Bayesian Super-Resolved Surface Reconstruction from Images

Conference Paper *in* Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition · February 2000

DOI: 10.1109/CVPR.2000.855843 · Source: IEEE Xplore

V. N. Smelyanskiy
NASA
153 PUBLICATIONS **3,025** CITATIONS
SEE PROFILE

Peter C Cheeseman
84 PUBLICATIONS **8,444** CITATIONS
SEE PROFILE

Robin D. Morris
University of California, Santa Cruz
61 PUBLICATIONS **1,029** CITATIONS
SEE PROFILE

BAYESIAN SUPER-RESOLVED SURFACE RECONSTRUCTION FROM IMAGES

V. N. Smelyanskiy, P. Cheeseman, D. A. Maluf and R. D. Morris

NASA Ames Research Center,
MS 269-2, Moffett Field, CA 94035, USA
[vadim,cheesem,maluf,rdm]@ptolemy.arc.nasa.gov

Abstract

Bayesian inference has been used successfully for many problems where the aim is to infer *the parameters of a model of interest. In this paper we formulate the three dimensional reconstruction problem as the problem of inferring the parameters of a surface model from image data, and show how Bayesian methods can be used to estimate the parameters of this model given the image data. Thus we recover the three dimensional description of the scene. This approach also gives great flexibility. We can specify the geometrical properties of the model to suit our purpose, and can also use different models for how the surface reflects the light incident upon it. In common with other Bayesian inference problems, the estimation methodology requires that we can simulate the data that would have been recorded for any values of the model parameters. In this application this means that if we have image data we must be able to* render *the surface model. However it also means that we can infer the parameters of a model whose resolution can be chosen irrespective of the resolution of the images, and may be super-resolved. We present results of the inference of surface models from simulated aerial photographs for the case of super-resolution, where many surface elements project into a single pixel in the low-resolution images.*

1 Introduction

Bayesian inference has proved to be the method of choice for many inference problems, enabling accurate estimation of parameters of interest from noisy and incomplete data, and also providing estimates of the errors associated with the inferred parameters. The general approach is illustrated in figure 1. The figure shows that synthetic observations of the model are made using a computer simulation of the observation process, and these are compared with the actual observations. The error between the actual and the simulated observations is used to adjust the parameters of the model, to minimize the errors. The estimation of the errors on the parameter estimates (or more generally, the estimation of the covariance matrix of the parameters) means that the parameter estimates can be updated when more observations become available.

In reconstruction of three dimensional surfaces, the model is a parameterized surface model, and the measurement system is a computer model of the image formation process.

The surface model is chosen to suit the desired application. For example it could be a CAD model if the data is images of machine parts, or a spline model if the data came from more general objects. In our application, we are interested in recovering planetary surfaces, and hence we choose to describe the surface by a triangular mesh. This is a standard model in computer graphics, and it allows the density of the mesh to vary spatially.

The model of the image formation process is the area of computer graphics known as rendering. A model of how light is reflected from the surface is used, together with an abstraction of how the image is formed, to synthesize the image that would have been recorded from the current surface model under the lighting conditions and camera position and orientation. The parameters describing surface reflectance properties are part of the surface model that are to be inferred together with the geometrical parameters. The theoretical exposition in this paper is valid for any parameterized reflection function. The results assume Lambertian reflection.

The formation of a synthetic image from a parameterized surface model, lighting and camera parameters is discussed in more detail in section 3. Here we just note that current rendering technology is unsuitable for our purpose because it operates in *image space*, and so the image formed has artifacts due to the rel-

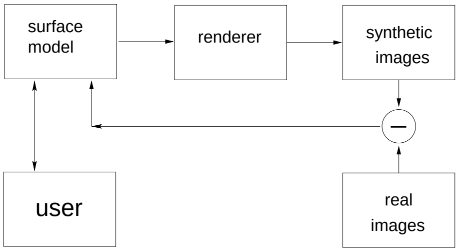

Figure 1: An outline of the Bayesian approach to surface reconstruction

ative sizes of the projection of the surface elements onto the image plane and their discretization into pixels. These artifacts are particularly noticeable along the edges of surface elements (aliasing). Also, if the projections of the surface elements in the pixel plane are very small, such as when we are trying to infer a high resolution mesh from many low resolution images, these artifacts dominate the 'standard' rendering process. The renderer that is required for our purposes is one that operates in *object space*, and in section 3 we briefly describe such a system. An object-space renderer can also compute the *derivatives* of the pixel values with respect to the surface model parameters. This is a crucial part of the inference process, and is described in detail in section 3.1.

One of the major goals of this paper is to develop the Bayesian inference approach for 3D super-resolved surface reconstruction when the resolution of inferred surface mesh is higher than the spatial resolution of input images. This also enables image super-resolution: synthetically produced images of the super-resolved surface model can be at higher resolution than the input data images.

1.1 Previous Work

Most previous work in the area of the estimation of three dimensional surfaces from image data has used, shape from shading, shape from motion and shape from stereo (more generally, 'shape from X').

In shape from shading [3], image gradients are related to surface derivatives under the assumption of orthographic projection. Using assumed boundary conditions, the surface derivatives can be integrated to produce an estimate of the surface heights (more strictly, the distance from the camera to the points on the surface).

Shape from stereo [8] uses correspondence matches between features in the images to give the disparity between these features. Finding these correspondence matches is aided by using the epipolar constraint [8]. If the camera geometry is known then the disparity can be directly related to the distance of the feature from the camera, and the feature can be located in space. The discrete points corresponding to the features matched in the images are then joined to form a representation of the surface.

A drawback of shape from stereo is that the density of points in the recovered surface is unknown a-priori, and is dependent on the number and density of features found in the images. A feature detector giving few features gives a very coarse surface representation; one that gives very many is likely to produce features that can be mismatched between images.

Shape from shading has the drawback that the spatial density of the recovered surface is fixed at the image density. It is also difficult to apply if the reflectance properties are spatially varying. Both of these approaches have difficulty incorporating new observations of the surface that become available after the initial estimate is made.

A Bayesian approach was used for image super-resolution in the series of papers beginning with [6]. In the earlier work in [7], input images were taken from roughly the same direction under the similar lighting conditions. In that work the surface model is essentially represented as 2D texture map. In our case of full 3D surface reconstruction this restriction is lifted: low resolution input images as well as high-resolution output images can correspond to very different values of registration parameters.

2 A Bayesian Framework

In this paper the surface geometry is represented by a triangular mesh and the surface reflectance properties (albedos) are associated with the vertices of the triangular mesh. We will consider the case of Lambertian surfaces. We will also assume known the camera parameters and the parameters of the lighting. The estimation of these parameters will be considered in a forthcoming paper using the same Bayesian framework.

Thus we represent the surface model by the pair of vectors $[\vec{z}\ \vec{\rho}]$. The components of these vectors correspond to the height and albedo values defined on a regular grid of points

$$[\vec{z}\ \vec{\rho}] = \{(z_{\mathbf{i}}, \rho_{\mathbf{i}}),\quad \mathbf{i} = \ell\ (q\,\hat{\mathbf{x}} + p\,\hat{\mathbf{y}})\}\quad q,p = 0,1,\ldots \tag{1}$$

where ℓ is the elementary grid length, $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$ are an orthonormal pair of unit vectors in the (x,y) plane and $\mathbf{i}$ indexes the position in the grid. The pair of vectors of heights and albedos represents a full vector for the surface model

$$\mathbf{u} = [\vec{z}\ \vec{\rho}]. \tag{2}$$

To estimate the values of $\vec{z}, \vec{\rho}$ from image data, we apply Bayes theorem which gives

$$p(\vec{z}, \vec{\rho}\,|I_1 \ldots I_F) \propto p(I_1 \ldots I_F|\vec{z}, \vec{\rho}\,)\, p(\vec{z}, \vec{\rho}\,),$$

where I_f $(f = 1, \ldots, F)$ is the image data. This states that the posterior distribution of the heights and the albedos is proportional to the likelihood – the probability of observing the data given the heights and albedos – multiplied by the prior distribution on the heights and albedos.

The prior distribution is assumed to be Gaussian

$$p(\vec{z}, \vec{\rho}\,) \propto \exp\left(-\frac{1}{2}\mathbf{u}\Sigma^{-1}\,\mathbf{u}^T\right), \qquad (3)$$

$$\Sigma^{-1} = \begin{bmatrix} \hat{Q}/\sigma_h^2 & 0 \\ 0 & \hat{Q}/\sigma_\rho^2 \end{bmatrix},$$

where the vector of the surface model parameters $\mathbf{u}$ is defined in (2). The inverse covariance matrix is constructed to enforce a smoothing constraint on local variations of heights and albedos. We penalize the integral over the surface of the curvature factor $c(x, y) = z_{xx}^2 + z_{yy}^2 + 2z_{xy}^2$, and similarly for albedos. We approximate the partial derivatives in $c(x, y)$ using finite differences of the height (albedo) values. Then coefficients of $\hat{Q}$ form a 5×5 template $\hat{\Gamma}$ and result from summing $c(x, y)$ over the surface

$$\hat{Q}_{n,m}^{q+n,p+m} = \Gamma_{q,p}, \quad q, p = -2, \ldots, 2. \qquad (4)$$

For brevity we do not provide here explicit values for the coefficients $\Gamma_{q,p}$. The two hyperparameters σ_h and σ_ρ in equation (3) control the expected values of the surface-averaged curvatures for heights and albedos.

This prior is placed directly over the height variables, z, but albedos are only defined over the range $[0-1]$. To avoid this, we use transformed albedos $\rho'_{\mathbf{i}}$ in the Gaussian (3), where $\rho'_{\mathbf{i}}$ are defined by:

$$\rho'_{\mathbf{i}} = \log(\rho_{\mathbf{i}}/(1-\rho_{\mathbf{i}})), \quad \mathbf{u} \rightarrow [\vec{z}\ \vec{\rho'}\,]. \qquad (5)$$

In the vector of model parameters $\mathbf{u}$ values of $\vec{\rho}$ are replaced by values of $\vec{\rho'}$.

For the likelihood we make the usual assumption that the differences between the observed data and the data synthesized from the model have a zero mean, Gaussian distribution, and also assume that the images I_f comprising the data are conditionally independent. This gives

$$p(I_1 \ldots I_F|\vec{z}, \vec{\rho}\,) \propto \exp\left(-\frac{\sum_{f,p}(I_{f\,p} - \hat{I}_{f\,p}(\vec{z}, \vec{\rho}\,))^2}{2\sigma_e^2}\right)$$

where $\hat{I}_{f\,p}(\vec{z}, \vec{\rho}\,)$ denotes the pixel intensities in the image f synthesized from the model, σ_e^2 is the noise variance and the summation is over the pixels (p) and over all images (f) used for the inference.

Consider the negative log-posterior.

$$L(\vec{z}, \vec{\rho}\,) \propto \frac{\sum_{f,p}(I_{f\,p} - \hat{I}_{f\,p}(\vec{z}, \vec{\rho}\,))^2}{\sigma_e^2} + \mathbf{x}\,\Sigma^{-1}\,\mathbf{x}^T, \quad (6)$$

where $\mathbf{x} = \mathbf{u} - \mathbf{u}_0$ is a deviation from a current estimate $\mathbf{u}_0$. L is a nonlinear function of $\vec{z}, \vec{\rho}$ and the MAP estimate is that value of $\vec{z}, \vec{\rho}$ which minimizes $L(\vec{z}, \vec{\rho}\,)$.

In the case of images with no shadows or visible occlusions which we consider here, the log-posterior is in general unimodal and gradient methods can be applied for minimizing $L(\vec{z}, \vec{\rho}\,)$. We linearize $\hat{I}(\vec{z}, \vec{\rho}\,)$ about the current estimate, $\vec{z}_0, \vec{\rho}_0$

$$\hat{I}(\vec{z}, \vec{\rho}\,) = \hat{I}(\vec{z}_0, \vec{\rho}_0\,) + \mathbf{D}\,\mathbf{x}, \quad \mathbf{D} \equiv \left\{\frac{\partial \hat{I}_{f\,p}}{\partial z_{\mathbf{i}}}, \frac{\partial \hat{I}_{f\,p}}{\partial \rho'_{\mathbf{i}}}\right\} \quad (7)$$

where $\mathbf{D}$ is the matrix of derivatives evaluated at z_0, ρ_0. Then the minimization of $L(\vec{z}, \vec{\rho}\,)$ is replaced by minimization of the quadratic form:

$$L' = \frac{1}{2}\mathbf{x}\,\hat{A}\,\mathbf{x} - \mathbf{b}\,\mathbf{x}, \quad \mathbf{x} \equiv \mathbf{u} - \mathbf{u_0},$$

$$\hat{A} = \Sigma^{-1} + \frac{\mathbf{D}\mathbf{D}^T}{\sigma_e^2}, \qquad (8)$$

$$\mathbf{b} = \frac{(I - \hat{I}(\vec{z}_0, \vec{\rho}_0\,))}{\sigma_e^2}\mathbf{D}.$$

Here $\hat{A}$ is the Hessian matrix of the quadratic form and vector $\mathbf{b}$ is the gradient of the likelihood L computed at the current estimate. We search for the minimum in $\mathbf{x}$ using a conjugate-gradient method. At the minimum we update the current estimate, $\mathbf{u}_1 = \mathbf{u}_0 + \mathbf{x}$, recompute $\hat{I}$ and $\mathbf{D}$, and repeat the minimization procedure iteratively until the current estimate $\mathbf{u}_k$ approaches the minimum of $L(\vec{z}, \vec{\rho}\,)$.

Thus to find the MAP estimate requires that we can render the image and compute the derivatives for any values of the surface model parameters. We discuss this computation in some detail in the next sections. Here it is sufficient to note that while forming $\hat{I}$ using only object space computation (see section 3) is computationally expensive, we can compute $\mathbf{D}$ at the same time for little additional computation. Also the derivative matrix is sparse with the number of nonzero entries a few times the number of model parameters. This makes the process described above a practical one. Convergence is also accelerated by using a multigrid approach.

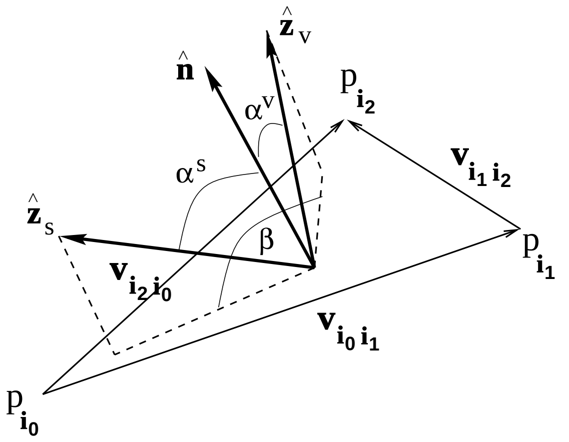

Figure 2: Geometry of the triangular facet, illumination direction and viewing direction. $\hat{\mathbf{z}}_s$ is the vector to the illumination source; $\hat{\mathbf{z}}_v$ is the viewing direction.

At convergence we compute a new inverse covariance matrix, $(\Sigma^{-1})' = \Sigma^{-1} + \mathbf{D}\mathbf{D}^T/\sigma_e^2$. This is then used as a *prior* inverse covariance matrix when new image data of the same surface is obtained, enabling a recursive update and integration of data recorded at different times. The posterior inverse covariance matrix gives information about the uncertainty of the estimated surface.

3 Formation of the image and the derivative matrix.

The task of forming an image, $\hat{I}$, given a surface description, $\vec{z}, \vec{\rho}$, and camera and illumination parameters is the area of computer graphics known as rendering [2]. Most current rendering technology is focused on producing images which are *visually* appealing, and producing them very quickly. As discussed in the introduction, this results in the use of image-space algorithms, with the fundamental assumption that each triangle making up the surface, when projected onto the image plane, is much larger than a pixel. This makes reasonable the assumption that any given pixel receives light from only one triangle, but does produce images with artifacts at the triangle edges. Standard rendering also produces inaccurate images if the triangles project into areas much smaller than a pixel on the image plane, as the pixel will then be colored with a value coming from just *one* of the triangles.

Clearly this approach is not suitable for high-resolution 3D surface reconstruction from multiple images. The triangles in a high-resolution surface may project onto an area much smaller than a single pixel in the image plane (sub-pixel resolution). Therefore, as discussed in the introduction, for our system we implemented a renderer for triangular meshes which performs all computation in *object space.* At present we neglect the blurring effect due to diffraction and due to the role of pixel boundaries in the CCD array. Then the light from a triangle as it is projected into a pixel contributes to the brightness of the pixel with a weight factor proportional to the fraction of the area of the triangle which projects into that pixel. This produces anti-aliased images and allows an image of any resolution to be produced from a mesh of arbitrary density, as required when the system performing the surface inference may have no control over the image data gathering.

Our renderer computes brightness $\hat{I}_p$ of a pixel p in the image as a sum of contributions from individual surface triangles $\triangle$ whose projections into the image plane overlap, at least partially, with the pixel p.

$$\hat{I}_p = \sum_{\triangle} f^p_{\triangle}\, \Phi_{\triangle}. \tag{9}$$

Here $\Phi_{\triangle}$ is a radiation flux reflected from the triangular facet $\triangle$ and received by the camera, and $f^p_{\triangle}$ is the fraction of the flux that falls onto a given pixel p in the image plane. In the case of Lambertian surfaces and a single spectral band $\Phi_{\triangle}$ is given by the expression

$$\begin{aligned} \Phi_{\triangle} &= \rho\, E(\alpha^s)\, \cos\alpha^v\, \cos^{\kappa}\theta\, \Delta\Omega, \qquad (10)\\ E(\alpha^s) &= \rho\, \mathcal{A}\left(\mathcal{I}^s \cos\alpha^s + \mathcal{I}^a\right).\\ \Delta\Omega &= S/d^2. \end{aligned}$$

Here ρ is an average albedo of the triangular facet. Orientation angles α^s and α^v are defined in figure 2. $E(\alpha^s)$ is the total radiation flux incident on the triangular facet with area $\mathcal{A}$. This flux is modeled as a sum of two terms. The first term corresponds to direct radiation with intensity $\mathcal{I}^s$ from the light source at infinity (commonly the sun). The second term corresponds to ambient light with intensity $\mathcal{I}^a$. The parameter θ in equation. (10) is the angle between the camera axis and the viewing direction (the vector from the surface to the camera); κ is the lens falloff factor. $\Delta\Omega$ in (10) is the spatial angle subtended by the camera which is determined by the area of the lens S and the distance d from the centroid of the triangular facet to the camera.

We identify the triangular facet $\triangle$ by the set of 3 indices $(\mathbf{i_0}, \mathbf{i_1}, \mathbf{i_2})$ from the vector of heights (1) that determines the vertices of the triangle in a counterclockwise direction (see figure 2). In the r.h.s of equation (10) we have omitted for brevity those indices from all the quantities associated with individual triangles. The average value of albedo for the triangle in (10) is computed based on the components of the albedo vector ρ corresponding to the triangle indices

$$\rho_\triangle \equiv \rho_{\mathbf{i_0},\mathbf{i_1},\mathbf{i_2}} = \frac{1}{3}(\rho_{\mathbf{i_0}} + \rho_{\mathbf{i_1}} + \rho_{\mathbf{i_2}}). \tag{11}$$

We note that using average albedo (11) in the expression for $\Phi_\triangle$ is an approximation which is justified when the albedo values vary smoothly between the neighboring vertices of a grid.

The area $\mathcal{A}$ of the triangle and the orientation angles in (10) can be calculated in terms of the vertices of the triangle $\mathbf{P_i}$ (see figure 2) as follows:

$$\begin{aligned} &\hat{\mathbf{n}} \cdot \hat{\mathbf{z}}^s = \cos\alpha^s, \quad \hat{\mathbf{n}} \cdot \hat{\mathbf{z}}^v = \cos\alpha^v, \\ &\hat{\mathbf{n}} = \frac{\mathbf{v}_{\mathbf{i_0},\mathbf{i_1}} \times \mathbf{v}_{\mathbf{i_1},\mathbf{i_2}}}{2\mathcal{A}}, \quad \mathbf{v}_{\mathbf{i},\mathbf{j}} = \mathbf{P_j} - \mathbf{P_i} \end{aligned} \tag{12}$$

Here $\hat{\mathbf{n}}$ is a unit normal to the triangular facet and vectors of the edges of the triangle $\mathbf{v}_{\mathbf{i},\mathbf{j}}$ are shown in figure 2.

We use a standard pinhole camera model with no distortion in which coordinates of a 3D world point $\mathbf{P} = (x, y, z)$ are first rotated with the rotation matrix $\hat{\mathbf{R}}$ and then translated by the vector $\mathbf{T}$ into camera coordinates, yielding $\mathbf{P_c} = (x_c, y_c, z_c)$

$$\mathbf{P_c} = \hat{\mathbf{R}}\,\mathbf{P} + \mathbf{T} \tag{13}$$

($\hat{\mathbf{R}}$ and $\mathbf{T}$ are expressed in terms of the camera registration parameters [2]. We do not give them explicitly here). After the 3D transformation given in (13), point $\mathbf{P_c}$ in the camera coordinate system is transformed using a perspective projection into the 2D image point $\bar{\mathbf{P}} = (\bar{x}, \bar{y})$ using a focal length f and aspect ratio a.

$$\begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix} = -\frac{f}{z_c} \begin{bmatrix} a\,x_c \\ y_c \end{bmatrix}. \tag{14}$$

We use 2D image projections of the triangular vertices $\mathbf{P_i}$ to compute the area fraction factors $f^p_\triangle$ for the surface triangles (cf. Eq. (9))

$$f^p_\triangle = \frac{\bar{A}_{\text{polygon}}}{\bar{A}_\triangle}. \tag{15}$$

Here $\bar{A}_\triangle$ is the area of the projected triangle on the image plane and $\bar{A}_{\text{polygon}}$ is the area of the polygon resulting from the intersection of the projected triangle and boundary of the pixel p (see figure 3).

3.1 Computation of the derivative matrix.

The inference of the surface model parameters depends on the ability to compute the derivatives of the modeled observations $\hat{I}$ with respect to the model parameters. According to equation (9), the intensity $\hat{I}_p$ of a pixel p depends on the subset of the surface parameters, (heights and albedos), that are associated with the triangles whose projections overlap the pixel area.

The derivatives $\hat{I}_p$ with respect to logarithmically transformed albedo values are easily derived from equations (5), (9) and (10).

In our object-space renderer, which is based on pixel-triangle geometrical intersection in the image plane, the pixel intensity derivatives with respect to the surface heights have two distinct contributions

$$\frac{\partial \hat{I}_p}{\partial z_{\mathbf{i}}} = \sum_\triangle \left(f^p_\triangle \,\frac{\partial \Phi_\triangle}{\partial z_{\mathbf{i}}} + \Phi_\triangle \frac{\partial f^p_\triangle}{\partial z_{\mathbf{i}}} \right) \tag{16}$$

Variation of the surface height $z_{\mathbf{i}}$ gives rise to variations in the normals of the triangles associated with this height (in a general triangular mesh, on average 6 triangles are associated with each height) and this produces the derivatives of the total radiation fluxe $\Phi_\triangle$ to the camera from those triangles. This is the first term in equation (16). Also, height variation gives rise to the displacement of the corresponding point which is the projection of this vertex on the image plane. This results in changes to the areas of the triangles and polygons with edges containing this point (see figure 3). This produces the derivatives of the fractions $f^p_\triangle$, the second term in equation 16.

When the triangle is completely inside the pixel its area fraction $f^p_\triangle = 1$ and according to (16) its contribution to the pixel intensity derivative equals the derivative of the corresponding radiation flux, $\partial\Phi/\partial z_{\mathbf{i}}$. The flux derivative can be computed directly from the coordinates of the triangle vertices and the camera position using Eqs. (10) and (12). For the surface triangle with vertices $(\mathbf{P_{i_0}}, \mathbf{P_{i_1}}, \mathbf{P_{i_2}})$ the flux derivative with respect to the z component of the vertex $\mathbf{P_{i_0}}$ equals

$$\frac{\partial \Phi}{\partial \mathbf{z_{i_0}}} = \frac{1}{2}\rho(\mathbf{P_{i_2}} - \mathbf{P_{i_1}}) \times \hat{\mathbf{z}} \cdot \mathbf{g}\,\frac{S}{d^2}, \tag{17}$$

where

$$\mathbf{g} = \mathcal{I}_s(\hat{\mathbf{z}}_v \cos\alpha_s + \hat{\mathbf{z}}_s \cos\alpha_v - \hat{\mathbf{n}} \cos\alpha_s \cos\alpha_v) + \mathcal{I}_a \hat{\mathbf{z}}_v$$

and $\hat{\mathbf{z}}$ is a unit normal in the vertical direction.

When the triangle is projected into more than one pixel then the height derivatives of the projected area fraction in (16) have to be computed for every pixel intersecting with the triangle. This can be done using the following chain rule arguments.

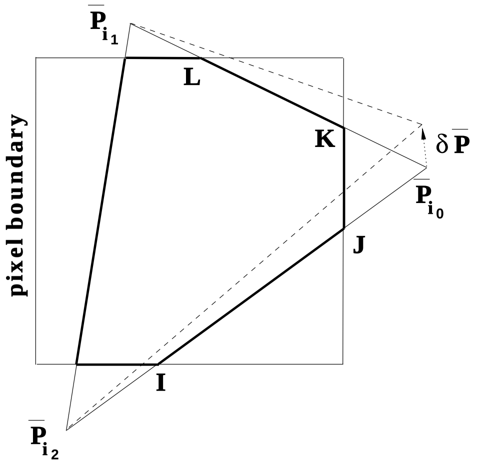
pixel boundary
$\bar{\mathbf{P}}_{\mathbf{i_1}}$
$\bar{\mathbf{P}}_{\mathbf{i_0}}$
$\bar{\mathbf{P}}_{\mathbf{i_2}}$
$\delta\bar{\mathbf{P}}$
L
K
J
I

Figure 3: The intersection of the projection of a triangular surface element $(\mathbf{i_0}, \mathbf{i_1}, \mathbf{i_2})$ onto the pixel plane with the pixel boundaries. Bold lines corresponds to the edges of the polygon resulting from the intersection. Dashed lines correspond to the new positions of the triangle edges when point $\mathbf{P}_{\mathbf{i_0}}$ is displaced by $\delta\mathbf{P}$

As mentioned above, when the z component of the vertex $\mathbf{P}_{\mathbf{i_0}}$ in the 3D world coordinate system varies by δz, the corresponding 2D point $\bar{\mathbf{P}}_{\mathbf{i_0}}$ in the image projection plane is displaced by $\delta\,\bar{\mathbf{P}}$ as shown in figure 3. The corresponding point displacement derivative equals:

$$\frac{\partial \bar{\mathbf{P}}}{\partial z} = -\frac{1}{z_c}\left[\begin{array}{c} a\,f\,R_{13} + \bar{x}\,R_{33} \\ f\,R_{23} + \bar{y}\,R_{33} \end{array}\right]. \qquad (18)$$

Here we have dropped the vertex index; R_{ij} are the components of the rotation matrix $\hat{\mathbf{R}}$ in (13) and z_c is the z component of the point $\mathbf{P}$ in the sensor coordinate system, given by equation (13).

Displacement $\delta\,\bar{\mathbf{P}}$ of the triangle vertex $\bar{\mathbf{P}}$ gives rise to the change $\delta\bar{A}_{\triangle}$ in the area of the projected triangle and also the change $\delta\bar{A}_{\text{polygon}}$ in the polygon area. It then follows from equation(15) that

$$\frac{\partial f^p_{\triangle}}{\partial z_{\mathbf{i_0}}} = \frac{1}{\bar{A}_{\triangle}}\left(\frac{\partial \bar{A}_{\text{polygon}}}{\partial \bar{\mathbf{P}}_{\mathbf{i_0}}} - f^p_{\triangle}\,\frac{\partial \bar{A}_{\triangle}}{\partial \bar{\mathbf{P}}_{\mathbf{i_0}}}\right)\frac{\partial \bar{\mathbf{P}}_{\mathbf{i_0}}}{\partial z_{\mathbf{i_0}}}. \qquad (19)$$

Here the point displacement derivative $\partial\bar{\mathbf{P}}_{\mathbf{i_0}}/\partial z_{\mathbf{i_0}}$ is given in (18).

Thus, the task of computing the derivative of the area fraction (19) is reduced to the computation of $\partial\bar{A}_{\triangle}/\partial\bar{\mathbf{P}}_{\mathbf{i_0}}$ and $\partial\bar{A}_{\text{polygon}}/\partial\bar{\mathbf{P}}_{\mathbf{i_0}}$. Note that the intersection of a triangle and a pixel for a rectangular pixel boundary can, in general, be a polygon with 3, 4, 5 or 6 edges with various possible forms. However the algorithm for computing the polygon area derivatives that we have developed is general, and does not depend on a particular polygon configuration. Consider, as an example, the polygon shown in figure 3 which is a part of the projected surface triangle with indices $\mathbf{i_0}, \mathbf{i_1}, \mathbf{i_2}$. We are interested in the derivative of the polygon area with respect to the point $\bar{\mathbf{P}}_{\mathbf{i_0}}$ that connects two edges of the projected triangle, $(\mathbf{P}_{\mathbf{i_2}}, \mathbf{P}_{\mathbf{i_0}})$ and $(\mathbf{P}_{\mathbf{i_0}}, \mathbf{P}_{\mathbf{i_1}})$. These triangular edges contain segments ($\mathbf{I}$, $\mathbf{J}$) and ($\mathbf{K}$, $\mathbf{L}$) that are sides of the corresponding polygon. It can be seen from figure 3 that when the point $\bar{\mathbf{P}}_{\mathbf{i_0}}$ is displaced by $\delta\bar{\mathbf{P}}_{\mathbf{i_0}}$ the change in the polygon area is given by the sum of two areas, $\delta A_{\mathbf{I,J}}$ and $\delta A_{\mathbf{K,L}}$, spanned by the two corresponding segments and taken with appropriate signs. Therefore the polygon area derivative with respect to the triangle vertex $\bar{\mathbf{P}}_{\mathbf{i_0}}$ is represented as a sum of the two "segment area" derivatives for the 2 segments adjacent to a given vertex. Using straightforward geometrical arguments one can calculate the areas $\delta A_{\mathbf{I,J}}$ and $\delta A_{\mathbf{K,L}}$ to first order in the displacement $\delta\bar{\mathbf{P}}_{\mathbf{i_0}}$. Then the polygon area derivative can be expressed directly in terms of the triangle vertices and the endpoints of the polygon segments: $\mathbf{I}$, $\mathbf{J}$, $\mathbf{K}$ and $\mathbf{L}$ (cf. figure 3). The details of the polygon area derivative computation will be presented elsewhere.

4 Adaptiveness and Super-resolution aspects

The correct choice of the smoothness prior (3) is necessary for inferring surfaces that have regions with high curvature (edges). It is also of special importance in the case of surface super-resolution where the spatial resolution of individual pixels is greater than the size of a surface triangle.

Clearly the values of the hyperparameters σ_h and σ_ρ in the smoothness prior (3) should be controlled by the relative sizes of surface triangles ℓ and spatial resolution of individual pixels $\Re$. One needs at least 2 images to infer heights and albedos at the same time. In this case σ_h and σ_ρ should be chosen in such a way to ensure the smoothness of a surface patches on a scale $\sim \Re \gg \ell$. For larger number of low-resolution images the inferred surface may have a smoothness scale smaller than $\Re$. This corresponds to a smaller values of σ_h and σ_ρ.

We note however that the smoothness prior remains very important even in the "over-constrained" case where the number of low-resolution images is relatively large and the total number of pixels in all images per height or albedo value is $\gtrsim 1$. The reason is that the spatial structure of the derivative matrix $\mathbf{D}$ can be very irregular in this regime and the smoothness prior essentially plays a role of *regularizer.*

Indeed, one can show based on the analysis from the previous section that the magnitude of the derivatives $|\partial \hat{I}_p / \partial z_{\mathbf{i}}|$ can be much larger for the surface vertices whose triangles intersect the boundary of the pixel p than for the vertices that are projected fully inside of the pixel p along with all triangles surrounding them. When the number of triangles per pixel increases (typically $> 10 - 15$) this can give rise to a strong spatial modulation of the derivative values that reflect the positions of the pixel boundaries.

We note that *misalignment* of the pixel boundaries from different data images of the same area is crucial for the surface reconstruction [6]. This gives rise to a very irregular spatial pattern of the coefficients of $\mathbf{DD}^T$ and $\mathbf{b}$ in equation (8) that results from summation over all input data images.

To achieve the regularization we adjust Σ^{-1} in equation (8) to be on the same scale as the matrix elements of $\mathbf{DD}^T/\sigma_e^2$. Scale of $\mathbf{DD}^T$ depends on a grid size ℓ. For height derivatives $\mathbf{DD}^T \sim \ell^2$ and for albedo derivatives $\mathbf{DD}^T \sim \ell^4$. This sets the scale of the hyperparameters σ_h and σ_ρ for different levels of the multigrid. We compute the values of σ_h and σ_ρ based on the average diagonal values of $\mathbf{DD}^T$. In particular, for the heights

$$\sigma_e^2/\sigma_h^2 \approx \text{trace}\left\{\mathbf{DD}^T\right\}_h/(N\,\Gamma_{0,0}), \qquad (20)$$

where the trace of $\mathbf{DD}^T$ is taken with respect to height and N is the number of heights. The value of σ_ρ is readjusted in a similar way. $\Gamma_{0,0}$ is a coefficient from the smoothness prior (see equation 4).

Finally, to achieve local adaptiveness of the prior we place the curvature penalty in equations (6), (8) on the **deviation** from the current surface estimate, $\mathbf{u} - \mathbf{u}_k$, not on the estimate itself.

5 Results

As a test example we used a triangulated surface whos heights correspond to the digital elevation model (DEM) of the Duckwater region (Nevada). We prepared the surface albedos synthetically to fit an existing Landsat image data of the same region. The surface is of dimension 297 × 297 heights and the same number of albedos. Sixteen low-resolution images of the surface were produced using the simple perspective sensor model (14), with differing lighting and camera orientations. Each image is 128×128 pixels. We used these synthetic images as the input data images I. Figure 4 (left) shows a portion of one of the input images of size (96×74) pixels.

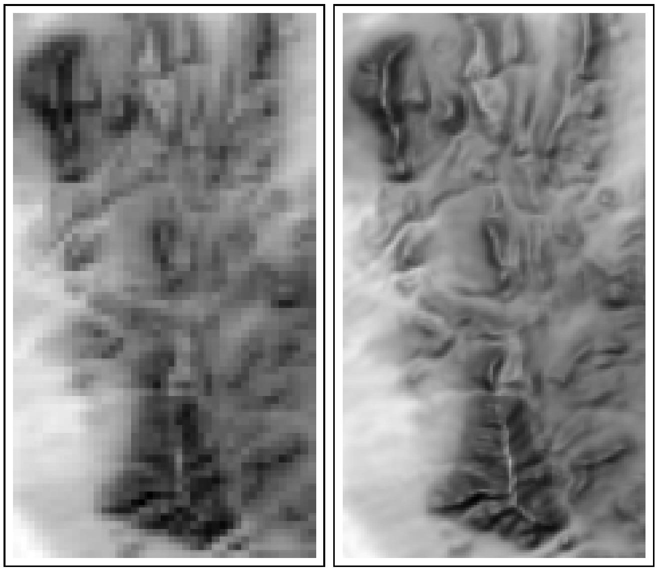

Figure 4: Low-resolution image, 96×74 pixels (left). High-resolution image, 960×740 pixels (right).

Starting from a mesh with all zero heights and all albedos set to 0.5, the conjugate gradient scheme described above was used to infer the surface shown in figure 5. The surface is of the same dimension as the original surface. Not that this is a dense triangulation – when projected into the pixel grid of figure 4 many triangles fall into one pixel. Thus we infer a super-resolved surface – a pixel lying on a mountain ridge does not imply a planar region in the inferred surface, rather, we infer a surface where highly curved regions may project into a single pixel. Figure 4 (right) shows a high-resolution image synthesized from the same region of the inferred surface as that corresponding to original data image at the left. This image was rendered at 10 times the resolution of the original data image. Comparison of the left and right images in figure 4 highlights the super-resolution aspect of our approach, but it should be emphasized that the output is the surface model in figure 5.

Because we know the original surface, the error maps can be computed for both heights and albedos (figure 6) to judge the quality of inference. Note the vertical scales compared with figure 5. The reconstruction is accurate, with most errors being in the regions of high curvature.

6 Conclusions and future extensions

We have developed a very general framework for the inference of general surface geometry and reflectance models from image data, where the model choice is determined by the physical properties of the surface we

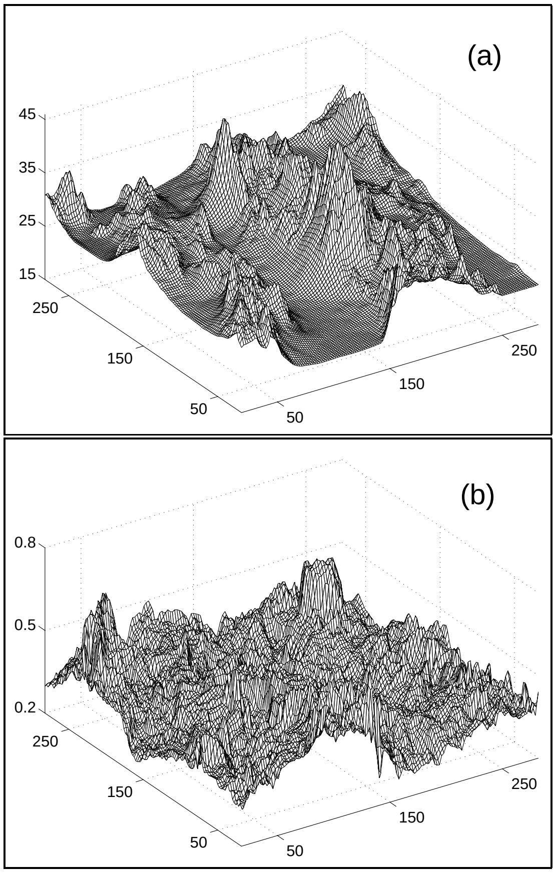

Figure 5: Inferred surface: (a) heights, (b) albedos

wish to infer. We have demonstrated that for the case of a triangulated surface and Lambertian reflectance the parameters of a surface model, namely the heights and albedos, can be inferred from a set of image data. We have developed a framework that allows easy inclusion of future data observed from the same surface, and easy incorporation of data from other sensing modalities.

In this paper we have assumed the registration parameters of input data images to be know in advance. In principle one can use the Bayesian approach developed above to infer the registration parameters of the data images along with surface heights and albedos. Such inference will include as an essential element the derivatives of the intensities of synthetic images with respect to registration parameters.

Another limitation of the current work is related to the absence of shadows and visible occlusions in input images. Future developments will include the ability to compute correctly both the image and its derivatives when this limitation is lifted. We note that the derivatives in the presence of shadows/occlusions are *nonlocal*, as points laying far apart on the surface can become correlated. These nonlocal derivatives are very informative as to the shape of the surface.

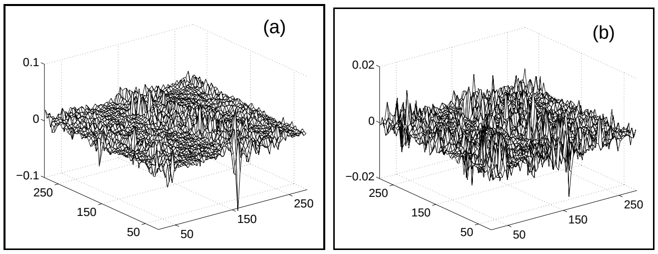

Figure 6: The errors between the inferred and the true surface: (a) error for heights, RMS = 5×10^{-3}; (b) error for albedos, RMS = 8×10^{-5}

Among the other extensions are more realistic reflection functions, blurring and modeling of different surface topologies. Limits to the accuracy of the superresolved surface reconstruction will also be explored.

References

[1] J. Bernardo and A. Smith. *Bayesian Theory.* Wiley, Chichester, New York, 1994.

[2] J. Foley, A. van Dam, S. Finer, and J. Hughes. *Computer Graphics, principles and practice.* Addison-Wesley, 2nd ed. edition, 1990.

[3] B. Horn and M. Brooks. *Shape from Shading.* MIT Press, 1989.

[4] S. Nayar and M. Oren. Visual appearance of matte surfaces. *Science*, Vol. 267, pp. 1153–1156, 1995.

[5] W. Rees. *Physical principles of remote sensing.* Cambridge University Press, 1990.

[6] P. Cheesman, B. Kanefsky, R. Kraft, J. Stutz, and R. Hanson. Super-resolved surface reconstruction from multiple images. In G.R. Heidberg, editor, *Maximum Entropy and Baysian Methods*, pp. 293-308, Kluwer, the Netherlands, 1996.

[7] A. Patti, M. Sezan, and A. Tekalp. Superresolution video reconstruction with arbitrary sampling lattices and nonzero aperture time. *IEEE Trans. Image Processing*, Vol. 6, pp. 1064-1076, 1997

[8] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. Technical Report No 2273, INRIA, Sophia Antipolis, 1994.