

# View Correspondence using Curvature and Motion Consistency

Gilbert Soucy

B. Eng., (Université de Sherbrooke), 1987

Department of Electrical Engineering

McGill University

Montréal

September, 1993

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

© Gilbert Soucy, 1993

## Abstract

This thesis deals with the problem of finding the correspondence of overlapping views. Correspondence is determined locally by finding the set of translations and rotations that maps a local neighborhood in one view to its corresponding neighborhood in another.

The solution is based on the concept of minimizing both the spatial and temporal variation of curvature between surfaces in adjacent views. Two specialized filters are used for this purpose: curvature and motion consistency. The first, curvature consistency, is applied individually on both views to recover a stable description of the local structure before correspondence. The second one, motion consistency, is used to control the rigidity of the relative motion of the object with respect to the observer.

This notion of the control of the rigidity of the motion is very important and constitutes one of the main contributions of this thesis. The implementation and the analysis of the limitations of the method represents also an important part of this work.

## Resumé

Cette thèse considère le problème de la mise en correspondance de différents points de vue d'un objet lisse quelconque. L'approche proposée évite une recherche globale de la solution en minimisant localement la différence structurelle des deux vues.

La méthode repose entièrement sur deux filtres spécialisés soient la "compatibilité des courbures" et la "compatibilité des déplacements". Le premier de ces filtres, la "compatibilité des courbures", est appliqué sur les deux vues séparément afin d'obtenir une description stable de la structure locale avant la comparaison comme telle. Le deuxième filtre, la "compatibilité de déplacement", permet de contrôler la rigidité du déplacement relatif de l'objet par rapport à l'observateur entre les vues.

Cette notion de contrôle de rigidité du déplacement occupe une place très importante dans cette thèse et constitue en fait une de ses principales contributions. L'implantation et l'analyse des limitations de la méthode représente aussi une part importante du travail présenté.

## Remerciements

D'abord et avant tout, je tiens à remercier mon superviseur, Frank Ferrie, pour son dynamisme débordant et son support moral, financier et technique. Sa "vision" globale des choses m'a souvent aidé à adapter des objectifs plus réalistes et à retomber sur terre.

Je tiens aussi à remercier tout particulièrement Peter Whaite pour sa grande patience, sa disponibilité et son inestimable assistance technique et organisationnelle. Mon seul vœux est de pouvoir un jour lui rendre la pareille...

Il me faut aussi souligner les encouragements mutuels de tout notre dynamique groupe "les Rangers" (particulièrement les vieux de la vieille: Pierre, Frank, Jean et la nouvelle génération: Duncan et André) et aussi de quelques "vétérans" (Mathieu, Jean Drolet, Marco, Nick, Christian). Merci particulier à Loc pour de nombreuses discussions fructueuses. Merci en général à tout le personnel et aux étudiants de McRCIM.

Je ne saurais passer sous silence le support moral de quelques "vieux" amis (entre autres Claude, Bruno et Sandra) et de toute ma famille.

Finalement, je remercie le CRSNG pour l'obtention d'une bourse d'étude supérieure.

## Table of Contents

<b>Chapter 1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation	1
1.2	Overview	2
1.3	Contributions	3
1.4	Organization of the thesis	4
<b>Chapter 2</b>	<b>The Problem of View Correspondence</b>	<b>5</b>
2.1	Introduction	5
2.2	Set-up Based Techniques	5
2.3	Discrete Feature Matching Techniques	6
2.4	Matching Techniques based on Continuous Characteristics	8
2.5	Summary	10
<b>Chapter 3</b>	<b>The Curvature Consistency Algorithm</b>	<b>11</b>
3.1	Introduction	11
3.2	Goals	12
3.3	Estimating the Differential Properties	13
3.4	Refining the Estimated Differential Properties	13
3.5	Discussion	18
3.6	Summary	21
<b>Chapter 4</b>	<b>Putting Views in Correspondence</b>	<b>22</b>
4.1	Introduction	22
4.2	Problem Definition	22
4.3	Refining Motion Estimates	23
4.3.1	Application of Curvature Consistency on Both Views Separately	27
4.3.2	Subdivision of the First View in Small Windows	28

4.3.3	Classification of the Patches . . . . .	28
4.3.4	Defining a Metric of Similarity . . . . .	31
4.3.5	Existence and Uniqueness of the Solution . . . . .	34
4.3.6	Minimizing the Metric $D$ . . . . .	37
4.3.7	Propagating the Motion Parameters . . . . .	39
4.4	Motion consistency . . . . .	41
4.4.1	Updating the Position . . . . .	43
4.4.2	Updating the Orientation . . . . .	46
4.4.3	Determining the Updated Motion Parameters . . . . .	51
4.5	Controlling the Rigidity of the Motion . . . . .	51
4.6	Relation Between this Method and Continuation Methods . . . . .	52
4.7	Summary . . . . .	55
<b>Chapter 5</b>	<b>Results and experimentation . . . . .</b>	<b>56</b>
5.1	Introduction . . . . .	56
5.2	Data Acquisition . . . . .	56
5.2.1	Range Finder . . . . .	57
5.2.2	Precision Stages . . . . .	57
5.2.3	Scanner Moved by a Robot . . . . .	60
5.2.4	Virtual Set-Up for Synthetic Data . . . . .	61
5.3	Results with Synthetic Data . . . . .	61
5.3.1	Noiseless Paraboloid . . . . .	62
5.3.2	Noisy Paraboloid . . . . .	68
5.3.3	Noiseless Hyperboloid . . . . .	72
5.3.4	Paraboloid with a Periodic Structure . . . . .	73
5.4	Results with Real Data . . . . .	76
5.4.1	A Small Owl Statuette . . . . .	76
5.4.2	A Scale Model of a Mercedes-300 . . . . .	80
5.5	Summary . . . . .	80
<b>Chapter 6</b>	<b>Conclusion . . . . .</b>	<b>83</b>

<b>Appendix A Determination of the Updated Frame in Curvature</b>	
Consistency . . . . .	85
<b>Appendix B Summary of Quaternion Algebra . . . . .</b>	<b>89</b>
<b>Appendix C Interpreting rotations . . . . .</b>	<b>92</b>
<b>Bibliography . . . . .</b>	<b>94</b>

## List of Figures

3.1	Local surface representation: the augmented Darboux Frame . . . . .	12
3.2	Osculating paraboloid at point $P$ . . . . .	13
3.3	Principal direction for a stone owl before and after curvature consistency . .	15
3.4	Different type of transport model . . . . .	16
3.5	Mechanism of transport process in curvature consistency . . . . .	17
3.6	Comparison of principal direction field found with curvature consistency and with convolution with a Gaussian . . . . .	19
3.7	Equivalent amount of smoothing by curvature consistency and convolution with a Gaussian . . . . .	20
4.1	Multiple solutions to matching a single Darboux frame on a paraboloid . . .	24
4.2	Multiple solutions to matching a neighborhood on a paraboloid . . . . .	25
4.3	Color-coding of the constant Gaussian curvature sign regions of a small owl .	29
4.4	Repetition of a local structure in a convex figure . . . . .	30
4.5	Typical organization of a neighborhood in two views. . . . .	32
4.6	Convexity of the metric $D$ for two views of a synthetic paraboloid . . . . .	36
4.7	Effect of curvature consistency on the convexity of $D$ . . . . .	37
4.8	Bilinear interpolation of a range image . . . . .	38
4.9	Effect of a bilinear interpolation on the metric $D$ . . . . .	39
4.10	Principle of propagation of the motion parameters . . . . .	40
4.11	Effect of motion consistency. . . . .	42
4.12	Principle of motion consistency for displacement . . . . .	44
4.13	Minimum sum of <i>distance</i> between two frames . . . . .	48
4.14	2D slices of the metric $D$ (displacement parameters) . . . . .	53
4.15	2D slices of the metric $D$ (rotation parameters) . . . . .	54
5.1	Schematic of the range finder used . . . . .	58
5.2	Precision Stages . . . . .	59

5.3	Robot (Puma 560) moving the range finder around a scene . . . . .	60
5.4	Noiseless paraboloid . . . . .	63
5.5	Correspondence of two views of a perfect paraboloid . . . . .	64
5.6	Starting point of the view correspondence process (noiseless paraboloid) . . . . .	65
5.7	Correspondence of views found for a noiseless paraboloid. . . . .	66
5.8	Noisy paraboloid . . . . .	70
5.9	Registration of two noisy views of a paraboloid . . . . .	71
5.10	Noiseless paraboloid . . . . .	72
5.11	Registration of two views of a synthetic hyperboloid . . . . .	73
5.12	Registration of two views of a paraboloid with a repetitive structure . . . . .	75
5.13	Registration of two views of a small owl statuette . . . . .	77
5.14	Different stages of motion consistency . . . . .	79
5.15	Starting point of the view correspondence process (Mercedes-300) . . . . .	80
5.16	Correspondence of views found for the Mercedes-300 . . . . .	81

This thesis proposes a novel approach to the problem of finding the correspondence of a sequence of overlapping views of an arbitrary smooth object. The data used for this work has been acquired with a laser range finder system.

The goal is to recover the motion parameters (translation and rotation) that locally map corresponding points in adjacent views. This is necessary to accommodate any non-rigid motion of the surface under view [8].

Surfaces are assumed to be piecewise smooth. This allows for the recovery of differential properties that are used in the process of determining correspondence. We accomplish this by using a reconstruction procedure based on the minimization of a functional form that embeds an implicit model of surface curvature.

One of the advantages of this approach is that the problem of correspondence can be cast as a convex minimization problem. Surface descriptions in adjacent views can be put in correspondence by minimizing another functional form that describes their similarity as a function of the motion parameters. This approach can work for non-rigid motion provided that the curvature structure varies smoothly. An additional filtering process called *motion consistency* is used to smooth out the variations in local motion estimates, reflecting the physical constraint that local descriptions are coupled through the surface in proportion to rigidity.

We show that this approach works for synthetic and real data.

### 1.1 Motivation

There are a number of motivations for this work.

On the biological side, even though the human representation of the 3D world is far from being clearly understood, it seems that our representation contains more than a single static view. In fact, while thinking of a known object, we can rotate it, look at it at different scales and with different levels of detail. There are many

different attributes associated with a representation such as color, texture, brightness, roughness and non-visual information such as weight, viscosity, odor. In order to survive in the environment we had to learn how to integrate all of this information.

Our representation of the world is dynamic, task-dependent and very flexible. Therefore, it is argued that the first step towards this representation is an integration of the different sources of information. Such a front-end process is required by any modeling system that does not have a priori knowledge of its environment. This thesis is not intended to explain biological vision, but rather proposes a mechanism for integrating data in artificial vision systems.

On a more practical side, since artificial vision systems are often used in association with robotic systems, the construction of a 3D model of the world is needed for obstacle avoidance and path planning. Autonomous robots are a good example of systems requiring integration of views of their environment since it is continuously changing.

### 1.2 Overview

The problem to be solved can be posed as finding the set of motion parameters that preserves the local structure of a surface across adjacent views. The proposed solution to this problem is a minimization of the variation of the differential properties of the surface across views. This minimization is based on the assumption of a locally smooth (piecewise) curvature field and is done without invoking a global rigidity assumption.

In order to get a complete mapping between two overlapping views, one could attempt to determine correspondence between successive pairs of images in pointwise fashion. However, the method is limited by computational complexity and the fact that not all points on a surface are good candidates for correspondence. Instead, our strategy is to select a subset of points satisfying a set of conditions related to the stability of their features. The solutions of these points are then used to initialize the solutions of neighboring points where the surface is visible, or to interpolate solutions

where the surface is occluded.

Because motion parameters are estimated locally, an additional stage of filtering is required. A useful constraint in this regard is provided by the physical make-up of the surface. Motion between adjacent surface patches is coupled in proportion to the rigidity of the surface. Thus, motion parameters are constrained to vary smoothly which in turn induces a smooth deformation of the surface.

The applicability of the method rests on two basic assumptions:

1. within each view, the curvature field is piecewise smooth;
2. deformations vary slowly with respect to motion.

The first assumption will allow for the local estimation of the motion parameters using a gradient descent method based on a comparison of the differential properties of the surface in adjacent views. This assumption is enforced through the “curvature consistency” filter that is described later in Chapter 3.

The second assumption expresses the fact that an object should not break as it moves. This assumption is enforced through the “motion consistency” filter that will be described in detail in Chapter 4.

To the previous assumptions, it is necessary to add a third one: the availability of an estimate of the motion parameters. The quality of this estimate is related to the distinctiveness and robustness of the features computed. Such estimates can be obtained from a manipulator system moving a camera or from an inertial navigation system on a mobile platform. A by-product of this process is an updated and refined set of motion parameters that can be used by an autonomous robot to further refine its knowledge of its own position.

### 1.3 Contributions

The overall contribution of this thesis is a novel approach to view correspondence of arbitrary smooth objects without the assumption of global rigid motion. More specifically, the contributions are:

1. a local formulation of the problem of view correspondence;
2. an implementation of its solution;
3. the design of a specialized filter to locally smooth out the motion parameters of neighboring points (*motion consistency* algorithm);
4. an analysis of the limitations of the method;
5. an analogy with standard continuation methods;
6. a set of experiments with synthetic and real data showing that the method gives accurate results.

## 1.4 Organization of the thesis

The next chapter presents a brief overview of the common methods used to find the correspondence of views. These methods are grouped in three main categories: set-up based, feature matching and structure comparison techniques. Each of these categories is explained and presented with few examples.

Chapter 3 can be seen as necessary background material for this thesis. It describes the *curvature consistency* algorithm which is used to enforce the essential condition of smoothness of the differential properties of the surface.

Chapter 4 is the heart of this thesis. It includes the complete and detailed description of the proposed solution to the view correspondence problem. It also describes the other specialized filter, “motion consistency”.

Chapter 5 includes many different examples of application of the method on both synthetic and real range images. These examples lead to an analysis of the behavior of the method under different circumstances.

Finally, the conclusion summarizes the main contributions of this thesis and outlines what can be done to improve the performance and the robustness of the method.

## 2.1 Introduction

In this chapter, I present a brief overview of the different types of methods commonly used to find the correspondence of views.

These different methods can be classified in three main categories: set-up based, matching of discrete features and matching of continuous characteristics.

Methods in the first category use a precisely calibrated set-up to obtain the correspondence of views. The second category of techniques consists of methods where specific features of the scene (such as lines or corners) are matched across views. Finally, methods in the third category try to compare some continuous characteristic of each view.

In the following sections, we consider these methods in more detail.

## 2.2 Set-up Based Techniques

To minimize the complexity of view correspondence, one can avoid the problem all together through precise registration of the different view points. This can be accomplished through a precisely calibrated mechanical set-up.

However, it is often difficult to obtain precise calibration, e.g. one obtains an estimate as opposed to an exact measurement. For a mobile robot for instance, the position is known up to a certain error but usually not precisely. In fact, one often needs to determine correspondence in order to refine the estimate of this position.

In [1] for instance, Bhanu uses a rotary stage to get multiple views of an object. The correspondence of the views is given by the orientation of the stage and the calibration of the set-up with respect to the sensor. A simple distance threshold is used to discard points scanned multiple times (visible from multiple viewpoints). The set of 3D points representing the object is then converted into a set of polygons for

eventual shape matching with a model in a database. The type of set-up used for data acquisition works better for convex objects.

Another example of this technique is given in [5] by Elfes and Matthies. In this case, a mobile robot tries to build an occupancy map of its environment by matching different views. Sonar and stereo range data are combined into a single representation using the robot position and orientation as a basis for view matching. A Bayesian model allows the fusing of information from the different sensors. The robot is assumed to know its absolute position so that the different views can be merged easily. However, if this is not the case, an additive error in the registration of the views can be produced. By using the visual feedback produced by an *active* matching of views (such as the one proposed in this thesis), one can avoid this type of additive error position (assuming that visual information is more precise than the robot encoders).

In fact, this type of technique will work only with a carefully calibrated set-up which is difficult to accomplish in the case of mobile robot.

In the next sections, we examine several methods that *actively* register multiple views of an object.

### 2.3 Discrete Feature Matching Techniques

Discrete feature matching techniques are applied in many different circumstances. For example, in stereo, one attempts to match discrete features from grey level images in order to obtain depth from triangulation. In the case of range images, matching features allows one to obtain the relative motion between adjacent views.

The features to be matched can take different forms. For example, discontinuities in color or grey level intensity can be used as features. In range images, one can use lines, planes or depth discontinuities.

What is common to all of these features is that they usually correspond to extremal values of a specific characteristic of the data. For this category of technique, it is important that a feature be at least locally unique and easy to detect.

Shah and Jain give an example of this type of technique in [17] where the idea is

to obtain structure from motion by matching time-varying corners between frames. A special operator, based on the variation of the grey-level of images, is used to detect the *cornerness* of each pixel. This characteristic is then compared across views to determine the time-varying corners. The method appears to be robust to noise. However, there still remains the problem of matching the features across frames, which is feasible if short displacements are assumed and more than two frames are used.

In [15], Bergevin, Laurendeau and Poussart estimate the motion between two range views using a hierarchical surface triangulation. It is assumed that the largest triangles on both views will represent more or less the same regions. A set of possible transformations mapping the two views results from the matching of a single pair. The amount of overlap existing between the views is used as a simple measure of the quality of fit of a given transformation and permits the isolation of the correct one. This transformation is usually coarse and needs some refinement. This is done with one of the three following techniques: hierarchical tracking of the best matching triangles, a method based on the heuristic search of Potmesil [14] and an adaptation of the iterative least-square computation of Chen and Medioni [3].

In the case of range images, the problem can be made easier. In [21], Vemuri and Aggarwal use a sensor combining range and intensity data to merge multiple views of an object. The object is placed on a base plane on which a specific pattern (a line) is drawn. This pattern must be at least partially visible from all view points. The pixel position of the line can be determined with the intensity part of the data while its 3D position can be determined with the range part of the data. The process of matching can be made automatic since all that is needed is the detection of a line in an intensity image. An advantage of this method is that no set-up calibration is needed (only the range finder is used to determine the position of the objects). The problem with this method is that it is not possible to get all the views of the object (the part touching the base plane will never be visible).

Because it operates on discrete and isolated features, this type of technique often involves exhaustive search (or at least a search in an area determined by the estimate

of the motion). This usually makes it computationally more expensive than the third type of technique which is described in the following section.

## 2.4 Matching Techniques based on Continuous Characteristics

Methods that perform a continuous comparison of functions across views have an important advantage over discrete methods: they use a metric of comparison that varies smoothly in the neighborhood of the solution. If a metric can be found for comparison that is both smooth and convex then correspondence can be determined using standard minimization techniques such as gradient descent. This can result in an efficient search.

To facilitate the design of such a metric, it is often useful to convert the input (sensor) data into a stable intermediate form. This process is referred to as visual reconstruction [2]. It consists of algorithms by which a piecewise-smooth description of the input signal is computed from a discrete set of sensor measurements. This step often involves processes such as the detection of discontinuities, Gaussian smoothing, least-squares fit with parametric functions, etc. In these techniques there is always a trade-off between preserving the original data and smoothing out the noise. Many different techniques are available and a good discussion of this topic is found in [2]. The method of reconstruction chosen should preserve the structural element to be compared across views (depth, curvature, normal...).

In order to avoid local minima while performing minimization, it is necessary to have a first order estimate of the solution. Starting too far from it might well cause the method to fall into a local minimum not corresponding to the correct solution.

An example of this type of method is given in [10] for stereo grey level images. Lucas and Kanade interpret the intensity images as functions  $I = f(x, y)$ . By smoothing the images (reconstruction) before the cross-comparison and by assuming that the images are in approximate registration (estimation of motion), the stereo matching of the intensity functions is performed using a Newton-Raphson method. They claim

that the technique is able to find the best match between two images with far fewer comparisons of images than techniques that examine the possible positions of registration in some fixed order. This method fails however for abrupt changes in disparity as is often the case for this type of technique at discontinuities.

In [14], Potmesil presents a method to obtain a 3D solid model by matching 3D surface segments. In this case again, it is assumed that an estimate of the motion is given. The surfaces are compared across views at a few evaluation points using position difference (distance in the surface normal direction), orientation (surface normal) and curvature (magnitude of surface curvature). A heuristic search algorithm with an evaluation function controls the generation of new transformations. The paper makes no mention of application of a reconstruction method to the raw data.

Another example of this kind of technique is given in [3] by Chen and Medioni. In this case, only the distance in the surface normal direction between the views is used to guide an iterative process. It is again assumed that a good estimation of the motion is given so that process can converge in a few iterations. The minimization of the difference of structure is done for few points selected on smooth areas (as determined by the local fit of planes). Here again, no specific algorithm for surface reconstruction is mentioned.

Szeliski presents in [20] a method to estimate the motion between views from sparse range data (terrain maps). The idea is to interpolate the surface between the sample points using 2D splines in order to get a dense description. A new set of data (new view) is then put in registration with the current model of the terrain by finding the geometric transformation that makes it most likely (in the Bayesian sense) that the points came from the same surface. The representation of the surface is refined as more views are obtained and the uncertainty of the estimate of motion is obtained from the shape of the energy function in the vicinity of the solution. Assumptions such as the availability of a good estimate of motion and the smoothness of the surface allow the use of a gradient descent technique to solve for the optimal motion between two views.

## 2.5 Summary

In this chapter, I have described some of the general approaches used to find the correspondence of views. The most direct method is the use of a calibrated set-up which permits a straightforward determination of motion parameters. However, its use is limited to a very well modeled and calibrated environment.

To deal with the general problem (imprecise estimation of motion), it is often necessary to perform an *active* registration of views such as the explicit matching of a few salient features across views (feature based techniques) or the comparison of continuous properties of the surface (curvature, surface normal, depth, etc) measured from different viewpoints.

We have seen that trying to automate the process of discrete feature matching can be very difficult. A big part of the problem in this case is that the neighborhood of a feature has no special characteristics that can help in determining the proximity of this feature (a feature usually being isolated). For this reason, an exhaustive search is often required.

On the other hand, by using smooth and convex metrics, it is sometimes possible to take advantage of the shape of the energy function in the vicinity of the solution, e.g. using a standard minimization procedure to solve the problem assuming that a good initial estimate of the motion is available. The method presented in this thesis falls into the latter category.

In the next chapter, we introduce and describe the particular method of reconstruction used in this thesis before the cross-comparison of views: curvature consistency.

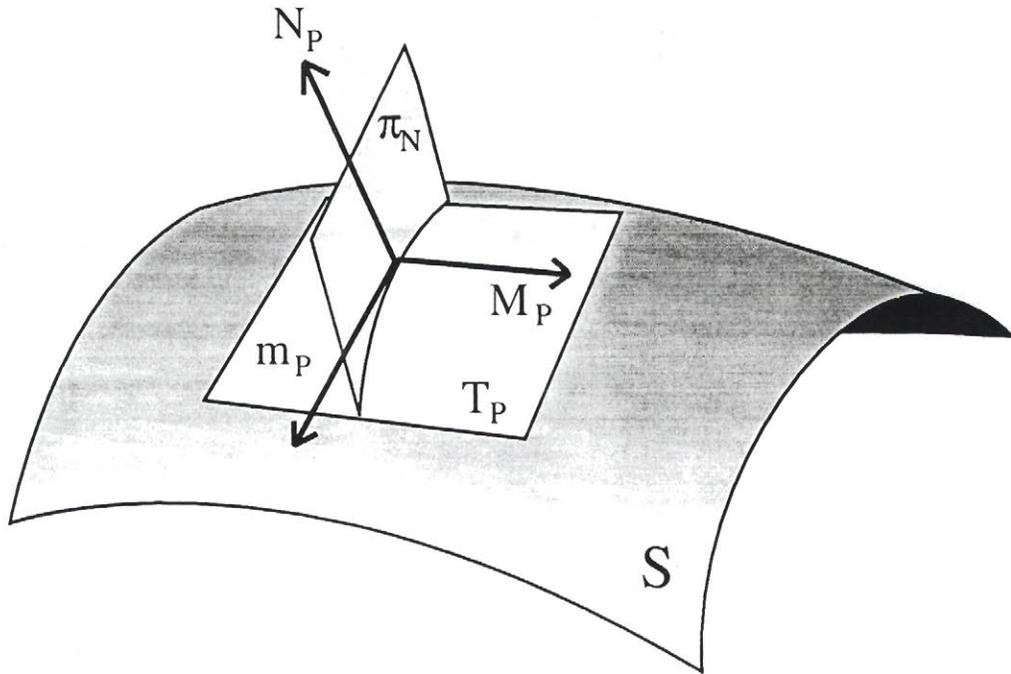
### 3.1 Introduction

Determination of correspondence on discrete surfaces often requires an accurate reconstruction of salient features. Such representations can permit the design of a comparison metric that has the desirable properties of smoothness (piecewise) and convexity of the error surface.

Our approach to the problem is to first obtain a piecewise smooth reconstruction of the underlying surface from which appropriate features (e.g. extremal points of curvature) can be reliably extracted. We are particularly interested in view invariant features that uniquely describe the surface.

Differential geometry can provide a convenient way to locally characterize the surface in such a manner. This characterization can be explained with the aid of figure 3.1. Two planes perpendicular to each other are shown:  $T_P$  tangent to the surface and  $\pi_N$  orthogonal to  $T_P$  and containing the surface normal  $N_P$ . The intersection of  $\pi_N$  with the surface produces a contour referred to as a normal section. The curvature of this section at point  $P$  is called normal curvature,  $\kappa_{N_P}$ . As the orientation of plane  $\pi_N$  is changed, the magnitude of the normal curvature  $\kappa_{N_P}$  varies. It reaches a minimum and a maximum value for two special directions (that can be proven to be perpendicular [4]) which are referred to as principal directions. The corresponding curvatures of these directions are referred to as the principal minimum and maximum curvatures respectively.

In this thesis, the association of the surface normal  $N_P$ , principal directions ( $M_P$  and  $\mathcal{M}_P$ ) and principal curvatures ( $\kappa_{M_P}$  and  $\kappa_{\mathcal{M}_P}$ ) is referred to as the *augmented Darboux frame*,  $\xi_P$  [16]. An *augmented Darboux frame* provides a convenient way to represent the orientation and the structure of the surface at each point. The problem is how to obtain a reliable estimate of the augmented Darboux frame from a range image. In this chapter, we present a solution to this problem: the curvature



**Figure 3.1:** Local surface representation: the augmented Darboux Frame consistency algorithm.

### 3.2 Goals

In its original form [16], the goal of the curvature consistency algorithm was to infer the surface trace points and obtain an estimate of the differential properties of the surface(s) in magnetic resonance images. When applied to a range image, the problem is reduced to the estimation of the differential properties as the trace is known. Specifically, the goal is to estimate and refine the differential properties (normals and principal curvatures and directions) of this unique surface.

Curvature consistency has, in general, a smoothing effect on the surface because it does not explicitly treat the discontinuities. A solution to this problem is proposed in [6] as an extension of the algorithm (adaptive localization of discontinuities).

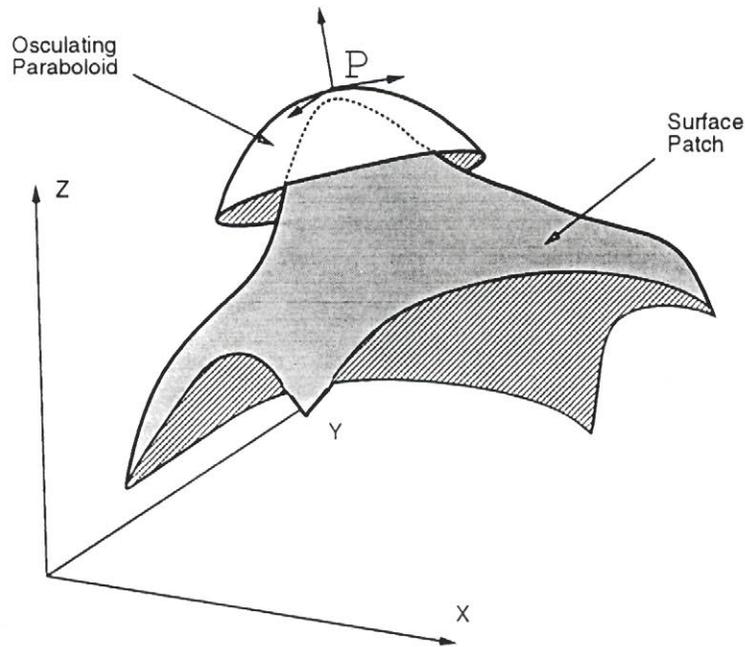


Figure 3.2: Osculating paraboloid at point  $P$

### 3.3 Estimating the Differential Properties

The differential properties estimated by this algorithm are the normal  $N_P$ , the principal curvatures  $\kappa_{M_P}$  and  $\kappa_{\mathcal{M}_P}$  and the associated principal directions  $M_P$  and  $\mathcal{M}_P$  (augmented Darboux frame  $\xi_P$ ).

An estimate of the augmented Darboux frame is required at each point of the surface so that it can be refined according to a local model of surface curvature.

In the case of range images, an estimate of the augmented Darboux frames is found at each point by the least-squares fit of an osculating paraboloid to the point and its immediate neighbors (figure 3.2). The fit includes both positional and surface normal information. The required properties are then simply computed using standard forms of differential geometry from the paraboloid equation [4].

### 3.4 Refining the Estimated Differential Properties

Because the estimates of augmented Darboux frames found at the first step of the algorithm are usually noisy as can be seen in figure 3.3, it is necessary to refine them.

The noise of the estimates is due to the imperfection of the data acquisition process and the computation of derivatives of the signal (curvature is a function of the second derivative of the surface) [9, p.23].

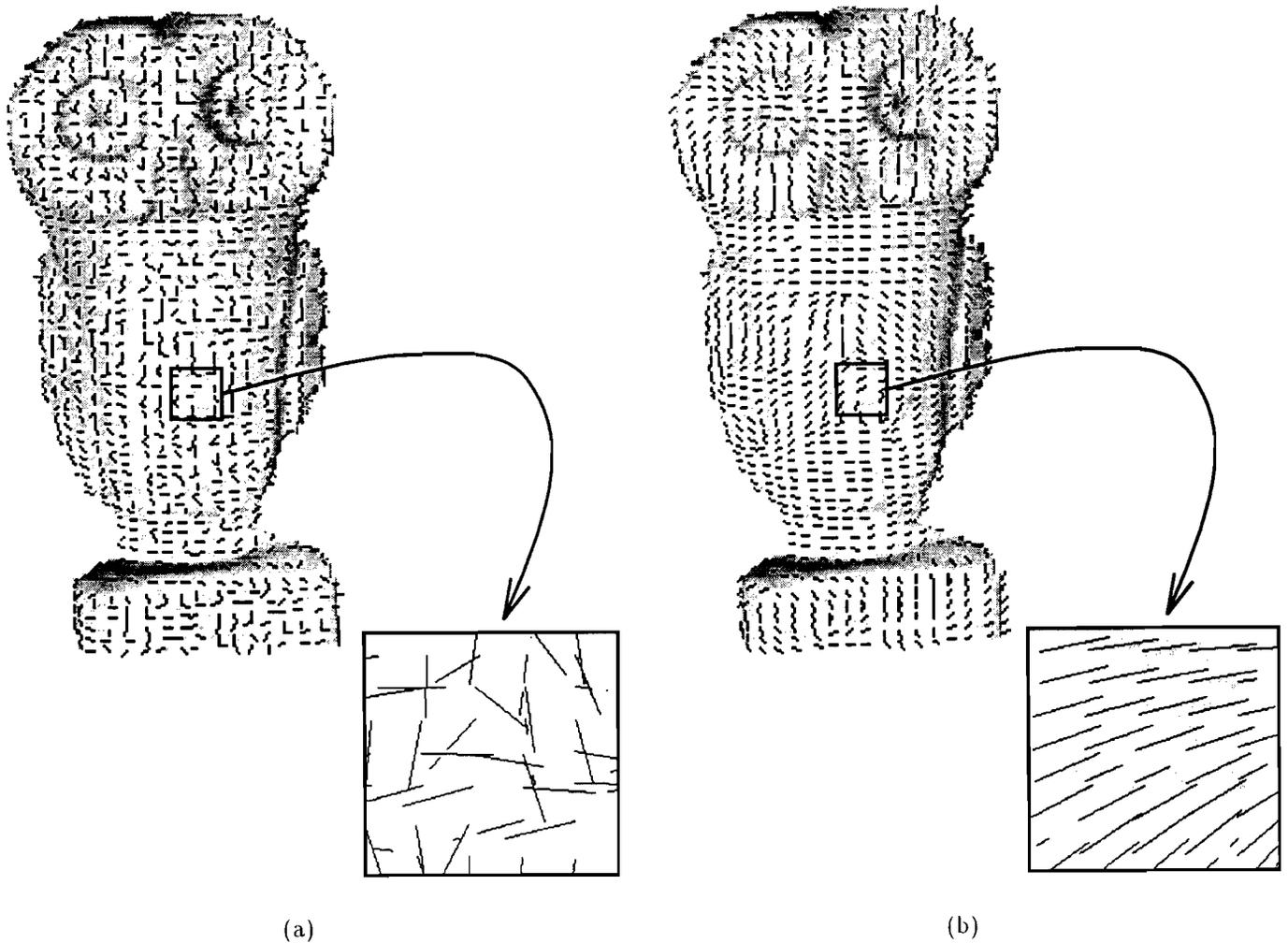
The refinement process consists of three components. The first is a description of the local neighborhood of a point  $P$ , a set of supporting neighbors or contextual neighborhood. The second is a transport model which defines how the structure at a neighboring point  $Q$  is extrapolated to  $P$ . The third is a procedure for determining a maximum likelihood estimate of  $P$  given estimates  $\hat{P}$  obtained from the contextual neighborhood using the transport model.

The contextual neighborhood of point  $P$  includes all the points that are within a certain distance of  $P$  and that are assumed to be part of the same surface. For a range image, this task is usually reduced to choosing the points within a certain window centered on  $P$  or within a certain cylinder [9] (in range data, surfaces with a large slope can have their points spread quite far apart). For this reason, distance alone is not a good criterion to determine the contextual neighborhood of a point. A cylinder of infinite extent centered on  $P$  is chosen to delimit the contextual neighborhood. The radius of this cylinder acts as a scale parameter.

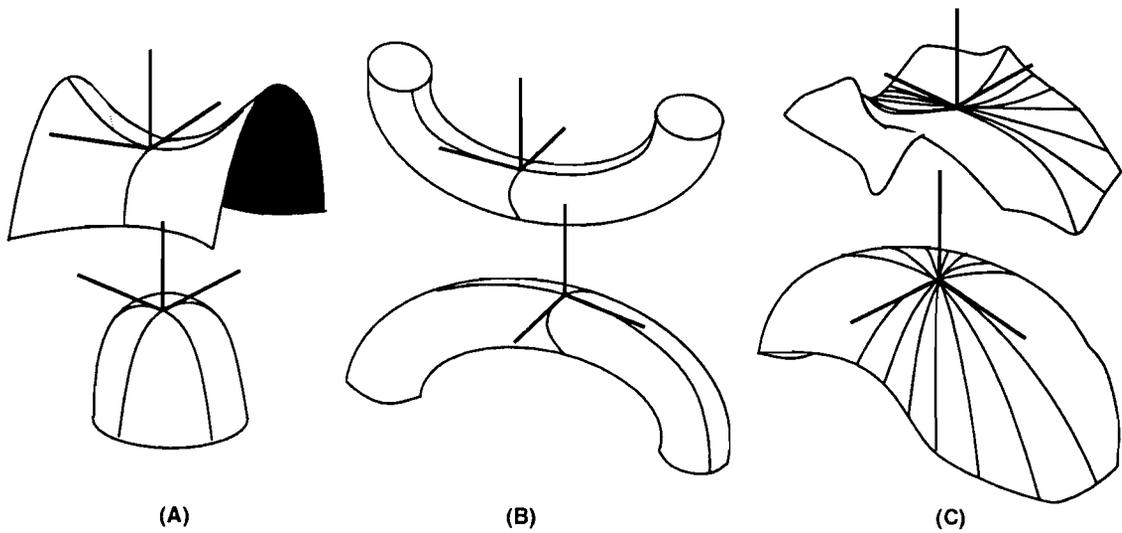
The transport process is summarized in figure 3.5. Frames at adjacent points  $Q_i$  are extrapolated along their respective transport surfaces to the vicinity of point  $P$ . For example, the estimate of the frame  $\xi_P$  determined from the frame at  $Q_4$  is  $\xi_{P_4}$  as shown in figure 3.5.

The transport surface embeds a local model of the surface which enforces a smooth variation of curvature, e.g. the locally constant curvature constraints [13].

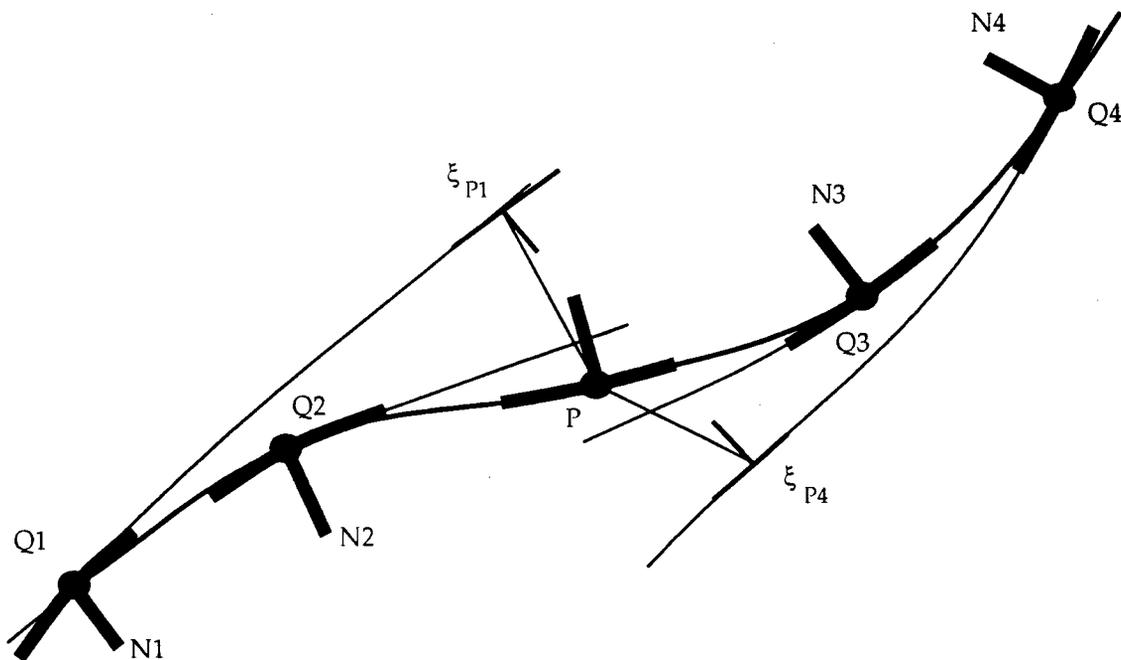
Many different types of surface can be used for this purpose, as shown in figure 3.4. In the implementation used in this thesis, the transport surfaces are paraboloids. Paraboloids have the advantage of being analytically simple. However, as is shown in [9], a paraboloid has higher curvature at its origin. Therefore, when using them for transporting Darboux frames, the curvature is reduced by a small amount at each iteration. For a large number of iterations, this effect often results in a large reduction of the curvature which produces a smoother surface.



**Figure 3.3:** (a) Estimation of one of the principal directions for a small owl statuette. The principal direction field is particularly noisy in low curvature regions. (b) Result of the application of five iterations of curvature consistency to the noisy field shown in (a) using a neighborhood of 5 by 5. Note the consistency in the principal direction field.



**Figure 3.4:** Different type of transport model. (a) shows parabolic patches. The magnitude of their normal curvature is maximum at the origin. (b) shows toroidal patches. They have constant curvature in only the two directions shown with black lines in the figure. (c) shows a more complex analytical surface with constant curvature in all directions.



**Figure 3.5:** Mechanism of parallel transport of Darboux frames in curvature consistency. This figure shows a 2D version of transport of the neighbors' normals. Here, information at point  $P$  is to be updated by the neighbors' support (points  $Q$ 's). As can be seen, transport is realized by extending the neighbor's local paraboloid (bold) until point  $P$  can be projected onto it in the direction of the neighbor's normal. The normal (and principal directions and curvatures) at the point of projection of  $P$  on the paraboloid patch is the parallel transported normal for the corresponding neighbor.

The final step of the process of refinement consists of using the transported Darboux frames to *update* the estimate at point  $P$ . This is accomplished by defining a maximum likelihood estimate of the properties at point  $P$  given the constraints resulting from the transport process (i.e. the set of augmented Darboux frames). Using standard methods (e.g. Lagrange multipliers), a set of equations enforcing the locally constant curvature and constraints relative the nature of the Darboux frame are minimized. The details of this process are given in Appendix A.

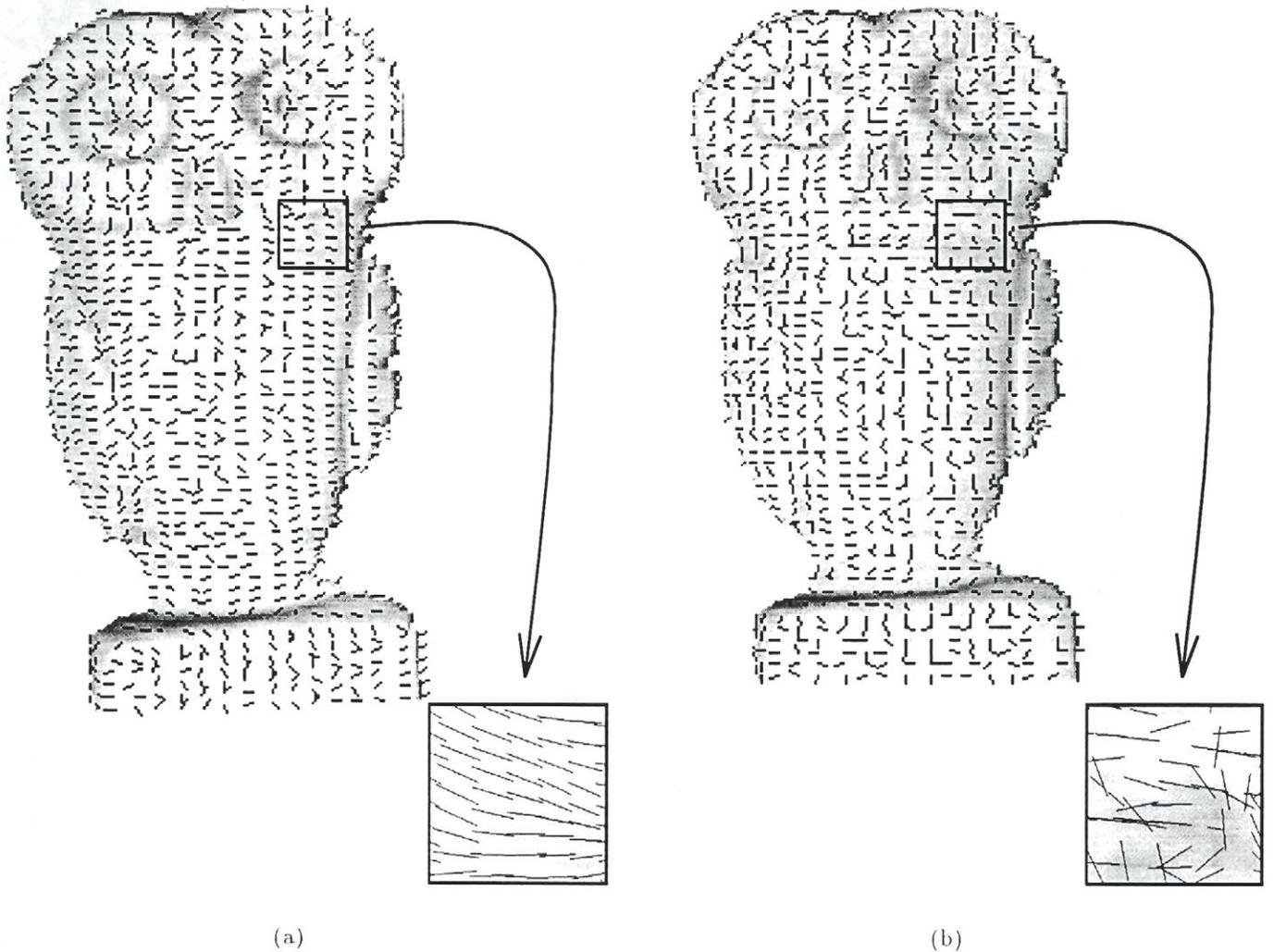
Figure 3.3b shows the result of applying five iterations of curvature consistency to the noisy field shown in figure 3.3a. More detailed examples of the method can be found in [9].

## 3.5 Discussion

Curvature consistency is a form of “feature-preserving” smoothing. The features preserved are the extended Darboux frames at each points. To see that this is the case, consider the result shown in figure 3.6. The left side shows one of the principal curvature fields as recovered by curvature consistency after 5 iterations. The right side also shows one of the principal curvature fields of the same surface after a convolution with a Gaussian. The field recovered with curvature consistency is clearly more coherent particularly in small curvature regions.

This shows an important aspect of this reconstruction algorithm: it can recover a stable and coherent description of differential properties of a surface. Even though other reconstruction algorithms have qualitatively the same smoothing effect on the surface (see figure 3.7), they usually fail in recovering a consistent description of the differential properties.

The differential properties of a surface have an important quality: they are view invariant. Because of that, they are used in this thesis as the basis of cross-comparison of views in order to determine correspondence. Curvature consistency is the way to estimate them in a stable and consistent manner.



**Figure 3.6:** Comparison of one of the principal direction fields found with curvature consistency and convolution with a Gaussian. The curvature field in the left image has been estimated with curvature consistency. The field for the Gaussian convolution (right image) is the result of an estimation with least-square fit of paraboloids (same process used to initialize curvature consistency). The field for curvature consistency is clearly more coherent particularly in small curvature regions.



**Figure 3.7:** Comparison between curvature consistency and convolution with a Gaussian for a small owl statuette. Left image is the result of curvature consistency. Right image is the result of convolution with a Gaussian with a sigma making an approximately “equivalent” amount of smoothing for both filters.

## 3.6 Summary

In this chapter, I have described the specific method used to reconstruct and stabilize the scenes before their integration, namely the curvature consistency algorithm. This method produces in few iterations a coherent curvature field that will be shown to be consistent enough to be compared across images.

Curvature consistency is a feature preserving smoothing technique that is different than a simple convolution with a Gaussian. Even though both methods can apply qualitatively the same amount of smoothing, curvature consistency has the ability to recover and preserve a consistent curvature field.

## 4.1 Introduction

In this chapter, I present in detail my solution to the problem of view correspondence. The discussion starts with a precise description of the problem at hand: the input needed, the output produced as well as a general description of the method.

Section 4.3 deals with refining the estimation of motion. Key concepts are presented and discussed in detail.

Next, the concept of motion consistency is introduced. We will see how this procedure can be used to improve local motion estimates by enforcing a smooth variation of motion parameters on the surface.

Section 4.5 presents a discussion on the meaning of non-rigid motion and on the different parameters affecting it (window size, number of iterations...). It will be seen that non-rigidity of motion is a concept tightly linked to scale.

Finally, in Section 4.6, the method presented in this chapter will be compared to classical continuation methods.

## 4.2 Problem Definition

The problem is to find the correspondence or spatial relationship of two overlapping views of an object. More precisely, because we don't assume globally rigid motion, we are looking for a *set* of rotation matrices  $\mathbf{Q}_i$  and translation vectors  $\mathbf{T}_i$  mapping each point or set of points (patch) of one view into another view coordinate frame.

Finding correspondence of views can be seen to be useful simply because it increases our knowledge of the world. In this sense, it may well be the first step of many other higher level tasks such as object grasping, mass center evaluation, path planning, modeling, object recognition, etc. The common point to all these specific applications is that they can be simplified because more information about the world

is available after the integration of multiple views [22].

The method presented in this chapter works with range images or with any high density depth representation. Two (or more) overlapping views are needed together with an estimate of the displacement of the viewer (and/or of the object). The precision of this estimate depends on the structure of the scene and is discussed later.

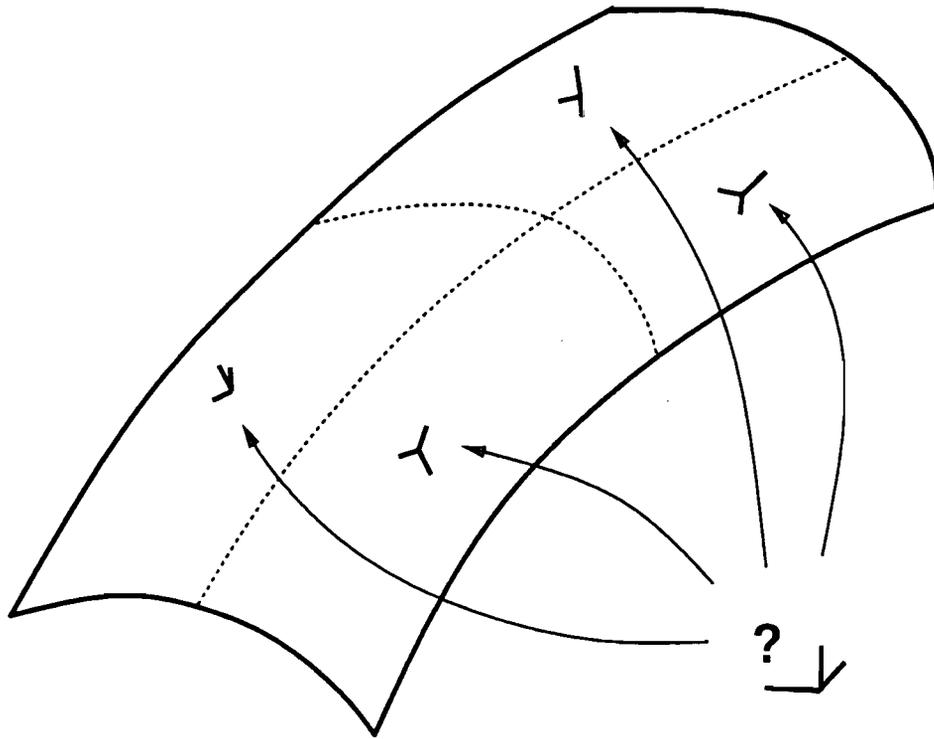
The method assumes that surfaces are piecewise smooth. However, they must contain sufficiently distinguished features such that correspondence can be determined uniquely. Surfaces such as a plane, a sphere or a cylinder for example will cause the method to fail (as a human would anyway if he (she) were allowed to use only the same clues as the method).

Assuming that the previous constraints are met, the problem is to find the correspondence of the overlapping regions in each view and to correctly map the unique information (present in only one of the views) to the other. Correspondence is determined locally without the assumption of globally rigid-body motion. The object is rather allowed to deform slightly from one view to the other. The correspondence is based on a comparison of curvature fields across views.

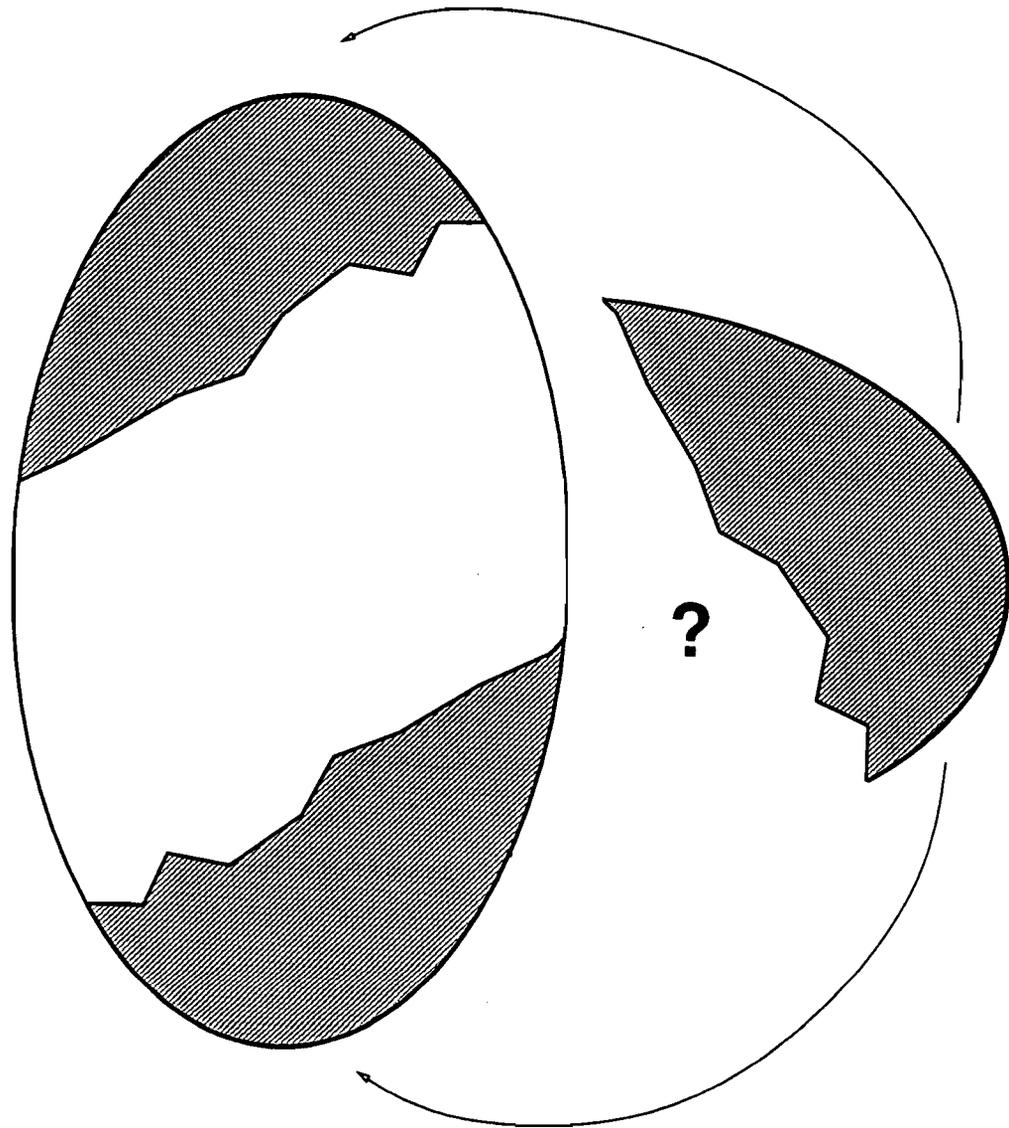
The output of the overall process is a set of motion parameters relating each patch (or point) of a surface seen in one view to the same patch (or point) seen (or occluded) in the other view. In the final representation, it is therefore common to have two points (one from each view) that describe the same part of an object. The *integration* (or fusion) of these corresponding multiple descriptions into a single one would be a natural next step of this work [12].

### 4.3 Refining Motion Estimates

Consider the problem depicted in Figure 4.1 which consists of finding (matching) a specific extended Darboux frame  $\xi_P$  on a parabolic patch. By using only the vector part of  $\xi_P$  (normal and principal directions), an infinite number of matches can be found (by changing its orientation, a frame can always be matched exactly with any other frame).



**Figure 4.1:** Because of symmetry, a single extended Darboux frame (curvature magnitude included) can be matched exactly at four locations on a paraboloid. This necessitates either a richer description (neighborhood) or of an estimate of the solution.



**Figure 4.2:** Again, because of symmetry, a neighborhood of Darboux frames can be matched exactly at two locations on a paraboloid. An estimate of the solution is required in order to obtain a unique match.

In order to restrict the number of solutions, it is necessary to use more information to do the matching e.g. using the principal curvatures. Again, as shown in Figure 4.1, this match is not unique. Because of the symmetry of the figure, four distinct matches can be found for  $\xi_P$  with exactly the same sign and magnitude for the principal curvatures.

Even a comparison based on the local neighbors of  $\xi_P$  would not lead to a unique solution (two solutions would still exist, again because of symmetry as shown in Figure 4.2). However, if we are given an estimate of the solution (let's say precise up to a quadrant of the paraboloid), the unique and exact match can now be found.

This example summarizes the requirements for finding a unique solution for the correspondence of two views. We need a rich (unique), stable and view invariant description of the surface. We also need an estimate of the solution for two principal reasons. First, it is *necessary* in order to determine the unique solution even on simple analytic surfaces. Second, it is *useful* in order to reduce the search space.

The steps required to determine the set of motion parameters relating two views can be summarized as follows:

1. **Curvature consistency on both images separately.** The goal of this step is to make the curvature field vary smoothly in both views to facilitate comparison later.
2. **Subdivision of one of the views into small windows of  $n \times n$  points.** The division is done according to the intrinsic grid of the image. All the points within a window will be moved together (same motion parameters). However, each patch will have its own and independent motion parameters mapping it to the other view.
3. **Selection of the largest patches of constant Gaussian curvature on the first view.** The selected windows will be the first ones for which correspondence will be solved because they are the most likely to result in a unique solution.
4. **Solution of local correspondence for these selected windows.** In the correspondence process, an initial estimate of motion is used as starting point

of a minimization based on a local comparison of differential properties across views.

5. **Propagation of motion parameters to the windows surrounding the selected patches.** The motion parameters determined for the selected windows are used as a starting point for determining the solutions of neighboring points.
6. **Application of motion consistency.** Refine motion estimates according to the smooth deformation constraint.

The remainder of this section describes these steps in detail.

#### 4.3.1 Application of Curvature Consistency on Both Views Separately

This first step is important in order to meet one of the essential conditions of the process, that is a piecewise smoothly varying curvature field on both views. This will help to simplify the minimization process described later.

It would be tempting to skip this pre-processing stage and try to recover both the motion and the local structure directly in the same step. In other words, why not use an extended neighborhood (points from both views) in the curvature consistency algorithm? The goal would then to try to minimize the residual error using different motion parameters (different neighbors).

In fact, this does not work. The reason is that, especially with a noisy set of images, it is possible to obtain a smaller residual error with the *wrong* neighbors, giving as a by-product the *wrong* motion parameters. In other words, these two problems namely surface reconstruction and view correspondence (structural and temporal problems) confound one another and cannot be solved concurrently.

This explains why it is important to get a stable description of the surface structure via curvature consistency before any attempt of cross-comparison of the views. The goal of the application of curvature consistency is to diminish the noise effect and to

get a stable and reliable estimate of the principal curvatures and directions. For this purpose, the algorithm is applied to both images separately, preparing them for the process of view correspondence.

The smoothness of the curvature field so obtained allows a comparison of the surfaces across views such that the associated motion parameters can be determined by a convex minimization procedure as described in the following sections.

### 4.3.2 Subdivision of the First View in Small Windows

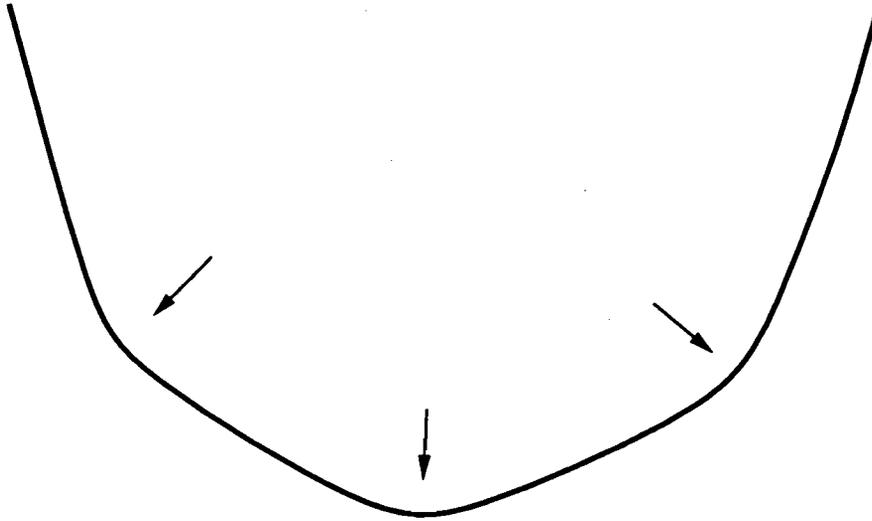
Following reconstruction, the next step is to divide the surface described by the first view in small windows of  $n \times n$  points. The number of regions and their size depends on the rigidity of the motion. In general, the size of these windows varies from  $3 \times 3$  to  $7 \times 7$  points.

Each of them is to be mapped independently of the others to the second view. At the end of the correspondence process, each window of  $n \times n$  points in the first view will have its own set of motion parameters (rotation and translation) mapping it to the other view. By solving correspondence *locally*, the method is able to cope with non-rigid motion, i.e. the surface can deform globally provided that the local structure defined by its curvature varies slowly.

Each patch or window will undergo a rigid motion i.e. each point within a patch will be assigned the same motion parameters. However, because patches can move independently, the structure of the surface as a whole can accommodate deformations. Besides providing a means of dealing with non-rigid motion, this subdivision is also useful in segmenting the surface into different categories of patches that can be treated differently.

### 4.3.3 Classification of the Patches

The Gaussian curvature (product of the minimum and maximum principal curvatures) is used to characterize the surface in the first view. The sign indicates whether the surface is elliptical (+), hyperbolic (-) or parabolic (zero Gaussian curvature).



**Figure 4.4:** 2D slice of a convex surface (constant KH-mapping). The three positions pointed to by the arrows have the same local structure in terms of curvature and are, in this sense, indistinguishable. It shows that KH-mapping is not a sufficient condition to guarantee a unique solution to correspondence.

parabolic patches do not fall into this category because their structural information content in one of the direction is too low (null curvature) and would complicate the matching process.

For the correspondence algorithm, it is necessary that the initial estimate results in a mapping onto the corresponding patch in the other view.

Even if the starting point of the comparison is in the same constant KH-mapping region, there is no guarantee that the result of the minimization will be the global solution. Figure 4.4 shows such an example. It is a convex figure (constant KH mapping) with a repetition of the same structure as shown by the three arrows. For example, a gradient descent procedure minimizing the difference between two regions would stop at the nearest solution and would not see the others. On a local basis and in terms of curvature only, there is no way to distinguish the three positions indicated by the arrows.

In this sense, trying to start in the same constant KH-mapping region gives only a limited guarantee of success. By landing on a similar patch, the smoothness of the curvature field is likely to guide the gradient descent to the correct answer which

would not be the case if we started in the wrong region. This might be viewed as a necessary but insufficient condition. The hope is that the stabilizing effect of curvature consistency will be strong enough within a region to make the curvature field vary smoothly and have a consistent structure in the immediate vicinity of the solution (which is verified experimentally).

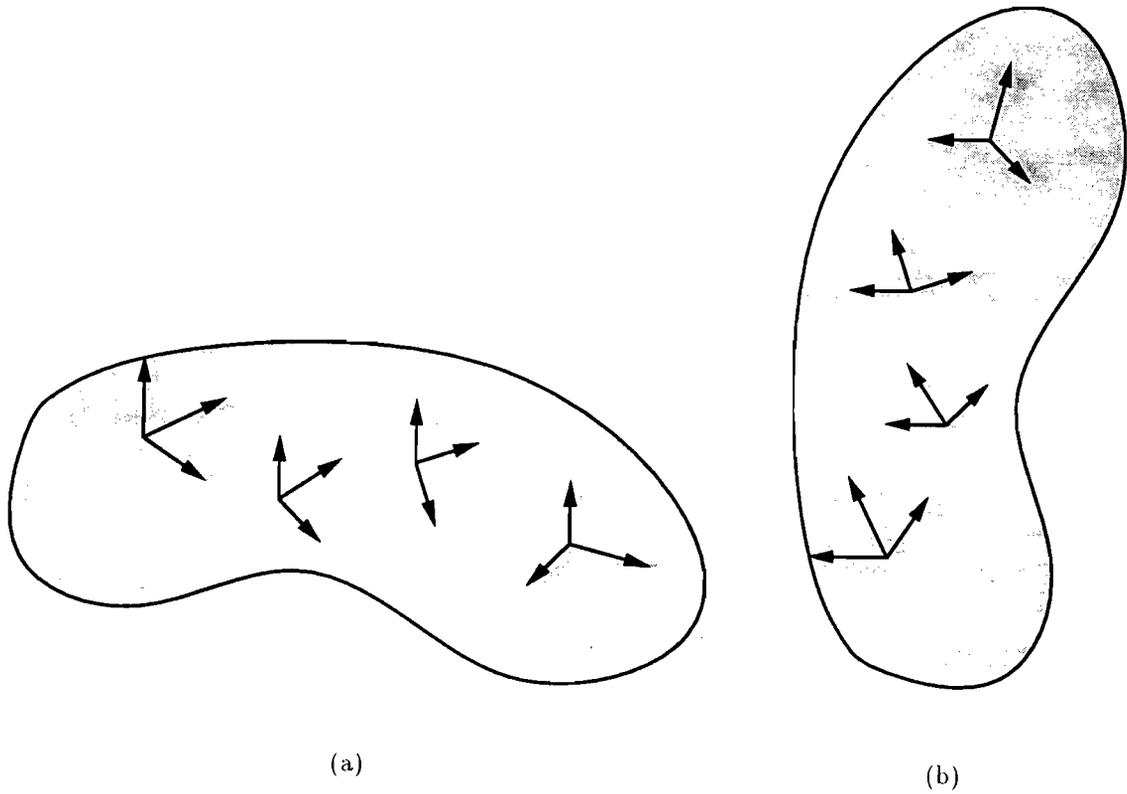
Usually, five to ten such patches are selected for the initial stage of the matching process which consists of minimizing the metric defined in the following section.

#### 4.3.4 Defining a Metric of Similarity

The basis of our approach is the minimization of a functional form that measures the difference between a local neighborhood in one image and a corresponding neighborhood in an adjacent image. We describe the local structure of a point  $P$  in the first view with the augmented Darboux frame  $\xi_P$  and, in the second view, with the augmented Darboux frame  $\xi_{P'}$ . The augmented Darboux frame at a point is determined as described in the previous chapter. A typical organization of the neighborhood at point  $P$  in both views is shown in Figure 4.5.

Assuming that the best estimate for the motion of point  $P$  is given by the rotation matrix  $\mathbf{Q}$  (containing the Euler angles  $\theta_x, \theta_y, \theta_z$ ) and the translation vector  $\mathbf{T}$  ( $x, y, z$ ), we can write the relationship between the components of the Darboux frames  $\xi_P$  and  $\xi_{P'}$  as follows:

$$\begin{aligned}
 \mathbf{p}' &\leftrightarrow \mathbf{Q}\mathbf{p} + \mathbf{T} \\
 \mathbf{M}_{P'} &\leftrightarrow \mathbf{Q}\mathbf{M}_P \\
 \mathcal{M}_{P'} &\leftrightarrow \mathbf{Q}\mathcal{M}_P \\
 \mathbf{N}_{P'} &\leftrightarrow \mathbf{Q}\mathbf{N}_P \\
 \kappa_{M_{P'}} &\leftrightarrow \kappa_{M_P} \\
 \kappa_{m_{P'}} &\leftrightarrow \kappa_{m_P}
 \end{aligned} \tag{4.1}$$



**Figure 4.5:** Typical organization of a neighborhood in two views. (a) shows a few Darboux frames in the neighborhood of a point seen in a first view. (b) shows exactly the same neighborhood seen from another viewpoint (after translation and rotation). The metric of similarity defined allows comparison of the structure of these neighborhoods by taking into account the motion parameters.

Observe that the principal curvatures ( $\kappa_{M_P}$  and  $\kappa_{m_P}$ ) are viewpoint independent. Therefore, they can be compared across views without any geometric transformation. In a similar manner, it is also important to note that the three vectors comprising the so-called *principal frame*, namely the vectors  $M$ ,  $\mathcal{M}$  and  $N$  are unit length and have no position content. One needs only to adapt their orientation through the rotation matrix  $\mathbf{Q}$  before comparing them across views. These vectors carry only a directional content; the translation vector  $\mathbf{T}$  has no effect on them. Finally, the position of point  $P$  in the second view with respect to the first one is determined with the complete set of motion parameters, rotation and translation.

The difference in the orientation of the Darboux frames and values of the principal curvatures is computed with equation 4.2. Each of the components of this equation is normalized to unit measure and weighted identically.

$$D_{\mathbf{Q}\mathbf{T}} = 3 - \langle M_P, M_{P'} \rangle^2 - \langle m_P, m_{P'} \rangle^2 - \langle N_P, N_{P'} \rangle^2 + \frac{|\kappa_{MP} - \kappa_{MP'}|}{|\kappa_{MP}| + |\kappa_{MP'}|} + \frac{|\kappa_{mP} - \kappa_{mP'}|}{|\kappa_{mP}| + |\kappa_{mP'}|} \quad (4.2)$$

A simple analysis of the previous metric shows that its value is contained in the range 0 to 5, varying from a perfect similarity of structure to a completely different one.

Change in the motion parameters  $\mathbf{Q}$  and  $\mathbf{T}$  leads to a different corresponding Darboux frame  $\xi'_P$  and therefore to a different value of the metric  $D$  that measures the similarity between  $\xi_P$  and  $\xi'_P$ . The idea is to minimize the functional  $D_{\mathbf{Q}\mathbf{T}}$  as a function of values of  $\mathbf{Q}$  and  $\mathbf{T}$ .

However, exhaustive search is not a good idea here because the search space is immense. The error surface, that is to say the metric  $D$  plotted for the various  $\mathbf{Q}$  and  $\mathbf{T}$ , is contained in a 6D space (3D rotations and 3D translations). In practice, however, the problem is reduced to a 5D space because the solution, the point minimizing the difference of structure, is known to lie on the surface defined by the second view, thereby imposing a constraint on the solution and removing one degree of freedom. The search space remains nevertheless too large for an exhaustive search.

In fact, there is no need for a global search because we assume that an estimate of the motion is available. We can therefore limit our search to a certain neighborhood in the vicinity of this position. We will demonstrate that a solution can be found using a gradient descent procedure provided that the conditions described in the next section are met.

### 4.3.5 Existence and Uniqueness of the Solution

The existence of a solution of the gradient descent formulation of the problem at hand (namely minimizing the metric  $D$ ) implies the existence of the metric itself. And the existence of this metric depends solely on the availability of estimates of the necessary differential properties of the surface (Darboux frame).

It is therefore important to be able to estimate the principal curvatures and directions. This means that the surface should be  $C^2$  smooth<sup>2</sup> since the curvature magnitude is related to its second derivative. We will assume that the surfaces used in this process meet this condition (at least locally - piecewise smooth surfaces).

In addition to its existence, the metric  $D$  must be convex to have a unique solution. The convexity of  $D$  is not easy to show for the general case. Even for a special case such as a paraboloid a formal proof is somewhat difficult.

For example, the principal curvatures of a paraboloid in standard position

$$z = 1/2 \alpha_1 x^2 + 1/2 \alpha_2 y^2 \quad (4.3)$$

are given by

$$\kappa_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (4.4)$$

where

$$a = 1$$

---

<sup>2</sup>Continuous to second order

$$b = -\frac{\alpha_1 + \alpha_2 + \alpha_1^2 \alpha_2 x^2 + \alpha_1 \alpha_2^2 y^2}{(1 + \alpha_1^2 x^2 + \alpha_2^2 y^2)^{3/2}}$$

$$c = \frac{\alpha_1 \alpha_2}{(1 + \alpha_1^2 x^2 + \alpha_2^2 y^2)^2}$$

and the principal direction (not normalized) corresponding to the principal curvature  $\kappa_1$  is given by

$$M = (\alpha_2 + \alpha_1^2 \alpha_2 x^2 - \mathcal{D}, \alpha_1^2 \alpha_2 xy, \alpha_1 x (\alpha_2 + \alpha_1^2 \alpha_2 x^2 - \mathcal{D}) + \alpha_1^2 \alpha_2^2 xy^2) \quad (4.5)$$

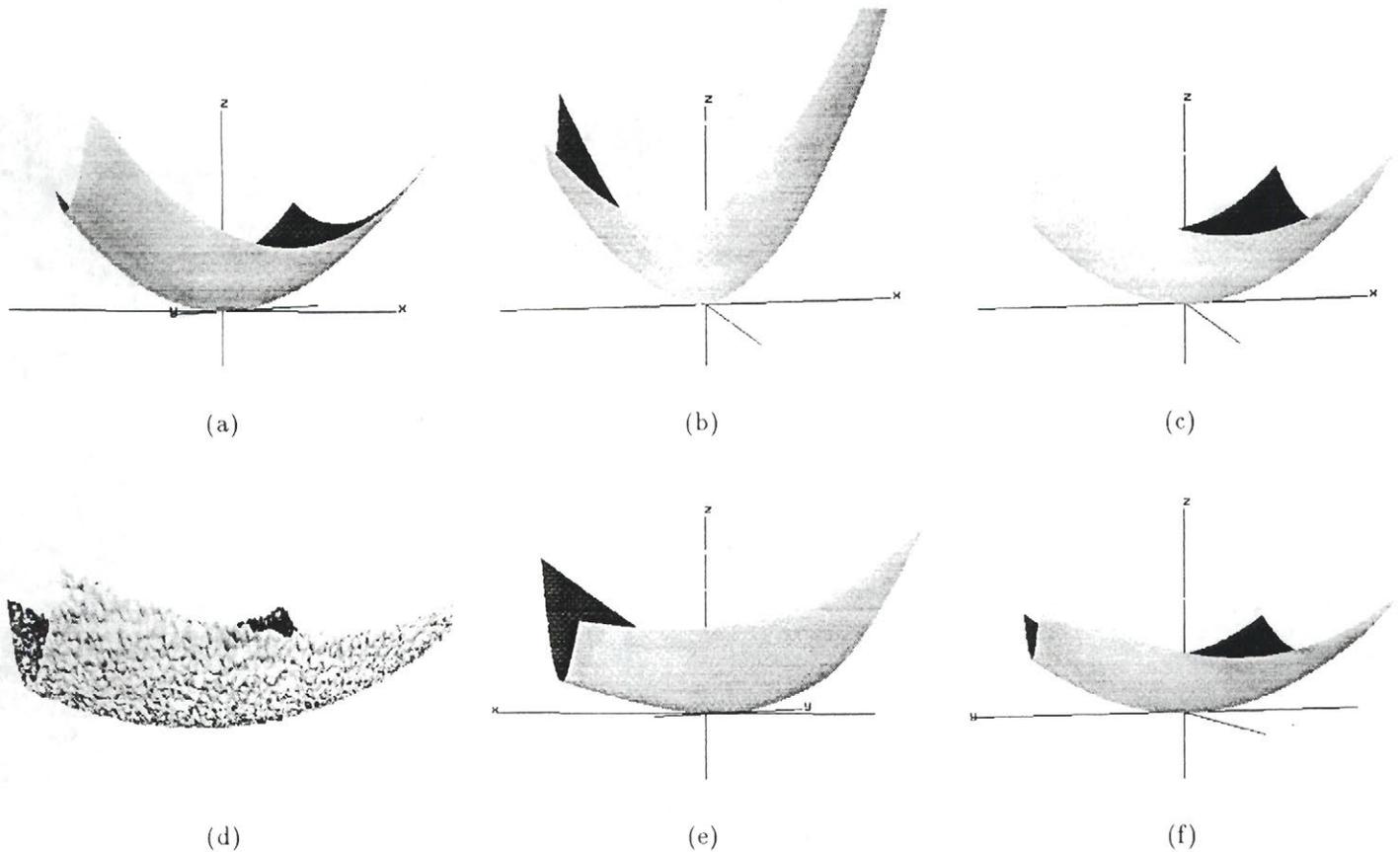
where

$$\mathcal{D} = \kappa_1 (1 + \alpha_1^2 x^2 + \alpha_2^2 y^2)^{3/2}. \quad (4.6)$$

In order to prove the convexity of  $D$  for this specific form, one would have to replace the variables  $x$  and  $y$  in equations 4.5 and 4.5 by functions having for arguments the motion parameters  $Q$  and  $T$ . The resulting equations could then be substituted in the metric  $D_{QT}$ .

Because of the complexity of the resulting form, we do not offer any formal proof of the convexity of the metric  $D$ . However an empirical demonstration is presented in Figure 4.14. It shows four slices of the error surface obtained using the metric  $D$  when trying to match an arbitrary Darboux frame from this surface (without and with noise added) with another point on the same surface. The convexity of the resulting error surface is apparent. A similar result has been obtained for an hyperboloid.

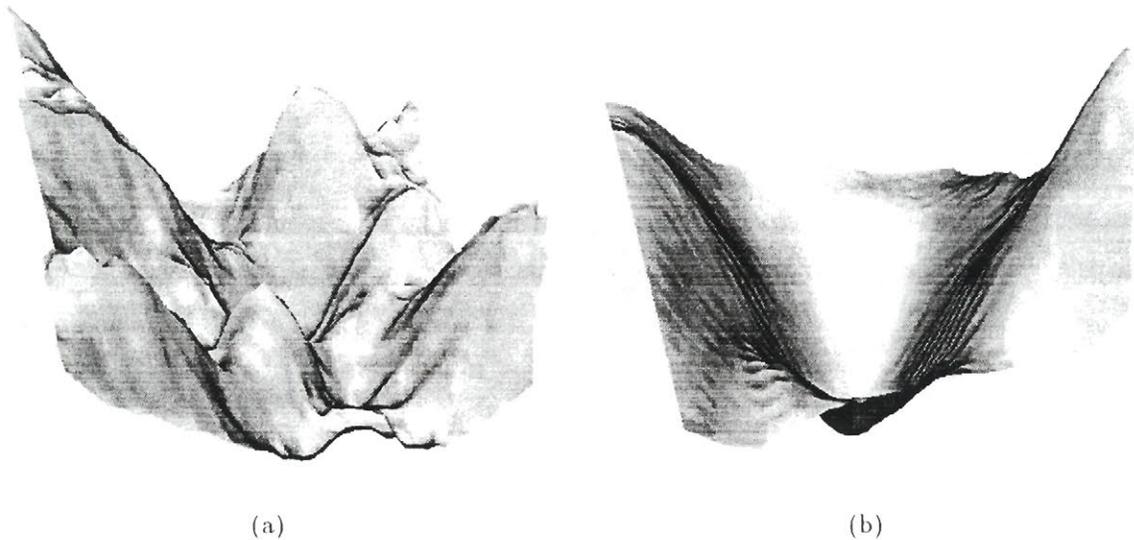
The interest of the parabolic case is the following. For any surface that can be locally approximated by a paraboloid (or an hyperboloid), the metric  $D$  is convex in the vicinity of the solution. The size of the region of convexity depends on how good the approximation is. For raw data, this approximation is, in general, good for a very limited area around the solution. However, application of an appropriate reconstruction procedure can extend this region significantly as is shown in Figure 4.7.



**Figure 4.6:** Convexity of the metric  $D$  for two views of a synthetic paraboloid. (a) shows a view of the noiseless paraboloid used. (b) is a slice (displacement in  $x$  and in  $y$ ) of the metric  $D$ . (c) is another slice (rotation in  $\theta_x$  and  $\theta_y$ ) of the metric  $D$ . Both slices are clearly convex. (d) shows the same paraboloid with 5% white noise added. (e) and (f) present the same result as (b) and (c) respectively for the paraboloid in (d). Results in (e) and (f) were obtained after 50 iterations of curvature consistency.

This demonstration serves to show that the method degrades gracefully (the same surfaces in Figure 4.6 (e) and (f) with fewer iterations would have many local minima).

The number of iterations of curvature consistency performed has a large effect on the shape of the metric  $D$ . Figure 4.7 shows a slice of the hyper-surface ( $6D$ ) produced by the metric  $D$  for two views of a small owl statue (Figure 5.13). The result shown is typical of the effect of curvature consistency on the convexity of  $D$ . This phenomenon and its consequences are presented in more detail later in Section 4.6.



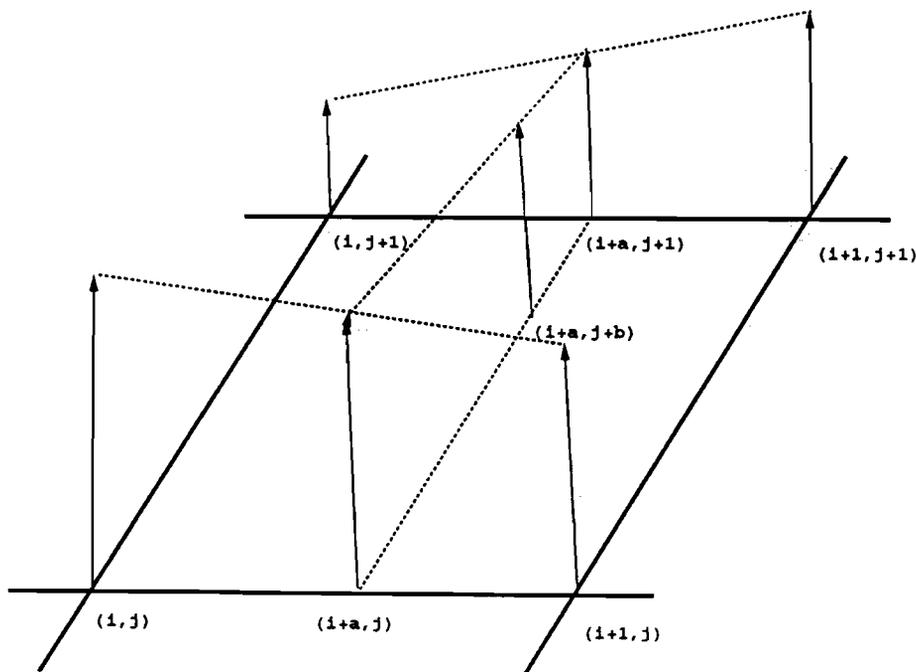
**Figure 4.7:** Effect of curvature consistency on the convexity of  $D$ . A 2D slice of the hyper-surface ( $6D$ ) produced by  $D$  for two views of a small owl statue is presented after different numbers of iterations of curvature consistency. (a) shows this surface after one iteration and (b) presents the same figure after five iterations. Many local minima present in (a) have disappeared in (b) where the convex shape is more clearly apparent.

#### 4.3.6 Minimizing the Metric $D$

The initial estimate of motion parameters is used to obtain a starting point for the process. The parameters are applied to a point  $P$ , mapping it from its position  $\mathbf{p}$  on the first view to its corresponding position  $\mathbf{p}'$  on the second view. The nearest position  $\mathbf{p}'$  is then chosen as the starting point of the minimization. If the distance from the position  $\mathbf{p}'$  to the nearest point on the second surface is larger than a specified threshold, point  $P$  is said to correspond to unique information visible in only the first view.

$D$  is minimized in gradient descent fashion. The gradient of  $D$  is estimated from discrete samples in the vicinity of  $\mathbf{p}'$ . Because of the convexity of the metric in this neighborhood, the minimal solution can be found quite efficiently (as compared to exhaustive search).

However, some care must be taken in applying the procedure. For example, a slight change in the rotation parameters does not induce a corresponding change in

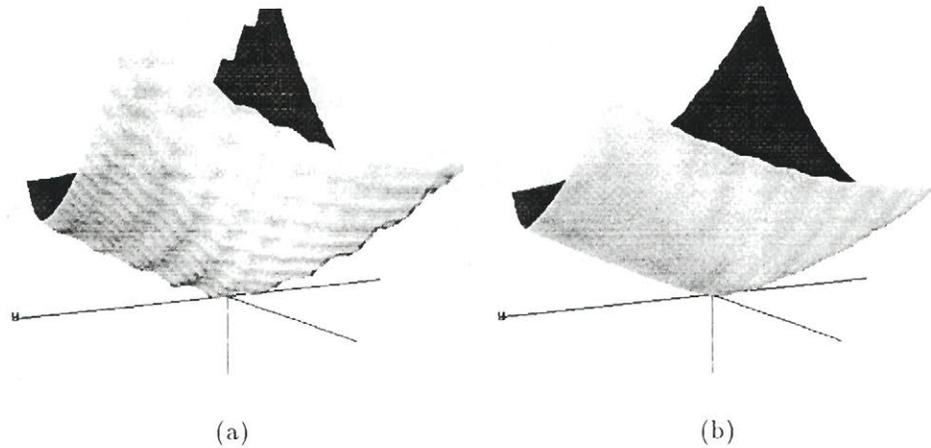


**Figure 4.8:** Bilinear interpolation of a range image. The range information is seen as a set of impulses at the grid corners. In this example, the value at position  $(i+a, j+b)$  is needed where  $a$  and  $b$  are smaller than 1. The first step consists in interpolating the values at position  $(i+a, j)$  and  $(i+a, j+1)$ . These two values are interpolated in their turn to obtain the desired value at  $(i+a, j+b)$ .

$\mathbf{p}'$  due to quantization. As a result, steps are induced in the error function, making it more difficult to minimize.

This type of behavior can be reduced by interpolating between the points. For example, a simple bilinear interpolation is sufficient to keep the error continuous and make it easier to minimize (see Figure 4.9). The meaning of bilinear interpolation for a range image is graphically explained in Figure 4.8.

The solution obtained by the gradient descent procedure is rejected if the value of the metric  $D$  is larger than a threshold determined by the size of the neighborhood, the noise present in the images and the deformability of the object. The motion parameters are deduced from the positions  $\mathbf{p}$  and  $\mathbf{p}'$  of point  $P$  in the first and second views respectively together with the corresponding preferred relative rotation.



**Figure 4.9:** Effect of a bilinear interpolation on the metric  $D$ . A slice of the 6D surface obtained with the metric  $D$  for two views of a synthetic (noiseless) paraboloid is shown in two circumstances. (a) shows the surface without interpolation (nearest point) which is bumpy and has many local minima. (b) shows the same surface found with bilinear interpolation. This time, the surface is smoother and easier to minimize

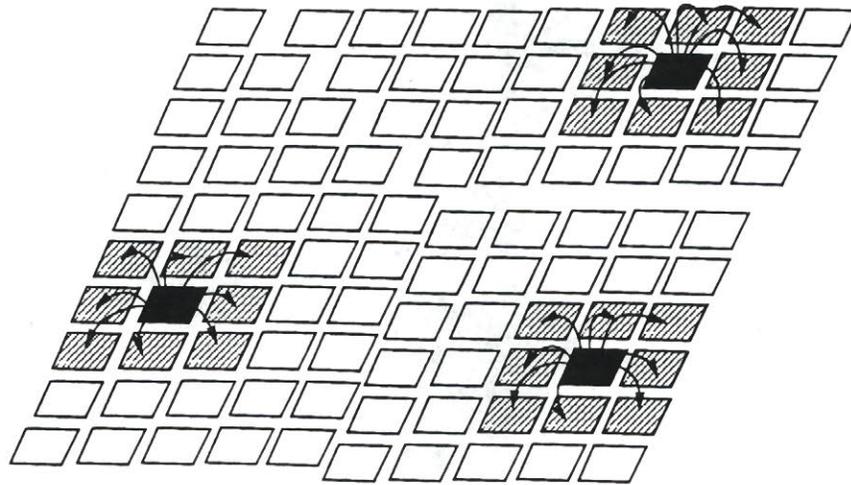
### 4.3.7 Propagating the Motion Parameters

At this stage, only few selected patches (see Section 4.3.3) have been matched in the second view through a complete minimization of the metric  $D$ . In order to associate a set of motion parameters to every patch in a view, a strategy of gradual propagation of information is adopted.

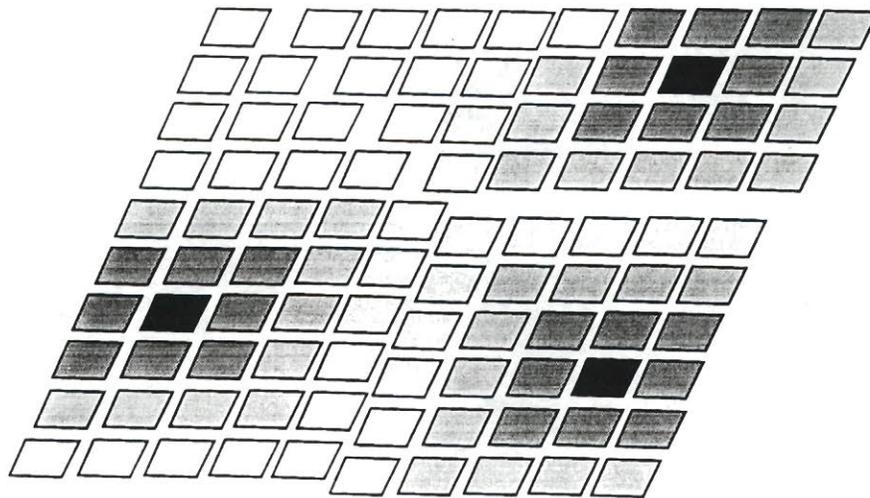
In this process, the selected patches (black patches in Figure 4.10) bequeath their motion parameters to their immediate neighbors. This information is then used by the neighbors as a starting point to find their own motion parameters through minimization of the metric  $D$ .

The process of *updating* the motion parameters is possible only if the neighbor is part of an overlapping region. This is determined by a distance threshold. If, when mapped with the parameters of the selected point, a neighbor is too *far* from the surface in the second view, it is said to be a unique information and its motion parameters can not be updated.

The advantage of the propagation is that the search space is reduced by using a very good starting point (motion parameters of an immediate neighbor). It also



(a)



(b)

**Figure 4.10:** Principle of propagation of the motion parameters. The patches are symbolized by small rectangles all having their own motion parameters. Only the *black* patches have had their motion parameters refined with the gradient descent method described earlier (because they were central points of the largest constant KH-mapping regions). At the initial stage of the propagation in (a), these *black* patches pass their motion parameters to their immediate neighbors. Each neighbor performs an update (minimization of the metric  $D$ ) of the inherited parameters and propagates the result to its own neighbors. At the end of the process (b), every patch has a set of motion parameters associated with it. The white patches in (b) were part of a non-overlapping region (unique information) and were simply given the nearest set of motion parameters (without any form of update). The noise present in the initial information (*black* patches) has an effect on a complete set of patches related to them and causes the surface to move in blocks (in this case three different blocks).

allows the association of a set of motion parameters to the points that are visible in only one of the views. For this process to give an appropriate result, it is necessary that the information of the selected points be accurate. An error in the matching of only one selected point may cause many other points to be badly matched (“motion in block” described in Figure 4.10). This is why it is so important to select candidates carefully and not hesitate to reject their match if some conditions (such as a threshold on the metric) are not met.

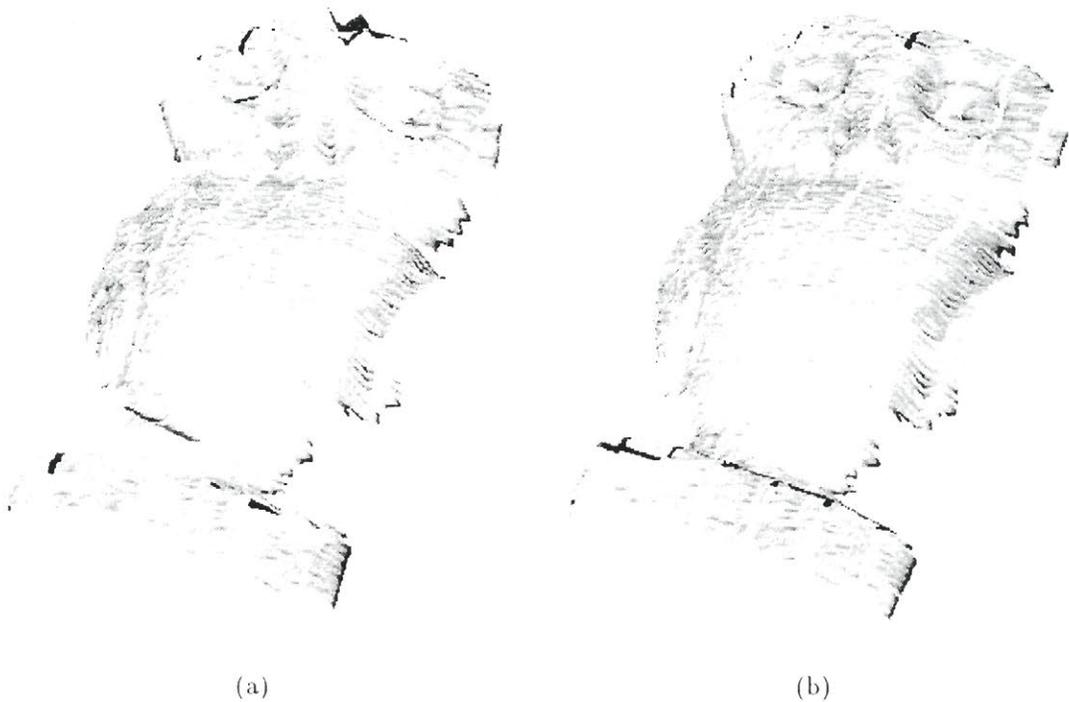
#### 4.4 Motion consistency

In the previous sections, we have seen how to obtain estimates of motion (rotation and displacement) that map all the points of one view to another. It has also been shown that because they are local and because there is noise present in the images, these estimates are usually noisy. In fact, when applied to an image they often lead to errors in reconstruction (see Figure 4.11).

A similar problem was seen in Chapter 3 with the noisy set of estimates of curvature and principal directions. The solution then was to look at the surrounding points and make the curvature field locally consistent.

The same approach is taken with the estimation of motion. The problem starts with a noisy set of motion parameters (translation and rotation at each point). The idea is to use a specialized “smoothing operator” to enforce a constraint of locally rigid motion. In other words, the object should not break as it is mapped from one view to another one. Its deformation, induced by noise or by a real distortion, should vary smoothly over its surface.

The problem can be separated in two parts: translation and rotation. These two parts can be dealt with separately but the separation must be done carefully. It is well known that there are an infinite number of ways to go from one position to another one by composing different rotations and translations. While the final position depends on the amount of rotation and translation applied, the final orientation depends solely on the rotation applied. This tells us how to separate the problem.



**Figure 4.11:** Effect of motion consistency. A view of a small owl statuette is mapped in the frame of a second view (each window of  $7 \times 7$  pixels has its own set of motion parameters). The second view itself is not shown for clarity. (a) shows the result obtained using locally determined motion parameters. (b) shows the same result after the application of 5 iterations of the motion consistency filter. The spatial relationship between the patches is now recovered correctly in (b).

In fact, locally rigid motion means that a point should see his neighbors at almost the same position and with the same relative orientation in the two views. If those conditions are met perfectly everywhere on the surface, we have a rigid body motion. However, if those conditions are met locally only, we have what we call a “locally rigid body motion”. The nature of *local* depends on the constant of rigidity of the body which, in a certain sense, defines the deformability of the object.

#### 4.4.1 Updating the Position

Let us look in detail at a point  $P$  and its  $3 \times 3$  neighborhood consisting of 8 points  $Q_i$ . The final mapping of point  $P$  onto the other view is to be updated using the motion parameters of its neighbors. In view 1, the points  $Q_i$  see point  $P$  at a relative position  $\mathbf{r}_{1i}$ . The position of  $P$  is thus given by

$$\mathbf{p}_1 = \mathbf{q}_{1i} + \mathbf{r}_{1i} \quad (4.7)$$

where  $\mathbf{p}_1$  is a vector representing the position of  $P$  in frame 1,  $\mathbf{q}_{1i}$  is a vector representing position of each of  $P$ 's immediate neighbors and  $\mathbf{r}_{1i}$  is a vector representing the relative displacement between  $P$  and each of the neighbors.

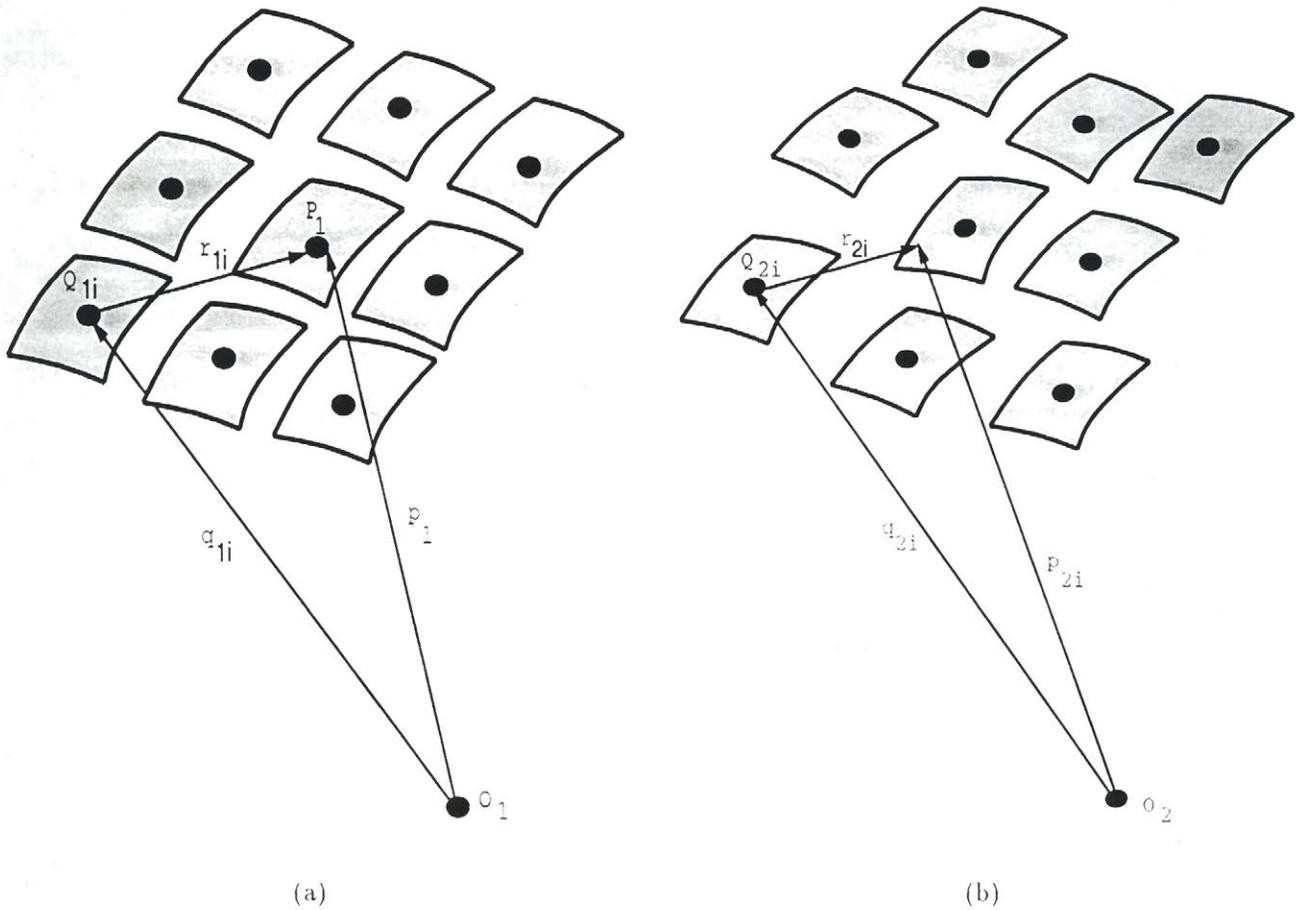
Assuming a locally rigid transformation, a similar description applies to the local neighborhood in frame 2, i.e.,

$$\mathbf{p}_2 = \mathbf{q}_{2i} + \mathbf{r}_{2i} \quad (4.8)$$

using a similar definition for vectors  $\mathbf{p}_2$ ,  $\mathbf{q}_{2i}$  and  $\mathbf{r}_{2i}$  in view 2. A typical representation of the spatial organization of those vectors and points is shown in Figure 4.12.

From equation 4.8, using the motion parameters of a point  $Q_i$  and the constraint of local rigidity, it is then possible to predict where point  $P$  should be in view 2 relative to this neighbor.

First, point  $Q_i$  is mapped onto view 2 using its own motion parameters  $\mathbf{Q}_{1i}^2$  (rotation matrix) and  $\mathbf{O}_1^2$  (displacement vector):



**Figure 4.12:** Principle of motion consistency for displacement. The motion parameters for point  $P$  have to be updated. (a) shows point  $P$  and its neighbors  $Q_{1i}$  in view 1. In (b), we see the same patches mapped in view 2 using their own motion parameters. Because the set of motion parameters for this neighborhood is noisy, the patches are misaligned when mapped in view 2. Vectors  $\mathbf{r}_{2i}$  and  $\mathbf{p}_{2i}$  show where point  $P$  should be when mapped in view 2 according to point  $Q_{2i}$  if we had a rigid-body type motion. Such a prediction is made by every neighbor of point  $P$ .

$$\mathbf{q}_{2i} = \mathbf{Q}_{1i}^2 (\mathbf{q}_{1i} - \mathbf{O}_{1i}^2) \quad (4.9)$$

Then the relative displacement between points  $Q_i$  and  $P$  is expressed in view 2 coordinates is:

$$\mathbf{r}_{2i} = \mathbf{Q}_{1i}^2 \mathbf{r}_{1i}. \quad (4.10)$$

The position of point  $P$  in view 2 as predicted by its neighbor  $Q_i$  is therefore

$$\mathbf{p}_{2i} = \mathbf{Q}_{1i}^2 (\mathbf{q}_{1i} - \mathbf{O}_{1i}^2) + \mathbf{Q}_{1i}^2 \mathbf{r}_{1i} \quad (4.11)$$

where  $\mathbf{Q}_{1i}^2$  and  $\mathbf{O}_{1i}^2$  are the rotation matrix and the displacement vector computed from the motion parameters at point  $Q_i$ .

This last equation, after simplification, simply means that a prediction of the position of point  $P$  in view 2 is found by applying the motion parameters of point  $Q_i$  to  $P$  in view 1,

$$\mathbf{p}_{2i} = \mathbf{Q}_{1i}^2 (\mathbf{p}_1 - \mathbf{O}_{1i}^2). \quad (4.12)$$

A prediction for point  $P$  on view 2 is obtained in this manner from each neighbor. Given the set of predictions and the constraint of local rigidity, the maximum likelihood estimate of  $P$  in view 2 is given by:

$$\mathbf{p}_2 = \frac{\sum_{i=1}^n w_i \mathbf{p}_{2i}}{\sum_{i=1}^n w_i} \quad (4.13)$$

where the  $w_i$  take into account the rigidity of the object and the distance of the neighbor  $Q_i$  to point  $P$ . In other terms, the weights  $w_i$  reflect the object deformability as well as the locality of the rigid-body constraint enforced.

Using Gaussian smoothing, one can determine these weights with the following equation:

$$w_i = e^{-\frac{d^2}{\sigma^2}} \quad (4.14)$$

where  $\sigma$  can be seen as the constant of rigidity of the object and  $d_i$  is the distance of point  $P$  to its neighbor  $Q_i$ . The larger  $\sigma$ , the more the object will be forced to move *rigidly*.

#### 4.4.2 Updating the Orientation

Next we consider orientation. As was the case with the relative displacement, the idea is to maintain a coherence in the *relative* orientation of point  $P$  with respect to its neighbors  $Q_i$  such that the local structure of the surface is preserved when mapped from one view to the other one.

Obviously, if the points  $P$  and  $Q_i$  had exactly the same rotational parameters, as is the case for a rigid body motion, their relative orientation would be preserved when mapped into a second view. However, the estimation of motion is noisy and is likely to vary even over a small neighborhood. These small variations in motion parameters from point to point are a cause of loss of structure. Therefore, trying to obtain a local coherence in the rotational parameters in a neighborhood is equivalent to trying to preserve the relative orientation of those points from view to view.

In fact, what we require is a maximum likelihood estimate of the rotation part of the neighbors' motion parameters. However, in doing so, one has to be careful to avoid a common problem in the representation of rotations, namely non-invariance with respect to the choice of reference frame.

Many different representations for rotations have been proposed. All of them are based on Euler's Theorem of rigid-body rotation: the attitude of a body after having undergone any sequence of rotations is equivalent to a single rotation of that body through an angle  $\theta$  about an axis  $\mathbf{n}$  [19]. Even though they all have the same origin, these different representations (rotation matrices, spinors, quaternions, Euler angles...) are not all equivalent in terms of invariance.

According to Euler's Theorem, one of the most natural ways to express a rotation is the quaternion. A quaternion is composed of a vector (axis of rotation) and a scalar function of the angle of rotation. Therefore, a quaternion expresses a rotation in its simplest form. A summary of quaternion algebra and its basic properties is given in

appendix B.

Because of their view invariant property [19] [18], quaternions are best to represent and deal (e.g. interpolate) with rotation and this is why they will be used to update the orientation of point  $P$ .

Without any loss of generality, consider only the rotation part of the neighbors' ( $Q_i$ ) motion parameters. It is easy to find, from the motion parameters, a quaternion  $\mathbf{q}_2^1$ , expressing the rotation of point  $Q_i$  from view 1 to view 2 (see Appendix B).

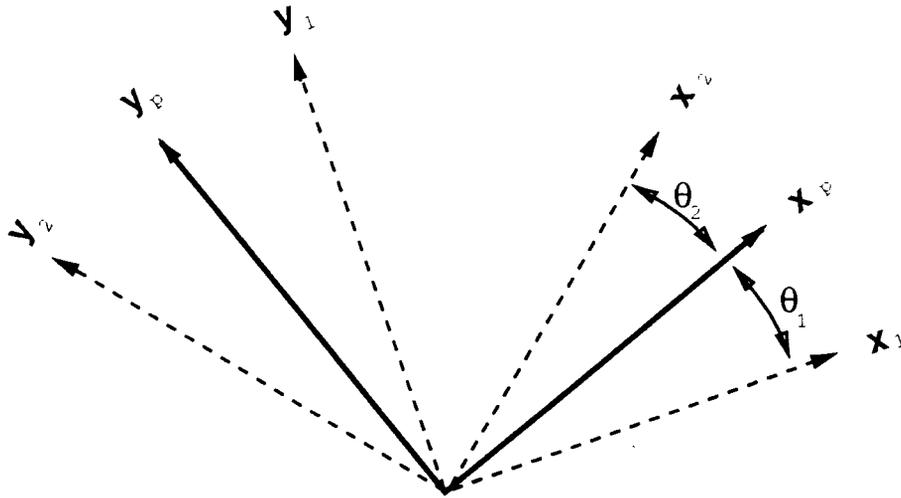
The goal here is to keep consistent the relative orientation between point  $P$  and its neighbors  $Q_i$  in view 1 and view 2. This goal would be achieved automatically if all the points ( $P$  and  $Q_i$ ) had the same rotation component in their own motion parameters. In this case, the patch composed of the whole neighborhood would be rotated by a constant amount and the relative orientation of the points on this patch would be preserved (rigid body motion).

Therefore, to solve our problem, we have to find a quaternion  $\mathbf{q}_{2P}^1$  defined as the minimum *distance* to the other rotations. *Distance* is defined here as the angle of rotation (a scalar) relating two different orientations independently of the axis of rotation as explained by Euler's theorem (see Figure 4.13).

Using the interpretation of quaternions given by the unit sphere (see Appendix B), we can set up a minimization in the following manner.

Let  $\mathbf{q}_{2P}^1$  be the quaternion that is the minimum distance (in the least-squares sense) to the quaternions  $\mathbf{q}_2^1$ , corresponding to the local neighborhood of  $P$ . The *distance* between the orientations expressed by the quaternions  $\mathbf{q}_{2P}^1$  (the solution to our problem) and  $\mathbf{q}_2^1$  (rotation of a point in the neighborhood) is a function of the scalar part of the product  $\bar{\mathbf{q}}_{2P}^1 \mathbf{q}_2^1$ , (where the quaternion  $\bar{\mathbf{q}}_{2P}^1$  is the inverse of  $\mathbf{q}_{2P}^1$  - see Appendix B). This number really gives the difference of rotation applied by the two quaternions since it is nothing but a composition of rotation. It should be noted that the scalar part of the resulting quaternion is  $\cos \frac{\theta}{2}$  where  $\theta$  is the angle of rotation between the two orientations.

If we define the quaternion  $\mathbf{q}_{2P}^1$ :



**Figure 4.13:** Principle of motion consistency for rotation in 2D. In this example, frame  $P$  has the minimum sum of squared distances (angles  $\theta_1$  and  $\theta_2$ ) to the frames 1 and 2. Frame  $P$  can be seen as an average of the two other orientations. The same principle is applied in 3D, using quaternions to obtain the angular displacement between the frames

$$\mathbf{q}_{2_P}^1 = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} \quad (4.15)$$

and the quaternions  $\mathbf{q}_{2_i}^1$ ,

$$\mathbf{q}_{2_i}^1 = \begin{pmatrix} x_i \\ y_i \\ z_i \\ s_i \end{pmatrix}, \quad (4.16)$$

we find that the scalar part of the product  $\bar{\mathbf{q}}_{2_P}^1 \mathbf{q}_{2_i}^1$  is

$$ax_i + by_i + cz_i + ds_i. \quad (4.17)$$

This information can be summed up over the neighborhood of point  $P$ , namely the points  $Q_i$ . A number inversely proportional to the coherence of rotation of  $\mathbf{q}_{2_P}^1$  with all the  $\mathbf{q}_{2_i}^1$  is obtained (sum of  $\cos \frac{\theta_i}{2}$ ). The goal is to maximize this number subject to the constraint of unit length for  $\mathbf{q}_{2_P}^1$ ,

$$a^2 + b^2 + c^2 + d^2 = 1. \quad (4.18)$$

Using a Lagrange multiplier to enforce the constraint, the complete function to be maximized is written as:

$$E = \sum (ax_i + by_i + cz_i + ds_i + \lambda(a^2 + b^2 + c^2 + d^2 - 1)) \quad (4.19)$$

As we did with the update of position (see Section 4.4.1), the terms of this sum can be weighted to take into account the object deformability as well as the distance of the neighbors to point  $P$ . Here again, the factors  $w_i$  defined by equation 4.14 will be used. Therefore, we have the following equation,

$$E = \sum(w_i(ax_i + by_i + cz_i + ds_i) + \lambda(a^2 + b^2 + c^2 + d^2 - 1)). \quad (4.20)$$

Solving for  $a, b, c$  and  $d$ , we obtain the following partial derivatives:

$$\frac{\delta E}{\delta a} = \sum(w_i x_i + 2a\lambda) = 0, \quad (4.21)$$

$$\frac{\delta E}{\delta b} = \sum(w_i y_i + 2b\lambda) = 0, \quad (4.22)$$

$$\frac{\delta E}{\delta c} = \sum(w_i z_i + 2c\lambda) = 0, \quad (4.23)$$

$$\frac{\delta E}{\delta d} = \sum(w_i s_i + 2d\lambda) = 0, \quad (4.24)$$

$$\frac{\delta E}{\delta \lambda} = \sum(a^2 + b^2 + c^2 + d^2 - 1) = 0. \quad (4.25)$$

After simplification we find

$$a = \pm \frac{\sum w_i x_i}{\sqrt{(\sum w_i x_i)^2 + (\sum w_i y_i)^2 + (\sum w_i z_i)^2 + (\sum w_i s_i)^2}}, \quad (4.26)$$

$$b = \pm \frac{\sum w_i y_i}{\sqrt{(\sum w_i x_i)^2 + (\sum w_i y_i)^2 + (\sum w_i z_i)^2 + (\sum w_i s_i)^2}}, \quad (4.27)$$

$$c = \pm \frac{\sum w_i z_i}{\sqrt{(\sum w_i x_i)^2 + (\sum w_i y_i)^2 + (\sum w_i z_i)^2 + (\sum w_i s_i)^2}}, \quad (4.28)$$

$$d = \pm \frac{\sum w_i s_i}{\sqrt{(\sum w_i x_i)^2 + (\sum w_i y_i)^2 + (\sum w_i z_i)^2 + (\sum w_i s_i)^2}}. \quad (4.29)$$

By looking carefully at the two possible solutions (plus or minus signs), we realize that they in fact correspond to two quaternions of opposite directions which represent the same rotation (see appendix B). Therefore, there is only one valid answer. However, it is a good practice to keep the scalar value positive to avoid any confusion caused by an angle of rotation larger than 180 degrees. The sign of the solution is then chosen so that  $d$  is positive.

### 4.4.3 Determining the Updated Motion Parameters

We now have all that is needed to update the motion parameters of point  $P$ . We have found its final position on view 2,  $\mathbf{p}_2$ , and its final orientation given by  $\mathbf{q}_{2P}^1$ .

The mapping of point  $P$  from view 1 to view 2 is represented by the following equation,

$$\mathbf{p}_2 = \mathbf{Q}_{1P}^2 (\mathbf{p}_1 - \mathbf{d}_1), \quad (4.30)$$

where  $\mathbf{Q}_{1P}^2$  is a rotation matrix deduced from the quaternion  $\mathbf{q}_{2P}^1$  and  $\mathbf{d}_1$  is the displacement we are looking for expressed in view 1 coordinates.

Hence,

$$\mathbf{d}_1 = \mathbf{p}_1 - \mathbf{Q}_{2P}^1 \mathbf{p}_2. \quad (4.31)$$

We now have updated all the parameters of point  $P$  needed to map it onto the second view. Such a process is applied to every point or patch in an iterative process.

## 4.5 Controlling the Rigidity of the Motion

Now that most of the proposed method to find correspondence of views has been described, it is appropriate to enumerate and describe the different *knobs* allowing control of the non-rigidity of motion.

Theses parameters are:

1. **The size of the patches.** It should be adapted to the object constant of rigidity ( $\sigma$ ). The bigger is  $\sigma$  the larger the patches can be. In a sense, one could always use small subwindows. This option simply means more computation in the case of an almost rigid motion.
2. **Motion consistency.** Three parameters can be defined in relation with this algorithm:

**Number of iterations.** The number of iterations should be proportional

to the rigidity of the motion and to the noise. At convergence (after a large number of iterations), motion consistency would produce a globally rigid motion. It is therefore important to apply it for only a few iterations, just enough to remove most of the noise, if the motion is known to be non-rigid.

**Window size.** The effect of the size of the neighborhood is very similar to the number of iterations. A larger neighborhood implies a more rigid motion.

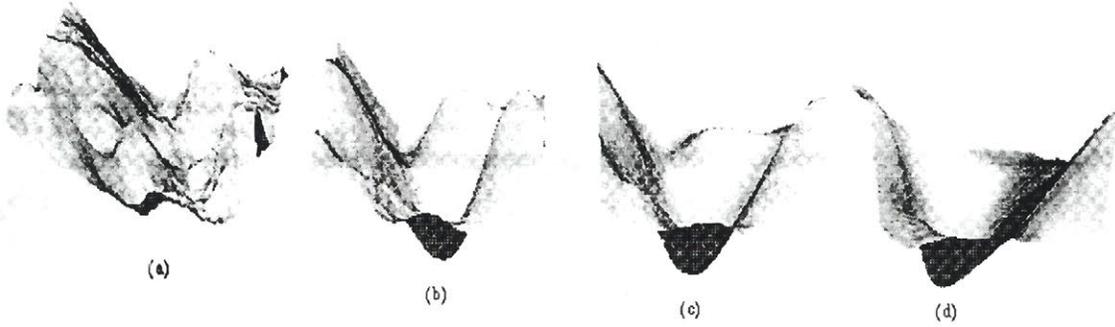
**Weighting factors ( $w_i$ ).** The weighting factors  $w_i$  as defined by equation 4.14 take into account the distance of a neighbor and the constant of rigidity of the object. Obviously, for a non-rigid motion, these weights and the constant of rigidity should be very small and set to higher values for a more constrained motion.

Together these parameters allow for a flexible control of the motion type. Unfortunately, they also express the need to determine many different constants. However, we have observed that, for the case of rigid objects, 3 iterations of motion consistency with a neighborhood of  $3 \times 3$  and a constant of rigidity of 1 for patches of  $5 \times 5$  pixels give good results (the spatial resolution of our test images was of about 1 sample per mm). Slight variations of these numbers do not appear to affect the final result.

## 4.6 Relation Between this Method and Continuation Methods

It is possible to establish a relationship between the method presented in this thesis and the classical continuation methods such as the GNC (Graduated Non-Convexity) method of Blake and Zisserman.

The idea behind the GNC is to turn a non-convex error function into a convex one by changing a control parameter. When the function to be minimized is convex, its solution (the minimum) does not correspond exactly to the original problem but is relatively near. As the function is brought back to the non-convex shape representative of the original problem, the minimum can be tracked and the global solution can be found.

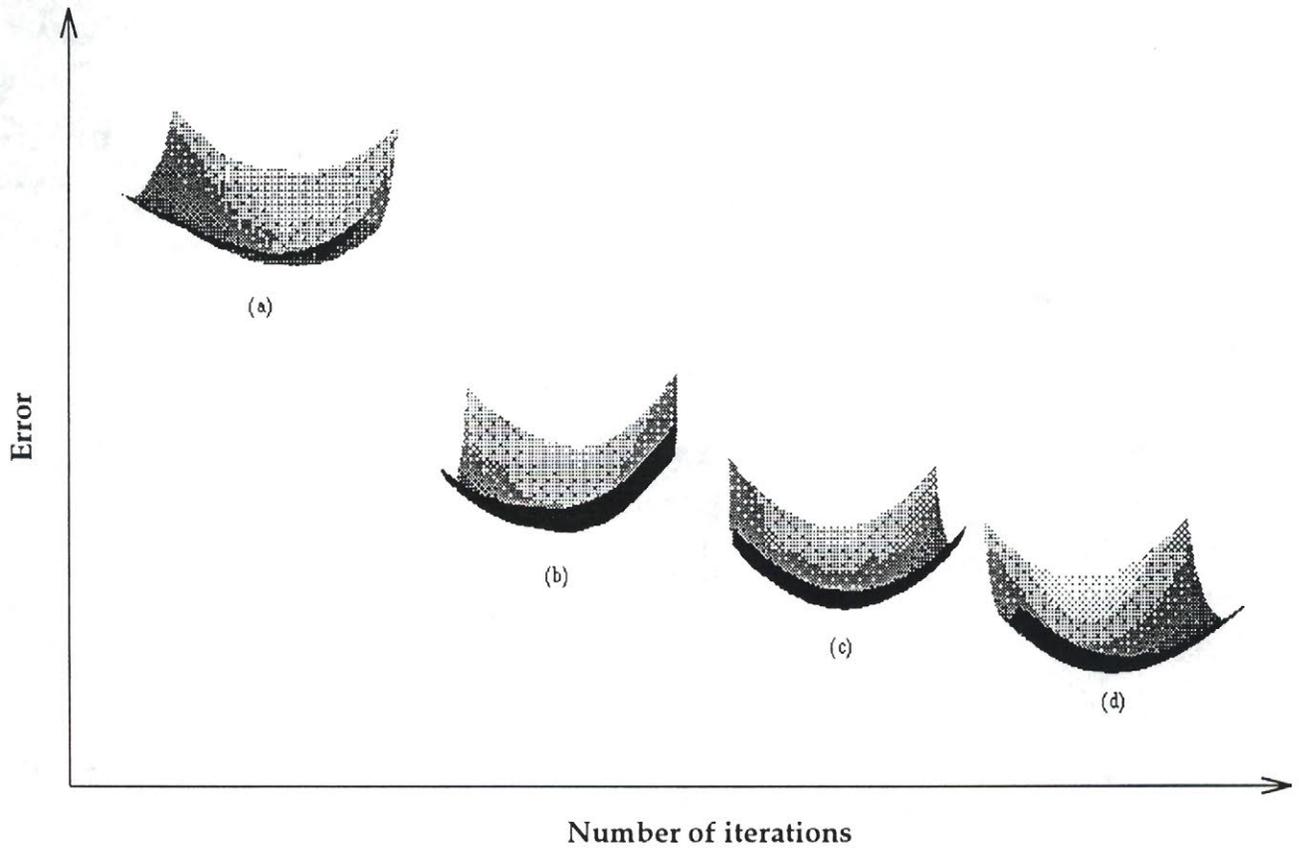


**Figure 4.14:** 2D slices of the surface produced with the metric  $D$  after different stages of curvature consistency. In this example, the displacement in  $x$  and in  $y$  are changed while the other motion parameters are kept constant. These surfaces are obtained for two views of a small owl statue presented in next chapter. (a) is after 1 iteration of curvature consistency, (b) is after 2, (c) is after 3 and (d) is after 5 iterations. The surface becomes clearly convex as more iterations are performed.

The same idea is applicable to our problem. We have seen in Section 4.3.5 the effect of the number of iterations of curvature consistency on the convexity of the metric  $D$ . Figures 4.14 and 4.15 show the same principle for different numbers of iterations of curvature consistency. The convexity of our metric is clearly controlled by the number of iterations of curvature consistency.

However, we face the same problem as in the GNC method, namely that the convex problem (after many iterations) is not exactly our original problem. It is possible to apply the same strategy as is done with the GNC: gradually reduce the number of iterations and go back to the original problem while tracking the minimum. For example, refer to Figure 4.15. Having found the minimum in (d), it is possible to determine the corresponding minima in Figure 4.15 (c) through (a). However, one should be careful doing so.

The estimation of the Darboux frames is very noisy before the application of curvature consistency and the metric  $D$  is corrupted and influenced by this factor. In this sense, it would be a bad idea to minimize the metric  $D$  before doing at least few iterations of curvature consistency because it would be difficult to infer the structure of the surface. Therefore, it is preferable to always perform a minimum of two or three iterations before attempting to determine correspondence.



**Figure 4.15:** 2D slices of the surface produced with the metric  $D$  after different stages of curvature consistency. In this example, the rotations in  $\theta_x$  and in  $\theta_y$  are changed while the other motion parameters are kept constant. These surfaces are obtained for two views of a small owl statue presented in next chapter. (a) is after 1 iteration of curvature consistency, (b) is after 2, (c) is after 3 and (d) is after 5 iterations. The surface also becomes more smooth and convex as the number of iterations is increased, but the most remarkable characteristic is the relative reduction of the error.

It is therefore possible to relate the GNC principle to our method. However, the previous practical considerations put some constraints on its application. The relationship between our method and classical continuation methods is nevertheless direct and straightforward.

### 4.7 Summary

In this chapter, we have presented and characterized a method to find the precise correspondence of views given an estimate of the motion parameters relating them. The method is local and allows for a distortion of the object from one view to another one since the mapping is done on a local basis.

We have also seen the limits of the method and the potential problems inherent to its application. The possibility of multiple solutions in the case of a noisy or highly textured surface has been examined and the necessity for reconstruction prior to the cross-comparison has been demonstrated.

To cope with noise inherent to local estimation, a new specialized filter has then been presented, motion consistency. We have shown that this filter enforces a coherent motion of the different parts of an object. It also expresses concretely the fact that an object should not break as it moves (even though it can change its shape).

An analogy of this method with classical continuation methods has finally been proposed. It shows an obvious link: the parameter controlling the convexity being the number of iterations of curvature consistency. Practical considerations limit the extent of the analogy.

In the next chapter, different examples of application of this method will be shown with synthetic and real range images.

## 5.1 Introduction

In this chapter, the results of experiments with real and artificial data are presented. The goal is to show the ability of the method to recover the motion parameters relating two views of an object in different circumstances.

Artificial range images will be divided in two categories: the ones for which the Darboux frames have been estimated using curvature consistency and the ones for which the Darboux frames have been found analytically. It will be seen that curvature consistency has a large effect on the precision of the recovered motion parameters.

The data acquisition process for synthetic and real surfaces will be described shortly in the following section. Even though these details are more at the technical level, they nevertheless determine the accuracy of the results and constrain the limit of what is achievable in practice with current hardware. They also help to understand concretely the problems faced when one tries to fuse different views of a single object.

The following section will present results obtained with synthetic data. It will be shown how the noise and the precision of the Darboux frame estimates can influence the method. Surfaces with repetitive structure (see Figure 4.4) are also examined to see what kind of result can be obtained in this case.

Results obtained with *real* range images will be presented in a following section. It will be seen that the estimate of motion obtained from the calibration of a robot moving a camera needs to be updated in order to get an acceptable registration of two views.

## 5.2 Data Acquisition

In this section, the set-up used for the acquisition of the data is described in more detail. In fact, three types of set-up will be presented (one of them being a virtual

one for the synthetic images). It will also be necessary to present a few technical details about the specific range finder used in these experiments.

### 5.2.1 Range Finder

The range finder used for the present experiments is the result of a collaboration between McGill University and the NRCC (National Research Council of Canada). It is a 2-axis scanner capable of scanning a complete surface from a fixed view point. The basic principle is explained in Figure 5.1.

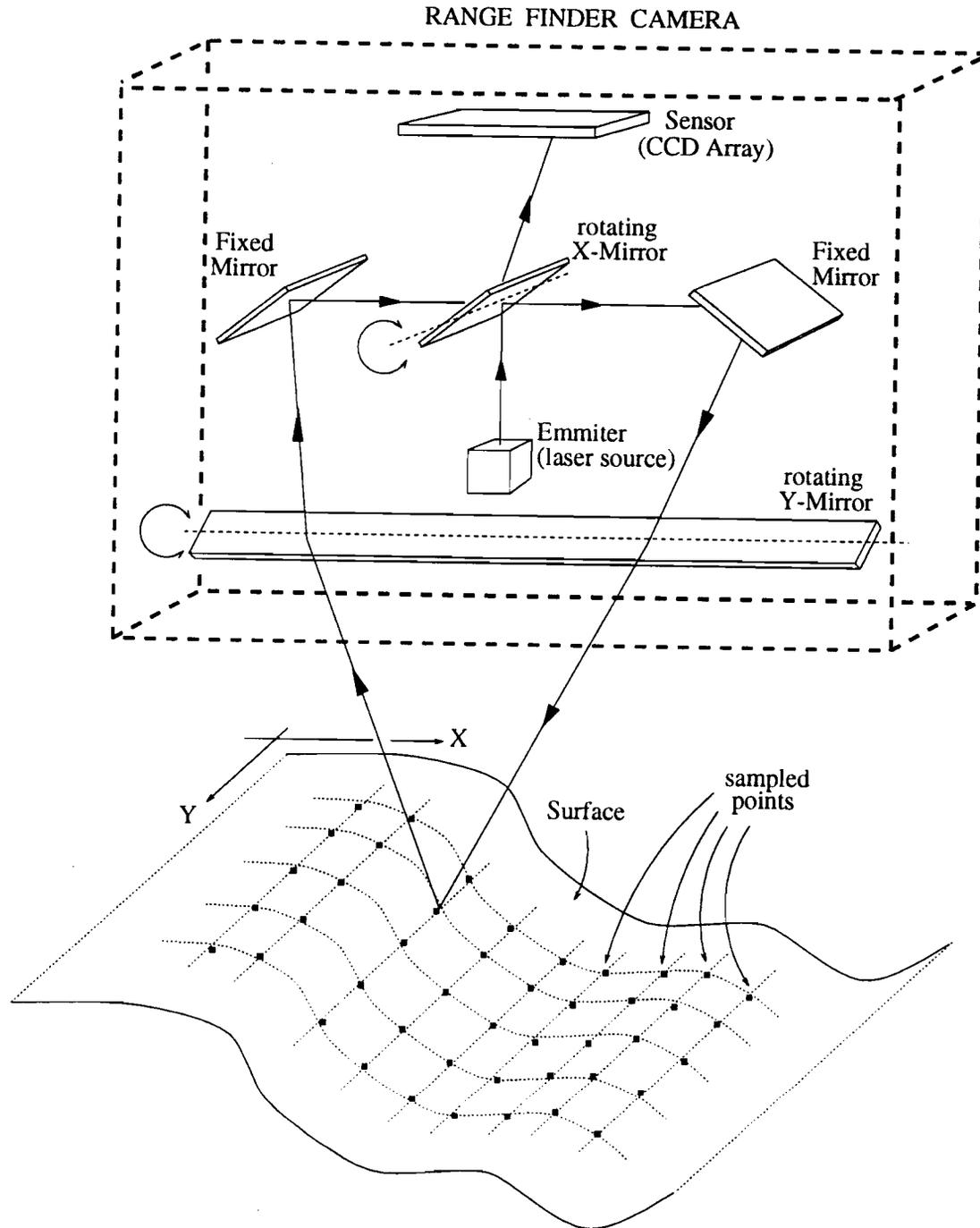
The range of this scanner is of approximately one meter with a field of view of 40 degrees in  $x$  and 28 degrees in  $y$ . Each of these directions ( $x$  and  $y$ ) can be divided into a maximum of 256 positions. The precision is in the order of 1 mm at 1 meter of distance and improves in a non-linear fashion as the distance is reduced. A good technical description of the construction and the principle of this scanner can be found in [11].

It is possible to obtain calibrated data (real triplets  $x, y$  and  $z$  in mm for each point) in terms of the camera coordinate frame. For this purpose, a calibration procedure has to be applied carefully. The result of the calibration is a set of look-up tables and few simple geometric equations relating the indices  $i$  and  $j$  of an image to their physical coordinates ( $x$  and  $y$  in mm in the camera coordinate frame).

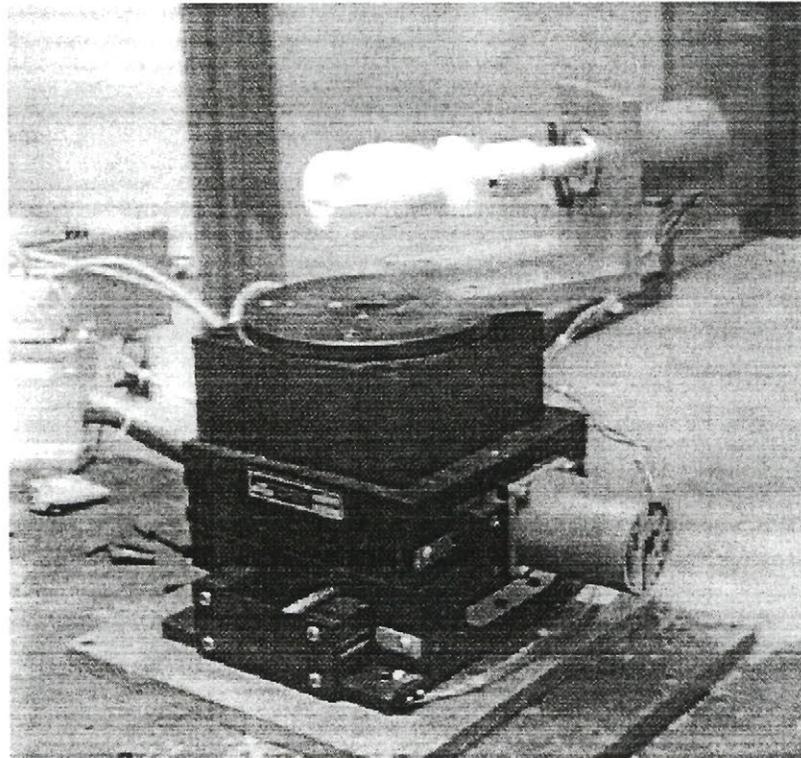
In addition, because of its principle of triangulation, this scanner is sometimes unable to read a point position because the light is occluded by an obstacle (shadow effect). This produces a shadow or a *hole* in the information from the surface. For a better discussion about shadow and other type of problems related to this scanner, see [7].

### 5.2.2 Precision Stages

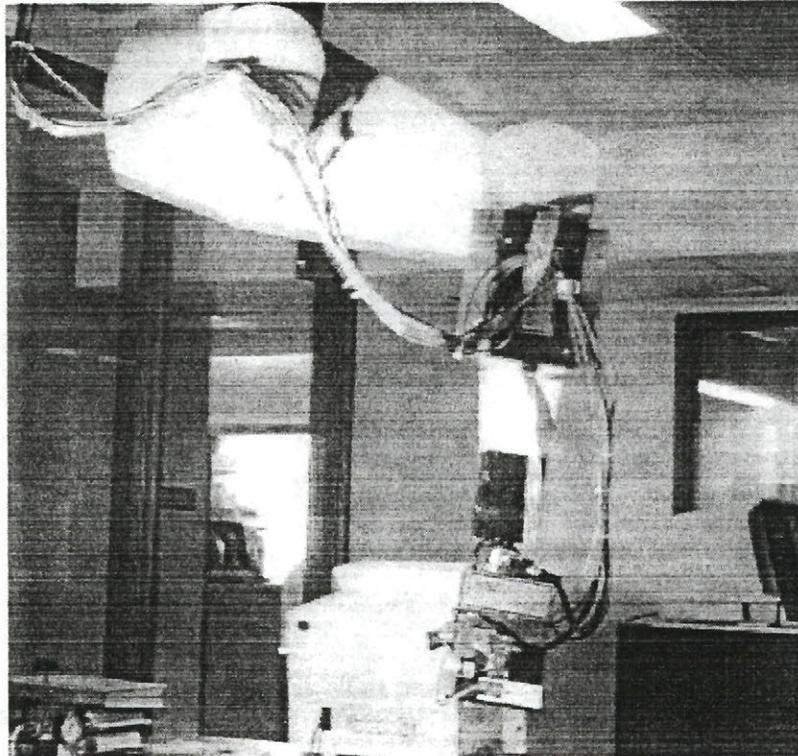
A set of precision stages controlled by stepper motors is used to precisely move an object in front of the range finder. This allows for four degrees of freedom (2 rotations and 2 translations). This set-up is shown in Figure 5.2.



**Figure 5.1:** Schematic of the range finder used. A laser beam is emitted and hits a first mirror (x-mirror) which is reflective on both sides. This mirror controls the  $x$  position of the beam on the surface after its reflection on two other mirrors (fixed mirror and y-mirror). From the surface, the light is diffused and a part of it goes back to the *detector* through a series of mirrors (y-mirror, fixed mirror and the other side of the x-mirror). The *detector* is in fact a CCD array where the displacement of the beam is proportional to the distance of the surface (triangulation based technique). By changing the angles of the x-mirror and y-mirror, it is possible to scan a 2 dimensional patch of the surface.



**Figure 5.2:** Precision stages used in the data acquisition process. These stages were used to precisely move the objects in front of the range finder. The combination of stages allows for four degrees of freedom (2 displacements and 2 rotations).



**Figure 5.3:** Robot (Puma 560) hanging from the ceiling and used to move the range finder around the scene. The motion parameters of the robot can be used to determine an estimate of the motion parameters between the views.

The theoretical precision obtained is about 79 steps per mm in displacement in  $x$  and in  $y$ , 100 steps per degree for the rotation about the  $z$ -axis and 0.56 step per degree for the rotation about the  $x$ -axis. However, the real precision is a bit lower if one takes into account the mechanical play in the gears (backlash).

In this set-up, multiple views of an object are obtained by keeping the camera fixed and by moving the object with the stages in front of the camera.

### 5.2.3 Scanner Moved by a Robot

A robot (Puma 560) mounted on the ceiling in an inverted configuration was also used to move the camera around a scene and collect data at different viewpoints. This set-up is shown in Figure 5.3.

The precision obtained in terms of motion parameters relating two views can vary considerably depending on the technique used for estimation. A first technique would

be to match by hand three (or more) points from the two views and deduce the motion parameters. This technique gives acceptable results but it requires intervention. It has, however, the advantage of being independent of the precision of the robot used (moreover, it is not necessary to use a robot to move the camera in this case).

Another technique is to calibrate the camera with respect to the end-effector of the robot. This calibration is a rather complicated process involving many views of a scene made of easily recognizable targets that can be matched across views. The advantage here is an *automated* process that needs only the robot positions to determine the relationship between two images. However, the precision obtained is not spectacular. It depends on both the precision of the robot and the quality of the calibration. An error of a few (5 to 10) millimeters is typical with the Puma 560.

### 5.2.4 Virtual Set-Up for Synthetic Data

In the case of synthetic surfaces, a *ray-tracing* technique has been applied to get different viewpoints from a computer-generated surface. It provides a very general method to scan any kind of surface and even allows simulation of the function of the range finder.

In this *virtual* set-up, the surface to be scanned is traced in space. The camera position and orientation relative to the surface are then given to the process as a parameter. For each point of the image, a ray starting at position  $(i, j)$  in the camera frame and ending at the surface is traced. The length of this ray gives the depth of the surface at this position.

The simulation of the range finder is done by adapting the direction of the rays traced accordingly with the calibration tables.

## 5.3 Results with Synthetic Data

The results presented in this section have been obtained with two views of synthetic surfaces. Because these surfaces are artificial and the process to *scan* them is purely at the software level, the motion parameters can be estimated perfectly and used to

validate our method.

The first two tests are done with a paraboloid and an hyperboloid. These noiseless figures represent two perfect constant KH-mapping surfaces which is one of the conditions discussed earlier in Section 4.3.3. The results obtained in these cases should be very accurate and will provide a minimal demonstration of the validity of the method.

Another series of tests is then performed on noisy paraboloids. The interest here is to determine the maximum noise for which the method gives an acceptable result.

A final test is made with a paraboloid to which a regular texture has been added. This demonstrates the limitation of the method in cases where the surface includes repetitive structure. The precision required for the estimation of the motion parameters is determined.

### 5.3.1 Noiseless Paraboloid

The equation of the paraboloid used in this case is

$$z = 0.01x^2 + 0.005y^2. \quad (5.1)$$

The two images produced from it were  $100 \times 100$  pixels. The shape of this surface is shown at Figure 5.4 from a general viewpoint.

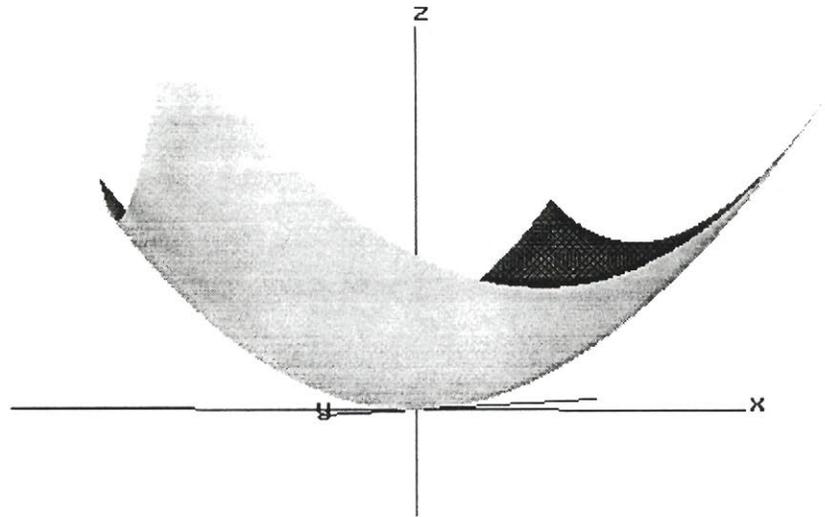
For the first view, the camera motion parameters relative to the surface were

$$\Theta_1 = \{-50, -50, -100, 0, 0, 0\} \quad (5.2)$$

where the three first numbers are the displacements  $x, y$  and  $z$  and the three last numbers express the rotation in Euler angles form ( $\theta_x, \theta_y$  and  $\theta_z$  in degrees). These six numbers give the position of the camera of the first view expressed in the surface coordinate frame. The second view has been *scanned* by placing the camera at

$$\Theta_2 = \{-25, -35, -125, 20, 10, 45\}. \quad (5.3)$$

From these two positions, it is possible to determine the *relative* position of the



**Figure 5.4:** General view of the noiseless paraboloid used to generate two views for a test.

two views:

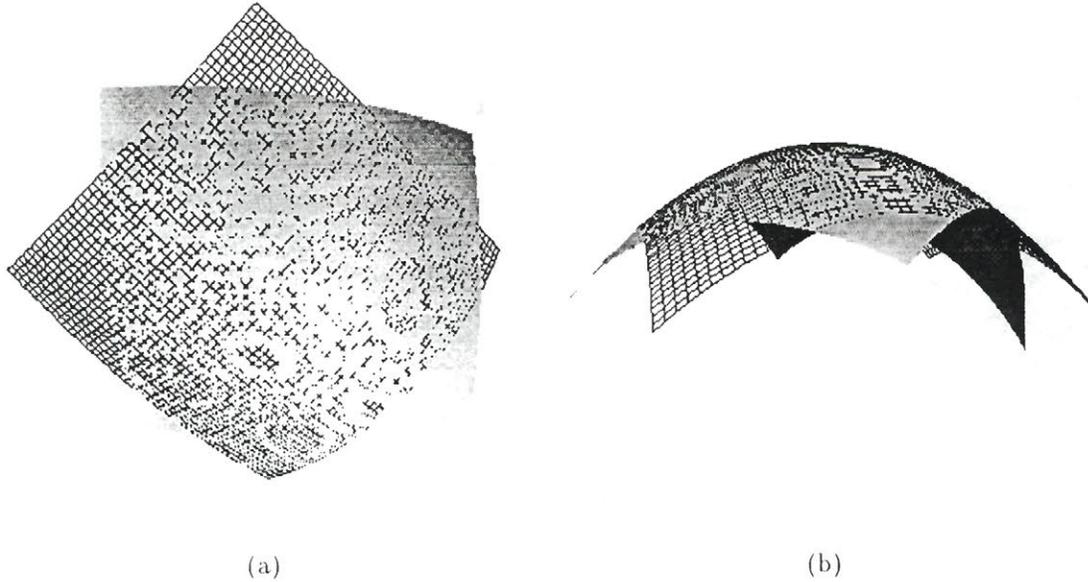
$$\Theta_r = \{25, 15, -25, 20, 10, 45\} \quad (5.4)$$

expressing the position<sup>1</sup> of the second view in the coordinate frame of the first view. The precision of these numbers is very good as is shown clearly in Figure 5.5. They will therefore be used as a basis of comparison with the results obtained by our method.

### Finding Correspondence Using Analytic Darboux Frames

For the first test, the Darboux frames at each point of the two views have been found analytically. It is possible to do so because a synthetic surface with a known mathematical function is used. In the general case however, it will not be possible to do so and the curvature consistency algorithm will have to be used for this purpose. The Darboux frames estimated using the curvature consistency algorithm tend to be less precise and can lead to an imprecision in the motion parameters recovered (due to smoothing).

<sup>1</sup>Here, the term *position* is used to designate the full set of motion parameters including the rotational part.



**Figure 5.5:** Correspondence of two views of the artificial paraboloid shown in figure 5.4. The estimation of motion ( $\Theta_r$ ) is used to put the views together. One of the views is shown as a shaded surface and the other is shown as a grid. The interlace of the grid with the shaded surface shows the accuracy and the precision of the motion parameters.

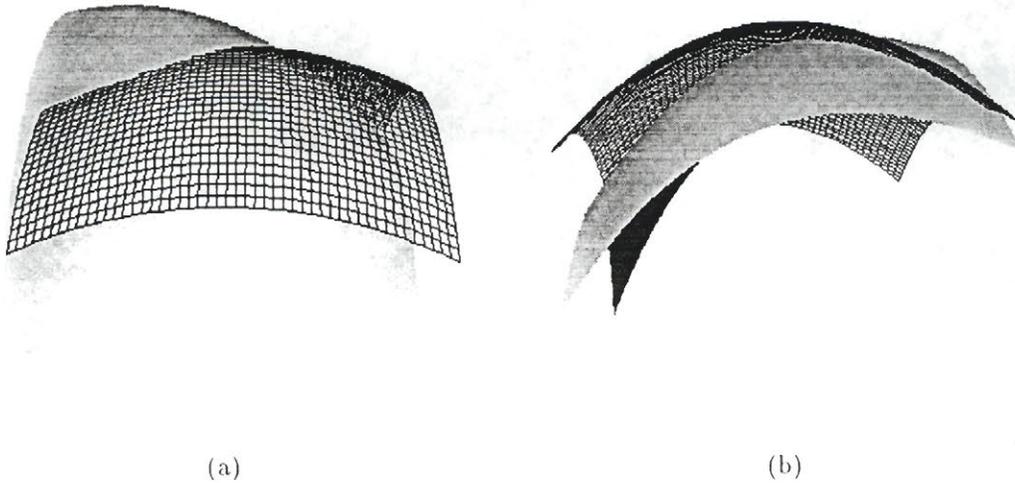
For the purpose of the test, the two surfaces have been misaligned by perturbing their known motion parameters ( $\Theta_r$ ). The *initial* motion parameters used as a starting point of the view correspondence process were:

$$\Theta_f = \{85, 5, -15, 0, 5, 90\} \quad (5.5)$$

which is better visualized with the help of Figure 5.6.

In this case, we have a surface of constant KH-mapping (convex figure) and, for this reason, there would be only one point selected for the initial minimization (see previous chapter). However, only for this example, the implementation uses 3 fixed points for the initial process of matching.

The motion parameters obtained for these three points after the initial step of minimization are:



**Figure 5.6:** Starting point of the view correspondence process for two views of the noiseless paraboloid shown in figure 5.4. The misalignment of the views is clearly visible.

$$\Theta_1 = \{24.573, 15.760, -25.249, 20.018, 9.993, 45.024\}$$

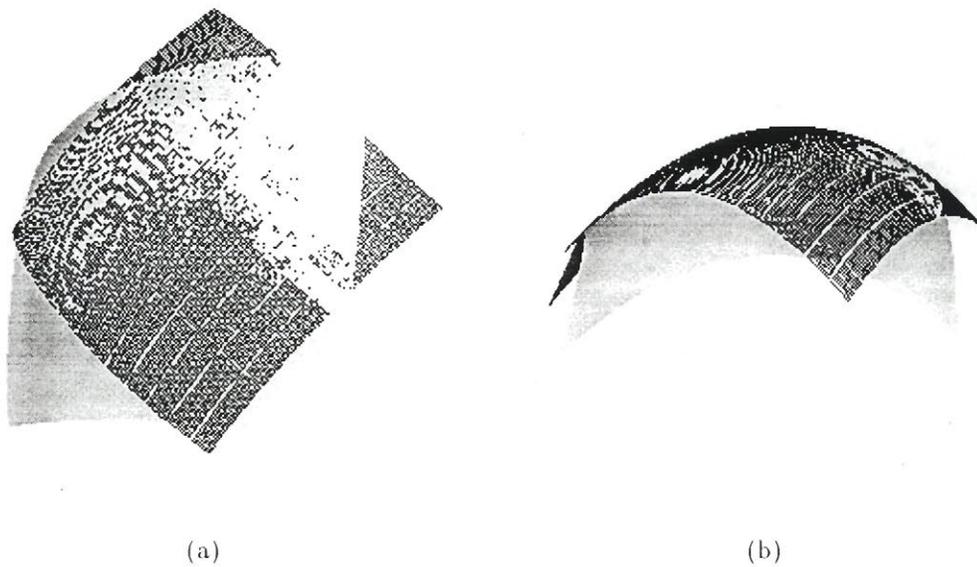
$$\Theta_2 = \{25.439, 15.741, -24.972, 20.007, 10.065, 44.960\}$$

$$\Theta_3 = \{24.792, 16.073, -24.860, 20.022, 10.024, 44.984\}.$$

The final result of the correspondence (including motion consistency) is shown in Figure 5.7. This represents an upper bound on the precision of the algorithm.

It is necessary to briefly comment the precision of the motion parameters  $\Theta_1$ ,  $\Theta_2$  and  $\Theta_3$  compared to the theoretical ones ( $\Theta_r$ ). It can be seen that these numbers are very similar in terms of rotation components but are slightly less precise in the case of the displacement components.

Even though the precision obtained for  $\Theta_1$ ,  $\Theta_2$  and  $\Theta_3$  is quite good, one could have expected better results since we have a noiseless figure and a set of analytically determined Darboux frames (the best of both worlds). The problem is that we have a discretized surface; interpolation is required to obtain a better precision. For this purpose, a bilinear interpolation is used (see Figure 4.8). It does a reasonable job



**Figure 5.7:** Correspondence of views found for the noiseless paraboloid using the starting point shown in figure 5.6. Again one of the views is shown as a shaded surface and the other one is shown as a grid. In this case however, the grid is composed of a set of independent  $3 \times 3$  patches. The quality of match can again be evaluated by looking at the interlace of the shaded surface with the grid.

but it is not perfect (as the numbers obtained tell). A better solution would be to apply the differential geometry concept of “parallel transport” [4] in interpolating the surface in between the samples. A better interpolated value would be so obtained allowing for a more precise determination of the motion parameters. This idea has not been implemented because it was felt that the gain in precision was too small (since we already have a dense representation) for the additional computational load.

The previous test has been repeated without interpolation (using the nearest point). The following results were obtained:

$$\Theta_1 n = \{26.222, 14.510, -25.081, 19.847, 9.719, 45.297\}$$

$$\Theta_2 n = \{24.783, 14.987, -24.968, 20.007, 10.063, 44.970\}$$

$$\Theta_3 n = \{24.778, 13.852, -24.747, 19.693, 9.976, 44.875\}$$

It can be seen that while being less precise, these results compare quite well with  $\Theta_r$  (the theoretical result). Therefore, interpolation does not yield a big improvement in precision. It does however make a difference in the convexity of the error function (metric  $D$ ) which is made easier to minimize (see Figure 4.9). This is why a bilinear interpolation will be performed on each of the subsequent tests.

### Finding Correspondence Using Estimated Darboux Frames

The idea here is to see at which precision the motion parameters can be recovered if we have only an *estimate* of the Darboux frames. For this purpose, the curvature consistency algorithm has been applied to both views. It should be recalled that curvature consistency estimates the Darboux frames by fitting paraboloids on sub-windows of the surface. This can lead to a noisy estimate because 1) the model is only approximate<sup>2</sup> and 2) because of quantization effects.

However, as more iterations of curvature consistency are performed, the noise induced by the estimation stage tends to smooth out and the motion parameters can

---

<sup>2</sup>Even a paraboloid cannot be represented perfectly by a set of overlapping paraboloids.

be recovered more precisely. This is shown in table 5.1 where the motion parameters have been found for three different points after different numbers of iterations of curvature consistency. From this table, it can be seen that the result becomes progressively more stable as more iterations are performed.

Another interesting result should be noticed in relation to this experiment. Even though the motion parameters obtained from the correspondence algorithm (before motion consistency) seem to be very noisy after few iterations of *curvature consistency*, they lead anyway to an acceptable result if a large number (e.g. 100) of iterations of *motion consistency* are applied. For this case, the effect of the quantization noise can be compensated with different combinations of curvature or motion consistency.

One should however be careful in interpreting this affirmation. In this case we have a noiseless paraboloid and the quantization noise is relatively low (low enough to allow the correspondence algorithm to work adequately). For higher levels of noise, the motion parameters estimated might be too far of the real answer and lead to a deformed surface even after the motion consistency filter. For this reason, it is always preferable to apply at least few (3) iterations of curvature consistency before trying to find the correspondence of views.

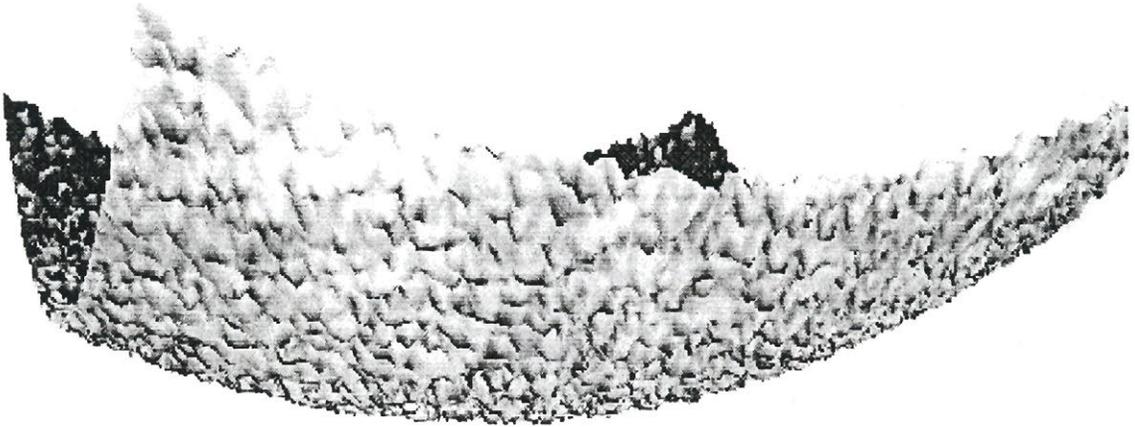
### 5.3.2 Noisy Paraboloid

For this experiment, the two views of the paraboloid have been corrupted with 5% Gaussian noise (different noise on each view). Figure 5.8 gives an idea of the amount of noise added. The goal here is to see up to what precision we can recover the motion parameters.

The first step is to try to get a good estimate of the local structure of the surface. In other words, we have to find the underlying smooth surface and estimate its differential properties. For this purpose, a first trial is realized using 20 iterations of curvature consistency with a  $5 \times 5$  window. The result obtained is showed in table 5.2. It turns out that using a larger window size for the correspondence algorithm does not much improve the result if the local structure of the surface has not been

Iteration number	$x$	$y$	$z$	$\theta_x$	$\theta_y$	$\theta_z$
0	23.038	7.025	-23.479	17.894	10.406	44.687
	28.195	5.941	-23.740	17.486	8.789	45.085
	25.975	31.343	-26.863	23.968	10.281	46.402
1	23.711	19.334	-25.615	20.929	10.406	45.171
	19.580	5.364	-22.331	17.398	11.764	44.843
	25.139	16.116	-24.888	20.023	9.937	45.109
2	25.086	18.029	-25.381	20.723	10.169	45.232
	28.128	19.151	-25.992	20.808	9.004	45.602
	25.399	16.466	-25.016	20.150	9.958	45.313
3	25.236	17.533	-25.375	20.493	10.088	45.218
	26.135	17.474	-25.519	20.422	9.516	45.308
	25.352	16.533	-25.037	20.186	9.945	45.264
4	25.461	17.295	-25.412	20.376	9.976	45.215
	24.554	10.261	-23.928	18.537	10.263	45.007
	25.295	16.610	-25.060	20.220	9.978	45.267
5	25.532	16.905	-25.392	20.193	9.917	45.167
	25.206	11.480	-24.443	19.080	9.957	45.076
	25.344	16.599	-25.058	20.214	9.957	45.273
6	25.655	16.715	-25.405	20.101	9.851	45.156
	25.901	12.369	-24.497	19.140	9.918	45.039
	25.334	16.608	-25.061	20.222	9.937	45.242
7	25.043	15.763	-25.378	20.014	9.824	45.156
	25.613	13.590	-24.733	19.343	9.698	45.091
	25.335	16.637	-25.071	20.235	9.937	45.242
8	25.144	15.601	-25.388	19.937	9.774	45.156
	25.511	14.050	-24.846	19.556	9.726	45.052
	25.343	16.612	-25.063	20.225	9.924	45.232
9	26.066	14.648	-25.056	19.898	9.723	45.148
	25.530	14.101	-24.865	19.578	9.704	45.036
	25.375	16.600	-25.060	20.220	9.906	45.230
10	26.196	14.563	-25.080	19.856	9.668	45.167
	25.473	14.110	-24.856	19.583	9.726	45.023
	25.349	16.593	-25.057	20.218	9.914	45.222

**Table 5.1:** Motion parameters recovered for three points of the noiseless paraboloid shown in Figure 5.4. Windows of  $5 \times 5$  have been used by the correspondence algorithm as well as by curvature consistency. The table shows the evolution of the results as more iterations of curvature consistency are performed. In theory, the motion parameters  $\Theta_r = \{25, 15, -25, 20, 10, 45\}$  should be obtained for each of the three points.



**Figure 5.8:** Paraboloid shown in Figure 5.4 corrupted with 5% of Gaussian noise. A similar amount of noise have been added to the other view of the paraboloid.

Window size	$x$	$y$	$z$	$\theta_x$	$\theta_y$	$\theta_z$
5	40.328	5.597	-22.230	15.639	17.388	60.615
7	56.293	13.971	-24.794	16.610	12.833	64.655
9	41.236	11.982	-24.392	17.456	17.935	61.549
11	40.574	13.201	-25.481	17.609	17.656	61.527

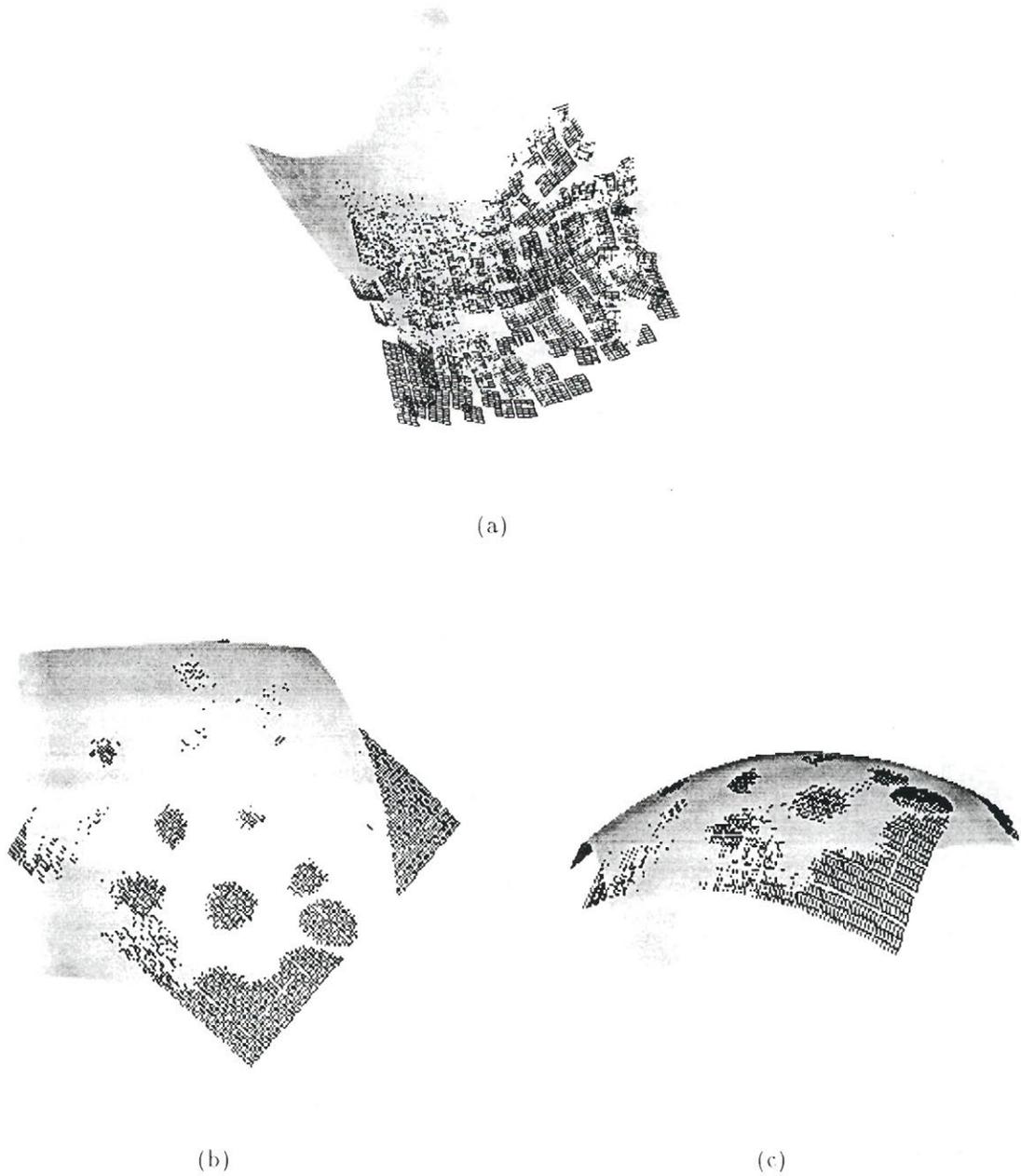
**Table 5.2:** Global set of motion parameters recovered for various window size. It can be observed that using a larger window at this stage does not improve the result much if the local structure of the surface has not been recovered properly.

properly recovered with curvature consistency.

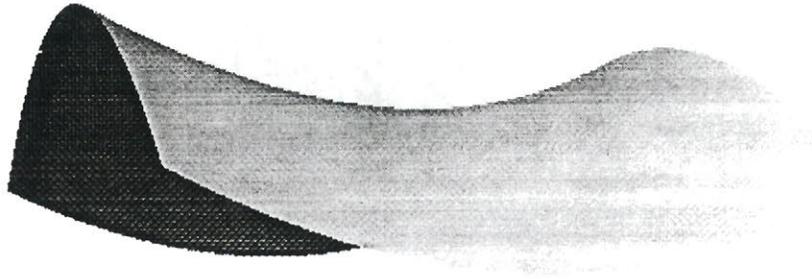
A second experiment has been performed using a window of  $11 \times 11$  for curvature consistency. The global motion parameters recovered after 12 iterations of curvature consistency and using a window of  $5 \times 5$  for the correspondence algorithm is this time quite acceptable:

$$\Theta_N = \{24.504, 15.465, -25.309, 20.215, 9.754, 44.060\} \quad (5.6)$$

The registration of the resulting smooth surfaces is shown in Figure 5.9 where the quality of the fit can again be judged by the interlace of the two views.



**Figure 5.9:** Registration of two noisy views of a paraboloid. (a) is after 0 iteration of motion consistency (estimation stage only) and (b) and (c) are after 100 iterations (top and side views). Even though the estimation of motion parameters at stage (a) is poor (disconnected patches), the motion consistency filter can recover an adequate registration of the surface. Because of the noise added to them, the surfaces do not touch each other everywhere but a good compromise has been found.



**Figure 5.10:** General view of the noiseless hyperboloid used to generate two views for a test

### 5.3.3 Noiseless Hyperboloid

The goal of this test with synthetic data is to show that the method presented here works with hyperbolic patches as well as it does for parabolic patches. For this reason, only one test will be performed with two noiseless views of an hyperboloid whose Darboux frames have been estimated and refined with 10 iterations of curvature consistency using a  $5 \times 5$  window.

The equation of the hyperboloid used was

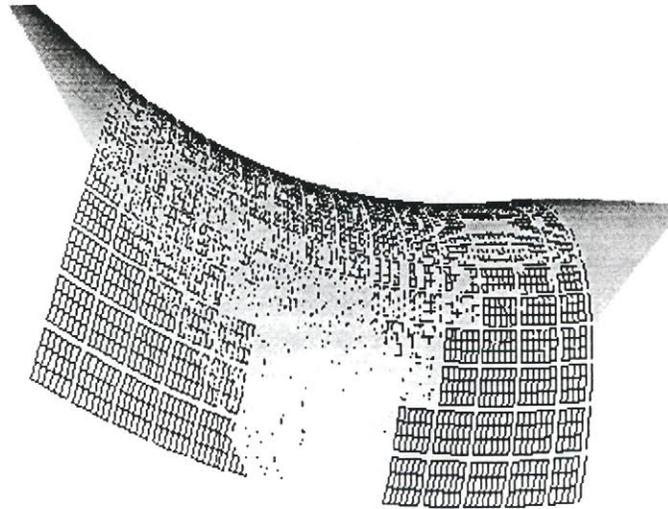
$$z = 0.01x^2 - 0.005y^2. \quad (5.7)$$

The two images produced with it were  $100 \times 100$  pixels. The shape of this surface is shown at Figure 5.10 from a general viewpoint.

The two views were taken at the same positions used for the paraboloid and, therefore, the theoretical relative motion parameters between these views is also

$$\Theta_r = \{25, 15, -25, 20, 10, 45\}. \quad (5.8)$$

The correspondence algorithm, started at  $\Theta_f$  defined for the paraboloid case, was applied to three points and the following motion parameters were recovered (before



**Figure 5.11:** Registration of the two views of the synthetic hyperboloid. Again, the interlace of the surfaces shows the quality of the fit.

motion consistency):

$$\Theta_1 = \{25.812, 15.758, -24.985, 20.044, 9.746, 45.046\}$$

$$\Theta_2 = \{25.703, 15.753, -25.023, 20.007, 9.960, 45.015\}$$

$$\Theta_3 = \{25.213, 15.354, -24.947, 19.824, 9.885, 44.996\}$$

The registration of the views is shown in Figure 5.11.

### 5.3.4 Paraboloid with a Periodic Structure

This test demonstrates the sensitivity of the method to periodic patterns or structures. For this purpose, a paraboloid to which a sinusoid has been added has been generated. The size of this image is  $100 \times 100$  and the period of the sinusoid is 15 (see Figure 5.12).

Again, the real motion parameters between the two views generated are

$$\Theta_r = \{25, 15, -25, 20, 10, 45\} \quad (5.9)$$

Ten iterations of curvature consistency were performed on each view separately and no noise was added.

In a first test, the process of view correspondence has been started with  $\Theta_r$  as estimate of the motion in order to determine a baseline for the second test. The motion parameters recovered were:

$$\Theta_{s1} = \{22.63, 15.52, -24.47, 20.29, 10.57, 44.14\} \quad (5.10)$$

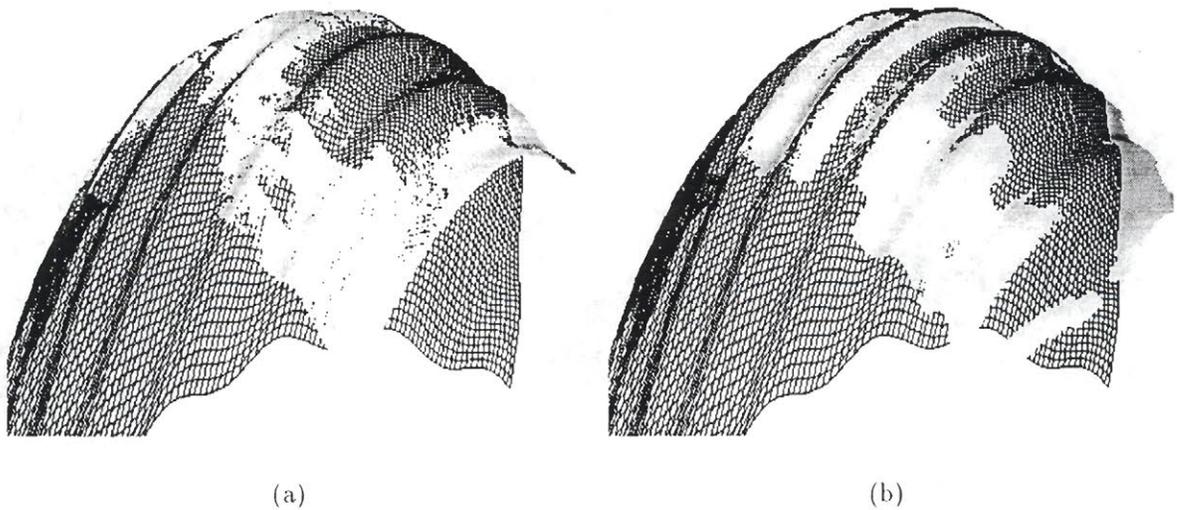
The difference between  $\Theta_r$  and  $\Theta_{s1}$  can be explained by the smoothing effect of the curvature consistency algorithm. Even though the numbers recovered are not perfect, the fit of the two views is nevertheless quite good as it is shown in Figure 5.12 a).

In the second test, the initial estimate of motion parameters was intentionally perturbed to confuse the correspondence algorithm. The idea was to show the *locking* effect of the repetitive structure. In fact, the perturbation of the motion parameters was made equal to the period of the repetitive pattern so that the process tries to match a point from the first view to a point on the next *period* of the second view. For this purpose, the following motion parameters were used as starting point:

$$\Theta_f = \{40, 15, -25, 20, 10, 45\} \quad (5.11)$$

As can be seen, only the displacement in  $x$  (direction of the sinusoid) was modified. Because of this modification, the correspondence algorithm made an error of one period of the repetitive structure (sinusoid). This means that the *local* structure of the sinusoid was seen as more important than the more *global* structure of the underlying paraboloid.

The result of this *mismatch* can be seen in Figure 5.12 (b). It is interesting to note that even though the match is wrong by one period of the sinusoid, the fit of the views is quite good. In a sense, this result shows the robustness of the method to non-rigid motion. In this case, if one was using rigid motion, it would be impossible to get a good fit of the surfaces in Figure 5.12 (b).



**Figure 5.12:** Registration of two views of a paraboloid with a repetitive structure (sinusoid). (a) shows the result of the process of view correspondence when started with the correct motion parameters ( $\Theta_r$ ). One of the views is shown as a shaded surface and the other one is shown as a grid. It is again possible to judge the quality of the fit by the interlace of the surfaces. (b) shows the same result but this time the initial estimate of motion parameters was perturbed so that two different periods of the sinusoid are compared across views. The structure of the sinusoid was strong enough (stronger than the underlying paraboloid) to cause an error of one period (relative to (a)). The fit of the surfaces is nevertheless quite good and shows the robustness of the method to non-rigid motion.

## 5.4 Results with Real Data

In this section, two results for real images are described. In one case, a small owl statuette has been rotated by the precision stages presented in section 5.2.2. In the other case, a scale model of a Mercedes-300 has been scanned with the set-up described in section 5.2.3 (Scanner Moved by a Robot). In both cases, the range finder described in section 5.2.1 was used.

### 5.4.1 A Small Owl Statuette

For this experiment, a small owl statuette was rotated by the precision stages described previously. A rotation of 45 degrees about the main axis of the owl was performed between the two views.

In order to get an estimate of the motion parameters, it is necessary to calibrate the set-up with respect to the range finder. The idea is to express the axis of rotation in the camera frame in order to be able to relate the two views. This process give us the following estimate of the relative motion parameters between the two views:

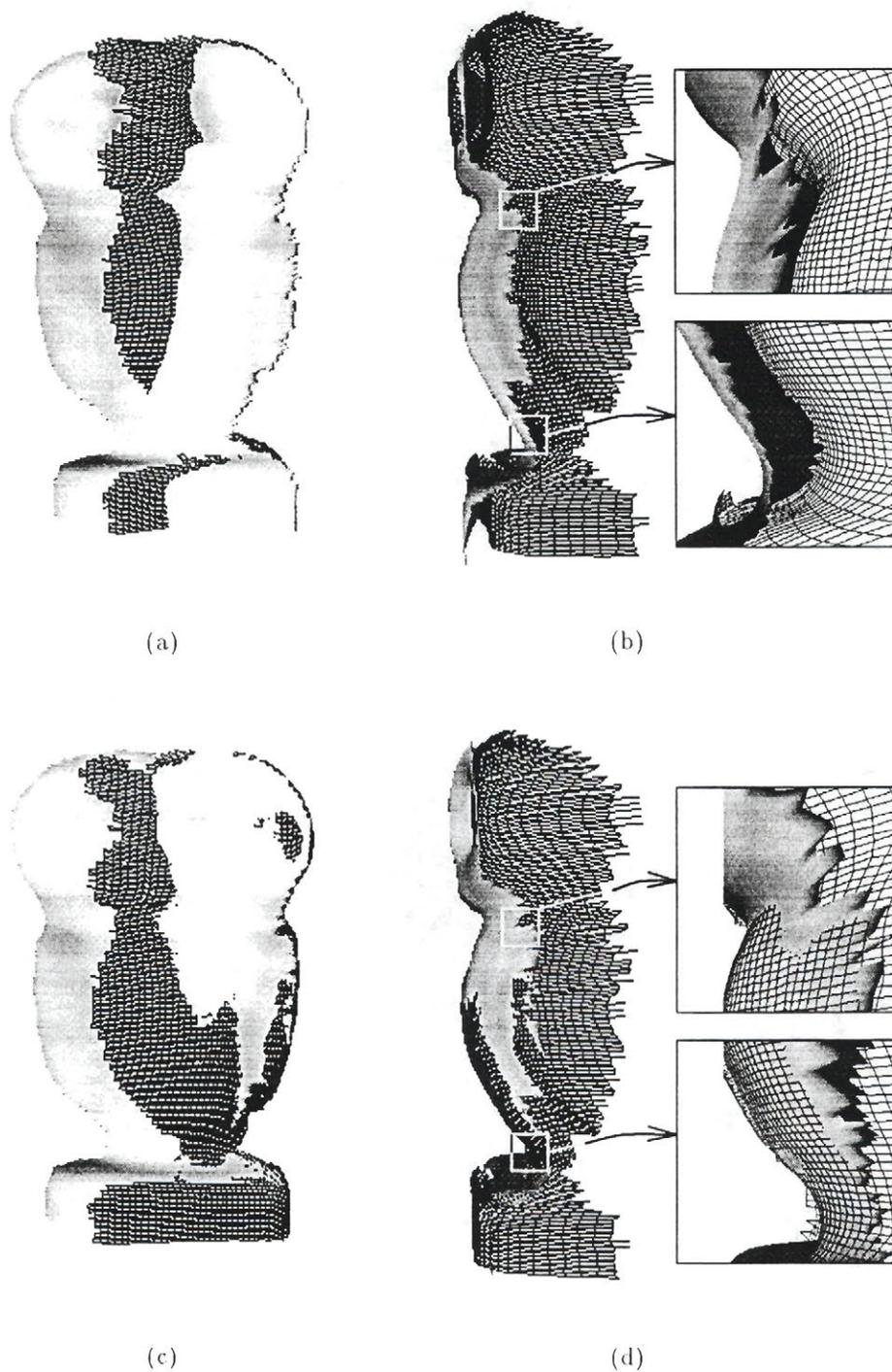
$$\Theta_e = \{-183.5, 3.2, 61.3, 1.3, 44.9, -3.6\} \quad (5.12)$$

In order to get a smooth estimate of the differential properties of the surface, fifteen iterations of curvature consistency were applied to both views separately using a neighborhood of  $5 \times 5$ . Figure 5.13 a) shows the resulting smooth surfaces put in correspondence using the estimated  $\Theta_e$ . This shows why we need to further refine this estimate.

In Figure 5.13 b), the two views are again fused, but this this time our correspondence algorithm is used instead. The fit is much more precise with the following motion parameters:

$$\Theta_u = \{-173.1, 11.0, 52.9, 3.47, 41.4, -4.38\} \quad (5.13)$$

The difference between  $\Theta_u$  and the estimation  $\Theta_e$  can be explained by a combina-



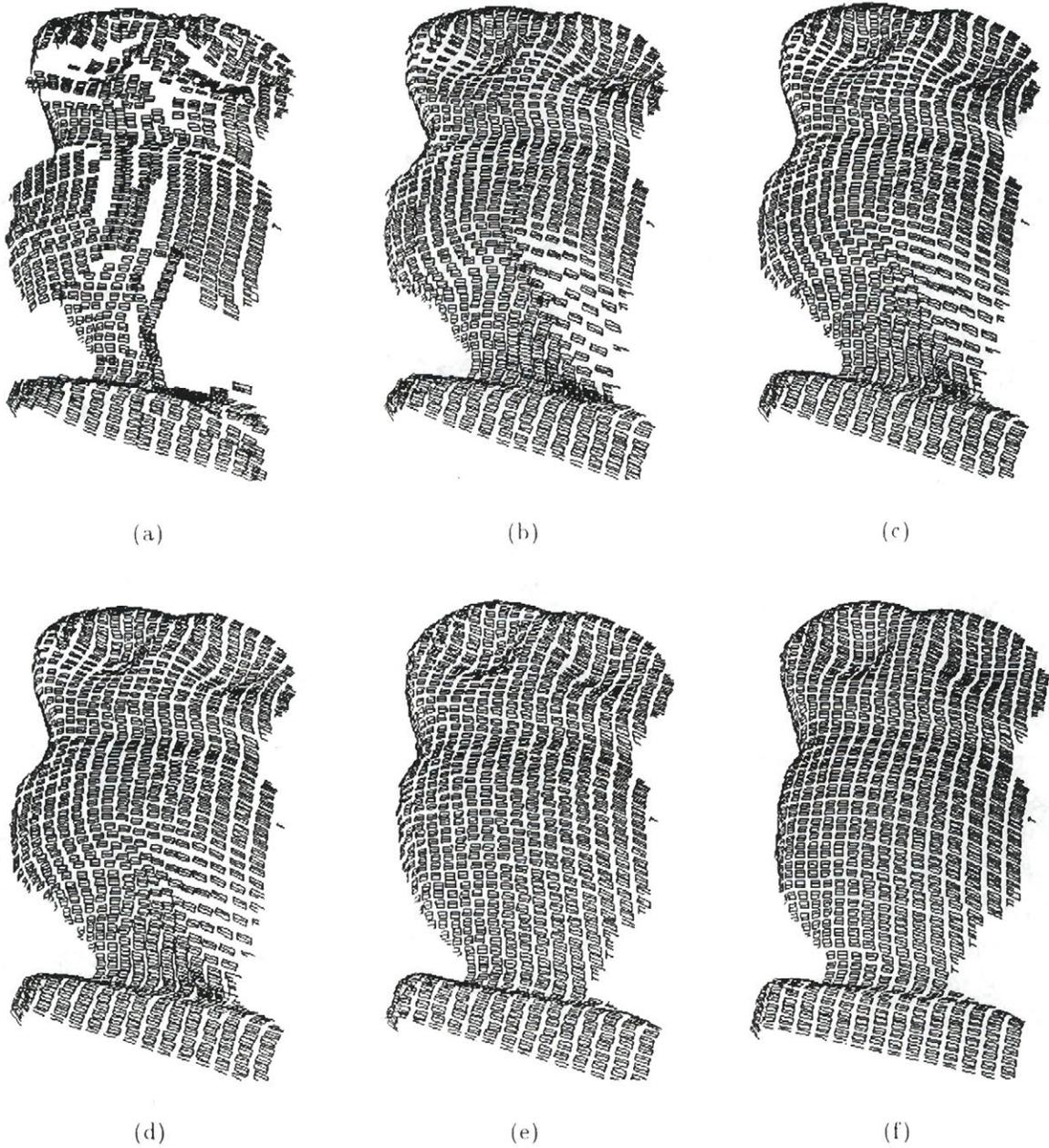
**Figure 5.13:** Registration of two views of a small statue owl. One of the views is shown as a shaded surface and the other one is shown as a grid. In (a) and (b), the estimated  $\Theta_e$  was used to put the views together. It is possible to see an imperfection in the match specially for one of the rotations. In (c) and (d) the result of our algorithm is shown after 50 iterations of motion consistency. In this case, a window of  $15 \times 15$  was used to for the correspondence algorithm. The interlace of the surfaces shows again the quality of the fit.

tion of factors. First, the calibration of the set-up is likely to be imperfect. Getting a good calibration requires careful set-up and many iterations of the calibration procedure. Another factor to consider is the imperfection of the range finder itself, that is to say its own calibration.

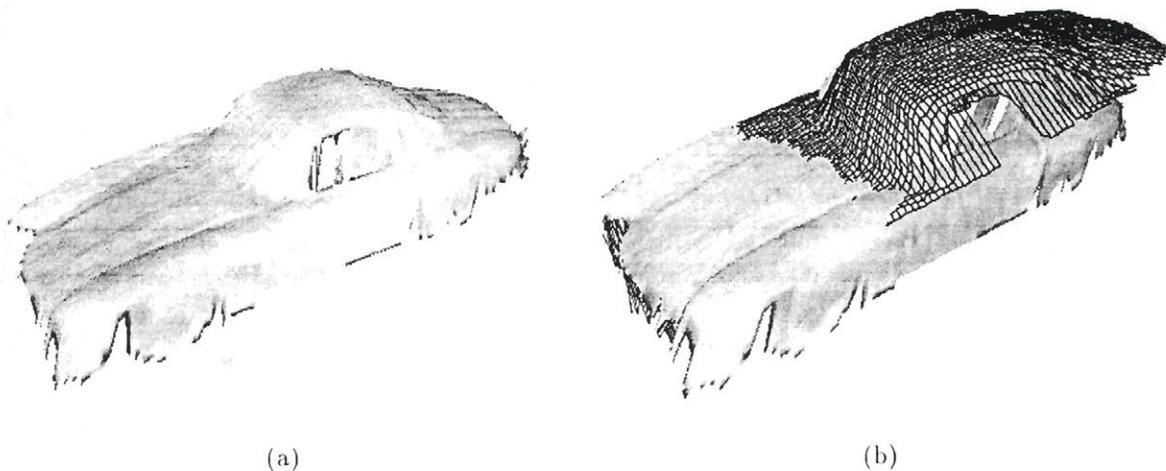
In Figure 5.14, different stages of motion consistency are shown. In this result, the first view of the owl is mapped in the coordinate frame of the second view (the second view itself is not shown). It is particularly interesting to note the imperfection of the motion parameters found for some patches on the belly before motion consistency. A closer look to this result shows that, in fact, the propagation of the motion parameters is the cause of this mismatch. The surface at these positions is almost umbilic. Because of this, it is hard to get a good estimate of the principal directions which leads to errors in correspondence of these points.

The solution to this problem is to avoid applying the correspondence algorithm on umbilic regions. We therefore need to detect these regions. A possible way to do so is to sum the ratio of the principal curvatures over a neighborhood for each point of the surface and to normalize the results obtained. This number can then be used in association with a threshold to detect the umbilic regions. As with any method involving thresholds, it is likely to fail on some occasions as it did in the case of a few points shown in Figure 5.14. It does however work most of the time and constitutes an interesting way to characterize the surface before selecting features for correspondence.

The sequence of images shown in Figure 5.14 demonstrates the robustness of the algorithm and its ability to gradually recover the correct structure of the surface. Obviously, there is a trade-off in the process. This trade-off is that some patches that were correctly positioned are perturbed slightly because of a few undetected mismatched patches. However, if most of the points are initially well positioned then the final result will not be affected much.



**Figure 5.14:** Mapping of the first view of the owl in the coordinate frame of the second view after different stages of motion consistency. The second view itself is not shown for clarity. (a) is the estimation of the motion parameters before motion consistency. It can be seen that the motion of some patches have been largely misevaluated especially on the belly. (b), (c), (d), (e) and (f) are respectively after 5, 10, 50, 100, and 1000 iterations of motion consistency. The sequence shows the gradual recovery of the correct structure of the surface.



**Figure 5.15:** Starting point of the view correspondence process for the reduced model of a Mercedes-300. (a) shows one of the views taken (left side plus the top). (b) shows the starting point of the process as obtained via the calibration of the set-up.

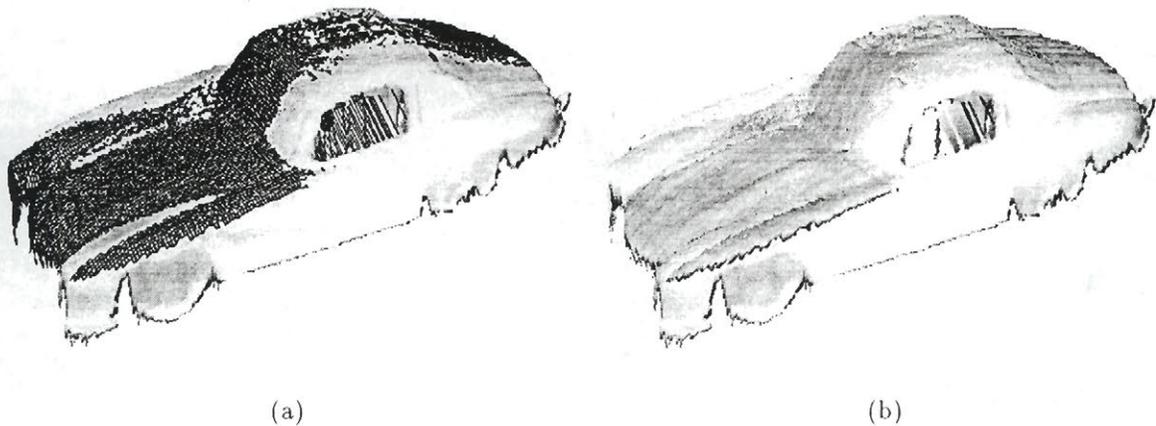
#### 5.4.2 A Scale Model of a Mercedes-300

For this experiment, two views of a scale model of a Mercedes-300 (scale 1/24) were taken using the set-up described in section 5.2.3 (Scanner Moved by a Robot). In this specific case, an estimate of the motion parameters was obtained via the calibration of the system (see Figure 5.15). As has been said in section 5.2.3, the motion parameters obtained in this manner are not very accurate because of the imprecision of the current calibration.

The two views were reconstructed using ten (10) iterations of curvature consistency with a  $5 \times 5$  window. And the final result of the process of view correspondence after 100 iterations of motion consistency is shown in Figure 5.16. A neighborhood of  $9 \times 9$  was used for the correspondence algorithm (initial comparison of structure).

### 5.5 Summary

In the first part of this chapter, we presented the different methods used to obtain multiple views of an object. The degree to which motion parameters can be estimated varies with the method.



**Figure 5.16:** Correspondence of views found for the Mercedes-300 using the starting point shown in Figure 5.15 b). A window of  $9 \times 9$  was used for the comparison of structure and 100 iterations of motion consistency were applied. (a) shows one of the views as a shaded surface and the other one as a grid. (b) shows the same result with both views as a shaded surface. It is again possible to judge the quality of the fit by the interlace of the surfaces.

The precision stages give a very good estimate of the displacement while the motion parameters obtained from the robot moving the range finder are a lot less precise (with the current calibration). It is also possible to use a manual matching of points to obtain motion estimates. The motion parameters obtained in this manner are usually good but require the intervention of a human (to effectively do the matching).

The correspondence of views obtained for synthetic data showed the validity of the method, its precision and its sensitivity. We have seen that the precision of the motion parameters recovered is largely dependent on the precision of the estimation of the Darboux frames. The effect of the discretization of the surface is important but it can be reduced with more iterations of curvature consistency. It should be noted that it is not useful to try to match views if the local structure of the surface has not been properly recovered beforehand.

The sensitivity of the method to repetitive structures has also been shown. In the presence of repetitive structures, the method can be easily confounded because it is local. In this case, the precision of the estimates should be better than the spatial frequency of the repetitive structure.

Finally, two results with real data were presented. The human intervention for these tests was limited to the figure/background separation and the selection of the window sizes for curvature consistency and the comparison of structure across views. These selections could eventually be made automatic by an analysis of the noise present in the views. The quality of the results obtained shows that the method is quite robust and can deal with noise from different sources (estimation of motion parameters, surface quantization, imprecision of sensors, etc.).

It has also been possible to see the convergence and the stability of motion consistency which goes gradually from non-rigid motion to completely rigid motion (1000 iterations).

A novel approach to the problem of view correspondence has been presented. The method proposed uses a rich description of the surface, i.e. the extended Darboux frame, to locally match views. The matching process exploits the fact that the metric of comparison is continuous and convex in the vicinity of a solution, allowing for rapid convergence. To improve the stability of local correspondence, an additional constraint on the recovered motion parameters was introduced which serves to filter out noise and reflect the continuity of motion of real objects. This constraint is referred to as *motion consistency*. The resulting algorithm has been shown to be accurate, stable and efficient in the recovery of rigid body motions given estimates of surface position. In addition, the local formulation of the algorithm combined with the motion blending afforded by motion consistency provides a means of dealing with non-rigid motion recovery.

The robustness of the algorithm is dependent on the accuracy of the local surface description as reflected by estimates of the extended Darboux frame at each sample point. A feature preserving smoothing filter based on the concept of curvature consistency was shown to provide sufficient accurate surface reconstruction for this purpose. In particular, it was demonstrated that the convexity of the metric of comparison was critically dependent on accurate recovery of the extended Darboux frame.

The efficiency of the search implied by the matching algorithm is a function of the metric of comparison. Although no formal proof was offered, it was demonstrated that the resulting function was convex in the vicinity of the correct solution. This permitted the use of a gradient descent algorithm for determining the set of motion parameters associated with a particular correspondence. Further gains in efficiency were made by applying the complete algorithm to a subset of points selected for the stability of their features (extremal points of curvature) and then propagating the solutions to adjacent points to be used to initialize new solutions. The *motion*

*consistency* algorithm was then applied to smooth out errors in motion parameters estimates resulting from errors in local correspondence.

The ability to deal locally with motion parameter estimation has two distinct advantages. First, the algorithm can accommodate the motion of non-rigid objects. Second, more reliable estimates can be determined for rigid objects. In the latter case, the *motion consistency* algorithm serves to integrate locally-determined estimates of the global rigid-body motion, providing a best estimate. The result can better tolerate local errors in correspondence that would confound more conventional rigid-body solutions.

Some interesting extensions to *motion consistency* are possible. One would be to give it the capability of detecting *motion discontinuities* and avoid smoothing over them. In this manner, objects composed of many moving parts could be dealt with more easily and more precisely. This can probably be accomplished by adapting some standard techniques (e.g. Kalman filtering) to the 6 dimensional space in which the motion parameters lie.

Another improvement would be to integrate *motion consistency* with the local estimation stage. In other words, the idea would be to devise a minimization based on two simultaneous constraints: similarity of structure and similarity of motion. At present these two constraints act at different levels and cannot interact properly together. A better result would likely be achieved by applying these constraints simultaneously rather than in tandem.

It would also be desirable to have a measure of confidence returned with estimates of each of the 6 motion parameters. This could be used in a weighting scheme applied to updating in the *motion consistency* algorithm. One possible approach would be to examine the shape of the error function in the vicinity of the solution. The metric value by itself is, in general, not a sufficient indicator of the quality of fit.

Finally, it should be said that the problem of view correspondence is and will likely continue to be a difficult but fascinating problem. It is hoped that some of the ideas put forward in this thesis might make a contribution towards solving this problem.

## Appendix A Determination of the Updated Frame in Curvature Consistency

In this Appendix, I describe the details of the updating rules for the set of transported Darboux frames  $Q_\alpha$ . As was stated in Chapter 3, the frames,  $Q_\alpha$ , are obtained by *parallel transport* of the neighbors' Darboux frames to the point to be updated (point  $P$ ). The  $Q_\alpha$  therefore represent the prediction of the Darboux frame at point  $P$  according to its neighbors.

The goal of the updating process is to determine a Darboux frame that minimizes the following functional form which embeds the constraint of minimum variation of curvature,

$$E = \sum_{k=1}^n (\kappa_{MP_\alpha}^k - \kappa_{MP})^2 + (\kappa_{\mathcal{M}P_\alpha}^k - \kappa_{\mathcal{M}P})^2 + (1 - \langle M_{P_\alpha}^k, M_P \rangle)^2 + (1 - \langle \mathcal{M}_{P_\alpha}^k, \mathcal{M}_P \rangle)^2 + (1 - \langle N_{P_\alpha}^k, N_P \rangle)^2 \quad (\text{A.1})$$

where  $N$ ,  $M$  and  $\mathcal{M}$  are unit length vectors.

Sander's approach [16] to solving this problem consisted of finding the normal and the principal curvatures in a first step and the principal directions afterwards. However, by doing so, the updated curvatures did not correspond to the updated principal directions. Rather, the curvatures associated with the updated frame were simply a mean of the raw curvatures of the translated frames and were not adapted to the updated principal directions. Lagarde showed in [9, p.35] that this introduced a loss of the local structure, specially near an umbilic point. Moreover, Sander, in his minimization, treated the principal directions as vectors i.e. as having an orientation content. Lagarde pointed out that principal directions have only a directional content and that a simple mean of their components was inappropriate.

In practice, the minimization of the metric  $E$  (equation A.1) is achieved in three steps. The first step of the updating procedure is the determination of the local

### A. Determination of the Updated Frame in Curvature Consistency

normal. In a range image, because the surface is of the form  $z = f(x, y)$ , the  $z$ -component of the normal is always of the same sign (positive or negative depending on convention). This leads to the following updating functional for  $N$

$$N^{i+1} = \frac{(\sum_{\alpha=1}^n N_{P\alpha x}^i, \sum_{\alpha=1}^n N_{P\alpha y}^i, \sum_{\alpha=1}^n N_{P\alpha z}^i)}{\sqrt{(\sum_{\alpha=1}^n N_{P\alpha x}^i)^2 + (\sum_{\alpha=1}^n N_{P\alpha y}^i)^2 + (\sum_{\alpha=1}^n N_{P\alpha z}^i)^2}}. \quad (\text{A.2})$$

The second step of the updating process consists of the determination of the principal direction  $M$ . This vector lies in the plane perpendicular to the normal, and can be written as a function of an angle  $\theta$  and of two arbitrary perpendicular vectors  $b_1$  and  $b_2$  lying in this plane

$$M(\theta) = b_1 \cos \theta + b_2 \sin \theta. \quad (\text{A.3})$$

The following conditions apply to  $b_1$  and  $b_2$

$$\langle b_1, b_1 \rangle = \langle b_2, b_2 \rangle = 1, \quad (\text{A.4})$$

$$\langle b_1, b_2 \rangle = \langle b_1, N \rangle = \langle b_2, N \rangle = 0. \quad (\text{A.5})$$

The following updating functional can be derived for  $E_M$ ,

$$E_M = \sum_{\alpha=1}^n 1 - \langle M^{i+1}, M_{\alpha}^i \rangle^2. \quad (\text{A.6})$$

The result of minimization is the angle  $\theta$  which determines the principal direction  $M$

$$\theta = \tan^{-1} \left[ \frac{(A_{22} - A_{11}) + \sqrt{(A_{22} - A_{11})^2 + 4A_{12}^2}}{2A_{12}} \right], \quad (\text{A.7})$$

where

$$A_{ij} = \sum_{\alpha=1}^n (M_{P\alpha} \cdot b_i)(M_{P\alpha} \cdot b_j). \quad (\text{A.8})$$

### A. Determination of the Updated Frame in Curvature Consistency

The other principal direction is determined by the cross-product

$$\mathcal{M} = M \times N. \quad (\text{A.9})$$

Finally, the third step of the updating process is the determination of the curvature magnitudes

$$\kappa_M = \frac{\sum_{\alpha=1}^n \kappa_{MP\alpha} \cos^2 \theta_{M\alpha M} + \kappa_{MP\alpha} \sin^2 \theta_{M\alpha M}}{n} \quad (\text{A.10})$$

$$\text{where } \theta_{M\alpha M} = \angle_{MP\alpha}^M$$

$$\text{and } \theta_{M\alpha M} = \angle_{MP\alpha}^M$$

and

$$\kappa_{\mathcal{M}} = \frac{\sum_{\alpha=1}^n \kappa_{MP\alpha} \cos^2 \theta_{M\alpha \mathcal{M}} + \kappa_{MP\alpha} \sin^2 \theta_{M\alpha \mathcal{M}}}{n} \quad (\text{A.11})$$

$$\text{where } \theta_{M\alpha \mathcal{M}} = \angle_{MP\alpha}^{\mathcal{M}}$$

$$\text{and } \theta_{M\alpha \mathcal{M}} = \angle_{MP\alpha}^{\mathcal{M}}$$

Recall that this process is done in parallel for all the points at a given iteration.

In addition to the normal and principal curvatures and directions, the depth of the points is also updated by interpolating the values suggested by the neighbors

$$z = \frac{\sum_{\alpha=1}^n z_{P\alpha}}{n}. \quad (\text{A.12})$$

Because curvature consistency is an iterative algorithm, it is important to have a way of characterizing convergence. A measure of the consistency of the Darboux frames over the image is given by the residual errors [16]. For a point  $P$ , the partial residual errors have the form

$$R_{\kappa} = \sum_{\alpha=1}^n (\kappa_M - \kappa_{MP\alpha})^2 + (\kappa_{\mathcal{M}} - \kappa_{MP\alpha})^2, \quad (\text{A.13})$$

### A. Determination of the Updated Frame in Curvature Consistency

$$R_N = \sum_{\alpha=1}^n (N_x - N_{xP\alpha})^2 + (N_y - N_{yP\alpha})^2 + (N_z - N_{zP\alpha})^2, \quad (\text{A.14})$$

$$R_M = \sum_{\alpha=1}^n (1 - \langle M, M_{P\alpha} \rangle)^2, \quad (\text{A.15})$$

$$R_{total} = R_\kappa + R_M + R_N. \quad (\text{A.16})$$

The summation of these partial errors for all the points updated at a given iteration provides a measure of the convergence.

## Appendix B

## Summary of Quaternion Algebra

This appendix presents the basics of quaternion algebra. It is not meant to be a complete reference but rather concentrates on notions needed in this thesis, specially for the updating of orientations in motion consistency.

A quaternion is four-dimensionnal entity composed of vector (3 components) and a scalar. The vector represents the axis of rotation while the scalar is function of the angle of rotation. Common representations include complex numbers form where  $s$  is the scalar and  $\mathbf{v}$  is the vector

$$\mathbf{q} = s + \mathbf{v}, \quad (\text{B.1})$$

and the  $4 \times 1$  vector form where  $\mathbf{n}$  is a  $3 \times 1$  unit length vector pointing in the direction of the axis of rotation and  $\theta$  is the angle of rotation about this axis

$$\{\mathbf{q}\} = \begin{Bmatrix} \sin \frac{\theta}{2} \mathbf{n} \\ \cos \frac{\theta}{2} \end{Bmatrix}. \quad (\text{B.2})$$

Note that this quaternion is unit length with a positive scalar part for rotations of  $+180$  to  $-180$  degrees. In fact, to avoid ambiguity caused by the equivalence of positive and negative rotations, the sense of rotation should be defined by the vector part using the positive sense determined by the right-hand rule.

Addition is associative and commutative but has no physical meaning:

$$\mathbf{q}_1 + \mathbf{q}_2 = (s_1 + s_2) + (\mathbf{v}_1 + \mathbf{v}_2). \quad (\text{B.3})$$

The conjugate  $\bar{\mathbf{q}}$  of a quaternion  $\mathbf{q}$  is defined as

$$\bar{\mathbf{q}} = s - \mathbf{v}. \quad (\text{B.4})$$

The conjugate of a quaternion is its inverse up to a scalar since it represents the

same rotation with a reversed (negated) axis of rotation. For a unit length quaternion (the length is computed as for a vector), the conjugate is exactly the reverse

$$\mathbf{q}\mathbf{q}^{-1} = \frac{\mathbf{q}\bar{\mathbf{q}}}{\|\mathbf{q}\|^2} = \mathbf{1}. \quad (\text{B.5})$$

The multiplication of quaternions is associative but not commutative. It is interpreted as the composition of rotations

$$\mathbf{q}_1\mathbf{q}_2 = (s_1s_2 - \mathbf{v}_1 \cdot \mathbf{v}_2) + s_2\mathbf{v}_1 + s_1\mathbf{v}_2 + \mathbf{v}_1 \times \mathbf{v}_2. \quad (\text{B.6})$$

The relationship between rotation matrices and quaternion is given as

$$\mathbf{Q} = (s^2 - \mathbf{v}^T\mathbf{v})\mathbf{1} + 2\mathbf{v}\mathbf{v}^T + 2s\mathbf{v}^\times \quad (\text{B.7})$$

where  $\mathbf{a}^\times$  is the matrix whose operation on a vector yields the same result as the vector cross-product  $[\mathbf{a}^\times]\mathbf{b} = \mathbf{a} \times \mathbf{b}$ .

The operation of a quaternion on a  $3 \times 1$  vector  $\mathbf{a}$  is defined as:

$$\mathbf{q}\mathbf{a} = -\mathbf{v} \cdot \mathbf{a} + (\mathbf{v} \times \mathbf{a} + s\mathbf{a}) \quad (\text{B.8})$$

and applying a rotation operation to this vector is done as follow:

$$\mathbf{Q}\mathbf{a} = \mathbf{q}\mathbf{a}\bar{\mathbf{q}}. \quad (\text{B.9})$$

Rotations are composed by multiplication of quaternions in the following manner

$$\mathbf{q}_C^A = \mathbf{q}_B^A\mathbf{q}_C^B. \quad (\text{B.10})$$

It is interesting to note that given two systems with common origin and a quaternion determining their relative orientation, the vector part of this quaternion (representing the axis of rotation) is expressed in the same manner in both systems ( $\mathbf{v}_A = \mathbf{v}_B$ ). It is in fact an eigenvector of the corresponding rotation matrix  $\mathbf{Q}_B^A$ .

The representation of the unit length quaternions on the unit 4D sphere is also

interesting and important. Any coherent interpretation of the quaternions should be done through this sphere which has the following properties.

The quaternions of opposite directions on this sphere represent the same rotation. For instance, the subset of unit quaternions with positive scalar part is sufficient to represent all the rotations that can be applied to an object.

An other important characteristic is the possibility of interpolating rotation. This characteristic is called “great arc in-betweening” in [18]. When moving from a quaternion  $q_1$  to an other one  $q_2$  along the great arc circle relating them on the sphere, the corresponding rotation of an object is changed smoothly between the orientations determined by  $q_1$  and  $q_2$ . In other words, when using rotations expressed as unit quaternions on the sphere, it is possible to manipulate orientation as easily as it is to manipulate vectors in 3-D space.

## Appendix C

## Interpreting rotations

This appendix gives the meaning for the notation related to rotation used in this thesis. In the meantime, it also tries to clarify the ambivalence of a rotation in its general sense.

A rotation matrix  $\mathbf{Q}_B^A$  can be interpreted in two ways. In its first interpretation (probably the most common one), it represents a way to express a physically fixed vector given in a frame of reference  $B$  into another frame of reference  $A$ :

$$\mathbf{a}_A = \mathbf{Q}_B^A \mathbf{a}_B \quad (\text{C.1})$$

The columns of the  $3 \times 3$  matrix  $\mathbf{Q}_B^A$  are the axes of the frame  $B$  expressed in frame  $A$

$$\mathbf{Q}_B^A = \left[ \begin{array}{ccc} \mathbf{x}_B & \mathbf{y}_B & \mathbf{z}_B \end{array} \right]_A \quad (\text{C.2})$$

The so-called invariants (angle of rotation and axis of rotation) of this rotation matrix:

$$\mathbf{Q}_B^A = \left[ \begin{array}{ccc} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{array} \right]_A \quad (\text{C.3})$$

are defined as

$$\text{tr}(\mathbf{Q}) = a_{11} + a_{22} + a_{33} = 1 + 2 \cos \theta \quad (\text{C.4})$$

and

$$\mathbf{v} = [\text{vect}(\mathbf{Q})]_A = \frac{1}{2} \begin{bmatrix} a_{32} - a_{23} \\ a_{13} - a_{31} \\ a_{21} - a_{12} \end{bmatrix} \quad (\text{C.5})$$

where  $\theta$  is the angle of rotation and  $\mathbf{v}$  is a vector parallel to the axis of rotation.

The other interpretation of a rotation matrix is the application of a *physical* rotation to a vector. The rotation applied is the same as the one that brings frame  $A$  in correspondance with frame  $B$ . The result is therefore *another* vector still expressed in frame  $A$ :

$$\mathbf{b}_A = \mathbf{Q}_B^A \mathbf{a}_A. \quad (\text{C.6})$$

In the same manner, the quaternion  $\mathbf{q}_B^A$  represents the rotation that brings frame  $A$  in correspondance with frame  $B$ . Therefore, this quaternion used in equation B.7 will produce the corresponding rotation matrix  $\mathbf{Q}_B^A$ . In terms of equations, we have

$$\mathbf{a}_A = \mathbf{Q}_B^A \mathbf{a}_B = \mathbf{q}_B^A \mathbf{a}_B \bar{\mathbf{q}}_B^A. \quad (\text{C.7})$$

Finally, given a reference frame  $R$  and two quaternions  $\mathbf{q}_A^R$  and  $\mathbf{q}_B^R$  relating the frames  $A$  and  $B$  to  $R$ , then the relationship between  $A$  and  $B$  is found by:

$$\mathbf{q}_B^A = \bar{\mathbf{q}}_A^R \mathbf{q}_B^R = \mathbf{q}_R^A \mathbf{q}_B^R \quad (\text{C.8})$$

One should note that when multiplying two quaternions to compose a rotation, a negative value for the scalar part can be obtained. The ambiguity possibly caused by this negative value can be removed using the fact that the negative of a quaternion still represents the same rotation. Therefore, when a negative scalar value is produced, the quaternion should be negated to keep the convention of positive scalar part and keep the angle of rotation in the interval of  $+180$  to  $-180$  degrees.

## Bibliography

- [1] Bir Bhanu. Representation and shape matching of 3-d objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(3):340–351, May 1984.
- [2] A. Blake and A. Zisserman. *Visual Reconstruction*. Cambridge: The MIT Press, 1987.
- [3] Yang Cheng and Gérard Medioni. Object modeling by registration of multiple range images. *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, 1991.
- [4] M.P. doCarmo. *Differential geometry of curves and surfaces*. Prentice Hall, 1976.
- [5] Alberto Elfes and Larry Matthies. Sensor integration for robot navigation: Combining sonar and stereo range data in a grid-based representation. *IEEE Conference on Decision and Control*, pages 1802–1807, December 1987.
- [6] Shailendra Mathur F.P. Ferrie and Gilbert Soucy. *Feature Extraction for 3-D Model Building and Object Recognition*. Elsevier Science Publisher BV., 1993.
- [7] G. Godin. Edge-based scene description using range imaging. Master's thesis, McGill University, 1989.
- [8] D.B. Goldgof, H. Lee, and T.S. Huang. Motion analysis of nonrigid surfaces. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 374–380, Ann Arbor, Michigan, June 1988. Computer Society of the IEEE, IEEE Computer Society Press.
- [9] Jean W. Lagarde. Constraints and their satisfaction in the recovery of local surface structure. Master's thesis, McGill University, 1990.
- [10] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. *IJCAI*, 1981.

- [11] F. Blais M. Rioux and J.-A. Beraldin. Laser range finder development for 3-d vision. In *Vision Interface '89*, London, Ontario, June 1989. Canadian Image Processing and Pattern Recognition Society.
- [12] D. Laurendeau M. Soucy, A. Croteau. A multi-resolution surface model for compact representation of range images. *Proceedings of the IEEE International Conference on Robotics and Automation*, 1992.
- [13] P. Parent and S.W. Zucker. Curvature consistency and curve detection. *J. Opt. Soc. Amer., Ser. A*, 2(13), 1985.
- [14] Michael Potmesil. Generating models of solid objects by matching 3d surface segments. *IJCAI*, 2(8), 1983.
- [15] Denis Laurendeau Robert Bergevin and Denis Poussart. Estimating the 3d rigid transformation between two range views of a complex object. *Proceedings of the 11th IAPR International Conference on Pattern Recognition*, 1992.
- [16] Peter T. Sander. *On Reliably Inferring Differential Structure from Three Dimensional Images*. PhD thesis, McGill University, 1988.
- [17] Mubarak A. Shah and Ramesh Jain. Detecting time-varying corners. *Computer Vision, Graphics and Image Processing*, pages 345–355, 1984.
- [18] Ken Shoemake. Animating rotation with quaternion curves. *ACM Computer Graphics*, 19(3), 1985.
- [19] K. Spring. Euler parameters and the use of quaternions algebra in the manipulation of finite rotations: a review. *Mechanism and machine Theory*, 21(5):365–373, 1986.
- [20] Richard Szeliski. Estimating motion from sparse range data without correspondence. *Proceedings of the International Conference on Computer Vision*, 1988.
- [21] B.C. Vemuri and J.K. Aggarwal. 3-d model construction from multiple views using range and intensity data. *Proceedings CVPR*, pages 435–437, 1986.

- [22] P. Whaite and F.P. Ferrie. From uncertainty to visual exploration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-13(10):1038-1050, October 1991.