Probabilistic Modeling of Surfaces

Richard Szeliski

Digital Equipment Corporation, Cambridge Research Lab
One Kendall Square, Bldg. 700, Cambridge, MA 02139

Abstract

Energy-based surface models are commonly used in computer vision to interpolate sparse data, to smooth noisy depth estimates, and to integrate measurements from multiple sensors and viewpoints. Traditionally, a single surface estimate is produced with such models. Probabilistic surface modeling, which describes distributions over possible surfaces, enables us to integrate such measurements in a statistically optimal fashion, to model the uncertainty in the surfaces, and to develop sequential estimation algorithms. When applied to 2-½-D surfaces, probabilistic modeling allows us to incrementally estimate depth maps from motion image sequences and to integrate sparse range data using elevation maps. To obtain more accurate models of depth, we show how to jointly model depth and intensity images.

To better represent the structure of the visual world, we must use full 3-D surface models. These are usually represented using parametric surfaces, which can create difficulties when the surface topology is unknown. To overcome these problems, we develop an incremental patch-based 3-D surface estimation algorithm. We then compare surface and feature-based methods, and propose a unified representation which encompasses both methods.

1. Introduction

Surfaces are widely used in computer vision as an intermediate representation for modeling the shape of our environment. When combined with other attributes such as color and reflectivity, the geometric description provided by surfaces is sufficient to account for most of the optical phenomena we see in real-world images. Surface descriptions can be used directly in a number of vision-based tasks such as navigation (obstacle avoidance), manipulation (choice of grasp points), and automatic 3-D model acquisition. They can also be used as an intermediate stage in object recognition algorithms, since they provide a more stable and useful description than the original intensity images.

Probabilistic models are a powerful tool for dealing with the noisy nature of real-world sensors and for incorporating external (*a priori*) knowledge about problem domains. They are particularly useful when we aggregate information from multiple modalities (multisensor fusion) and/or incorporate information over time (sequential estimation). Probabilistic models not only allow us to compute optimal estimates, but also give us a quantitative measure of the *uncertainty* in these estimates.

The application of probabilistic models to surface descriptions is a fairly recent development [GG84, Sze89]. This is because the probabilistic modeling of surfaces poses a number of fundamental problems and limitations. First, the high dimensionality of surface-based descriptions makes it expensive to model higher order statistics such as covariances. Fortunately, Markov Random Field models [GG84] can help us here. Second, while parametric models of surfaces have attractive properties such as viewpoint invariance, the automatic selection of parameterizations remains an open problem. Third, while researchers have developed optimal sequential estimation algorithms for pure geometric entities such as points, lines, and planes [Aya91], we cannot yet estimate surfaces with comparable accuracy.

This paper surveys the probabilistic modeling of two and three-dimensional surfaces and provides novel solutions to some of these outstanding problems. We begin with a review of visible surface modeling, Bayesian modeling, and uncertainty modeling applied to 2½-D surfaces. We show how these probabilistic models can be used to incrementally estimate surfaces from multiple optic flow and range data measurements. We introduce a new coupled depth and intensity estimation model and show how it improves the convergence rate of incremental surface estimation. Turning to 3-D surfaces, we review parametric surfaces and discuss the difficulty of determining good parameterizations for complex objects. We propose a solution which involves sequentially estimating surface elements and then postprocessing them with a particle-based surface interpolator. We present a comparison of surface-based and feature-based representations, and close with a proposal for merging the two.

 0-8194-0698-8/91/$4.00

2. Visible surface modeling

The traditional approach to modeling visible surfaces in low-level and intermediate-level vision is to use a collection of two-dimensional piecewise continuous functions computed directly from input images. Such representations were first suggested by Marr [Mar78], whose *2 ½-dimensional (2 ½-D) sketch* encodes local surface orientation and distance to the viewer as well as discontinuities in the orientation and distance maps, and by Barrow and Tenenbaum [BT78], whose *intrinsic images* represent scene characteristics such as distance, orientation, reflectance, and illumination in multiple retinotopic maps.

The computational theory of visible surfaces has been formalized using a number of techniques, including variational principles [Gri83], regularization [PTK85, Ter88], and physically-based modeling [TWK87]. In regularization, a visible surface is computed from a set of constraints d (such as those provided by stereo matches) by finding the function u which minimizes the weighted sum of two energy functionals

$$\mathcal{E}(u) = \mathcal{E}_{\mathrm{d}}(u, d) + \lambda \mathcal{E}_{\mathrm{s}}(u). \tag{1}$$

The *data compatibility* functional $\mathcal{E}_{\mathrm{d}}(u, d)$ measures the distance between the solution and the sampled data d, the *stabilizing* functional $\mathcal{E}_{\mathrm{s}}(u)$ measures the smoothness of the solution. The regularization parameter λ controls the amount of smoothing performed.

An example of regularization applied to visible surface modeling is the interpolation of a piecewise continuous surface $u(x, y)$ through a sparse set of data points $\{(x_i, y_i, d_i)\}$. The data compatibility term in this case is a weighted sum of squares

$$\mathcal{E}_{\mathrm{d}}(u, d) = \frac{1}{2} \sum c_i [u(x_i, y_i) - d_i]^2, \tag{2}$$

where the confidence c_i is inversely related to the variance of the measurement d_i, i.e., $c_i = \sigma_i^{-2}$. The smoothness functional is

$$\mathcal{E}_{\mathrm{s}}(u) = \frac{1}{2} \int \int \rho(x, y)\{[1 - \tau(x, y)][u_x^2 + u_y^2] + \tau(x, y)[u_{xx}^2 + 2u_{xy}^2 + u_{yy}^2]\}\, dx\, dy, \tag{3}$$

where $\rho(x, y)$ is a *rigidity* function, and $\tau(x, y)$ is a *tension* function [Ter86b]. The rigidity and tension functions are used to introduce depth ($\rho(x, y) = 0$) and orientation ($\tau(x, y) = 0$) discontinuities (Figures 1a and 1b). The minimum energy solution of the above system is a *thin plate surface under tension* [Ter86b].

To compute a numerical solution to a regularized problem, we first convert the functionals $\mathcal{E}_{\mathrm{d}}(u, d)$ and $\mathcal{E}_{\mathrm{s}}(u)$ to discrete energy functions using finite element analysis [Ter88, Sze89]. If we fix the continuity control functions $\rho(x, y)$ and $\tau(x, y)$ and discretize the surface using a fine rectangular mesh, these energy functions are quadratic with a simple regular structure.[1] The data compatibility function becomes

$$E_{\mathrm{d}}(\mathbf{u}, \mathbf{d}) = \frac{1}{2}(\mathbf{u} - \mathbf{d})^T \mathbf{A}_{\mathrm{d}} (\mathbf{u} - \mathbf{d}), \tag{4}$$

where $\mathbf{u}$ is the discretized surface, $\mathbf{d}$ are the data points, and $\mathbf{A}_{\mathrm{d}}$ is a diagonal matrix (for uncorrelated sensor noise). The discrete smoothness energy is

$$E_{\mathrm{s}}(\mathbf{u}) = \frac{1}{2} \mathbf{u}^T \mathbf{A}_{\mathrm{s}} \mathbf{u}, \tag{5}$$

where $\mathbf{A}_{\mathrm{s}}$ is sparse and banded, with a bandwidth equal to one of the image dimensions.[2] The rows of $\mathbf{A}_{\mathrm{s}}$ can be described in terms of computational molecules [Ter88].

The resulting total energy function $E(\mathbf{u})$ is quadratic in $\mathbf{u}$

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

with

$$\mathbf{A} = \mathbf{A}_{\mathrm{d}} + \lambda \mathbf{A}_{\mathrm{s}} \text{ and } \mathbf{b} = \mathbf{A}_{\mathrm{d}} \mathbf{d}. \tag{7}$$

The energy function has a minimum at $\mathbf{u}^*$, the solution to the linear system of algebraic equations

$$\mathbf{A}\mathbf{u} = \mathbf{b}. \tag{8}$$

[1] We prefer discretizations that do not depend on the data since these are easier to use in sequential estimation applications.

[2] In finite element analysis, $\mathbf{A}_{\mathrm{s}}$ is called the *stiffness matrix*.

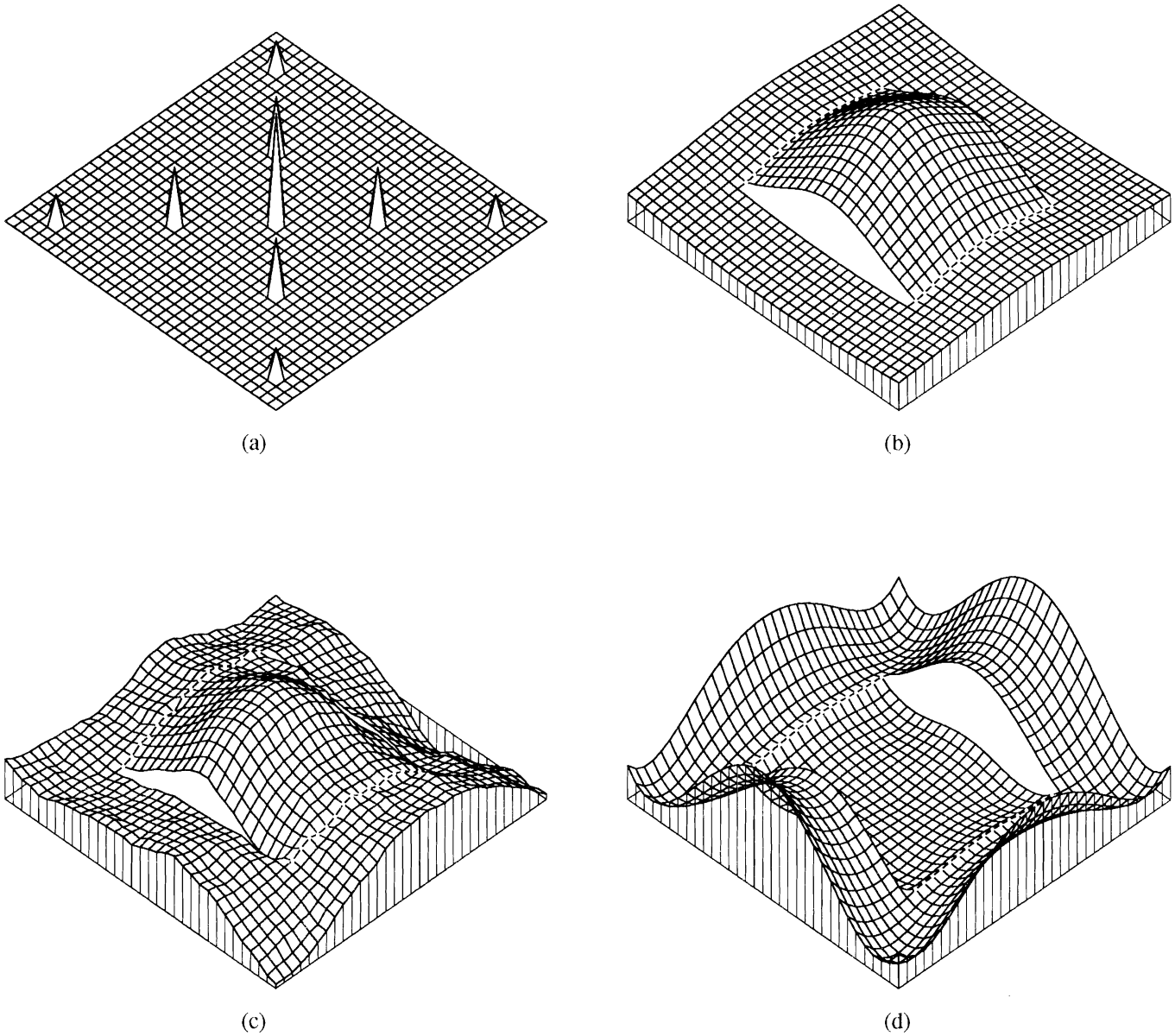


Figure 1: Sample data and interpolated surface: (a) data points, (b) thin plate solution with two depth and two orientation discontinuities, (c) random fractal sample from posterior distribution, (d) uncertainty (variance) field (see Section 4). The depth discontinuities are shown as missing line segments, while the orientation discontinuities appear as white dots at the nodes.

The energy can thus be rewritten as

$$E(\mathbf{u}) = \frac{1}{2}(\mathbf{u} - \mathbf{u}^*)^T \mathbf{A}(\mathbf{u} - \mathbf{u}^*) + k. \tag{9}$$

Once we have derived the discrete energy function, we can use a variety of techniques to find the minimum energy solution $\mathbf{u}^*$. For large sparse systems such as the ones obtained with our fine discretization, the most efficient and parallelizable techniques are iterative relaxation algorithms such as successive overrelaxation [BZ87], multigrid relaxation [Ter86a], and hierarchical basis conjugate gradient descent [Sze90a]. Alternative multiresolution representations of the surface can also be used [ST89b, Sze89]. The design of efficient and robust algorithms for computing visible surface representations, which includes the important problem of discontinuity detection, has been the subject of a great deal of research in computer vision (see [BZ87, Ter88, Sze89] for reviews).

3. Bayesian modeling

A Bayesian model is a statistical description of an estimation problem that consists of two separate components. The first component, the *prior model*, $p(\mathbf{u})$, describes the probability distribution of our state $\mathbf{u}$ (in this case, a surface) in the absence of any sensed data. The second component, the *sensor model*, $p(\mathbf{d}|\mathbf{u})$, describes the probability of sensing values $\mathbf{d}$ if the surface $\mathbf{u}$ is viewed. These two probabilistic models can be combined to obtain a *posterior model*, $p(\mathbf{u}|\mathbf{d})$, which allows us to draw the backward inference, describing the probability that the surface $\mathbf{u}$ has been viewed given that data values $\mathbf{d}$ have been sensed. To compute this posterior model we use Bayes' Rule

$$p(\mathbf{u}|\mathbf{d}) = \frac{p(\mathbf{d}|\mathbf{u})\, p(\mathbf{u})}{p(\mathbf{d})}, \tag{10}$$

with the normalizing denominator

$$p(\mathbf{d}) = \sum_{\mathbf{u}} p(\mathbf{d}|\mathbf{u}).$$

When applied to surface representations, the prior model is used to bias the solutions towards smooth surfaces, i.e., to encode the smoothness constraint [Sze87, Sze89]. This can be done conveniently by using a Gibbs (or Boltzmann) distribution of the form

$$p(\mathbf{u}) = \frac{1}{Z_s} \exp(-E_s(\mathbf{u})), \tag{11}$$

where $E_s(\mathbf{u})$ is the discrete smoothness energy defined in (5), and Z_s (called the *partition function*) is a normalizing constant. Because the energy function $E_s(\mathbf{u})$ can be written as a sum of local clique energies, the prior distribution (11) is a Markov Random Field [GG84].

For surface interpolation, the sensor model is a discrete sampling of the surface $\mathbf{u}$ with white (independent) Gaussian noise added to each measurement. This multivariate Gaussian distribution can be written as

$$p(\mathbf{d}|\mathbf{u}) = \frac{1}{Z_d} \exp(-E_d(\mathbf{u}, \mathbf{d})), \tag{12}$$

where $E_d(\mathbf{u}, \mathbf{d})$ is given by (4). More sophisticated sensors models can easily be developed and used within this Bayesian framework [Sze89, ST91a].

We are now in a position to derive the posterior distribution $p(\mathbf{u}|\mathbf{d})$ using Bayes' Rule. From (10), (11) and (12) we have

$$p(\mathbf{u}|\mathbf{d}) = \frac{p(\mathbf{d}|\mathbf{u})\, p(\mathbf{u})}{p(\mathbf{d})} = \frac{1}{Z} \exp(-E(\mathbf{u})), \tag{13}$$

where

$$E(\mathbf{u}) = E_d(\mathbf{u}, \mathbf{d}) + E_s(\mathbf{u}). \tag{14}$$

The *maximum a posteriori* (MAP) estimate $\hat{\mathbf{u}}$, i.e., the value of $\mathbf{u}$ that maximizes the conditional probability $p(\mathbf{u}|\mathbf{d})$, is the same as the surface $\mathbf{u}^*$ which minimizes the discrete energy $E(\mathbf{u})$ obtained from regularization.

4. Probabilistic surface modeling

While energy-based and Bayesian modeling may ultimately yield the same estimate, there are several advantages to the probabilistic formulation. First, the statistical assumptions corresponding to a smoothness constraint can be explored by randomly generating samples from the prior model [Sze87]. This also gives us a powerful method for generating stochastic surfaces such as fractals (Figure 1c) [ST89a]. Second, the data constraint energies can be derived in a principled fashion from the known noise characteristics of the sensors [Sze89, ST91a]. Third, the uncertainty in the posterior model can be quantified, as we show below. Fourth, Bayesian modeling can be used to integrate multiple measurements [MKS89], as we show in Section 5. Additional uses and advantages of the probabilistic modeling of visual primitives can be found in [Sze89, ST91a, Aya91].

To compute the uncertainty in our posterior estimate, we note that the posterior distribution $p(\mathbf{u}|\mathbf{d})$ given by (13) is a Gibbs distribution with a quadratic energy given by (9). This distribution is a multivariate Gaussian with mean $\mathbf{u}^*$ and covariance $\mathbf{A}^{-1}$. Thus, to characterize the uncertainty, we need only invert the matrix $\mathbf{A}$.

In practice, computing and storing $\mathbf{A}^{-1}$ is not feasible for surfaces, because while $\mathbf{A}$ is sparse and banded, $\mathbf{A}^{-1}$ is not. We can obtain a reduced description of the uncertainty if we compute only the diagonal elements of $\mathbf{A}^{-1}$, i.e., the variance at each point on the surface [Sze89]. We have developed two methods to compute this variance. The first involves computing the values sequentially by replacing the right hand side of (8) with unit vectors. The second method uses a Monte-Carlo approach which generates random samples from the posterior distribution and accumulates the desired statistics [Sze89]. The uncertainty (variance) map for our interpolated surface is shown in Figure 1d.

The uncertainty modeling method we have developed is just one of several possible approaches. Spatial likelihood maps have been developed for surfaces represented in spherical coordinates [Chr87]. Occupancy maps [EM87] indicate the likelihood of a surface being present in a two- or three-dimensional array. The advantage of our energy-based formulation is that it explicitly models the correlation between adjacent points on the surface.

Uncertainty maps can be used to grow a "confidence region" around the surface estimate, indicating an envelope within which the surface is likely to lie. This can be useful in navigation and manipulation applications, and can also be used to determine where additional sensing would be helpful (active vision). However, the most useful application of uncertainty modeling is in the sequential estimation of surface shape, as we discuss next.

5. Incremental surface estimation

Bayesian models of surfaces are particularly well suited to integrating information from multiple measurements (multisensor fusion) or estimating surface shapes over time (sequential estimation). When the surfaces or observer are moving or changing over time, our problem becomes one of dynamic system estimation. Such problems can be solved using either batch processing techniques such as epipolar plane image analysis [BBM87] or by sequential estimation algorithms such as the Kalman filter [Gel74]. The advantage of the Kalman filter is that estimates are available immediately and that the storage costs are reduced.

The Kalman filter extends the Bayesian model by adding a *system model* to the prior and sensor models. In the Kalman filter, the prior model is a multivariate Gaussian with mean $\hat{\mathbf{u}}_0$ and covariance $\mathbf{P}_0$ denoted by

$$\mathbf{u} \sim N(\hat{\mathbf{u}}_0, \mathbf{P}_0). \tag{15}$$

For surface estimation problems, we set $\hat{\mathbf{u}}_0 = \mathbf{0}$ and $\mathbf{P}_0^{-1} = \mathbf{A}_s$. The sensor model relates each new measurement vector $\mathbf{d}_k$ to the current state $\mathbf{u}_k$ through a measurement matrix $\mathbf{H}_k$ and the addition of Gaussian noise $\mathbf{r}_k$,

$$\mathbf{d}_k = \mathbf{H}_k \mathbf{u}_k + \mathbf{r}_k, \quad \mathbf{r}_k \sim N(0, \mathbf{R}_k). \tag{16}$$

For surface interpolation, we let $\mathbf{H}_k = \mathbf{I}$ and $\mathbf{R}_k^{-1} = \mathbf{A}_d$. The system model describes the evolution of the current state vector $\mathbf{u}_k$ over time using a known transition matrix $\mathbf{F}_k$ and the addition of Gaussian noise $\mathbf{q}_k$,

$$\mathbf{u}_k = \mathbf{F}_k \mathbf{u}_{k-1} + \mathbf{q}_k, \quad \mathbf{q}_k \sim N(0, \mathbf{Q}_k). \tag{17}$$

To compute the current state estimate $\hat{\mathbf{u}}_k$, we first *predict* or *extrapolate* the old estimate $\hat{\mathbf{u}}_{k-1}$ and its covariance $\mathbf{P}_{k-1}$

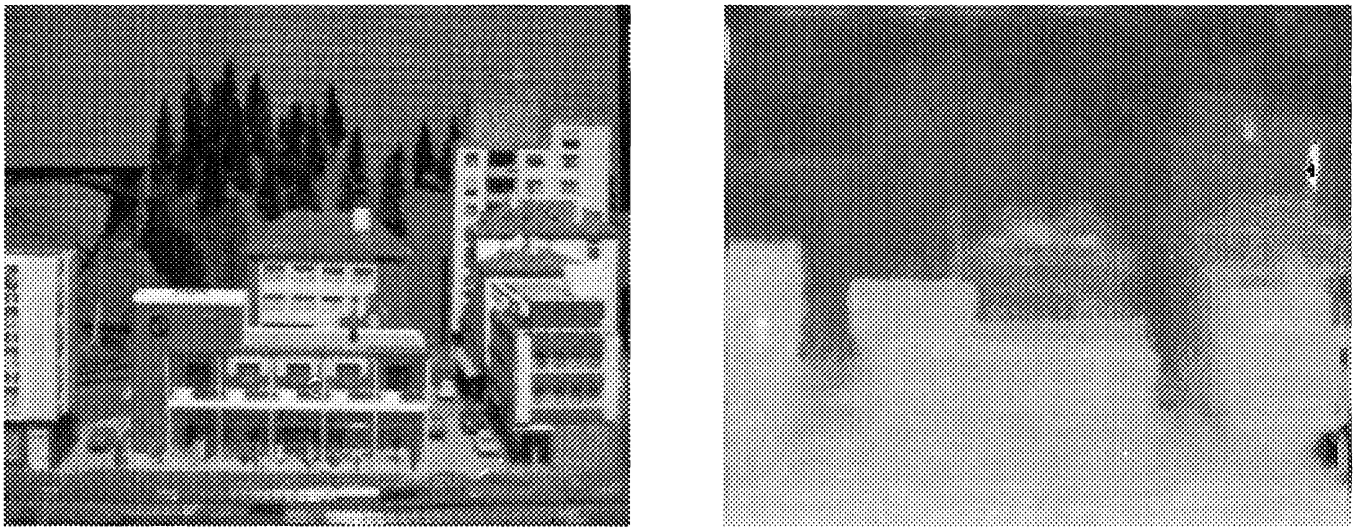
$$\tilde{\mathbf{u}}_k = \mathbf{F}_k \hat{\mathbf{u}}_{k-1} \tag{18}$$

$$\tilde{\mathbf{P}}_k = \mathbf{F}_k \mathbf{P}_{k-1} \mathbf{F}_k^T + \mathbf{Q}_k. \tag{19}$$

We then *correct* or *update* the estimate and its covariance

$$\hat{\mathbf{u}}_k = \tilde{\mathbf{u}}_k + \mathbf{P}_k \mathbf{H}_k^T \mathbf{R}_k^{-1} (\mathbf{d}_k - \mathbf{H}_k \tilde{\mathbf{u}}_k) \tag{20}$$

$$\mathbf{P}_k^{-1} = \tilde{\mathbf{P}}_k^{-1} + \mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k. \tag{21}$$



(a) (b)

Figure 2: Depth map computed from image sequence: (a) first frame of image sequence, (b) intensity-coded depth map computed from combined sequence of horizontal and vertical motions

Of the above four equations, the first three are expressed in terms of the covariance matrices $\mathbf{P}_k$, while the fourth uses the inverse covariance (or *information*) matrices (which we can denote by $\mathbf{A}_k$). For surface modeling, we showed in Section 4 that the information matrices are sparse, while the covariance matrices are not. Our implementation of the Kalman filter for surfaces therefore uses information matrices to model the uncertainty. This involves performing a linear system solution of

$$\mathbf{A}_k \Delta\hat{\mathbf{u}}_k = \mathbf{H}_k^T \mathbf{R}_k^{-1} (\mathbf{d}_k - \mathbf{H}_k \tilde{\mathbf{u}}_k)$$

in (20), and finding a way to implement (19) using inverse covariances. This can be achieved by partitioning the information matrices into a diagonal matrix arising from the measurements ($\mathbf{H}_k^T \mathbf{R}_k^{-1} \mathbf{H}_k^T$ in (21)), and a banded matrix encoding the smoothness constraint ($\mathbf{P}_0^{-1}$) which is assumed not to vary over time [Sze89]. Similar ideas can be applied to other physically-based models in computer vision [ST91a].

To demonstrate the utility of the sequential estimation of surfaces, we will briefly review two applications. The first algorithm builds a dense depth map from a sequence of images where the motion of the observer is known [MKS89]. The input to this algorithm consists of optic flow fields computed from successive pairs of images. These flow fields are converted into *disparity* (inverse depth) maps, which are then aggregated over time using the 2-D Kalman filter. Regularization-based smoothing is used to reduce the noise in the flow measurements and to fill in areas where flow was not reliably estimated. A key feature of this method is that the variance of each flow measurement is estimated locally from the shape of the correlation surface, and this variance is propagated through the Kalman filter [MKS89]. To keep the representation *iconic* (2-D image-based), the disparity and uncertainty maps are warped (re-sampled) between frames using the current disparity estimates to predict the amount of inter-frame motion.

A second application is the registration and fusion of multiple range data images obtained from a moving vehicle [Sze88]. A single stationary terrain map is used to fuse the multiple measurements. The estimated position of the observer is refined by finding the motion which minimizes the distance between the new measurements and the surface. Because large areas of this map may be "shadowed" (not visible) from earlier viewpoints (Figure 3), the direct matching of new measurements to the old surface leads to incorrect motion estimates. Using a Bayesian model which takes into account the spatially varying uncertainty in the surface, we can derive a statistically optimal metric which uses both the old and new surface estimates in the computation of the distance [Sze88]. An additional advantage of the Bayesian model is that we can compute the uncertainty in the motion estimate directly from the shape of the error surface.

6. Joint estimation of depth and intensity

While the incremental computation of visible surfaces based on probabilistic surface modeling has produced impressive results, its theoretical convergence rate is poorer than that of feature-based approaches [MKS89]. An intuitive way to see why is to

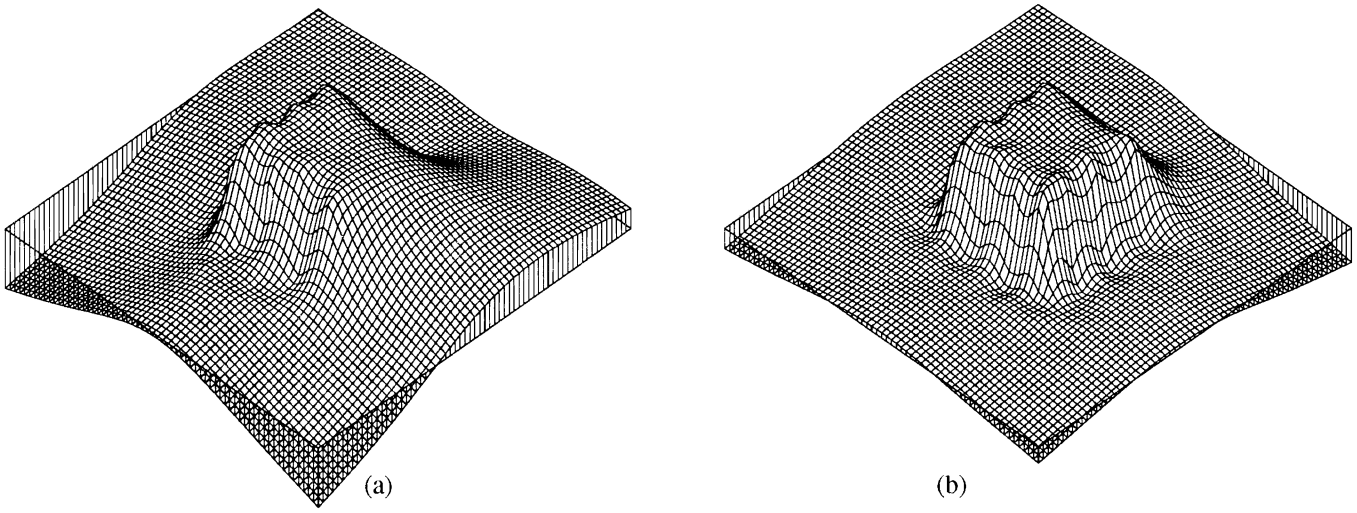


Figure 3: Motion estimation from range data [Sze88]. Two data sets (not shown) are used to incrementally compute the surface. Shown are the interpolated surface computed from (a) first sparse block data set (b) both sparse block data sets

consider disparity estimation as line fitting in a spatio-temporal cube of image data. An edge-based estimator which estimates both the disparity (slope) and sub-pixel location (intercept) of these lines has a variance of $\sigma_F^2(n) \propto 1/n^3$, where n is the number of images [MKS89]. An iconic algorithm which averages successive displacements is equivalent to a stereo match between the first and last image, and therefore has $\sigma_I^2(n) \propto 1/n^2$.

The reduced rate of convergence occurs because we ignore the temporal correlations between successive flow measurement [MKS89]. One way of compensating for such correlated measurements is to introduce additional state variables [Gel74]. In our case, the most natural choice of variable is the intensity distribution over the surface. To achieve the optimal rate of convergence, we estimate both the disparity and intensity fields and model the correlation between these two fields.

In our new formulation, it is no longer necessary to use a separate flow computation module (although we may wish to use one initially to provide more robust disparity estimates). Instead, we use a single measurement equation which relates new images to the underlying disparity and intensity fields. The equations defining the temporal evolution of the intensity f and disparity d fields (assuming no occlusions) for a camera translating horizontally are

$$f(x + \Delta t\, d(x,y,t), y, t + \Delta t) = f(x,y,t) \tag{22}$$

$$d(x + \Delta t\, d(x,y,t), y, t + \Delta t) = d(x,y,t). \tag{23}$$

Given a sequence of noisy sampled images

$$g(x,y,t) = f(x,y,t) + n(x,y,t), \tag{24}$$

we could solve for the intensity and disparity at time T using a batch minimization algorithm. With the addition of appropriate smoothness constraints on f and d, this would be *regularized depth from motion*.

If we wish to estimate the current intensity and disparity images in an incremental fashion, we can use the *extended Kalman filter* [Gel74, p. 188]. In this model, we replace (17) and (16) with

$$\mathbf{u}_k = \mathbf{f}_k(\mathbf{u}_{k-1}) + \mathbf{q}_k, \quad \mathbf{q}_k \sim N(0, \mathbf{Q}_k) \tag{25}$$

$$\mathbf{d}_k = \mathbf{h}_k(\mathbf{u}_k) + \mathbf{r}_k, \quad \mathbf{r}_k \sim N(0, \mathbf{R}_k). \tag{26}$$

We then use the same updating equations as before with

$$\mathbf{F}_k = \frac{\partial \mathbf{f}_k}{\partial \mathbf{u}}(\hat{\mathbf{u}}_{k-1}) \tag{27}$$

$$\mathbf{H}_k = \frac{\partial \mathbf{h}_k}{\partial \mathbf{u}}(\tilde{\mathbf{u}}_k) \tag{28}$$

To apply the extended Kalman filter to (22) and (23), we must discretize the f and d functions in both space and time. This leads to a set of warping equations

$$f_k(i,j) = \text{interpolate}(\{(i + d_{k-1}(i,j), j, f_{k-1}(i,j)\})(i,j) \quad (29)$$

$$d_k(i,j) = \text{interpolate}(\{(i + d_{k-1}(i,j), j, d_{k-1}(i,j)\})(i,j), \quad (30)$$

i.e., the new intensity and disparity fields are obtained by interpolating through the collection of shifted intensity and disparity estimates and then re-sampling [MKS89]. The system transition matrix $\mathbf{F}_k$, which is the Jacobian of the above set of non-linear equations, models the dependence of the new states on the old states. In particular, $\mathbf{F}_k$ contains entries which link the new intensities to the old disparities through an approximation to the intensity gradient.

Because the covariance matrix of the predicted fields models the correlations between the intensity and disparity estimates, a simple discrete measurement equation based on (24) is sufficient to update both fields. A disadvantage of the above formulation is that if the previous estimate $\hat{\mathbf{u}}_{k-1}$ of $f(x,y)$ and $d(x,y)$ was not accurate, then the value of $\mathbf{F}_k$ will not be very good. A better solution is to use (29) itself as the measurement equation. We can then use the *iterated extended Kalman filter* [Gel74, p. 190] to repeatedly calculate $\hat{\mathbf{u}}_k$ and $\mathbf{H}_k$ until good convergence is obtained.

While it has yet to be tested empirically, the joint modeling of intensity and disparity has the potential for improving the accuracy of depth from motion algorithms and for simplifying their implementation.

7. 3-D surface modeling

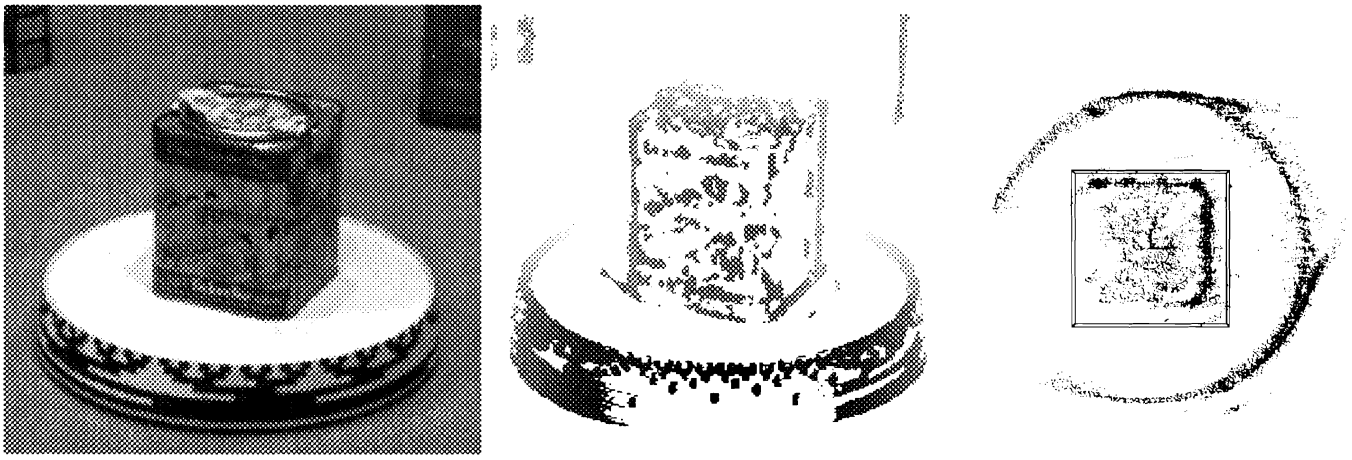
The computation of visible surface representations (2-D depth or elevation maps) is a useful step in the construction of higher-level shape descriptions, and can also be used directly in a number of applications such as obstacle avoidance. For many vision applications, however, we need a representation that can integrate surface descriptions from widely disparate views. Such applications require the use of full 3-D surface models, which are generally viewpoint invariant and can represent parts of the surface that are not currently visible. For example, we may wish to represent the full 3-D shape of an object as it rotates in front of a camera [Sze91]. Alternatively, we may wish to model the shape of the environment behind a mobile robot which was seen from previous viewpoints.

Choosing an appropriate representation for 3-D surface is not as straightforward as it was for visible surfaces, where scalar functions defined over two-dimensional domains provided a natural and convenient representation. The simplest way to extend visible surfaces to 3-D is to use *parametric surfaces*, where the 3-D coordinates of a surface $\mathbf{x} = [X\ Y\ Z]$ are functions of the underlying parameters $(u,v) \in [0,1]^2$. The elastic properties of the surface can be specified by applying the same smoothness energies as were used for the piecewise continuous spline under tension (3) to each component of $\mathbf{x}$ independently. The resulting patch behaves somewhat like a deformable sheet of paper or a thin stretchable membrane.

To obtain a 3-D surface more suited to modeling true 3-D objects, we can "seam" together the two opposite edges of the parametric sheet $u = 0$ and $u = 1$ to obtain a deformable tube model. The *symmetry-seeking models* of Terzopoulos, Witkin, and Kass [TWK87], couple this tube model $\mathbf{x}_T(u,v)$ with a deformable spine $\mathbf{x}_S(v)$ to obtain a physically-based model that responds to image forces. A simpler, though less flexible, version of this model is a cylindrical representation, where the radius is a function of angle and height $r(\theta, z)$. Other primitives that may be suitable for modeling 3-D surfaces include *generalized cylinders* [BGB79], where an arbitrary cross section is swept along a spine curve, and *superquadrics* [Pen86].

Probabilistic surface modeling can be applied to 3-D parametric surfaces just as easily as it was to 2-½-D surfaces. Because surface modeling is always a combination of internal smoothness constraints and external data fitting constraints, we can still build prior and sensor models to reflect these two components, using the Gibbs distribution to link energies with probabilities. As the energies become more complicated, we may no longer be able to model the surface as a correlated Gaussian field (because the energies are not quadratic), but we can still develop appropriate Bayesian models. Sequential estimation algorithms based on the extended Kalman filter can still be developed, although will no longer be optimal compared to batch algorithms. Nevertheless, the same advantages originally obtained using probabilistic models, which include the ability to sensibly weight new measurements and to model the uncertainty in our estimates, are still available.

A potential disadvantage of a parametric representation is that many different parameterizations can be used to represent the same surface, which can complicate the matching of surfaces in recognition tasks. A solution to this problem is to reparameterize the sheet using canonical parameters based on local curvature [VTL89]. A more serious disadvantage is that the topology of the parametric surface (and sometimes even its rough shape) must be known in advance before it can be fitted to data. In computer graphics, where the emphasis is on modeling, this problem has attracted a fair deal of attention [LD90]. In computer vision, the application of parametric surfaces has been limited to simple topologies such as sheets [VTL89] or cylinders [TWK87]. A more general solution, based on systems of interacting particles, will be presented at the end of the next section.



a b c

Figure 4: assam image sequence: (a) first image (b) depth map from flow (darker is nearer) (c) top view of 3-D point cloud

8. Incremental 3-D patch/point estimation

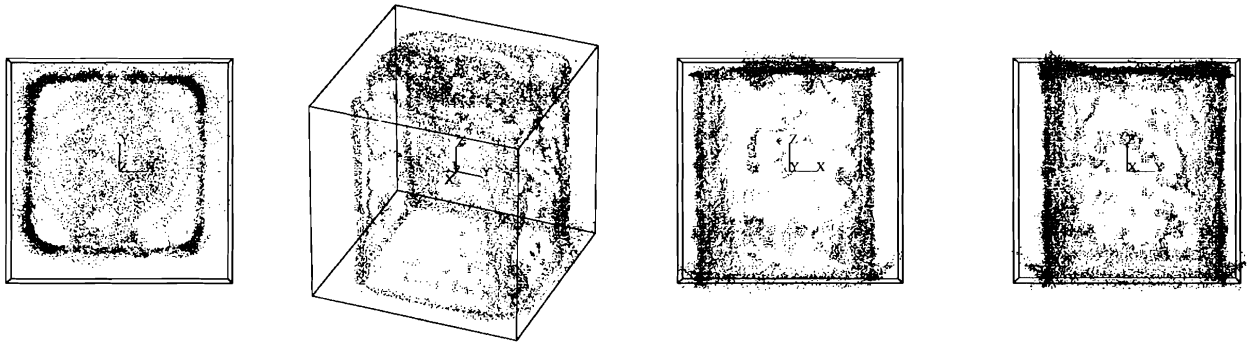
To investigate the feasibility of incremental 3-D surface modeling, we have been studying the construction of such models from image sequences of objects rotating in front of a camera [Sze90a]. In our setup, a single object rotates on a turntable which has been marked so that the current angle of rotation can easily be determined (Figure 4a). Because the camera parameters have already been determined in a pre-calibration phase, we know for each image the exact 3-D transformation relating the turntable (object) coordinates to the camera coordinates. Our problem is thus the standard depth from known motion problem, except that we wish to recover a full 3-D shape description. We call this problem *shape from rotation*, to emphasize that we wish to integrate and represent shape information from a full 360° range of views.

Many different techniques could be used to extract 3-D information from this sequence of images. One of the simplest is to compute a bounding volume for the object by intersecting the volumes formed by the binary object silhouettes and the camera centers [Sze90b]. Other approaches involve tracking curves on the object's silhouette and surface [CB90]. The technique which we describe here uses optic flow measurements to compute a dense and detailed surface model [Sze91]. It is thus similar to incremental iconic depth from motion estimation [MKS89], except that the surface shape is not represented as a 2-D map.

Ideally, we would like to represent our 3-D surface in parametric form. However, constructing such as representation before a rough surface shape is known is not possible. We therefore adopt a different approach, where each optic flow measurement is converted to a 3-D point with an associated 3-D covariance matrix (Figures 4b and 4c). Each of these points represents a small patch of the surface (alternative methods for estimating and tracking such patches can be found in [HCCF88, RW91]). Even though these points are not explicitly connected into a surface (since it may be difficult to reliably segregate points on different surfaces), they form a dense model of shape, unlike pure edge- or feature-based representations.

As successive image pairs are processed, we wish to integrate and merge our 3-D measurements in order to reduce positional errors and to build a full 3-D model. Instead of warping our description to keep it iconic, we keep a list of 3-D points, where each point has both a position and a 3×3 covariance matrix that reflects the confidence in that measurement. This allows us to represent points that are not currently visible, and avoids reducing resolution as the surface slants away from the camera. To avoid an excessive buildup of points and to increase the accuracy of point locations, we merge points from adjacent viewpoints if their projected centers lie within a $\frac{1}{2}$ pixel in the image plane, and if their difference in depth (weighted by their joint uncertainty) is below a threshold [Sze91]. The resulting algorithm incrementally builds a surface description represented as a cloud of points whose accuracy improves over time (Figure 5).

The next step in building our 3-D surface model is to take this collection of 3-D points and to interpolate a surface through them. If we do not know the desired parametric form or a rough shape for the surface, this problem can be quite difficult. To solve this dilemma, we have developed a new 3-D surface interpolation model based on interacting *oriented particles* [ST91b]. These particles, which represent local surface patches, have energy functions which favor the alignment of normals of neighboring particles, thus endowing the surface with an elastic resistance to bending. The particles also have a preferred inter-particle spacing distance, which encourages a uniform sampling density over the surface. We can think of these particle as being a mesh-based (finite element) description of the surface, and the inter-particle energies as discrete approximations to some smoothness metric computed over the surface. To interpolate across gaps in the surface where there are no data points, we add new particles where it is energetically favorable. We can also delete points in areas where the sampling density is too



a b c d

Figure 5: Final merged data from `assam` image sequence: (a) top view (b) oblique view (c) front view (d) side view. The wireframe cube represents the object coordinate system.

high. Once the particles cover the whole object in a uniform manner, we can obtain an explicit surface model by triangulating the point locations.

9. Comparison with feature-based methods

An alternative to directly modeling and estimating surfaces is to compute a visual description based on simpler geometric entities such as points, lines, and planes [Aya91] or 3-D space curves [KWT88]. These simpler primitives have several potential advantages when compared to full surface descriptions. First, because the reduced description completely describes the primitive (e.g, the 5 parameters for a 3-D line) we can obtain better accuracy in the estimation of these parameters. Second, matching features may be less computationally expensive than computing quantities such as optic flow. Third, the parameters describing each feature can be updated independently, which can be much faster than the iterative smoothing required when using correlated fields. Fourth, because the description does not have to be re-sampled, the implementation is much simpler and there is no potential loss of accuracy. Fifth, features such as edges may be more stable under changes of illumination or viewing directions than raw intensities. Sixth, edges or other features may be a sufficient representation for many vision-based tasks such as recognition and positioning [Low85].

On the other hand, feature-based descriptions have a number of limitations and potential disadvantages. It may not always be easy to find features such as lines or curves reliably in images, especially in smoothly varying or in highly textured areas. In particular, methods based on line segments may only work in restricted man-made environments. Features may also shift position depending on the local structure of the image or may disappear and reappear under small changes in viewpoint. The correspondence problem of matching features may sometimes be more difficult or expensive than image-based correlation, especially when the local intensity structure around the feature is ignored. Perhaps the most serious drawback is that if a surface-based description is to be computed from a collection of 3-D geometric primitives, this process may be more difficult and error-prone than estimating the surface from the beginning.

10. Towards robust 3-D surface estimation

How do we determine which of these representations is more suitable for vision applications? While the answer will often be task-dependent, we can consider building a representation which simultaneously models both surfaces and discrete features lying on these surfaces. Features such as edges can be represented as discontinuities in the intensity distribution, and their sub-pixel 3-D location can be explicitly modeled. Intensity edges can serve as potential candidates for depth discontinuities or creases [GP87]. The position of intensity edges can be updated by matching their projection to the output of image-based edge detectors. Edges also locally modify the smoothness constraints on the intensity and shape fields [Ter88].

A possible discrete implementation of such a representation would consist of a collection of 3-D points which define a mesh (triangulation) lying on the surface. Each 3-D point has a position, an intensity, possibly a normal, and a covariance matrix characterizing the uncertainty in these parameters. Each point also has a list of neighbors in the mesh, which may vary over time. Certain points are tagged as edge points, and these are linked together to form 3-D curves. Edge points modify the smoothness in their neighborhood. In the case of depth (shape) discontinuities, edge points are only linked to surface points on the upper side of the discontinuity. Free-floating curves, and even isolated point, are also possible.

Higher level primitives such as line segments or planes can also be added to this representation. A line segment has a list of edge points which belong to it, and its representation is updated by doing a weighted least squares fit to the positions of the individual points. Another way to implement this is to add additional forces coupling the constituent points to the line. A probabilistic formulation is also possible, where the likelihood of a point belonging to a line moderates its interactions with the line (for example, using robust statistics [Hub81]).

This generalized surface representation can be built up incrementally from a sequence of images. As certain areas become more visible, additional mesh points are added to keep the resolution fairly uniform (weak inter-node forces can also be used to keep the spacing uniform). As it becomes evident that surfaces are separate, their meshes can break apart. Free-floating edges can attach themselves to surfaces if they are sufficiently close. Measurements inconsistent with the rest of the model can be thrown out. Of necessity, the behavior of this system is built on a heuristically chosen physically-based model. As the description becomes more stable, however, probabilistically-based updating rules can be used to ensure good convergence for the surface model parameters.

The implementation of a complete modeling system based on this new representation will obviously be quite challenging. However, the potential for accurately modeling both surface shape and sparse geometric features makes this a promising approach to robust and accurate shape recovery.

11. Conclusions

The modeling of visible and 3-D surfaces is an essential component of many computer vision tasks. Developing probabilistic models of such surfaces allows us to obtain robust estimates, to integrate information from different sensors and vision modules, and to accumulate information over time. For visible surfaces (2-½-D depth maps), we can develop probabilistic models using the Gibbs distribution to relate energies (such as smoothness) to probabilities. This in turn allows us to develop sequential estimation algorithms based on the Kalman filter which incrementally build surface descriptions from image sequences. To improve the convergence rate of these algorithm, we have shown how to jointly estimate depth and intensity.

Our ultimate goal is to develop estimation algorithms for full 3-D surface models. Such models are viewpoint independent and allow us to model parts of the visual world that are not currently visible. Building such models is difficult if we do not know the parameterization or topology ahead of time. Our current solution to this problem is to estimate local surface patches, and to later sew these patches into complete surfaces using a mesh-based representation. Our long-term goal is to build complete 3-D surfaces models directly from images, using a representation which models both continuous functions such as shape and intensity, and sparse features such as edges and points, within a single probabilistic framework.

References

[Aya91] N. Ayache. *Artificial Vision for Mobile Robots: Stereo Vision and Multisensory Perception*. MIT Press, Cambridge, Massachusetts, 1991.

[BBM87] R. C. Bolles, H. H. Baker, and D. H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, 1:7–55, 1987.

[BGB79] R. A. Brooks, R. Greiner, and T. O. Binford. The ACRONYM model-based vision system. In *Sixth International Joint Conference on Artificial Intelligence (IJCAI-79)*, pages 105–113, Tokyo, Japan, August 1979.

[BT78] H. G. Barrow and J. M. Tenenbaum. Recovering intrinsic scene characteristics from images. In Allen R. Hanson and Edward M. Riseman, editors, *Computer Vision Systems*, pages 3–26. Academic Press, New York, New York, 1978.

[BZ87] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, Cambridge, Massachusetts, 1987.

[CB90] R. Cipolla and A. Blake. The dynamic analysis of apparent contours. In *Third International Conference on Computer Vision (ICCV'90)*, pages 616–623, Osaka, Japan, December 1990. IEEE Computer Society Press.

[Chr87] J. P. Christ. *Shape Estimation and Object Recognition Using Spatial Probability Distributions*. PhD thesis, Carnegie Mellon University, April 1987.

[EM87] 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.

[Gel74] Arthur Gelb, editor. *Applied Optimal Estimation*. MIT Press, Cambridge, Massachusetts, 1974.

[GG84] 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.

[GP87] E. Gamble and T. Poggio. Visual integration and detection of discontinuities: the key role of intensity edges. A. I. Memo 970, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, October 1987.

[Gri83] W. E. L. Grimson. An implementation of a computational theory of visual surface interpolation. *Computer Vision, Graphics, and Image Processing*, 22:39–69, 1983.

[HCCF88] Y.-P. Hung, D. B. Cooper, and B. Cernushi-Frias. Bayesian estimation of 3-D surfaces from a sequence of images. In *IEEE International Conference on Robotics and Automation*, pages 906–911, Philadelphia, Pennsylvania, April 1988. IEEE Computer Society Press.

[Hub81] P. J. Huber. *Robust Statistics*. John Wiley & Sons, New York, New York, 1981.

[KWT88] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, January 1988.

[LD90] Charles Loop and Tony DeRose. Generalized B-spline surfaces of arbitrary topology. *Computer Graphics (SIGGRAPH'90)*, 24(4):347–356, August 1990.

[Low85] D. G. Lowe. *Perceptual Organization and Visual Recognition*. Kluwer Academic Publishers, Boston, Massachusetts, 1985.

[Mar78] D. Marr. Representing visual information. In Allen R. Hanson and Edward M. Riseman, editors, *Computer Vision Systems*, pages 61–80. Academic Press, New York, New York, 1978.

[MKS89] L. H. Matthies, T. Kanade, and R. Szeliski. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3:209–236, 1989.

[Pen86] A. P. Pentland. Perceptual organization and the representation of natural form. *Artificial Intelligence*, 28(3):293–331, May 1986.

[PTK85] T. Poggio, V. Torre, and C. Koch. Computational vision and regularization theory. *Nature*, 317(6035):314–319, 26 September 1985.

[RW91] J. Rehg and A. Witkin. Visual tracking with deformation models. In *IEEE International Conference on Robotics and Automation*, pages 844–850, Sacramento, California, April 1991. IEEE Computer Society Press.

[ST89a] R. Szeliski and D. Terzopoulos. From splines to fractals. *Computer Graphics (SIGGRAPH'89)*, 23(4):51–60, July 1989.

[ST89b] R. Szeliski and D. Terzopoulos. Parallel multigrid algorithms and computer vision applications. In *Fourth Copper Mountain Conference on Multigrid Methods*, pages 383–398, Copper Mountain, Colorado, April 1989. Society for Industrial and Applied Mathematics.

[ST91a] R. Szeliski and D. Terzopoulos. Physically-based and probabilistic modeling for computer vision. In *SPIE Vol. 1570 Geometric Methods in Computer Vision*, San Diego, July 1991. Society of Photo-Optical Instrumentation Engineers.

[ST91b] R. Szeliski and D. Tonnesen. Particle systems for surface interpolation. Technical report, Digital Equipment Corporation, Cambridge Research Lab, (in preparation) 1991.

[Sze87] R. Szeliski. Regularization uses fractal priors. In *Sixth National Conference on Artificial Intelligence (AAAI-87)*, pages 749–754, Seattle, Washington, July 1987. Morgan Kaufmann Publishers.

[Sze88] R. Szeliski. Estimating motion from sparse range data without correspondence. In *Second International Conference on Computer Vision (ICCV'88)*, pages 207–216, Tampa, Florida, December 1988. IEEE Computer Society Press.

[Sze89] R. Szeliski. *Bayesian Modeling of Uncertainty in Low-Level Vision*. Kluwer Academic Publishers, Boston, Massachusetts, 1989.

[Sze90a] R. Szeliski. Fast surface interpolation using hierarchical basis functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6):513–528, June 1990.

[Sze90b] R. Szeliski. Real-time octree generation from rotating objects. Technical Report 90/12, Digital Equipment Corporation, Cambridge Research Lab, December 1990. For ordering information, please send a message to techreports@crl.dec.com with the word `help` in the Subject line.

[Sze91] R. Szeliski. Shape from rotation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'91)*, pages 625–630, Maui, Hawaii, June 1991. IEEE Computer Society Press.

[Ter86a] D. Terzopoulos. Image analysis using multigrid relaxation methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(2):129–139, March 1986.

[Ter86b] D. Terzopoulos. Regularization of inverse visual problems involving discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(4):413–424, July 1986.

[Ter88] D. Terzopoulos. The computation of visible-surface representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-10(4):417–438, July 1988.

[TWK87] D. Terzopoulos, A. Witkin, and M. Kass. Symmetry-seeking models and 3D object reconstruction. *International Journal of Computer Vision*, 1(3):211–221, October 1987.

[VTL89] B. C. Vemuri, D. Terzopoulos, and P. J. Lewicki. Canonical parameters for invariant surface representation. In *SPIE, Advances in Intelligent Robotics Systems*, Philadelphia, Pennsylvania, November 1989. Society of Photo-Optical Instrumentation Engineers.