CONSTRAINTS AND THEIR SATISFACTION IN THE RECOVERY OF LOCAL SURFACE STRUCTURE

Jean W. Lagarde

Department of Electrical Engineering
McGill University

February 1997

A Thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfilment of the requirements for the degree of
Master of Engineering

Abstract

This thesis deals with the problem of recovering the local structure of surfaces from discrete range data. It is assumed that this recovery is done mostly in a bottom-up fashion, that is, without the help of a priori knowledge about the viewed surface.

Because the problem is ill-posed, we nevertheless need to place constraints on the recovered structure to get a unique solution. In a bottom-up approach, these constraints must come from generic assumptions that apply to all surfaces.

Many methods of bottom-up surface reconstruction have been proposed up to now, some of them dealing with intensity surfaces, some with range surfaces. Each of these methods either explicitly or implicitly applies a set of constraints on the data. The way in which the constraints are applied also varies from method to method. The main contribution of this thesis is some success at unifying a number of those methods under a common formalism of energy minimization, which will permit to better compare the choice of constraints between methods. We also show that the most successful surface reconstruction methods form idempotent operators, which we argue is to be expected.

One method, Sander's curvature consistency, is studied in more detail than the others because it has not been studied much elsewhere yet.

TABLE OF CONTENTS

LIST OF FIGURES

Abstract

This thesis deals with the problem of recovering the local structure of surfaces from discrete range data. It is assumed that this recovery is done mostly in a bottom-up fashion, that is, without the help of a priori knowledge about the viewed surface.

Because the problem is ill-posed, we nevertheless need to place constraints on the recovered structure to get a unique solution. In a bottom-up approach, these constraints must come from generic assumptions that apply to all surfaces.

Many methods of bottom-up surface reconstruction have been proposed up to now, some of them dealing with intensity surfaces, some with range surfaces. Each of these methods either explicitly or implicitly applies a set of constraints on the data. The way in which the constraints are applied also varies from method to method. The main contribution of this thesis is some success at unifying a number of those methods under a common formalism of energy minimization, which will permit to better compare the choice of constraints between methods. We also show that the most successful surface reconstruction methods form idempotent operators, which we argue is to be expected.

One method, Sander's curvature consistency, is studied in more detail than the others because it has not been studied much elsewhere yet.

Sommaire

Cette thèse considèrele problème de la récupération de la structure locale des surfaces à partir de données télémétriques discrètes. Il est supposé que cette récupération est faite principalement du bas vers le haut, c'est-à-dire, sans l'aide de connaissances preétablies sur la surface observée.

Parce que le problème est mal-posé, nous devons cepandant placer des contraintes sur la structure récupérée, de manière à obtenir une solution unique. Dans une approche du bas vers le haut, ces contraintes doivent venir de suppositions génériques qui sont applicables à toutes surfaces.

Plusieurs méthodes de reconstruction de surface ont été proposées jusqu'ici, certaines traitant avec des surfaces d'intensité, certaines avec des surfaces de mesures télémétriques. Chaqune de ces méthodes appliquent un ensemble de contraintes sur les données, que ce soit explicitement ou implicitement. L'approche utilisée pour appliquer ces contraintes est différente d'une méthode à l'autre. La contribution principale de cette thèse est quelque succès à l'unification d'un nombre de ces méthodes sous un formalisme commun de minimisation d'énergie, ce qui va permettre de mieux comparer le choix de contraintes entre les méthodes. Nous montrons également que les méthodes de reconstruction de surface qui ont le plus de succès sont des opérateurs idempotents, ce à quoi on devrait s'attendre, selon nous.

Une des methodes, celle de la compatibilité des courbures de Sander, est étudiée en plus de détails que les autres, parce qu'elle n'a pas été beaucoup étudiée ailleurs jusqu'ici.

Acknowledgements

First and foremost, I would like to thank my thesis supervisor, Frank Ferrie, for making my two years (and more) at McRCIM very worthwile. As his first graduate student, I would like to encourage him to keep involving his other students in his research as much as he involved me.

I thank Peter Whaite and Lee Iverson for the programming support they have provided me with from time to time, and the whole McRCIM gang in general for all the stimulating discussions.

It is obvious to me that I would not have accomplished this work in two years without the support of my parents. I have the feeling that I used them as a bed and breakfast service for a while, but I know that we were all happy to see each other even during the short times I was home.

This thesis would not have been possible if the Department of National Defence had not allowed me to undertake graduate studies. I thank those who made the regulations that gave me the time when I needed it. I also acknowledge the financial support provided by an NSERC scholarship.

CHAPTER 1

Introduction

This thesis deals with the problem of recovering the local structure of surfaces from discrete range data, as a bottom-up process.

Many methods of bottom-up surface reconstruction have been proposed up to now, some of them dealing with intensity surfaces, some with range surfaces. Each of these methods either explicitly or implicitly applies a set of constraints on the data. The way in which the constraints are applied usually varies from method to method. The main contribution of this thesis is some success at unifying a number of those methods under a common formalism of energy minimization, which will permit to better compare the choice of constraints between methods. We also show that the most successful surface reconstruction methods are idempotent operators, which we argue is to be expected.

One method, Sander's curvature consistency, is studied in more detail than the others because it has not been studied much elsewhere yet.

1. Origins of this work

In his doctoral thesis Peter Sander [**56**] presented a novel method for recovering the differential structure of surfaces from noisy discrete data. One major difference with other surface reconstruction work is that his method dealt with *three-dimensional images* produced from magnetic resonance imaging, rather than with graph surfaces such as intensity or range images.

Sander suggested, as a possible research topic, to investigate the applicability of his method for rangefinder data. The method should apply straightforwardly, because range data can be represented as a three-dimensional image in which the surface is restricted to be a graph; the problem is therefore less general and the method directly applicable. Furthermore, the trace is explicitly given in the case of range data, so trace inference is not required, only the recovery of the differential properties of the surface need to be achieved.

One of the aspects of this thesis is the forementionned application of Sander's method to range data. This was to be the main contribution of the thesis, but early in the research, it became quite clear that some fundamental questions were left unanswered in Sander's original work. The application of Sander's method to range data was therefore the spring board to research on more fundamental issues of surface reconstruction in general. The consideration of these issues (to be detailed later) is now the main contribution of this thesis. The usual review of previous work on the subject, normally part of the introduction, will also be part of the main body of this thesis. It will be shown that most major paradigms for surface reconstruction can be unified through the use of an energy minimization formalism.

2. Background

2.1. Surfaces define the 3D world. Vision (or perception in general) permits an agent to interact with the world that surrounds it. Here is an enumeration of some commonly cited visual tasks:

- Object recognition,
- object manipulation,
- obstacle avoidance,
- highly specialized tasks such as reading, recognizing faces, etc.

Depending on the task at hand, the world might be more or less restricted. For example, when reading text, one may consider his visual world as being strictly two-dimensional. In general, however, our world is three-dimensional, and objects within it occupy volume. Because objects are usually opaque, most of what we can visually perceive of them is their surfaces, which are the two-dimensional boundaries that separate matter types. This is why the perception and the representation of surfaces is a good starting point for the perception of objects in general, as was first strongly proclaimed by Gibson [**28**, p.8#1].

2.2. Cooperation between curve and surface sensing. There has been and still is a lot of research in the perception of two-dimensional curves in intensity images [**54, 51**] [**46, 42, 4**][**34**]; this research is interesting in its own right, because as for the reading example above, there are a number of visual tasks that can be restricted to a two-dimensional world. But methods that are successful in some dimensionality are not necessarily easily extended to higher dimensions,[1] so there is a need to tackle the perception of surfaces directly.

Surface reconstruction and curve reconstruction must act cooperatively. The occluding contour of a viewed object (a 2D curve on the image plane) gives information about the surface of that object [**5, 52**]. In the other way, the structure of an intensity surface or of a range surface may be used to find 2D information, such as the contours of characters embossed in a metal plate.

[1]For example, Leclerc [**36**] designed a very effective method for finding discontinuities in graph curves. His extension of the method to graph surfaces necessitated much too many constraining assumptions to be practical, however.

2.3. Intensity versus range surfaces. The now widespread availability of range-sensors gives researchers the possibility to work on data that directly relates to the geometry of surfaces. The geometric perception can therefore be studied without the need to consider surface reflectance and light source descriptions.

A number of the surface reconstruction paradigms studied in this thesis were designed to reconstruct intensity surfaces rather than range surfaces. Intensity surfaces do not directly represent the geometry of the scene which generated it. For example, surface markings will produce features in the intensity surface but not in the range surface. Nevertheless, we will treat the problem of intensity surface reconstruction and of range surface reconstruction as equivalent for the purpose of this thesis.

2.4. The recovery of local surface structure as a bottom-up process. It is accepted that vision makes a great deal of use of high level reasoning. Perception, a "controlled hallucination" [**17**], uses general contextual knowledge to restrict and to guide recognition. Nevertheless, it is also accepted that there is a fair part of perception that is bottom-up (rather than top-down). [**10**] cites psychophysical studies suggesting that

> . . . in the first stage of visual recognition the image is categorized on perceptual grounds only, whereas a perceptual category is given semantic content in the second stage.

The bottom-up parts of perception use generic assumptions that do not rely on a specific context in order to make certain features of the world explicit, and available for higher level processes to use. These low-level processes can be "hard/wet-coded", and therefore more efficient. Biological support for such low level processes has been found. For example, orientation, disparity, motion, and curvature detectors have been detected [**32, 20**], which all make explicit some features implicitly present in the visual field.

In this thesis, the recovery of local surface structure will also be considered as a low-level, data-driven process, which should nevertheless be open to suggestions or constraints imposed by a high level process. An example of such top-down influence on low-level processes can be seen in [**38**, Fig. 14, plate 2].

2.5. Semantics. Finally, let us interpret the meaning of "recovery of local surface structure" in the context of this thesis. Seen as a low-level process, it is to make explicit features that are only implicitly present in the input data. The features that relate to the local (geometric) structure of surfaces are of two kinds. First are the differential properties, up to a certain order, of the continuous parts of the surfaces. Seen as geometric entities, the first two low order differential properties translate to surface orientation and surface curvature. Second are the surface discontinuities, also up to a certain order. Steps and creases (height and orientation discontinuities) are usually considered [**59, 26, 13, 29**], as well as curvature discontinuities sometimes [**49, 36**]. Global properties, such as how a surface may enclose a volume, are not considered here.

The expression "surface reconstruction", which I have already used above, is more popular in the literature. It is less specific than the previous expression, and its meaning varies slightly from one author to the other. However, because it is simpler to write, I will use it here to mean the same thing as the recovery of local surface structure.

3. Constraints and their satisfaction

Given that the initial data consists in (possibly noisy) samples of a surface, and that the end product of our low level process is a local differential structure, there still lacks something to the problem definition. In terms of regularization theory, the problem is still *ill-posed* [**47**]. Informally put, there could be a very large number of answers explaining any particular set of data. If the goal is to get stable and meaningful solutions, additional constraints must be added to enforce stability and "meaningfulness", but the information content of the data must not be lost. For example, it is natural to assume that "nothing extraordinary" happens to the surface in between the points that were sampled, even if anything could happen in reality.

In this thesis, we consider the surface reconstruction problem as one of constraint satisfaction. That is, in order to make the ill-posed problem well-posed, we constrain the answer to lie within a region (a *constraint subset*) determined by a set of constraints, and for given data, we choose the point in the subset closest to that data. Such a closest point problem forms an *idempotent* operator, and we argue that idempotency is a required property of surface reconstruction algorithms.

Once a problem is well-posed, one remains to find the now unique solution to it. Even if the solution is known to exist, to be unique and to be stable, it is not necessarily simple to find. For example, the traveling salesman problem is well-posed, but it is also NP-complete. The satisfaction of the constraints must therefore be attained by the use of some computationally practical algorithm, and many such algorithms have been proposed in the computer vision literature. Most of these methods do not express the problem as one of constraint satisfaction, and a number of them do not dissociate between the problem statement and the problem solution process. Within our framework, the dissociation is straightforward: The problem solution is the minimization of some functional, and the problem statement is the functional to minimize and the type of minimization to perform.

4. Organization of the thesis

The formalism to be used throughout this thesis when talking of constraints and their satisfaction is presented in the next chapter. It is based on constraint subsets and energy minimization. Detailed analysis of the work of Sander has not been published on a large scale, and its adaptation to range imagery has only been given an overview in papers using the method [**22, 23, 24**]. It is therefore a good choice as a case of a surface reconstruction method to be described in detail. This is done in chapter 3. The description of Sander's method is followed by a comparison with other methods in chapter 4. It is shown in that chapter that a number of methods are actually equivalent. The conclusion first summarizes the findings of the preceding chapter.

Then, suggestions for future work that could originate from this thesis are presented. A number of appendices are included. A review of differential geometry is relatively standalone. The other appendices relate directly to the contents of some chapters. These will be introduced within the relevant sections.

5. Contributions

The two most important contributions of this thesis are the unification of popular surface reconstruction methods, based on the functional minimization framework, and the proof of idempotency of some surface reconstruction methods. Also, the adaptation of Sander's locally constant curvature updating to 2–$1/2$D (range) data and an in-depth analysis of the method are worthwile contributions.

CHAPTER 2

Idempotent operators, constraint subsets, and energy minimization

1. Introduction

This chapter presents the formalism that will be used in the following chapters in order to analyze and compare different surface reconstruction methods. This formalism uses constraint satisfaction and energy minimization. Many surface reconstruction methods are not formulated as energy minimization problems from the outset, and we consider useful that a number of *generating theories*, consisting of the problem definition, context, starting assumptions, that lead to a particular formulation, be presented. These different generating theories can provide insight into the interpretation of the basic problem. Such theories include physical models such as membrane and plates under tension [**60**], [**13**], stochastic models such as [**26**], information theoretic models such as Minimum Description Length (MDL) [**36**], evolutionary theories such as dynamic shape theories [**35**], etc. However, once a theory has been established, a paradigm designed to solve a problem within this theory, and computer code generated to implement the paradigm, say, on a Turing machine, one notices that there are much less different patterns in the code as there are generating theories. It is in an attempt to reduce a number of theories into a small number of problem classes that we use constraint satisfaction and energy minimization as the formalism in which to view surface reconstruction problems.

Also, we claim that a surface reconstruction operator should be *idempotent*. Under some conditions, an idempotent operator can be expressed as an energy minimizer, so it can be expressed within our formalism. Not all surface reconstruction methods that have been proposed have this property.

In this chapter, we first present the notion of idempotency and its properties. Then, the associated notion of constraint subset is introduced, followed by examples of idempotent operators. In the following section, we describe three classes of energy minimization problems and show how they relate together and to idempotent operators. These classes will be used as a basis for comparing different surface reconstruction methods in the following chapter.

2. Idempotency requirement

A surface reconstruction algorithm may be considered as an operator[1] in functional analysis terminology. An operator has a domain and a range, such that it performs a mapping from an element in its domain to an element in its range. In this chapter, it is argued that one of the required properties of an operator that is to perform surface reconstruction is *idempotency*. An operator g is idempotent when its composition with itself yields itself[2]

$$(g \circ g)(\mathbf{d}) = g(\mathbf{d}). \tag{2.1}$$

The argument in favor of the idempotency requirement is informal: If the goal of the operator is to recover a valid interpretation from initial data, then applying the process one more time on the supposedly valid interpretation should not change it. Otherwise, how can we say the first interpretation was valid in the first place?

A *perpendicular projection* is an operator that is both idempotent and *self-adjoint*. [**61**] was among the first to advocate that a signal restoration operator should be a perpendicular projection. More recently, [**45**] argues that edge detection should be a projection, and [**25**] enforce an integrability constraint as a projection, in a shape from shading algorithm.

3. Constraint subsets

If a surface reconstruction operator is idempotent, then we will denote its range as its *constraint subset*[3], because the function constrains the answer to lie within that set, and does not modify arguments that already lie in it. We use "subset" rather than "set" because the range of an idempotent operator is a subset of its domain. If a point lies outside of the range of the operator (but is in its domain), the operator will cause the answer to lie in the constraint subset (its range). If the argument of the operator lies in its range, then the operator will act as the identity operation, because it is idempotent.[4] In other terms, all the points in the range of an idempotent operator are "fixed points" of the operator [**1**]. If a surface reconstruction operator is not idempotent, then we cannot identify its range with a constraint subset: Even if the argument to the function lies in its range, the result is not necessarily equal to the argument.

For the surface reconstruction function to make sense, the meaning of elements of its domain must be the same as the meaning of the elements of its range. This is not to say that the domain and the range must have the same dimensionality in the initial problem statement. Rather, the cardinality of the range must be lower than or equal to the cardinality of the domain [**37**]. A few examples are in order.

---

[1] An operator is more general than a functional or a function. It will therefore be used unless we want to be more specific.

[2] See [**37**] for a review of functional analysis.

[3] Which is not quite the same as the *constraint set*, or *feasible region*, of nonlinear programming [**33**, p. 1096].

[4] The proof is trivial. If x lies in the range of f, then there exists a y in its domain such that $f(y) = x$. By the idempotency of f, $f(f(y)) = f(x) = x$, which completes the proof.

In [**45**], the function is to produce a binary feature map corresponding to the edges and lines of the input image. In this case, even if the input image is not a binary image, the result is nevertheless a valid image that consists only of binary lines. Applying the function once more on this image should therefore give the same result back.

The function $f : \Re^n \rightarrow \Re$ consisting in taking the norm of a vector in $\Re^n$ lowers the dimensionality of the result to one. However, the meaning of the components of the argument and of the result may be the same. For example, the average of a set of heights is a height. In order to represent this function as an idempotency, we view the result as the parameter t of the one-dimensional manifold $\mathbf{u}(t) = (t, t, \ldots, t)$, $\mathbf{u} \in \Re^n$. Taking the average can then be seen as the perpendicular projection of the argument onto the constraint subset defined above. The operator can then be applied on its result, which produces no effect, since the argument now already satisfies the constraint (that all the components of the vector be equal).

Now, let us consider an operator which has a range with a higher dimensionality than its domain. This is an important consideration in surface reconstruction in the case of sparse data and multi-sensor integration. In this case, I slightly generalize the definition of idempotency. As long as a subspace of the range corresponds to the domain, it is possible to test for a *generalized idempotency* of the operator. For example, consider an operator that takes as argument a vector $\boldsymbol{y}$ representing the depth of points sampled on a planar curve, and that produces a piecewise continuous polygonal estimate of the underlying curve, while satisfying some set of constraints. If the input consists of n depth values, then the result will consist in n new depths and in the n orientations of the polygonal segments. If the operator is idempotent with respect to the depth information, applying the operator on the resulting depth should give the same polygonal estimate. In this example, the reason for the idempotency requirement is again clear. If the goal is to obtain the best (under a fixed set of constraints) n segment piecewise continuous polygonal approximation to the underlying curve, it would not make sense that the data generated by the first application of the operator generate a different approximation upon a second application of the constraints.

Here are some more properties of constraint subsets. A constraint subset does not have to be a smooth manifold in the domain $A \subseteq \Re^n$. It could consist of the union of any number of sets of any dimensionality, as long as they can be embedded in A. For example, it could consist of a C^k regular surface of dimension $m < n$, either bounded or of infinite or partially infinite extent. It could consist of a subset of $\Re^n$, of dimension n. It could also consist of a disconnected set of m points in $\Re^n$. Figure 2.1 shows examples of the three. The cardinality of the constraint subset is one indication of how information reducing the operator is, but the "volume" occupied by a bounded subset is also informative in the case of an n dimensional subset of $\Re^n$, because the cardinality of both these sets is the same [**37**].

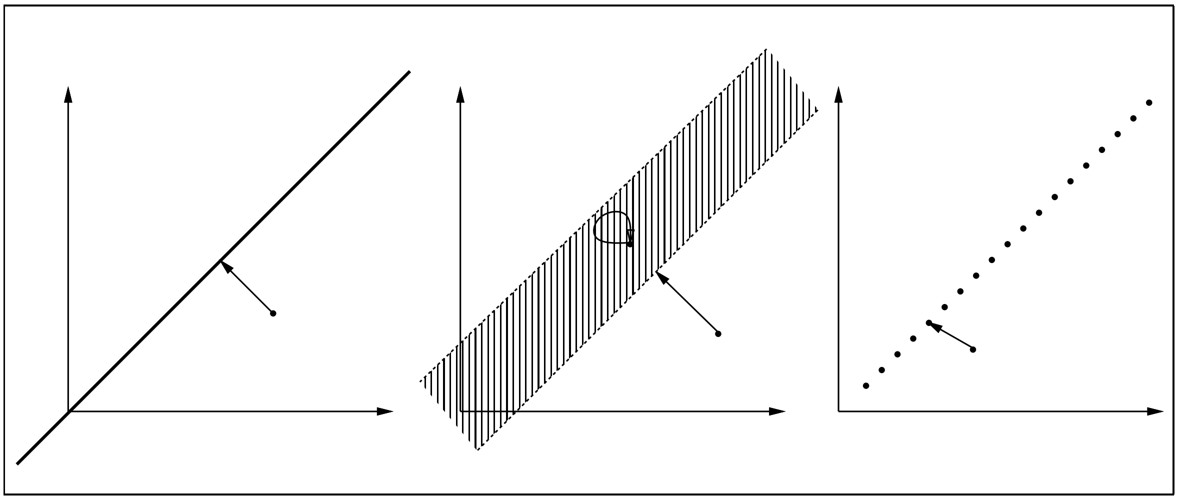


FIGURE 2.1. Examples of constraint surfaces. From left to right, a manifold in $\Re^n$, an n dimensional subset of $\Re^n$, and a set of disconnected points in $\Re^n$. The arrows represent the solutions of closest point problems on these constraint subsets.

4. Some types of idempotent operators

The *perpendicular projection* of a point $\boldsymbol{d}$ on a convex constraint subset is idempotent. When a set is not convex, $\boldsymbol{d}$ may perpendicularly project on more than one point, and the operator is therefore not well defined.

Given an arbitrary constraint subset, the *closest point* operator is the one that finds the point in the constraint subset closest to the data point $\boldsymbol{d}$. The closest point problem for a convex constraint subset is equivalent to the projection operator. Figure 2.1 gives examples of the closest point operator for different types of constraint subsets. If the constraint subset is not convex, it is possible that a data point has more than one closest point on the constraint subset. The operator is unstable at these points.

When restricted to Euclidean spaces, the closest point operator is of limited use. We borrow on the terminology of field theory [**43**] to conceptualize the closest point operator in any curvilinear orthogonal coordinate system. In such a system, we designate one set of coordinates as *flux lines*, and all the other coordinates as *equipotential*, with the constraint subset having a constant potential by definition. From a data point in this space, the idempotent operator consists in following the flux line from the data point to the constraint subset.

5. Three classes of energy minimization

We now present the energy minimization problem, and divide it into three classes. The division between the classes depends both on the problem to be solved and on the approach used to solve it. That is, the same problem usually can be formulated by more than one class of energy minimization.

5.1. Minimization of convex energies. The simplest energy minimization problem is the minimization of convex energies. Numerical methods for such minimizations abound, and the process is well understood [**64**]. It is sometimes possible to solve the problem analytically, especially when the parameter space is small. In most cases, however, numerical methods are more practical. Some idempotent operations can be formulated as minimization of convex energies. For example, perpendicular projection of a point $\boldsymbol{d}$ on a convex manifold $\boldsymbol{r}(\boldsymbol{u})$ can be expressed as $min_{\boldsymbol{u}}(\boldsymbol{d} - \boldsymbol{r}(\boldsymbol{u}))^2$, where the function to minimize is convex. This problem can be solved numerically using gradient descent approaches, that use local differential properties of the energy function, such as its first, and even second, partial derivatives. Note that the problem is completely defined by the energy function to minimize, and that this energy function combines both the constraints of and the argument to the operator.

5.2. Global minimization of non-convex energies. The second family of energy minimization is the most difficult to solve, and many methods of surface reconstruction currently fall in this energy minimization class. Problems in this class consist of finding the global minimum of a non-convex energy function. For the problem to be well posed, there must be no more than one point with the minimum value of the function over the domain of interest. What differentiates between this class and convex minimization problems is the problem itself. The formulation of the minimization is the same, but some problems will produce convex energies, while some will not. As for convex energy minimizations, the argument to the operator is part of the energy function, in this case.

When the parameter space is small, it is possible to solve this problem exactly using dynamic programming approaches [**2**]. However, this approach becomes inefficient very fast as the dimensionality of the parameter space and the required accuracy increases. Numerical methods fall mostly in two categories. One is simulated annealling, which performs a kind of informed random search on the energy function [**26**]. The other approach has actually been developed independently for computer vision algorithms [**13**], and is generally known as the *continuation method* [**1**]. It consists in smoothing the energy function to make it convex, and to track the minimum as the function is gradually unsmoothed back to its original shape. The tracking is similar to zero crossing tracking in the scale-space of Witkin [**65**].

The closest point in a constraint subset $\boldsymbol{r}(\boldsymbol{u})$ to a point $\boldsymbol{d}$ can be found by solving $min_{\boldsymbol{u}}(\boldsymbol{d} - \boldsymbol{r}(\boldsymbol{u}))^2$. If $\boldsymbol{r}(\boldsymbol{u})$ is not convex, this problem is a non-convex global minimization.

5.3. Local minimization of non-convex energies. This is the second simplest energy minimization problem to solve. In this case, the problem data consists of a possibly non-convex energy function, together with a starting point in the parameter space of the function, called an *initial estimate* to the solution. The problem is to find the minimum of the function local to that point. Local, in this case, may have two meanings. In one case, the local minimum is considered as the minimum *closest* to the initial estimate in the parameter

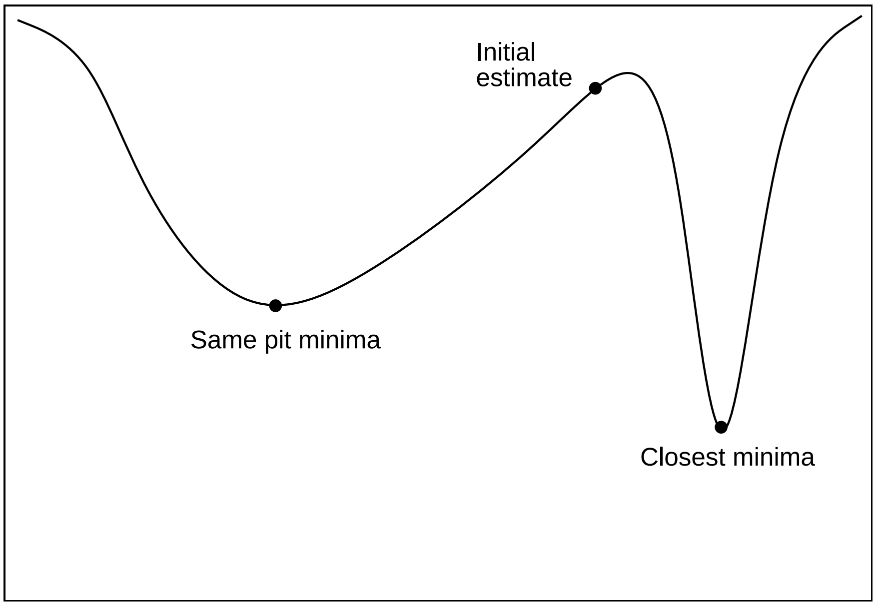


FIGURE 2.2. The closest local minimum to the initial point is not necessarily the same as the same-pit local minima, as this example shows. The point to the left is much further from the initial estimate than the point to the right, but it lies in the same pit as the initial estimate.

space. We call this problem the *closest minimum* problem. In the other case, the solution is the local minimum that one would obtain by gradient descent from the initial point. We call this problem the *same-pit minimum* problem. Figure 2.2 shows that these two problems do not necessarily give the same solution. If the energy function has some symmetry with respect to its minima then both problems may be equivalent, as is the case for energy defined by a *distance transform* [**14**]. In any case, the same-pit minimum makes more sense as an energy minimization problem because the closest point minimum makes no use of the energy function else than to find the set of local minima. In terms of problem complexity, the same-pit minimum problem is equivalent to a convex minimization, since it can be solved by using only local properties of the energy function. The second case is not as easy because the local structure of the energy function does not provide any information as to the direction where the solution lies.

The same-pit energy minimization will be the one used for the local energy minimization class in the rest of this thesis. It is trivial to show that a local energy minimization is an idempotent operator. For the closest point problem, for example, the corresponding energy function, mentioned above, is the distance transform of the constraint subset. Contrary to the previous classes, the function to (locally) minimize is only defined by the constraints, and not by the argument to the operator. The argument is taken as the initial estimate, that determines the "local pit" of the energy function.

5.4. Comparison between global and local energy minimization methods. The difference between the global and the local minimization class is partly in how the problem is posed. We saw in section 4 that a local energy minimization problem can be expressed as a distance transform in an appropriate curvilinear orthogonal coordinate system. Any local minimization can therefore be posed as a global minimization by constructing the parametric equation $\boldsymbol{c}(\boldsymbol{t})$ of the boundary of the constraint subset in the distance transform space, and by expressing the problem as

$$\min_{\boldsymbol{t}}(\boldsymbol{d} - \boldsymbol{c}(\boldsymbol{t}))^2. \tag{2.2}$$

Figure 2.3 illustrates this duality between global minimization and local minimization problems.

However, not all global energy minimization problems, even convex ones, can be expressed as local minimizations, because these operators are not necessarily idempotent, and we know that a local minimization problem is idempotent. A local minimization is much simpler to solve than a non-convex global minimization, but the local energy function itself might not be easily computable, however.

In local minimization problems one needs an initial estimate of all the features to be explicitly produced by the operator. However, some features can be implicitly represented. For example, the position of discontinuities in a surface can be derived from the junction of the Monge patch surfaces that were explicitly computed, as is done in [**36**]. In contrast, global minimization methods permits for data that is not explicitly provided to be explicitly produced, such as the line process in [**13**].

5.5. Stability. In any problem that has to be solved numerically comes the question of stability. Even if a minimization is solved exactly, there can be instabilities inherent to the problem itself, for certain data. Slight changes in unstable data may produce very large changes in the result, and that may be considered as a type of ill-posedness [**47**].

In terms of an energy minimization framework, numerical stability has to do with the shape of the energy functional near the global or the local minimum that is to be found. A very small curvature in the neighbourhood of the minimum makes the exact localization of the minimum hard to find numerically, even if it is actually well defined. A gradient descent process will tend to wander around the minimum if it sits in an almost flat area. In surface reconstruction, the minimization takes place in a multidimensional space where each dimension usually represents a particular pixel site in the image. In that case, the minimum may be very localized along some axes, but not along others.

The data instabilities differ in their instantiation depending on the type of energy minimization. No such instability can exist for a convex minimization, because such a minimization does not take any decision [**13**]. In the case of a global but non convex minimization, data that is unstable (with respect to the problem to solve) will have some of its local minima at almost the same value (or even at the same value) as the "global" one, in at least one direction in the problem space. The process looking for the global minimum will therefore not know which one to choose. In local minimization, the energy functional does not change with the data. Therefore, it is the position of the data point on the energy functional that determines its stability. Data will

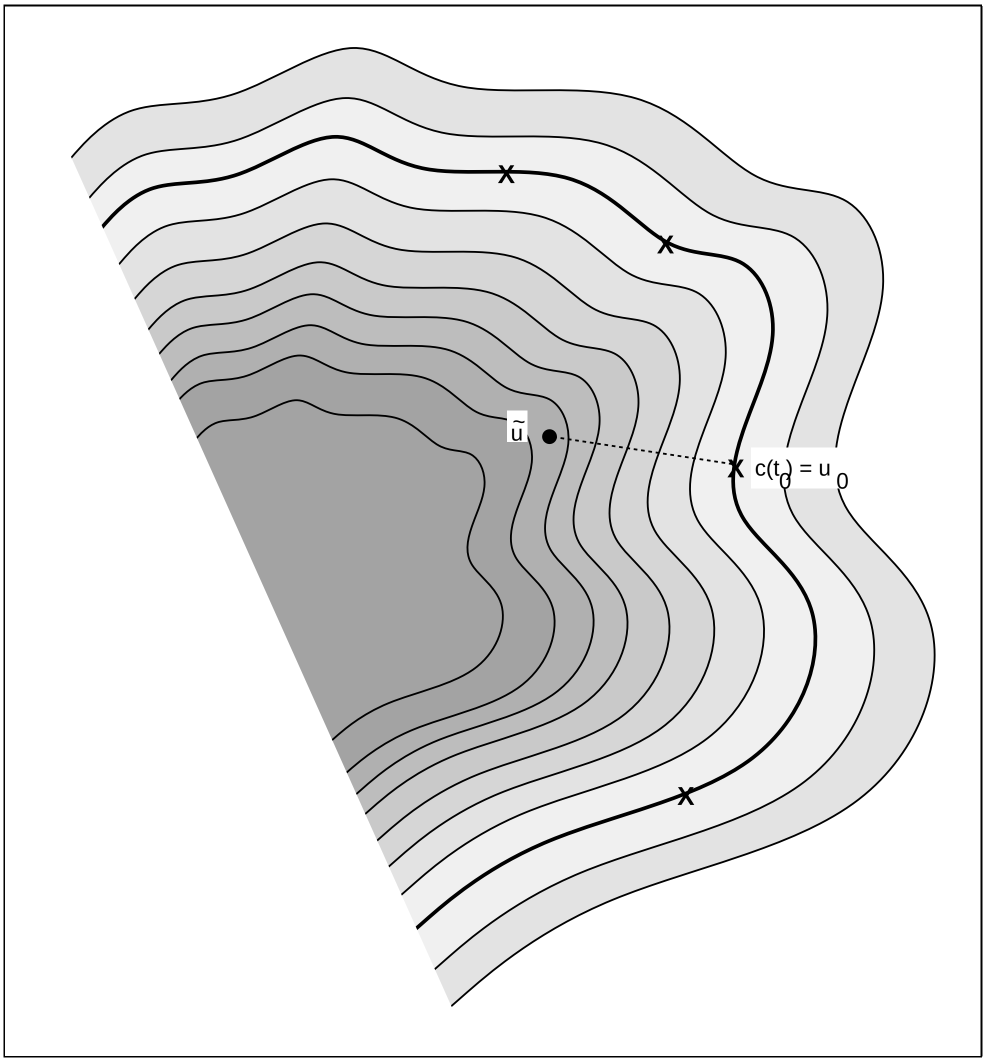


FIGURE 2.3. Relation between global minimizations and local minimizations. The point $\tilde{\boldsymbol{u}}$ is the data. The thick curve is the constraint subset, expressed parametrically as $\boldsymbol{c}(\boldsymbol{t})$: only points on this surface satisfy all the constraints. The problem is to find the closest point $\boldsymbol{u}_0$ to $\tilde{\boldsymbol{u}}$ that lies on the constraint surface. One way to find $\boldsymbol{u}_0$ is to build the distance transform of the constraint surface, pictured by the shaded surface with level curves, and to find the local minimum of this function. This is done by moving from $\tilde{\boldsymbol{u}}$ along the path of steepest descent (the dashed line) to the solution. The other way is to find the global minimum of $(\tilde{\boldsymbol{u}} - \boldsymbol{c}(\boldsymbol{t}))^2$ in terms of $\boldsymbol{t}$. However, the function to minimize is non-convex (the crosses indicate where the local minima would occur).

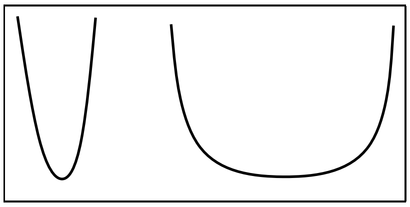


FIGURE 2.4. Numerical stability in one dimension. The minimum of the left energy will be much more stable than the minimum of the right energy.

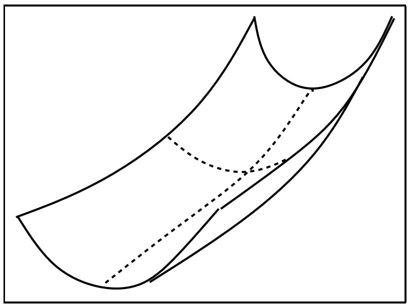


FIGURE 2.5. Numerical stability in two dimensions. The minimization will be more stable in the highly curved direction than in the other direction.

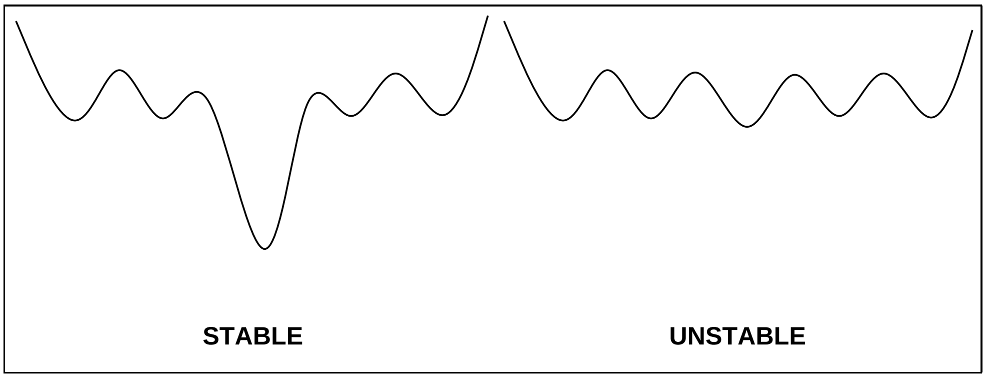


FIGURE 2.6. Stability in global minimizations.

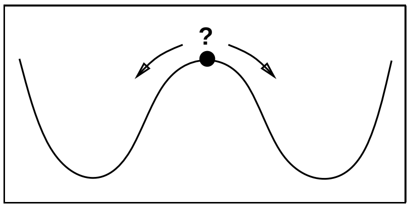


FIGURE 2.7. Stability in local minimizations.

be unstable if it is at a local *maximum* of the functional, along at least one direction in the problem space. In that case, a gradient descent approach can decide to proceed on either side of the ridge, giving two completely different results.

What I have termed numerical instabilities and data instabilities are in fact related. If the local minima of an unstable global energy functional close up together to the point where they touch in the limit, the result will be a large flat minimum, causing numerical instability. Also, when a continuation method is used to "smooth" an unstable global energy functional, the resulting functionals will tend to be flatter, as can be seen in [**36**, Appendix B]. In the case of a local energy functional, if the local minima are connected, then there are no ridges separating them, and the unstability will be a numerical one.

6. Summary

I have put forth arguments for the requirement that a surface reconstruction operator be idempotent. In such a case, the range of the operator can be considered as a constraint subset. An argument that is already in the range of the operator is considered to satisfy the constraints of the operator, and an argument that is not in the range will be made to adhere to the constraint subset.

In order to compare surface reconstruction operators, either idempotent or not, I have adopted an energy minimization formalism. The three classes of energy minimization I consider are global convex minimization, local minimization, and global non-convex minimization. The distinction between the global convex and the global non-convex energy minimizations depend on the type of problem to solve. The distinction between local and global minimizations depends on the formalism adopted to solve a given problem. Local minimization problems always correspond to idempotent operators and can always be expressed as a global minimization problem, although the reverse is not necessarily true.

In the next chapter, we present the locally constant curvature constraint algorithm of Peter Sander [**56**], which I have adapted for surfaces presented as a graph rather than as a density image. This method is presented here in more detail because it has only been outlined in the literature [**23, 22, 24**], and because it involves all the key issues that will be considered in the following chapters. Immediately following is a chapter comparing a number of well known surface reconstruction methods under the formalism of this chapter. In particular, we are interested to know which methods are idempotent operators. Or, equivalently, which methods are associated with a well defined constraint subset.

CHAPTER 3

Sander's variational relaxation

1. Introduction

This chapter presents the adaptation of the locally constant curvature approach of Peter Sander to the reconstruction of graph surfaces. The first part gives an overview of the original work of Sander [**56**]. The rest of the chapter presents its adaptation to graph surfaces and an analysis of the algorithm.

2. Summary of Sander's original work

This section briefly overviews the original work presented in Peter Sander's doctoral thesis [**56**]. The emphasis is on the implementation of the method, as it was applied to magnetic resonance images. The *fibre bundle* formalism introduced by Sander will not be used at all, since it is felt it does not add anything to the understanding of the method.[1]

2.1. Goals. The goals of Sander's method are to infer the trace points (see appendix A) of surfaces in three-dimensional images (analogous to inferring the trace points of curves in standard two-dimensional intensity images), and to estimate the differential geometric properties of these surfaces, in the form of an augmented Darboux frame (to be defined later) at every sample point. Sander's method does not attempt to find step and crease edges in the surfaces, as many other methods do. Rather, the surfaces are assumed everywhere smooth.[2]

2.2. Representation of the surface. The differential geometric properties that are to be explicitly recovered by the method are the surface orientation and principal curvatures [**19**]. Specifically, an *augmented Darboux frame* $\xi_{\boldsymbol{P}}$[3] is stored for every trace point $\boldsymbol{P}$. The frame consists of the coordinate $\boldsymbol{P}$ of its origin, the surface normal $\boldsymbol{N}$ at $\boldsymbol{P}$, the principal directions $\boldsymbol{M}_M$ and $\boldsymbol{M}_m$, and principal curvatures κ_M and κ_m,

[1] Actually, even Sander does not refer much to the formalism outside of the section where it is presented.
[2] Referring to appendix A, smooth is interpreted as C^2, since this is sufficiently regular to perform all the computations of the algorithm.
[3] Actually, *principal frame* [**6**, p. 20] would be a better designation, since Darboux frames do not have to correspond to frames on the lines of curvatures. Nevertheless, this notation will be kept for compatibility with Sander's work.

at $\boldsymbol{P}$ [**56**]. The vectors $\boldsymbol{N}$, $\boldsymbol{M}_M$, and $\boldsymbol{M}_m$ are all unit vectors, and form a *direct orthonormal frame* in $\Re^3$. Although the principal directions are represented as vectors, the only information content is in their direction. Appendix A reviews the relevant differential geometric concepts.

2.3. Initial Estimates. It is necessary for this method to have initial estimates of all the quantities to be produced by the algorithm. This does not automatically prove that the method falls into the *local minimization* family. The estimates are the coordinates of the trace points, and their associated differential properties in the form of their augmented Darboux frames, to be described later.

The initial estimate of the position of the trace points and of their associated normals is obtained in a way similar to the use of edge operators on two-dimensional images, except that the operators are now three-dimensional. Details can be found in [**67**]. Basically, the image is convolved with a 3D edge operator at a set of discrete orientations. The image positions at which an operator response is above a threshold, and which survive a local maxima selection are chosen as the trace points, and the normal is taken as the orientation of the operator which produced the response. It still remains to estimate the principal curvatures and directions. These could be found by estimating the second order derivatives using difference operations over the trace points, but Sander argues that better estimates can be achieved by using a more robust surface fitting method [**56**, Chapter 3]. From differential geometry, we know that the osculating paraboloid to a surface has same principal curvatures and principal directions at the point of osculation [**48**]. The required quantities are therefore estimated by fitting a paraboloid to the putative trace point and its neighbours. The fit includes both positional and surface normal information.

2.4. Refining estimated differential properties. The initial estimates obtained in the previous section are not yet satisfactory. This can easily be seen in examples, such as the one in figure 3.12. It is well known that signal noise is magnified in the computation of its derivatives, and second order quantities have to be computed in this method. The principal directions are known to be especially noisy [**56**]. The initial estimates are computed locally, and consistency between neighbours is not insured. The initial estimates are therefore further refined by an iterative process which enforces a *locally constant curvature* constraint over the trace points. Sander attempts to do this in a framework similar to the relaxation labelling process used to extract image curves in [**46**] and [**34**]. However, the complexity of the representation precludes the use of a discrete set of labels, as in relaxation labelling. Instead, Sander applies the same local minimization principle using continuous values and variational methods, thus the name *variational relaxation*.

The iterative updating process is now described. Most of the figures will represent planar curves instead of surfaces. This is clearer, and the principles are easily transferred in 3D through the mind's eye. The updating process refines the set of augmented Darboux frames $\xi^i_{\boldsymbol{P}}$ for every point $\boldsymbol{P}$ that is part of the putative trace. The superscript i indicates the iteration at which the Darboux frame was produced. The data estimated from the preceding section provide the updating algorithm with $\xi^0_{\boldsymbol{P}}$. At any given iteration, all the points are updated in parallel.

At iteration $i+1$, the new frame $\xi^{i+1}_{\boldsymbol{P}}$ at point $\boldsymbol{P}$ is determined from the frames at points $\boldsymbol{Q}_\alpha$, that are members of the contextual neighbourhood $\mathcal{N}^i_{\boldsymbol{P}}$ of point $\boldsymbol{P}$ at iteration i. This contextual neighbourhood consists of the points within a sphere of radius r centered on $\boldsymbol{P}$ that satisfy a contextual neighbourhood criteria on the point. This criteria is that $\boldsymbol{P}$ lies within a "thick trace" of width ϵ of the dual paraboloid patch (see appendix A) $\boldsymbol{s}_\alpha$ of $\xi^i_{\boldsymbol{Q}_\alpha}$. Issues concerning the thick trace will be addressed in the next section.

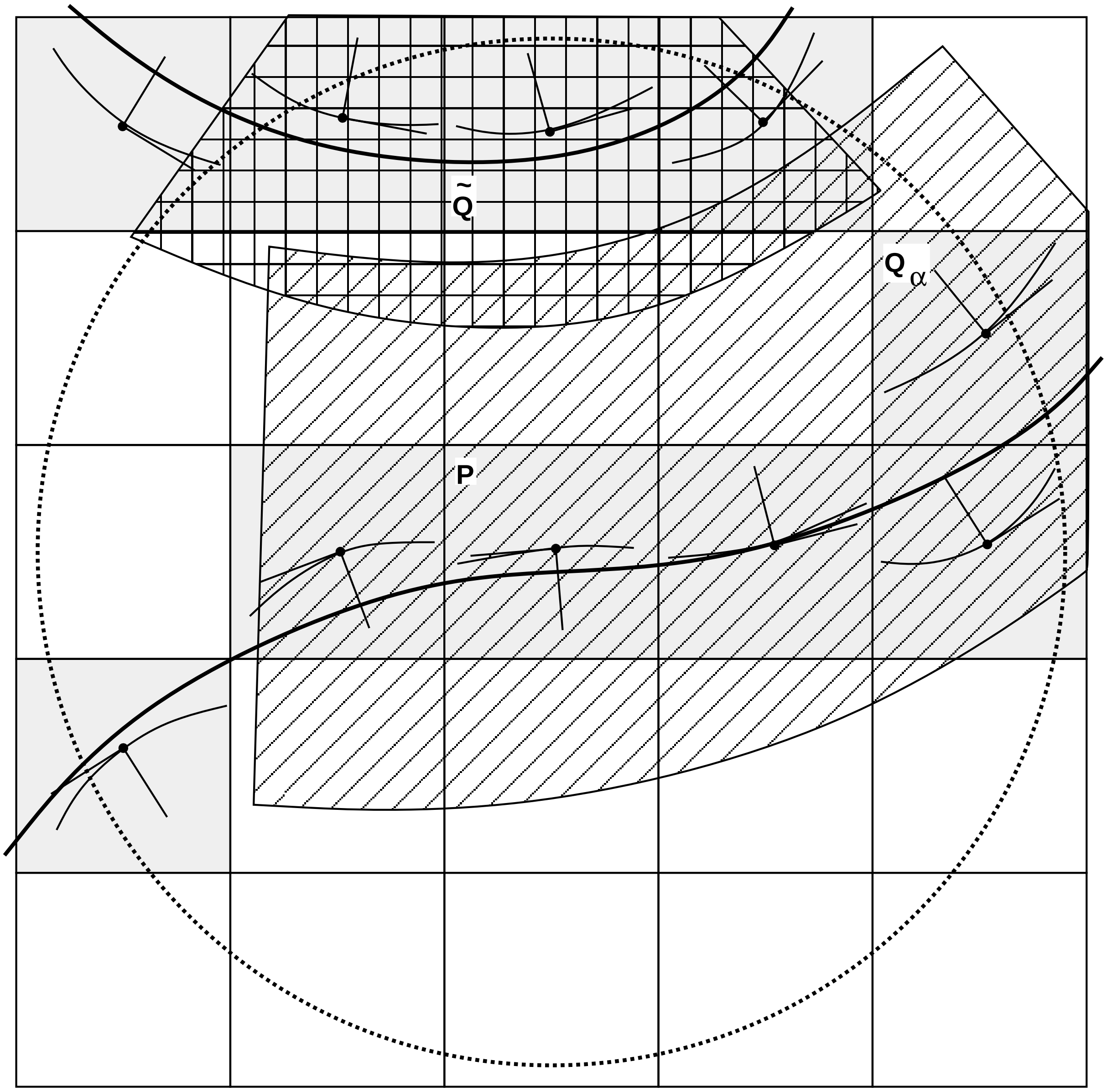


FIGURE 3.1. Darboux frames of the trace points, and how to determine the contextual Neighbourhood.

Figure 3.1 illustrates the ideas presented above. In the figure, the thick curves are the actual surfaces that were sampled using the grid shown. The grey grid elements are the trace of the curve, as determined by the 3D edge detectors. The coordinate axes at every trace point represent the vector components of the Darboux frame, and a section of the osculating paraboloid is included to represent the magnitude of the curvature at the trace points. All the Darboux frames are for iteration i, so the superscripts are not shown.

The contextual neighbourhood of point $\boldsymbol{P}$ is a subset of the trace points included in a sphere of radius r (2 pixels in this case, as indicated by the dashed circle) about $\boldsymbol{P}$. For a neighbouring point $\boldsymbol{Q}_\alpha$ to be included in the contextual neighbourhood, point $\boldsymbol{P}$ must lie in the thick trace of the osculating paraboloid to the surface at $\boldsymbol{Q}_\alpha$. The thick trace of point $\boldsymbol{Q}_\alpha$, which is two pixel wide in this case, is indicated by the hashed pattern. The thick trace at point $\tilde{\boldsymbol{Q}}$, indicated by a grid pattern, does not include point $\boldsymbol{P}$, so point $\tilde{\boldsymbol{Q}}$ is therefore not part of the contextual neighbourhood of point $\boldsymbol{P}$.

Given the contextual neighbourhood, $\xi_{\boldsymbol{P}}^{i+1}$ is obtained by minimizing a functional implementing a locally constant curvature constraint on the neighbours. The first step is to "transport" the Darboux frames at the neighbouring points to the point to be updated. Given $\boldsymbol{Q}_\alpha^i \in \mathcal{N}_{\boldsymbol{P}^i}$, and its associated Darboux frame $\xi_{\boldsymbol{Q}_\alpha}^i$, the updating frame at $\boldsymbol{P}$ from $\boldsymbol{Q}_\alpha^i$, denoted $\xi_{\boldsymbol{P}_\alpha}^i$, is obtained as the Darboux frame of a point of the patch $\boldsymbol{s}_\alpha$. This point on the patch is obtained by projecting $\boldsymbol{P}$ on the patch along the direction of $\boldsymbol{N}_{\boldsymbol{Q}_\alpha}$. In figure 3.2, where the iteration subscripts were omitted, point $\boldsymbol{P}$ is projected on the osculating paraboloid of point $\boldsymbol{Q}_\alpha^i$, along the normal of $\boldsymbol{Q}_\alpha^i$, in order to obtain the updating frame $\xi_{\boldsymbol{P}_\alpha}^i$. The transported frames are shown in dashed lines for $\alpha = 1, 4$, and 5. The arrows indicate the projection of point $\boldsymbol{P}$ onto the osculating paraboloids of points $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_5$, along the normals of the neighbours.

The updated frame $\xi_{\boldsymbol{P}}^{i+1}$ is the one that minimizes the square of the euclidean normed distance to all the updating frames $\xi_{\boldsymbol{P}_\alpha}^i$.[4] The components of $\xi_{\boldsymbol{P}}^{i+1}$ are found separately, as the quantities that satisfy the following least-squared minimizations.

$$\kappa_{MP}^{i+1} = \min_{\kappa_M} \sum_{\alpha=1}^{n} (\kappa_M - \kappa_{MP\alpha})^2, \tag{3.1}$$

$$\kappa_{mP}^{i+1} = \min_{\kappa_m} \sum_{\alpha=1}^{n} (\kappa_m - \kappa_{mP\alpha})^2, \tag{3.2}$$

$$\boldsymbol{N}_P^{i+1} = \min_{\boldsymbol{N},\lambda} \sum_{\alpha=1}^{n} (\boldsymbol{N} - \boldsymbol{N}_{P\alpha})^2 + \lambda(\boldsymbol{N}^2 - 1), \tag{3.3}$$

$$\boldsymbol{M}_{MP}^{i+1} = \min_{\boldsymbol{M}_M,\lambda_1,\lambda_2} \sum_{\alpha=1}^{\hat{n}} (\boldsymbol{M}_M - \boldsymbol{M}_{MP\alpha})^2 + \lambda_1(\boldsymbol{M}_M^2 - 1) + \lambda_2(\boldsymbol{M}_M \cdot \boldsymbol{N}). \tag{3.4}$$

The Lagrange multipliers λ, λ_1, and λ_2 enforce the required conditions on the vector quantities. $\hat{n}$ is the number of non-umbilic points in the contextual neighbourhood of $\boldsymbol{P}$. The value of $\boldsymbol{M}_{mP}^{i+1}$ is directly determined

[4] Actually, the frames $\xi_{\boldsymbol{P}_\alpha}^i$ that are determined to come from *umbilic points* do not contribute to the update of the principal directions.

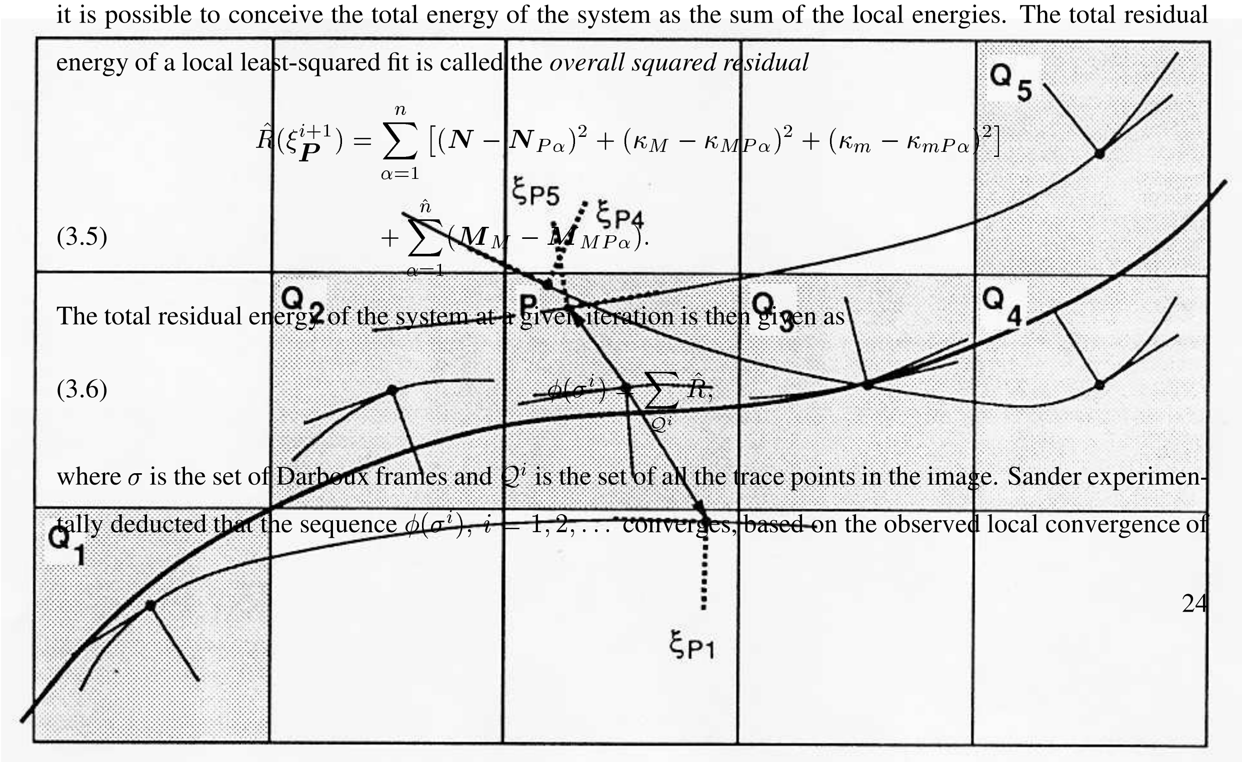


FIGURE 3.2. Obtaining the updating frames from the neighbours.

by cross product since it is required to form an orthonormal frame with $\boldsymbol{M}_{MP}^{i+1}$ and $\boldsymbol{N}_{P}^{i+1}$. Although it is indicated in Sander's thesis [**56**, p. 34] that the set of trace points are also updated, it is not clear how this is achieved (this is not important for the adaptation of Sander's method to range images).

Although the energy minimizations are done locally at each point, and not globally over the entire image, it is possible to conceive the total energy of the system as the sum of the local energies. The total residual energy of a local least-squared fit is called the *overall squared residual*

$$\hat{R}(\xi_{\boldsymbol{P}}^{i+1}) = \sum_{\alpha=1}^{n} \left[(\boldsymbol{N}-\boldsymbol{N}_{P\alpha})^2 + (\kappa_M-\kappa_{MP\alpha})^2 + (\kappa_m-\kappa_{mP\alpha})^2\right] + \sum_{\alpha=1}^{\hat{n}}(\boldsymbol{M}_M-\boldsymbol{M}_{MP\alpha}). \tag{3.5}$$

The total residual energy of the system at a given iteration is then given as

$$\phi(\sigma^i) = \sum_{Q^i} \hat{R}, \tag{3.6}$$

where σ is the set of Darboux frames and Q^i is the set of all the trace points in the image. Sander experimentally deducted that the sequence $\phi(\sigma^i)$, $i = 1, 2, \ldots$ converges, based on the observed local convergence of the residuals $\hat{R}(\epsilon_P^{i+1})$. The number of iterations to be performed by the algorithm can therefore automatically be controlled by a test on the convergence of the global residual.

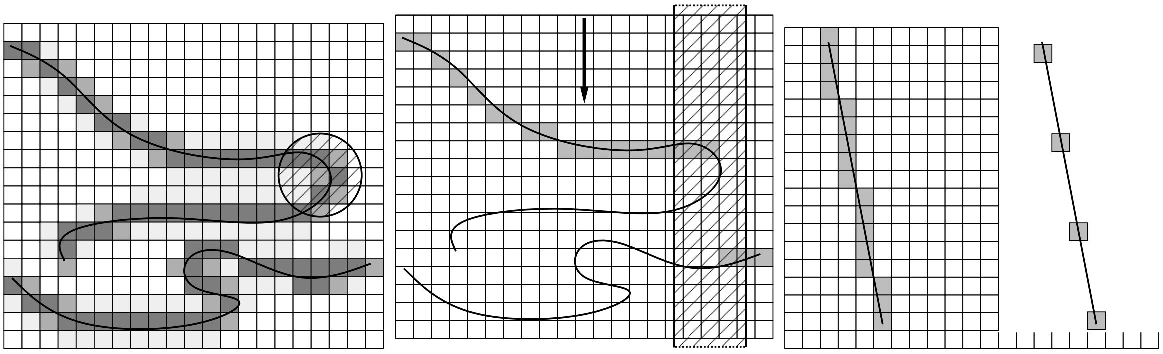
3. Adaptation of Sander's method for range images

In this section, the method previously described is adapted to data presented as a two-dimensional graph, where the trace of the unique surface is given explicitly. The text will specifically refer to data in the form of range images, but the adapted algorithm can deal with other input data, such as surface orientation graphs produced by shape-from-X algorithms [**22**]. Some of the modifications to the original algorithm are due to the difference in the type of data. However, some modifications are to correct deficiencies that also apply to the original algorithm. Whether modifications apply only to the adapted algorithm or to both will be specified in the text. In any case, the modifications keep the method within the original paradigm, that is, an iterative refinement of initial estimates of surface depth, orientation and curvature, using a locally constant curvature constraint.

3.1. The general outline. The outline of the adapted algorithm is mostly the same as the original one. First, initial estimates of the augmented Darboux frame must be found at every point of the map. The difference is that the trace points are already assumed to consist of all the discrete points of the two-dimensional map. Then, the augmented Darboux frames are iteratively updated in parallel, as in the original method. The main difference lies in how the contextual neighbourhood is determined. The transport and updating are conceptually identical.

3.2. Initial estimates. In the case of range data, the initial estimates of the trace points are explicitly given in the data. The initial estimates of surface orientation and curvature can be obtained by using finite difference operations, or by performing local fitting of quadric surfaces, of which the differential properties can be obtained analytically [**6**]. Fitting local quadrics is less sensitive to noise than using finite differences, and using the smallest possible symmetric neighbourhoods does not overly smooth the data. Since the method assumes the surface to be everywhere smooth, there is no need to use robust fitting methods [**7**], [**66**]; least squared fitting is currently used in the implementation. The goal at this stage of the development does not include the detection of discontinuities; however, in order ensure the validity of the smoothness assumption in the experiments, the bounding contours of surfaces may be explicitly given, and the local neighbourhoods are then restricted to smooth connected regions of the surface. The same explicit discontinuity information is also used to prevent cross-boundary updating in the iterative updating.

3.3. Determining the contextual neighbourhood. In the original algorithm, which dealt with a three dimensional grid of pseudo intensity data, the neighbours of the frame were taken from a sphere (actually a cube) centered on the point to be updated. Furthermore, points within the sphere would be included in the



(a) (b) (c)

FIGURE 3.3. Difference between an intensity surface and a range surface (illustrated with 2D curves). In (a) is a continuous surface, together with the discrete intensities it could generate as the output of an "intensity camera". The neighbourhood relation existing at points of such an image is a sphere. In (b) is the same surface sampled by a range camera in the direction of the arrow. The neighbourhood relation for a point is now a cylinder of infinite extent. (c) illustrates that the sampling separation of a regular surface by an intensity camera is bounded, while the sampling separation due to a range camera is unbounded.

averaging only if the point to be updated was inside a *thick trace* of the extrapolating patch of the neighbour. In the case of range data, the neighbourhood rather consists of a disk (actually a square) since the data are two dimensional. If the data are seen as "$2 - 1/2$D", the neighbourhood can be seen as a cylinder. There can be no limit to the extent of the cylinder, because surfaces with a high slope have a sparse sampling of the surface. This is not the case for sampling along three axes, in which case surfaces with any orientation are sampled almost identically (surfaces at $45°$ are sampled more sparsely by a factor of $\sqrt{2}$), and the neighbourhood can be constrained along every axis.

Also, the thick trace was not used in the implementation of the algorithm for range data. The only valid purpose of the thick trace is to prevent frames coming from points that are in the same spatial neighbourhood (which consists of a sphere in this case), but not in the same neighbourhood on the trace or the traces on which they lie, to interact with each other. See figure 3.1 for such an example. Therefore, it is mostly to handle the case of two parallel traces that are close to each other; it is not intended to handle discontinuities within the same trace, as Sander assumes the surfaces are smooth. Range data samples the surface of objects as a graph, thus providing a single trace covering the sampling grid. The case of two nearby but distinct traces cannot occur.

Sander cited, as another reason for the thick trace paradigm, the inclusion of spurious data points.[5] Again, in the case of range data, since the connectivity of points is assumed to correspond to the sampling grid, even the initial estimates do not handle spurious points such as those created by shadow edges [**30**, p. 27].

4. Extrapolating patches

In his original algorithm, Sander used a parabolic patch to extrapolate the darboux frame of a neighbour to the location of the point to be updated. This patch is one of the easiest to implement since it only involves second order polynomials, but it has the disadvantage that the origin of the paraboloid is where curvature is maximum; all other points of the patch have lower curvature. If extrapolation is done locally to the origin of the paraboloid, curvature will only be approximately constant, and however small the bias, it will always be towards lower values. A large cumulative bias is likely to occur if a large number of iterations is performed. This is true for both the original method and its modification.

The equation of a paraboloid in a local coordinate frame based on the Darboux frame of the neighbour is given below. The w axis is along the normal and the u and v axes are along the principal directions.

$$w = \frac{1}{2}(\kappa_{M\alpha}u^2 + \kappa_{m\alpha}v^2) \tag{3.7}$$

Paraboloid patches for positive and negative Gaussian curvature are shown in figure 3.5(a).

One of the easiest possible extrapolations is not to extrapolate at all, i.e., the darboux frame of the neighbour is simply used directly as the update frame. The principal curvatures are therefore considered locally constant, as required, but the normal is not extrapolated either, while it obviously should, if the surface is to remain consistent (if the surface has curvature, then the normal vector should not remain constant). In order to satisfy the constant curvature constraint consistently, the normal should be extrapolated on an arc of a circle with constant curvature. Although such an extrapolation is well defined for one component of the normal, one does not know how to update the component of the normal in the direction perpendicular to the direction of extrapolation. One component of the normal is undetermined because there is no knowledge of the underlying surface to which the constant curvature arc belongs. It is like attempting *parallel transport* [**19**], [**56**, p. 38] along a curve without knowing the embedding surface. Figure 3.4 illustrates this problem. That is why it is necessary to use an extrapolating patch, to completely determine the extrapolated normal.

In an attempt to find a surface patch that satisfies a constant curvature constraint as much as possible by itself, we considered a torus patch, and another patch having a constant normal curvature property. A torus patch with origin on the extremal outer or inner rim has constant normal curvature along the principal directions at the origin. However, normal curvatures in other directions than the principal directions are not constant. Torus

[5] As another method of dealing with spurious data points Sander mentions cross-validation techniques [**18**]. But cross-validation does not deal with spurious data point in the sense of "outliers", it is rather aimed at estimating the "optimum" smoothness parameter of smoothing splines for data with additive uncorrelated white noise.

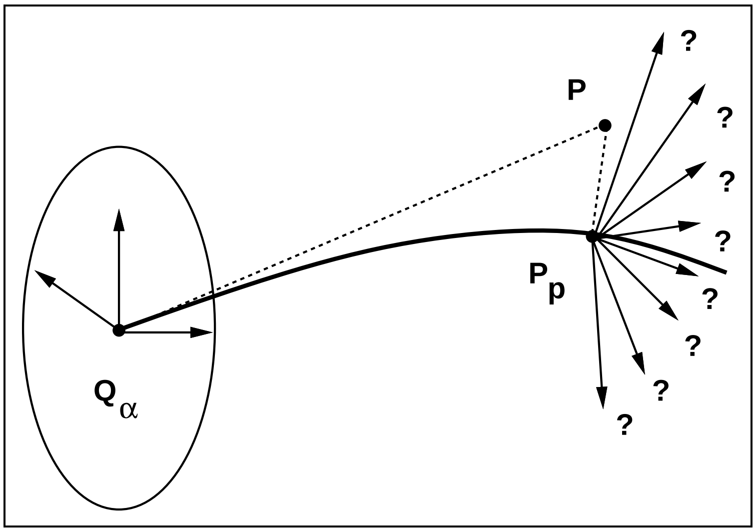


FIGURE 3.4. Transport of the normal on a constant curvature curve leaves one component undetermined. The surface shown has normal curvature as indicated by the thick arc of circle, in the direction in which the extrapolation is to take place. The extrapolated normal could lie anywhere in the normal plane to the curve at the projected point P_p, unless the surface embedding the arc is known.

patches with positive and negative Gaussian curvature are shown in figure 3.5(b), and the equation is given below. (In the equation found in [**19**], the w axis passes through the donut hole; in this equation, the w axis passes across the plane of the donut.)

$$w = \frac{1}{\kappa_M}\left[\sqrt{\left[1 - \frac{\kappa_M}{\kappa_m}\left(1 - \sqrt{1 - \kappa_m^2 v^2}\right)\right]^2 - u^2\kappa_M^2} - 1\right] \tag{3.8}$$

It is simple to derive the equation of a surface patch which satisfies the requirement that every normal section passing through the origin has constant curvature. This equation is presented below, and its illustration can be found in figure 3.5(c).

$$w = \frac{1}{\kappa_M u^2 + \kappa_m v^2}\sqrt{(u^2+v^2)\left[(u^2+v^2) - (\kappa_M u^2 + \kappa_m v^2)^2\right]} - \frac{u^2+v^2}{\kappa_M u^2 + \kappa_m v^2} \tag{3.9}$$

It should be noted that neither of these two types of surfaces have umbilic points, whereas a paraboloid patch with positive Gaussian curvature may have lemon umbilics [**6**]. Also, projecting the point to be updated on the extrapolating surface may miss for these two, since they do not have infinite extent, as the paraboloid. It is fair in this case not to consider the corresponding neighbour in the update, i.e., to exclude it from the contextual neighbourhood.

5. Least square fit of curvature fields

A curvature field is not a vector field as such since it is not *oriented*. Only the direction matters. Furthermore, it is not possible in general to assign an orientation to every line of a direction field in order to obtain an

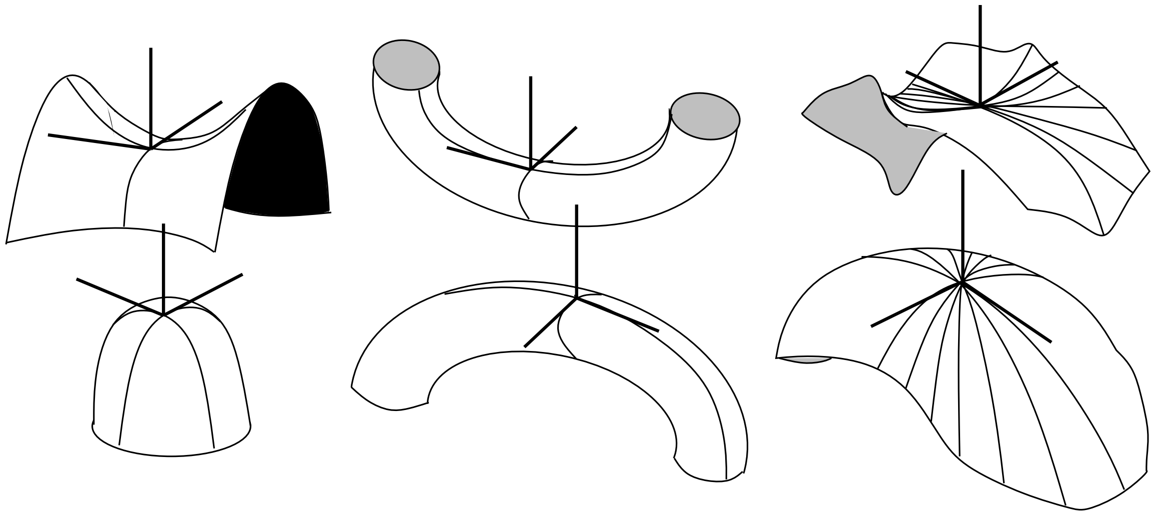


FIGURE 3.5. Possible extrapolation patches. For all patches, the top patch has negative Gaussian curvature, while the bottom one has positive Gaussian curvature. (a) shows parabolic patches. Their undesirable property is that the magnitude of normal curvatures is maximum at the origin. (b) shows torus patches. The two surface curves passing through the origin in the principal directions are arc of circles. They are therefore closer to enforcing a constant curvature constraint. However, other planar surface curves passing through the origin do not have constant curvature. (c) shows patches for which all planar surface curves passing through the origin are arcs of circles. It therefore enforces a constant normal curvature constraint at the origin.

everywhere smooth vector field, because of the presence of umbilic points. Figure 3.6 illustrates such a case.

One would be tempted to think that this problem was avoided in the original method by detecting umbilic points and not using them in the update of principal directions. However, as figure 3.6 shows, the discontinuities in the vector field do not only occur at the umbilic point, but along complete contours on the surface. Also, the detection of umbilics is not robust in the first iterations of the algorithm. We chose to include every point of the contextual neighbourhood in the updating of the direction field, without attempting to detect umbilics at this step. However, the updating rule for the principal directions has been modified to ignore the orientation of the vectors representing the principal directions. This modification is presented in appendix B. With the new updating rule, the results are very stable and preserve the umbilics without having to explicitly detect them.

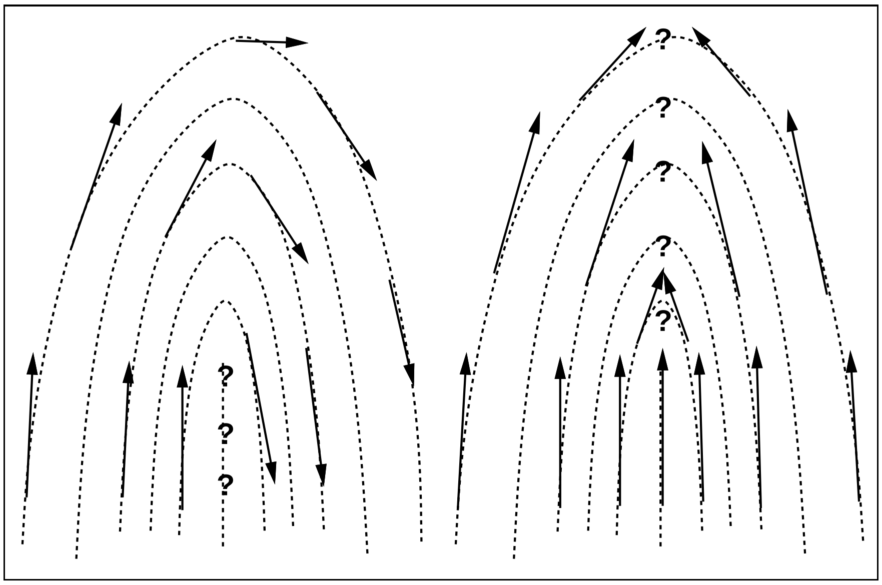


FIGURE 3.6. A direction field cannot always produce a smooth vector field. The figure shows a direction field (dotted lines) in the neighbourhood of a lemon umbilic. The two smoothest vector fields produced from this direction field are shown as arrows. In both cases, there is a line of discontinuity where the vector field is undefined. Note that the field can be undefined at points that are not umbilics.

As for other methods of refining the direction fields, those that consider only smooth vector fields, such as [**41**], are not useful, because our vector fields may have discontinuities. Work as also been done on curvature fields, for example, [**62**].

5.1. Extrapolation of the curvature magnitudes. In Sander's updating of the principal curvatures, the new principal curvature at a point was simply taken as the average of the principal curvatures of the extrapolating patches at the projected neighbouring points. This has been modified, because it is felt that this quantity does not represent the normal curvature along the principal direction at the point to be updated. Figure 3.7 demonstrates this point. The solid lines represent the *extrapolated* principal directions of the neighbours of the middle point. The dashed lines at the middle point represent the updated principal directions. Now, the updated principal curvatures at the central point should correspond to the normal curvature of the surface along those principal directions. Therefore, the principal curvatures should be the average of the normal curvatures of the neighbours in the required directions (the dashed lines) rather than the average of their extrapolated principal curvatures (in the direction of the solid lines). The updating of the principal curvatures at a point must therefore be done after the update of the principal directions is known. Given the updated principal directions M_{MP}^{i+1} and M_{mP}^{i+1}, the curvatures to be averaged are then

$$\kappa_{MP\alpha}^{i} = \kappa_{MQ\alpha}^{i}\cos^2\theta_M + \kappa_{mQ\alpha}^{i}\sin^2\theta_M, \quad \text{where} \quad \theta_M = \angle M_{MP}^{i+1} \tag{3.10}$$

$$\kappa_{mP\alpha}^{i} = \kappa_{MQ\alpha}^{i}\cos^2\theta_m + \kappa_{mQ\alpha}^{i}\sin^2\theta_m, \quad \text{where} \quad \theta_m = \angle M_{mP}^{i+1}, \tag{3.11}$$

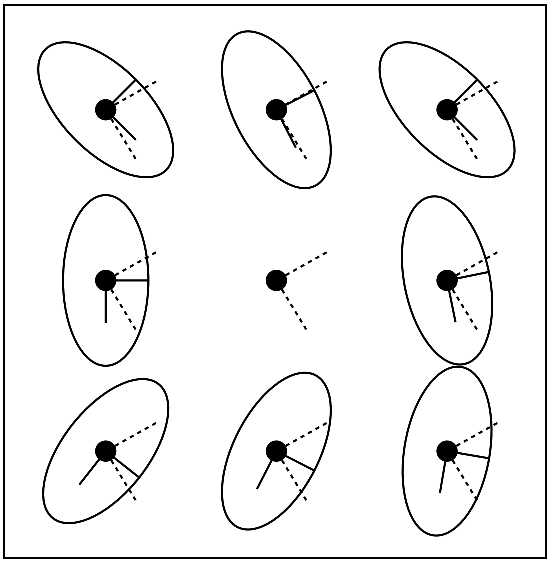


FIGURE 3.7. The principal curvatures of the neighbours do not directly correspond to the principal curvatures at the updated point.

instead of the principal curvatures themselves.

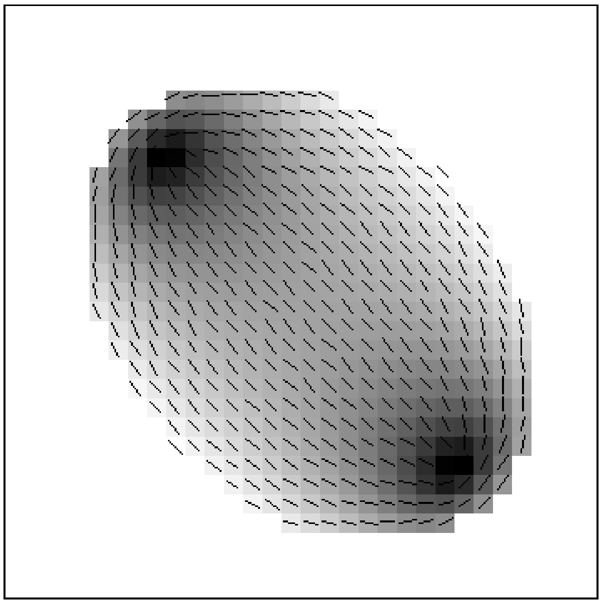
As for the modification of the updating of the principal direction fields, this modification is significant only around umbilic points. Elsewhere, the direction field does not vary much in a neighbourhood, and both methods give similar results. However, the consistency between the curvature magnitudes and the curvature field at umbilic points is greatly improved. Figure 3.8 illustrates the improvement on an ellipsoid.

5.2. coordinate system and projection method. One detail that has not been discussed in detail yet is how to determine the coordinate of the point at which we want the extrapolation. The method used by Sander consists in projecting the point $\boldsymbol{P}$ to be updated onto the extrapolating surface along the local w axis (in (u, v, w) coordinates) of the neighbour $\boldsymbol{Q}_\alpha$, which is actually $N_{Q\alpha}$. Other choices are possible, however. The projection used to obtain this coordinate should be either in a coordinate frame intrinsic to the surface, or in a meaningful extrinsic frame. Figure 3.9 shows three possible choices.

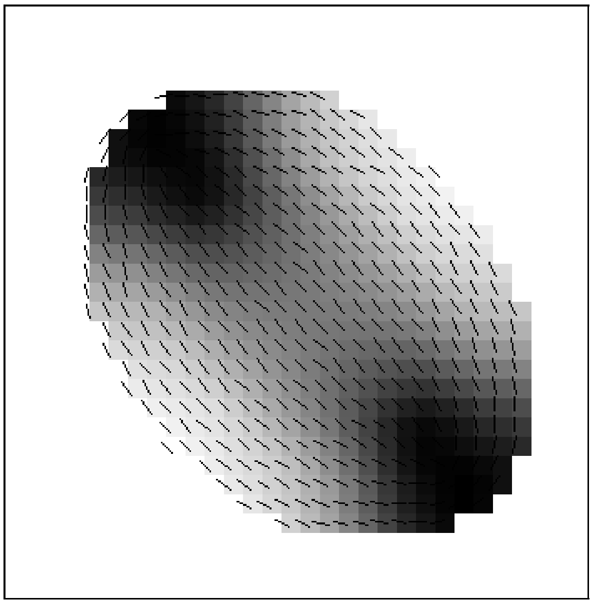
The first choice, represented as $\boldsymbol{P}_{pw}$ in the figure, is the one used by Sander. It consists in projecting along the normal of the neighbour $\boldsymbol{Q}_\alpha$. Another choice is to project perpendicularly on the interpolating surface. This case is denoted as $\boldsymbol{P}_{pn}$. Finally, yet another choice is to project the point to be updated along the global z axis onto the interpolating surface patch. This case is denoted by $\boldsymbol{P}_{pz}$. In the two first cases it makes a difference whether the global z position of the point $\boldsymbol{P}$ is known or not. However, this value is not known from the outset if the initial Darboux frames come from a shape from shading process [**22**] rather than from a range sensor. These two projection methods can be qualified as intrinsic, since they do not depend on the embedding of the surface in space. The first method, however, is extrinsic to the surface, and will give different

$\frac{\|\kappa_M - \kappa_m\|}{\max(\|\kappa_M\|, \|\kappa_m\|)}$	min		max	
analytic	0.064	(7,7)	0.691	(22,8)
old	0.301	(6,5)	0.649	(22,8)
new	0.014	(7,6)	0.647	(22,8)

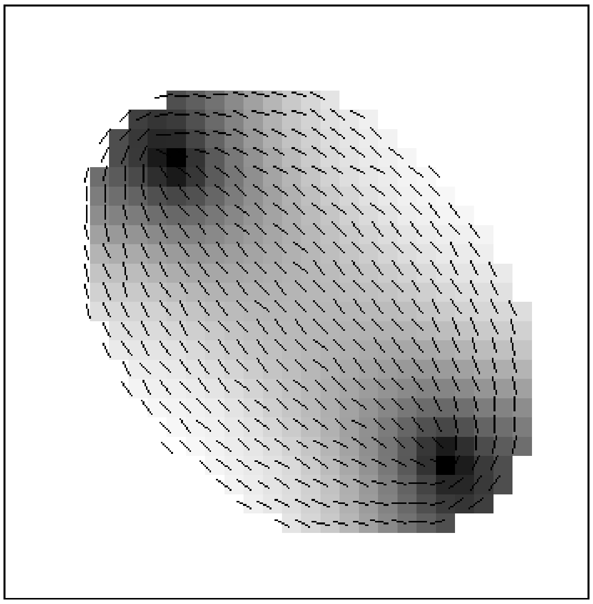
(a)



(b)



(c)



(d)

FIGURE 3.8. Comparison between old and new extrapolation of curvatures for an ellipsoid. The line segments indicates one of the principal direction, the intensities correspond to $\|\kappa_M - \kappa_m\| / \max(\|\kappa_M\|, \|\kappa_m\|)$. (b) are the results analytically computed from the equation of the ellipsoid. (c) , the old method, uses the principal directions as updates. (d) uses normal curvatures in appropriate directions. Both results correspond to 2 iterations of the updating process using torus extrapolation and the new field update rule. (a) indicates the min and the max values, as well as the coordinate at which they occur.

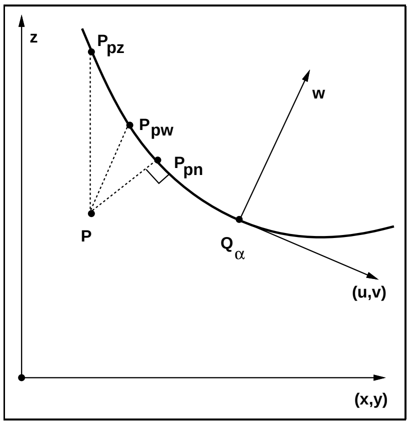


FIGURE 3.9. Possible projections onto an extrapolating patch. The illustrations are in 2D, but they generalize directly to 3D projections.

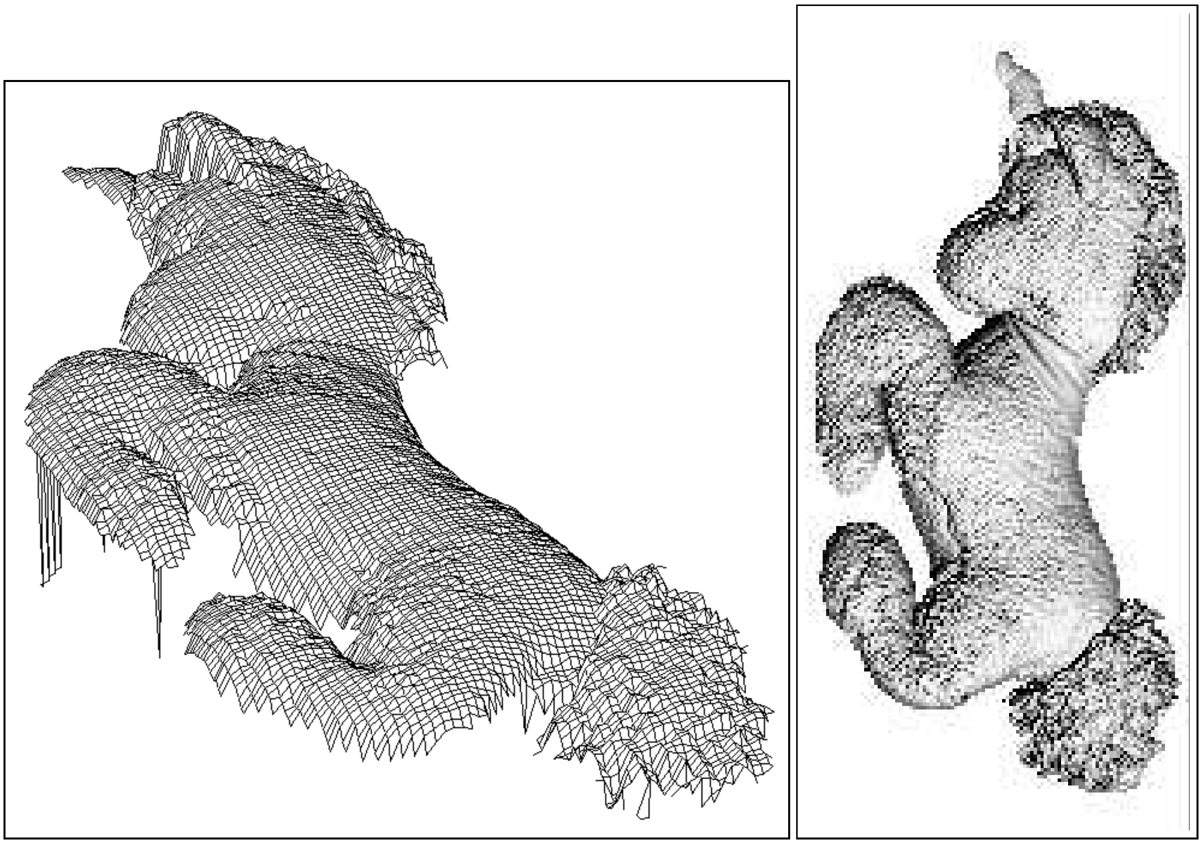
results if the surface is imaged at different orientations. The reasons for using such an extrinsic projection is because it is nevertheless meaningful with respect to the sampling of the data and its noise contents (mostly along the view axis of the range sensor). One choice that does not seem useful is projecting along the normal of the point to be updated. Indeed, as Sander argues, one should not rely on the very information that is to be updated, in principle.

Projection normal to the extrapolating surface is non trivial, even for a parabolic patch. It is possible, however, to use approximate methods, such as the one in [**44**].

The update of the depth of a point is obtained by averaging the z component of the projections on the extrapolation patches. In the two first projection methods, the global x and y coordinates of the point may also be modified by the projection operation, but those are ignored to preserve the initial format of the range image, for computational simplicity.

6. Results

Results of some experiments are now presented for the modified method. These are limited to one range image (taken from the CNRC database [**53**, image #139], and one analytically generated surface. Other results can be found in [**23**]. Large range images (256×256) are useful for qualitative evaluation of the method, and for discovery of unsuspected behaviors. However, testing the convergence properties of the method on such large images proves impractical due to the amount of time needed to execute the updating on a serial machine. Furthermore, once a peculiar behavior has been observed, it is required to trace its source. Small numerically generated surfaces were therefore used to test the method more rigorously. Small images have the advantage of permitting a large number of iterations of the method. Numerically generated surfaces permit to compare the differential quantities computed by the system with those computed analytically from the generating functions.



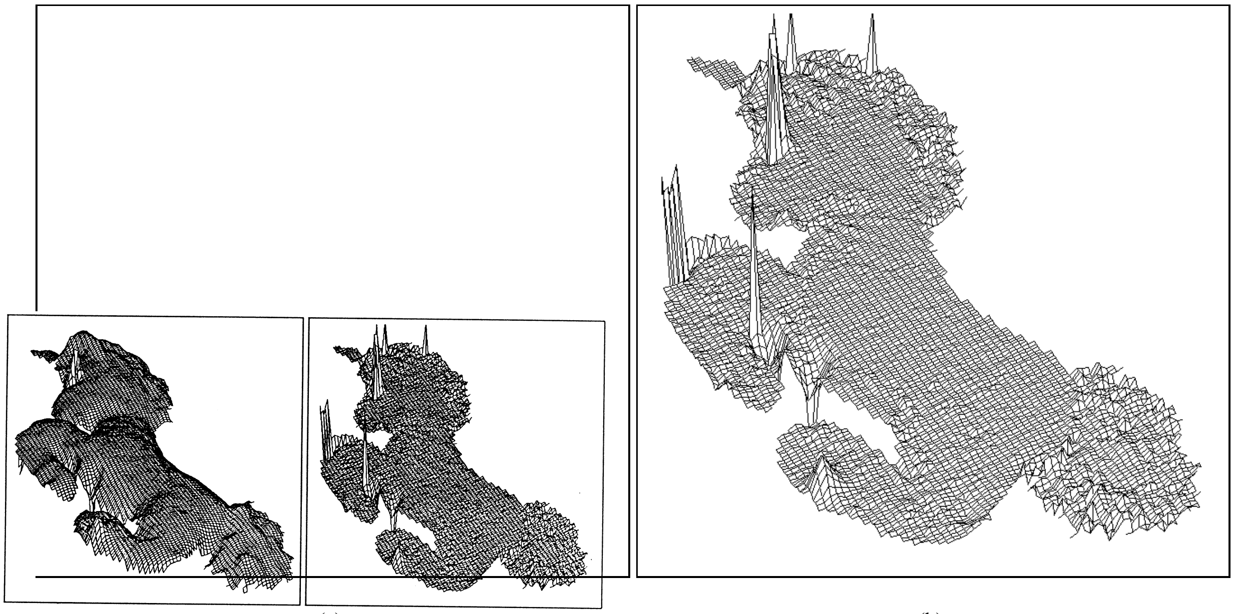
(a) Raw range data. (b) Approximate shaded image.

FIGURE 3.10. Initial data – fluffy unicorn.

6.1. A fluffy unicorn. Figure 3.10(a) shows the initial range data presented as a mesh, and (b) shows a shaded image generated from finite difference estimates of the surface normal (part of the initial Darboux frame). This image was chosen among those we experimented with because of the very textured surface, the richness of the shape, and the satisfaction of the smooth surface assumption at almost every scale of observation.[6] A rough occluding contour of the unicorn was obtained beforehand by thresholding the range data, and the algorithm was provided with this information so that it did not smooth over the boundary. The drops in the range data caused by the specular surface of the eye, and by a shadow edge in the fold of the upper leg were left as they were. Parts of the shadow edges also remained on the boundary of the shape.

Four iterations of the algorithm were performed on a Symbolics Lisp Machine. The projection method was along the w axis. The new field update rule and the new principal curvature updates were used. The extrapolating patch was a paraboloid, because using a torus patch considerably increases the running time of

[6]From the shaded image, we humans can detect some step discontinuities, even the ones on either side of the ribbon on the neck.



(a) (b)

FIGURE 3.11. Refined depths for the unicorn. (a) shows the new depth map after refinement. The initial data can be seen in figure 3.10. (b) is the difference between the two maps.

the algorithm, and four iterations were not sufficient to produce a significant bias in the curvature magnitudes. The information contained in the initial estimates of the augmented Darboux frames and in those refined by the algorithm is presented below in many forms.

Figure 3.11 shows the refined depth, and the difference with the

initial depth data. In the difference map, the bump on the lower leg is unexplained, and does not seem to appear in the initial data. All the spikes correspond to the shadow boundaries and the specularity of the eye, which produced step discontinuities in the original data. It is therefore normal that the algorithm deviated considerably from the data to enforce the smoothness assumption of locally constant curvature. Note however that these effects have remained very localized.[7] As for the rest of the difference map, it shows that no significant warp of the shape occurred.

Figure 3.12 shows one of the principal direction field. It is clear that the initial estimate of the field does not have any apparent structure, besides, maybe, on the horn and on the back. The results of four iterations of the algorithm show a lot more structure. From our own qualitative appreciation of the curvature of the surface of the unicorn, the field seems satisfactory on the legs, the front of the head, the neck and the body. The structure of the ribbon has apparently been lost, but this is understandable since the algorithm assumes

[7]Using a neighbourhood size of 5×5 for 4 iterations can influence points within an 8 pixel radius, while we observe much more localized spikes in the results.

(a) (b)

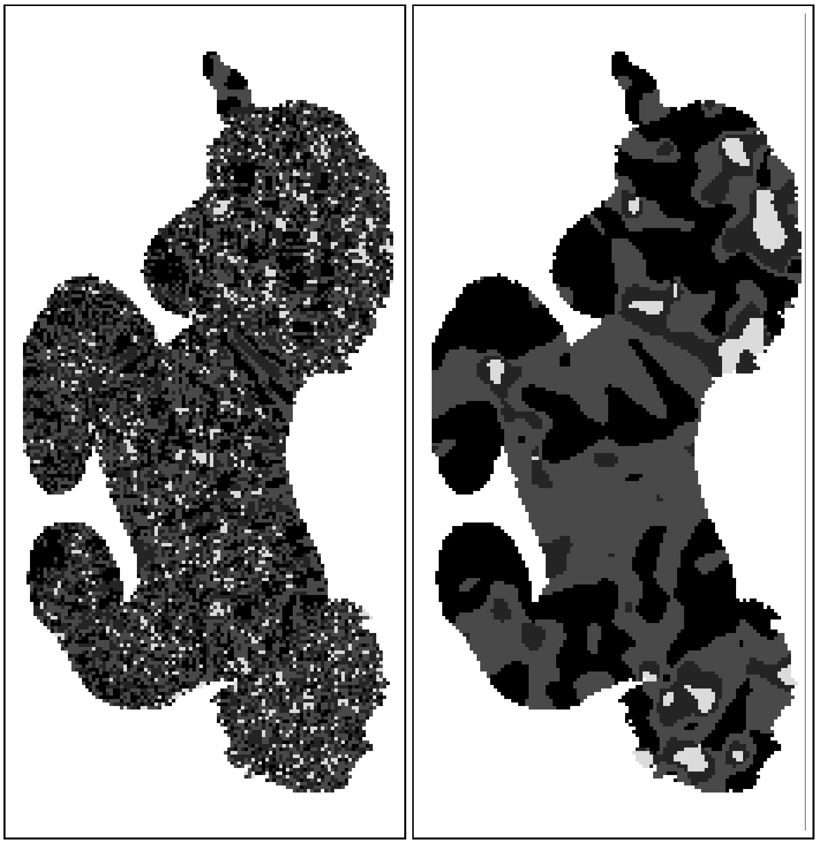
FIGURE 3.12. Refined principal directions for the unicorn, sampling every three grid points. (a) shows one of the initial principal direction field. (b) shows the same field after refinement.

the surface is smooth. The tail and the back of the head present noisier fields, but these areas were noisier in the data itself.

The segmentation of smooth surfaces following regions determined by the sign of mean and Gaussian curvatures has been proposed by many authors [**8, 66, 15, 55**]. Figure 3.13 shows the effect of the algorithm on those regions. Qualitatively, the results are comparable to those obtained by [**55**] (the experiment has also been run on the indian mask).

It has also been proposed to segment surfaces at local maxima of positive curvature (adopting the convention that the surface normal points outside of the surface, or towards positive z values, in the case of range data) [**10**]. Figure 3.14 shows how useless such a segmentation would be from the initial estimates, but that the refinement would permit it. Note that by explicitly storing the principal directions, the local maxima of positive curvature are assigned a direction as well as a position. This facilitates their aggregation into continuous surface curves, as was done in [**23**].

6.2. Experiments on a ellipsoid. An ellipsoid was used as a demonstration of the convergence properties of the algorithm. In this case, the augmented Darboux frame field of the shape can be computed from the



(a) (b)

FIGURE 3.13. Refined KH sign map for the unicorn. (a) shows the map produced from the initial estimates. (b) is after refinement.

analytic expressions of the surface derivatives. The initial estimates fed to the algorithm were these analytically computed ones. Hopefully, the algorithm should leave this data unchanged, since it is already consistent. This is not the case at convergence, however, as the following figures show. This means that if the algorithm forms an idempotent operator, then ellipsoids are not part of the corresponding constraint subset. Unless otherwise noted, the same methods for the unicorn were used here.

First, we compare the effect of the choice of extrapolation patches. Figure 3.15 shows the results. That the results at 100 iterations are close to the results at convergence can be seen from figures 3.17 to 3.21, which shows the global residual energy of the system, and its breakdown over the individual terms of equations 3.5. As expected, we see that using the paraboloid patch tends to flatten the surface at convergence. Recall that this is due to the bias of a paraboloid patch towards lowering the curvature of its neighbours. The surface obtained from torus extrapolations has comparable curvature magnitudes, but the shape of the surface has

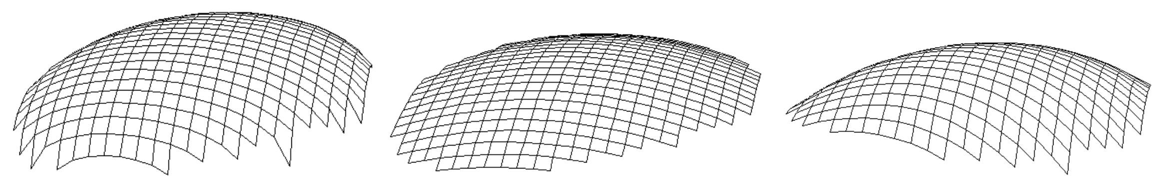
(a) (b)

FIGURE 3.14. Refined local maxima of positive curvature for the unicorn. (a) shows the trace segments produced from the initial estimates. (b) is after refinement.

obviously been warped considerably. The exact nature of this warp and the quantitative changes in curvature have not been studied in detail.

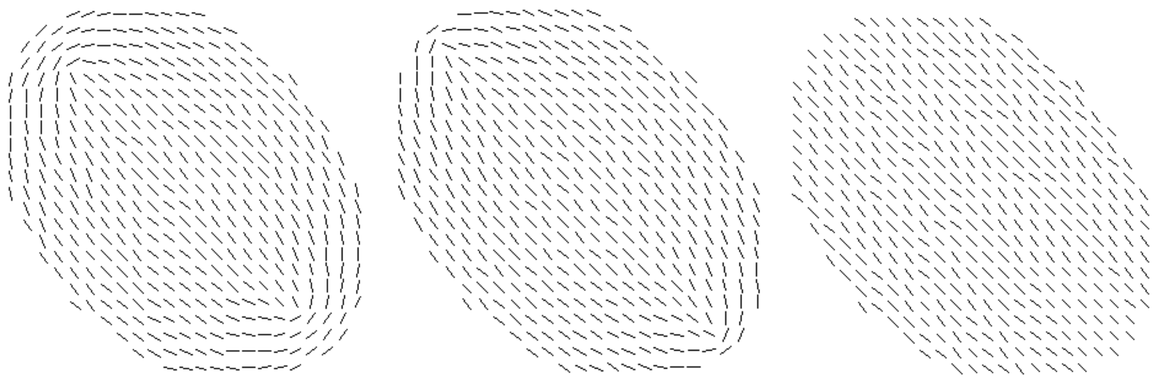
As figure 3.16 shows, the structure of the principal direction field is totally lost after 100 iterations. From the result after 10 iterations, we see that the umbilics are progressively pushed away until they disappear. The corresponding drop in residual can be seen in figure 3.19(b). Considering that a torus patch does not have umbilics, better results would be expected for a paraboloid patch, but the field behaves similarly when a paraboloid patch is used (notice the same energy drop in figure 3.19(a)).

Figures 3.17 to 3.21 show the sums of the local residuals during the updating process. In figure 3.17, we see that the total energy does not decrease monotonically in the case of the paraboloid, (there is a local maxima around the 40th iteration). In both cases, however, it is clear that the process will asymptotically converge to some result. In figure 3.18, we see that the residuals of the least-squared fits for the curvature magnitudes quickly go to zero. This is not surprising, considering that the update consists in a simple averaging of scalar quantities. As was already noted, the residuals for the principal directions updates quickly converge to zero



(a) (b) (c)

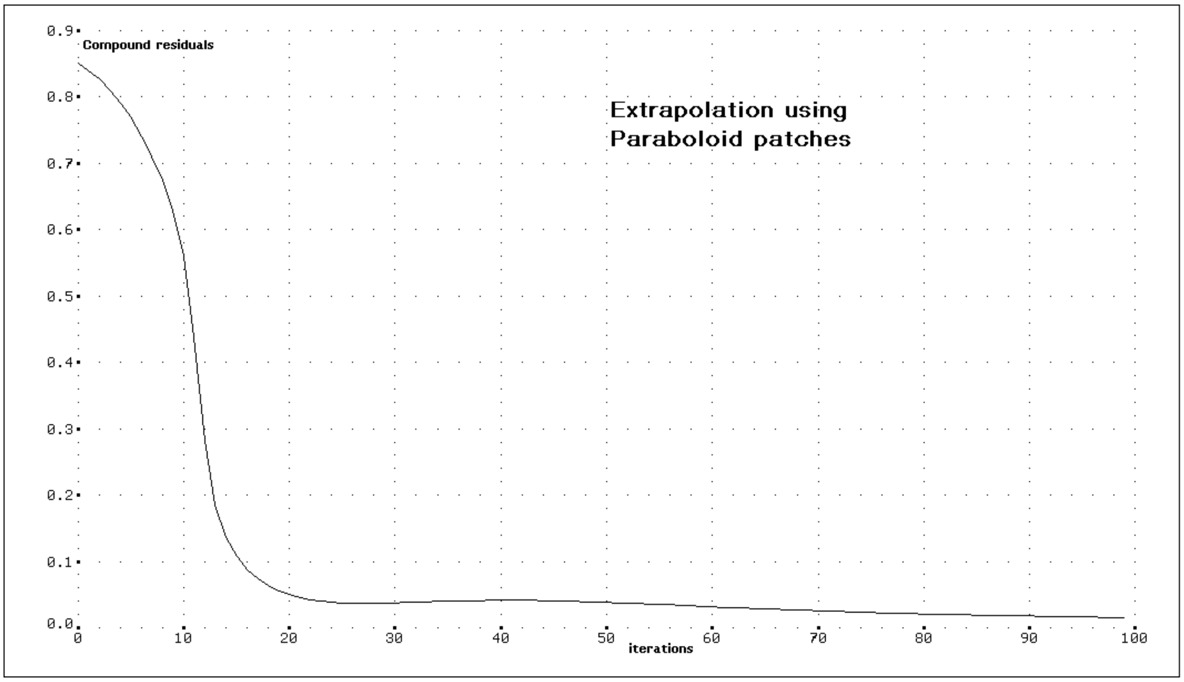
FIGURE 3.15. This figure compares the refined depth at convergence using paraboloid and torus extrapolation patches. (a) is the original depth data. (b) is the result of 100 iterations using a paraboloid patch, and (c) is the result of 100 iterations using a torus patch.



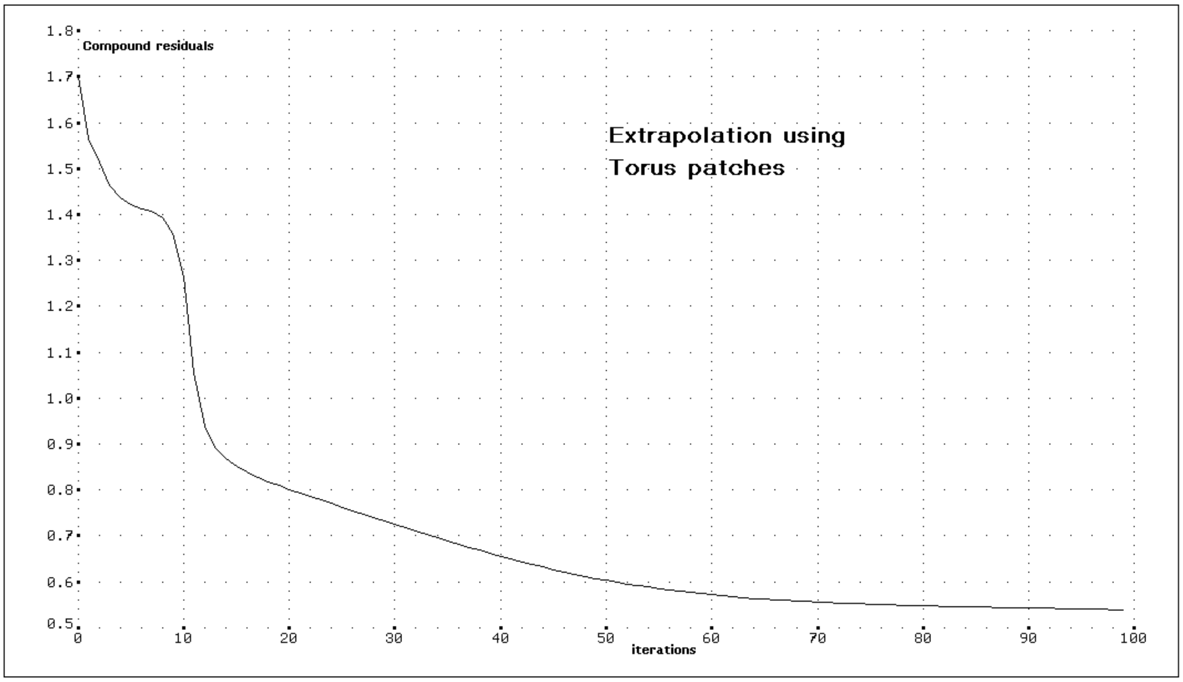
(a) (b) (c)

FIGURE 3.16. Loss of structure of the principal direction fields at convergence. (a) is the initial field, as computed analytically from the equation of the ellipsoid. (b) is the field after 10 iterations (using a torus patch). (c) is the field after 100 iterations.

once the umbilic structures have disappeared. Note also that these residuals form a significant part of the compound residuals, which was not the case for the principal curvature residuals. The two last residual plots are for the normals (figure 3.20) and for the depth (figure 3.21). The depth residuals are slightly more significant than the normal ones, but both are less significant than the principal directions residuals. The convergence of these residuals is much more eventful than the two previous ones. In general, we have not found any correlation between the positions of the local extremas and the observable behavior of the augmented Darboux frames, as was done in the case of the principal directions residuals. We noted, however, that the torus patch extrapolation produced convergence sooner than with the paraboloid patch, as can be seen by observing the position of the last local maximum, and the slope of the curve at the 100th iteration.

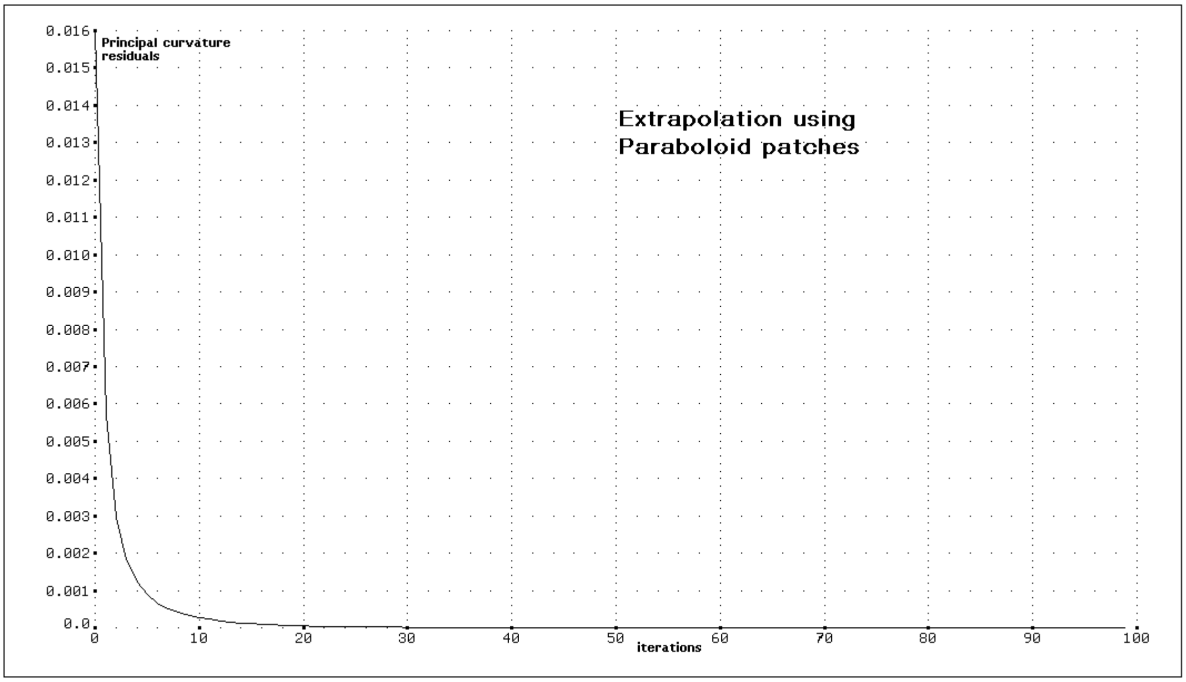


(a)

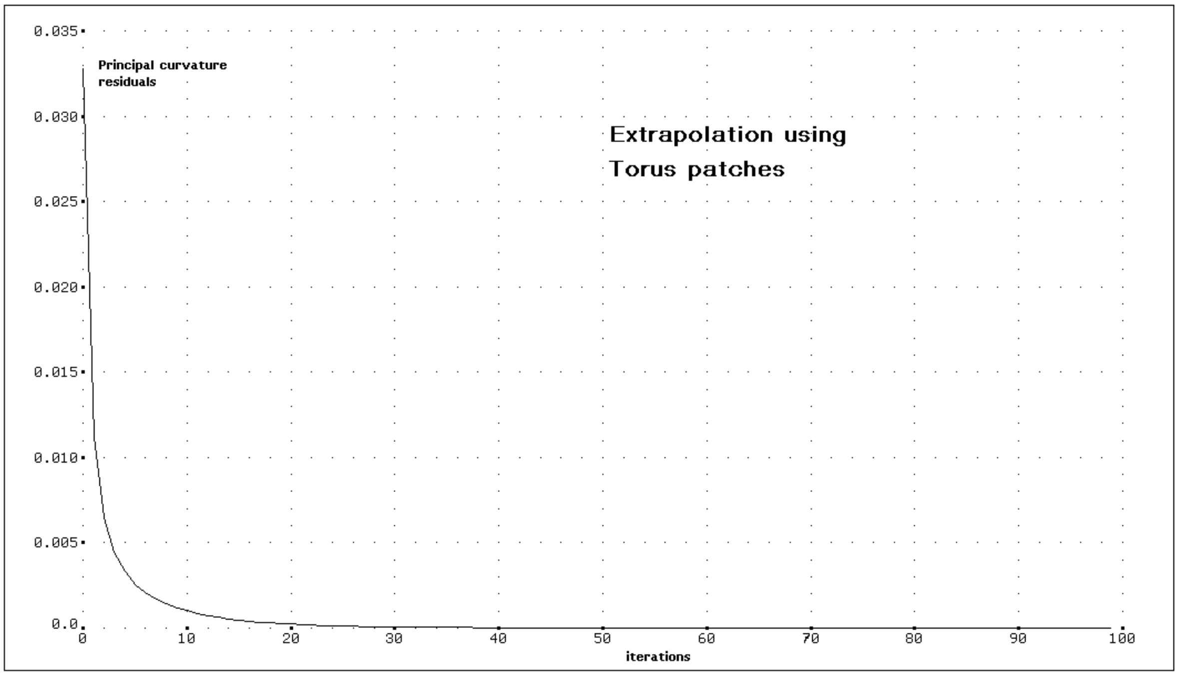


(b)

FIGURE 3.17. Compound residuals



(a)



(b)

FIGURE 3.18. Residuals of principal curvature updates

Finally, we show the result of the updating process on the ellipsoid with 10% additive Gaussian noise in the z direction. Figure 3.22 shows the initial data, as well as the refined depth after 5 iterations. The surface is smooth, but is obviously badly warped compared with the noise free ellipsoid. The local features do disappear after more iterations, but the global warping observed above takes place instead.

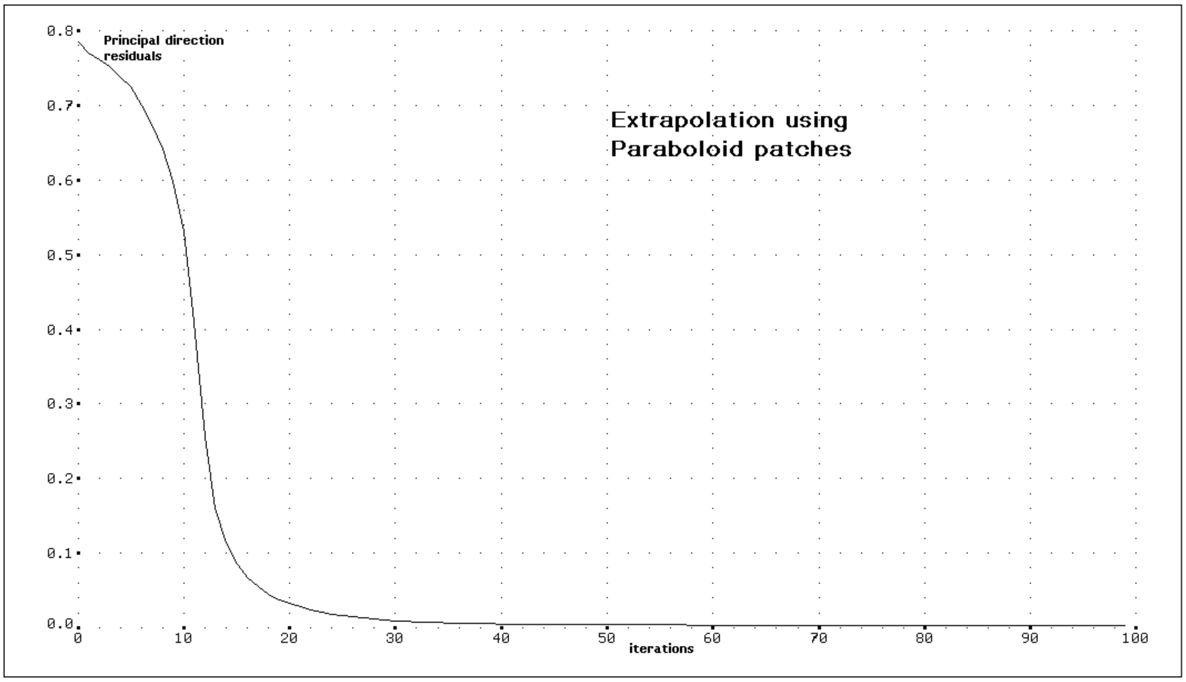
Figure 3.23 shows how the method can recover the structure of the principal direction field. In this case, the initial augmented Darboux frames were computed numerically, with the result shown in 3.23(a). The next figure shows the field corresponding to the fifth iteration. One of the umbilics has been recovered, but not the other. In any case, as was shown above, even the recovered umbilic would disappear if the algorithm were left to iterate to convergence.

7. Discussion

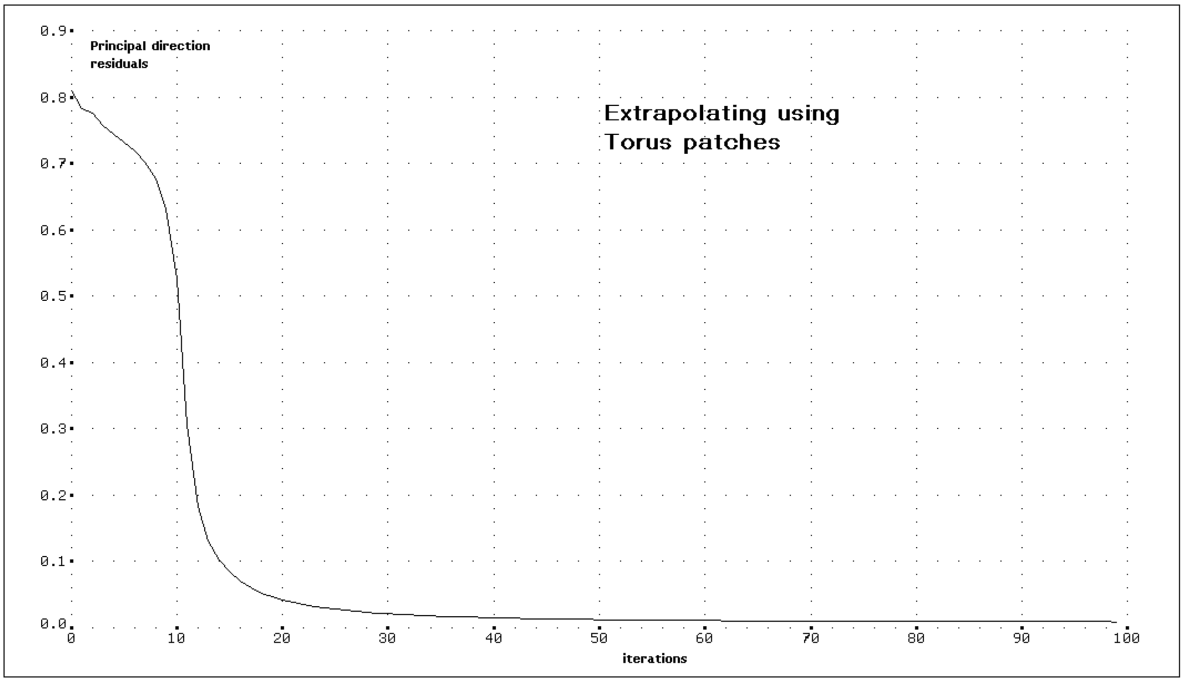
7.1. The hidden scale parameter: number of iterations. Let us discuss the effect of varying the number of iterations on the updating process. From the experiment with the ellipsoid and other analytically generated surfaces, it is obvious that the "locally constant curvature" constraint becomes a *globally constant curvature* constraint at convergence. It is not clear at this point how the normal curvature can vary in different directions on the surface at convergence, i.e., we do not expect the results to be always spheres. For example, a cylinder already satisfies the constraints if a torus patch is used (note that the osculating torus to a point on a cylinder degenerates into a cylinder), so a cylinder would not be warped by the process. Even then, the constraint seems too restrictive if the process is ran to convergence. All the small scale features of the surface will be lost. If the locally constant curvature constraint was applied to plane curves, then it is clear that the result of the algorithm at convergence would always be a circular arc, the radius being the only free parameter of the class of curves satisfying the constraint. It is not as straightforward to find the complete class of surfaces that satisfies the locally constant curvature constraint, because the extrapolating surfaces (such as the torus) are much more complex than the arc of circle that would be used for curves.

One apparent solution to the globality of the constraints is to control the number of iterations. The number of iterations is at least in part a scale parameter, because the updating process is local, and the only means for a point on the surface to influence a point further than its local neighbourhood is through a number of iterations. However, we are now back to the same problem than with regularization theory: the choice of the scale (or smoothness) parameter.

Another solution is to attempt to find a better set of constraints, that would not be as restrictive as the constant curvature one. Then, the process could be run to convergence every time. However, this does not solve the question of scale. Scale is an important component of perception, and if the updating process is always run to convergence, a new scale parameter has to be found, possibly in the computation of the initial estimates.

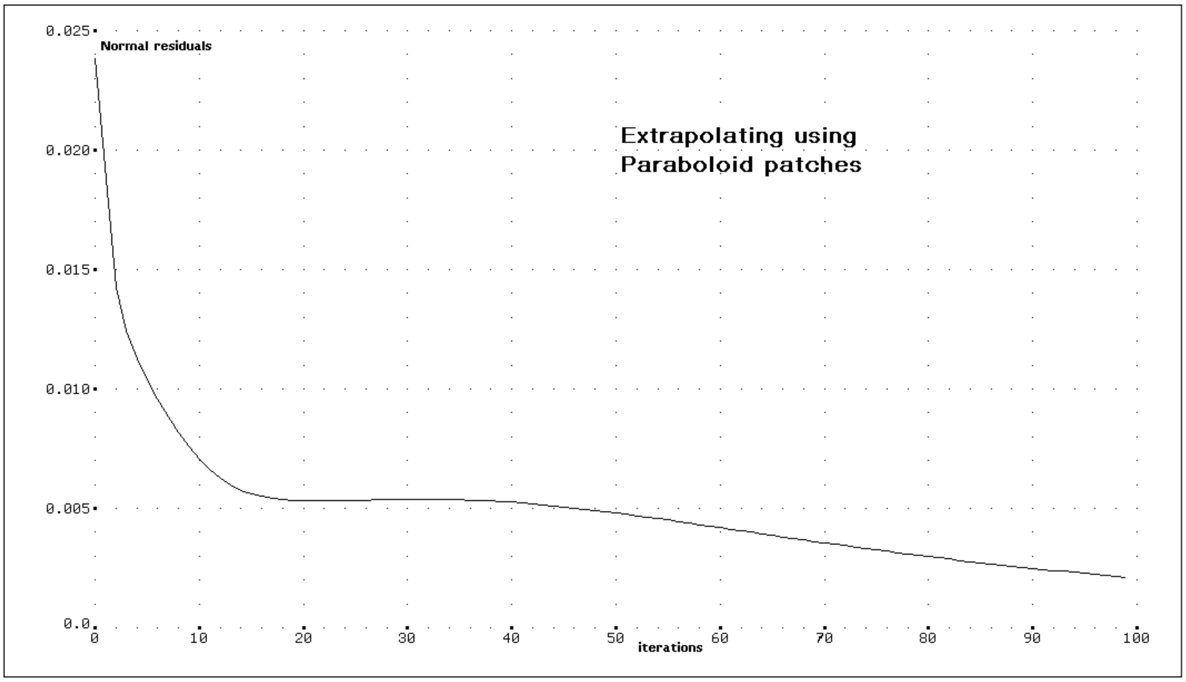


(a)

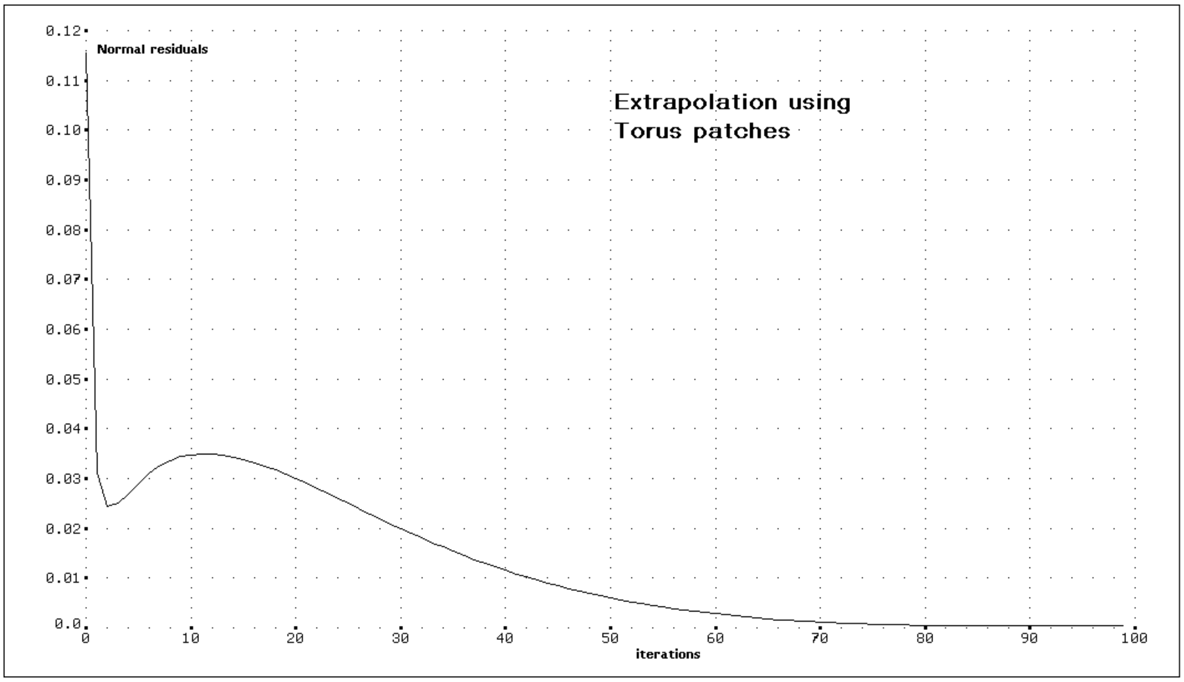


(b)

FIGURE 3.19. Residuals of principal directions updates

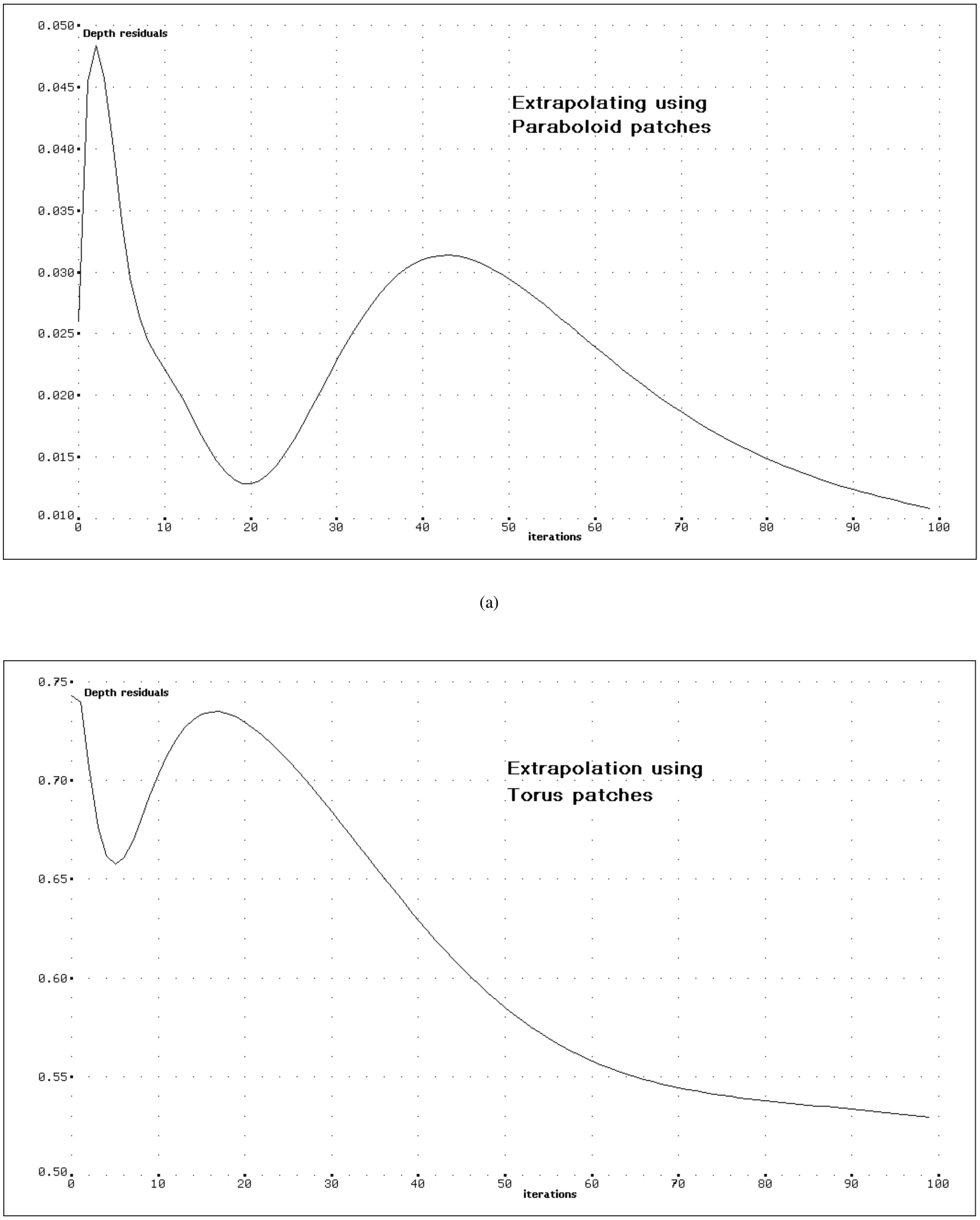


(a)



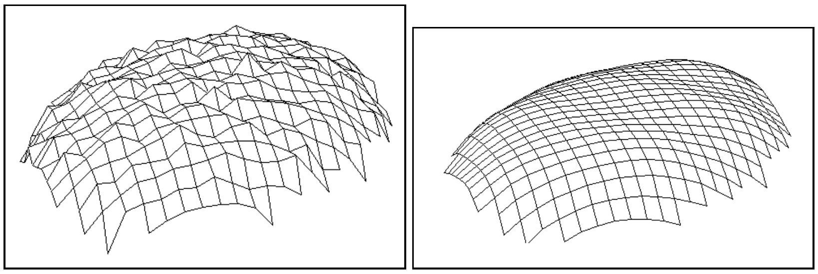
(b)

FIGURE 3.20. Residuals of normal updates



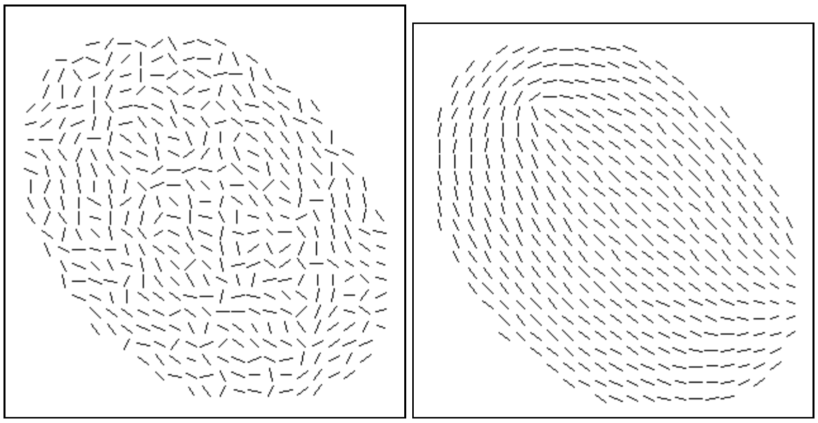
(b)

FIGURE 3.21. Residuals of depth updates



(a) (b)

FIGURE 3.22. (a) is the previous ellipsoid with 10% Gaussian noise added. (b) is the result of 5 iterations of the updating process.



(a) (b)

FIGURE 3.23. (a) is the initial estimates of the principal direction field. (b) is the result of 5 iterations of the updating process.

7.2. Type of energy minimization. In the previous section, we saw that letting the iterations run to convergence gives some globally simple surface as a result, and that controlling the number of iterations amount to setting the value of a scale parameter. When the process is ran to convergence, it is obviously an idempotent operator. Indeed, taking the result at convergence as the initial estimate for another application of the process is the same thing as attempting another iteration on the first application of the process once convergence has been reached. This will let the result unchanged. It is therefore possible to express this particular application of the algorithm as a local energy minimization.

The global residual energy computed to measure convergence is given by the sum of

$$\hat{R}(\xi_{\boldsymbol{P}}^{i+1}) = \sum_{\alpha=1}^{n} \left[(\boldsymbol{N} - \boldsymbol{N}_{P\alpha})^2 + (\kappa_M - \kappa_{MP\alpha})^2 + (\kappa_m - \kappa_{mP\alpha})^2 \right] + \sum_{\alpha=1}^{\hat{n}} (\boldsymbol{M}_M - \boldsymbol{M}_{MP\alpha}). \tag{3.12}$$

at every point of the surface. The function will be zero for surfaces that can be exactly fitted by the chosen extrapolation patch, at every point. In the case of a torus patch, the function will be zero for spheres and cylinders. These surfaces can be considered as part of the constraint subset of the idempotent operator. The iterative process does not perform a local minimization on this function however, because the residual does not monotonically decrease during the process, as can be seen in figure 3.17. Therefore, although the process is idempotent, it is not the global residual energy computed by Sander that is the function to locally minimize.

When the number of iterations is controlled, then the operator is not idempotent. If a second application of the process is attempted with the result of the first one as initial value, the second process will simply continue where the first one left off, to bring the result closer to the result at convergence. This operator cannot therefore be expressed as a local energy minimization.

7.3. A secondary effect of the thick trace. In view of the considerable warping of surfaces after a large number of iterations of the updating process, we wondered why such an effect was not detected in Sander's experiments. One possible reason is that a large number of iterations were never performed. Another reason is the use of the thick trace to determine the contextual neighbourhoods, in the case of Sander's original method. There were reasons to use a thick trace paradigm in the 3D case, but this thick trace does not only perform the intended task, it may also prevent the algorithm from running to the intended convergence. Because of the distortion introduced in the surface after a given number of iterations, the extrapolating patches are less and less like their initial estimates, which were obtained by least-squared fit to the neighbours, and therefore were necessarily passing close to the neighbours. As the surface distorts, less and less neighbours will satisfy the thick trace constraint, and at some point, updating will cause no change in the Darboux frame to be updated, because not a single neighbour satisfies the thick trace constraint. The thickness of the thick trace will artificially limit the amount of distortion caused by the algorithm, and this distortion effect may go unnoticed. By not having the thick trace constraint in this implementation, the distortions were noticed more easily. Note that in this case, the thickness of the thick trace can be considered as another version of a scale parameter.

7.4. Advantages of Sander's approach. Because the iterative updating does not use finite difference equations, it is easily extended to non regular, or non orthogonal grids. It was noted in section 5.2 that the update of the depth was done along the global z axis. This was done because it was more computationally practical, and also because it is known that most of the image noise produced by the triangulation range sensor used is in the depth measurements [**3**]. However, in principle, there is nothing requiring a regular sampling grid or an update of the depth in the z direction. Even sparse sampling of the surface is acceptable, as long as the surface is smooth and the sampling still gives sufficient information about the larger scale shape of the object.

The method uses curvature constraints, which are view independent. Regularization methods, such as the thin membrane or the thin plate models [**59**], [**13**], can be moved one order up to smooth the third derivative of the surface, but the constraints will not be view independent. There are some regularization methods that have approximately view independent constraints.

7.5. Possible improvements. The following improvements are meant to leave the philosophy of the method intact, and are therefore mostly technical (as were those that were implemented and discussed above). The following chapters discuss more fundamental problems of surface reconstruction.

One modification to the variational relaxation that would take it closer to relaxation labelling is to keep a confidence value for each Darboux frame in the grid. When updating a point, the extrapolated Darboux frames would be obtained in the same way, but they would be weighted by their confidence when doing the variational updating (averaging). The new confidence in the update frame would be a function of the residual of the local update. The original confidence value would be a function of the residual of the fit needed to find the initial estimate of the darboux frames.

There should also be a confidence value associated with the principal directions, that would be based on the closeness of the point to being an umbilic. Indeed, an umbilic point does not carry any directional information on the curvature of the surface. Its confidence in its "principal directions" (which are computed by the algorithm at every points) should therefore be zero. At points that are far from being umbilic, i.e., that have a large absolute difference between their principal curvatures, the confidence in the principal directions should be high. The confidence in the principal directions could therefore simply be the absolute value of the difference between the two principal curvatures. It is not necessary to normalize this confidence, because it will vary smoothly among neighbouring points of a smooth surface. But a normalized confidence could always be formed by

$$\frac{||\kappa_M - \kappa_m||}{\max(||\kappa_M||, ||\kappa_m||)}. \tag{3.13}$$

We feel that this would be much better than trying to explicitly detect umbilics, as Sander did, in order to prevent umbilics to contribute to an update. For example, a lot of umbilics are detected in the initial estimates of a very noisy surface [**56**, p. 70]. Using Sander's approach, a lot of points would be prevented from contributing to the local updates. Using the confidence scheme, most points would have more or less the same (low) confidence, and would therefore contribute equally to the updates, which is what we want until some structure is found in the data.

The reason for not having implemented these improvements is that it was clear that the approach is deficient in more fundamental ways than such technicalities. It was decided to consider more theoretical subjects rather. The next chapter considers other popular surface reconstruction methods, and attempts to unite them along the energy minimization framework developed earlier.

CHAPTER 4

Study of well known methods

Now that Sander's method and my adaptation of it have been described in detail, I will compare it with a number of other surface reconstruction methods. There is a need to study the methods in isolation and to compare them together in various ways. I have decided to structure the chapter by placing the emphasis on one method at a time, and to include in appendices any involved comparison between them. This permits the discussion on one method to freely make cross references through the appendices.

The methods that I included are Terzopoulos' regularization method [**59**], Geman and Geman's MAP method [**26**], Blake and Zisserman's weak continuity method [**13**], Sander's variational relaxation method [**56**], Besl and Jain variable order surface fitting [**9**], and Leclerc's MDL method [**36**]. These methods are only a selection of available ones, but they are considered representative of the principles that have evolved in surface reconstruction. They are presented in their chronological order of appearance, in order to show that evolution. These methods are preceded by a discussion on regularization methods in general.

For each method, a brief description of the problem is first given, followed by a description of how the problem is solved. The adequacy of the problem itself is considered more important in this thesis, so the method used to solve the problem, or even the feasibility of the solution will not be described in great depth.

The method is then compared to others, and the properties of the method are discussed in general. Finally, the method is classified in the energy minimization formalism.

1. Regularization based approaches

1.1. Overview of regularization.

1.1.1. *Problem description.* Regularization, as used by [**47**], denotes:

> . . . any method used to make an ill-posed problem well posed.

Since surface reconstruction is an inherently ill-posed problem [**59**], [**31, 47**], and since the goal of any surface reconstruction method is to obtain a solution to a well-posed problem (even those including the detection of discontinuities), any surface reconstruction method is a regularization method in that sense. In this section, we study the "standard regularization methods" [**47**]. In standard regularization, the problem is to make well-posed (to regularize) the problem of solving the equation $A\boldsymbol{z} = \boldsymbol{y}$ for $\boldsymbol{z}$, by introducing a norm $||\cdot||$ and a stabilizing functional $||Pz||$. For standard regularization, A and P are linear operators, and $||\cdot||$ is quadratic. The well posed problem is then to solve

$$\min_{\boldsymbol{z}}(||A\boldsymbol{z} - \boldsymbol{y}||^2 + \lambda||P\boldsymbol{z}||^2), \tag{4.1}$$

where λ is the regularization parameter.

In computer vision, the data $\boldsymbol{y}$ is usually a discrete set of measurements. However, $\boldsymbol{z}$ can be either a discrete set of values or a function to be evaluated at discrete points for comparison with $\boldsymbol{y}$. In the text below, the theory is usually presented in terms of functions, that will explicitly be denoted as such, e.g. $z(x_i)$. The regularization functionals used in computer vision usually use the discrete case, which is explicitly denoted using indices, e.g. z_i.

1.1.2. *Problem solution.* The literature on solving convex optimization problems such as the ones produced by regularization is extensive. The most practical methods in computer vision are iterative updating of an arbitrary initial value by the use of some kind of gradient descent along the energy functional. [**57**] is one of the many standard books on such methods.

1.2. Characterization of the method.

1.2.1. *Comparison with other methods.* Terzopulos's method, when discontinuities are included a priori or are detected after the surface reconstruction process, is standard regularization. The stabilizing functional P is non-linear in Terzopoulos' variational continuity control, in Blake and Zisserman's weak continuity method, and in Leclerc's MLD method; these are therefore not standard regularization methods.

Similarities between Terzopoulos' variational continuity control and Blake and Zisserman's weak continuity are discussed in section 4.

1.2.2. *Choice of stabilizing functional.* The particular norms and stabilizing functionals used up to now in the reconstruction of surfaces were often chosen rather ad hoc. Grimson [**31**, section 6.3] performed an analysis of second order functionals, and has concluded that all may be expressed as a linear combination of the square Laplacian and the quadratic variation. Boult [**16**] studied a large range of stabilizing functionals, based on a subjective ordering of the functionals by human observers who rated the reconstructed surface from what they personally perceived from raw (pointwise) data. One choice of stabilizing functional that I consider particularly ad hoc is the use of physical model analogies such as thin membranes or thin plates under tension.[1] As Blake argues [**11**], there is no reason in favor of using such physical models for visual surface reconstruction. In the same paper, Blake proposes a stabilizing functional based on curvature, which is an intrinsic property of surfaces.

[1]Actually, the models are the *small deflection* approximations [**58**].

1.2.3. *Characterization of the solution to regularization problems.* In this section, we will first consider the one-dimensional regularization with stabilizing functional

$$J(z) = \int_{xa}^{xb} (z^{(m)}(x))^2 \, dx. \tag{4.2}$$

The regularization problem is to find z that minimizes

$$\frac{1}{n}\sum_{j=1}^{n}(z(x_j) - d_j)^2 + \lambda J(z). \tag{4.3}$$

In general, the solution which minimizes this functional is a polynomial smoothing spline of order $2m-1$. That is, every curve segment from x_i to x_{i+1} is a polynomial of order $2m-1$, where each polynomial is joined such that the overall curve is C^{2m-2}. There are well known particular cases [**63**]:

- If $\boldsymbol{d}$ can be interpolated exactly by some polynomial of degree less than m then that interpolating polynomial is the solution of the minimization.
- If $\lambda \to \infty$ and $\boldsymbol{d}$ cannot be exactly interpolated by a polynomial of degree less than m, then the solution is a single polynomial of degree $m-1$ best fitting the data in a least-squares sense.
- If $\lambda \to 0$ and $\boldsymbol{d}$ cannot be exactly interpolated by a polynomial of degree less than m, then the solution is the polynomial spline of degree $2m-1$ that interpolates the data $\boldsymbol{d}$.

The two last cases are very special in that the corresponding operators can be shown to be perpendicular projections, and thus, idempotent. This is obvious for the first case, since a least-squares fit is by definition a closest point problem [**50**, p. 197]. Proof for the other case can be found in [**40**]. As is shown in section 1, however, the operator is not idempotent in general.

The two-dimensional case is slightly more complicated, especially if the sampling grid is not uniform. Also, the solution to the minimization cannot be expressed as a piecewise polynomial surface in general. However, the same principles hold.

For the two-dimensional case, the stabilizing functional is given by

$$J(\boldsymbol{z}) = \int_{xa}^{xb}\int_{ya}^{yb}\sum_{l=0}^{m}\left(\frac{\partial^m \boldsymbol{z}}{\partial x^l \partial y^{m-l}}\right)^2 \, dxdy, \tag{4.4}$$

which corresponds to the thin membrane model for $m = 1$ and to the thin plate for $m = 2$. Models similar to those were used by [**31**] and [**59**] for surface reconstruction.

We now generalize the remarks made earlier for the one-dimensional case. In general, the solution which minimizes this functional is a $C^{2(m-1)}$ surface, which is part of the Sobolev space $\mathcal{H}^2$ of order two [**58**], and is given by [**21**]:

$$\sum_{i=1}^{n} c_i r_i^{2(m-1)} ln(r_i) + p_m(x,y), \quad m \geq 2, \tag{4.5}$$

where $p_m(x, y)$ is a polynomial of degree $< m$, and $r_i^2(x, y) = (x - x_i)^2 + (y - y_i)^2$. n is the number of data points. Note that the solutions were C^{2m-1} in the one-dimensional case. Nevertheless, the three cases above generalize directly to two dimensions [**63**]:

- If $\boldsymbol{d}$ can be interpolated exactly by some polynomial $P(x, y)$ of degree less than m ($P(x, y) = \sum_{i+j=0}^{m-1} a_{ij} x^i y^j$) then that interpolating polynomial is the solution of the minimization.
- If $\lambda \rightarrow \infty$ and $\boldsymbol{d}$ cannot be exactly interpolated by a polynomial of degree less than m, then the solution is a single polynomial of degree $m - 1$ best fitting the data in a least-squares sense.
- If $\lambda \rightarrow 0$ and $\boldsymbol{d}$ cannot be exactly interpolated by a polynomial of degree less than m, then the solution is the $C^{2(m-1)}$ surface given by equation 4.5 that interpolates the data $\boldsymbol{d}$.

Again, the two last cases correspond to perpendicular projections [**40**], but the regularization operators are not idempotent in general.

1.2.4. *Choice of regularization parameter.* In surface reconstruction, the stabilizing functional usually imposes a maximum order constraint on the surface. The value of λ can then be seen as an arbitrator between a "closeness to data" constraint and a "zero order m derivative" constraint, for some given integer m. As was implied in the previous section, if λ is very small, then the resulting surface will not necessarily be very smooth, but it will closely interpolate the original data. If λ is very large, the resulting surface may poorly interpolate the data, but it will be $C^{2(m-1)}$ smooth. Setting λ at one of these extremes is considered uninteresting in terms of regularization (although it is seen as useful by other formalisms such as Leclerc's and Geman and Geman's, as will be seen later). λ has also been interpreted as a scale parameter [**13**]. This is intuitively correct, since at one extreme, a global polynomial is fitted to the data, and as the value of λ moves toward zero, more and more local features are permitted, up to the point where every known local feature (the sampled points) are interpolated.

The most important problem of regularization methods is to find the "optimal" value of the parameter, and to define the meaning of optimality. In the particular case when the noise characteristics of the signal are known a priori, optimality may mean to have the smoothest result that retains the known noise characteristics of the data. There are theoretical results to handle this case [**18**].

If however the "noise" characteristics of the signal are not known (which I will claim later is always the case in computer vision), they may be estimated by statistical means. [**18**] developed such a method, called *generalized cross-validation (GCV)*. The idea is to use the ability of regularized functionals *done without one of the data points* at predicting the value of the missing point, as a measure of the goodness of the regularization parameter. Another approach is the Min-Max principle, which is succinctly presented by the authors [**27**]:

> The main difficulty in choosing λ is that if it is either too low or too high, one of the objective functions will be inadequately represented in the total cost, and the total cost will be too low.

> One way to ensure that the total cost will not be too low is to pick *(λ giving)* the maximum cost solution.

Here, the "cost" is the residual of the solution of the regularization problem, and the objective functions correspond to the closeness to data term and the smoothness term.

Here, I argue that knowing the acquisition model of the data is not sufficient to find an optimum value of λ. Furthermore, I claim that there is no single optimum value of λ for the purpose of surface reconstruction. The reasons for both these claims is the same. When using regularization as a data processing tool, the goal is to ease the interpretation of the data by specialists, or to interpolate or extrapolate the signal. In that case, what constitutes the signal is well defined, especially if the noise process is known. In the case of computer vision, the goal is not only one of signal processing, but also of *automatic signal interpretation*. What constitutes the signal (in the sense of the information we are interested in) is not as well defined in this case. For example, if the surface under observation has some small scale texture (relative to the scale of interest), then this texture is part of the signal in a data processing sense, but not in a surface reconstruction sense, where we may consider the texture as insignificant. Estimating one optimal value of λ is not necessarily good either. A surface may be covered by a "mid-scale texture" itself covered by a small-scale texture, and we may consider any combination of those as the signal of interest. Basically, this is the problem of *scale-space* [**65**]. It will be discussed in other sections as well.

1.2.5. *Method classification.* In standard regularization, the choice of stabilizing functional and of a quadratic norm is done to ensure the convexity of the functional to minimize. The standard regularization methods therefore fall in the global minimization of convex energies class.

The attempts made at finding the right stabilizing functional for regularization were concerned in a large part with the limitations of the regularization approach, such as the need for a semi-norm property [**47**]. What is most important however is the knowledge of the constraint subset, if it exists, or as expressed by Terzopoulos [**60**, p. 434]:

> . . . the surface model can be related to expectations regarding the class of admissible surfaces.

It is true that there is a restricted class of surfaces that satisfy a particular regularization problem. For example, the admissible class for thin plate splines is the Sobolev space of order two [**58**]. However, even when the initial data can be interpolated exactly by one of the surfaces in the admissible class, the regularization operator may choose another (approximating) function of the admissible class instead of the interpolating one. More generally, it is shown in section 1 that an operator defined by standard regularization (with fixed values of λ and other parameters) is not idempotent. The class of admissible functions is therefore not a constraint subset in the sense of section 3.

However, we have seen in a previous section that when the regularization parameter is either 0 or ∞, the regularization method becomes a perpendicular projection. In these extreme cases, then, the method is idempotent, and can be expressed as a local minimization.

2. Terzopoulos' controlled continuity splines

2.1. Overview of the method.

2.1.1. *Problem description.* In his method, Terzopoulos wanted to be able to handle the presence of known step and crease discontinuities on the surface. In order to achieve this, the stabilizing functional may be locally

- disabled,
- the thin membrane functional, or
- the thin plate functional.

Therefore, the value of the stabilizing functional at a point depends not only on the differential properties of the solution surface at the point, but also on the knowledge of the desired continuity at that point.

The regularization is discrete, and is the solution of the minimization of

$$\mathcal{E}_{\boldsymbol{\rho\tau}}(\boldsymbol{v}) = \mathcal{S}_{\boldsymbol{\rho\tau}}(\boldsymbol{v}) + \mathcal{P}(\boldsymbol{v}) \tag{4.6a}$$

where $\mathcal{S}_{\rho\tau}(\boldsymbol{v})$ is a combination of the thin membrane and of the thin plate energies, where $\boldsymbol{\rho}$ and $\boldsymbol{\tau}$ control the local continuity of the surface. $\mathcal{P}(\boldsymbol{v})$ is the closeness to data term. It can also include surface orientation information, but this is not included here for simplicity.

$$\begin{aligned}\mathcal{S}(\boldsymbol{v}) = \frac{1}{2}\sum_{i,j}\rho_{i,j}\Bigg\{ & \tau_{i,j}\Big[(v_{i+1,j} - 2v_{i,j} + v_{i-1,j})^2 \\ & +2\left(v_{i+1,j+1} - v_{i,j+1} - v_{i+1,j} + v_{i,j}\right)^2 \\ & + \left(v_{i,j+1} - 2v_{i,j} + v_{i,j-1}\right)^2\Big] \\ & + [1-\tau_{i,j}]\Big[\left(v_{i+1,j} - v_{i,j}\right)^2 + \left(v_{i,j+1} - v_{i,j}\right)^2\Big]\Bigg\},\end{aligned} \tag{4.6b}$$

$$\mathcal{P}(\boldsymbol{v}) = \frac{1}{2}\alpha\sum_{i\in D}(v_{i,i} - d_{i,i})^2. \tag{4.6c}$$

Depending on the required continuity at point i, j, the values of ρ and τ are set as

$$\rho_{i,j} = 0, \quad \text{at a } C^0 \text{ discontinuity,} \tag{4.7}$$

$$\rho_{i,j} = 1, \tau_{i,j} = 0, \text{at a } C^1 \text{ discontinuity,} \tag{4.8}$$

$$\rho_{i,j} = 1, \tau_{i,j} = 1, \text{otherwise.} \tag{4.9}$$

Terzopoulos also introduced a variational continuity control method, in which the energy functional is minimized with respect to $\boldsymbol{\rho}$ and $\boldsymbol{\tau}$ as well as with respect to $\boldsymbol{v}$. A new term $\mathcal{D}(\boldsymbol{\rho}, \boldsymbol{\tau})$ is added to the energy functional in order to penalize the introduction of discontinuities:

$$\mathcal{E}(\boldsymbol{v}, \boldsymbol{\rho}, \boldsymbol{\tau}) = \mathcal{S}(\boldsymbol{v}, \boldsymbol{\rho}, \boldsymbol{\tau}) + \mathcal{P}(\boldsymbol{v}) + \mathcal{D}(\boldsymbol{\rho}, \boldsymbol{\tau}) \tag{4.10a}$$

where

$$\mathcal{D}(\boldsymbol{\rho}, \boldsymbol{\tau}) = \sum_{i,j} \left[\beta_d D_{i,j}(\rho) + \beta_O O_{i,j}(\tau)\right]. \tag{4.10b}$$

$D_{i,j}(\rho)$ and $O_{i,j}(\tau)$ are weights that depend on the structure of the discontinuities near the central point, and β_d and β_O are scaling parameters. The configuration of those weights is similar to Geman and Geman's line process, although in this case, discontinuities can be either steps or creases. Steps and creases do not interact with each others in these weighting functions. Such interactions should be included if all possible surface intersections are to be considered.

If one does not want to consider the structure of discontinuities, the simplest expression of $\mathcal{D}$ is

$$\mathcal{D}(\boldsymbol{\rho}, \boldsymbol{\tau}) = \sum_{i,j} \left[\beta_d[1 - \rho_{i,j}] + \beta_O[1 - \tau_{i,j}]\right]. \tag{4.10c}$$

Even then, the energy function is not convex anymore, and the problem cannot be considered one of standard regularization.

2.1.2. *Problem solution.* The first method presented by Terzopoulos produces a convex minimization problem. Therefore, all the standard methods of convvex optimization can be used. The variational continuity control produces a non convex functional, for which the global minimum has to be found. Terzopoulos attempts to find the global minimum by performing a number of minimizations over a family of functionals

$$\mathcal{E}_{\beta_d, \beta_O}(\boldsymbol{v}, \boldsymbol{\rho}, \boldsymbol{\tau}) \tag{4.11}$$

that are controlled by the values of β_d and β_O. These values are first set very high, disabling discontinuities, and making the functional convex. This functional is minimized using a gradient descent method, and the result is used as the initial value for a gradient descent on a new functional with lower values of β_d and β_O (first lowering β_d in steps and then lowering β_0 in steps), and so on, until a result is obtained for the desired functional. Although this approach was not formulated as a continuation method by Terzopoulos, this updating process is very similar to the graduated non-convexity solution of the first order plate of Blake and Zisserman.

2.2. Characterization of the solutions.

2.2.1. *Comparison with other methods.* The first method presented by Terzopoulos is standard regularization, where α is the regularization parameter. Since this is discrete regularization, the result of the algorithm is not a surface, but a new set of discrete points. The intent of Terzopoulos' method was to interpolate in between sparsely sampled data points, such as those produced by feature based stereo [**28, 39**]. If the regularization parameter is set at the high extreme, then the solution will exactly interpolate the initial data, and will also provide depth estimates at the grid points where no data was available. In the case of dense data, i.e. every grid point has a value, then discrete regularization with $\alpha \to \infty$ will behave as an identity operator. For either dense or sparse data, setting α close to zero will fit planes to the data within these regions delimited by the known discontinuities. As was said earlier, this produces an idempotent operator.

In the variational continuity control method, two other parameters, β_d and β_O, control the formation of discontinuities. As a variational method, the weak continuity constraint of Blake and Zisserman comes closest to this method. However, the weighting of discontinuities based on their local structure has more similarity with Geman and Geman's method (as acknowledged by Terzopoulos).

3. Geman and Geman's MAP estimate and stochastic relaxation method

3.1. Overview of the method.

3.1.1. *problem statement.* Geman and Geman use a Markov Random Field (MRF) model of the underlying image and formulate the problem as a maximum aposteriori probability (MAP) estimation. Because of the equivalence between the MRF model and a Gibbs distribution for the probability of the underlying surface, the problem is equivalent to minimizing a Gibbs energy functional dependent on the sampled data and the known data acquisition process.

Given a degraded image $\boldsymbol{G}$, the method seeks the image $\boldsymbol{f}$ that has the MAP probability of being the underlying image $\boldsymbol{F}$ that has produced $\boldsymbol{G}$. Using Bayes rule, this probability, for any $\boldsymbol{f}$, is given by

$$P(\boldsymbol{F}=\boldsymbol{f}|\boldsymbol{G}=\boldsymbol{g}) = \frac{P(\boldsymbol{G}=\boldsymbol{g}|\boldsymbol{F}=\boldsymbol{f})P(\boldsymbol{F}=\boldsymbol{f})}{P(\boldsymbol{G}=\boldsymbol{g})}. \tag{4.12}$$

Because we want to maximize this expression with respect to $\boldsymbol{f}$, we can ignore the denominator on the RHS, since it does not depend on $\boldsymbol{f}$. The second term in the numerator on the RHS depends on some a priori assumptions made on the probability of existence of an underlying image. Geman and Geman use a MRF model to determine this probability. In such a model, the probabilit that a pixel has a certain value, given the values of all the other pixels in the image, is only a function of the pixels in a finite neighbourhood of that pixel. This probability is given by

$$P(\boldsymbol{F} = \boldsymbol{f}) = \frac{e^{\frac{-U(\boldsymbol{f})}{T}}}{Z} \tag{4.13}$$

where Z is a normalizing factor, T is a *temperature* parameter used to control the simulated annealling process, and $U(\boldsymbol{f})$ is an energy function based on local properties of the image $\boldsymbol{f}$.

In order to compute the conditional probability in the numerator on the RHS of equation 4.12, the image formation process must be known. If the original image is given as $\boldsymbol{F}$, the degraded image, $\boldsymbol{G}$, is given by

$$\boldsymbol{G} = \Psi(\phi(H(\boldsymbol{F})), N) \tag{4.14}$$

where H is a blurring matrix, ϕ is some non-linear transformation, N is some arbitrary noise, and $\Psi(a, b)$ is some invertible function with inverse $\Phi(\boldsymbol{g}, \phi(H(\boldsymbol{f})))$. If the noise is known to be additive white noise, the conditional probability function is given by

$$P(\boldsymbol{G} = \boldsymbol{g}|\boldsymbol{F} = \boldsymbol{f}) = (2\pi\sigma^2)^{-M/2} e^{-(\mu - \Phi(\boldsymbol{g}, \phi(H(\boldsymbol{f}))))^2} \tag{4.15}$$

where M is the number of pixels in the image, μ and σ are the known mean and variance of the white noise. If we multiply this conditional probability with $P(\boldsymbol{F} = \boldsymbol{f})$, we obtain

$$P(\boldsymbol{F} = \boldsymbol{f}|\boldsymbol{G} = \boldsymbol{g}) = \frac{e^{\frac{-U^P(\boldsymbol{f})}{T}}}{Z^P} \tag{4.16}$$

where Z^P is a normalizing factor, and

$$U^P(\boldsymbol{f}) = U(\boldsymbol{f}) + \frac{(\mu - \Phi(\boldsymbol{g}, \phi(H(\boldsymbol{f}))))^2}{2\sigma^2}. \tag{4.17}$$

It is equivalent to maximizing the Gibbs distribution of equation 4.16 and to minimizing the Gibbs energy of equation 4.17.

3.1.2. *problem solution.* The Gibbs energy to minimize is not convex in general, so gradient descent methods are not sufficient to find the global minimum. Geman and Geman's formulation of the problem as a MAP estimate permits to directly use a stochastic relaxation method to find the global minimum. This method, as for the continuation methods used by Blake and Zisserman and Leclerc, produces a sequences of images that converges to the global minimum. In this method, the sequencing parameter is the temperature T, which is first set at a high value, and then gradually lowered towards zero. When interpreted as a statistical sampling method, the process evolves from a purely random sampler ($T = \infty$) to a purely deterministic one ($T = 0$). Geman and Geman have given a lower bound on the *cooling schedule* of T for which they have shown that the MAP estimate will be found by the method. [**12**] compares the stochastic relaxation method with the graduated non-convexity method.

3.2. Characterization of the method. The three models used to compute the a priori probability in [**26**] have been admittedly chosen ad hoc by Geman and Geman. The first of these models, the "blob process", is shown to give an energy functional equivalent to the piecewise constant case of Leclerc in section 1. In this comparison the data acquisition model was assumed to be additive Gaussian noise, but both Geman and Geman's and Leclerc's formulations can accept more complex image formation models in the energy functional. Since Blake and Zisserman's energy with $\lambda \to \infty$ has been shown to be equivalent to the piecewise constant case of Leclerc, it is also equivalent to the blob process of Geman and Geman.

The blob process does not make use of line processes at all. When such a model is used in the second example of Geman and Geman, consisting of constant intensity rectangles (with sides aligned with the sampling grid) degraded by additive Gaussian noise, the result of their algorithm gives rather jerky sides. Application of Leclerc's algorithm on similar data (also with additive Gaussian noise), also gives jerky sides at low contrast boundaries.

The addition of a line process in the a priori model permits the use of knowledge about the types of images processed. If the Gibbs energy is expressed algebraically (as in appendix C), it results in the addition of more Kronecker deltas to the function for the blob process. The energy, corresponding to the a priori model used in their second example, is presented in section 2. Since the line variables have only a few discrete values, it would not make much sense to use Leclerc's continuation method on the Kronecker delta terms involving the line variable. But this is a detail of how to solve the problem. On the study of the problem statement alone, one should note that the line process used fits very well with the data in the example. A more complex line process has to be used for the third example, which consists in a more natural "roadside scene" [**26**]. Among other things, line elements with different orientations must be allowed to take into account diagonal lines.

It is not obvious if a line process could be found that would be more useful than no line process at all if no priori assumptions can be made on the contents of the image. For example, it would have been interesting to see the result of using the blob process on the roadside scene.

One should note that the number of discrete intensity levels handled by the method is rather small (Due to having to sample a probability density), contrary to other methods such as Blake and Zisserman an Leclerc, where the intensity can be practically considered continuous.

The a priori models used by Geman and Geman in their examples are very restricted in that they assume piecewise constant images. There is nothing preventing the use of higher order models, but this would in turn require the distinction between different types of line processes, one for each type of discontinuity (as Terzopoulos does). This would involve rather complex expressions.

Because it can be expressed as the minimization of a gibbs energy functional, the method enters the category of global minimizations. Due to the equivalence between the blob process and the piecewise constant case of Leclerc, the method with the blob process gives an idempotent operator, as is proven for Leclerc in section 3. This method could therefore be expressed as a local minimization problem. Idempotency has not been demonstrated for the other line processes, but we are tempted to think that these methods are idempotent as well.

4. Blake and Zisserman's weak continuity method

4.1. Overview of the method. This method is very well described in the book [**13**]. The method is very similar to the variational continuity control of Terzopoulos, but it facilitates the determination of the global minimum by a clever combination of the smoothness functional and of the line process functional. How this is done will be shown here for the detection of step discontinuities in 1D, using a *weak string* model. Detection of 1D creases and of step and creases in 2D, using the same approach, are also discussed in the book.

In the weak string model, the energy to be globally minimized is

$$E = D + S + P, \tag{4.18}$$

where

$$D = \sum_{i=0}^{N} (u_i - d_i)^2, \tag{4.19}$$

$$S = \lambda^2 \sum_{i=1}^{N} (u_i - ui - 1)^2 (1 - l_i), \tag{4.20}$$

$$P = \alpha \sum_{i=1}^{N} l_i. \tag{4.21}$$

(4.22)

and l_i is one if there is a discontinuity between site $i-1$ and i, zero otherwise. The energy E is to be minimized with respect to $\boldsymbol{u}$ and $\boldsymbol{l}$.

This problem is equivalent to minimizing an energy F with respect to $\boldsymbol{u}$ alone, where

$$F = D + \sum_{i=1}^{N} g(u_i - u_{i-1}) \tag{4.23}$$

where

$$g_{\alpha,\lambda}(t) = min_{l\in\{0,1\}}\left(\lambda^2(t)^2(1-l)+\alpha l\right) \tag{4.24}$$

The energy F is not convex in general. Blake and Zisserman attempt to find the global minimum to F by using a *continuation method* they call *graduated non-convexity*. It consists in sequentially finding the minima of the family of functions

$$F^{(p)} = D + \sum_{i=1}^{N} g^{(p)}(u_i - u_{i-1}) \tag{4.25}$$

starting at $p = 1$, and using the result of one minimization as the starting point for the next. The family of functions is chosen such that $F^{(1)}$ is convex, and such that $F^{(p)}, p \to \infty$ tends to F. This is achieved by using

$$g^{(p)}_{\alpha,\lambda}(t) = \begin{cases} \lambda^2 t^2, & \text{if } |t| < q \\ \alpha - c(|t|-r)^2/2, & \text{if } q \leq |t| < r \\ \alpha, & \text{if } |t| \geq r \end{cases} \tag{4.26}$$

where

$$c = \frac{c^*}{p}, r^2 = \alpha\left(\frac{2}{c}+\frac{1}{\lambda^2}\right), \text{and } q = \frac{\alpha}{\lambda^2 r}. \tag{4.27}$$

and c^* is chosen such that $F^{(1)}$ is convex.

4.2. Characterization of the results.

4.2.1. *Relation with other methods.* The graduated non-convexity introduced by Blake and Zisserman, although not a novel concept (see [**1**] for a review of continuation methods), introduces a useful formalism in the domain of automatic discontinuity detection for surface reconstruction. Besides the formalism, however, their method is very similar to the variational continuity control of Terzopoulos, at the algorithmic level. The difference is mostly that Blake and Zisserman do not mix membrane and plate energies, whereas Terzopoulos does. Although it is not presented as formally as in Blake and Zisseram's work, Terzopoulos also uses some kind of a continuation method in order to find the global minimum of the regularizing spline with step and crease line processes. Indeed, the scaling factors for depth and orientation discontinuities are first set to a high value [**59**, P. 49]. The depth discontinuity factor is then gradually lowered, a minimization being performed each time, the preceding result used as a starting point for the current minimization. Only after the depth discontinuity factor is low enough is the orientation discontinuity factor lowered in steps, repeating the same sequential minimization process. Practically, the sequential lowering of the discontinuity factors is similar to the lowering of p in graduated non-convexity, and keeping the orientation discontinuity factor high during the detection of steps is very similar to the 1st order plate approach of Blake and Zisserman [**13**, p. 106].

Leclerc's method and Geman and Geman's method are still to be presented, but we state at this point that Blake and Zisserman's method is very similar to those methods when the regularization parameter tends to infinity. These similarities between the methods are presented in sections 1 and 2.

4.2.2. *Stability measure.* Although it is not stated explicitly in their book, the discontinuities found in Blake and Zisserman's method can be assigned a stability measure. Indeed, at any given minimization while the graduated non-convexity parameter p is lowered from 1 to a very small value, a discontinuity will be either unambiguous or ambiguous [**13**, p. 139]. A discontinuity that is unambiguous at a given step will never disappear during subsequent minimizations, while an ambiguous one may disappear. It is therefore natural to assign the value of p at which a discontinuity becomes unambiguous as its stability. This measure of stability is similar to the one described in [**36**].

4.2.3. *Computational feasibility.* Blake and Zisserman have extensively studied the feasibility of their method. They have showed that their algorithm did find the global minimum of the energy function in the case of the weak string with non-interacting discontinuities [**13**, appendix E]. They have not been able to prove that the global minimum is found in other cases, however.

4.2.4. *Gearing a problem statement from the available solution method.* The approach of Blake and Zisserman is an example of producing problem statements geared towards a particular method of solution. The graduated non-convexity method was naturally introduced for a weak string. A number of concessions had to be made however, to apply the functional to higher order energies, or in two dimensions. For example, the smoothing energy of the membrane is separated into the x and y components of the gradient, so that the one dimensional function $g(t)$ can be used [**13**, p.115]. The method of Blake and Zisserman is therefore limited in the choices of regularizing functionals and of line processes, due to the requirements of the graduated non-convexity process.

5. Sander's variational relaxation method

5.1. Overview of the method. The variational relaxation method of Sander is described in detail in chapter 3. Recall that the method consists in iteratively satisfying a locally constant curvature constraint, starting from an initial estimate of an augmented Darboux frame at every image point, and iterating until a convergence criterion is satisfied. However, it was noticed that iterating to convergence removed most of the local information content of the data. In experiments with computer generated and real data, it was observed that best results are obtained when the iterations are stopped possibly well before convergence. The number of iterations performed were interpreted as a scale parameter.

5.2. Characterization of the method. When using the parabolic extrapolation patch, we saw in chapter 3 that the result at convergence is equivalent to the least square fit of a plane, globally, to the data points.[2] As was noted in chapter 3, the parabolic patch does not implement the constant curvature constraint anyway, after a large number of iterations. At least, it serves the purpose of disproving the possibly intuitive conclusion that the surface at convergence is a gobal instantiation of the local extrapolating surfaces (a plane *is* a parabolic patch, but it is not as general).

When using the torus patch, the constant curvature constraint is better implemented, but what constitutes the constraint subset is not clear anymore. All spheres and all cylinders are obviously elements of the subset, because a torus patch may degenerate into a sphere and into a cylinder, and also because of the symmetry properties of these shapes. It is equally trivial to show that planes are included in the constraint subset. Besides these, it is not even obvious if torus surfaces are part of the constraint subset. This is because the torus patch osculating a general point of another torus does not necessarily osculate the torus everywhere else. Even without knowing the constraint subset exactly, we can nevertheless say that its element surfaces may have nonzero second partial derivatives, from the inclusion of sphere and cylinders, as well as intuitively. If we associate the effect of running Sander's algorithm to convergence with the effect of regularization with $\lambda \to \infty$, then the stabilizing functional of the equivalent regularization method should involve at least third order partial derivatives of the surface, which is one order higher than for the thin plate.

Experimentally, the method gives good results when it is not iterated to convergence. Only a few iterations are usually sufficient to remove noise from the principal direction fields. Also a local minimum of the overall residual, over the number of iterations, usually occurs after a few iterations, but what these local minima correspond to has not been found yet.

5.2.1. *Comparison with other methods.* As was mentioned in chapter 3, controlling the number of iterations of Sander's algorithm can be seen as controlling a scale parameter. The parameter λ of regularization methods can also be interpreted as a scale parameter [**13**]. The way in which the number of iterations of Sander's iterative updating and the regularization parameter control the scale is not quite the same, however. An attempt at comparing these controls is presented in appendix C. In the appendix, we note that using simpler constraints than constant curvature, we can find similarities between the iterative updating formulas of Sander and the iterative solution of regularization by traditional relaxation methods. In that sense, Sander's iterative updating is somewhat equivalent to the iterative solution of regularization problems with $\lambda \to \infty$, which as we know is an idempotent operation. Then, controlling the number of iterations of Sander's algorithm would be similar to the iterative solution of regularization problem, where the initial value is taken as the data (which usually does not matter in the iterative solution of regularization problems), and the algorithm

[2] This is a pretty expensive way of doing a least square fit!

is prevented from running to convergence. The effect of preventing the iterative solution of a regularization problem from approaching convergence is not discussed in the mathematical literature.

5.2.2. *Method classification.* Sander's variational relaxation, since it starts from an initial estimate of the augmented Darboux frame at every surface point, and iteratively progresses toward a set of frames that satisfy the constraints, is by definition an idempotent operator, if it is ran to convergence. The process could therefore be expressed as a local minimization, but the functional to locally minimize is not expressed explicitly in this method, since it consists of a number of convex minimizations at every point rather than a global one. A global energy functional is nevertheless defined in order to examine the convergence of the algorithm. This functional is defined as equation 3.5. This energy function does not describe a constraint surface that has an obvious geometrical meaning. Among other things, it has not been found yet if the constraint surface is convex or not. Examination of this global energy shows that is is not strictly decreasing as the algorithm progresses. That may be explained in part because the computed energy is not necessarily the one being implicitly minimized, at least not before a large number of iterations.

The operator corresponding to running Sander's algorithm without waiting for convergence is not idempotent. Indeed, if the algorithm is ran again on a previously obtained result, the iterations will simply pick up the minimization of energy were they left off, and the result after a given number of iterations will be closer to the result at convergence.

6. Besl and Jain's variable order surface fitting

6.1. Overview of the method.

6.1.1. *Problem description.* The goal of Besl and Jain's variable order surface fitting [**8**] is to recover a piecewise-smooth surface from noisy samples of it. The smooth regions of the recovered surface are modeled by polynomial patches in their paper. The order and distribution of these patches is to be adapted to the underlying surface, such that the number of regions is small (minimizing the number of discontinuities), and such that the difference between the reconstructed surface and the data is small. This is about as formal as the problem description of this method can be. The authors have purposefully not attempted to weight the closeness to data and the minimization of discontinuity terms, so no formal functional minimization is possible. Rather, the method seems to have been developed more as a series of procedures, that are described in the following section.

6.1.2. *Problem solution.* This method consists of two distinct steps. The first one consists in obtaining a preliminary surface segmentation (an initial guess) based on classifying surface regions in one of eight surface curvature types: peak, ridge, valley, saddle ridge, saddle valley, flat, and minimal. The curvature properties of the surface are determined by estimations of partial surface derivatives using convolution operators.

These initial regions are then used as seeds for a region growing process, based on function approximation over the regions being grown. The process proceeds iteratively one surface region at a time, starting with the largest region in the initial guess, and ending with the smallest one that has not been merged with some other one.

For a given region from the initial guess, the following steps are performed: First, the region is eroded to a size sufficient to perform the polynomial fits. Then polynomial surfaces are fitted to the seed region, starting with the lowest order surface (a plane), and continuing up to a maximum order, until a fit satisfies some error criteria. By proceeding in this way, the lowest order surface that gives satisfactory results will be used. Once the order of the polynomial has been selected, the seed region is grown by adding connected neighbouring pixels that are compatible to the fitted surface, under some criteria.

In this iterative procedure, regions that are grown at a given point may include a pixel that had previously been included in another region, if the surface fitted to that region better represents that pixel. The image segmentation process is therefore reversible while the iterations are taking place.

After the iterative region growing process, the authors of the method mention some other steps that are used to tidy up the results. Among other things, non adjacent regions that have compatible surface patches may be merged to become a single region.

6.2. Characterization of the method.

6.2.1. *Comparison with other methods.* This method is one of the few one presented in this thesis that does not permit to directly express the problem as an energy minimization. The method of Leclerc gives an end result that is very similar to this one, that is, a segmentation of the surface into polynomial patches of the minimum possible order. Leclerc's method is stated as an optimization, and the parameters that control it are clearly defined. However, by freeing themselves of an optimization approach, Besl and Jain can more easily include more global concepts in their method, such as the inclusion of non adjacent regions into a single surface patch model.

6.2.2. *Classification of the method.* Obviously, because it is not stated as a minimization problem, it is not straightforward to classify this method as one type of energy minimization. Because of the amount of steps in the process, and of the number of parameters and estimations involved, it was not attempted here to classify the method or to determine if it was idempotent

7. Leclerc's Minimum Description Length method

7.1. Overview of the method. The principle behind Leclerc's method [**36**] is to achieve the minimum description length (MDL) of a degraded and sampled image. Informally, it is much more efficient to describe the structured contents of a signal using a structured language, and to describe the noise contents of the signal using a statistical language. By obtaining the MDL description of a signal, the structure and noise contents of the signal will be naturally separated by the choices of descriptive language. Practically, the available languages to be used for the description must be enumerated beforehand, because there exists an infinity of possible languages in theory.

Leclerc chose to describe the structured parts of images with a polynomial model in which each image pixel is assigned a polynomial patch. Leclerc used from 0th (piecewise constant) to 2nd (piecewise quadratic) models. Leclerc does not use explicit line processes in [**36**]. Discontinuities are implicitly represented by discontinuities between neighbouring polynomial patches. The description language used by Leclerc for the non structured parts of images (the "noise") includes Gaussian blur, sampling, quantization, and additive white noise.

The goal of the method is therefore to describe a "real" image $\boldsymbol{z}$ as the sum of an underlying image $\boldsymbol{u}$ and of noise $\boldsymbol{r}$, such that the choice of $\boldsymbol{u}$ (and implicitly the choice of $\boldsymbol{r}$) minimizes the total description length of $\boldsymbol{z}$. The problem is then another one of global energy minimization, where the functional to minimize with respect to $\boldsymbol{u}$ is

$$|\mathcal{L}_u(\boldsymbol{u})| + |\mathcal{L}_r(\boldsymbol{z}-\boldsymbol{u})|, \tag{4.28}$$

where $\mathcal{L}_u$ and $\mathcal{L}_r$ are the languages used to describe the underlying image and the noise, respectively.

7.1.1. *Problem solution.* The above energy is not convex in general. Leclerc uses a continuation method which replaces Kroneker delta terms in the original energy by exponentials of the form

$$\exp\left(\frac{-t^2}{(s\sigma)^2}\right). \tag{4.29}$$

Note that this expression tends to $\delta(t)$ when s tends to zero. The parameter s is therefore varied to provide a sequence of minimizations, from a convex one, to the desired one. Leclerc has not proven that his continuation method finds the global minimum in general, but he showed that his method did find the global minimum of problems that could manageably be solved exactly by dynamic programming.

7.2. Characterization of the method.

7.2.1. *Comparison with other methods.* As was already stated in the preceding sections, it is shown in appendix C that the piecewise constant case of Leclerc with known additive white noise is equivalent to Geman and Geman's blob process, and to Blake and Zisserman's process when λ tends to ∞. In terms of the understanding of the parameters of the weak continuity methods, this would seem to say that the contrast sensitivity threshold $h_0 = \sqrt{2\alpha/\lambda}$ of the piecewise constant case is 0, but let us not forget that the contrast sensitivity of steps a apart, where $a \ll \lambda$ is increased by $\sqrt{\lambda/a}$, and that this is always the case for $\lambda \to \infty$. It is therefore consistent that the contrast sensitivity of MDL be in a useful range. The same argument can be used to predict that the gradient limit g_l is not zero. As for the immunity to noise, Blake and Zisserman found that no spurious discontinuities will be generated when $\alpha > 2\sigma^2$, approximately. Under the proportionality relation $\alpha = (b/a)\sigma^2$ of section 2, and using the values of $a = 1/(2\log 2)$ and $b = 2$ derived in Leclerc's thesis [**36**, p. 86], we obtain $\alpha = 1.44\sigma^2$, which shows that the two methods are in close agreement on that point.

The variable order surface fitting of Besl and Jain [**9**] is similar to the work of Leclerc in that the underlying surface is to be described by a small number of lowest possible order polynomial graph surfaces. However, the method used to obtain this result is very different from the one of Leclerc, and its bases are much less formal. Basically, a locally computed sign of curvature segmentation is used as the initial estimates for a region growing process that finds the boundaries of the final polynomial patches. The order of a polynomial patch is increased as the algorithm progresses, if the current order does not describe the data adequately. There is no explicit control of the continuity between the patches. That may cause the algorithm to produce interpatch discontinuities of a lower order than required by the maximum order of the polynomial patches. The algorithm does permit non connected regions to be represented with the same polynomial patch, which is a feature not yet implemented in the method of Leclerc. Finally, Besl does not seem to make a distinction between underlying discontinuities and model discontinuities, as described below.

7.2.2. *Underlying discontinuities and description discontinuities.* Leclerc's descriptive language is composed of two parts: One describes the *underlying surface*, the other describes the noise (including in one bag sensor noise, model induced distortion, surface noise, etc.), which is basically what the first part of the description cannot account for. He does not distinguish, however, between *underlying discontinuities* and *model discontinuities*. My claim is that if a piecewise polynomial surface of order n is used to describe the underlying surface, then the underlying discontinuities are those of order $0 \leq m < n$, and the model discontinuities are those of order n.

If an order n discontinuity is detected, then there is no way to know if it was caused by a discontinuity in the underlying surface, or by a too low order of the model. If it is known a priory that the model is adequate (piecewise), and only then, can the nth order discontinuities be considered to reflect discontinuities of the underlying surface. Also, in the former case, the position of nth order discontinuities is likely to be arbitrary and unstable. Examples of this can be found in the results presented in [**36**]. In the piecewise constant examples, the underlying surfaces *are* piecewise constant, so it is expected that the discontinuities will be located accurately. In the description of a face with piecewise first order polynomials, there are a number of unstable, apparently meaningless discontinuities which I assume are first order discontinuities due to the inadequacy of this model. These discontinuities should obviously not be considered as features of the image. A support of this interpretation is that these meaningless discontinuities disappear when a second degree description is used.

As for lower order discontinuities (n), those are necessarily reflexive of discontinuities in the underlying surface, because the model has one or more degrees of freedom that could be used to model the data instead of introducing a discontinuity.

7.2.3. *Where is the scale parameter?* Scale relates to rejecting the insignificant and keeping the significant in visual data. What is considered significant and what is considered insignificant depend in part on the scale of observation. At some extreme, even the "noise" may be considered a significant part of the data. In the description languages used by Leclerc, one part of the language describes the significant – the part using piecewise continuous polynomials – and another part accounts for the insignificant – which is considered as noise *even if there actually is structure in it*. An example of this last statement is the application of the piecewise constant case on a plane with a small enough slope for no discontinuities to be detected. In that case, the description of the data will consist in an horizontal plane, and the "noise" will be the inclined plane with zero average value.

In Leclerc's method, the various coefficients a, b, c, etc., are determined from an information theoretic point of view. The noise variance σ, however, may either be specified or determined as part of the problem, in the unknown noise case. In the case where the spatially dependent noise variance is automatically determined, the resulting values depend on how adequate the language for the description of the underlying surface is. In a sense, relating to the previous paragraph, the values of σ_i are found that leave as little structure as possible in the noise part of the description language. However, if we intend the underlying surface to be described at possibly different scales of interpretations, we might *want* some small scale structure to actually be included in the description of the noise. There are therefore reasons for using different values of the σ_i's for the description language.

Nevertheless, not *all scales* of observation are informative of the contents of the data. There exists a finite set of discrete scales at which the data presents some meaningful structure. This discrete segmentation of the scale axis must be combined with a segmentation of space to be the most useful. For example, while observing a wall with a plaque, one may simply consider the wall itself, ignoring the presence of the plaque. One may consider the plaque as a raised rectangle on the wall, and one may consider the writing on the plaque. It is useless to use the scale required to segment the writing from the plaque on the rest of the wall. We are therefore considering a discrete area in space and in scale at the same time.

Although it seems feasible to find the smallest informative scale, as does Leclerc, it is more difficult to automatically find the complete set of interesting scales (and spatial regions). Witkin [**65**] attempts to obtain a meaningful segmentation of the scale-space of a signal by generating an *interval tree* of the features tracked through scale-space, and by introducing a measure of *stability* of an interval. Uses of this automatic segmentation method of scale-space have not appeared yet. In the case of the MDL problem with unknown noise, it would be interesting to study the behavior of the local minima of the description length as the values of σ_i are changed. Hopefully, the value of these local minima seen as a function of σ_i, may have themselves a number of local minima that correspond to the meaningful scales of observation of the image. How to search for these local minima is not obvious though.

Finally, one note on the decision of using a constant $\boldsymbol{\sigma}$ within a continuous region. It has been noted in section 7.2.2 that the discontinuities in the highest order of the description language for the underlying data cannot be considered as discontinuities in the underlying data. Therefore, there should be no segmentation of $\boldsymbol{\sigma}$ due to these highest order discontinuities.[3]

7.2.4. *Classification of the method.* As it is stated in is PhD thesis, the method of Leclerc falls in the global minimization category. Also, in section 3, we prove that the method gives an idempotent operator for the piecewise constant description language with known noise process. It is therefore possible that the method can be represented as a local energy minimization.

[3]This introduces a problem, however. Although the discontinuities found by Leclerc's process form closed contours, these contours are not necessarily composed of the same order of discontinuity. By not considering order n discontinuities, regions may not be closed anymore. What to do with the value of σ in this case is not obvious.

CHAPTER 5

Conclusion

Computer vision research balances between specific techniques being developed, implemented and tested, and a kind of research brainstorm, where questions are asked about what has been achieved and what should be done next. In this thesis, the first part consisted in the adaptation of Sander's method to range data. In the other part, an organizing framework was established, methods were compared, and questions were asked. This chapter is mostly a summary of this second part, followed by suggestions on what to do next.

1. Summary of framework

It was proposed to use an energy minimization framework as a tool to compare surface reconstruction methods. The two main minimization problems are global minimizations, where the problem data is part of the functional to globally minimize, and local minimization, where the problem data is used as an initial point for the local minimization. Global minimizations can be convex, when the problem does not involve decision making. It is believed, but has not been proven, that all local minimization problems can be expressed as global minimizations. However, we know that not all global minimizations can be expressed as local minimizations. Finally, local minimizations have the desirable property of being idempotent.

2. About surface reconstruction methods

Standard regularization methods (more particularly Terzopoulos' controlled continuity splines), Terzopoulos' variationally controlled continuity splines, Blake and Zisserman's weak membrane and plate, Geman and Geman's MAP estimate, and Leclerc's MDL method can all easily be expressed in terms of a global energy minimization. The first method, which does not detect discontinuities, gives convex minimizations. The three last methods are related in the following way: The weak membrane, when the regularizing parameter tends to infinity, solves the same problem as Geman and Geman's MAP estimate, and as Leclerc's piecewise constant case with known noise. Also, because it was proven that this configuration of Leclerc's method is idempotent, the two other method configurations are as well. This also means that they could possibly be expressed as local minimization problems.

Sander's variational relaxation method and Besl and Jain's variable order surface fitting cannot easily be expressed in terms of an energy minimization. Due to the updating process used, we know that Sander's method is idempotent (if iterated to convergence), but because we have not been able to express it as an energy minimization, we have not been able to completely identify the constraint subset of the method. We know only that it contains spheres, cylinders, and probably other less known surfaces with some globally constant curvature properties. In the case of variable order surface fitting, the only similarity with other methods is in the form of the resulting surface, which is similar to Leclerc's method and its equivalent.

The goal of the comparison was to obtain a bette understanding of the problems they were attempting to solve, it was not to find a best method out of those compared. Because of the argument presented for idempotency, however, we think that more research should be done on those methods that are idempotent.

3. Directions for research

The result of applying Sander's method is still very much misunderstood. In particular the behaviour of the overall residual, as the iterations progress, is not understood. Are the local minima in this residual caused by numerical instabilities, or by the propagation of conflicting constraints from one surface neighbour to the other?

The energy minimization framework needs to be studied in much more depth and more formally. Most importantly, it should be determined if this formalism can in theory describe every surface reconstruction method possible. (That Sander's method and Besl and Jain' method were not found an equivalent representation within our framework does not mean that one does not exist.) It would be interesting to determine the relative complexity of using local or global minimizations to actually solve an idempotent problem. Current idempotent methods use global minimization. The problems would be easier to solve by local minimization if the local energy functional was available, but it is expected that this functional would be very expensive to compute and to store (although it need be computed only once since it does not depend on the data).

More formal reasons for the idempotency requirement are needed. The informal reason presented in this thesis is that a valid interpretation of some data should not be changed by another application of the interpretation process. Also, the other properties required by an idempotent operator for it to be expressible as a local minimization problem should be identified.

It is hoped that the questions raised in this thesis, on the similarity of surface reconstruction methods, on the establishment of a standard framework for their comparison, and on the requirement for idempotency of surface reconstruction methods, will contribute to foster more work in this field.

REFERENCES

[1] E. Allgower and K. Georg. Simplicial and continuation methods for approximating fixed points and solutions to systems of equations. *SIAM Review*, 22(1):28–85, Jan. 1980.

[2] A. Amini, T. Weymouth, and R. Jain. Using dynamic programming for solving variational problems in vision. Technical Report CSE-TR-05-88, Department of electrical engineering and computer science, the university of Michigan, Ann Arbor, Michigan, 48109-2122, 1988.

[3] C. Archibald and C. Merrit. Wrist mounted range profile scanners and applications. In *Workshop on range image processing, vision interface '89*, London, Ontario, Canada, June 1989. Canadian image processing & pattern recognition society.

[4] H. Asada and M. Brady. The curvature primal sketch. *IEEE Trans. PAMI*, 8:2–14, 1986.

[5] H. Barrow and J. Tenenbaum. Interpreting line drawings as three-dimensional surfaces. *Artificial intelligence*, 17:75–116, 1981.

[6] P. J. Besl. Describing geometric signals. In *workshop on range image processing, vision interface '89*, London, Ontario, Canada, June 1989. Canadian image processing & pattern recognition society.

[7] P. J. Besl, J. B. Birch, and L. T. Watson. Robust window operators. In *Second International Conference on Computer Vision*, Tampa, Florida, Dec. 1988. IEEE Computer Society.

[8] P. J. Besl and R. C. Jain. Invariant surface characteristics for 3d object recognition in range images. *Computer vision, graphics, and image processing*, 33:33–80, 1986.

[9] P. J. Besl and R. C. Jain. Segmentation through variable-order surface fitting. *IEEE Transactions on Pattern Anal. and Machine Intell.*, 10(2):167–192, Mar. 1988.

[10] J. M. Beusman, D. D. Hoffman, and B. M. Bennet. Description of solid shape and its inference from occluding contours. *Journal of the optical society of america a*, 4:1155–1167, July 1987.

[11] A. Blake. Reconstructing a visible surface. In *Proc. nat. conf. ai (aaai-84)*, pages 23–26, austin, Texas, 1984.

[12] A. Blake. Comparison of the efficiency of deterministic and stochastic algorithms for visual reconstruction. *IEEE transactions on pattern analysis and machine intelligence*, 11(1):2–12, Jan. 1989.

[13] A. Blake and A. Zisserman. *Visual Reconstruction*. The MIT Press, Cambridge, 1987.

[14] H. Blum. Biological shape and visual science (part i). *J. Theor. Biol.*, 38:205–287, 1973.

[15] P. Boulanger. Label relaxation technique applied to the topographic primal sketch. In *1988 Canadian vision conference*, pages 426–431, June 1988.

[16] T. E. Boult. What is regular in regularization? In *First international conference on computer vision*, pages 457–462, London, England, June 1987. ieee computer society.

[17] M. Clawes. On seeing things. *Artificial intelligence*, 2:79–116, 1971.

[18] P. Craven and G. Wahba. Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross-validation. *Numer. Math*, 31:377–403, 1979.

[19] M. P. do Carmo. *Differential geometry of curves and surfaces*. Prentice Hall, 1976.

[20] A. Dobbins, S. Zucker, and M. Cynader. Endstopped neurons in the visual cortex as a substrate for calculating curvature. *Nature*, 329:438–441, Oct. 1987.

[21] N. Dyn and D. Levin. Bell-shaped basis functions for surface fitting. In Z. Ziegler, editor, *Approximation Theory and Applications*, pages 113–129. Academic Press, 1981.

[22] F. Ferrie and J. Lagarde. Robust estimation of shape from shading. In *Image Understanding and Machine Vision Topical Meeting*, pages 24–27, Cape Cod, Massachusetts, June 1989.

[23] F. Ferrie, J. Lagarde, and P. Whaite. Darboux frames, snakes, and super-quadrics: Geometry from the bottom-up. In *Proceedings Workshop on interpretation of 3D scenes*, pages 170–176, Nov. 1989.

[24] F. Ferrie, J. Lagarde, and P. Whaite. Towards sensor-derived models of objects. In *Proceedings Vision Interface '89*, pages 166–174, June 1989.

[25] R. T. Frankot and R. Chellappa. A method for enforcing integrability in shape from shading algoerithms. *IEEE Trans. PAMI*, 10(4):439–451, July 1988.

[26] S. Geman and D. Geman. Stochastic relaxation, gibbs distribution, and the bayesian restoration of images. *IEEE Trans. PAMI*, 6:712–741, 1984.

[27] M. A. Gennert and A. L. Yuille. Determining the optimal weights in multiple objective function optimization. In *Second international conference on computer vision*, pages 87–89, Tampa, Florida, December 1988. IEEE computer society.

[28] J. Gibson. *The Perception of the Visual World*. Houghton Mifflin Company, Boston, 1950.

[29] G. Godin and M. D. Levine. Building the edge-junction graph from a range image of curved objects. In *Workshop on Range Image Understanding – Vision Interface '89*, 1989.

[30] G. D. Godin. Edge-based scene description using range imaging. Master's thesis, Departement of Electrical Engineering, McGill University, May 1989.

[31] W. E. L. Grimson. *From Images to Surfaces: A Computational Study of the Human Early Visual System*. The MIT Press, 1981.

[32] D. Hubel and M. S. Livingstone. Segregation of form, color, and stereopsis in primate area 18. *Journal of neuroscience*, 7(11):3378–3415, Nov. 1987.

[33] K. Itô, editor. *Encyclopedic Dictionary of Mathemathics*. The MIT Press, second edition, 1987.

[34] L. A. Iverson. The description of image curves: Discrete forms of continuity in space. Master's thesis, Dep. Elec. Eng., McGill University, 1988.

[35] J. Koenderink and A. van Dorn. Dynamic shape. *Biological Cybernetics*, 53:383–396, 1986.

[36] Y. G. Leclerc. *The local structure of image intensity discontinuities*. PhD thesis, Dept. Elec. Eng. McGill University, 1989.

[37] S. Lester J. Heider and J. E. Simpson. *Theoretical Analysis*. W.B. Saunders company, Philadelphia–London, 1967.

[38] D. Marr. Early processing of visual information. *phil. trans. r. soc. lond*, 275:483–519, Oct. 1976.

[39] D. Marr and T. Poggio. A computational theory of human stereo vision. *Proc. Roy. Soc. Lond., B*, 204:301–328, 1979.

[40] J. Meiguet. Multivariate interpolation at arbitrary points made simple. *Journal of applied mathematic physics*, 30, 1979.

[41] A. Mitiche, R. Grisell, and J. Aggarwal. On smoothness of a vector field – application to optical flow. *IEEE Trans. PAMI*, 10(6):943–949, Nov. 1988.

[42] F. Mojhtarian and A. Mackworth. Scale-based description and recognition of planar curves and two-dimensional shapes. *IEEE Trans. PAMI*, 8:34–43, 1986.

[43] P. Moon and D. E. Spencer. *Field Theory for engineers*. D. Van Nostrand Company, 1961.

[44] V. S. Nalwa and E. Pauchon. Edgel aggregation and edge description. *Computer Vision, Graphics, and Image Processing*, 40:79–94, 1987.

[45] R. Owens, S. Venkatesh, and J. Ross. Edge detection is a projection. *Pattern Recognition Letters*, 9:233–244, 1989.

[46] P. Parent and S. W. Zucker. Trace inference, curvature consistency, and curve detection. Technical Report CIM-86-3, Computer Vision and Robotics Laboratory, McGill Research Center for Intelligent Machines, McGill University, Québec, Canada, June 1985.

[47] T. Poggio, V. Torre, and C. Koch. Computational vision and regularization theory. *Nature*, 317(26):314–319, Sept. 1985.

[48] A. Pogorelov. *Differential geometry*. Noordhoff, Groningen, Netherlands, 1958.

[49] J. Ponce and M. Brady. Toward a surface primal sketch. A.I. Memo 824, Massachusetts Institute of Technology, Apr. 1985.

[50] P. Prenter. *Splines and Variational Methods*. John Wiley & Sons, New York, 1975.

[51] W. Richards and D. Hoffman. Codon constraints on closed 2d shapes. *Computer vision, Graphics, and Image Processing*, 31:265–281, 1985.

[52] W. Richards, J. J. Koenderink, and D. Hoffman. Inferring three-dimensional shapes from two-dimensional silhouettes. *Journal of the optical society of america a*, 4(7):1168–1175, July 1987.

[53] M. Rioux and L. Cournoyer. The nrcc three-dimensional image data files. Technical Report CNRC no 29077, Division of Electrical Engineering, National Research Council of Canada, Ottawa, Ontario, Canada, K1A 0R8, June 1988.

[54] A. Rosenfeld and M. Thurston. Edge and curve detection for visual scene analysis. *IEEE Transactions on Computers*, C-20:562–569, 1971.

[55] G. Roth. Range image segmentation based on differential geometry and refined with relaxation labelling. In *Workshop on Range Image Understanding – Vision Interface '89F*, 1989.

[56] P. T. Sander. *On Reliably Inferring Differential Structure from Three-Dimensional Images*. PhD thesis, Dept. Elec. Eng. McGill University, 1988.

[57] D. R. Smith. *Variational methods in optimization*. Prentice-Hall, 1974.

[58] D. Terzopoulos. The role of constraints and discontinuities in visible-surface reconstruction. In *Proc. 8th Int. Joint Conf. AI*, pages 1073–1077, Karlsruhe, W. Germany, 1983.

[59] D. Terzopoulos. Computing visible-surface representations. A.I. Memo No. 800, Massasucetts Institute of Thechnology Artificial Intelligence laboratory, Mar. 1985.

[60] D. Terzopoulos. The computation of visible-surface representations. *IEEE Trans. PAMI*, 10:417–438, 1988.

[61] H. J. Trussel and M. R. Civanlar. The feasible solution in signal restoration. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-32(2):201–212, Apr. 1984.

[62] B. Vemuri, A. Mitiche, and J. Aggarwal. 3d object representation utilising intrinsic surface properties. In *Proc SIAM conference on geometric modelling and robotics*, July 1985.

[63] G. Wahba. Improper priors, spline smoothing and the problem of guarding against model errors in regression. *Journal of the Royal Statistical Society., ser. B*, 40(3):364–372, 1978.

[64] H. William. *Numerical recipes: the art of scientific computing*. Cambridge university press, 1986.

[65] A. P. Witkin. Scale-space filtering. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, pages 1019–1021, 1983.

[66] N. Yokoya and M. D. Levine. Range image segmentation based on differential geometry: A hybrid approach. Technical Report McRCIM-TR-CIM 87-16, Computer Vision and Robotics Laboratory, McGill Research Center for Intelligent Machines, McGill University, Québec, Canada, Sept. 1987.

[67] S. Zucker and R. Hummel. A three-dimensional edge operator. *IEEE trans. PAMI*, 3(3):324–331, 1981.

APPENDIX A

Review of differential geometry concepts

1. Introduction

In this appendix, we will review concepts of classical differential geometry, which is the study of local properties of curves and surfaces [**19**]. Differential geometry takes singularities, or critical points, of curves and surfaces into consideration, but a general theory of critical points is a field by itself [**eells: ?**]. Global differential geometry links differential geometry and topology.

The concepts will not necessarily be presented in all their formality. The reader is referred to [**19**] and [**48**] for a more formal and complete treatment. Besl [**6**] presents a nice summary of differential geometry from the point of view of computer vision research. It is also only intended to cover those parts of the field that are of interest for this thesis.

The appendix is organized in the following way. First, definitions are given for the representation of curves and surfaces. Then, those differential properties that are of use in this thesis are presented. Singularities are first defined, followed by the properties of regular patches.

2. Notation

The following notational conventions apply to the whole thesis:

Inner Product: The general inner product will be denoted by $<, >$. However, the inner product in Euclidean space will be denoted by $\cdot$.

Cross Product: $\times$ will denote the vector cross product.

Derivatives: The derivatives of a function of a single variable will be given by $f'(t)$ to denote a first derivative, $f''(t)$ for a second derivative, etc.

Partial Derivatives: The partial derivatives of a function of several variables $f(u, v)$ will be given by f_u for the first partial derivative with respect to u, f_{uv} for the second order partial derivative in u and v, etc.

Orientation and Direction: To be sure that there are no confusions, we will adhere to the convention that a *direction* is defined by a line, and that an *orientation* is defined by a vector. This is the convention usually used in mathematics.

3. Representing curves and surfaces

Although curves and surfaces are represented in similar ways, it is impractical to discuss both at the same time. The representation of curves will first be discussed, followed by the representation of surfaces. The section on surfaces, however, will directly generalize from the one on curves, where it applies. The reason for considering curves at all is twofold. First, the differential geometric concepts are more easily introduced with curves, and second, even if this thesis is about surfaces, we might want to consider curves embedded in these surfaces.

3.1. Curve representation.

DEFINITION A.1. *An* elementary curve [**48**] *may be considered as the* image *of a* one-to-one *and* bicontinuous *mapping* $\boldsymbol{r} : I \to \Re^3$ *of an open interval* $I = (a, b)$ *of the real line* $\Re$ *into* $\Re^3$.

If the image of a curve is a planar subset of $\Re^3$, i.e., $\boldsymbol{r} : I \to \Re^2$, then the curve is furthermore called a *plane curve*.[1] A curved defined by such a mapping is called a *parametric curve* and is written as

$$\boldsymbol{r}(t) = (x(t), y(t), z(t)) \,, \tag{A.1}$$

or, for a plane curve,

$$\boldsymbol{r}(t) = (x(t), y(t)) \,. \tag{A.2}$$

One talks of $\boldsymbol{r}(t)$ as the equation of the curve, and of $x = x(t)$, $y = y(t)$, and $x = x(t)$ as the equations of the curve.

DEFINITION A.2. *A* simple curve *has the same definition as an elementary curve, except that the mapping may be from either an open interval or a* circumference*, i.e., the curve may be* closed*.*

DEFINITION A.3. *A* general curve *is the image of a simple curve under a continuous and* locally *one-to-one mapping.*

A curve that does not have a globally one-to-one mapping will have self intersections in its image.

DEFINITION A.4. *Finally, a* regular curve (of class C^k) *is one for which the parametric equation is* k*-times continuously differentiable* ($k \geq 1$) *and* $\boldsymbol{r}\,'(t) \neq 0$ *for all* $t \in I$.

[1] There is more similarity between a plane curve and a surface than for a general curve, because the plane curve is a mapping from $\Re^n$ to $\Re^{n+1}$, as for a surface. For example, there is no torsion associated with a surface or a plane curve, but there is one for 3D curves.

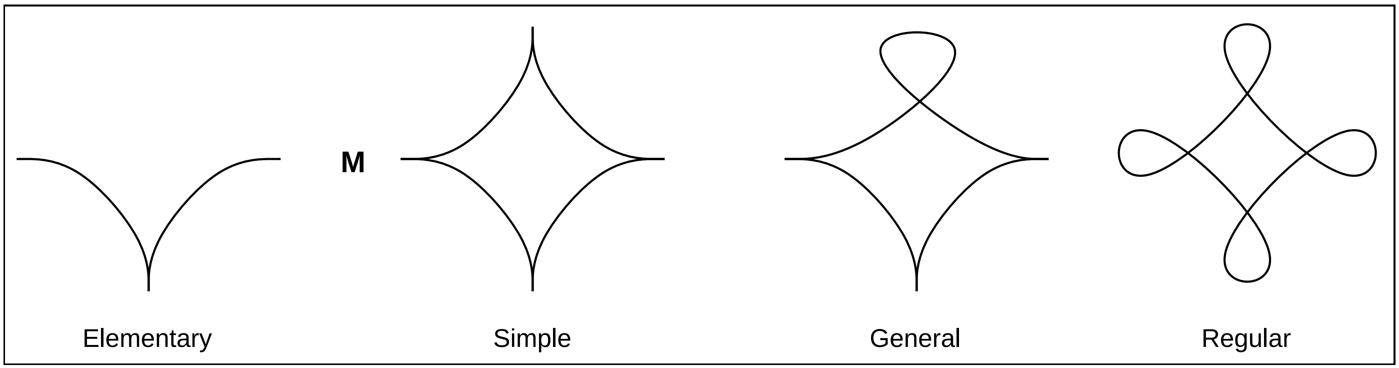


FIGURE A.1. Different types of curves. The first three are each subsets of each others, and they do not need to be C^1. A regular curve must be at least C^1, but it may be elementary, simple, or general, as the one pictured here.

A regular C^1 curve is called *smooth*.[2] A curve which is C^0 is not regular, but is called *continuous*. Figure A.1 gives examples of the types of curves defined above.

DEFINITION A.5. *The image set* $\boldsymbol{r}(I) \subset \Re^3$ *is also called the* trace *of* $\boldsymbol{r}$ [**19**].

One must make the distinction between the mapping that produced the curve and the curve itself, which is only the image of the map. It is important to consider the trace of curves and surfaces in detail, because a sensor will at most be able to produce a sampling of the trace of a curve or surface; the parametrization is meaningless in this context. An infinity of mappings may produce the same trace. In the case of elementary and simple curves, curves that have the same trace are considered equivalent; the parametrization is therefore not important in the geometric interpretation of the curve. For general curves, however, two curves with the same trace may be considered as distinct curves depending on their parametrization. For example the middle of the loop in figure A.2 may be seen as two crossing lines, as in A.2(a), or as two parts of the curve touching, as in A.2(b). Depending on the parametrization, the differential properties of the curve will change, and the curves will be considered distinct.

DEFINITION A.6. *As a convention, when context is not sufficient, a* geometric curve *will specifically refer to the trace of a curve, while an* analytic curve *will refer to its analytic representation, which is usually understood as parametric, although it could also be implicit, or otherwise.*

It is sufficient to consider only parametric representations in this thesis.

Note that the difference between a simple curve and a general curve has to do with *global geometry*, but the distinction is necessary to ensure that the trace of the simple geometric curve has only one corresponding analytic curve.

[2] Here, we use the definition of *smooth* found in [**48**], because of its qualitative appeal in the context of perception. Indeed, as long as a curve is C^1, it will be qualified as smooth by humans [**koend2: ?**]. [**19**] defines a smooth (or *differentiable*) curve as one which is C^∞.

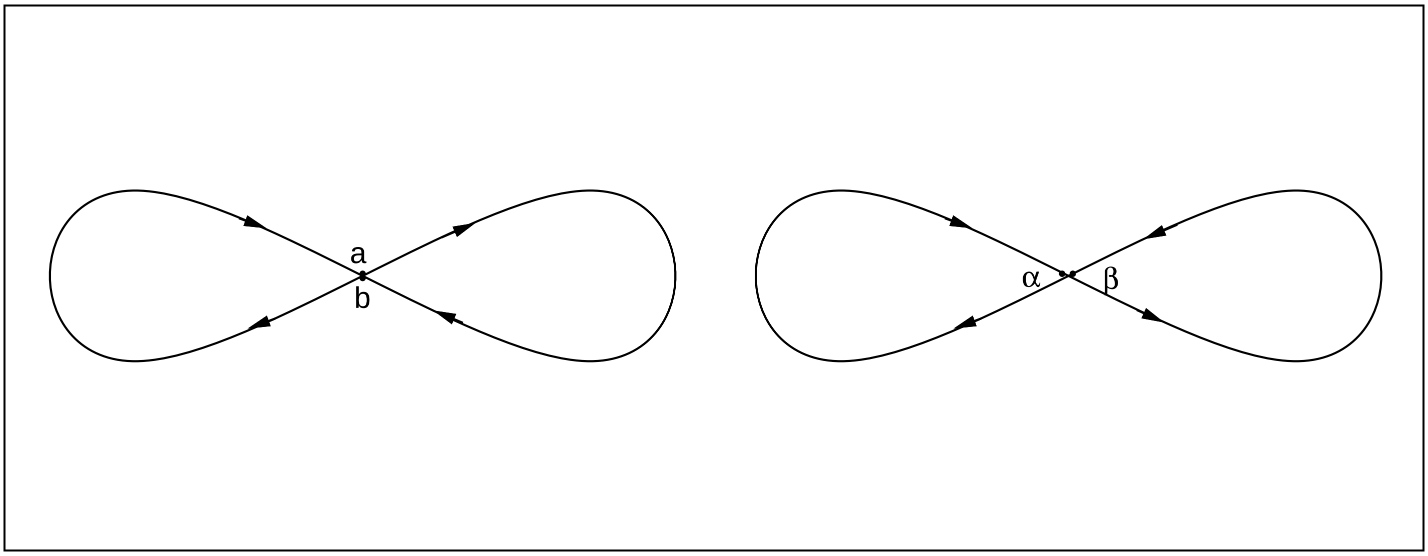


FIGURE A.2. These two curves have the same trace, but the arrows, indicating the direction of traversal of the domain circles, show that the parametrization are different. In particular, points a and b are singular, where points α and β are regular.

3.1.1. *special parametrizations.* A particular parametric representation which is useful in the context of computer vision is a *graph* representation. If a planar curve can be expressed parametrically in the form

$$x = t,\ y = \phi(t),\ t \in I \tag{A.3a}$$

(allowing for the choice of coordinate axes), then the equation of the curve can be written in the graph form

$$y = \phi(x). \tag{A.3b}$$

Note that for a regular curve, it is always possible, in a small enough neighbourhood, and with the appropriate choice of coordinates, to represent the curve in graph form [**48**].

Although it was said before that the parametrization of simple curves is irrelevant to the geometry of the curve itself, it is possible to make the parametrization relevant. For example the *unit speed*, or *natural*, parametrization identifies the parameter with the *arc length* of the curve. This other special parametrization will be presented in section A.4.1. Other ways to represent curves will not be discussed here.

3.2. Surface representation.

DEFINITION A.7. *An* elementary surface [**48**]*, similarly to an elementary curve, is the image of a one-to-one and bicontinuous mapping* $\boldsymbol{r} : U \to \Re^3$ *of an open set (or* elementary region*)* $U \in \Re^2$ *into* $\Re^3$.

Its equation is usually written as

$$\boldsymbol{r}(u,v) = (x(u,v), y(u,v), z(u,v))\,. \tag{A.4}$$

A *simple surface* cannot be defined as simply as was done with curves. A simple curve can always be expressed as either the image of an open segment, or as the image of a circle. However, a simple surface may not be representable as the image of a disk or of a sphere.

DEFINITION A.8. *A surface is a* simple surface *if it is connected and if at every of its points there is a neighbourhood such that the neighbourhood and the point form an elementary surface.*[3]

A simple surface may be closed, but the set difference between simple and elementary surfaces does not only consist of closed surfaces, as was the case for curves [**48**].

DEFINITION A.9. *The definition of a* general surface *generalizes directly from the one of a general curve.*

DEFINITION A.10. *A* regular surface (of class C^k) *is one for which the parametric equation is k-times continuously differentiable ($k \geq 1$) and $\boldsymbol{r}_u \times \boldsymbol{r}_v \neq 0$ for all $(u,v) \in U$, i.e. $\boldsymbol{r}_u, \boldsymbol{r}_v$ form a linearly independent set.*

Figure A.3 shows different types of surfaces.

DoCarmo [**19**, P.54] defines a regular surface in terms of a subset of $\Re^3$ with conditions on the set preventing self-intersection, but then he mentions (P. 78) that regular *parametric* surfaces could have self-intersections. We rather keep the parametric definition, and qualify the so defined regular surface as elementary, simple, or general (in which case self-intersections are permitted).

The image set $\boldsymbol{r}(U) \subset \Re^3$ will also be called the *trace* of the surface. The same comments as for curves apply, that is, the geometry of simple surfaces is completely characterized by their trace, but this is not true for general surfaces. Two simple surfaces are considered the same if their image set correspond, but a one-to-one and bicontinuous correspondence must also be established between the domain sets for two general surfaces to be considered equivalent. *Geometric surface* and *Analytic surface* will be used to explicitly differentiate between a surface defined by its trace, and one defined as a mapping.

Because surfaces in the world are usually not perceived as self intersecting, as witnessed in our everyday visual experiences, we will assume we deal with simple regular surfaces (as opposed to general regular surfaces), unless stated otherwise.

[3] From now on, when speaking of a neighbourhood on a surface, it will be understood to be that neighbourhood that forms an elementary surface.

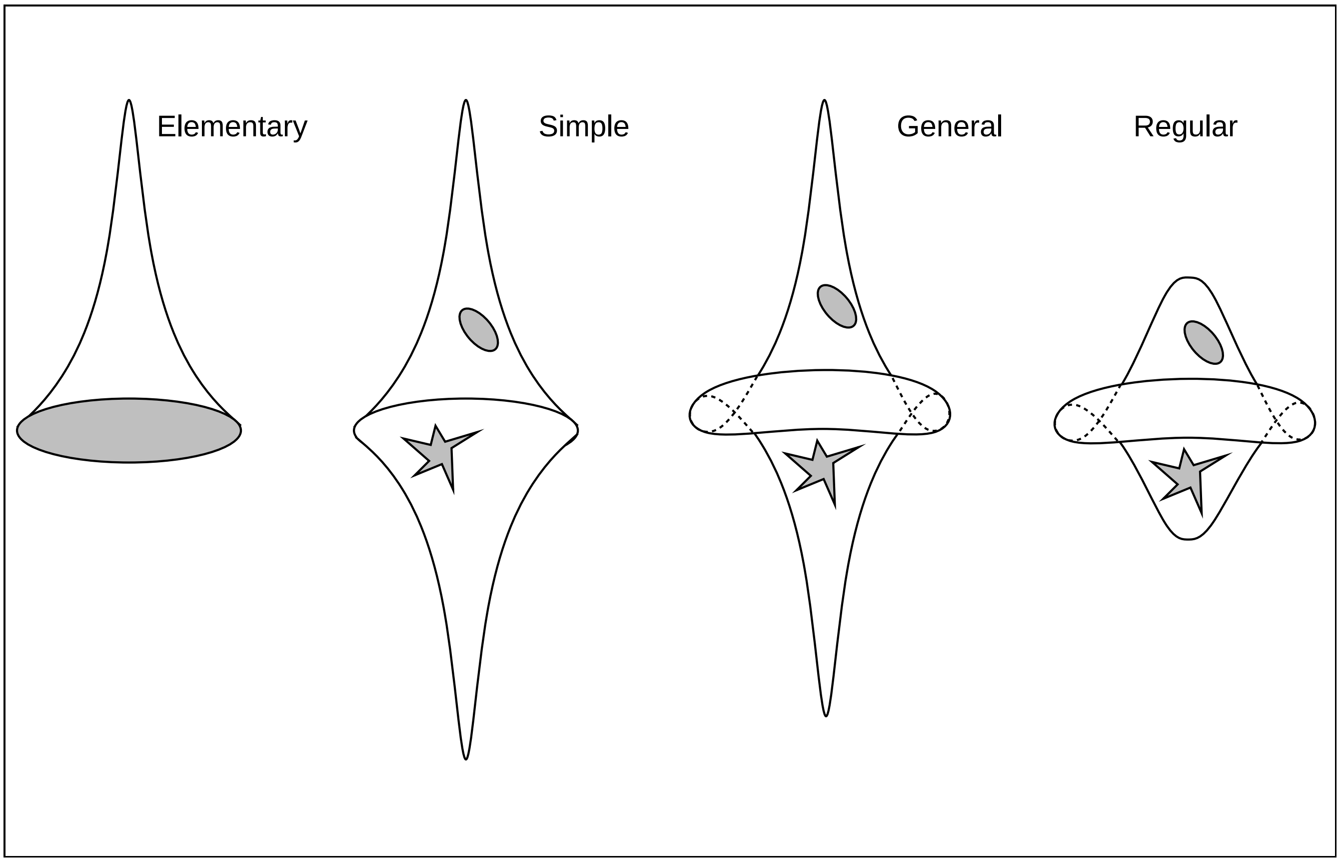


FIGURE A.3. Different types of surfaces. The first three are each subsets of each others, and they do not need to be C^1. A regular surface must be at least C^1, but it may be elementary, simple, or general, as the one pictured here.

3.2.1. *Special parametrizations.* The generalization of the *graph* representation for curves applies to surfaces represented as

(A.5a) $$x = u,\ y = v,\ z = \phi(u, v),\ (u, v) \in U$$

which can also be written as

(A.5b) $$z = \phi(x, y).$$

It is always possible to represent a neighbourhood of a simple surface in graph form, with the appropriate choice of coordinates. A graph representation of a surface is often called a *Monge patch*.

The *implicit representation* of a surface, defined as the set of points (x, y, z) which satisfy the equation $f(x, y, z) = 0$, is often useful as a compact definition of closed simple surfaces, and was often used to generate test data for this thesis.

4. Singularities

Singularity theory [**eells: ?**] is required in many fields, and the one of computer vision is not spared. Indeed, curves and surfaces in the world are not always perceived as regular, or smooth.[4] Recall that one of the conditions for the regularity of curves and surfaces was that $\boldsymbol{r}\,' \neq 0$ and $\boldsymbol{r}_u \times \boldsymbol{r}_v \neq 0$, respectively. If this condition is not met at a point, then this point is called a *singular point* of the curve or surface (for a more rigorous definition of surface singular points, see [**48**, p. 76]. In the case of surfaces, a set of connected singular points forms a *singular curve* on the surface. In general, we will consider surfaces that are *simple piecewise regular surfaces*. A piecewise C^k regular curve is a curve $\boldsymbol{r}$ defined on an open interval $[a, b]$ such that the curve is C^k regular on every open interval $[s_i, si + 1]$, for a finite set of points $a = s1, s2, \dots, b = sn$. Fractal curves and surfaces, which are studied in the context of computer vision [**pent,ben:frac: ?, pent,ben:frac: ?**], are not piecewise regular, but we rather consider the fractal nature of a surface as a texture that is to be separated from an underlying regular surface at a given scale of observation. Finally, note that we often talk of *almost everywhere* regular curves and surfaces in computer vision. This is an important distinction which summarizes the practical expectation that a surface has "not many" singular points. This expectation is enforced in one way or another in all surface reconstruction methods that permits singularities, for example [**26, 13, 56, 9**], to name a few.

5. differential properties

We are now ready to specifically discuss the differrential geometric properties of simple regular curves and surfaces. As before, the discussion will be separate for curves and surfaces.

5.1. differential properties of curves. The *unit tangent vector* $\boldsymbol{\tau}(t)$ of a curve at point t is given by the equation

$$\boldsymbol{\tau}(t) = \frac{\boldsymbol{r}\,'}{\|\boldsymbol{r}\,'\|}. \tag{A.6}$$

The unit tangent vector is unique, up to its orientation, which is dependent on the parametrization of the curve. Therefore, the *tangent line*, or tangent, is independent of parametrization for simple curves. The tangent is the line passing through $\boldsymbol{r}(t)$ and parallel to $\boldsymbol{\tau}(t)$. The meaning of the tangent can be intuitively understood in many ways; here, we will note that the tangent is itself a curve, and that it is the best linear approximation to the curve at point t. We can see that the tangent of a curve is defined only at regular points, where $\boldsymbol{r}\,' \neq 0$.

[4]The qualifying "perceived" is necessary, because, curves and surfaces are only mathematical concepts. After all, a physical surface is really mostly composed of "empty" space, at the sub-atomic scale.

We already introduced the arc-length of a curve when we talked about special parametrizations. The *arc length* of a segment $\tilde{\gamma}(a \leq t \leq b)$ of a simple regular curve $\gamma : \boldsymbol{r}(t)$ is given by

$$s(\tilde{\gamma}) = \int_a^b \|\boldsymbol{r}'(t)\| dt. \tag{A.7}$$

The arc length can be interpreted as the length of a polygon inscribed in the curve, as the number of polygonal arcs tends to infinity [**48**]. Since the arc length is independent of parametrization, and defines a *metric* on the curve, it is possible to use a standard parametrization, called the *natural parametrization*, to represent any given geometric curve. This parametrization is the one that satisfies $s(\tilde{\gamma}) = (b - a)$, or equivalently, $\|\boldsymbol{r}'(t)\| \equiv 1 \; \forall t$. A standard notation is to change the parameter t for s if the curve is naturally parametrized. The natural parametrization of a regular geometric curve always exists and is unique up to a sign difference in the parameter.

We shall now introduce the concept of *curvature* of a curve. The curvature measures the rate of change of the angle of the tangent line in terms of a displacement on the curve. Because the second derivative $\|\boldsymbol{r}''(s)\|$ measures the rate of change of the tangent vector, and because the parameter s precisely measures displacement on the curve (which is not the case for other parametrizations), the curvature κ of a naturally parametrized curve is simply

$$\kappa(s) = \|\boldsymbol{r}''(s)\|. \tag{A.8}$$

Curvature is an unsigned quantity by definition but a sign is sometimes assigned to the curvature of a plane curve, because the orientation of such a curve can be defined.

Since the graph parametrization of a plane curve is another parametrization that can be considered special, we shall show how curvature is interpreted in the context of a graph representation. First, note that the graph parametrization of a straight line segment parallel to the parameter axis is equivalent to the natural parametrization of the curve. Also, the graph parametrization of a straight line segment in general is only a constant factor away from the natural parametrization (for the curve $x(t) = t,\; y(t) = a,\; s = t\sqrt{1 + a^2}$). These cases may seem uninteresting, since the curvature of a straight segment is zero everywhere. However, these remarks also apply to the instantaneous orientation of the curve, as given by the tangent vector. Indeed, at a point t_0 where the tangent vector of a plane curve parametrized as in equation A.3b is parallel to the parameter axis, the curvature will be

$$\kappa(t_0) = \|\boldsymbol{r}''(t_0)\| = \left\| \frac{\partial^2 y}{\partial x^2} \right\|. \tag{A.9}$$

Therefore, the curvature of a curve at any point t_0 may be seen as the magnitude of the second derivative of the graph parametrization of the curve in a coordinate frame such that the tangent of the curve at point t_0 is parallel to the parameter axis. The idea of assigning a different graph parametrization at every point of a curve is implemented by a *moving frame*. One such moving frame, the *Frenet trihedron*, is very useful in differential geometry. Explaining the Frenet trihedron, as well as the torsion of a curve, will be presented here for the sole purpose of introducing the Fundamental theorem of the local theory of curves.

A point s such that $\boldsymbol{r}\,''(s) = 0$ is called a singular point of order 1. At all other points of a curve, there is a well defined unit vector $\boldsymbol{N}(s) = \boldsymbol{r}\,''(s)/\|\boldsymbol{r}\,''(s)\|$, called the *normal vector* to the curve at s. The plane spanned by $n(s)$ and $t(s) = \boldsymbol{r}\,'(s)$ (which are always perpendicular for a naturally parametrized curve) is called the *osculating plane* at s. These two vector define a third vector $\boldsymbol{b}(s) = \boldsymbol{t}(s) \times \boldsymbol{N}(s)$ called the *binormal vector* at s. These three vectors form an orthonormal moving frame parametrized by arc length, called the Frenet trihedron. The *torsion* $\tau(s) = -\boldsymbol{b}'(s) \cdot \boldsymbol{N}(s)$ of a curve measures the rate of change of the osculating plane in terms of a displacement on the curve. The vectors $\boldsymbol{t}'(s)$, $\boldsymbol{N}'(s)$, and $\boldsymbol{b}'(s)$ are all interrelated through the curvature and torsion, by the use of the *Frenet formulas*, which will not be presented here. We are now ready to present the following theorem, cited from [**19**], which sums up our geometric knowledge of a curve.

FUNDAMENTAL THEOREM OF THE LOCAL THEORY OF CURVES. Given arbitrary regular functions $\kappa(s) > 0$ and $\tau(s)$, $s \in I$, and $\kappa(s) > 0$, there exists a unique (up to position in space) curve $\boldsymbol{r} : I \to \Re^3$ such that s is the arc length, $\kappa(s)$ is the curvature, and $\tau(s)$ is the torsion of $\boldsymbol{r}$. [**48**]

What this theorem says, basically, is that the geometry of a curve can be uniquely characterized in terms of its curvature $\kappa(s)$ and torsion $\tau(s)$, *parametrized by arc length*. In some other parametrization, we would need the three quantities $\kappa(t)$, $\tau(t)$, and $s(t)$, that we will call the *intrinsic equation of the curve*. It is not surprising that we need three parametrized quantities in general, since the extrinsic parametrization of a curve is represented by three quantities $x(t)$, $y(t)$, and $z(t)$, and three bases are required to represent manifolds in 3D space. What is interesting is that we are able to use a basis where each base expresses a specific intrinsic geometric concept about the curve , rather than extrinsic coordinates, which depend on the representation of the embedding space. Finally, note that having *discrete* measures of curvature and torsion of a curve (even of arc length) does not characterize the curve uniquely, no more than having discrete measures of its coordinates in space.

5.2. Differential properties of surfaces. As for curves, we will begin this section with the notion of tangency. The *tangent plane* to a surface S at a point $P(u, v)$, denoted by $T_P(S)$, is the plane trough P spanned by the basis $(\boldsymbol{r}_u(u, v), \boldsymbol{r}_v(u, v))$, which are two linearly independent vectors tangent to the surface (recall the regularity requirements). Any linear combination of this basis lies in the tangent plane, and is called a *tangent vector* to the surface at P. The vector $\boldsymbol{N} = (\boldsymbol{r}_u \times \boldsymbol{r}_v)/\|\boldsymbol{r}_u \times \boldsymbol{r}_v\|$ perpendicular to the tangent plane is called the *unit normal* to the surface at P. The tangent plane of a surface does not depend on its parametrization. The set $(\boldsymbol{r}_u, \boldsymbol{r}_v, \boldsymbol{N})$ forms a moving frame on the surface, but it is not necessarily orthonormal, and more importantly, it is not completely intrinsic to the surface.

A curve C *lies on a surface* when every of its points belongs to the surface. Since the curve is embedded in a two-dimensional space (the surface), it can be expressed by only two equations, even if it is not planar in space. A practical way of doing this is to express the curve in terms of the parameters of the surface, in the following way. Let $C : \boldsymbol{r}(t)$ be a curve on the surface $S : \boldsymbol{r}(u,v)$. Then, the path of the curve on the surface can be given as $\boldsymbol{r}(t) = \boldsymbol{r}(u(t), v(t))$, which is often simply denoted $\boldsymbol{r}(t) = (u(t), v(t))$. The *parametric curves* of a surface are given by $\boldsymbol{r}(u,b)$ and $\boldsymbol{r}(a,v)$, a and b constants.

Now that there is a unique tangent space assigned to every point of a simple regular surface, we can introduce an operator, the *first quadratic form*, which is defined in this tangent space. This operator simply performs the inner product of vectors that lie in the tangent space of a point P of the surface, i.e., vectors that can be expressed in the basis $(\boldsymbol{r}_u, \boldsymbol{r}_v)$ of the tangent space. This quadratic form is defined as:

$$I_P(\boldsymbol{d}, \boldsymbol{\delta}) \equiv du\delta u\, {\boldsymbol{r}_u}^2 + du\delta v\, \boldsymbol{r}_u \cdot \boldsymbol{r}_v + dv\delta u\, \boldsymbol{r}_u \cdot \boldsymbol{r}_v + dv\delta v\, {\boldsymbol{r}_v}^2 \tag{A.10}$$

$$= E\, du\delta u + F\, du\delta v + F\, dv\delta u + G\, dv\delta v \tag{A.11}$$

$$= \begin{pmatrix} du & dv \end{pmatrix} \begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix} \begin{pmatrix} \delta u \\ \delta v \end{pmatrix}, \tag{A.12}$$

and let $I_P(\boldsymbol{d})$ denote $I_P(\boldsymbol{d}, \boldsymbol{d})$. In this equation, we use the notation of [**48**], which interprets the form as a differential operator. Indeed, $I_P(\boldsymbol{d})$ measures the infinitesimal distance covered on the surface at point P in the direction $\boldsymbol{d}$, and is called the *element of arc length* ds on S. The coefficients of the first quadratic form are usually denoted by E,F, and G, and (g_{ij}) is called the *metric tensor* of the surface at P. The form itself is independent of parametrization, but the individual coefficients are dependent. The first quadratic form permits to measure the length of curves on the surface, the angle of intersection of curves on the surface, and the area of the surface bounded by a closed curve on the surface. The curves of parametrization of a surface are orthogonal when $F = 0$ in this parametrization. The first quadratic form, alone, does not uniquely characterize a surface; more than one geometric surface may have the same first quadratic form.

There is another quadratic form of interest that is also an inner product of two vectors in the tangent space of S. However, this quadratic form, called the *second quadratic form*, restricts one of the vectors to be $d\boldsymbol{N}_P(\boldsymbol{v})$, the differential of the normal at P in the direction of a vector $\boldsymbol{v}$ of the tangent space of the surface. The form is defined as

$$II_P(\boldsymbol{v}) \equiv -d\boldsymbol{N}_P(\boldsymbol{v}) \cdot \boldsymbol{v} \tag{A.13}$$

$$= (\boldsymbol{r}_{uu} \cdot \boldsymbol{N})du^2 + 2(\boldsymbol{r}_{uv} \cdot \boldsymbol{N})dudv + (\boldsymbol{r}_{vv} \cdot \boldsymbol{N})dv^2 \tag{A.14}$$

$$= Ldu^2 + 2Mdudv + Ndv^2 \tag{A.15}$$

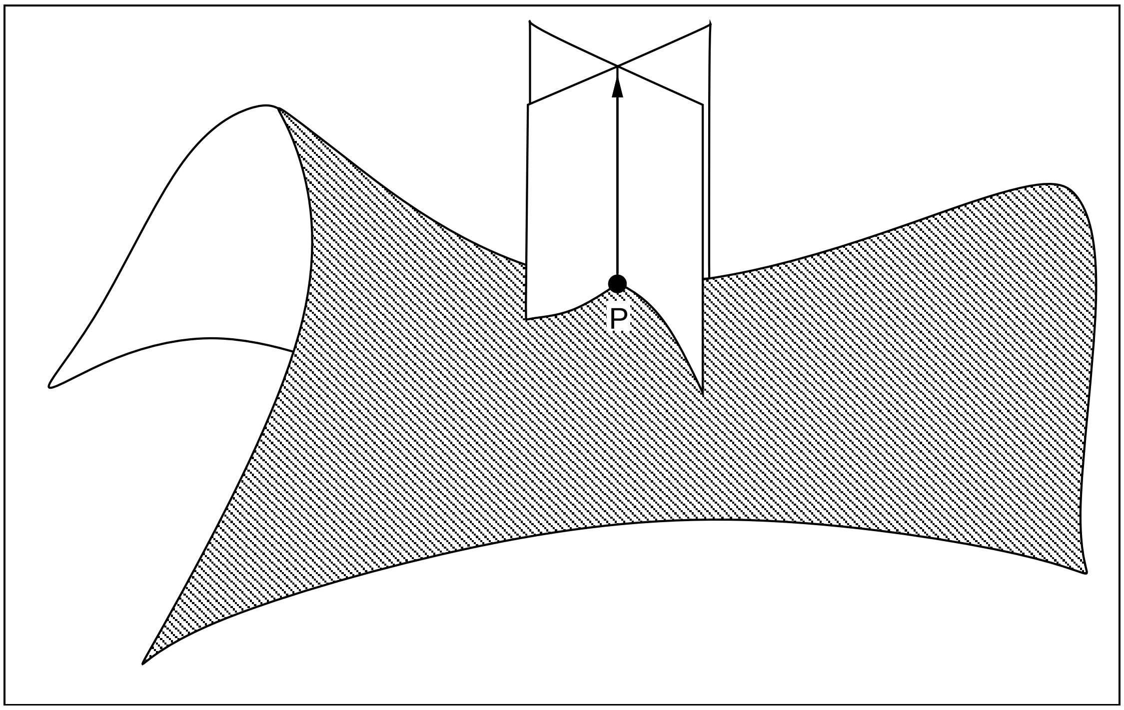


FIGURE A.4. Normal curvature from the normal section of a surface.

When $\boldsymbol{v}$ is a unit vector, II_P measures the rate of change of the surface normal in the direction $\boldsymbol{v}$ on the surface. This quantity looks a lot like the intuitive definition of curvature for a curve, and it is called the *normal curvature* κ_n of the surface at P in direction $\boldsymbol{v}$. Figure A.4 shows an intuitive interpretation of the normal curvature, in term of a *normal section*. A normal section is the intersection of the surface with a plane spanned by the surface normal and a tangent vector $\boldsymbol{v}$ at point P. This section is a plane curve, and its curvature at point P is the absolute value of the normal curvature of the surface at this point and in this direction. The sign is determined from the orientation of the surface normal. By rotating the intersecting plane about the normal, one obtains all possible values for the normal curvature at point P. The directions of that plane at which the normal curvature is minimum and maximum are called the *principal directions* of the surface at point P. The actual value of the normal curvature at these directions are the *principal curvatures* of the surface, and are respectively denoted by $II_P(\boldsymbol{M}_m) = \kappa_m$ and $II_P(\boldsymbol{M}_M) = \kappa_M$, $\boldsymbol{M}_m$ and $\boldsymbol{M}_M$ unit vectors. These two directions are always perpendicular. A curve C in S is a *line of curvature* if at every of its points, its tangent is a principal direction of the surface. The lines of curvatures form an orthogonal net on S, except at *umbilic points*, where no lines of curvature ever pass. An umbilic point of a surface is a point where $\kappa_m = \kappa_M$, i.e., where the surface is locally spherical, or locally planar. The *Gaussian curvature* K is equal to $\kappa_m \kappa_M$. The *mean curvature* H is equal to $\frac{\kappa_m + \kappa_M}{2}$. The

We end this section with the following theorem:

FUNDAMENTAL THEOREM OF THE LOCAL THEORY OF SURFACES (Bonnet). Suppose

$$Edu^2 + 2Fdudv + Gdv^2 \tag{A.16}$$

$$Ldu^2 + 2Mdudv + Ndv^2 \tag{A.17}$$

are two arbitrary quadratic forms the first of which is positive definite. Suppose the coefficients of these forms satisfy the Gauss and Peterson-Codazzi equations [**48**] Then there exists a surface, unique to within position in space, for which these forms are the first and second quadratic forms respectively. [**48**]

Because the six coefficients E, F, G, L, M, and N must satisfy the Gauss formula and the two Peterson-Codazzi formulas, there are really only three independent coefficients in the first and second quadratic forms. This makes sense, since three coordinates must be used to represent surfaces.

APPENDIX B

Finding the principal direction updates

As its name indicates, the principal direction of maximum or minimum curvature on a surface is only a direction – it is not oriented. If one uses vector notation to represent these directions, care should be taken to ensure that the orientation of the vectors is always treated as irrelevant. The updating formula for the principal directions originally used by Sander [**56**] uses the same minimizing norm as the one used to update the normals. The constrained minimization seeks the minimum of

$$J = \sum_{\alpha=1}^{n} \|\boldsymbol{M} - \boldsymbol{M}_{P\alpha}\|^2 + \lambda_1(\boldsymbol{M}^2 - 1) + \lambda_2(\boldsymbol{M} \cdot \boldsymbol{N}), \tag{B.1}$$

where the two Lagrange multipliers introduce the constraints that the principal direction vector be unit length, and that it be perpendicular to its associated normal (which was previously found from another minimization). The minimization of J with respect to $\boldsymbol{M}$ gives

$$\boldsymbol{M} = \pm \left(\frac{\boldsymbol{S} - (\boldsymbol{S} \cdot \boldsymbol{N})\boldsymbol{N}}{\|\boldsymbol{S} \times \boldsymbol{N}\|} \right), \tag{B.2}$$

where

$$\boldsymbol{S} = \sum_{\alpha=1}^{n} \boldsymbol{M}_{P\alpha}. \tag{B.3}$$

It is simply the normalized vector average of the principal directions $\boldsymbol{M}_{P\alpha}$ extrapolated from the neighbours, and projected on the plane defined by the corresponding normals $\boldsymbol{N}_{P\alpha}$. Because a vector average treats the orientation of the vectors as relevant, the resulting vector will not necessarily be directed along the average *direction* of the vectors, as is required. Figure B.1 demonstrates this fact with a simple example involving only two vectors.

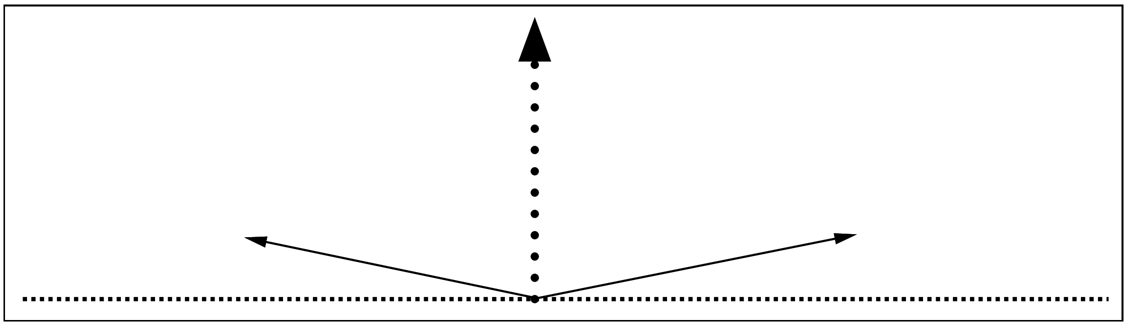


FIGURE B.1. Vector averages are not the same as direction averages. The dotted vector is the normalized vector average of the two solid ones. The dashed line is along the average *direction* of the two solid vectors.

The reason why the vector average gives an inadequate result in the example above is that the *vector field* (composed of only two vectors in this case) is discontinuous in its orientation, while the *direction field*[1] is continuous. It is not possible in general to produce a globally smooth vector field by assigning an orientation to the principal direction field of a surface. Figure 3.6 shows an example where any choice of orientation would produce set of discontinuities, and *not only at the umbilic point itself*, even if the umbilic is the cause of the non-orientability of the field. Therefore, the minimizing functional was reformulated to take only the direction of the vectors into account.

$$J = \sum_{\alpha=1}^{n} 1 - (\boldsymbol{N} \cdot \boldsymbol{M}_{P\alpha})^2, \tag{B.4}$$

which only measures the difference in directions, assuming all vectors are unit length.

Using Lagrange multipliers to enforce the required constraints on $\boldsymbol{M}$ proved impractical when J had to be minimized. Instead, the constraints were explicitly enforced on the answer by expressing $\boldsymbol{M}$ as

$$\boldsymbol{M}(\theta) = \boldsymbol{b_1} \cos\theta + \boldsymbol{b_2} \sin\theta, \tag{B.5}$$

where $\boldsymbol{b_1}$ and $\boldsymbol{b_2}$ satisfy

$${\boldsymbol{b_1}}^2 = {\boldsymbol{b_2}}^2 = 1, \tag{B.6}$$

$$\boldsymbol{b_1} \cdot \boldsymbol{b_2} = \boldsymbol{b_1} \cdot \boldsymbol{N} = 0. \tag{B.7}$$

Minimizing J with respect to θ then gives the update rule

[1]A direction field assigns to each $P \in U$ a line $\boldsymbol{r}(t)$ in $\Re^2$ passing through P [**19**].

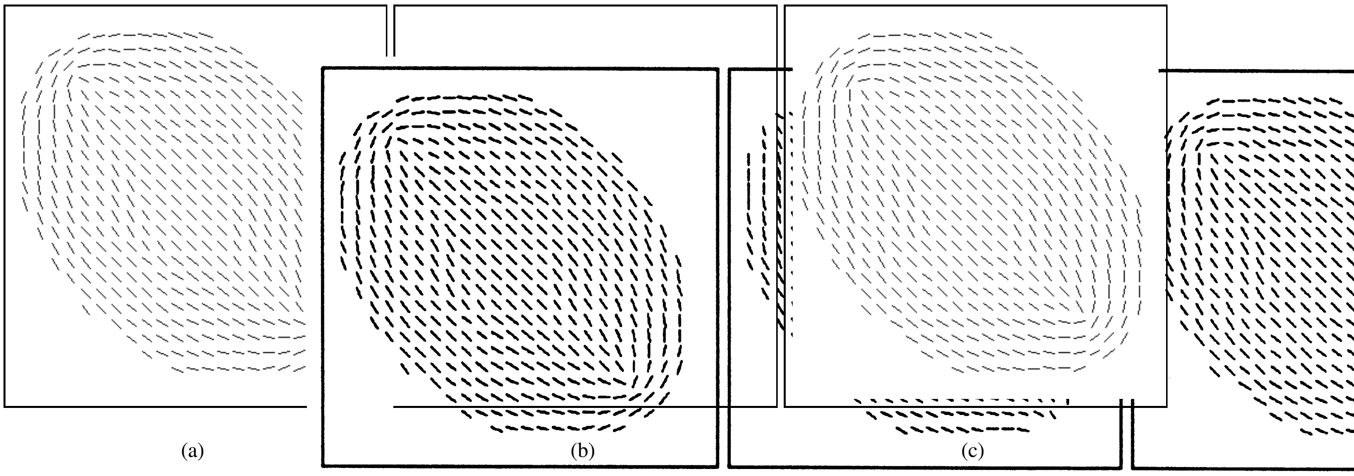
$$\theta = \tan^{-1}\left[\frac{(A_{22} - A_{11}) + \sqrt{(A_{22} - A_{11})^2 + 4A_{12}^2}}{2A_{12}}\right] \tag{B.8}$$

$$\boldsymbol{M}(\theta) = \boldsymbol{b_1}\cos\theta + \boldsymbol{b_2}\sin\theta, \tag{B.9}$$

where

$$A_{ij} = \sum_{\alpha=1}^{n}(\boldsymbol{M}_{P\alpha}\cdot\boldsymbol{b_i})(\boldsymbol{M}_{P\alpha}\cdot\boldsymbol{b_j}). \tag{B.10}$$

Results of using the old and the new updating rule on two iterations of the curvature consistency algorithm on an ellipsoid are shown in figure B.2. The new updating rule is clearly better at preserving the structure of the principal direction field.



(a) (b) (c)

FIGURE B.2. Figure (a) shows the analytically calculated principal direction field of an ellipsoid. This field is provided to the updating process as initial data. Figure (b) shows the result of two iterations of the algorithm using Sander's original update rule, which take orientation information into account. Figure (c) shows the result of two iterations using the new update rule presented in this appendix.

APPENDIX C

Details of comparisons between methods

This appendix contains a collection of detailed analyses of surface reconstruction methods, in completion of the discussion of chapter 4

1. Equivalence between piecewise constant MDL and Geman and Geman's "blob process"

The a priori Gibbs energy used in the first example of [**26**] used the eight-neighbor system and cliques $C = \{r, s\}$ of size two, for which the potential was

$$V_C(\boldsymbol{f}) = \left\{ \begin{array}{ll} \frac{1}{3}, & f_s = f_r \\ -\frac{1}{3}, & f_s \neq f_r \end{array} \right\} = \frac{1}{3}(2\delta(f_s - f_r) - 1). \tag{C.1}$$

There is no line process, so the Gibbs energy of the a priory probability distribution is given by

$$\begin{aligned} U(\boldsymbol{f}) &= \sum_C V_C(\boldsymbol{f}) && \text{(C.2)} \\ &= \frac{2}{3} \sum_{i,j=1}^{n} \Big\{ [2\delta\left(f_{i,j} - f_{i-1,j}\right) - 1] + [2\delta\left(f_{i,j} - f_{i,j-1}\right) - 1] \\ &\qquad + [2\delta\left(f_{i,j} - f_{i-1,j-1}\right) - 1] + [2\delta\left(f_{i,j} - f_{i+1,j-1}\right) - 1] \Big\} && \text{(C.3)} \\ &= \frac{4}{3} \sum_{i,j=1}^{n} \Big\{ [\delta\left(f_{i,j} - f_{i-1,j}\right) - 1] + [\delta\left(f_{i,j} - f_{i,j-1}\right) - 1] \\ &\qquad + [\delta\left(f_{i,j} - f_{i-1,j-1}\right) - 1] + [\delta\left(f_{i,j} - f_{i+1,j-1}\right) - 1] \Big\} \\ &\qquad - \frac{8n^2}{3}, && \text{(C.4)} \\ & && \text{(C.5)} \end{aligned}$$

where the $2/3$ term is due to every clique being counted twice, one for each pixel site. If we assume that the noise model is additive Gaussian noise with zero mean and variance σ, without blurring or nonlinearities, the a posteriori Gibbs energy is given by

$$U^P(\boldsymbol{f}) = U(\boldsymbol{f}) + \sum_{i,j} (g_{i,j} - f_{i,j})^2 / 2\sigma^2, \tag{C.6}$$

where $\boldsymbol{g}$ is the data. This energy is linearly proportional to the energy of the piecewise constant case of MDL, under the requirement that $a/b = 3/8 = 0.38$. The parameters used by Leclerc were $a = 1/(2\ln 2)$ and $b \approx 2$, such that $a/b \approx 0.36$. The energies to minimize, and therefore the operators, are very similar.

2. Equivalence between piecewise constant MDL and Blake and Zisserman's weak membrane

The solution to the piecewise constant minimum description length problem (with known constant variance) [**36**] is the minimizer of the functional

$$\begin{aligned} E_{\text{pcMDL}} = {} & \frac{a}{\sigma^2} \sum_{(i,j)\in I} (z_{i,j} - u_{i,j})^2 \\ & + \frac{b}{2} \sum_{(i,j)\in I} \sum_{(k,l)\in N_{(i,j)}} (1 - \delta(u_{i,j} - u_{k,l})) \end{aligned} \tag{C.7}$$

$$\begin{aligned} = {} & \frac{a}{\sigma^2} \sum_{(i,j)\in I} (z_{i,j} - u_{i,j})^2 \\ & + b \sum_{i,j} (1 - \delta(u_{i,j} - u_{i-1,j})) + b \sum_{i,j} (1 - \delta(u_{i,j} - u_{i,j-1})), \end{aligned} \tag{C.8}$$

when $N_{i,j} = \{(i-1,j), (i+1,j), (i,j-1), (i,j+1)\}$ (4 neighbours are used here, but Leclerc uses 8 neighbours).

The solution to the weak membrane model is the minimizer of the functional

$$\begin{aligned} E_{\text{weak membrane}} = {} & \sum_{i,j} (u_{i,j} - d_{i,j})^2 \\ & + \sum_{i,j} g_{\alpha,\lambda}(u_{i,j} - u_{i-1,j}) + \sum_{i,j} g_{\alpha,\lambda}(u_{i,j} - u_{i,j-1}), \end{aligned} \tag{C.9a}$$

where

$$g_{\alpha,\lambda}(t) = \begin{cases} \lambda^2 t^2 & \text{if } |t| < \sqrt{\alpha}/\lambda \\ \alpha & \text{otherwise} \end{cases} \tag{C.9b}$$

First, we observe that

$$\lim_{\lambda\to\infty} g_{\alpha,\lambda}(t) = \alpha(1 - \delta(t)) \tag{C.10}$$

Proof: First, $\lambda^2 t^2$ evaluated at $|t| = \sqrt{\alpha}/\lambda$ is equal to α. Second, $\lim_{\lambda\to\infty} \sqrt{\alpha}/\lambda = 0$. therefore, $g_{\alpha,\lambda\to\infty}$ is equal to α everywhere except at $t = 0$. ⊡

Therefore, $E_{\text{pcMDL}} \propto E_{\text{weak membrane}}$ when $\alpha = (b/a)\sigma^2$ and $N_{i,j} = \{(i-1,j),(i+1,j),(i,j-1),(i,j+1)\}$.

Proof: By using equation C.10, we directly obtain

$$E_{\text{weak membrane},\lambda\to\infty} = \sum_{i,j}(u_{i,j} - d_{i,j})^2$$

$$+\alpha\sum_{i,j}(1-\delta(u_{i,j}-u_{i-1,j})) \tag{C.11}$$

$$+\alpha\sum_{i,j}(1-\delta(u_{i,j}-u_{i,j-1})), \tag{C.12}$$

and the proportionality relation is straightforward to compute. ⊡

3. Comparison between Sander's iterative updating and regularization

As explained previously, Sander's algorithm is based on a constraint of locally constant curvature [**56**]. This constraint is enforced by iteratively taking local neighbourhood averages of the magnitude of the principal curvatures, of extrapolated normals, and of extrapolated principal directions, where the extrapolations take place on a surface patch that satisfies the constant curvature constraint. This approach earned the name of *variational relaxation* in section 2.4. The proof of convergence of such an algorithm is not obvious, because of the many interpolated quantities being averaged. It is even less obvious to describe what the algorithm is expected to converge to if it does.

In this appendix, we compare Sander's iterative updating approach with a discrete solution to regularization problems. In order to keep the complexity of the comparison down, we lower the dimensionality and the order of the constraints of Sander rather than formulating more complex regularization problems. We therefore consider the one dimensional string and rod models as regularization problems, and we compare their solutions by finite element methods with iterative updating methods comparable to the one used by Sander. The goal of this comparison is not to measure the effectiveness of the constraints used by Sander, since we use lower order constraints here. It is rather to gain an understanding of the iterative updating process he uses: Local averaging of quantities extrapolated from neighbouring patches instantiating a set of local constraints.

3.1. Comparison between a string model and a constant depth assumption. For the Sander-like updating to be of the same order as the iterative solution of a string model, let us use a constraint of locally constant depth (instead of locally constant curvature) in one dimension. Then, following Sander's approach, the algorithm would iteratively take local averages of the depth of neighbours (implicitly using an horizontal line as the extrapolating patch in this case). Using a neighbourhood of size 3, the update formula would be

$$u_i^{(k+1)} = \frac{u_{i-1}^{(k)} + u_{i+1}^{(k)}}{2}. \tag{C.13}$$

Now, we will show that enforcement of a continuous string model gives a similar iterative solution. The energy of the string model is given by

$$E = D + S \tag{C.14}$$

$$= \sum_{i=1}^{n} (u_i - d_i)^2 + \lambda^2 \sum_{i=1}^{n} (u_i - u_{i-1})^2 , \tag{C.15}$$

where D is the faithfulness to data ($\boldsymbol{d}$) measure, and S is the smoothness measure. Since this energy function is convex, its only minimum with respect to $\boldsymbol{u}$ can be found using the checkerboard Gauss Seidel iterative relaxation method. Using this method, the updating formula is given by

$$u_i^{(k+1)} = u_i^{(k)} - \frac{1}{T_i} (\partial E / \partial u_i) , \tag{C.16a}$$

where $T_i = \partial^2 E / \partial u_i^2$ in order to ensure convergence. Computing the partial derivatives, we obtain

$$u_i^{(k+1)} = u_i^{(k)} - \frac{1}{(1+2\lambda^2)} \left(u_i^{(k)} (1+2\lambda^2) - \lambda^2 u_{i+1}^{(k)} - \lambda^2 u_{i-1}^{(k)} - d_i \right) \tag{C.16b}$$

$$= \frac{\lambda^2 u_{i-1}^{(k)} + \lambda^2 u_{i+1}^{(k)} + d_i}{1+2\lambda^2}. \tag{C.16c}$$

This is a weighted average that is very similar to equation C.13. The main difference is that the data at node i is not included in the average of equation C.13, and that there is therefore no weighting of the terms. However, if we let $\lambda \to \infty$, then the updating rules C.13 and C.16c are identical.

It is now clear that a solution to satisfying a locally constant depth assumption can be reformulated as a local energy minimization. In the case of equation C.13, the energy function would simply be

$$E = \sum_{i=1}^{n} (u_i - u_{i-1})^2 \tag{C.17}$$

which only has, as its local minima, surfaces satisfying u_i =constant, $i = 1 \ldots n$.

Applying a locally constant depth constraint iteratively, without constraining the answer to stay near the initial data, achieves a globally constant depth at convergence. Thus, although things are not so simple for a locally constant curvature constraint, it appears that it will also be applied globally at convergence.

There is a final point that can still be discussed using the simpler constant depth assumption. This is the effect of varying the neighbourhood size in Sander's algorithm. Let the neighbourhood size be $2m+1$. Then, the updating rule will be

$$u_i^{(k+1)} = \frac{u_{i-m}^{(k)} + \cdots + u_{i-1}^{(k)} + u_{i+1}^{(k)} + \cdots + u_{i+m}^{(k)}}{2m} \tag{C.18}$$

$$= u_i^{(k)} - \frac{1}{2m} \left(2m u_i^{(k)} - u_{i-m}^{(k)} - \cdots - u_{i-1}^{(k)} - u_{i+1}^{(k)} - \cdots - u_{i+m}^{(k)} \right) . \tag{C.19}$$

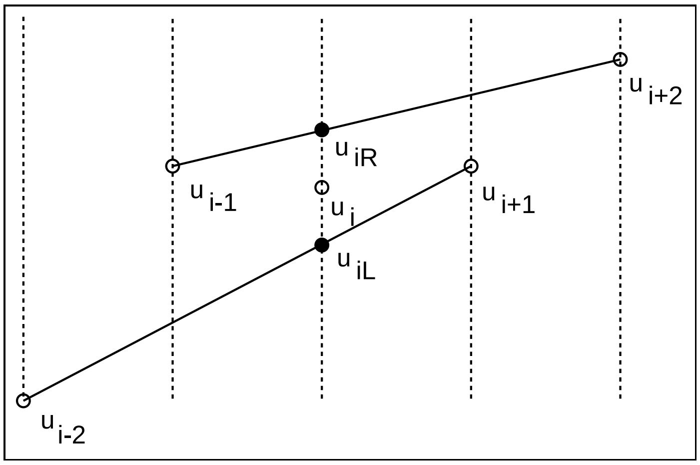


FIGURE C.1. Update values of u_i using a constant slope assumption.

We attempt to find a global energy functional E such that

$$\frac{\partial E}{\partial u_i} = 2mu_i - u_{i-m} - \cdots - u_{i-1} - u_{i+1} - \cdots - u_{i+m} \tag{C.20a}$$

and

$$\frac{\partial^2 E}{\partial u_i^2} = 2m. \tag{C.20b}$$

The corresponding functional (up to the integration constant) is given by

$$E = \sum_{i=1}^{n} (u_i - u_{i-1})^2 + \cdots + \sum_{i=1}^{n} (u_i - u_{i-m})^2 \, . \tag{C.21}$$

Applying the iterative updating rule C.18 will find the u_i's that minimize this energy functional. If the squared terms are to be interpreted as estimates of the squared gradient at u_i, then the estimates are less accurate as the value of m increases. Indeed, the first term is a true finite difference, but all the other terms are not normalized properly and are not as well localized.

3.2. Comparison between rod model and constant slope assumption. Moving one level up, we can attempt to compare the updating rule of the Gauss Seidel solution to the rod model and the updating rule of a constant slope constraint, where the data being updated is again the depth itself.

In that case the updating of the depth values follows a method similar to the one used to update the normals and the depths in Sander's algorithm. First, interpolated u_i's are produced by assuming a constant slope line passing through the neighbours. The projection is along the global z axis, as shown in figure C.1. Then, these interpolated depths are averaged in order to produce the updated u_i. Taking a neighbourhood size of 3, and taking the slope estimate at u_{i+1} to be $(u_{i+2} - u_{i-1})/3$ and the slope estimate at u_{i-1} to be

$(u_{i+1} - u_{i-2})/3$, the interpolated u_i's would be as follows (we omit the superscripts up to the final updating rule): u_{iR} interpolated from u_{i+1} satisfies

$$u_{i+1} - u_{iR} = \frac{u_{i+2} - u_{i-1}}{3} \Rightarrow u_{iR} = \frac{3u_{i+1} + u_{i-1} - u_{i+2}}{3} \tag{C.22a}$$

and u_{iL} interpolated from u_{i-1} satisfies

$$u_{iL} - u_{i+1} = \frac{u_{i+1} - u_{i-2}}{3} \Rightarrow u_{iL} = \frac{3u_{i-1} + u_{i+1} - u_{i-2}}{3}. \tag{C.22b}$$

Taking the average of these two values gives

$$u_i^{k+1} = \frac{4u_{i+1}^k + 4u_{i-1}^k - u_{i+2}^k - u_{i-2}^k}{6}. \tag{C.22c}$$

We would now want to compare this to the Gauss Seidel solution of the rod model. The energy of the rod model is given by

$$E = D + S \tag{C.23}$$

$$= \sum_{i=1}^{n} (u_i - d_i)^2 + \mu^4 \sum_{i=1}^{n} (u_{i-1} + u_{i+1} - 2u_i)^2 \tag{C.24}$$

where the smoothness term is now a quadratic term. The Gauss Seidel solution to minimizing this energy function (refer to the more detailed derivation for the string model) is given by

$$u_i^{(k+1)} = u_i^{(k)} - \frac{1}{(1+6\mu^4)} \left(u_i^{(k)} \left(1 + 6\mu^4\right) - 4\mu^4 u_{i+1}^{(k)} - 4\mu^4 u_{i-1}^{(k+1)} + \mu^4 u_{i+2}^{(k)} + \mu^4 u_{i-2}^{(k+1)} - d_i \right) \tag{C.25}$$

$$= \frac{4\mu^4 u_{i+1}^{(k+1)} + 4\mu^4 u_{i-1}^{(k)} - \mu^4 u_{i+2}^{(k)} - \mu^4 u_{i-2}^{(k+1)} + d_i}{1 + 6\mu^4}. \tag{C.26}$$

Again, this iterative solution is very similar to the one satisfying a constant slope assumption by local averagings, as in equation C.22c. Again, the only difference lies in the absence of the closeness to data term, which disappears if we let $\mu \to \infty$. This comparison of the rod model and of the application of a constant slope assumption using Sander's approach is not quite perfect. Indeed, if Sander's approach was to be followed exactly, the value of the slope of the curve would be explicitly stored at every point, and this slope value would be explicitly updated, rather than being implicitly updated by the depth updates. The slope would be estimated from finite differences only at the first iteration, not at all subsequent iterations, as was done here. Also, the projection of the point to be updated onto the extrapolation line would be done perpendicularly to the line.

The goal of the above comparisons was to demonstrate the similarity between Sander's approach and regularization method, when the regularization parameter tends to infinity. Finding a regularization expression which is technically closer to Sander's locally constant curvature constraint is surely possible, but it was not considered necessary here for the purpose of showing the similarities between the two methods.

4. Comparison between Sander's method and regularization in terms of constraint subsets

In the previous section the approach of Sander was lowered to a constraint of locally constant depth and of locally constant slope, in order to compare this constraint with the one of a string model. Here, we relate the methods in the context of a constraint subset.

The constraint subset of a constant depth assumption is the one parameter hypercurve $\boldsymbol{v}(t) = t$. The data vector $\boldsymbol{d}$ is considered as the initial value for Sander's constant depth constraint algorithm. The value of the solution $\boldsymbol{u}$ to Sander's constant depth algorithm is the perpendicular projection of $\boldsymbol{d}$ onto the hypercurve $\boldsymbol{v}(t)$, and is given by

$$1/n \sum_{i=1}^{n} d_i. \tag{C.27}$$

The solution to the string model for $\lambda = 0$ is $\boldsymbol{u} = \boldsymbol{d}$ [**63**, P. 366], which is equivalent to stopping Sander's algorithm before doing any iterations at all. The solution for $\lambda = \infty$ is $1/n \sum_{i=1}^{n} d_i$ [**63**], which is equivalent to letting Sander's algorithm iterate to convergence. There are four major ways in which one can get intermediate results between these two extremes (a scale-space of some sort). One is to vary the λ parameter between 0 and ∞ in the case of the string model. The second, also in the case of the string, is to use some numerical method (such as Gauss-Seidel) to achieve the minimum, but not to let the algorithm run to convergence. Normally, the initial value used in iteratively finding the global minimum is unimportant. However, if the initial value is taken as the data point, the intermediate results trace a path from the data to the global minimum of the regularisation problem. It is likely that the scale of the results increases along this path, but the properties of this particular scale-space are unknown. Similarly, and thirdly, the number of iterations performed by Sander's algorithm can be controlled, with the same question being asked: what do these intermediate values correspond to? Finally, in view of the perpendicular projection interpretation, one could take intermediate values along the straight hyperline joining the initial data vector to its perpendicular projection onto the constraint surface. In all these cases, pictured in figure 4, the intermediate solutions are obtained by moving along a one parameter hypercurve joining the initial data and its perpendicular projection onto the constraint surface, except in the case of regularization solved by a numerical method.

The values obtained along the perpendicular projection to the constraint surface are given by

$$u_{pi} = \frac{d_i}{1+\tau} + \frac{(\tau/n)\sum d_i}{1+\tau}, \tag{C.28}$$

where the parameter τ is zero when $u_{pi} = d_i$ and is infinity when $u_{pi} = u_i$. Interpreting the formula shows that these intermediate solutions are not very interesting. Indeed, the first term performs a uniform scaling

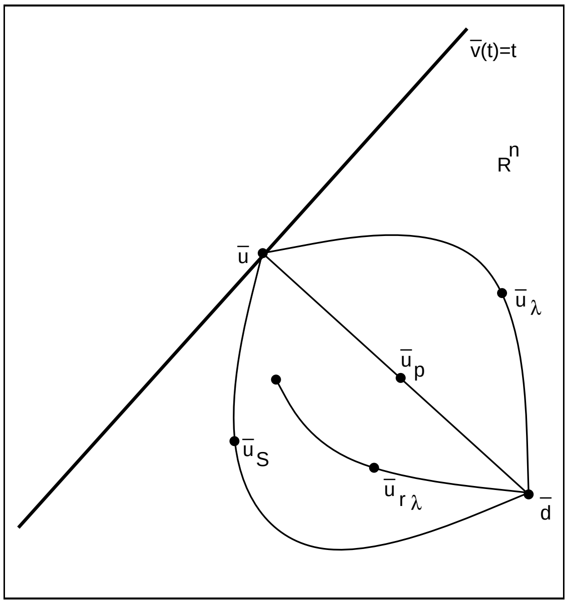


FIGURE C.2. Intermediate values between initial data and its perpendicular projection on a constraint surface. The data is composed of a vector of n points $\boldsymbol{d}$. The constant depth constraint is embodied in the hyperline $\boldsymbol{v}(t) = t$. The value corresponding to projecting $\boldsymbol{d}$ perpendicularly on the constraint surface, or to applying the string model with $\lambda = \infty$, or to running Sander's algorithm with a constant depth assumption up to convergence, is the point $\boldsymbol{u}$ on the constraint surface. The intermediate values along the perpendicular projection are $\boldsymbol{u}_p$; the intermediate values obtained by varying λ for the string model are $\boldsymbol{u}_\lambda$; the intermediate values obtained by running Gauss-Seidel for the string model and a given λ, but stopping the iterations before convergence are $\boldsymbol{u}_{r\lambda}$; the intermediate values obtained by iterating a limited amount of times using Sander's algorithm are $\boldsymbol{u}_S$.

for all the points, and the second term is the translation required to ensure that the average depth of the points of the intermediate value is the same as the average depth of the initial data. As for the other two cases, the behavior they have on the surface is not clear, but it seems obvious that the intermediate values will be closer to the constraint surface than the initial value. Knowing what happens when limiting the number of iterations of Sander's method is important, because it is clear that this is the only way in which the method will give interesting results. The result of stopping the iterative solution of the string model before convergence, is also not discussed in the standard literature, but it would be interesting to kown how such results differ from changing the value of the regularization parameter.

APPENDIX D

Supplementary detail on surface reconstruction methods

1. A class of non-idempotent minimizations

We claim that the operator

$$f(\boldsymbol{d}) = \min_{\boldsymbol{u}} \sum_{i=1}^{n} (u_i - d_i)^2 + S(\boldsymbol{u}), \qquad S(\boldsymbol{u}) > 0, \tag{D.1}$$

is not idempotent in general (with some continuity restrictions on $S(u)$).

Proof: First, apply the operator once on some data $\boldsymbol{y}$, obtaining the result $\boldsymbol{z}$.

$$\boldsymbol{z} = \min_{\boldsymbol{u}} E_{\boldsymbol{y}}(\boldsymbol{u}) = \min_{\boldsymbol{u}} \left(\sum_{i=1}^{n} (u_i - y_i)^2 + S(\boldsymbol{u}) \right) \tag{D.2}$$

Whether $E_{\boldsymbol{y}}(\boldsymbol{u})$ is convex or not, as long as its first partial derivatives are defined in the neighbourhood of the solution,

$$\left. \frac{\partial E_{\boldsymbol{y}}(\boldsymbol{u})}{\partial u_i} \right|_{\boldsymbol{u}=\boldsymbol{z}} = 0 \tag{D.3}$$

for every u_i in $\boldsymbol{u}$.

Now, let us use the result as the argument to the same operator. The operator is then to minimize the energy (with respect to $\boldsymbol{u}$)

$$E_{\boldsymbol{z}}(\boldsymbol{u}) = \sum_{i=1}^{n} (u_i - z_i)^2 + S(\boldsymbol{u}) \tag{D.4}$$

$$= E_{\boldsymbol{y}}(\boldsymbol{u}) - \sum_{i=1}^{n} (u_i - y_i)^2 + \sum_{i=1}^{n} (u_i - z_i)^2 \tag{D.5}$$

$$= E_{\boldsymbol{y}}(\boldsymbol{u}) + 2 \sum_{i=1}^{n} ui(y_i - z_i) + \sum_{i=1}^{n} (z_i^2 - y_i^2). \tag{D.6}$$

Again, at the minimum of this function, the following equation must be satisfied for i from 1 to n

$$\frac{\partial E_{\boldsymbol{z}}(\boldsymbol{u})}{\partial u_i} = \frac{\partial E_{\boldsymbol{y}}(\boldsymbol{u})}{\partial u_i} + 2(y_i - z_i) = 0. \tag{D.7}$$

From equation B.15, the only way that this equation can be satisfied at $\boldsymbol{u} = \boldsymbol{z}$ is if $y_i - z_i = 0$ for i from 1 to n which is not true in general. Since idempotence of the operator requires that $\boldsymbol{u} = \boldsymbol{z}$, these types of minimizations are therefore not idempotent. ⊡

Methods that fall in this category are the standard regularization methods and the weak continuity methods of Blake and Zisserman. The functional to minimize is similar to equation D.1 in the case of Leclerc's MDL with known spatially varying noise. However, the partial derivatives of $S(\boldsymbol{u})$ are not defined at the local minima of the functional, because of the kroneker delta functions. Therefore, the above proof does not apply for this case. In fact, we present a proof in section 3 that Leclerc's MDL piecewise constant operator is idempotent. Finally, note that using the continuation method in MDL problems makes the intermediate $S(\boldsymbol{u})$'s differentiable, which means that these intermediate operators are not idempotent.

2. Gibbs energy of an a priori model with a line process

In this section, we express the Gibbs energy of the a priori model used in example two of [**26**]. Although it is not specified in the paper, I assume they use a four-neighbour system for the pixel sites, because this is in accordance with the neighbour system for the line sites.

The line site between two pixel sites $(i1, j1)$ and $(i2, j2)$ is denoted by $l_{(i1,j1),(i2,j2)}$, and the value of a line site is either 0 or 1, signifying the absence or presence of a line (with fixed orientation).

$$U^P(\boldsymbol{f}, \boldsymbol{l}) = U(\boldsymbol{f}|\boldsymbol{l}) + U(\boldsymbol{l}) + \sum_{i,j} \left(\mu - \Phi\left(g_{i,j}, \phi\left(H(f_{i,j})\right)\right)\right)^2 / 2\sigma^2, \tag{D.8a}$$

where the energy of the underlying surface (dependent on the line process) and the energy of the line process (which together constitute the Gibbs energy of the a priori probability of $\boldsymbol{f}$) are given as

$$U(\boldsymbol{f}|\boldsymbol{l}) = 2\sum_{i,j}\Big\{\delta(l_{(i,j),(i-1,j)})\left[2\delta\left(f_{i,j} - f_{i-1,j}\right) - 1\right]$$

$$\delta(l_{(i,j),(i,j-1)})\left[2\delta\left(f_{i,j} - f_{i,j-1}\right) - 1\right]\Big\}, \tag{D.8b}$$

and

$$U(\boldsymbol{l}) = 0.9 \sum_{i,j} \Big\{ 3\delta(l_{(i,j)1} + l_{(i,j)2} + l_{(i,j)3} + l_{(i,j)4} - 4)$$

$$2\delta(l_{(i,j)1} + l_{(i,j)2} + l_{(i,j)3} + l_{(i,j)4} - 3)$$

$$3\delta(l_{(i,j)1} + l_{(i,j)2} + l_{(i,j)3} + l_{(i,j)4} - 1)$$

$$2\delta(l_{(i,j)1} + l_{(i,j)2} + l_{(i,j)3} + l_{(i,j)4} - 2)$$

$$\big(1 - \delta(l_{(i,j)1} + l_{(i,j)3}) - \delta(l_{(i,j)2} + l_{(i,j)4})\big)\Big\}, \tag{D.8c}$$

$$l_{(i,j)1} = l_{(i,j),(i-1,j)}, \tag{D.8d}$$

$$l_{(i,j)2} = l_{(i,j),(i,j-1)}, \tag{D.8e}$$

$$l_{(i,j)3} = l_{(i,j-1),(i-1,j-1)}, \tag{D.8f}$$

$$l_{(i,j)4} = l_{(i,j),(i-1,j-1)}, \tag{D.8g}$$

which is the sum of the potentials for all the four element cliques in the neighbourhood system of the line process.

3. MDL problems are idempotent

It is intuitively understandable that the standard regularization methods are not idempotent, because these methods smooth the data only by a finite amount: Subsequent applications of a method just keeps on smoothing the data further. However, Leclerc's MDL method smooths all the noise out of the data in one application.

Section 1 does not prove or disprove that Leclerc's method is idempotent, because Leclerc's energy functional is discontinuous at the global minimum. Let us consider the piecewise constant case with known variance white noise. By definition, the result of the minimum description consists of constant value patches (plus the description of the noise that has been removed). Therefore, if the operator that produced this data is applied again and none of the previously detected discontinuities disappear, then it is obvious that nothing will change within the regions either, since these can be fitted perfectly by the constant patch models. That means that no new discontinuities can be created within the continuous regions by a second application of the operator. The only possiblity for non-idempotency is that previously detected discontinuities are removed at subsequent applications of the operator. If we can show that previously detected discontinuities are not removed by subsequent applications of the operator, then we will have proved that the operator is idempotent. This is what we do below.

Given a vector $\boldsymbol{x}$ of initial data, Leclerc's method seeks the vector $\boldsymbol{y}$ that minimizes an energy of the form

(D.9) $$E_{\boldsymbol{x}}(\boldsymbol{y}) = \sum_{i \in I}(x_i - y_i)^2 + \lambda \sum_{i \in I} \sum_{j \in N_i} (1 - \delta(y_i - y_j))$$

If $\boldsymbol{y}$ is the global minimum of this functional, and if $\boldsymbol{y}'$ is a vector identical to $\boldsymbol{y}$ except for m less discontinuities, then

(D.10) $$E_{\boldsymbol{x}}(\boldsymbol{y}') = \sum_{i \in I}(x_i - y_i')^2 + \lambda \sum_{i \in I} \sum_{j \in N_i} (1 - \delta(y_i - y_j)) - 2m\lambda$$

(the 2 in $2m\lambda$ is due to each discontinuity being counted twice) and

(D.11) $$E_{\boldsymbol{x}}(\boldsymbol{y}) < E_{\boldsymbol{x}}(\boldsymbol{y}') \Rightarrow \sum_{i \in I}(x_i - y_i')^2 - 2m\lambda > \sum_{i \in I}(x_i - y_i)^2$$

Since we assumed that $\boldsymbol{y}$ is the global minimum of the energy function. To verify if the operator is idempotent, we perform another minimization with $\boldsymbol{y}$ as initial data. In that case,

(D.12) $$E_{\boldsymbol{y}}(\boldsymbol{y}) = \lambda \sum_{i \in I} \sum_{j \in N_i} (1 - \delta(y_i - y_j))$$

(D.13) $$E_{\boldsymbol{y}}(\boldsymbol{y}') = \sum_{i \in I}(y_i - y_i')^2 + \lambda \sum_{i \in I} \sum_{j \in N_i} (1 - \delta(y_i - y_j)) - 2m\lambda$$

One of the conditions for the global minimum to be $\boldsymbol{y}'$ is that

(D.14) $$E_{\boldsymbol{y}}(\boldsymbol{y}') < E\boldsymbol{y}(\boldsymbol{y})$$

or, from equations D.10 and D.11

(D.15) $$\sum_{i \in I}(y_i - y_i')^2 - 2m\lambda < 0$$

Assuming that the vectors $(\boldsymbol{x} - \boldsymbol{y})$ and $(\boldsymbol{y} - \boldsymbol{y}')$ are perpendicular (this will be shown later) we can write Phytagorean theorem:

(D.16) $$\sum_{i \in I}(x_i - y_i')^2 = \sum_{i \in I}(x_i - y_i)^2 + \sum_{i \in I}(y_i - y_i')^2.$$

Subtracting $2m\lambda$ on either side of this equation, we obtain

(D.17) $$\sum_{i \in I}(x_i - y_i')^2 - 2m\lambda = \sum_{i \in I}(x_i - y_i)^2 + \left[\sum_{i \in I}(y_i - y_i')^2 - 2m\lambda\right].$$

We know from equation D.15 that the term in square brackets is negative. If we remove this term, we obtain the relation

$$\sum_{i\in I}(x_i - y_i')^2 - 2m\lambda < \sum_{i\in I}(x_i - y_i)^2 \tag{D.18}$$

but this equation contradicts equation D.11, which shows that any $\boldsymbol{y}'$ with less discontinuities than $\boldsymbol{y}$ is not the global minimum of a second application of the operator, which proves that the operator is idempotent.

This proof is valid only under the assumption that $(\boldsymbol{x}-\boldsymbol{y})$ and $(\boldsymbol{y}-\boldsymbol{y}')$ are perpendicular. This assumption will be shown to hold in general here.

In the piecewise constant case, the result $\boldsymbol{y}$ of MDL consists of a partition of the image, with each partition a constant value patch which is the average of the pixel values in the original image. If some of the partitions are merged together in a second application resulting in $\boldsymbol{y}'$, the merged partition will then be further averaged.

Assume that the elements of a vector $\boldsymbol{x}$ of length k are segmented in m groups of n_i elements per group, $i = 1\ldots m$:

$$\begin{aligned}\boldsymbol{x} &= (\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_m}) \\ &= (\{x_{1,1}, x_{1,2}, \ldots, x_{1,n_1}\}, \{x_{2,1}, x_{2,2}, \ldots, x_{2,n_2}\}, \ldots, \{x_{m,1}, x_{m,2}, \ldots, x_{m,n_m}\}) .\end{aligned} \tag{D.19}$$

If we perform an averaging in each of the m groups, we obtain the vector

$$\boldsymbol{y} = \left(\left\{\frac{\boldsymbol{x}_1\cdot\boldsymbol{i}}{n_1}, \ldots, \frac{\boldsymbol{x}_1\cdot\boldsymbol{i}}{n_1}\right\}, \left\{\frac{\boldsymbol{x}_2\cdot\boldsymbol{i}}{n_2}, \ldots, \frac{\boldsymbol{x}_2\cdot\boldsymbol{i}}{n_2}\right\}, \ldots, \left\{\frac{\boldsymbol{x}_m\cdot\boldsymbol{i}}{n_m}, \ldots, \frac{\boldsymbol{x}_m\cdot\boldsymbol{i}}{n_m}\right\}\right) \tag{D.20}$$

where $\boldsymbol{i}$ is the identity vector. Finally if we average all the groups together, we obtain

$$\boldsymbol{y}' = \frac{(\boldsymbol{y}\cdot\boldsymbol{i})}{k}. \tag{D.21}$$

The path taken in vector space during these two averaging steps is given by

$$\begin{aligned}\boldsymbol{x}-\boldsymbol{y} = \Bigg(&\left\{x_{1,1} - \frac{\boldsymbol{x}_1\cdot\boldsymbol{i}}{n_1}, x_{1,2} - \frac{\boldsymbol{x}_1\cdot\boldsymbol{i}}{n_1}, \ldots, x_{1,n_1} - \frac{\boldsymbol{x}_1\cdot\boldsymbol{i}}{n_1},\right\}, \\ &\left\{x_{2,1} - \frac{\boldsymbol{x}_2\cdot\boldsymbol{i}}{n_2}, x_{2,2} - \frac{\boldsymbol{x}_2\cdot\boldsymbol{i}}{n_2}, \ldots, x_{2,n_2} - \frac{\boldsymbol{x}_2\cdot\boldsymbol{i}}{n_2},\right\}, \\ &\vdots \\ &\left\{x_{m,1} - \frac{\boldsymbol{x}_m\cdot\boldsymbol{i}}{n_m}, x_{m,2} - \frac{\boldsymbol{x}_m\cdot\boldsymbol{i}}{n_m}, \ldots, x_{m,n_m} - \frac{\boldsymbol{x}_m\cdot\boldsymbol{i}}{n_m},\right\}, \Bigg)\end{aligned} \tag{D.22}$$

and

$$\begin{aligned}\boldsymbol{y}-\boldsymbol{y}' = \Bigg(&\left\{\frac{\boldsymbol{x}_1\cdot\boldsymbol{i}}{n_1}-\frac{\boldsymbol{y}\cdot\boldsymbol{i}}{k}(\times n_1 \text{ times})\right\},\\ &\left\{\frac{\boldsymbol{x}_2\cdot\boldsymbol{i}}{n_2}-\frac{\boldsymbol{y}\cdot\boldsymbol{i}}{k}(\times n_2 \text{ times})\right\},\\ &\vdots\\ &\left\{\frac{\boldsymbol{x}_m\cdot\boldsymbol{i}}{n_m}-\frac{\boldsymbol{y}\cdot\boldsymbol{i}}{k}(\times n_m \text{ times})\right\}\Bigg).\end{aligned} \tag{D.23}$$

These two vectors are perpendicular, because their dot product, given below, is zero:

$$\begin{aligned}(\boldsymbol{x}-\boldsymbol{y})\cdot(\boldsymbol{y}-\boldsymbol{y}') = &\left(\frac{\boldsymbol{x}_1\cdot\boldsymbol{i}}{n_1}-\frac{(\boldsymbol{y}\cdot\boldsymbol{i})}{k}\right)(\boldsymbol{x_1}\cdot\boldsymbol{i})-\left(\frac{\boldsymbol{x}_1\cdot\boldsymbol{i}}{n_1}-\frac{(\boldsymbol{y}\cdot\boldsymbol{i})}{k}\right)(\boldsymbol{x_1}\cdot\boldsymbol{i})+\\ &\left(\frac{\boldsymbol{x}_2\cdot\boldsymbol{i}}{n_2}-\frac{(\boldsymbol{y}\cdot\boldsymbol{i})}{k}\right)(\boldsymbol{x_2}\cdot\boldsymbol{i})-\left(\frac{\boldsymbol{x}_2\cdot\boldsymbol{i}}{n_2}-\frac{(\boldsymbol{y}\cdot\boldsymbol{i})}{k}\right)(\boldsymbol{x_2}\cdot\boldsymbol{i})+\\ &\vdots\\ &\left(\frac{\boldsymbol{x}_m\cdot\boldsymbol{i}}{n_m}-\frac{(\boldsymbol{y}\cdot\boldsymbol{i})}{k}\right)(\boldsymbol{x_m}\cdot\boldsymbol{i})-\left(\frac{\boldsymbol{x}_m\cdot\boldsymbol{i}}{n_m}-\frac{(\boldsymbol{y}\cdot\boldsymbol{i})}{k}\right)(\boldsymbol{x_m}\cdot\boldsymbol{i})\\ =&\ 0.\end{aligned} \tag{D.24}$$

4. Constraint subset for the piecewise constant case of Leclerc's method in a two point configuration

In this section, we give an illustration of how such a local minimization can be used through a trivial example with an image composed of only two points, under the piecewise constant description. The functional to minimize over the two variables is then

$$\begin{aligned}E(u_1,u_2) = &\frac{a}{\sigma^2}(z_1-u_1)^2+\frac{a}{\sigma^2}(z_2-u_2)^2\\ &+b[1-\delta(u_1-u_2)].\end{aligned} \tag{D.25a}$$

Equivalently the energy can be reformulated as

$$\begin{aligned}E(h) = &\frac{a}{\sigma^2}(z_0+h/2-u_1)^2+\frac{a}{\sigma^2}(z_0-h/2-u_2)^2\\ &+b[1-\delta(u_1-u_2)].\end{aligned} \tag{D.25b}$$

There are obviously two possible solutions to this minimization. Either $u_1 = u_2 = (z_1+z_2)/2$, or $u_1 = z_1$ and $u_2 = z_2$. By comparing the energies of these two cases, we find that the second case will occur if $h >$

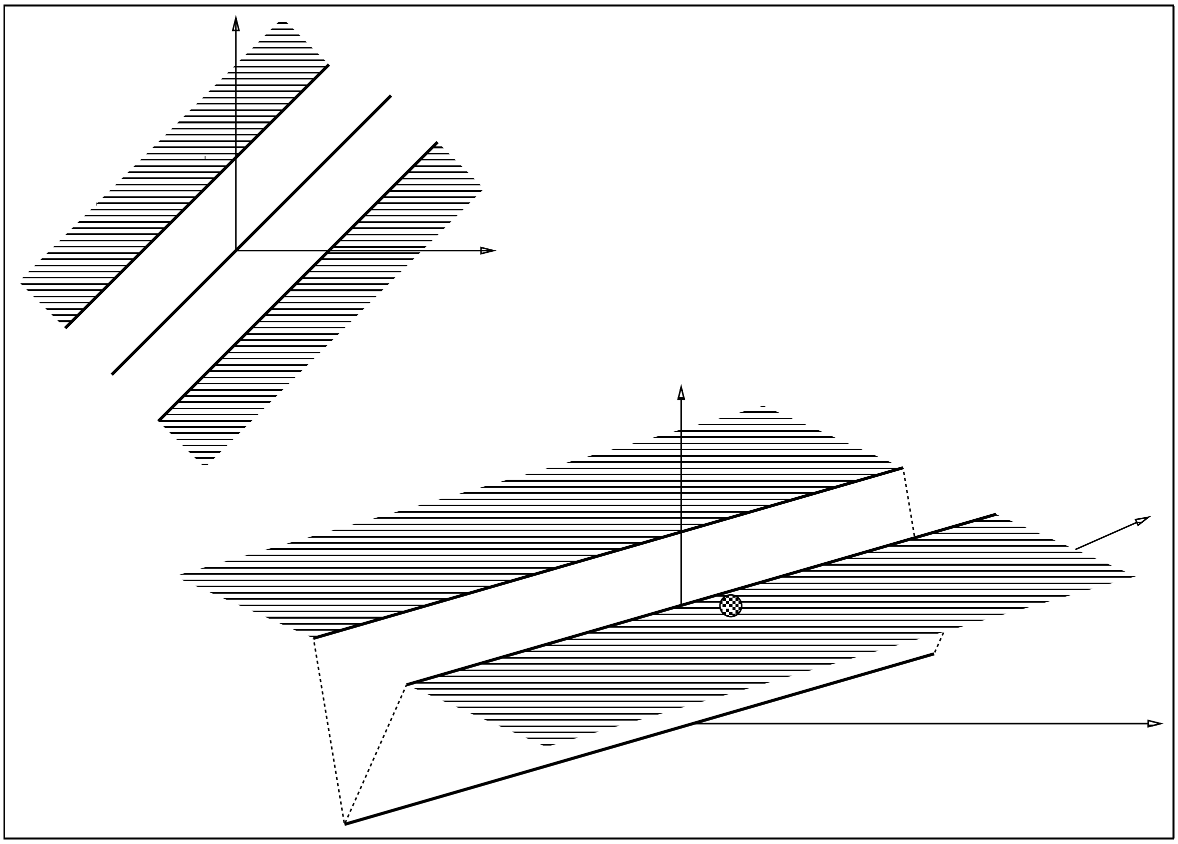


FIGURE D.1. Leclerc's MDL method seen as a local minimization problem. The left figure shows the constraint subset of the valid combinations of u_1 and u_2, including the line passing through the origin. The right shows one possible energy surface used for a local minimization statement of the global minimization problem.

$h_\tau \sqrt{2b\sigma^2/a}$. We can easily express this problem as a local energy minimization, as illustrated in figure D.1. The energy function in the figure is simply

$$\min(|u_1 - u_2|, \sqrt{\frac{2b\sigma^2}{a}}). \tag{D.26}$$

The absolute value could as well have been replaced by $(u_1 - u_2)^2$.

By adding more pixels to the image, the dimensionality of the energy functional would increase, and the topology of the constraint subset would be much more complex. However the energy functional of the equivalent local minimization problem has same dimensionality as the global one, and the topology of the global energy is also very complex. Also, the local energy functional is likely to be much more symmetric, since it includes only the constraints, not the data. In section 5 the constraint subset and a local energy function are found for a three point case.

5. Constraint subset for the piecewise constant case of Leclerc's method in a three point configuration

If the "image" consists of three points arranged in a triangle (i.e. all points are neighbours), then only three categories of solution may occur. Either the three points are averaged, only two are averaged and the other one is left unchanged, or all three points are left unchanged. The description length of such an image is given by

$$\begin{aligned} E = {} & \frac{a}{\sigma^2}(z_1-u_1)^2 + \frac{a}{\sigma^2}(z_2-u_2)^2 + \frac{a}{\sigma^2}(z_3-u_3)^2 \\ & +b[1-\delta(u_1-u_2)] + b[1-\delta(u_2-u_3)] + b[1-\delta(u_3-u_1)]. \end{aligned} \tag{D.27}$$

The first case is for three points separated by an equal height h_1

$$z_1 = z_0 + h_1, \tag{D.28}$$

$$z_2 = z_0, \tag{D.29}$$

$$z_3 = z_0 - h_1. \tag{D.30}$$

The value of h_1 for which the description of the tree distinct points and of a single region is equal is $h_1 = \sqrt{(3b\sigma^2)/(2a)}$.

The second case is when one point as a value very different from the two others

$$z_1 = z_0 + h_2/2, \tag{D.31}$$

$$z_2 = z_0 - h_2/2, \tag{D.32}$$

$$z_3 = z_0 + \infty. \tag{D.33}$$

The threshold h_2 for merging z_1 and z_2 is given by $h_2 = \sqrt{(2b\sigma^2)/a}$.

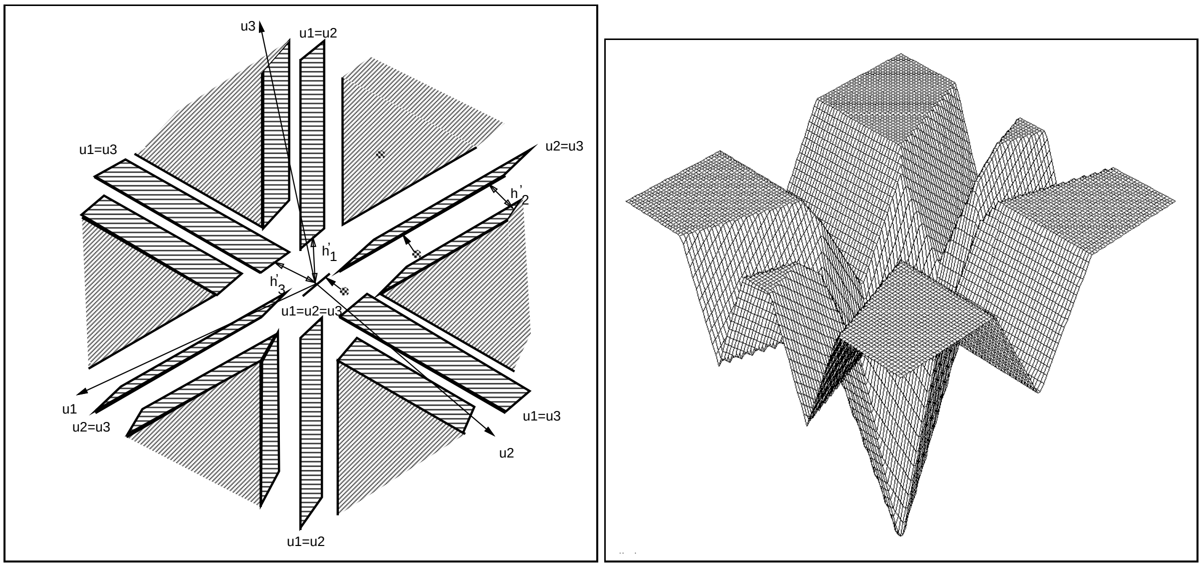
Finally, if two points already have same value, but the third one does not,

$$z_1 = z_2 = z_0 + h_3/3, \tag{D.34}$$

$$z_3 = z_0 - 2h_3/3. \tag{D.35}$$

The threshold for merging the third point is $h_3 = \sqrt{(3b\sigma^2)/(2a)}$.

Figure D.2 shows the constraint subset for this case. It consists of a line corresponding to the case $u_1 = u_2 = u_3$, 3 pairs of (coplanar) half planes corresponding to the cases where only two points have same value, and subsets of 3-space corresponding to the cases where the points keep their initial values. The distances h'_i correspond to the three thresholds computed above (the primes are because the distances as shown are at an angle, and are therefore not equal to the actual thresholds). The right figure is a slice of the local energy



(a) (b)

FIGURE D.2. Constraint subset and local energy for a three point case. The left figure is the constraint subset, consisting of a line, planes, and regions in three space. The right figure is a slice out of a possible four dimensional local energy functional.

function given by

$$
\begin{aligned}
E = \min(&|u_1 - u_2| + |u_1 - u_3| + |u_2 - u_3|,\\
&k_2|u_1 - u_2| + k_1, k_2|u_1 - u_3| + k_1, k_2|u_2 - u_3|) + k_1,\\
&k_3), \qquad \text{(D.36)}
\end{aligned}
$$

$$k_1 = 2\sqrt{(3b\sigma^2)/a}, \qquad \text{(D.37)}$$

$$k_2 = (2 - \sqrt{2})\sqrt{3}, \qquad \text{(D.38)}$$

$$k_3 = 4\sqrt{(3b\sigma^2)/(2a)} \qquad \text{(D.39)}$$

taken on a plane perpendicular to the line $u_1 = u_2 = u_3$. Note that the vector from the initial value to the local minima will always be parallel to that plane anyway, because the average intensity of the image is preserved everywhere within the plane.

Document Log:

Manuscript Version 0

Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeX — 7 February 1997

Jean W. Lagarde

Center for Intelligent Machines, McGill University, 3480 University St., Montréal (Québec) H3A 2A7, Canada, *Tel.* : (514) 398-7158

Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-LaTeX