Surface Recovery Using Curvature and Motion Consistency

G. Soucy F.P. Ferrie

Computer Vision and Robotics Laboratory
McGill Research Center for Intelligent Machines
Montréal, Québec, CANADA

April 1995

A key problem in the recovery of scene descriptions from multiple views is the fusion of information from different vantage points. The contribution of this paper is a set of algorithms for reconstructing surfaces obtained from overlapping range images in a common frame of reference. Surfaces are assumed to be piecewise-smooth but not necessarily rigid. Motion parameters, rotations and translations that describe correspondence between views, are recovered locally under the assumption that the curvature structure at a point on a surface varies slowly under transformation. The recovery problem can thus be posed as finding the set of motion parameters that preserves curvature across two views. We show that an appropriate similarity functional can be devised that is convex in the vicinity of the true motion parameters. This leads to an efficient local algorithm that recovers motion parameters in gradient descent fashion.

Fusion of information from different viewpoints is accomplished by applying local motion estimates to map data points between frames. However, because these estimates are determined locally, they are subject to the usual effects of noise and quantization error. To increase the robustness of this reconstruction procedure the additional constraint of motion consistency is introduced, that variations in the velocities of adjacent regions are also piecewise-smooth. This is cast as a second local minimization problem which seeks to find the set of motion parameters that minimizes differences in the relative positions and orientations at adjacent points. The resulting algorithm serves to smooth out local perturbations and blend adjacent surface patches. In contrast to global rigid body motion approaches, our procedure for reconstructing surfaces from different viewpoints is tolerant of local errors in correspondence and can accommodate objects that are articulated.

Acknowledgments

This research was supported in part by the Natural Sciences and Engineering Research Council of Canada under Grant OGPIN 011 as well as a post graduate scholarship. The authors wish to thank S.W. Zucker and P. Whaite for their helpful discussions and comments during the course of this work. Additional thanks to P. Whaite for systems and computational support.

Postal Address: 3480 University St., Montréal, Québec, CANADA H3A 2A7
Telephone: (514) 398-6042 Fax: (514) 398-7348
Network Address (Internet): ferrie@mcrcim.mcgill.edu

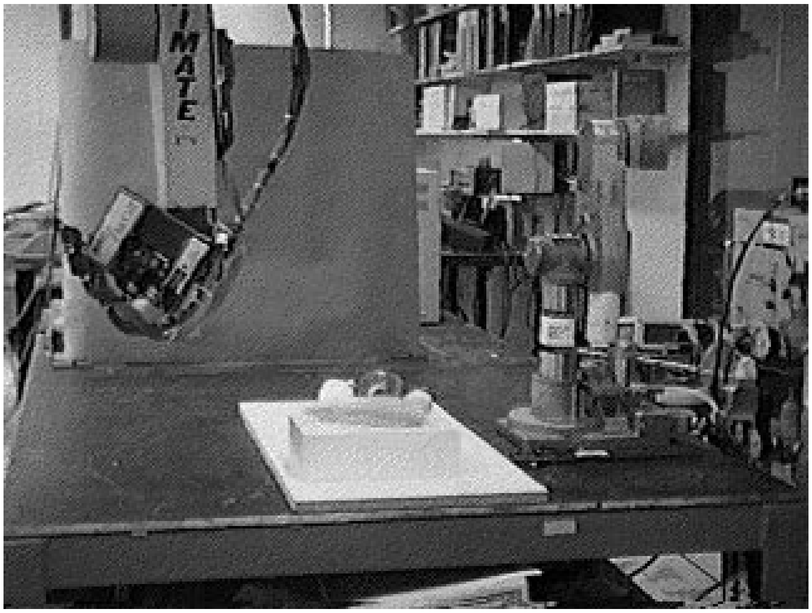


Figure 1: The mobile sensor consists of a laser rangefinding system with a 1 m^3 field of view mounted on the end-effector of an inverted PUMA 560 robot.

1 Introduction

Figure 1 depicts a problem common in dynamic scene analysis, the integration of scene information from multiple viewpoints. In the example shown, a laser rangefinding system attached to the end-effector of an inverted PUMA 560 robot is used to explore the surrounding workspace. Tasks such as these are becoming increasingly more common with the current interest in active vision and mobile robotics [41]. In this paper we develop a solution for the specific problem of fusing together in a common 3-D reference frame a set of overlapping range images obtained either from laser range measurements or from stereo. The traditional approach to solving this problem consists of first determining the motion parameters (rotations and translations) used to describe the correspondence between points in adjacent images, and then reconstructing a composite surface by mapping each data point into a common frame of reference. The contribution of this paper is two-fold. One is an algorithm for the recovery of local motion parameters on non-rigid surfaces, and the other is a second algorithm for refining these estimates according to a physical motion model.

Motion parameter estimation is cast as a local correspondence problem in terms of differential geometry. Surfaces are assumed to be piecewise-smooth and locally deformable, encompassing a broad range of natural and man-made scenes. The result of this formulation is an algorithm which is efficient, robust, and void of overly restrictive assumptions, e.g., a single rigid moving object. In fact, where the surface is known a priori to correspond to a rigid structure, local motion estimates can be combined into a single estimate that is very tolerant of errors in local correspondence.

The estimation problem is posed as finding the sets of local motion parameters that best preserve the curvature structure of a surface across adjacent views. Each set describes

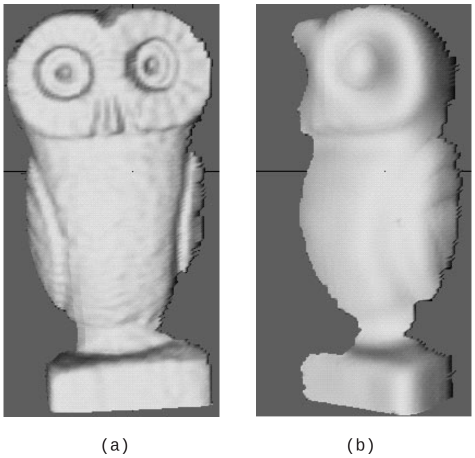


Figure 2: Two views of a statue of an owl corresponding to a rotation of approximately 45° about the longitudinal axis. Data were acquired using a laser rangefinding system and rendered as shaded images corresponding to the two depth maps.

the motion of a single point or region depending on the rigidity of the underlying surface. Non-rigid motion is accommodated provided that the curvature structure of the surface varies slowly with respect to local deformations. In fact, because motion parameters are estimated locally, articulations are possible in which different sections of the surface can move with different velocities. The key to solving the estimation problem is devising a functional $D_{\Omega T}$, that measures the similarity between two local neighbourhoods centered at points $\mathbf{x} \in I(i,j,t)$ and $\mathbf{x}' \in I(i,j,t+1)$ in adjacent images, that is continuous in motion parameters $\mathbf{\Omega}$ and $\mathbf{T}$. These are respectively the rotation matrix and translation vector which map $\mathbf{x}$ to $\mathbf{x}'$. In particular we show that by exploiting some well-known forms from differential geometry [9], a $D_{\Omega T}$ can be devised that is convex in the vicinity of the true motion parameters. This leads to an efficient local algorithm that recovers motion parameters $\mathbf{\Omega}$ and $\mathbf{T}$ in gradient descent fashion.

The second contribution concerns the robust interpretation of $\mathbf{\Omega}$ and $\mathbf{T}$ and is best illustrated by example. Figure 2 shows two views of a statue of an owl corresponding to a rotation of approximately 45° about the longitudinal axis. The task at hand is to fuse the corresponding range images into a single coordinate frame. To do so we first sub-divide each image into a set of rectangular sub-regions[1] and then apply our local algorithm to estimate $\mathbf{\Omega}$ and $\mathbf{T}$ for each. By applying the estimated motion parameters to each sub-region, one can reconstruct a composite surface in the coordinate frame of either view (a) or view (b). Figure 3a shows the result of this operation where each element of Figure 2b has been mapped into the coordinate frame of Figure 2a. Notice that the results are quite noisy. In fact, the situation gets worse as successive transformations are applied over a sequence of images. This result is not surprising considering that motion estimates are based on regions with limited support. Clearly some further interpretation is required.

One approach would be to apply a second stage of reconstruction to the composite surface in order to smooth out perturbations due to errors in motion parameters [4, 21]. The problem

[1] This is done for computational expediency.

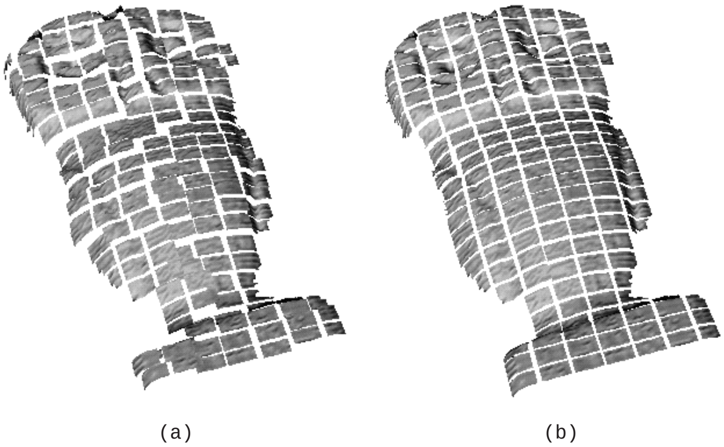


Figure 3: (a) Composite surface generated by directly applying local motion estimates. (b) The same surface after first applying the motion consistency algorithm to the local motion estimates.

here is that such errors tend to introduce structural changes which are difficult to separate out, e.g. the introduction of surface discontinuities. In addition care must be taken not to over-smooth the surface, resulting in a net loss of information. However, one can gain some insight into how to further interpret local motion estimates by considering the behaviour of physical surfaces. The key observation is that adjacent particles are coupled and cannot move independently. This implies that the corresponding motion parameters must be locally coupled as well, and provides an important constraint that can be used to further refine local motion estimates. We call this constraint *motion consistency* and show how it can be used to devise a procedure that iteratively minimizes the local variation of motion parameters such that the relative displacements and orientations at adjacent points are consistent with the local surface model. Figure 3b shows the composite surface that results after applying the motion consistency algorithm to the set of local motion estimates used to obtain the surface shown in Figure 3a. The result clearly shows a marked improvement. In addition to refining motion estimates, the algorithm also serves as a means of interpolating motion parameters into regions of the surface for which local estimates cannot be obtained, e.g. umbilic regions which are void of directional information.

1.1 Background and Methodology

Whether an image sequence is generated from the observation of a mobile observer in motion about a stationary scene (Figure 1), or a mobile scene in motion about a fixed observer, or some combination of the two, the essential task is the same - to determine the motion parameters relating corresponding points on surfaces in adjacent views. This problem has a long history in the computer vision literature. Approaches have included the determination of point correspondences and analysis of point configurations [5, 19, 27, 46, 45, 8, 48, 42, 6, 30, 44, 7, 12, 11, 13, 25], the analysis of flow fields [33, 29, 47, 2, 35, 36, 43], and photometric methods [32, 31, 23, 1]. The method chosen is a function of both the data and how they are

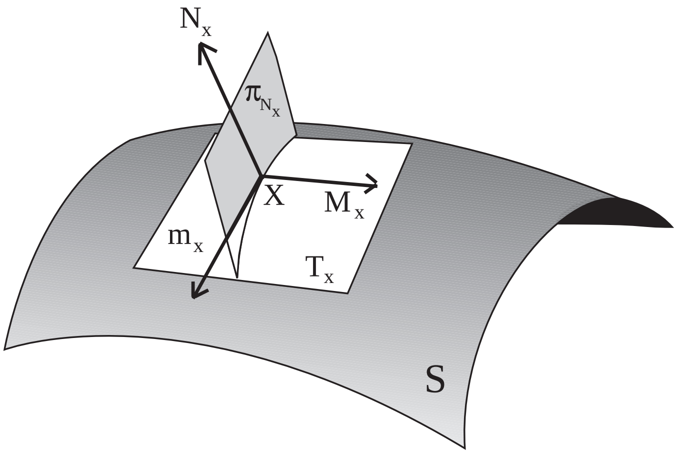


Figure 4: Local surface representation - the augmented Darboux frame

acquired.

In our application surfaces are assumed to be piecewise-smooth, hence salient features can be difficult to localize in the presence of noise. This suggests a continuous approach to the recovery of motion parameters along the lines of flow analysis. The requirement of determining local motion parameters further suggests that this flow should be expressed in terms of the differential structure of the surface. A convenient representation in this regard is what Sander [38] refers to as an *augmented* Darboux frame [9]. To each point $\mathbf{x} \in I(i,j,t)$ an augmented Darboux frame, $\mathcal{D}(\mathbf{x})$, is associated which describes the orientation and curvature of the local neighbourhood of $\mathbf{x}$. Each frame is composed (Figure 4) of three orthonormal vectors, the unit normal, $N_{\mathbf{x}}$, and principal direction vectors $M_{\mathbf{x}}$ and $\mathcal{M}_{\mathbf{x}}$ lying in the plane tangent to $\mathbf{x}$, $T_{\mathbf{x}}$. $M_{\mathbf{x}}$ and $\mathcal{M}_{\mathbf{x}}$ correspond to the directions for which the normal curvature at $\mathbf{x}$ (a directional property) takes on maximum and minimum values, $\kappa_{M_{\mathbf{x}}}$ and $\kappa_{\mathcal{M}_{\mathbf{x}}}$ respectively. The latter scalar quantities are referred to as the principal curvatures at $\mathbf{x}$. Collectively, the elements $(\mathbf{x}, M_{\mathbf{x}}, \mathcal{M}_{\mathbf{x}}, N_{\mathbf{x}}, \kappa_{M_{\mathbf{x}}}, \kappa_{\mathcal{M}_{\mathbf{x}}})$ are referred to as the augmented Darboux frame at $\mathbf{x}$, $\mathcal{D}(\mathbf{x})$.

With this representation in place, the problem of estimating local motion parameters can now be precisely formulated as finding that set of parameters, $\mathbf{\Omega}$ and $\mathbf{T}$ which maps $\mathcal{D}(\mathbf{x})$ to $\mathcal{D}(\mathbf{x}')$ for each $\mathbf{x} \in I(i,j,t)$. However, correctly and efficiently recovering local motion parameters also depends on being able to accurately recover $\mathcal{D}(\mathbf{x})$. Such methods have been investigated in [38, 21, 14, 16, 17, 15, 26], and form the basis of our current work. Briefly, the idea is to first use local approximation techniques to build initial estimates of $\mathcal{D}(\mathbf{x})$. Next a specialized filter is applied to these estimates which results in a surface whose variation in curvature is piecewise-smooth. This filter is obtained by minimizing a functional form that represents local differences in $\mathcal{D}(\mathbf{x})$ under the assumption of locally constant curvature. The principle is often referred to as *curvature consistency* [34].

With estimates of $\mathcal{D}(\mathbf{x})$ $\forall \mathbf{x} \in I(i,j,t)$ and $\mathcal{D}(\mathbf{x}')$ $\forall \mathbf{x}' \in I(i,j,t+1)$ computed, a similar minimization strategy can be applied to solve for the local motion parameters. In this case differences in $\mathcal{D}(\mathbf{x})$ and $\mathcal{D}(\mathbf{x}')$ are minimized over variations in $\mathbf{\Omega}$ and $\mathbf{T}$. From a computational vantage point it is tempting to simply embed this temporal minimization within the curvature consistency framework. However, this approach does not work because surface recovery is confounded by motion, i.e. the motion parameters that produce the smoothest surface in terms of temporal curvature variation do not necessarily produce the correct one. Thus our strategy is instead to first apply the curvature consistency algorithm [38, 18] independently to $I(t)$ and $I(t+1)$ to obtain stable surface representations. The next stage of processing then attempts to find for each selected candidate point in $I(t)$ a set of motion parameters that yields a corresponding point in $I(t+1)$ such that variation in the associated frames $\mathcal{D}(\mathbf{x})$ and $\mathcal{D}(\mathbf{x}')$ is minimum.

The strategy used to smooth out variations in motion parameters is related to curvature consistency. Rather than minimizing the variation of curvature over a spatial neighbourhood, the idea is to minimize the variation of the surface over a temporal neighbourhood such that local motion parameter variation is minimized, hence the term *motion consistency*. In curvature consistency the augmented Darboux frame serves as a basis for comparison between two local neighbourhoods and provides the analytical form on which an appropriate minimization functional is devised [38, 21]. Motion consistency relies on a similar analog to represent motion parameters, namely the rotation matrix $\mathbf{\Omega}$ and translation vector $\mathbf{T}$. By expressing rotations as their equivalent *quaternions* [40, 39], and following a similar line of analysis, one obtains a set of updating functions that forms the basis of an iterative filter. Finally, just as curvature consistency preserves the local structure of the underlying surface, motion consistency preserves the structure of the rotation matrix $\mathbf{\Omega}$ and translation vector $\mathbf{T}$ at each sample point.

1.2 Overview

The remainder of the paper provides a theoretical justification for our approach, describes the technical details of the algorithms, and presents some of the results obtained on laser rangefinder image sequences. We begin in Section 2 by deriving a functional used to compare local surface descriptions as a function of motion parameters and show that it is convex in the vicinity of the true parameters. This leads, in Section 3, to the first algorithm that exploits the convexity of this functional in estimating motion parameters. Section 4 describes how the concept of motion consistency can be used to refine and interpolate motion parameter estimates, and leads to the motion consistency algorithm. We also present the results of experiments, in Section 5 that show how these two algorithms can be used to reconstruct several overlapping range images in a common frame.

2 Estimating Local Motion Parameters

We begin by considering the relationship between two local neighbourhoods, $\mathcal{N}(\mathbf{x})$ centered at point $\mathbf{x} \in I(i,j,t)$ and $\mathcal{N}(\mathbf{x}')$ centered at point $\mathbf{x}' \in I(i,j,t+1)$. The images $I(i,j,t)$ and $I(i,j,t+1)$ are discrete samples of piecewise-smooth surfaces $\mathcal{S}$ and $\mathcal{S}'$ respectively. Surfaces $\mathcal{S}$ and $\mathcal{S}'$ need not be rigid on a global scale, but are assumed to be locally rigid within $\mathcal{N}(\mathbf{x})$ and $\mathcal{N}(\mathbf{x}')$, i.e. we can describe the displacement of each neighbourhood with a single set of parameters. Thus for any $\mathbf{x}_i \in \mathcal{N}(\mathbf{x})$ there exists an $\mathbf{x}'_i \in \mathcal{N}(\mathbf{x}')$ such that

$$\mathbf{x}'_i = \mathbf{\Omega}\mathbf{x}_i + \mathbf{T}, \tag{1}$$

where

$$\mathbf{\Omega} = \begin{bmatrix} C(\theta_y)C(\theta_z) & C(\theta_x)S(\theta_z) + S(\theta_x)S(\theta_y)C(\theta_z) & S(\theta_x)S(\theta_z) - C(\theta_x)S(\theta_y)C(\theta_z) \\ -C(\theta_y)S(\theta_z) & C(\theta_x)C(\theta_z) - S(\theta_x)S(\theta_y)S(\theta_z) & C(\theta_x)S(\theta_y)S(\theta_z) + S(\theta_x)C(\theta_z) \\ S(\theta_y) & -S(\theta_x)C(\theta_y) & C(\theta_x)C(\theta_y) \end{bmatrix},$$

$$\mathbf{T} = \begin{bmatrix} T_x \\ T_y \\ Tz \end{bmatrix}, \tag{2}$$

θ_x, θ_y, θ_z are rotations and T_x, T_y, T_z translations about/along the X, Y, and Z axes respectively, and the abbreviations $C()$ and $S()$ are used in place of cos() and sin() respectively.

Knowledge of $\mathbf{x}_i$ and $\mathbf{x}'_i$ alone does not provide a sufficient basis from which to determine $\mathbf{\Omega}$ and $\mathbf{T}$. Instead, consider the relationship between two corresponding frames $\mathcal{D}(\mathbf{x})$ and $\mathcal{D}(\mathbf{x}')$ at $\mathbf{x}$ and $\mathbf{x}'$ respectively,

$$\mathcal{D}(\mathbf{x}') = \mathcal{T}(\mathcal{D}(\mathbf{x}), \mathbf{\Omega}, \mathbf{T}), \tag{3}$$

where $\mathcal{T}()$ is defined as follows:

$$\begin{aligned} \mathbf{x}' &= \mathbf{\Omega}\ \mathbf{x} + \mathbf{T}, \\ M_{\mathbf{x}'} &= \mathbf{\Omega}\ M_{\mathbf{x}}, \\ \mathcal{M}_{\mathbf{x}'} &= \mathbf{\Omega}\ \mathcal{M}_{\mathbf{x}}, \\ N_{\mathbf{x}'} &= \mathbf{\Omega}\ N_{\mathbf{x}}, \\ \kappa_{M_{\mathbf{x}'}} &= \kappa_{M_{\mathbf{x}}}, \\ \kappa_{\mathcal{M}_{\mathbf{x}'}} &= \kappa_{\mathcal{M}_{\mathbf{x}}}. \end{aligned} \tag{4}$$

As is shown in Figure 5, each frame has a position and orientation which is sufficient to completely specify the rotation matrix $\mathbf{\Omega}$ and translation vector $\mathbf{T}$. Thus given the correspondence between two frames the motion parameters are completely determined. However in order to determine this correspondence in the first place, a view-invariant measure is required which serves to identify corresponding frames. Observe in (4) that the principal curvatures $\kappa_{M_{\mathbf{x}'}}$ and $\kappa_{\mathcal{M}_{\mathbf{x}'}}$ are independent of $\mathbf{\Omega}$ and $\mathbf{T}$. We now consider the conditions under which correspondence may be uniquely determined.

2.1 Determining Correspondence

A number of constraints serve to simplify the problem of determining the correspondence between $\mathcal{D}(\mathbf{x})$ and $\mathcal{D}(\mathbf{x}')$:

- $\mathcal{D}(\mathbf{x})$ and $\mathcal{D}(\mathbf{x}')$ are constrained to lie on $\mathcal{S}$ and $\mathcal{S}'$ respectively, limiting the space of motion parameters that must be searched.

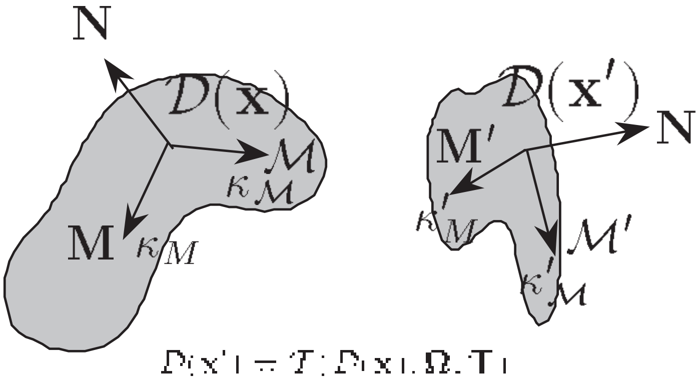

N
$\mathcal{D}(\mathbf{x})$
$\mathcal{M}$
$\kappa_{\mathcal{M}}$
$\mathbf{M}$
κ_M
$\mathcal{D}(\mathbf{x}')$
$\mathbf{N}'$
$\mathbf{M}'$
κ'_M
$\mathcal{M}'$
$\kappa'_{\mathcal{M}}$
$\mathcal{D}(\mathbf{x}') = \mathcal{T}(\mathcal{D}(\mathbf{x}), \mathbf{\Omega}, \mathbf{T})$

Figure 5: Mapping between $\mathcal{D}(\mathbf{x}) \in \mathcal{S}$ and $\mathcal{D}(\mathbf{x}') \in \mathcal{S}'$.

- $\mathcal{D}(\mathbf{x})$ and $\mathcal{D}(\mathbf{x}')$ are continuous functions of $\mathcal{S}$ and $\mathcal{S}'$ [9].
- Initial estimates of $\mathbf{\Omega}$ and $\mathbf{T}$, $\hat{\mathbf{\Omega}}$ and $\hat{\mathbf{T}}$, can often be determined through operational constraints, e.g. motion parameters provided by the manipulator system used to position the sensor, or specific knowledge about how the sensor or object can move in relation to each other. These further serve to limit the space of motion parameters that must be searched.

If $\mathbf{\Omega}$ and $\mathbf{T}$ are such that $\mathbf{x}' \in \mathcal{S}'$, then the foregoing suggests the following. For a given $\mathcal{D}(\mathbf{x})$ the corresponding $\mathcal{D}(\mathbf{x}')$ can be found without having to explicitly determine correspondence by minimizing the following continuous functional,

$$\min_{\mathbf{\Omega T}} \|\mathcal{D}(\mathbf{x}') - \mathcal{T}(\mathcal{D}(\mathbf{x}), \mathbf{\Omega}, \mathbf{T})\|. \tag{5}$$

However for such to be the case we must first ensure the existence and uniqueness of $\mathcal{D}(\mathbf{x})$ and $\mathcal{D}(\mathbf{x}')$.

First, in order for $\mathcal{D}(\mathbf{x})$ to exist, the surface $\mathcal{S}$ at $\mathbf{x}$ must be C^2 differentiable. Second, the surface must be such that $M_{\mathbf{x}}$ and $\mathcal{M}_{\mathbf{x}}$ have unique directions, that is, $\mathbf{x}$ is not an umbilic point on $\mathcal{S}$. Although these conditions ensure that $\mathcal{D}(\mathbf{x})$ exists and is uniquely directed at a point $\mathbf{x}$, this does not guarantee that no other point on $\mathcal{S}$ exists with similar attributes. One way of visualizing this is by means of the Gaussian sphere, a mapping of the unit surface normals of $\mathcal{S}$ to the center of a unit sphere $\mathcal{G}$ (Figure 6) [24]. In the example shown (Hilbert & Cohn-Vossen 52, p199) $\mathcal{S}$ is a bell-surface consisting of an upper elliptical region and a lower hyperbolic region separated by a parabolic curve along the circle of latitude shown.

Within each elliptic and hyperbolic region the orientation of the surface $\mathcal{S}$ expressed by $N_{\mathbf{x}}$ is unique since normals cannot be parallel on a curved surface for which $\kappa_{M_{\mathbf{x}}} \times \kappa_{\mathcal{M}_{\mathbf{x}}} \neq 0$. The same true for the directions of curvature $M_{\mathbf{x}}$ and $\mathcal{M}_{\mathbf{x}}$, provided that in addition $\kappa_{M_{\mathbf{x}}} \neq \kappa_{\mathcal{M}_{\mathbf{x}}}$. However, consider the case of a region that contains both elliptic and parabolic points, e.g. the region bounded by points 1-8 in Figure 6a. Notice that the mapping of this region onto the Gaussian sphere shown in Figure 6b is not unique. The region is folded over the parabolic line that separates it into elliptic and hyperbolic sections, leading to ambiguity in computing correspondence. To ensure that this does not occur, comparisons are restricted

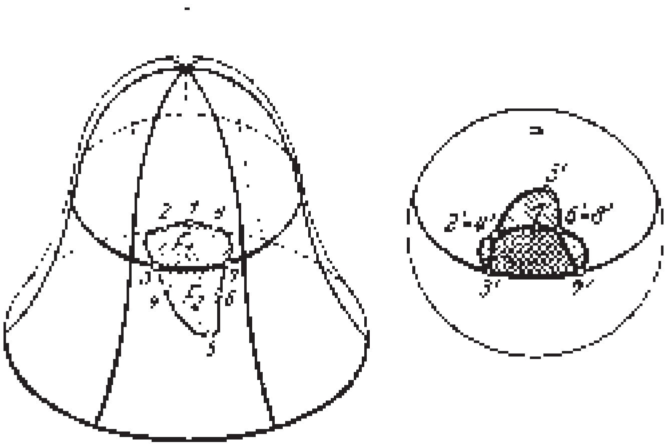


Figure 6: (a) A bell surface containing both elliptic and hyperbolic points. (b) Mapping of the bell surface onto the Gaussian sphere $\mathcal{G}$ (From Hilbert & Cohn-Vossen 52, p. 199).

to surfaces having Gaussian curvatures (the product of $\kappa_{\mathcal{M}}$ and κ_M) of the same sign. Now it is still possible that $\mathcal{S}$ may be composed of a number of elliptic or hyperbolic patches with similar attributes that can ultimately confound any matching process. The only way in which correspondence can still be estimated locally, other than assuming the uniqueness of each patch on $\mathcal{S}$ (not a very realistic assumption), is to assume that initial estimates $\hat{\mathbf{\Omega}}$ and $\hat{\mathbf{T}}$ can be determined which limit the aperture of correspondence for $\mathbf{x}$ to the patch in $\mathcal{S}'$ containing $\mathbf{x}'$.

In this paper we consider two methods of determining $\hat{\mathbf{\Omega}}$ and $\hat{\mathbf{T}}$. The first assumes that $\hat{\mathbf{\Omega}}$ and $\hat{\mathbf{T}}$ are provided by the sensor positioning system. Referring back to Figure 1, $\hat{\mathbf{\Omega}}$ and $\hat{\mathbf{T}}$ can usually be computed from displacements of the robot end-effector. Where such information is not available, a second (default) method is used that assumes a limit on the relative displacement between $\mathbf{x}$ and $\mathbf{x}'$ due to a limit on the relative velocities between camera and object. In other words, if $I(x, y, t)$ are the coordinates of $\mathbf{x}$, then $I(x, y, t+1)$ is assumed to lie on the patch in $\mathcal{S}'$ containing $\mathbf{x}'$. The value of $\hat{\mathbf{\Omega}}$ is then calculated from the difference in orientations between the surface normals at $I(x, y, t)$ and $I(x, y, t+1)$ respectively, and the translation component $\hat{\mathbf{T}}$ is set to $[0, 0, 0]$.

2.2 Similarity Functional

In addition to existence and uniqueness considerations, the similarity functional defined in equation (5) should also be convex to be of practical benefit. We now demonstrate that such is the case provided that certain restrictions are met. However, rather than apply (5) directly we introduce the following modification,

$$\min_{\mathbf{\Omega T}} \sum_i \|\mathcal{D}_i(\mathbf{x}') - \mathcal{T}(\mathcal{D}_i(\mathbf{x}), \mathbf{\Omega}, \mathbf{T})\|, \tag{6}$$

which improves the robustness of the measure by extending it over a larger local neighbourhood that encompasses neighbouring frames. If (6) can be shown to be convex, then $\mathbf{\Omega}$ and

$\mathbf{T}$ can efficiently be determined using an appropriate gradient descent procedure from the starting values $\hat{\mathbf{\Omega}}$ and $\hat{\mathbf{T}}$. Let $\lambda \subset \mathcal{S}$ be a patch containing a point $\mathbf{x}$ and a set of frames $\mathcal{D}_i(\mathbf{x})$ that describe the local neighbourhood of $\mathbf{x}$. The patch λ is now displaced according to $\mathbf{\Omega}$ and $\mathbf{T}$. Specifically, we observe that if

1. $\hat{\mathbf{\Omega}}$ and $\hat{\mathbf{T}}$ are such that the point corresponding to $\mathbf{x}$ on $\mathcal{S}'$ lies somewhere on the image of λ on $\mathcal{S}'$, λ',
2. λ meets existence and uniqueness requirements with respect to $\mathcal{D}_i(\mathbf{x})$,
3. λ is completely elliptic or hyperbolic,

then a similarity functional, $D_{\Omega T}$, corresponding to (6) can be found that is convex in the vicinity of the true values of $\mathbf{\Omega}$ and $\mathbf{T}$. We define $D_{\Omega T}$ as follows:

$$D_{\Omega T} = \sum_i \left\{ 3 + \frac{|\kappa_{M_{\mathbf{x}_i}} - \kappa_{M_{\mathbf{x}'_i}}|}{|\kappa_{M_{\mathbf{x}_i}}| + |\kappa_{M_{\mathbf{x}'_i}}|} + \frac{|\kappa_{\mathcal{M}_{\mathbf{x}_i}} - \kappa_{\mathcal{M}_{\mathbf{x}'_i}}|}{|\kappa_{\mathcal{M}_{\mathbf{x}_i}}| + |\kappa_{\mathcal{M}_{\mathbf{x}'_i}}|} \right. \\ \left. - (M_{\mathbf{x}i} \cdot M_{\mathbf{x}'i})^2 - (\mathcal{M}_{\mathbf{x}i} \cdot \mathcal{M}_{\mathbf{x}'i})^2 - (N_{\mathbf{x}i} \cdot N_{\mathbf{x}'i})^2 \right\} \quad (7)$$

where $(\kappa_{M_{\mathbf{x}_i}}, \kappa_{\mathcal{M}_{\mathbf{x}_i}}, M_{\mathbf{x}i}, \mathcal{M}_{\mathbf{x}i}, N_{\mathbf{x}i})$, and $(\kappa_{M_{\mathbf{x}'_i}}, \kappa_{\mathcal{M}_{\mathbf{x}'_i}}, M_{\mathbf{x}'i}, \mathcal{M}_{\mathbf{x}'i}, N_{\mathbf{x}'i})$ are the components of $\mathcal{D}_i(\mathbf{x})$ and $\mathcal{D}_i(\mathbf{x}')$ respectively. The functional is essentially a direct expansion of Equation (6), but implemented to yield a value of 0 for optimal values of $\mathbf{\Omega}$ and $\mathbf{T}$ through an appropriate choice of norms.

In order to prove this assertion it is necessary to substitute analytical expressions for κ_M, $\kappa_{\mathcal{M}}$, M_i, $\mathcal{M}_i$, N_i in terms of a parabolic quadric of the form $h(u,v) = au^2 + buv + cv^2$ into (7) and show that

$$\frac{\partial D_{\Omega T}}{\partial \mathbf{\Omega}} = \frac{\partial D_{\Omega T}}{\partial \mathbf{T}} = 0,$$

and that

$$\begin{pmatrix} \frac{\partial^2 D_{\Omega T}}{\partial \mathbf{\Omega}^2} & \frac{\partial^2 D_{\Omega T}}{\partial \mathbf{\Omega} \partial \mathbf{T}} \\ \frac{\partial^2 D_{\Omega T}}{\partial \mathbf{T} \partial \mathbf{\Omega}} & \frac{\partial^2 D_{\Omega T}}{\partial \mathbf{T}^2} \end{pmatrix} \quad (8)$$

is positive-definite. This analysis is quite complex because of the forms involved and we have as yet been unable to derive a complete analytic proof. However, we have investigated the behaviour of this functional through extensive numerical simulation.

Intuitively on examination of the mapping of $\mathcal{S}$ on the Gaussian sphere $\mathcal{G}$, one would expect (7) to vary monotonically since the orientations N_i and directions M_i and $\mathcal{M}_i$ vary monotonically in the range $[0, \pi]$. The same holds true for κ_M and $\kappa_{\mathcal{M}}$ since they are in turn related to the variations of N_i. This expectation is confirmed by our simulation results, some of which are shown below. In each of these experiments $D_{\Omega T}$ is sampled by selecting a 5×5 neighbourhood from an elliptic or hyperbolic surface, applying the transformation according to $\mathbf{\Omega}$ and $\mathbf{T}$, and comparing the transformed result against the original surface from which it was sampled according to (7).

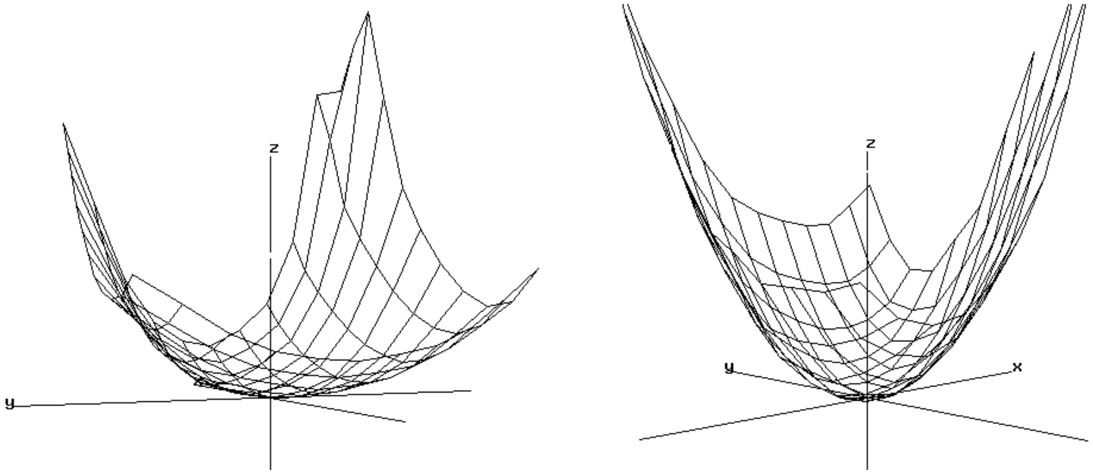


Figure 7: Convexity of $D_{\Omega T}$ as a function of (a) translation along the X and Y axes and (b) rotation about the X and Z axes for an elliptic surface.

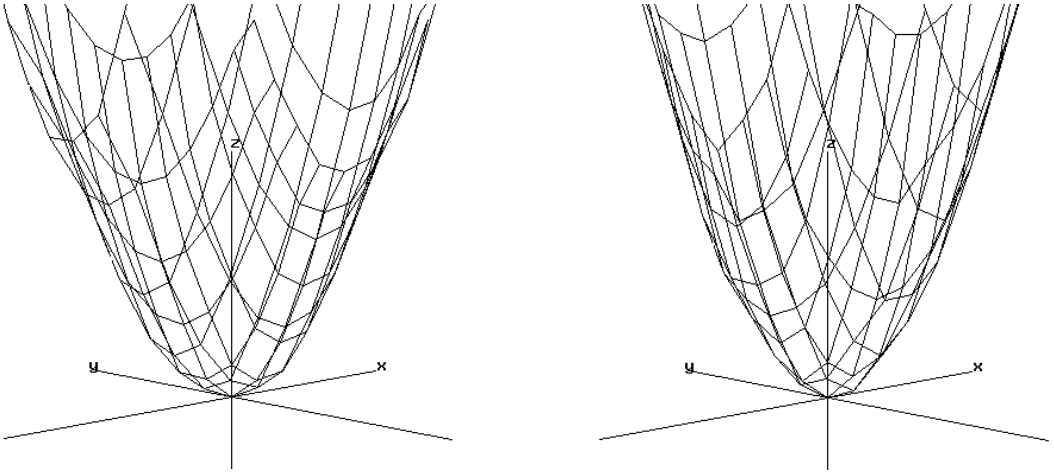


Figure 8: Convexity of $D_{\Omega T}$ as a function of (a) translation along the X and Y axes and (b) rotation about the X and Z axes for a hyperbolic surface.

In the first example shown in Figure 7a, $D_{\Omega T}$ is plotted against translations along the X and Y axes with all other parameters held constant. The same behaviour is also apparent in Figure 7b which shows $D_{\Omega T}$ plotted against rotations about the X and Z axes. In both cases, the location of the minimum corresponds *exactly* to the correct location of the selected neighbourhood. The above experiments are now repeated for a hyperbolic surface and the results shown below in Figures 8a and 8b. Again, the same results are observed.

3 Implementation of the Local Algorithm

The results of the previous section suggest an algorithm for the local recovery of $\mathbf{\Omega}$ and $\mathbf{T}$ at a point $\mathbf{x}$. It is assumed that the input to this process consists of two range images $I(i,j,t)$ and $I(i,j,t+1)$ corresponding to surfaces $\mathcal{S}$ and $\mathcal{S}'$ respectively. In addition, it is assumed that the augmented Darboux frames $\mathcal{D}(\mathbf{x})$ and $\mathcal{D}(\mathbf{x}')$ can be computed for each discrete sample point $\mathbf{x}$ and $\mathbf{x}'$ in $I(i,j,t)$ and $I(i,j,t+1)$. The following steps are performed for each point $\mathbf{x}$ for which $\mathbf{\Omega}$ and $\mathbf{T}$ are required.

1. Determine $\hat{\mathbf{\Omega}}$ and $\hat{\mathbf{T}}$ from either manipulator displacements or velocity limit assumptions. It is assumed that $D_{\Omega T}$ is convex in the vicinity of $\hat{\mathbf{\Omega}}$ and $\hat{\mathbf{T}}$.

2. Minimize $D_{\Omega T}$ for each parameter of $\mathbf{\Omega}$ and $\mathbf{T}$, i.e. θ_x, θ_y, θ_z, T_x, T_y, and T_z. Because $D_{\Omega T}$ is assumed to be convex, this computation can be performed efficiently.

3. Apply the resulting $\mathbf{\Omega}$ and $\mathbf{T}$ to $\mathcal{D}(\mathbf{x})$ to obtain $\mathcal{D}(\mathbf{x}')$. Validate the result by verifying that $\mathcal{D}(\mathbf{x})$ and $\mathcal{D}(\mathbf{x}')$ have non-zero Gaussian curvatures of the same sign.

We now elaborate on some of the technical details required for the algorithm to function properly.

3.1 Computing $\mathcal{D}(\mathbf{x})$ and $\mathcal{D}(\mathbf{x}')$

It is essential that the directional properties represented by $N_{\mathbf{x}}$, $M_{\mathbf{x}}$, and $\mathcal{M}_{\mathbf{x}}$ at each point $\mathbf{x}$ in $I(i,j)$ be recovered with suitable accuracy. More specifically, it is the variation in the orientations and directions of $\mathcal{D}_i(\mathbf{x})$ in the local neighbourhood of $\mathbf{x}$ that are of importance. The problem of recovering orientation and curvature has been investigated by a number of authors. A common approach is to locally approximate $\mathcal{S}$ and estimate the components of $\mathcal{D}(\mathbf{x})$ from the parameters of the approximation [3, 10, 20, 22]. Various techniques can be used to optimize estimation in the presence of noise, but they largely fall short when it comes to estimating directions [38]. For this reason we apply the curvature consistency algorithm [38, 21] to precisely recover $\mathcal{D}(\mathbf{x})$ and $\mathcal{D}(\mathbf{x}')$ for each discrete sample in $I(i,j,t)$ and $I(i,j,t+1)$ respectively.

The importance of this pre-processing stage is illustrated by Figure 9 which shows several slices through $D_{\Omega T}$ as a function of different numbers of iterations of curvature consistency for a single 5×5 neighbourhood in Figure 2. Figures 9a to 9d correspond to 0, 1, 3, and 5

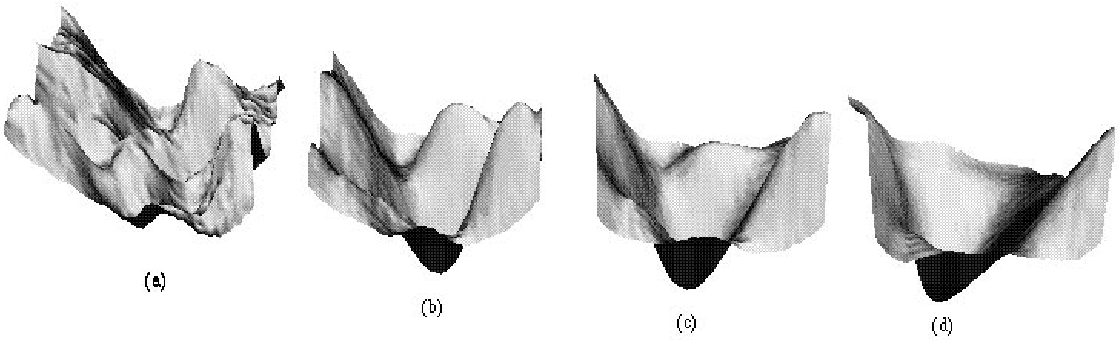


Figure 9: The effect of curvature consistency on the convexity of $D_{\Omega T}$. (a) 0 iterations (b) 1 iteration (c) 3 iterations (d) 5 iterations.

iterations of curvature consistency respectively. Without curvature consistency (Figure 9a) it is not only difficult to detect but also difficult to localize the global minimum. Furthermore the result is not robust to perturbations due to noise. With successive iterations, however, the minimum becomes increasingly more distinct and better localized to the point where a simple gradient descent approach suffices. We will not go into the details of the curvature consistency algorithm here, but will summarize the basic ideas for completeness. The reader is referred to [38, 21, 18] for specific details.

The overall procedure consists of two stages:

1. Initial estimates of $\mathcal{D}(\mathbf{x})$ and $\mathcal{D}(\mathbf{x}')$ are computed for each $\mathbf{x}$ and $\mathbf{x}'$ respectively. This is done by fitting a quadric patch to the neighbourhood of each discrete sample point and then computing the components of $\mathcal{D}(\mathbf{x})$ and $\mathcal{D}(\mathbf{x}')$ from the quadric parameters. The approach is essentially that described by [3].

2. The curvature consistency algorithm [38, 18] is then applied to these initial estimates.

The algorithm is implemented as an iterative filter which can be explained with the aid of Figure 10. Two frames are shown, $\mathcal{D}_P$ at point P and $\mathcal{D}_Q$ at a neighbouring point Q. Now if the surface is locally continuous in the vicinity of each point[2], then one can transport $\mathcal{D}_{Q_\alpha}$ at Q along S to P and obtain $\mathcal{D}_{P_\alpha}$. In effect $\mathcal{D}_{P_\alpha}$ is an extrapolation of the surface at Q to P according to the transport model defined by S. Formally, this mechanism is referred to as parallel transport [9]. We use this transport model to enforce the constraint of locally constant curvature [34]. By performing this operation for each neighbour of P one obtains a set of frames, $\mathcal{D}_{P_\alpha}$, each providing an estimate of P from its associated neighbour.

The algorithm functions by iteratively applying transport and updating $\mathcal{D}_P$ with a maximum likelihood estimate computed from $\mathcal{D}_{P_\alpha}$. Technical details regarding the updating functions for $\mathcal{D}_P$ in terms of $\mathcal{D}_{P_\alpha}$ along with the transport model are described in [38] and [21]. At each iteration a residual error is computed and summed for each frame on S. The process is allowed to iterate until the change in this error converges to a threshold value, generally within 5 to 10 iterations [21].

[2] The case of local discontinuities is discussed in [21].

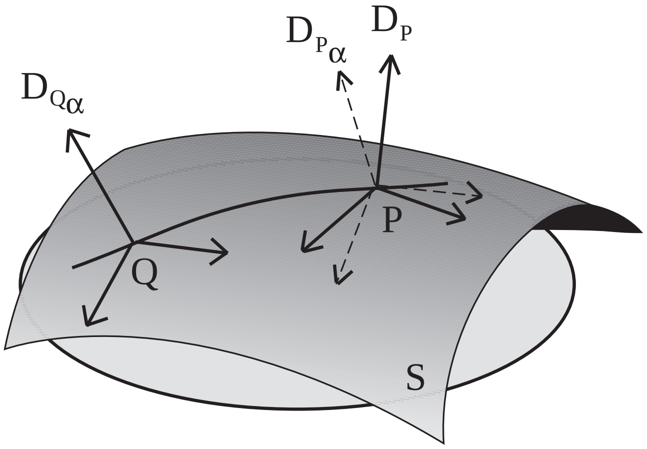


Figure 10: Local extrapolation using a quadric surface of the form $au^2 + buv + cv^2$.

3.2 Determining $\mathbf{\Omega}$ and $\mathbf{T}$

The convexity of (7) allows the parameters of $\mathbf{\Omega}$ and $\mathbf{T}$ to be determined in gradient descent fashion. An efficient strategy is to apply a Newton-Rhapson procedure to equation (7) on each of the 6 parameters. Recall that the effect of varying $(\theta_x, \theta_y, \theta_z, T_x, T_y, T_z)$ is to select a new local neighbourhood in $\mathcal{S}'$. The procedure converges very quickly as it is already constrained by having to map each $\mathbf{x}'$ onto $\mathcal{S}'$. The only parameter that must be supplied to the algorithm is the size of the local neighbourhood represented by the index variable i in (7). This is in turn dependent on a priori knowledge about the elasticity of $\mathcal{S}$ as the size of the neighbourhood is inversely proportional to it. Size also affects the robustness of (7) as larger neighbourhoods tend to have more unique structures. However, there is a limit to this size as one must ensure that the patch λ is completely visible in both range images.

Once a set of parameters has been found the $\mathbf{x}'$ corresponding to $\mathbf{x}$ is also determined. Two additional checks are made on the result to validate the correspondence. First, the sign of the Gaussian curvature at $\mathbf{x}'$ is compared to that at $\mathbf{x}$ as they are required to be identical. Second, the magnitude of $D_{\Omega T}$ is compared against a threshold to ensure a minimum similarity. The overall accuracy of local parameter recovery is very dependent on reconstruction error in the presence of noise. This is especially true for smooth surfaces with slowly varying curvature, and tends to either broaden the minimum of $D_{\Omega T}$ or else introduce additional minima for which a more complex minimization procedure is required. In the former case, the effect is to reduce the resolution of correspondence.

4 Refining Local Motion Estimates

The example shown earlier in Figure 3 demonstrated that for purposes of image reconstruction, local estimation of motion parameters is inadequate for precise surface recovery. In this section we show how the concept of motion consistency can be used to derive an iterative filtering procedure analogous to curvature consistency for the precise recovery of motion parameters.

Motion consistency can be thought of as a temporal analog of curvature consistency. Even though surfaces can undergo deformations as they move, motions, and thus motion parameters, are constrained by the coupling between local surface elements. In curvature consistency the frames $\mathcal{D}_i(\mathbf{x})$ and $\mathcal{D}_i(\mathbf{x}')$ were iteratively filtered such that curvature variation was minimized according to the assumption of locally constant curvature [34, 38, 21]. A similar strategy is employed in the motion consistency filter, except that it is $\mathbf{\Omega}$ and $\mathbf{T}$ which are varied to preserve the local neighbourhood structure of $\mathbf{x}$.

While the final position of $\mathbf{x}$ depends on both $\mathbf{\Omega}$ and $\mathbf{T}$, the final orientation of the local neighbourhood at $\mathbf{x}$ depends solely on $\mathbf{\Omega}$. This provides the necessary insight into how to separate the problem. First, given values of $\mathbf{\Omega}$ and $\mathbf{T}$ for $\mathbf{x}$ and each of its neighbours Q_{1i} (Figure 11), we determine a maximum likelihood estimate for each $\mathbf{x}'$ that preserves the relative displacements between $\mathbf{x}$ and each Q_{1i}. $\mathbf{T}$ is then updated by adding the differences between $\mathbf{x}'$ and its updated value as an offset. Next, updated values for each $\mathbf{\Omega}$ are determined by using a similar strategy to find rotation matrices that preserve the relative orientation between $\mathbf{x}$ and each of its neighbours Q_{1i} upon transformation.

4.1 Updating T

Within the local neighbourhood of $\mathbf{x}$ we assume that motion is approximately rigid, i.e. that the motion of $\mathbf{x}$ and its near-neighbours can be described by the same $\mathbf{\Omega}$ and $\mathbf{T}$. Now consider a 3×3 neighbourhood centered at $\mathbf{x}$ consisting of the 8 points Q_{1i} (Figure 11). The position of $\mathbf{x}$ relative to any Q_{1i} is given by

$$\mathbf{p}_1 = \mathbf{q}_{1i} + \mathbf{r}_{1i}, \tag{9}$$

where $\mathbf{p}_1$ and $\mathbf{q}_{1i}$ are vectors corresponding to $\mathbf{x}$ and Q_{1i} in view 1, and $\mathbf{r}_{1i}$ is the relative displacement of point $\mathbf{x}$ from point Q_{1i}. Under the rigid motion assumption a similar relationship must exist upon application of $\mathbf{\Omega}$ and $\mathbf{T}$, i.e.,

$$\mathbf{p}_2 = \mathbf{q}_{2i} + \mathbf{r}_{2i}, \tag{10}$$

where the vectors $\mathbf{p}_2, \mathbf{q}_{2i}$ and $\mathbf{r}_{2i}$ are defined as before except for the indices indicating view 2.

The relative spatial arrangement defined by (10) suggests a method of updating the position of point $\mathbf{x}'$, given estimates of $\mathbf{\Omega}_i$ and $\mathbf{T}_i$ computed independently for each Q_{1i}. It follows that

$$\mathbf{q}_{2i} = \mathbf{\Omega}_i(\mathbf{q}_{1i} - \mathbf{T}_i). \tag{11}$$

Similarly, the relative displacement between points Q_{1i} and $\mathbf{x}$ in the coordinates of view 2 is given by

$$\mathbf{r}_{2i} = \mathbf{\Omega}_i \mathbf{r}_{1i}. \tag{12}$$

Then the position of $\mathbf{x}$ in view 2, i.e. $\mathbf{x}'$, as predicted by its neighbour Q_{1i} is therefore

$$\mathbf{p}_{2i} = \mathbf{\Omega}_i(\mathbf{q}_{1i} - \mathbf{T}_i) + \mathbf{\Omega}_i \mathbf{r}_{1i}. \tag{13}$$

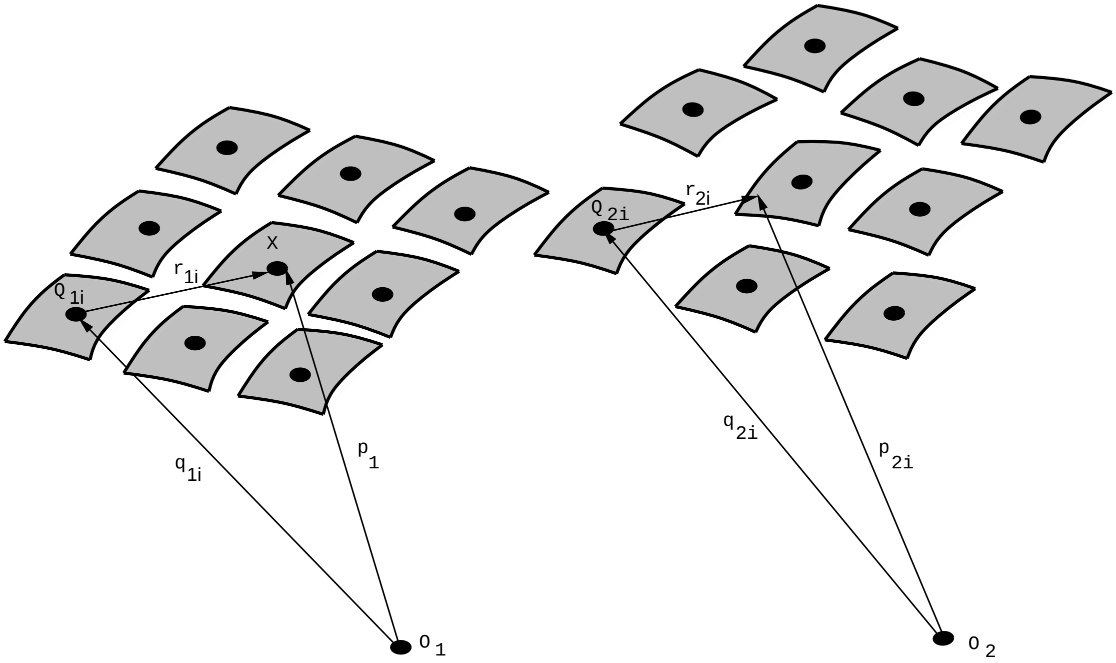


Figure 11: Principle of motion consistency for displacement. The relative displacements between $\mathbf{x}$ and its neighbours Q_{1i} (left) are corrupted under transformation (right). Displacements are represented by the vectors $\mathbf{r}_{1i}$ and $\mathbf{q}_{1i}$ for each neighbour. By applying that neighbour's associated motion parameters, one obtains the vectors $\mathbf{r}_{2i}$ and $\mathbf{q}_{2i}$ that provide an independent prediction for $\mathbf{x}'$. A maximum likelihood estimate of $\mathbf{x}'$ can be computed from the set of predictions.

A maximum likelihood estimate of $\mathbf{x}'$ can be obtained by computing the predicted position of $\mathbf{x}$ for each of its neighbours and taking a weighted mean, i.e.,

$$\hat{\mathbf{x}}' = \frac{\sum_{i=1}^{n} w_i \mathbf{p}_{2i}}{\sum_{i=1}^{n} w_i}, \tag{14}$$

where the w_i take into account the rigidity of the object and the distance between neighbouring points. The weights w_i and the size of the local neighbourhood determine the rigidity of the reconstructed surface. In our experiments a Gaussian weighting was used over neighbourhood sizes ranging from 3×3 to 11×11. $\mathbf{T}$ is then updated with an offset computed from the difference of $\mathbf{x}'$ and $\hat{\mathbf{x}}'$.

4.2 Updating $\mathbf{\Omega}$

The second part of the updating procedure seeks to enforce the relative orientation between point $\mathbf{x}$ and each of its neighbours Q_{1i}. However, under the assumption of locally rigid motion, this is equivalent to saying that each point in the local neighbourhood should have the same rotation component in its individual motion parameters. This would ensure that the relative orientations between $\mathbf{x}$ and Q_{1i} are preserved. The task, then, is to find the optimal set of rotations that best preserves the local orientation structure given $\mathbf{x}$ and Q_{1i}.

One difficulty in dealing with representations for rotation is that care must be taken to preserve the structure of the representation. For example, an operation such as an average of several rotation matrices has no physical meaning since it destroys important properties of the rotation matrix in the process. To get around this problem we convert estimates of $\mathbf{\Omega}_i$ into their equivalent *quaternions* [40, 39]. According to Euler's theorem of rigid body motion, a body having undergone any sequence of rotations is equivalent to a single rotation of that body through an angle θ about an axis $\mathbf{n}$. These parameters are easily determined from $\mathbf{\Omega}$. The unit quaternion $\mathbf{Q}$ is then defined as follows,

$$\mathbf{Q} = \left\{ \begin{array}{l} \sin(\theta/2)\ n_x \\ \sin(\theta/2)\ n_y \\ \sin(\theta/2)\ n_z \\ \cos(\theta/2) \end{array} \right\}. \tag{15}$$

The locus of unit quaternions traces out a unit sphere in 4-dimensional space. One of their desirable properties is that metric distance between two quaternions is given by the great circle distance on this sphere. For small distances, which is precisely the case for the rotations associated with each Q_{1i}, a reasonable approximation is given by their scalar product.

The computational task is now to estimate the quaternion corresponding to $\mathbf{x}$, $\mathbf{Q_x}$, given the quaternions $\mathbf{Q}_{1i}$ at each Q_{1i} by minimizing the distance between $\mathbf{Q_x}$ and each $\mathbf{Q}_{1i}$. If we define the quaternions $\mathbf{Q_x}$ and $\mathbf{Q}_{1i}$ respectively as

$$\mathbf{Q_x} = \left\{ \begin{array}{c} a \\ b \\ c \\ d \end{array} \right\}, \qquad \text{and} \qquad \mathbf{Q}_{1i} = \left\{ \begin{array}{c} x_i \\ y_i \\ z_i \\ s_i \end{array} \right\}, \tag{16}$$

then the distance between $\mathbf{Q}_{\mathbf{x}}$ and $\mathbf{Q}_{1i}$ is

$$D\left(\mathbf{Q}_{\mathbf{x}}, \mathbf{Q}_{1i}\right) = ax_i + by_i + cz_i + ds_i. \tag{17}$$

The goal is to maximize this functional subject to the constraint

$$a^2 + b^2 + c^2 + d^2 = 1. \tag{18}$$

Using Lagrange multipliers to enforce the constraint, the function to be maximized can be written as

$$E = \sum(ax_i + by_i + cz_i + ds_i + \lambda(a^2 + b^2 + c^2 + d^2 - 1)). \tag{19}$$

As was the case earlier in updating position, the terms of this sum can be weighted to take into account object deformability as well as the distance between $\mathbf{x}$ and Q_{1i}, i.e.,

$$E = \sum(w_i(ax_i + by_i + cz_i + ds_i) + \lambda(a^2 + b^2 + c^2 + d^2 - 1)). \tag{20}$$

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

$$\frac{\delta E}{\delta a} = \sum(w_i x_i + 2a\lambda) = 0, \tag{21}$$

$$\frac{\delta E}{\delta b} = \sum(w_i y_i + 2b\lambda) = 0, \tag{22}$$

$$\frac{\delta E}{\delta c} = \sum(w_i z_i + 2c\lambda) = 0, \tag{23}$$

$$\frac{\delta E}{\delta d} = \sum(w_i s_i + 2d\lambda) = 0, \tag{24}$$

$$\frac{\delta E}{\delta \lambda} = \sum(a^2 + b^2 + c^2 + d^2 - 1) = 0, \tag{25}$$

which after simplification yields

$$a = \pm\frac{\sum w_i x_i}{\sqrt{(\sum w_i x_i)^2 + (\sum w_i y_i)^2 + (\sum w_i z_i)^2 + (\sum w_i s_i)^2}}, \tag{26}$$

$$b = \pm\frac{\sum w_i y_i}{\sqrt{(\sum w_i x_i)^2 + (\sum w_i y_i)^2 + (\sum w_i z_i)^2 + (\sum w_i s_i)^2}}, \tag{27}$$

$$c = \pm\frac{\sum w_i z_i}{\sqrt{(\sum w_i x_i)^2 + (\sum w_i y_i)^2 + (\sum w_i z_i)^2 + (\sum w_i s_i)^2}}, \tag{28}$$

$$d = \pm\frac{\sum w_i s_i}{\sqrt{(\sum w_i x_i)^2 + (\sum w_i y_i)^2 + (\sum w_i z_i)^2 + (\sum w_i s_i)^2}}. \tag{29}$$

The two possible solutions correspond to two quaternions of opposite direction which, in fact, correspond to the same rotation. By convention $\mathbf{\Omega}$ is updated accordingly with the positive solution.

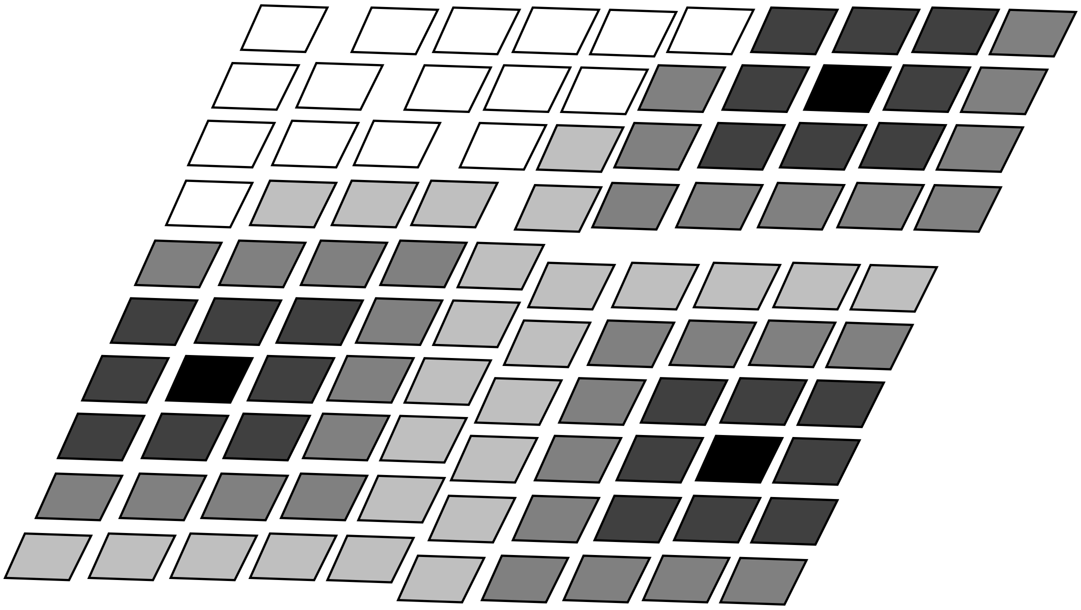
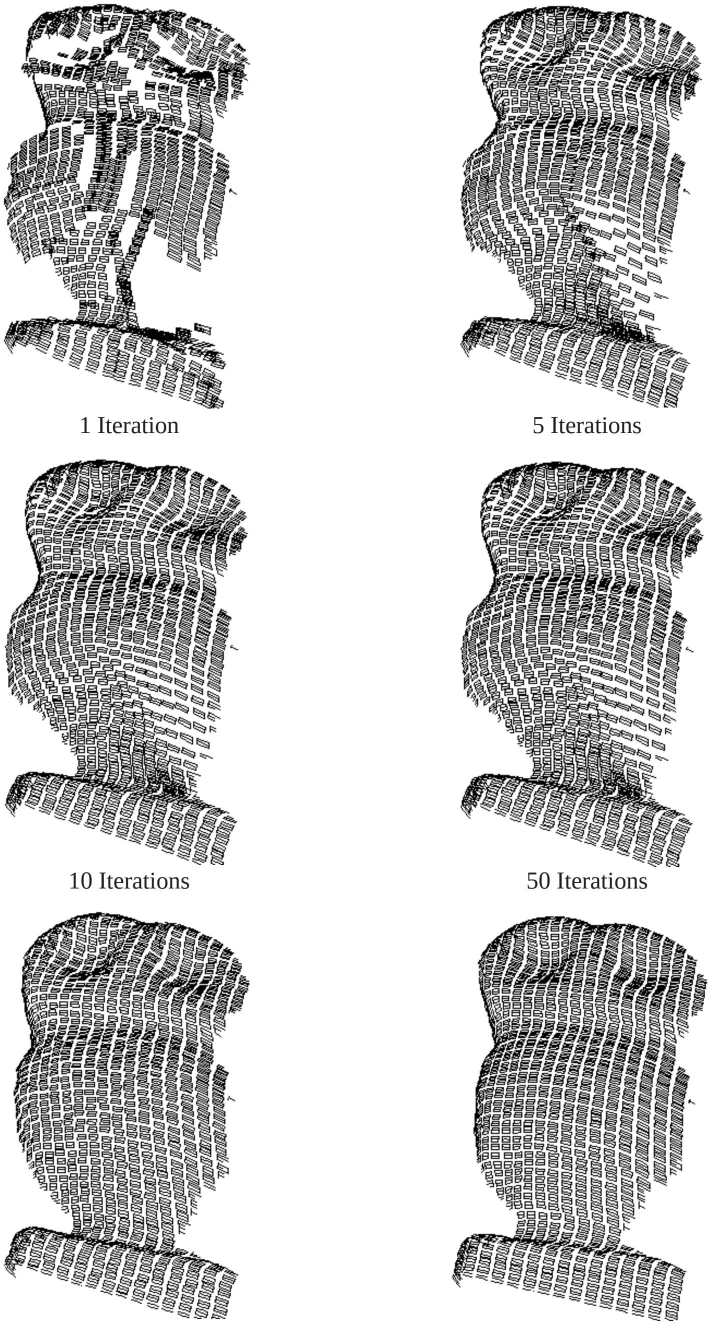


Figure 12: Interpolating motion parameters. The three black squares represent points on the surface for which local motion estimates are available. A grass fire algorithm is used to propagate these estimates outward until they meet up with other wavefronts. The motion consistency algorithm is then applied to smooth out the resulting field.

4.3 Interpolation and Refinement

The updating functions for $\mathbf{\Omega}$ and $\mathbf{T}$ are applied iteratively to the set of motion estimates obtained using the local algorithm. Since the effect of the motion consistency filter is to smoothly blend adjacent motions, this suggests a means of interpolating motion parameters over those portions of the surface for which no estimates are available. This is illustrated in Figure 12 which shows a surface consisting of three points (the black squares) for which motion estimates are available. A grass fire algorithm is used to propagate these estimates outward until they run into adjacent wavefronts generated by other the points. The motion consistency algorithm is then applied to smooth out the resulting field.

Figure 13 shows the effect of motion consistency on the owl surface shown earlier in Figure 3. In this example we have also shown how motion parameters are iteratively propagated over the surface in order to fill holes for which no estimates are available. At 0 iterations, the surface resulting from the fusion of Figure 2a and Figure 2b is sparse and noisy. With successive iterations, however, holes become filled and the surface becomes smooth in accordance with the real surface. At about 1000 iterations the process converges to the point where each coordinate has the same motion parameters, i.e. rigid-body motion. The number of iterations acts as a λ parameter, varying between the extremes of independent and rigid-body motion. Where objects are known to be rigid a priori, the process can be allowed to run its course to convergence (no further significant change in parameters). In practice, the number of iterations applied is data-dependent and is tied to the rigidity of the objects in question.



1 Iteration

5 Iterations

10 Iterations

50 Iterations

100 Iterations

1000 Iterations

Figure 13: Evolution of the surface of the owl as a function of varying degrees of motion consistency.

5 Experiments: Fusing Multiple Views

We now present the results of four experiments on fusing together range images from different viewpoints. The first two experiments address respectively the stability and robustness of the algorithm and its ability to deal with non-rigid correspondence. In the second two experiments, range images obtained with a laser rangefinding system are fused together from several different viewpoints using the experimental set-up shown earlier in Figure 1.

The laser rangefinding system is a National Research Council of Canada design [37] built in cooperation with McGill. It has a working field of view of approximately 1.0 m^3 with a resolution of $256 \times 256 \times 1024$ in the X, Y, and Z directions respectively. Accuracy is a function of the Z distance to the object. At closest approach, roughly 10.0 cm, accuracy is approximately 0.2 $mm/pixel$. At maximum distance, 100.0 cm, it falls off to about 1.0 $mm/pixel$. The rangefinder is mounted on the end-effector of a PUMA-560 robot inverted from the ceiling. Absolute positioning accuracy in Cartesian coordinates is on the order[3] of 1.0 cm. Consequently, $\mathbf{\Omega}$ and $\mathbf{T}$ cannot be accurately determined from changes in the position and orientation of the robot end-effector[4].

The procedure we use to fuse range images is as follows:

1. The curvature consistency algorithm is applied to each range image in the input sequence.
2. A set of token points is selected in $I(i,j,t)$ that spans $\mathcal{S}$.
3. The algorithm described in Section 3 is applied and $\mathbf{\Omega}$ and $\mathbf{T}$ estimated for each token point.
4. The motion consistency algorithm is applied to the result of Step 3.
5. Steps 2-4 are repeated for $I(i,j,t+1)\ldots I(i,j,t+n)$.
6. Using successive transformations, data points are mapped into a chosen coordinate frame $I(i,j,t+m), m \in [1,n]$.

The selection process begins with the tessellation of $I(i,j,t)$ with an odd sized grid where the size of each cell is inversely proportional to the elasticity of $\mathcal{S}$. Selection is made on the basis of the attributes associated with the frame $\mathcal{D}(\mathbf{x})$ corresponding to each point. For example, frames corresponding to local maxima and minima of curvature are good candidates for matching because they are easy to localize [19]. On the other hand, frames associated with low magnitudes of κ_M and $\kappa_{\mathcal{M}}$ or umbilic points where $\kappa_M = \kappa_{\mathcal{M}}$ are bad candidates as the surface has few distinguishing characteristics within their vicinity. Other regions to avoid are those where the surface normal N is near to being orthogonal to the direction of view, e.g. in the vicinity of an occluding contour, as these often correspond to regions of $\mathcal{S}$ that are changing rapidly [28].

[3] On a good day when the robot manages to stay in calibration.

[4] However this does provide an initial estimate that can be used by the motion estimation algorithm.

Accurate detection of occluding contours is also important because the latter can serve to delineate regions undergoing different motions. This is particularly important in the case of self-occlusion of an articulated object or occlusion between two (or more) different objects moving at different velocities. The degree to which occluding contours can be detected accurately is an important factor in the accuracy of surface reconstruction.

Once a suitable token point has been found, the set of frames $\mathcal{D}_i(\mathbf{x})$ comprising its local neighbourhood is used for matching against $\mathcal{S}'$. If a suitable token cannot be found, the parameters of the corresponding grid cell are interpolated as described in Section 4.3.

5.1 Correspondence with Additive Noise

In the first experiment two identical paraboloids ($z = 0.01x^2 + 0.005y^2$) are generated and displaced relative to each other with $\mathbf{\Omega} = (20^\circ_x, 10^\circ_y, 45^\circ_z)$ and $\mathbf{T} = (25_x, 15_y, -25_z)$. Each paraboloid is defined on a 100×100 grid and has a maximum height (z) of 50 pixels. The algorithm is applied with 30 iterations of curvature consistency using a 5×5 window and 5 iterations of motion consistency using a 10×10 tiling of the coordinate grid. These surfaces are relatively smooth and not highly curved, thus one would expect motion estimates to be very sensitive to additive noise. The algorithm is started at the correct solution, i.e. $\mathbf{\Omega} = (20^\circ_x, 10^\circ_y, 45^\circ_z)$ and $\mathbf{T} = (25_x, 15_y, -25_z)$, and allowed to run to termination.

Termination criteria are defined by convergence of the curvature and motion consistency algorithms respectively. In the case of the former, the algorithm operates by minimizing a functional form that is related to differences in $\mathcal{D}(\mathbf{x})$ under the assumption of locally constant curvature. The algorithm is terminated when the derivative of this functional falls below a user-specified threshold. It has been shown elsewhere that this algorithm converges rapidly, generally within 5-10 iterations [18, 38]. A similar termination criterion is used for motion consistency and is defined as the point at which $\frac{d\hat{\mathbf{\Omega}}}{dt}$ and $\frac{d\hat{\mathbf{T}}}{dt}$ fall below a preset threshold [18]. In practice the number of iterations applied for both algorithms is generally a function of sensor noise[5].

Results are shown in Table 1 for 4 different additive noise levels. Random variables are sampled from a standardized normal distribution, $N(0,1)$, and scaled by 0%, 1.4%, 2.7% and 5.4% of the maximum z range (50). The RMS error listed in the table corresponds to

$$Error_{rms} = \sqrt{\frac{1}{n}\sum_{i,j}(\mathbf{x}_{i,j} - \mathbf{x}'_{i,j})^2}, \tag{30}$$

where $\mathbf{x}'$ corresponds to the displacement of $\mathbf{x}$ incurred by errors in the transformation parameters $\mathbf{\Omega}$ and $\mathbf{T}$. The summation is applied to the n surface coordinates of the synthetic paraboloid. As expected, error increases linearly with noise (graceful degradation) up to the point where the noise level obscures the surface curvature. A qualitative description of this error is shown in Figure 14 where each result is displayed as an overlay of the two paraboloids.

[5] For the experiments involving laser rangefinder data, typical values are 5 iterations of curvature consistency followed by 100 iterations of motion consistency.

True Parameters $(\mathbf{\Omega}),(\mathbf{T})$	*Estimated Parameters* $(\hat{\mathbf{\Omega}}),(\hat{\mathbf{T}})$	*Noise*	*Error*
$(20^\circ_x, 10^\circ_y, 45^\circ_z)$, $(25_x, 15_y, -25_z)$	$(20.0^\circ_x,\ 9.1^\circ_y, 43.9^\circ_z)$, $(25.9_x, 14.4_y, -25.7_z)$	0%	1.93
$(20^\circ_x, 10^\circ_y, 45^\circ_z)$, $(25_x, 15_y, -25_z)$	$(23.4^\circ_x, 14.0^\circ_y, 45.2^\circ_z)$, $(20.2_x, 20.0_y, -25.7_z)$	1.4%	4.76
$(20^\circ_x, 10^\circ_y, 45^\circ_z)$, $(25_x, 15_y, -25_z)$	$(30.4^\circ_x, 8.92^\circ_y, 38.2^\circ_z)$, $(19.0_x, 35.0_y, -26.0_z)$	2.7%	11.19
$(20^\circ_x, 10^\circ_y, 45^\circ_z)$, $(25_x, 15_y, -25_z)$	$(-8.6^\circ_x,\ 8.3^\circ_y, 59.6^\circ_z)$, $(47.6_x,\ -4.1_y,\ 2.0_z)$	5.4%	18.5

Table 1: Error in determining correspondence as a function of additive noise. Noise is expressed as a percentage of the maximum range value (50 pixels). The error indicated is RMS, units = pixels.

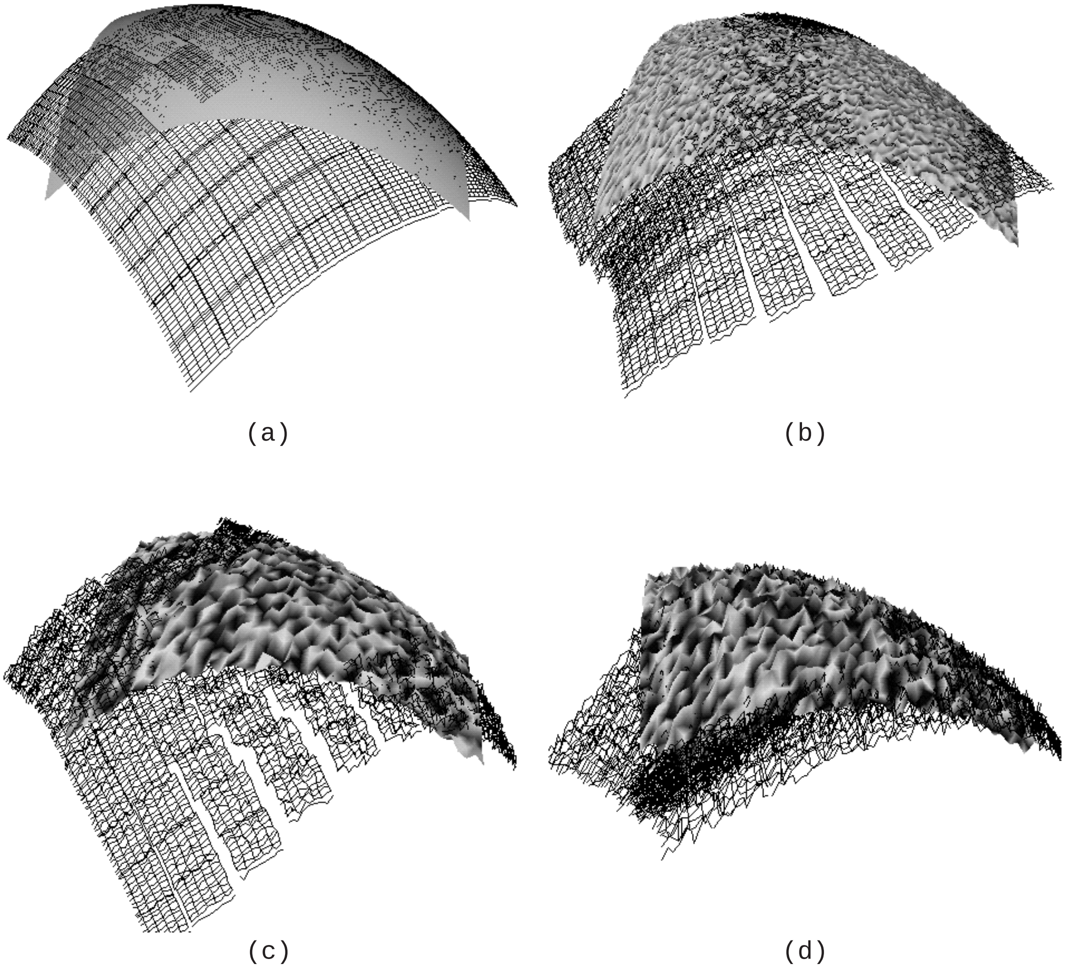


Figure 14: Effects of additive noise on the recovery of motion parameters for a synthetic paraboloid. Results are shown as overlays with one surface rendered as a mesh and the other as a shaded surface. (a) 0% additive noise (b) 1.4% additive noise (c) 2.7% additive noise (d) 5.4% additive noise.

5.2 Correspondence with Non-Rigid Surfaces

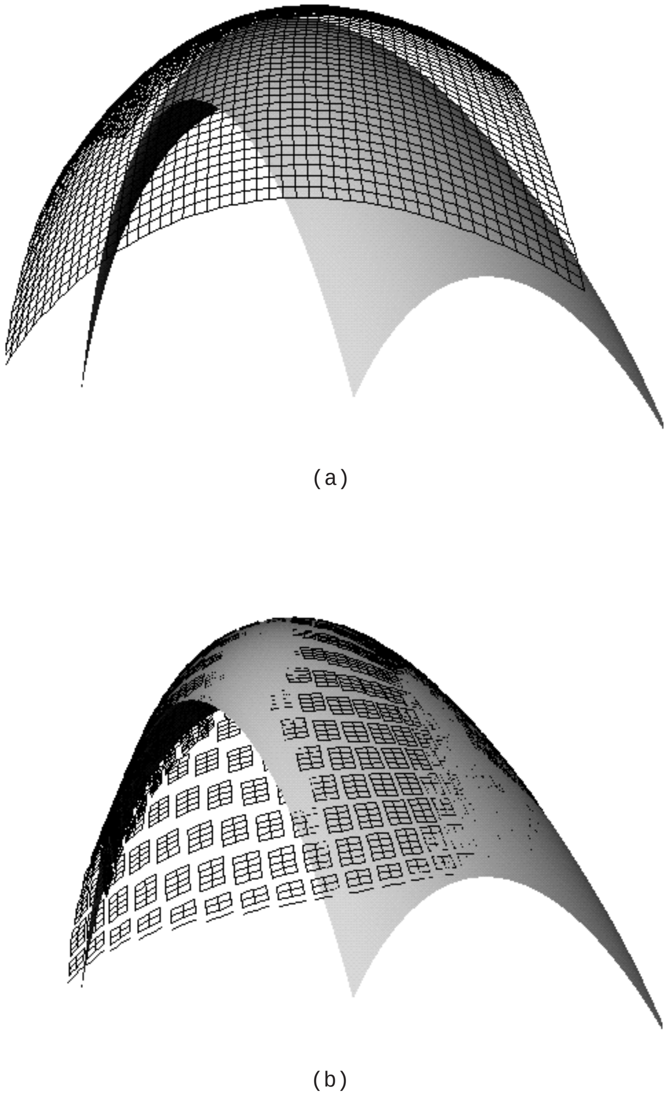
In the second experiment the algorithm is applied to two different paraboloids with equations $z_1 = 0.01x^2 + 0.005y^2$ and $z_2 = 0.02x^2 + 0.01y^2$ respectively which correspond to the deformation of a surface by bending and stretching. In addition the two surfaces are displaced relative to one another by $\mathbf{\Omega} = (20^\circ_x, 10^\circ_y, 45^\circ_z)$ and $\mathbf{T} = (25_x, 15_y, -25_z)$. The algorithm is applied with the same parameters as in the first experiment and the results shown in Figure 15 are obtained. Figure 15a shows the result corresponding to a rigid-body solution - there is no way that the deformation can be characterized in terms of a single set of motion parameters. The actual solution obtained in this case, $\mathbf{\Omega} = (16.4^\circ_x, 10.8^\circ_y, 54.0^\circ_z)$,$\mathbf{T} = (37.6_x, 16.3_y, -17.6_z)$, approximates the relative displacement between the two surfaces. In Figure 15b, motion parameters corresponding to each local deformation are estimated independently. The motion consistency algorithm ensures that local estimates blend smoothly and produces the result shown in the figure where the overlap between the two surfaces is on average within a single pixel.

5.3 Piecing Together a Mercedes

In the third experiment two views of a model car, Figures 16a and b, are shown rendered as shaded images. An initial estimate of the motion parameters, $\hat{\mathbf{\Omega}}$ and $\hat{\mathbf{T}}$ relating the two views is obtained from the robot. However due to the inaccuracies of the robot calibration and mechanical alignment, $\hat{\mathbf{\Omega}}$ and $\hat{\mathbf{T}}$ have uncertainties on the order of 10° and 10 mm respectively. Using $\hat{\mathbf{\Omega}}$ and $\hat{\mathbf{T}}$ as starting points, local motion parameters are estimated for each 5×5 pixel neighbourhood of the view corresponding to Figure 16a. The motion consistency filter is then applied to these local estimates, and the resulting motion parameters are then used to map the view corresponding to Figure 16b into the coordinates of the first view. The resulting surface is shown in Figure 17 with elements from the first view rendered as a shaded surface, and elements from the second view overlaid as a mesh pattern. We have verified that the accuracy of point placement is generally within a single pixel.

5.4 Reconstructing the Owl From Three Views

Figure 18 shows three views of the owl statue at (a) 0°, (b) 45°, and (c) 90° respectively. In this experiment all three views are mapped into a single composite surface. The owl is reconstructed by first mapping the 0° view into the 45° view and merging the two surfaces. Next the 45° view is mapped into the 90° view and these surfaces merged. Finally, the intermediate surface from the $0^\circ - 45^\circ$ views is mapped into the one corresponding to the $45^\circ - 90^\circ$ views and the two merged, resulting in a complete reconstruction that spans approximately 180°. The neighbourhood size corresponding to $\mathcal{D}_i(\mathbf{x})$ in this experiment is 7×7 and approximately 30 token points were selected automatically be the algorithm, which was implemented on a SUN 470. Solutions for $\mathbf{\Omega}$ and $\mathbf{T}$ for each token required approximately 10 seconds, excluding the time required for reconstruction. As presently implemented, the major bottleneck in the process is the curvature consistency algorithm, which requires several minutes for each 256×256 range image.



(a)

(b)

Figure 15: (a) Two surfaces corresponding to a deformation of a paraboloid with equation $z_1 = 0.01x^2 + 0.005y^2$ (opaque) into a second with equation $z_2 = 0.02x^2 + 0.01y^2$ (mesh). A rigid-body solution cannot bring these two surfaces into correspondence. (b) Local motion estimates suffice to characterize deformation - motion consistency ensures that they blend smoothly producing the result shown.

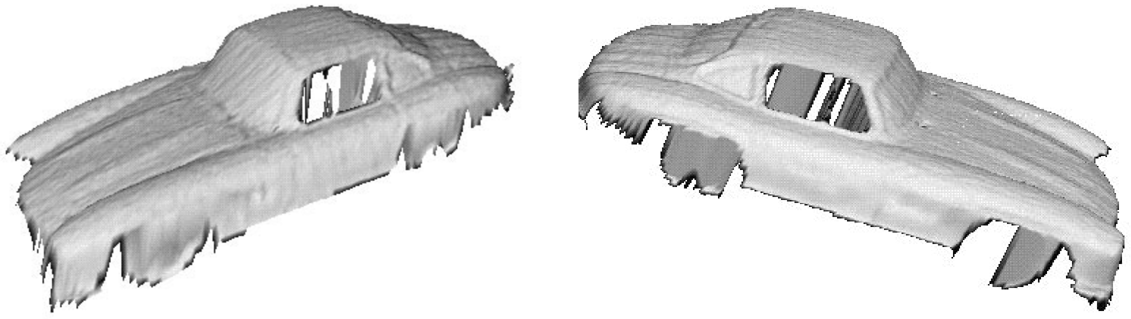


Figure 16: Two views of a model Mercedes obtained with a laser rangefinding system and rendered as shaded surfaces.

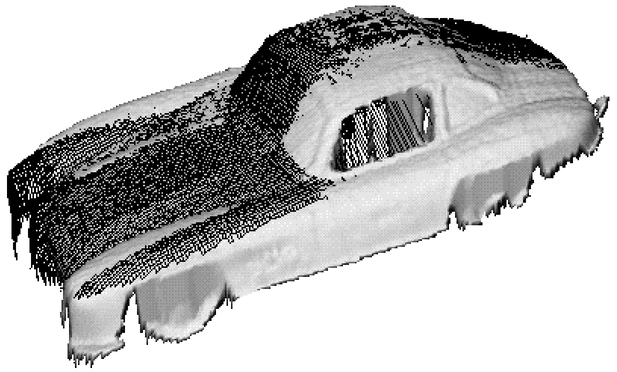


Figure 17: Two views of a model Mercedes fused into a single surface. Elements from the first view are rendered as a shaded surface and elements from the second view are overlaid as a mesh pattern to highlight the alignment of the two surfaces.

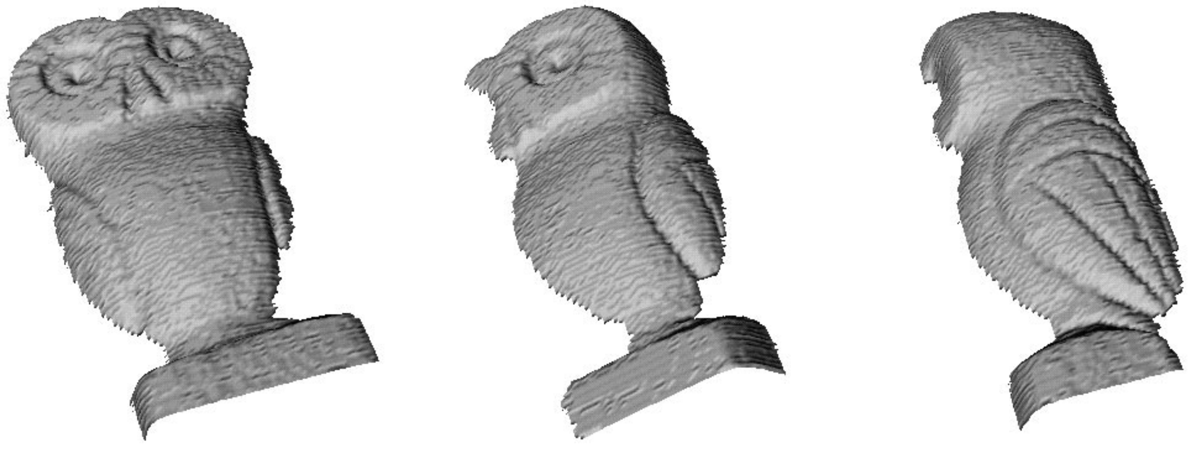


Figure 18: Laser rangefinder images of an owl statuette at (a) 0°, (b) 45° and (c) 90°. Resolution is 256×256 by 10 bits/rangel.

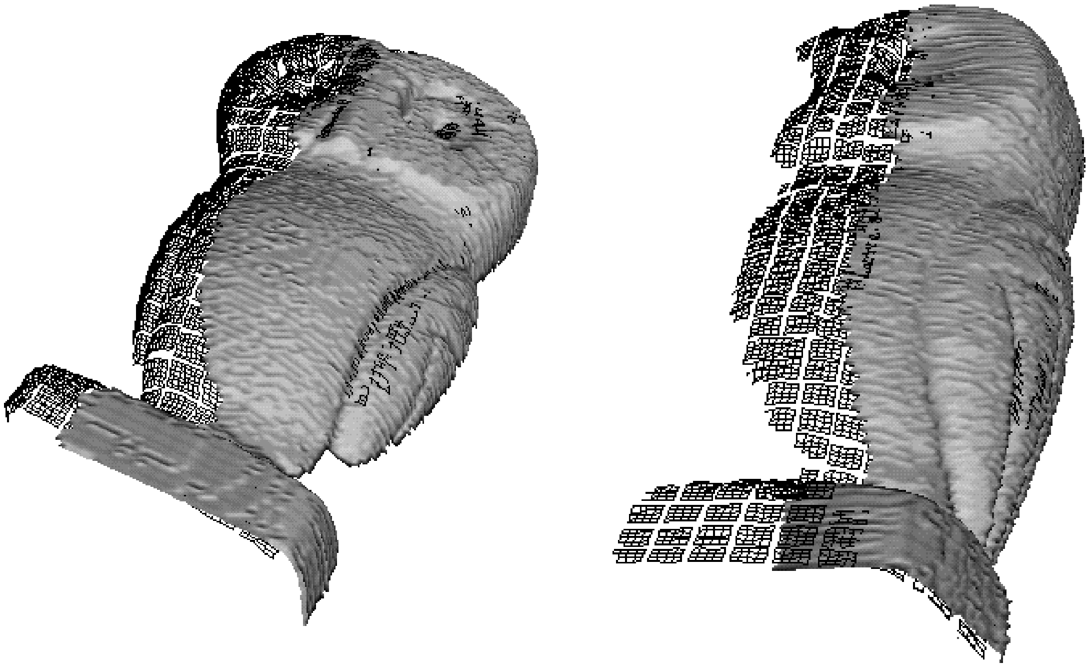


Figure 19: Reconstruction of two pairs of laser rangefinder images of the owl statuette. In each example the surface from the first view is rendered as a mesh and the surface from the second view as a shaded surface. Reconstruction of (a) the 0° and 45° views, (b) the 45° and 90° views.

The results of this experiment are shown in Figures 19, 20, and 21. In Figure 19a, the 0° view, shown as a mesh surface, is mapped into the coordinates of the 45° view which is rendered as a shaded surface. We have rendered the resulting surface from a different viewing position from which it can be compared against the three viewpoints shown in Figure 18. The alignment of the two surfaces is precise almost everywhere except near the edge of pedestal where the surface is slightly distorted. This is most likely due to the almost planar shape of the pedestal which makes the localization of surface features difficult. Nevertheless the results are quite good, especially in light of the large angle of rotation. Similar results are obtained for the mapping of the 45° view into the 90° view shown in Figure 19b.

Finally the surface corresponding to the complete reconstruction is shown rendered as a shaded image in Figure 20b. Next to it in Figure 20a is a laser rangefinder scan of the owl taken from the viewpoint at which the reconstructed surfaces are rendered. As can be seen, the three surface patches align quite well, especially considering that the surfaces of the object are quite smooth and the displacement angles large. This is further evident in Figure 21 which shows two enlarged views corresponding to the reconstructed surfaces in profile.

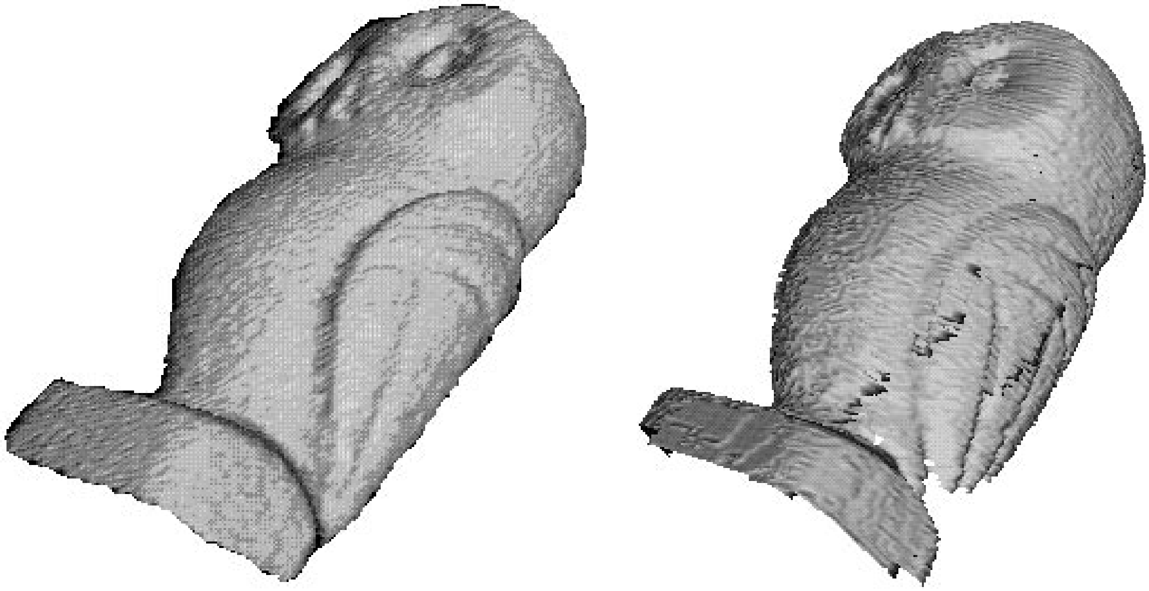


Figure 20: (a) A laser rangefinder image rendered as a shaded surface showing the owl from the viewpoint of the reconstructed surfaces shown in (b). (b) Reconstruction of three views of the owl taken at 0°, 45° and 90° and rendered as a shaded surface.

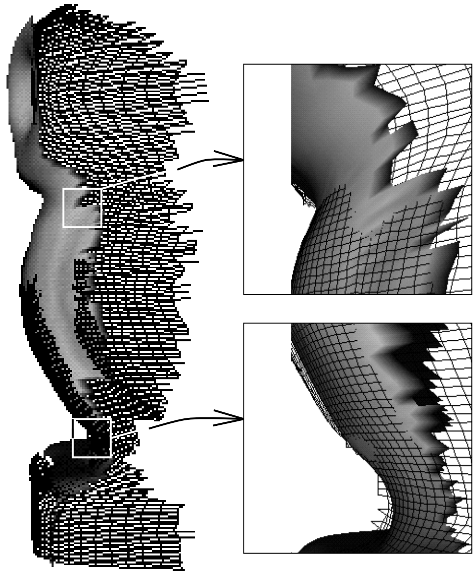


Figure 21: Two enlarged views corresponding to the profile of the reconstructed surfaces shown in Figure 20b. As in previous displays, overlapping surfaces are rendered using meshes and shading to highlight registration accuracy.

6 Conclusions

The ability to integrate information from different sources and vantage points is a recurring theme in the literature and still poses significant challenges in the realization of functional systems. In this paper we have described two algorithms that together provide a means of fusing a sequence of range images into a single 3-D representation of a scene. What makes this a particularly interesting problem is that surfaces are assumed to be piecewise-smooth, void of the kind of features such as lines and corners that are often used to determine correspondence in polyhedral scenes. In addition, the assumption of rigidity applies only to the local neighbourhood of a point on the surface, requiring that motion parameters be determined locally.

We showed how the local curvature structure of a surface could be used to determine motion parameters. In particular, we demonstrated that this could be accomplished by employing a similarity functional relating the differential geometry of two corresponding local neighbourhoods as a function of these parameters. Furthermore, by exploiting some well-known forms, i.e. the augmented Darboux frame $\mathcal{D}(\mathbf{x})$, we showed that a functional could be devised that was convex in the vicinity of the true parameters. This lead to an efficient local algorithm that recovers motion parameters in gradient descent fashion.

The second contribution of this work was a procedure, the motion consistency algorithm, for filtering local motion estimates according their expected variation on real surfaces. We showed that the algorithm significantly improved the recovery of local motion parameters and could also be used to interpolate motions over those portions of a surface for which no local estimates were available. For the case of rigid-body motion, the algorithm provides a novel means of arriving at consensus, i.e., combining local motion estimates spanning the surface into one single set of parameters that is both accurate and robust.

Together, the two algorithms provide a successful means of reconstructing a scene from several views, and we believe that the results presented justify our basic approach. The result shown in Figure 17 demonstrates that the procedure can take advantage of motion estimates provided by a positioning system and arrive at a positioning accuracy limited only by the sensor. A more telling result was shown in Figure 20, which concatenates information from three views spanning approximately 180° to an accuracy that is within a pixel over most of the surface.

Current work is aimed at improving the efficiency of the algorithms to the point were they operate closer to data acquisition speeds. However our experience to date suggests that the complexity of the procedure is well warranted against the results obtained.

References

[1] M. Asada and S. Tsuji. Inferring motion of cylindrical object from shading. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 240–245, Washington, D.C., June 1983. Computer Society of the IEEE, IEEE Computer Society Press.

[2] F. Bergholm. A theory on optical velocity fields and ambiguous motion of curves. In *Proceedings, 2ND International Conference on Computer Vision*, pages 165–176, Tampa, Florida, Dec. 1988. Computer Society of the IEEE, IEEE Computer Society Press.

[3] P. Besl and R. Jain. Segmentation through symbolic surface description. In *Proceedings IEEE Conf. Computer Vision and Pattern Recognition*, pages 77–85, Miami Beach, Florida, June 1986.

[4] A. Blake and A. Zisserman. *Visual Reconstruction.* MIT Press, Cambridge, Massachusetts, 1987.

[5] H. Chen and T. Huang. An algorithm for matching 3-d line segments with application to multiple-object motion estimation. In *Proceedings of the IEE Computer Society Workshop on Computer Vision*, pages 151–157, Miami Beach, Florida, Dec. 1987. Computer Society of the IEEE, IEEE Computer Society Press.

[6] N. Cui, J. Weng, and P. Cohen. Extended structure and motion analysis from monocular image sequences. In *Proceedings, 3RD International Conference on Computer Vision*, pages 222–229, Osaka, Japan, Dec. 1990. Computer Society of the IEEE, IEEE Computer Society Press.

[7] L. Davis. Contour-based motion estimation. *Computer Vision, Graphics, and Image Processing*, 23:313–326, Sept. 1983.

[8] C. Debrunner and N. Ahuja. A direct data approximation based motion estimation algorithm. In *Proceedings of the 10TH International Conference on Pattern Recognition*, pages 384–389, Atlantic City, New Jersey, June 1990. Computer Society of the IEEE, IEEE Computer Society Press.

[9] M. do Carmo. *Differential Geometry of Curves and Surfaces.* Prentice-Hall, Inc., Englewood Cliffs,New Jersey, 1976.

[10] T. Fan, G. Medioni, and R. Nevatia. Description of surfaces from range data using curvature properties. In *Proceedings IEEE Conf. Computer Vision and Pattern Recognition*, pages 86–91, Miami Beach, Florida, June 1986.

[11] J. Fang and T. Huang. Estimating 3-d movements of a rigid object: Experimental results. In *Proc. International Joint Conference on Artificial Intelligence*, pages 1033–1037, 1983.

[12] J. Fang and T. Huang. Solving three-dimensional small-rotation motion equations. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 253–258, Washington, D.C., June 1983. Computer Society of the IEEE, IEEE Computer Society Press.

[13] J. Fang and T. Huang. Some experiments on estimating the 3-d motion parameters of a rigid body from two consecutive image frames. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(5):545–554, Sept. 1984.

[14] F. Ferrie and J. Lagarde. On computing stable surface descriptions from range images. In *Proceedings 5th International Conference on Image Analysis*, Positano, Italy, September 20-22 1989.

[15] F. Ferrie and J. Lagarde. Curvature consistency improves local shading analisys. In *Proceedings 10th International Conference on Pattern Recognition*, pages 70–76, Atlantic City, New Jersey, jun 1990.

[16] F. Ferrie, J. Lagarde, and P. Whaite. Darboux frames, snakes, and super-quadrics: Geometry from the bottom-up. In *Proceedings IEEE Workshop on Interpretation of 3D Scenes*, pages 170–176, Austin, Texas, Nov. 27-29 1989.

[17] F. Ferrie, J. Lagarde, and P. Whaite. Recovery of volumetric object descriptions from laser rangefinder images. In *Proceedings First European Conference on Computer Vision*, Antibbes, France, April 1990.

[18] F. Ferrie, J. Lagarde, and P. Whaite. Darboux frames, snakes, and super-quadrics: Geometry from the bottom-up. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(8), Aug. 1993. to appear.

[19] F. Ferrie and M. Levine. Integrating information from multiple views. In *Proceedings of the IEE Computer Society Workshop on Computer Vision*, pages 117–122, Miami Beach, Florida, Dec. 1987. Computer Society of the IEEE, IEEE Computer Society Press.

[20] F. Ferrie and M. Levine. Deriving Coarse 3D Models of Objects. In *IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition*, pages 345–353, University of Michigan, Ann Arbor, Michigan, June 1988.

[21] F. Ferrie, S. Mathur, and G. Soucy. *3D Object Recognition Systems*, chapter Feature Extraction for 3-D Model Building and Object Recognition. Elsevier, Amsterdam, 1993.

[22] G. Godin and M. Levine. Structured edge map of curved objects in a range image. In *Proceedings IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition*, San Diego, California, June 4-8 1989.

[23] B. Hayashi and S. Negahdaripour. Direct motion stereo: Recovery of observer motion and scene structure. In *Proceedings, 3RD International Conference on Computer Vision*, pages 446–449, Osaka, Japan, Dec. 1990. Computer Society of the IEEE, IEEE Computer Society Press.

[24] D. Hilbert and S. Cohn-Vossen. *Geometry and the Imagination.* Chelsea, New York, 1952.

[25] C. Jerian and R. Jain. Determining motion parameters for schemes with translation and rotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(4):523–529, 1983.

[26] J. Lagarde. Constraints and their satisfaction in the recovery of local surface structure. Master's thesis, Dept. of E.E., McGill Univ., 1989.

[27] C. Lee. Finding point correspondences and determining motion of a rigid object from two weak perspective views. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 398–403, Ann Arbor, Michigan, June 1988. Computer Society of the IEEE, IEEE Computer Society Press.

[28] D. Marr. *Vision*. W.H. Freeman & Co., San Francisco, 1982.

[29] A. Mitchie, X. Zhuang, and R. Haralick. Interpretation of optical flow by rotation decoupling. In *Proceedings of the IEE Computer Society Workshop on Computer Vision*, pages 195–199, Miami Beach, Florida, Dec. 1987. Computer Society of the IEEE, IEEE Computer Society Press.

[30] N. Navab, R. Deriche, and O. Faugeras. Recovering 3d motion and structure from stereo and 2d token tracking cooperation. In *Proceedings, 3RD International Conference on Computer Vision*, pages 513–516, Osaka, Japan, Dec. 1990. Computer Society of the IEEE, IEEE Computer Society Press.

[31] S. Negahdaripour and C. Ho Yu. Robust recovery of motion: Effects of surface orientation and field of view. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 404–410, Ann Arbor, Michigan, June 1988. Computer Society of the IEEE, IEEE Computer Society Press.

[32] S. Negahdaripour and B. Horn. Using depth-ia-positive constraint to recover translational motion. In *Proceedings of the IEE Computer Society Workshop on Computer Vision*, pages 138–143, Miami Beach, Florida, Dec. 1987. Computer Society of the IEEE, IEEE Computer Society Press.

[33] R. Nelson and J. Aloimonos. Finding motion parameters from spherical flow fields. In *Proceedings of the IEE Computer Society Workshop on Computer Vision*, pages 145–149, Miami Beach, Florida, Dec. 1987. Computer Society of the IEEE, IEEE Computer Society Press.

[34] P. Parent and S. Zucker. Curvature consistency and curve detection. *J. Opt. Soc. Amer., Ser. A*, 2(13), 1985.

[35] K. Prazdny. Egomotion and relative depthmap from optical flow. *Biological Cybernetics*, 36:87–102, 1980.

[36] K. Prazdny. Determining the instantaneous direction of motion from optical flow generated by a curvilinearly moving observer. *Computer Graphics and Image Processing*, 17:238–248, 1981.

[37] M. Rioux. Laser range finder based on synchronized scanners. *Applied Optics*, 23(21):3837–3844, Nov. 1984.

[38] P. Sander and S. Zucker. Inferring differential structure from 3-d images: Smooth cross sections of fiber bundles. *IEEE Trans. PAMI*, 12(9):833–854, 1990.

[39] K. Shoemake. Animating rotations with quaternion curves. *ACM Computer Graphics*, 21(5):365–373, 1985.

[40] K. Spring. Euler parameters and the use of quaternion algebra in the manipulation of finite rotations: a review. *Mechanism and Machine Theory*, 21(5):365–373, 1986.

[41] M. Swain. Ijcv special issue on active vision. 1993.

[42] R. Szeliski. Estimating motion from sparse range data without correspondence. In *Proceedings, 2ND International Conference on Computer Vision*, pages 207–215, Tampa, Florida, Dec. 1988. Computer Society of the IEEE, IEEE Computer Society Press.

[43] R. Tsai and T. Huang. Estimating three-dimensional motion parameters of a rigid planar patch. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-30:525–534, Aug. 1982.

[44] T. Viéville and O. Faugeras. Feed-forward recovery of motion and structure from a sequence of 2-d line matches. In *Proceedings, 3RD International Conference on Computer Vision*, pages 517–521, Osaka, Japan, Dec. 1990. Computer Society of the IEEE, IEEE Computer Society Press.

[45] J. Weng, T. Huang, and N. Ahuja. Estimating motion and structure from line matches: performance obtained and beyond. In *Proceedings of the 10TH International Conference on Pattern Recognition*, pages 168–172, Atlantic City, New Jersey, June 1990. Computer Society of the IEEE, IEEE Computer Society Press.

[46] G. Young and R. Chellappa. 3-d motion estimation using a sequence of noisy stereo images. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 710–716, Ann Arbor, Michigan, June 1988. Computer Society of the IEEE, IEEE Computer Society Press.

[47] G. Young and R. Chellappa. Statistical analysis of inherent ambiguities in recovering 3-d motion from a noisy flow field. In *Proceedings of the 10TH International Conference on Pattern Recognition*, pages 371–377, Atlantic City, New Jersey, June 1990. Computer Society of the IEEE, IEEE Computer Society Press.

[48] Z. Zhang, O. Faugeras, and N. Ayache. Analysis of a sequence of stereo scenes containing multiple moving objects using rigidity constraints. In *Proceedings, 2ND International Conference on Computer Vision*, pages 177–186, Tampa, Florida, Dec. 1988. Computer Society of the IEEE, IEEE Computer Society Press.