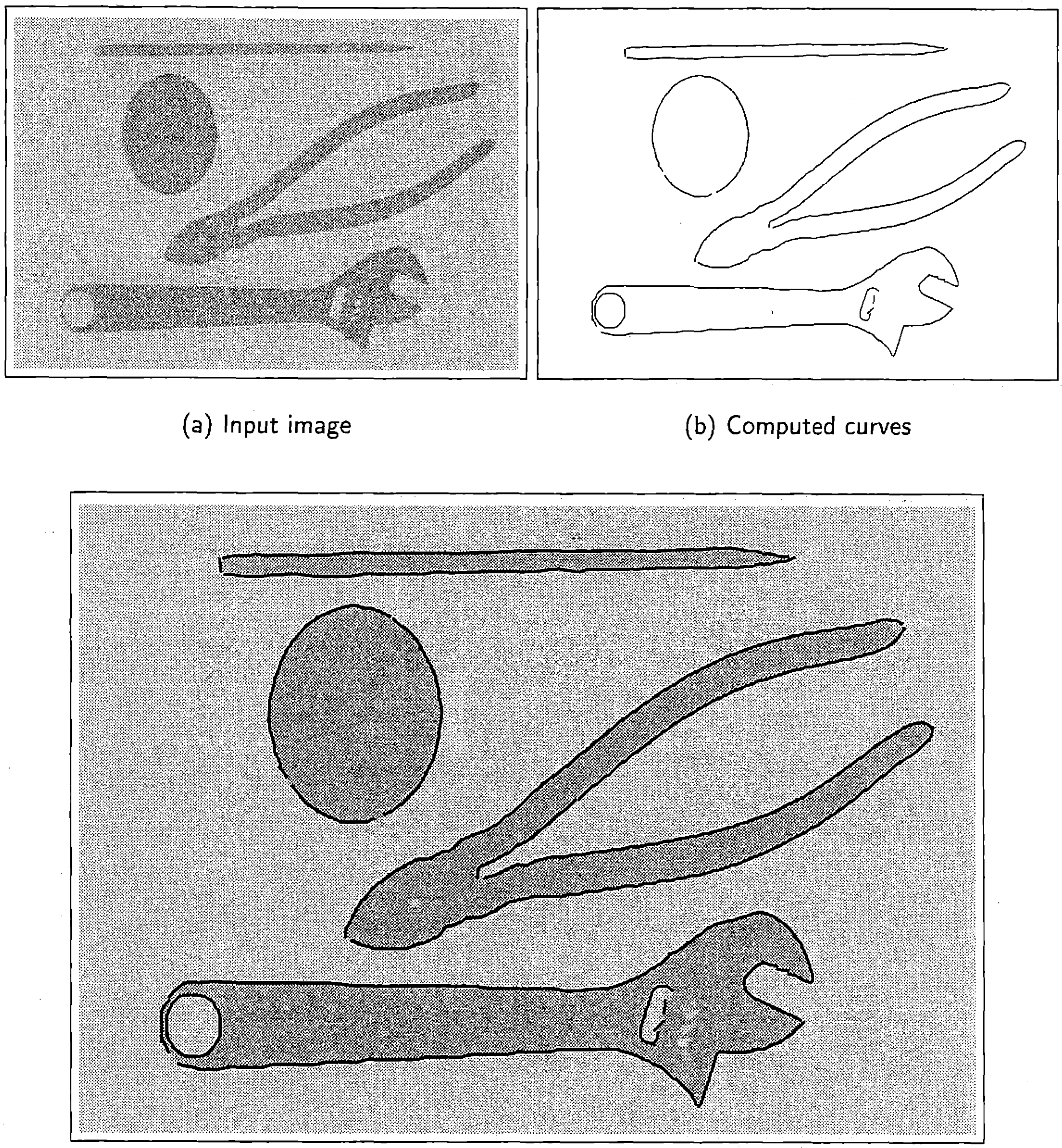


Figure 5-5: Result of the proposed algorithm on the whole telephone image. $\overline{\Theta} = 15°$, $a = 0.5$, $d_1 = 0.5$, $d_2 = 1.15$, $h_1 = 0.768$, $h_2 = 0.95$.

5.4 Experiment 4: three tools on a table



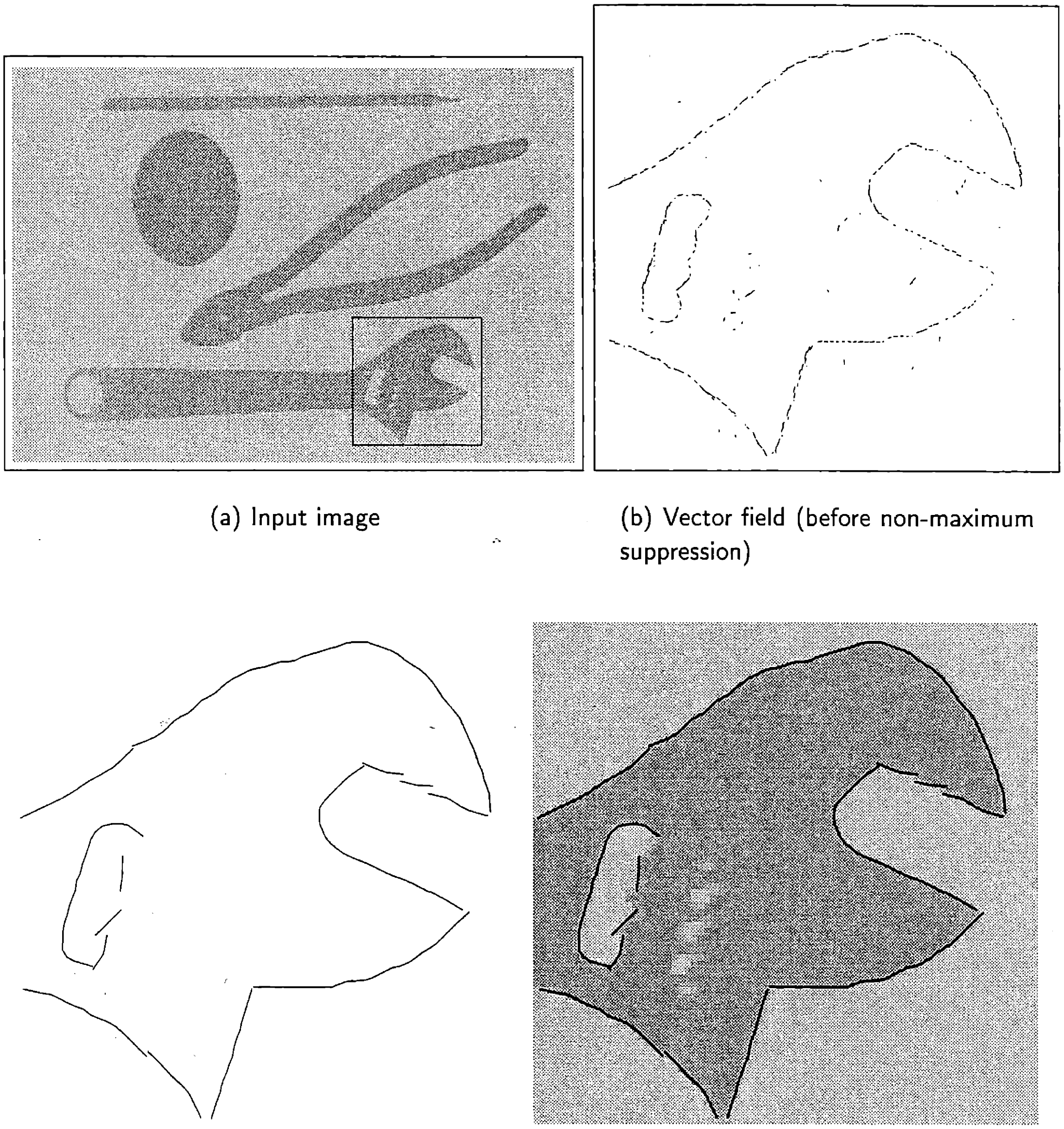
(a) Input image

(b) Computed curves

(c) Computed curves on brigthness image

Figure 5-6: Result of experiment 4. $\overline{\Theta} = 15^\circ$, $a = 0.5$, $d_1 = 0.5$, $d_2 = 1.15$, $h_1 = 0.768$, $h_2 = 0.95$.

5.5 Experiment 5: three tools on a table (detail)



(a) Input image

(b) Vector field (before non-maximum suppression)

(c) Computed curves

(d) Computed curves on brightness image

Figure 5-7: Result of experiment 5. $\overline{\Theta} = 15^\circ$, $a = 0.5$, $d_1 = 0.5$, $d_2 = 1.15$, $h_1 = 0.768$, $h_2 = 0.95$.

Chapter 6

Conclusions

Computing edges by local techniques — i.e. searching for local maxima of brightness change — has two limitations. First, the computed representation is often inaccurate since edges can be present where there is no brightness change and, conversely, brightness change also occurs away from edges. Second, the type of representation obtained — namely a set of edge points or short curve fragments — is local and fragmented and does not make explicit important information in the image which requires more complex geometric descriptors such as curves and regions.

To improve edge detection, global information and complex geometric descriptors (such as curves and regions) have to be introduced into the process. An important assumption which underlies the work presented in this thesis, is that, in order to control computational complexity without sacrificing robustness, global information should be introduced gradually. To do so, complex descriptors should be constructed hierarchically from simpler ones and each "step" in this hierarchy should be as small as possible.

We have considered the problem of representing the edges in the image by means of a set of curves. These curves are constructed by grouping together local hypotheses about edge localization represented by vectors in the real plane. Two well-known difficulties in edge detection are curve singularities (junctions and corners) and continuation of curves to regions where there is no brightness change. According to the hierarchical strategy, these issues should be dealt with only after a good representation of the edges where these difficulties are not present has been obtained. This led us to consider the problem of computing curves which

- do not contain any singularities
- have sufficiently large brightness change (relative to the noise level) at every point

To formulate the problem rigorously, we defined models of ideal curves in terms of both the vector field of local edge hypotheses and the brightness image. Then our objective was to detect all the curves in the image whose distance from an ideal model is below a threshold. We proved that, in the worst case, the proposed algorithm detects and reconstructs all these curves with an error which vanishes linearly when the upper bound on the deviation from the ideal model goes to zero.

Further work remains to be done to employ this curve-based edge representation to tackle the problems of junctions and edge continuation. Also, curve information should be integrated with region information, such as brightness homogeneity, to construct more complex, two dimensional descriptors.

Bibliography

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

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

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

Mumford, D. and Shah, J. (1989). Boundary detection by minimizing functionals. *Image Understanding 1989*, 1:19–43.

Nitzberg, M. and Mumford, D. (1990). The 2.1-d sketch. In *Proceedings of the Third International Conference of Computer Vision*, pages 138–144.

Parent, P. and Zucker, S. W. (1989). Trace inference, curvature consistency, and curve detection. *IEEE trans. Pattern Anal. Mach. Intell.*, 11.

Perona, P. and Malik, J. (1990). Detecting and localizing edges composed of steps, peaks and roofs. In *Proceedings of the Third International Conference of Computer Vision*, pages 52–57, Osaka. IEEE Computer Society.

Richardson, T. J. (1990). Scale independent piecewise smooth segmentation of images via variational methods. Technical Report LIDS-TH-1940, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology.

Zucker, S. W., David, C., Dobbins, A., and Iverson, L. (1988). The organization of curve detecion: Coarse tangent fields and fine spline coverings. In *Proceedings of the Second International Conference of Computer Vision.* IEEE Computer Society.