Estimating Motion from Sparse Range Data without Correspondence[1]

Richard Szeliski
Schlumberger Palo Alto Research
3340 Hillview Ave.
Palo Alto, CA 94304

Abstract

Estimating observer motion from time-varying range data and fusing this data into a coherent map of the environment are two important problems in robot navigation. Current methods first determine a correspondence between range measurements acquired from different viewpoints, and then compute a motion estimate from this correspondence. In this paper, we present an alternative technique which does not assume that any such correspondence exists. Instead, a smooth surface assumption is used, i.e., the sensed points are assumed to lie on some piecewise smooth surface. A motion estimate is obtained by finding the geometric transformation which makes it most likely (in a Bayesian sense) that the points came from the same surface. We derive an energy equation which measures the distance between the new data points and the dense interpolated depth map which is being incrementally refined. The shape of the energy equation in the neighborhood of the optimal motion estimate is used to compute the uncertainty in the estimate. The resulting motion estimation algorithm can be used in conjunction with other motion estimation systems, and provides a flexible and robust method for computing motion from sparse range data.

1 Introduction

As computer vision moves from the analysis of static images towards full integration into real-time robotics systems, the issues of dynamic motion estimation and on-line sensory integration are becoming increasingly important. Traditionally, motion estimation has been performed by extracting image features, finding correspondences, and then jointly solving for the rigid motion and the three-dimensional position of the features [Ullman79, Tsai84]. More recently, the advent of direct range sensing has made it possible to determine motion directly from a set of three-dimensional points with known correspondence [Faugeras87]. While these motion estimation methods have been successful in highly structured man-made environments, they are less applicable for outdoor navigation. The unstructured nature of the outdoor environment makes it more difficult to reliably extract features for matching, and occlusions or limited areas of overlap cause many of the sensed range points to have no correspondence.

In this paper, we introduce a new method for determining observer motion from sparse range data which does not assume any correspondences between sensed points. In addition to determining the motion, the method also "registers" the two sets of points so that they may be integrated into an updated surface estimate. Our method is based on a *smooth surface assumption*, i.e., the points which are sensed with the range finder (from two or more viewpoints) are assumed to lie on a piecewise smooth surface. Our algorithm finds the motion which makes it most likely that these sets of points are from the same piecewise smooth scene. In practice, the logarithm of the likelihood measure turns out to be closely related to the weighted sum of squares distance between the new data points and the current surface estimate. This method thus fits well with incremental sensing strategies, where dense depth estimates are obtained by integrating measurements taken from a moving camera [Matthies88].

The method presented in this paper shows how to measure the the likelihood of a particular collection of transformed points being properly "registered", and how to find a locally optimal motion estimate using gradient descent. This paper does not, however, address the issue of how to search the large space of possible transformations for the "best" motion. Our method is thus meant to be used in conjunction with some other motion estimation system — such as an inertial navigation system — which is used to start the gradient descent algorithm in the vicinity of the solution. Our method will also work if the range of possible motions is small, which can be ensured by sampling the data sufficiently rapidly (as is the case in real-time robot control).

The motion estimation algorithm we develop can be applied to both mobile robot navigation and robot manipulation. As part of a mobile robot system, the algorithm is used to refine or improve motion estimates obtained from other sources such as inertial navigation, dead reckoning or landmark recognition. The algorithm also builds up and maintains a dense depth map of the environment which can be used for integration with other sensors. This map can either be a retinotopic (image-based) depth map or a terrain-based elevation map [Olin88]. Our algorithm is particularly well suited for terrain maps since it can handle data points that are irregularly spaced (from perspective de-projection), incorporate prior knowledge from cartographic data, and fuse data with only limited areas of overlap. In robot manipulation, our algorithm can be used to determine object or observer motion from sparse tactile data.

The general approach used in this paper is to incrementally build up a dense depth map by interpolating and integrating sparse range data, and to match new points to this surface to perform the motion estimation. We thus start by reviewing

[1] This research was performed while the author was a graduate student at Carnegie Mellon University.

CH2664-1/88/0000/0207$01.00 © 1988 IEEE

spline-based surface interpolation algorithms and show how these are implemented by minimizing a quadratic energy function defined over a set of nodal variables. Next, we introduce a Bayesian model of dense depth estimation, and show that the posterior estimate obtained from this model is equivalent to the interpolated surface obtained using splines. We then derive a simple matching criterion based on the distance between the new set of points and the old interpolated surface, and show why this criterion is sub-optimal. A statistically optimal criterion is then derived from the Bayesian model and is shown to be a sensible generalization of the simple matching criterion which we first proposed. Using this new criterion, we show how to calculate the uncertainty in the motion estimate and how to integrate this estimate with other data. We then demonstrate the performance of our algorithm on a number of synthetic two-dimensional range data examples. We close by comparing our method to a number of alternative approaches, and mention some of the possible extensions of our work.

2 Interpolating sparse data

To determine motion from a sparse set of depth values which do not have any correspondence, we must assume that the points have some spatial coherence or that they came from the same smooth surface. One way to do this is to match the points to a set of known models [Grimson84]. A more general approach is to use two-dimensional spline interpolation to derive a smooth surface from the sparse data. The estimate of this surface can be refined as more data is acquired, and the surface (or dense depth map) can be'used for integration with other sensors.

To formulate the spline approximation problem, we combine two weak constraints: a data compatibility constraint, and a smoothness constraint. The data compatibility constraint measures the distance between the collection of depth values $\{\mathbf{p}_i\} = \{(x_i, y_i, d_i)\}$ and the interpolated surface $u(x,y)$ using an energy measure

$$\mathcal{E}_d(u, \{\mathbf{p}_i\}) = \frac{1}{2}\sum_i w_i\,(u(x_i,y_i) - d_i)^2 \qquad (1)$$

where the weights w_i are inversely related to the variance of the measurements ($w_i = \sigma_i^{-2}$). The smoothness constraint, which is usually expressed as functional or norm of $u(x,y)$, encodes the *variation* in the surface. A simple example of such a constraint is the *thin plate* model, whose smoothness functional is

$$\mathcal{E}_p(u) = \frac{1}{2}\int\int \left(u_{xx}^2 + 2u_{xy}^2 + u_{yy}^2\right)\,dx\,dy. \qquad (2)$$

This model can be derived using either variational principles [Grimson83], physical models [Terzopoulos84] or regularization [Poggio85]. In this paper, we use a thin plate with controlled depth and orientation discontinuities, as discussed in [Terzopoulos86]. This interpolator was chosen because it produces piecewise smooth (C^2) surfaces. The possibility of using different interpolators is discussed in Section 8.

To find the approximating spline, the two constraints (1) and (2) are combined into a single energy functional

$$\mathcal{E}(u) = \mathcal{E}_d(u, \{\mathbf{p}_i\}) + \lambda\mathcal{E}_p(u) \qquad (3)$$

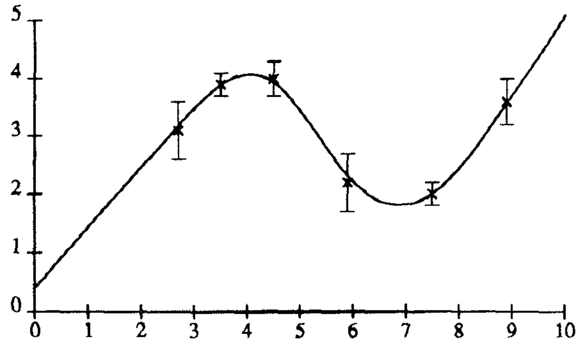

Figure 1: One-dimensional cubic spline interpolation
A cubic spline is fit to the data points shown with ×'s. The error bars indicate the standard deviation of each measurement.

where λ is called the regularization parameter. The minimum energy solution of (3) is called the approximating spline. Figure 1 shows a simple one-dimensional example of spline approximation. The curves drawn in this Figure are cubic splines (piecewise cubic polynomials), and are obtained using a one-dimensional version of the thin plate model.

To implement the minimization of (3) on a digital or analog computer, it is necessary to discretize the domain of the solution $u(x,y)$ using a finite number of *nodal variables*. The usual and most flexible approach is to use finite element analysis [Terzopoulos84]. In this paper, we use a rectangular domain on which a rectangular "fine grained" mesh has been applied. Since the topology of this mesh is fixed, it does not depend on the location of the data points, and can thus be used for integrating data from various sensors or from various viewpoints. The fine grained nature of the mesh also leads to a natural implementation on a massively parallel array of processors.

When finite element analysis is applied to the smoothness constraints defined by quadratic functionals such as (2), the resulting energy is a quadratic form

$$E_p(\mathbf{u}) = \frac{1}{2}\mathbf{u}^T\mathbf{A}_p\mathbf{u} \qquad (4)$$

where $\mathbf{u}$ is the vector of nodal variables, i.e., $\mathbf{u} = \{u(i,j)\}$. The *prior model* matrix $\mathbf{A}_p$ is extremely sparse, having only 13 entries per row for a thin plate. The quadratic structure of the energy function is true even for the controlled-continuity spline (see [Terzopoulos84] or [Szeliski88] for implementation details). The data constraint in (1) can also be re-written as a quadratic form in terms of $\mathbf{u}$ and the zero-padded vector of depth values $\tilde{\mathbf{d}}$,

$$E_d(\mathbf{u}, \mathbf{d}) = \frac{1}{2}(\mathbf{u} - \tilde{\mathbf{d}})^T\mathbf{A}_d(\mathbf{u} - \tilde{\mathbf{d}}). \qquad (5)$$

The diagonal matrix $\mathbf{A}_d$ consists of the data weights w_i where data points coincide with nodal variables and 0s elsewhere.

Using (4) and (5), we can write the combined energy as a single quadratic form

$$E(\mathbf{u}) = \frac{1}{2}\mathbf{u}^T\mathbf{A}\mathbf{u} - \mathbf{u}^T\mathbf{b} + c \qquad (6)$$

where

$$\mathbf{A} = \lambda\mathbf{A}_p + \mathbf{A}_d \quad\text{and}\quad \mathbf{b} = \mathbf{A}_d\tilde{\mathbf{d}}.$$

$$\mathbf{u}^{*} = \mathbf{A}^{-1}\mathbf{b}. \tag{7}$$

The solution of the linear system of equations (7) can be found using either direct or iterative methods (see [Terzopoulos84] for a survey of the latter). While the choice of solution method is not very important in the context of this paper, we note that iterative methods have the possibility of implementation on fine-grained parallel machines, which may be desirable for real-time applications. A second advantage of iterative schemes is that they can be used in conjunction with gradient-descent non-linear optimization techniques (Section 5).

3 Bayesian estimation

Spline-based interpolation provides one possible mechanism for estimating a smooth surface from a sparse set of depth values. An alternative approach is to use Bayesian modeling. In this framework, we assume a prior distribution $p(\mathbf{u})$ for the surface that we are estimating, and model the measurement process by a conditional probability distribution $p(\mathbf{d}|\mathbf{u})$. The posterior distribution $p(\mathbf{u}|\mathbf{d})$ is derived from these two models using Bayes' Rule

$$p(\mathbf{u}|\mathbf{d}) = \frac{p(\mathbf{d}|\mathbf{u})p(\mathbf{u})}{p(\mathbf{d})}. \tag{8}$$

In its usual application [Geman84], Bayesian modeling is used to find the *Maximum A Posteriori* (MAP) estimate, i.e., the value of $\mathbf{u}^*$ which maximizes the conditional probability $p(\mathbf{u}|\mathbf{d})$.

For estimating a dense depth map from sparse depth measurements, we will assume that the surface can be modeled as correlated Gaussian noise (the justification for this assumption is given later)

$$u(x,y) = h(x,y) * n(x,y)$$

where $h(x,y)$ is a filter impulse response, $n(x,y)$ is a white Gaussian noise process and $*$ is the two-dimensional convolution operator. A discrete version of $u(x,y)$ can be obtained by sampling the signal on a rectangular grid. This discrete signal, $\mathbf{u} = \{u(i,j)\}$, is a correlated multivariate Gaussian. In this paper, we will use the notation $\mathbf{x} \sim N(\mathbf{m},\mathbf{P})$ to denote that $\mathbf{x}$ is a multivariate normal variable with mean $\mathbf{m}$ and covariance $\mathbf{P}$ [Gelb74]. The probability density function for $\mathbf{x}$ can be written as

$$p(\mathbf{x}) = |2\pi\mathbf{P}|^{-1/2}\exp\left[-\frac{1}{2}(\mathbf{x}-\mathbf{m})^T\mathbf{P}^{-1}(\mathbf{x}-\mathbf{m})\right] \tag{9}$$

where $|\mathbf{P}|$ is the determinant of the matrix $\mathbf{P}$ (we use $|2\pi\mathbf{P}|^{-1/2}$ instead of $(2\pi)^{-n/2}|\mathbf{P}|^{-1/2}$ for notational succinctness). We can thus write the distribution of our discrete prior model as

$$\mathbf{u} \sim N(0,\mathbf{P}_0). \tag{10}$$

The covariance matrix $\mathbf{P}_0$ can be derived from the correlation function if it is known, or it can be chosen heuristically. In this paper, we follow the latter route. We chose the *inverse* covariance matrix $\mathbf{P}_0^{-1}$ to have the same sparse structure as the matrix $\mathbf{A}_p$ used in (4). This choice ensures that the mean estimate produced by the Bayesian approach is the same as will be important later in deriving a simple likelihood measure on the match. Our choice of $\mathbf{P}_0^{-1}$ also means that the prior model is a Markov Random Field [Geman84] and that the power spectrum of the prior is fractal [Szeliski87].

For our sensor model, we use a linear system with additive Gaussian noise

$$\mathbf{d} = \mathbf{H}\mathbf{u} + \mathbf{r}, \quad \text{with} \quad \mathbf{r} \sim \mathrm{N}(0,\mathbf{R}). \tag{11}$$

This form of sensor modeling is common in the Kalman filtering literature [Gelb74]. The measurement matrix $\mathbf{H}$ encodes the sparse sampling which converts from the dense depth map $\mathbf{u}$ to the sparse set of depth values $\mathbf{d}$. This rectangular matrix usually contains 1's where the nodal variables coincide with the data points, and 0's elsewhere. If the data points do not lie on the grid (e.g., if we have "sub-pixel" positioning accuracy), the $\mathbf{H}$ matrix can be derived from the local interpolation function and will contain non-integer values. The $\mathbf{R}$ matrix encodes the covariance of the measurement noise process, and is usually diagonal with $r_{ii} = \sigma_i^2$ (uncorrelated sensor noise).

From the sensor model (11) we can derive the distribution of the data $\mathbf{d}$ conditioned on the initial state $\mathbf{u}$ as

$$\mathbf{d}|\mathbf{u} \sim N(\mathbf{H}\mathbf{u},\mathbf{R}). \tag{12}$$

Similarly, we can derive the marginal distribution of the data by integrating over all possible initial states to obtain

$$\mathbf{d} \sim N(0,\mathbf{H}\mathbf{P}_0\mathbf{H}^T + \mathbf{R}) \tag{13}$$

(this result can also be derived using the formula for sums of Gaussian random variables).

The posterior estimate $\mathbf{u}_1$ after the first set of measurements can be derived from (10) and (12) using Bayes' Rule. This estimate is a multivariate Gaussian

$$\mathbf{u}_1 \sim N(\hat{\mathbf{u}}_1,\mathbf{P}_1) \tag{14}$$

with a mean

$$\hat{\mathbf{u}}_1 = (\mathbf{P}_0^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{d}$$

and a covariance

$$\mathbf{P}_1 = (\mathbf{P}_0^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}$$

(see [Gelb74] or [Maybeck79] for derivations).

The mean estimate $\hat{\mathbf{u}}_1$, which is also the MAP estimate, corresponds to the minimum energy solution of the spline approximation (7) if we make the following correspondences:

$$\mathbf{P}_0^{-1} = \lambda\mathbf{A}_p, \quad \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} = \mathbf{A}_d, \quad \text{and} \quad \mathbf{H}\mathbf{d} = \tilde{\mathbf{d}}.$$

The estimate obtained with Bayesian modeling is thus the same as that obtained with spline approximation. The Bayesian model also provides a description of the *uncertainty* or variance in this estimate [Szeliski87]. The Bayesian formulation of the estimation problem forms the basis of the optimal motion detection algorithm described in Section 5.

The Bayesian model can be extended to include multiple measurements of the same surface using

$$\mathbf{d}_k = \mathbf{H}_k\mathbf{u} + \mathbf{r}_k, \quad \text{with} \quad \mathbf{r}_k \sim \mathrm{N}(0,\mathbf{R}_k). \tag{15}$$

The estimation equations can be written in a recursive form using

$$\mathbf{A}_k = \mathbf{P}_0^{-1} + \sum_{j=1}^{k} \mathbf{H}_j^T \mathbf{R}_j^{-1} \mathbf{H}_j = \mathbf{A}_{k-1} + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k \qquad (16)$$

and

$$\mathbf{b}_k = \sum_{j=1}^{k} \mathbf{H}_j^T \mathbf{R}_j^{-1} \mathbf{d}_j = \mathbf{b}_{k-1} + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{d}_k \qquad (17)$$

to accumulate the inverse covariance and cumulative weighted data vector. The $\mathbf{A}_k$'s and $\mathbf{b}_k$'s are the same as are used in spline interpolation of multiple data sets. The optimal estimate at time k can be obtained by solving

$$\hat{\mathbf{u}}_k = \mathbf{A}_k^{-1} \mathbf{b}_k. \qquad (18)$$

4 Matching points to a surface

Once we have obtained a dense depth map from the first set of sparse depth measurements, we can match the second set of points to this surface to compute the likelihood of a particular motion. We start by assuming that the first set of data points $\mathbf{p}_1$ is registered with respect to the world coordinate frame. We then compute an interpolated surface $u(x, y)$ by splitting each point into its location (x_i, y_i), which is incorporated into the $\mathbf{H}_1$ matrix, and its depth value d_i, which becomes part of $\mathbf{d}_1$. The variance assigned to each point is obtained by projecting the three-dimensional covariance matrix Σ_i onto the z-axis.

The second set of points $\mathbf{p}_2$ is obtained by taking the sensor-based set of depth measurements $\mathbf{p}_2'$ and transforming them through a geometric transformation T which is parameterized by Θ

$$\mathbf{p}_2 = T(\mathbf{p}_2', \Theta).$$

These transformed points are then used to derive $\mathbf{H}_2$, $\mathbf{R}_2$ and $\mathbf{d}_2$, which are now all functions of Θ. The exact form of Θ is not important here since we are not trying to obtain a closed-form solution. In this paper, we will use the translation vector (t_x, t_y, t_z) and three rotation angles $(\theta_x, \theta_y, \theta_z)$ as our parameters (see [Tsai84] and [Faugeras87] for alternate formulations).

We can measure the likelihood of the new points coming from the interpolated surface $\hat{\mathbf{u}}_1$ using (12). Substituting into (9), we obtain

$$p(\mathbf{d}_2|\Theta) = |2\pi\mathbf{R}_2|^{-1/2} \exp\left[-\frac{1}{2}(\mathbf{d}_2 - \mathbf{H}_2\hat{\mathbf{u}}_1)^T \mathbf{R}_2^{-1} (\mathbf{d}_2 - \mathbf{H}_2\hat{\mathbf{u}}_1)\right]. \qquad (19)$$

We can find the maximum likelihood estimate of Θ by maximizing the above equation, or equivalently, minimizing the negative log likelihood

$$\begin{aligned} E_s(\mathbf{d}_2) &= -\log p(\mathbf{d}_2|\Theta) & (20) \\ &= \frac{1}{2}\log|2\pi\mathbf{R}_2| + \frac{1}{2}(\mathbf{d}_2 - \mathbf{H}_2\hat{\mathbf{u}}_1)^T \mathbf{R}_2^{-1} (\mathbf{d}_2 - \mathbf{H}_2\hat{\mathbf{u}}_1). \end{aligned}$$

In general, (20) does not have a closed-form solution for the minimum energy transformation (the vertical translation t_z is the exception). This energy equation (the subscript 's' stands for "simple") must thus be minimized using non-linear optimization techniques such as gradient descent [Press86].

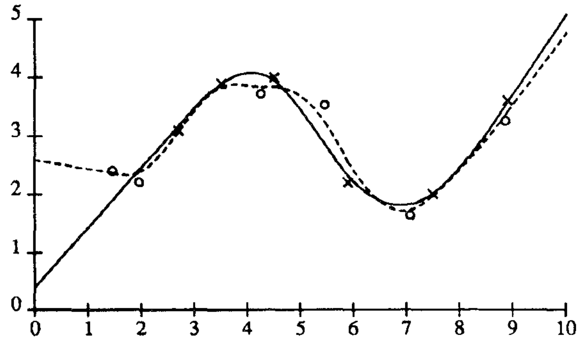

Figure 2: Result of simple motion estimation
The set of data points indicated by o's is matched to the solid curve interpolated through the ×'s. The misalignment is due to matching the extrapolated curve on the left. The dashed curve indicates the "updated" fit to both data sets.

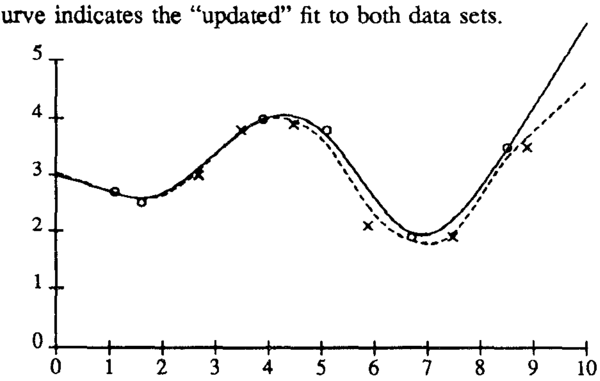

Figure 3: Simple motion estimation with points reversed
The set of data points indicated by ×'s is matched to the solid curve interpolated through the o's. The alignment differs from that of Figure 2.

Figure 2 shows a one-dimensional example of this matching technique. The smooth spline interpolated from the first set of data points (×'s) is shown as a solid curve. The second set of data points (o's) is shown in its "most likely" position and orientation. By examining this example, we can observe several problems with our simple method. The matching of new data points to the "extrapolated" parts of the curve is inaccurate, since little is known about the curve in these areas. This is symptomatic of the more general problem with this technique, which is that it does not incorporate any knowledge about the uncertainty in the original spline approximation. For example, range data will often have "shadowed" areas where the extrapolated data can be extremely uncertain. This method is also not symmetrical with respect to the data acquisition, since changing the role of the ×'s and o's results in a different registration (Figure 3). To overcome these problems, we have to go back to the original Bayesian formulation, and derive an optimal motion estimator.

5 Optimal motion estimation

To derive an optimal estimate of the motion, we must find the value of Θ which makes it most likely that the two sets of data points $\mathbf{p}_1$ and $\mathbf{p}_2$ came from the same smooth surface.

To compute the likelihood of observing the depth values $\mathbf{d}_2$ (through the measurement matrix $\mathbf{H}_2$), we note that the distribution $\mathbf{u}_1 \sim N(\hat{\mathbf{u}}_1, \mathbf{P}_1)$ describes all that is known about the smooth surface after the first set of data points has been incorporated. The second set of data points $\mathbf{d}_2$ must thus be drawn from the distribution

$$\mathbf{d}_2 \sim N(\mathbf{H}_2\hat{\mathbf{u}}_1, \mathbf{H}_2\mathbf{P}_1\mathbf{H}_2^T + \mathbf{R}_2)$$

(this same result can be obtained by writing down the joint probability function $p(\mathbf{u}, \mathbf{d}_1, \mathbf{d}_2)$ and calculating the conditional probability $p(\mathbf{d}_2|\mathbf{d}_1)$). The negative log likelihood function in this case is

$$E_o(\mathbf{d}_2) = \frac{1}{2}\log|2\pi(\mathbf{H}_2\mathbf{P}_1\mathbf{H}_2^T + \mathbf{R}_2)| + \frac{1}{2}(\mathbf{d}_2 - \mathbf{H}_2\hat{\mathbf{u}}_1)^T(\mathbf{H}_2\mathbf{P}_1\mathbf{H}_2^T + \mathbf{R}_2)^{-1}(\mathbf{d}_2 - \mathbf{H}_2\hat{\mathbf{u}}_1) \qquad (21)$$

(the subscript 'o' stands for "optimal"). Unfortunately, (21) is difficult to evaluate, since it involves the covariance matrix $\mathbf{P}_1$, which is not sparse (for a 512 × 512 image, this matrix has more than 10^{10} entries!) However, using some algebraic manipulation (Appendix A), we can obtain a more computationally tractable formula for the log likelihood

$$E_o(\mathbf{d}_2) = E_1(\mathbf{d}_2) + E_2(\mathbf{d}_2) \qquad (22)$$

where

$$E_1(\mathbf{d}_2) = \frac{1}{2}\log|2\pi\mathbf{R}_2^{-1}| + \frac{1}{2}\log|\mathbf{P}_1^{-1}| - \frac{1}{2}\log|\mathbf{P}_2^{-1}| \qquad (23)$$

and

$$E_2(\mathbf{d}_2) = \frac{1}{2}(\mathbf{d}_2 - \mathbf{H}_2\hat{\mathbf{u}}_1)^T\mathbf{R}_2^{-1}(\mathbf{d}_2 - \mathbf{H}_2\hat{\mathbf{u}}_2). \qquad (24)$$

The first component of the energy, E_1, measures the reduction in likelihood due to the sensor noise as traded off against the increase in posterior information. In practice, this component of the energy varies fairly slowly with the transformation parameter Θ and can usually be ignored. The second part of the energy, E_2, measures the distance between the new data points $\mathbf{d}_2$ and the surfaces $\hat{\mathbf{u}}_1$ and $\hat{\mathbf{u}}_2$. Note how (24) is similar to (20) except that one side of the quadratic form now involves the *new* surface estimate. Points which lie closer to the new surface estimate $\hat{\mathbf{u}}_2$ than to the old estimate $\hat{\mathbf{u}}_1$ are thus penalized less by the optimal energy measure. In this way, areas where the surface values are originally uncertain (because there is little data, the area is "shadowed", or the surface is being extrapolated) contribute less to the matching criterion.

A related energy measure which could be used instead of E_2 as a matching criterion is

$$E_3(\mathbf{d}_2) = \frac{1}{2}\hat{\mathbf{u}}_2^T\mathbf{P}_0^{-1}\hat{\mathbf{u}}_2 + \frac{1}{2}\sum_{i=1}^{2}(\mathbf{d}_i - \mathbf{H}_i\hat{\mathbf{u}}_2)^T\mathbf{R}_i^{-1}(\mathbf{d}_i - \mathbf{H}_i\hat{\mathbf{u}}_2) \qquad (25)$$

which is derived from the joint probability density function of the data and the surface (Appendix A). This new energy clearly shows the symmetry of the formula with respect to data set ordering. This alternate form also has a simple interpretation as the weighted distance between the data points and the optimal surface estimate ("spring energy") plus the smoothness value of the surface ("strain energy"). The determination of an optimal motion estimate is thus equivalent to minimizing the energy of the composite surface and spring system.

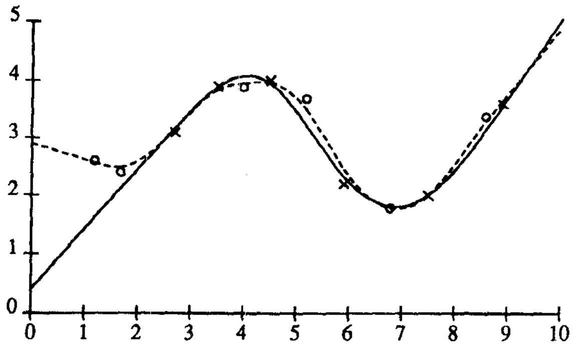

Figure 4: Optimal motion estimation
The solid curve shows the spline fit to the first set of points (×'s), and the dashed curve shows the spline fitted to all of the points (×'s and o's). Note how the updated estimate (dashed curve) more closely matches the new data points than the previous estimate (solid curve).

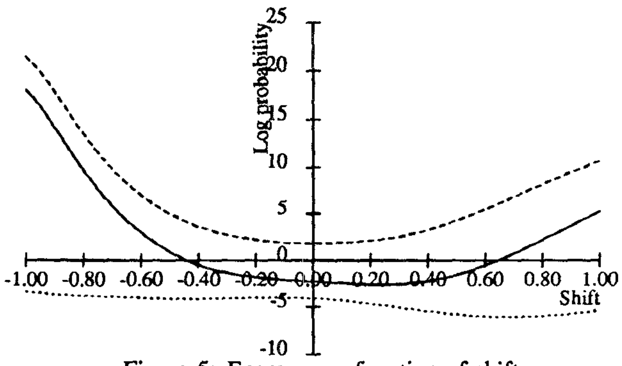

Figure 5: Energy as a function of shift
The dotted curve shows E_1 (log determinants), the dashed curve shows E_2 (spline and spring energy), and the solid curve shows E_o which is the negative log likelihood.

The implementation of optimal motion estimation is somewhat more complicated than the simple version that we presented in the previous section. This is because the updated surface estimate $\hat{\mathbf{u}}_2$ must be re-computed each time a new transformation Θ is generated. Fortunately, if successive transformations are close, the new surface estimate can be obtained from the previous estimate using only a few relaxation iterations. More importantly, since $\hat{\mathbf{u}}_2$ actually corresponds to the minimum energy solution of (25), we can jointly optimize $\hat{\mathbf{u}}_2$ and Θ using a continuation method similar to [Witkin86].

The optimal motion estimate for the one-dimensional example which we introduced previously is shown in Figure 4. The new registration is "better" than the ones previously obtained since the outlying data points do not "pull" the solution towards the previous estimate as much. The energy profile for one of the parameters (the shift t_x) is shown in Figure 5. As we mentioned before, the variation of E_1 with Θ (dotted curve) is small with respect to that of E_2 (dashed curve), and can thus be neglected.

6 Modeling motion uncertainty and optimal integration

The method which we have just described computes an optimal motion estimate by finding the transformation which minimizes an energy (negative log likelihood) function. This motion estimate is itself uncertain, i.e., it has a variance that can be determined from the shape of the "error surface". We can justify estimating the distribution of the transformation Θ from the likelihood of $\mathbf{d}_2$ using Bayes' Rule

$$p(\Theta|\mathbf{d}_2) = \frac{p(\mathbf{d}_2|\Theta)p(\Theta)}{p(\mathbf{d}_2)} \propto p(\mathbf{d}_2|\Theta)$$

if the prior distribution of the transformation $p(\Theta)$ is uniform. We can thus estimate a complete distribution for $p(\Theta|\mathbf{d}_2)$ by simply normalizing $p(\mathbf{d}_2|\Theta)$, which is the negative exponential of our energy function $E_o(\mathbf{d}_2)$. This distribution can encode multiple hypotheses about Θ by having multiple "humps".

In practice, estimating a complete distribution for Θ may be impractical. Instead, we can take the optimal motion estimate $\hat{\Theta}$ and augment it with a covariance matrix Σ_Θ. One way of calculating Σ_Θ is to first sample the complete distribution $p(\Theta|\mathbf{d}_2)$ and to then compute its first and second moments. A simpler, though potentially less accurate method is to simply fit a "parabola" (second order polynomial) to the energy surface. In this paper, we adopt this latter method, i.e., we fit the surface

$$E'(\Theta) = \frac{1}{2}(\Theta - \hat{\Theta})^T\mathbf{M}(\Theta - \hat{\Theta}) + c'$$

to the error surface $E_o(\mathbf{d}_2|\Theta)$ in the vicinity of the minimum energy. We use least squares fitting to perform this fit since it has a simple closed-form solution. The optimal motion estimate is then equated to $\hat{\Theta}$, and the inverse covariance Σ_Θ^{-1} is set to $\mathbf{M}$.

Using the parabolic fit has several advantages over the moment-based approach. It allows us to estimate the covariance of the motion estimate using only a few sample Θ values, and also provides a $\hat{\Theta}$ estimate that has better precision than the original sampling of Θ values. Estimating the inverse covariance can also be more stable since this matrix is often near-singular (e.g., when certain types of motion are being confounded or when there is insufficient data for a good estimate). The values for $\hat{\Theta}$ and Σ_Θ that were computed for our one-dimensional example are

$$\hat{\Theta} = \begin{bmatrix} 0.0782 \\ -0.1119 \\ -0.0031 \end{bmatrix} \qquad \Sigma_\Theta = \begin{bmatrix} 0.0906 & 0.0183 & 0.0013 \\ 0.0183 & 0.0280 & 0.0000 \\ 0.0013 & 0.0000 & 0.0002 \end{bmatrix}.$$

The three parameters are shift (t_x), offset (t_y) and rotation (θ_z, in radians).

Having a statistical description of the motion estimate allows us to integrate this information with other estimates, such as those provided by dead reckoning or inertial navigation. If each measurement has a mean $\hat{\Theta}_i$ and an associated inverse covariance matrix $\Sigma_{\Theta_i}^{-1}$, we can obtain an updated estimate from

$$\hat{\Theta} = (\Sigma_\Theta^{-1})^{-1} \sum_i \Sigma_{\Theta_i}^{-1}\hat{\Theta}_i \quad \text{and} \quad \Sigma_\Theta^{-1} = \sum_i \Sigma_{\Theta_i}^{-1}.$$

The overall uncertainty in the motion estimate can thus be reduced significantly, especially if the directions of maximum uncertainty are orthogonal for different sensing modalities.

The scheme we have just developed for estimating the motion between two frames can easily be extended to estimate a whole sequence of motions. The formulas given in (23) and (24) can be converted into a recurve form by simply substituting k for the subscript 2 and $k-1$ for the subscript 1. After each motion $\hat{\Theta}_k$ is computed, the new surface estimate $\hat{\mathbf{u}}_k$ and its inverse covariance $\mathbf{P}_k^{-1}$ are updated (we could also use the recursive formulas given in (16) and (17)). The Bayesian scheme which we have developed can also be used to incorporate domain-specific prior knowledge, such as cartographic information. The easiest way to do this is to introduce a "phantom" range observation before the first real set of data.

7 A synthetic range data example

To investigate the performance of our algorithm on sparse noisy range data, we have tested it on a number of synthetic range images. For each synthetic scene, a number of nearby viewpoints are selected, and for each viewpoint, a small rectangular array of depth values is generated using perspective projection. These depth values are then corrupted by adding white Gaussian noise with a standard deviation proportional to the range value (this more closely models laser range finder data than uniform noise). The noisy depth values are then input to the motion estimation algorithm.

Two of the synthetic scenes that we use are shown in Figure 6 ("digital terrain") and Figure 10 ("blocks world"). We can generate a sparse (9×7) depth map from the digital terrain by modeling a range finder with a pan of ±40° and a tilt of 15° to 45° (Figure 7). Applying the spline approximation algorithm to this sparse data, we obtain the surface shown in Figure 8. Note how this surface is quite different from the original terrain in those areas where there are no sample points. Fortunately, the lack of precision in these "hallucinated" (shadowed or invisible) areas does not affect the matching very much (for reasons discussed in Section 5). A second set of sparse depth points (Figure 9) is then generated by moving the range finder "forward" by 0.05 units (our digital terrain is 1.0×1.0 units large). Note that this set of points does not closely coincide with the first set, and that there is incomplete overlap between the two sets.

The result of our motion estimation algorithm on the sampled terrain range data is

$$\Theta^* = \begin{bmatrix} 0.0000 \\ 0.0500 \\ 0.0000 \\ 0.0000 \\ 0.0000 \\ 0.0000 \end{bmatrix} \qquad \hat{\Theta} = \begin{bmatrix} 0.0042 \\ 0.0501 \\ 0.0000 \\ 0.0002 \\ 0.0050 \\ 0.0053 \end{bmatrix}$$

where the true motion is Θ^*, and the optimal estimate is $\hat{\Theta}$. The six parameters are t_x, t_y, t_z (units normalized to model size) and θ_x, θ_y and θ_z (radians). As we can see the method provides a good estimate of the camera motion, even when little information is given. Unfortunately, the computation of the

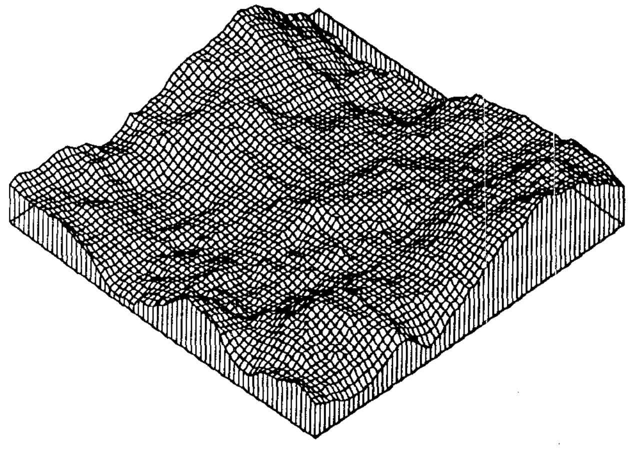

Figure 6: "Digital terrain" synthetic scene

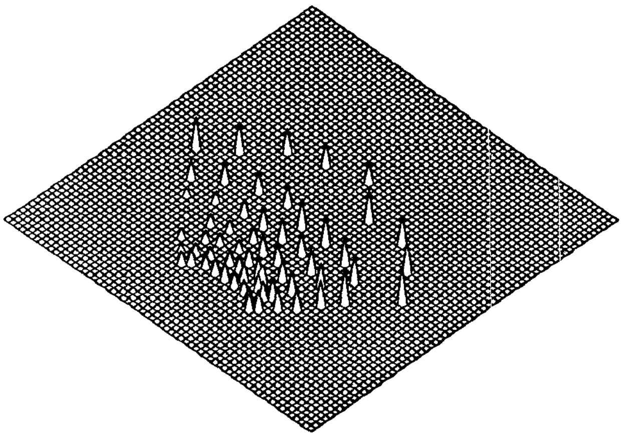

Figure 7: Sparse terrain depth data from viewing position 1

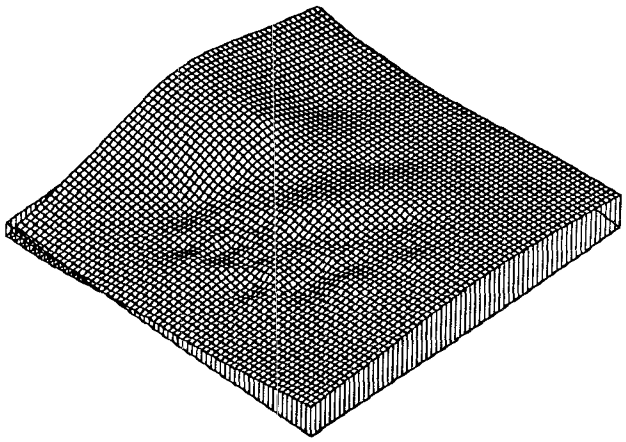

Figure 8: Interpolated surface from first set of data

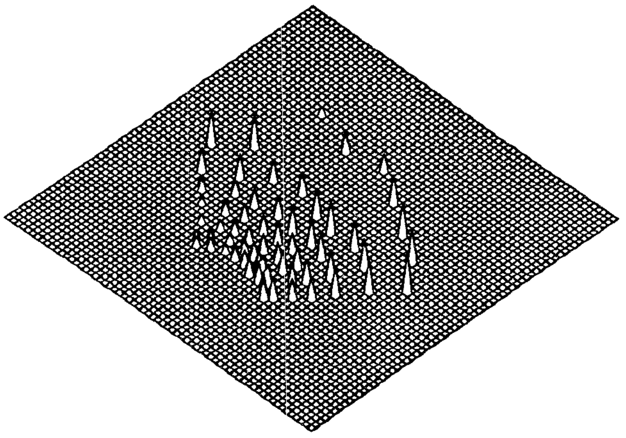

Figure 9: Sparse terrain depth data from viewing position 2

information matrix **M** from the energy values in the vicinity of the optimal estimate leads to a matrix that is not positive definite (and hence non-invertible). To overcome this, a quadratic programming technique could be used, but this has not yet been implemented.

To demonstrate that our motion estimation technique can work over a wider range of viewpoints, we can generate the two point sets shown in Figures 11 and 13. Here, the blocks world scene is being viewed from two different quadrants. The interpolated solution from the first set (Figure 12) is quite inaccurate in the "shadowed" area behind the block, and yet the motion estimate obtained from this data is still very good

$$\Theta^* = \begin{bmatrix} 0.8000 \\ -0.1000 \\ -0.1000 \\ -0.0952 \\ -0.1000 \\ -1.6345 \end{bmatrix} \quad \hat{\Theta}_o = \begin{bmatrix} 0.8005 \\ -0.1000 \\ -0.1000 \\ -0.0864 \\ -0.0975 \\ -1.6320 \end{bmatrix} \quad \hat{\Theta}_s = \begin{bmatrix} 0.9100 \\ -0.1100 \\ -0.9980 \\ -0.1002 \\ -0.0950 \\ -1.6345 \end{bmatrix}$$

where the true motion is shown on the left, and the optimal estimate in the middle, and the "simple" method estimate on the right. The final surface reconstructed from the combined data is quite reasonable (Figure 14).

We can compare this motion estimate to that obtained using the "simple" matching criterion of Section 4. As shown in above, this estimate is quite far from the true motion. This is mostly due to the mismatch between the ground plane data seen from the second viewpoint (Figure 13) and the solution interpolated from the first viewpoint in the shadowed area (Figure 12).

While these preliminary results are encouraging, more experimentation is required to fully characterize the performance of our algorithm. The region of convergence [Lucas84] of the gradient descent algorithm should be studied. The effects of varying surface shape, sampling density and amount of noise should also be examined. The method should be tested on an on-line estimation example (e.g., traversing some terrain) to see if the depth map improves as more samples are added, and whether the accuracy of the motion estimate increases as a result.

8 Discussion

The motion estimation method which we have developed in this paper can be compared to a number of alternative range-based motion estimation techniques. One such alternative is di-

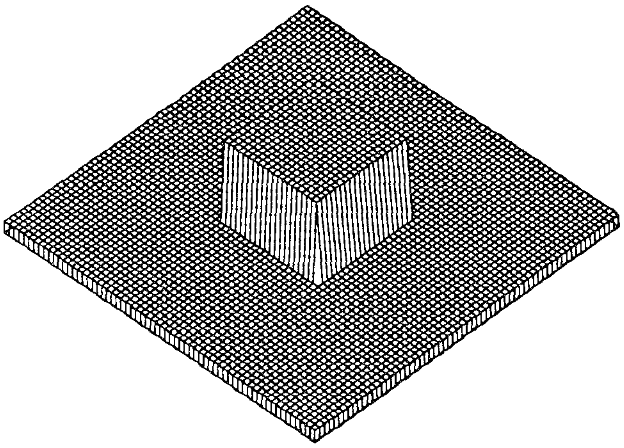

Figure 10: "Blocks world" synthetic scene

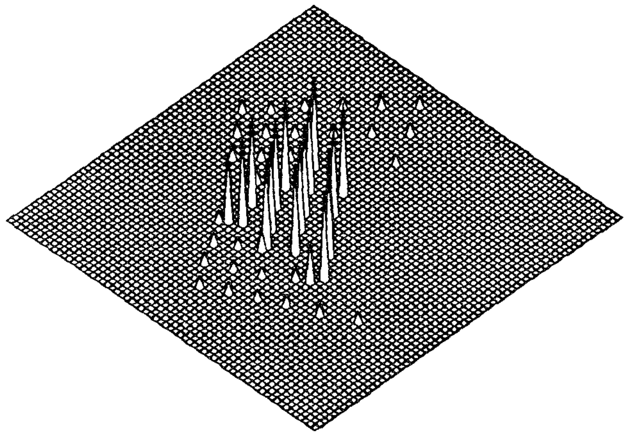

Figure 11: Sparse block depth data from viewing position 1

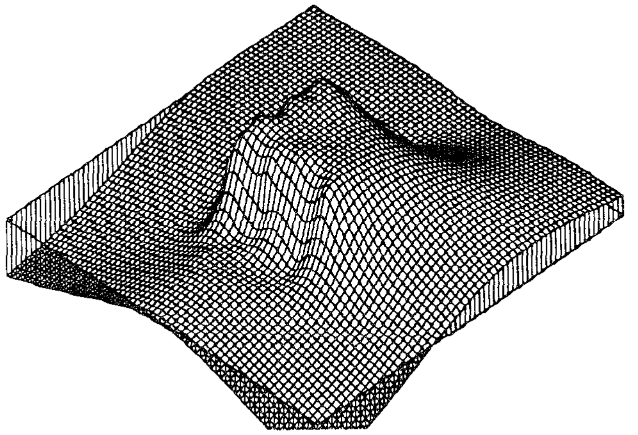

Figure 12: Interpolated surface from first sparse block data set

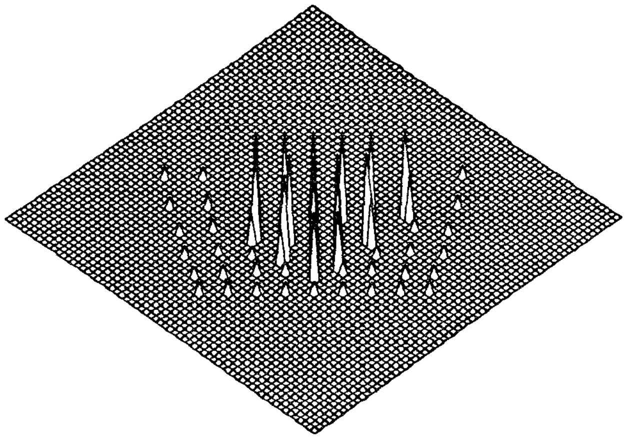

Figure 13: Sparse block depth data from viewing position 2

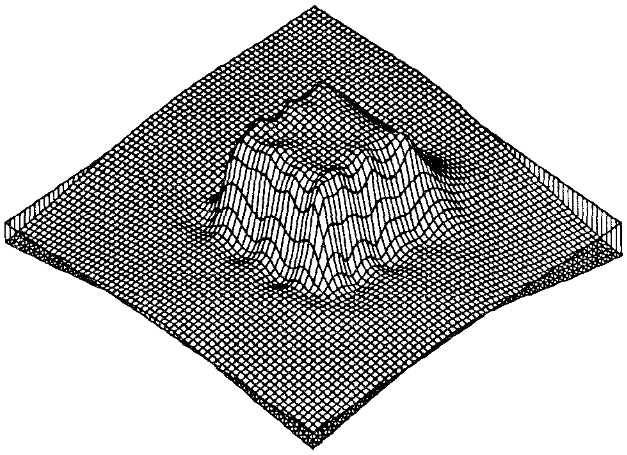

Figure 14: Interpolated surface from both block data sets

rect point or plane matching [Faugeras87]. This method works well if the sampling density is high enough so that the matched points are nearby, or if the object consists of planar facets. In the latter case, the method described in [Faugeras87] may actually give better results than our method, since a plane fit gives a more accurate position estimate than a spline fit. Spatial likelihood maps [Christ87] could also be used, but these require the use of spherical coordinates and do not have an explicit smoothness constraint. Another alternative is occupancy maps [Elfes87, Stewart87]. This method has the advantage that it can represent any three-dimensional scene without the need for a reference plane. However, the accuracy of these representations is limited to the grid or cell size, and combining new measurements with the existing estimate is often difficult. A similar representation is the elevation map [Hebert88], which is closely related to the spline representation used in this paper. Recent work [Kweon88] examines how to convert between camera-based range images and elevation maps taking into account the shape of the sensor noise.

Comparing our new method to these previous approaches, we see that it has several advantages. Our approach can handle sparse and irregularly spaced data. It does not require any correspondence between sensed data points, and can handle data with only limited overlap. The dense depth map that is incrementally built up becomes increasingly more accurate as more data is acquired, since the height is represented as a real value. The depth map can also be "extrapolated" to shadowed or invisible areas without affecting the performance of the motion estimation algorithm. This is because our framework implicitly models the uncertainty of the interpolated surface. Finally, the uncertainty in the motion estimate can be calculated from the shape of the energy surface in the vicinity of the optimal estimate.

The work which we have described could be extended in several directions. A discontinuity detection process (perhaps based on [Blake87]) should be incorporated into our algorithm. The regularization parameter λ which controls the amount of smoothing could be derived from the data by maximizing the likelihood of the observations, i.e., finding the λ value which minimizes (22) [Szeliski88]. A related possibility is to vary the amount of smoothing being used while searching for the optimal motion parameters. This coarse-to-fine strategy, which has been used successfully in stereo matching [Lucas84, Witkin86], can be used to broaden the "region of convergence" of our gradient-descent optimization technique.

The surface and sensor models used in our algorithms could also be extended. For many applications, viewpoint invariant interpolators [Blake87] might be a better choice than our viewpoint dependent splines. Another possibility would be to use full three-dimensional energy-based surface models [Terzopoulos87]. Both of these models no longer have quadratic energy functions, so the resulting probability distributions are no longer Gaussian. The sensor models can be extended using a full three-dimensional noise covariance matrix, in which case the data (compatibility) constraint becomes non-linear [Szeliski88].

Finally, our algorithms should be tested on real range images, such as the ones obtained from the ERIM laser range finder used on the NavLab at Carnegie Mellon. This would conclusively determine the viability of our approach, and could also bring to light unforeseen problems with the method.

9 Conclusions

In this paper, we have presented a new method for estimating observer motion from sparse range data. This method is based on a smooth surface assumption used in conjunction with Bayesian analysis. A dense depth estimate is obtained from the sparse range data using two-dimensional spline fitting. This dense estimate is then used for matching the new set of data points. To implement the correct Bayesian likelihood measure, the *new* updated surface estimate must be used as part of the match criterion. The optimal motion estimate is obtained using gradient descent on the match energy as a function of the motion parameters. The shape of the energy function in the vicinity of this optimal motion is used to estimate the covariance matrix of the motion estimate. This statistical description of our motion estimate can then be used to integrate this information with other position sensors. Our results demonstrate that the resulting algorithm is useful for real-time robot navigation. They also suggest that Bayesian analysis can in general improve the performance of vision algorithms and extend their domain of applicability.

References

[Blake87] A. Blake and A. Zisserman. *Visual Reconstruction.* MIT Press, Cambridge, Massachusetts, 1987.

[Christ87] J. P. Christ. *Shape Estimation and Object Recognition Using Spatial Probability Distributions.* PhD thesis, Carnegie Mellon University, April 1987.

[Elfes87] A. Elfes and L. Matthies. Sensor integration for robot navigation: combining sonar and stereo range data in a grid-based representation. In *IEEE Conference on Decision and Control*, IEEE Computer Society Press, 1987.

[Faugeras87] O. D. Faugeras and M. Hebert. The representation, recognition and positioning of 3-D shapes from range data. In Takeo Kanade, editor, *Three-Dimensional Machine Vision*, pages 301–353, Kluwer Academic Publishers, Boston, MA, 1987.

[Gelb74] Arthur Gelb, editor. *Applied Optimal Estimation.* MIT Press, Cambridge, Massachusetts, 1974.

[Geman84] S. Geman and D. Geman. Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741, November 1984.

[Grimson83] W. E. L. Grimson. An implementation of a computational theory of visual surface interpolation. *Computer Vision, Graphics, and Image Processing*, 22:39–69, 1983.

[Grimson84] W. E. L. Grimson and T. Lozano-Pérez. Model-based recognition and localization from sparse range data. *International Journal of Robotics Research*, 3(3):3–35, Fall 1984.

[Hebert88] M. Hebert and T. Kanade. 3-D vision for outdoor navigation by an autonomous vehicle. In *Image Understanding Workshop*, pages 365–382, Morgan Kaufmann Publishers, Cambridge, MA, April 1988.

[Kweon88] I. S. Kweon, M. Hebert, and T. Kanade. Perception for rugged terrain. In *SPIE, Advances in Intelligent Robotics Systems*, Society of Photo-Optical Instrumentation Engineers, Cambridge, Massachusetts, November 1988.

[Lucas84] B. D. Lucas. *Generalized Image Matching by the Method of Differences.* PhD thesis, Computer Science Department, Carnegie-Mellon University, July 1984.

[Matthies88] L. Matthies, R. Szeliski, and T. Kanade. Incremental estimation of dense depth maps from image sequences. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 366–374, IEEE Computer Society Press, Ann Arbor, MI, June 1988.

[Maybeck79] P. S. Maybeck. *Stochastic Models, Estimation, and Control.* Volume 1, Academic Press, New York, NY, 1979.

[Olin88] K. E. Olin, M. J. Daily, J. G. Harris, and K. Reiser. Knowledge-based vision technology progress at Hughes AI Center. In *Image Understanding Workshop*, pages 88–93, Morgan Kaufmann Publishers, Cambridge, MA, April 1988.

[Poggio85] T. Poggio, V. Torre, and C. Koch. Computational vision and regularization theory. *Nature*, 317(6035):314–319, 26 September 1985.

[Press86] W. Press et al. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, Cambridge, 1986.

[Stewart87] W. K. Stewart. A non-deterministic approach to 3-D modeling underwater. In *Fifth International Symposium on Unmanned Untethered Submersible Technology*, pages 283–309, University of New Hampshire Marine Systems Engineering Laboratory, June 1987.

[Szeliski87] R. Szeliski. Regularization uses fractal priors. In *AAAI-87: Sixth National Conference on Artifical Intelligence*, pages 749–754, Morgan Kaufmann Publishers, Seattle, Washington, July 1987.

[Szeliski88] R. Szeliski. *Bayesian Modeling of Uncertainty in Low-Level Vision*. PhD thesis, Carnegie Mellon University, August 1988.

[Terzopoulos84] D. Terzopoulos. *Multiresolution Computation of Visible-Surface Representations*. PhD thesis, Massachusetts Institute of Technology, January 1984.

[Terzopoulos86] D. Terzopoulos. Regularization of inverse visual problems involving discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(4):413–424, July 1986.

[Terzopoulos87] D. Terzopoulos, A. Witkin, and M. Kass. Symmetry-seeking models for 3-D object reconstruction. In *First International Conference on Computer Vision*, pages 269–276, IEEE Computer Society Press, London, England, June 1987.

[Tsai84] R. Y. Tsai and T. S. Huang. Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(1):13–27, January 1984.

[Ullman79] S. Ullman. *The Interpretation of Visual Motion*. MIT Press, Cambridge, MA, 1979.

[Witkin86] A. Witkin, D. Terzopoulos, and M. Kass. Signal matching through scale space. In *AAAI-86: Fifth National Conference on Artificial Intelligence*, pages 714–719, Philadelphia, August 1986.

A Derivation of probability distributions

To re-write the negative log likelihood equation (21) in terms of sparse quantities such as $\mathbf{R}_2^{-1}$, $\mathbf{P}_1^{-1}$ and $\mathbf{P}_2^{-1}$, we will need to use two Lemmas. The first is a matrix inversion Lemma ([Maybeck79], p. 280)

$$(\mathbf{H}\mathbf{P}\mathbf{H}^T+\mathbf{R})^{-1} = \mathbf{R}^{-1}-\mathbf{R}^{-1}\mathbf{H}(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}+\mathbf{P}^{-1})^{-1}\mathbf{H}^T\mathbf{R}^{-1} \quad (26)$$

where $\mathbf{H}$ need not be a square matrix. The second Lemma is related to matrix determinants

$$|\mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{R}|^{-1} = |\mathbf{R}^{-1}||\mathbf{P}^{-1}||\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{P}^{-1}|^{-1} \quad (27)$$

([Maybeck79], p 280).

To apply the above two lemmas, we note that

$$\mathbf{P}_2^{-1} = \mathbf{P}_1^{-1} + \mathbf{H}_2^T\mathbf{R}_2^{-1}\mathbf{H}_2 \quad (28)$$

and

$$\mathbf{P}_2^{-1}\hat{\mathbf{u}}_2 = \mathbf{P}_1^{-1}\hat{\mathbf{u}}_1 + \mathbf{H}_2^T\mathbf{R}_2^{-1}\mathbf{d}_2. \quad (29)$$

The equation for E_1 (23) is obtained immediately by applying (27) and (28) to the first term in (21). The equation for E_2 (24) is derived from the second term of (21) by applying (26), (28) and (29)

$$\begin{aligned}
E_2(\mathbf{d}_2) &= \frac{1}{2}(\mathbf{d}_2 - \mathbf{H}_2\hat{\mathbf{u}}_1)^T[\mathbf{R}_2^{-1} \\
&\quad - \mathbf{R}_2^{-1}\mathbf{H}_2(\mathbf{P}_1^{-1} + \mathbf{H}_2^T\mathbf{R}_2^{-1}\mathbf{H}_2)^{-1}\mathbf{H}_2^T\mathbf{R}_2^{-1}](\mathbf{d}_2 - \mathbf{H}_2\hat{\mathbf{u}}_1) \\
&= \frac{1}{2}(\mathbf{d}_2 - \mathbf{H}_2\hat{\mathbf{u}}_1)^T\mathbf{R}_2^{-1}[(\mathbf{d}_2 - \mathbf{H}_2\hat{\mathbf{u}}_1) \\
&\quad - \mathbf{H}_2\mathbf{P}_2(\mathbf{H}_2^T\mathbf{R}_2^{-1}\mathbf{d}_2 - \mathbf{H}_2^T\mathbf{R}_2^{-1}\mathbf{H}_2\hat{\mathbf{u}}_1)] \\
&= \frac{1}{2}(\mathbf{d}_2 - \mathbf{H}_2\hat{\mathbf{u}}_1)^T\mathbf{R}_2^{-1}[(\mathbf{d}_2 - \mathbf{H}_2\hat{\mathbf{u}}_1) \\
&\quad - \mathbf{H}_2\mathbf{P}_2((\mathbf{P}_2^{-1}\hat{\mathbf{u}}_2 - \mathbf{P}_1^{-1}\hat{\mathbf{u}}_1) - (\mathbf{P}_2^{-1} - \mathbf{P}_1^{-1})\hat{\mathbf{u}}_1)] \\
&= \frac{1}{2}(\mathbf{d}_2 - \mathbf{H}_2\hat{\mathbf{u}}_1)^T\mathbf{R}_2^{-1}(\mathbf{d}_2 - \mathbf{H}_2\hat{\mathbf{u}}_1) \\
&\quad - \frac{1}{2}(\mathbf{d}_2 - \mathbf{H}_2\hat{\mathbf{u}}_1)^T\mathbf{R}_2^{-1}\mathbf{H}_2(\hat{\mathbf{u}}_2 - \hat{\mathbf{u}}_1) \\
&= \frac{1}{2}(\mathbf{d}_2 - \mathbf{H}_2\hat{\mathbf{u}}_1)^T\mathbf{R}_2^{-1}(\mathbf{d}_2 - \mathbf{H}_2\hat{\mathbf{u}}_2)
\end{aligned}$$

which is the desired result.

To derive the second form for E_2 (25), we start with the energy (negative log probability) of the joint probability density

$$E(\mathbf{u}, \mathbf{d}_1, \mathbf{d}_2) = \frac{1}{2}\mathbf{u}^T\mathbf{P}_0^{-1}\mathbf{u} + \frac{1}{2}\sum_{i=1}^{2}(\mathbf{d}_i - \mathbf{H}_i\mathbf{u})^T\mathbf{R}_i^{-1}(\mathbf{d}_i - \mathbf{H}_i\mathbf{u}).$$

To marginalize with respect to $\mathbf{u}$, we simply replace this variable with its minimum energy solution with respect to the other variables, which is $\hat{\mathbf{u}}_2$.

Acknowledgements

I would like to thank In So Kweon and Martial Hebert for discussions which inspired this work, and Takeo Kanade for suggesting some of the experiments. I also thank Takeo Kanade and Martial Hebert for helpful comments on the paper.