

## Learning a Near-Optimal Estimator for Surface Shape from Shading\*

DAVID C. KNILL

*Department of Psychology, Brown University, Providence, Rhode Island 02912*

AND

DANIEL KERSTEN<sup>†</sup>

*Department of Cognitive and Linguistic Sciences, Brown University,  
Providence, Rhode Island 02912*

Received April 6, 1988; revised May 3, 1989

The problem of determining surface shape from shading is formulated in terms of Bayesian estimation. The goal is to select an estimate of surface shape that best fits some criterion on the posterior probability of the surface conditional on the image data. This conditional probability is a function of the imaging function and the prior probability of the surface. A gradient descent technique is used to compute the best linear estimator of the mean of the conditional distribution from a set of random fractal surfaces and their images. Simulations show that the derived estimator works well for a wide range of surfaces, including non-fractals. Once learned, the estimator could be used to implement a fast, non-iterative parallel device for estimating shape from shading with arbitrary light source directions. The estimator has similar perceptual deficits to human observers. In particular, it exhibits a loss of coarse-scale surface shape, while accurately reconstructing fine-scale details. © 1990 Academic Press, Inc.

### 1. INTRODUCTION

The two-dimensional retinal image providing information about a scene to the visual system is a complex function of the three-dimensional structure of the environment, the viewing conditions, and the illumination. The visual system must invert this “imaging” function, in order to estimate three-dimensional properties of the scene. Many of the problems of early vision fall naturally into the framework of inverse problems [2, 3]. Examples include shape from shading, structure from stereo, structure from motion, and reflectance estimation.

A notable characteristic of many of these problems is that they are ill-posed, either because they are under-determined (i.e., no unique solution exists) or because of noisy and sparse image data. The solution spaces of the problems must somehow be constrained in order to derive a unique solution. Computational models for solution of these problems typically employ heuristic constraints which have some intuitive appeal to the modellers. The most popular of these is some form of smoothness constraint, having been applied to problems as diverse as shape from shading [1], structure from stereo [4], and motion detection [5].

\*This research was supported by BRSG PHS2 507 RR 07085 and NSF Grant BNS-8708532 to Daniel Kersten, and by NSF BNS-85-18675 and ONR N-00014-86-K-0600 to James A. Anderson. The authors would like to acknowledge the help and support of the neural modelling group at Brown University. We would especially like to thank Jim Anderson and Mike Rossen for many helpful discussions and suggestions.

<sup>†</sup>Present address: Department of Psychology, University of Minnesota, Minneapolis, MN 55455.

This approach has two problems. The first is the determination of which constraints to use in the solution. This is not trivial, as it requires accurate knowledge of the structure of the environment. The second is the translation of this knowledge into a mathematical representation suitable for model building. For practical reasons, the latter problem often guides the selection of constraints.

An alternative approach to solving these inverse problems is to use an adaptive system which learns a desired inverse mapping through associative pairing of real images with real scene characteristics [6–8]. The constraints existent on the scene characteristics will be embodied in the learned mapping, effectively unburdening the modeller from the task of specifying them. How well the mapping captures the natural constraints will depend on the representations used, the nature of the constraints, and the power of the learning algorithm (e.g., linear, polynomial, etc.).

In this paper, we derive a linear estimator of surface shape from shading using the Widrow–Hoff associative learning algorithm. Though we only consider the problem of shape from shading, the techniques studied here may be applied to a range of problems, and, hopefully, the results we report will serve to guide research into these areas. We view the problem as one in statistical estimation and develop the model within this framework. In the second section, we introduce the problem and review some of the past modelling work on it. The third section contains a brief overview of Bayesian estimation theory. The fourth section introduces the learning algorithm. In the fifth section we describe a statistical model of natural surfaces which we use in the implementation of the model. The sixth section describes the implementation of the model. In the seventh section, we present simulation results. The final section contains a summary and discussion.

## 2. SHAPE FROM SHADING

### 2.1. *Problem Definition*

The shading pattern on a surface provides information about the shape of the surface. Recent studies of human perception of shape from shading focus on simple, convex objects with occluding boundaries, the most common being ellipsoids and spheres [9–10]. Without the information provided by the boundaries, these images appear very flat. Shaded images of more complex surfaces, with several peaks and valleys, are, on the other hand, perceived as having three-dimensional shape (see Fig. 1). The work presented in this paper will concentrate on the estimation of shape from shading away from boundaries and other contours, such as self-occlusions and shadows.

In general, an image will be a complex function of many variables; however, with the simplifying assumption of a point light source and matte surfaces, the light energy reflected to the viewer from points on a surface is given by the Lambertian shading equation

$$L = \rho\lambda(\mathbf{N} \cdot \mathbf{I}), \quad (1)$$

where  $L$  is the luminance,  $\rho$  is the surface reflectance,  $\lambda$  is the light energy flux incident on the surface,  $\mathbf{N}$  is the surface normal vector at a point, and  $\mathbf{I}$  is the unit vector in the direction of the light source. We ignore the problem of detecting variations in surface reflectance,  $\rho$ , which relates to work on lightness constancy [12], and assume it is a constant. In most image formation models,  $\rho$  is assumed to

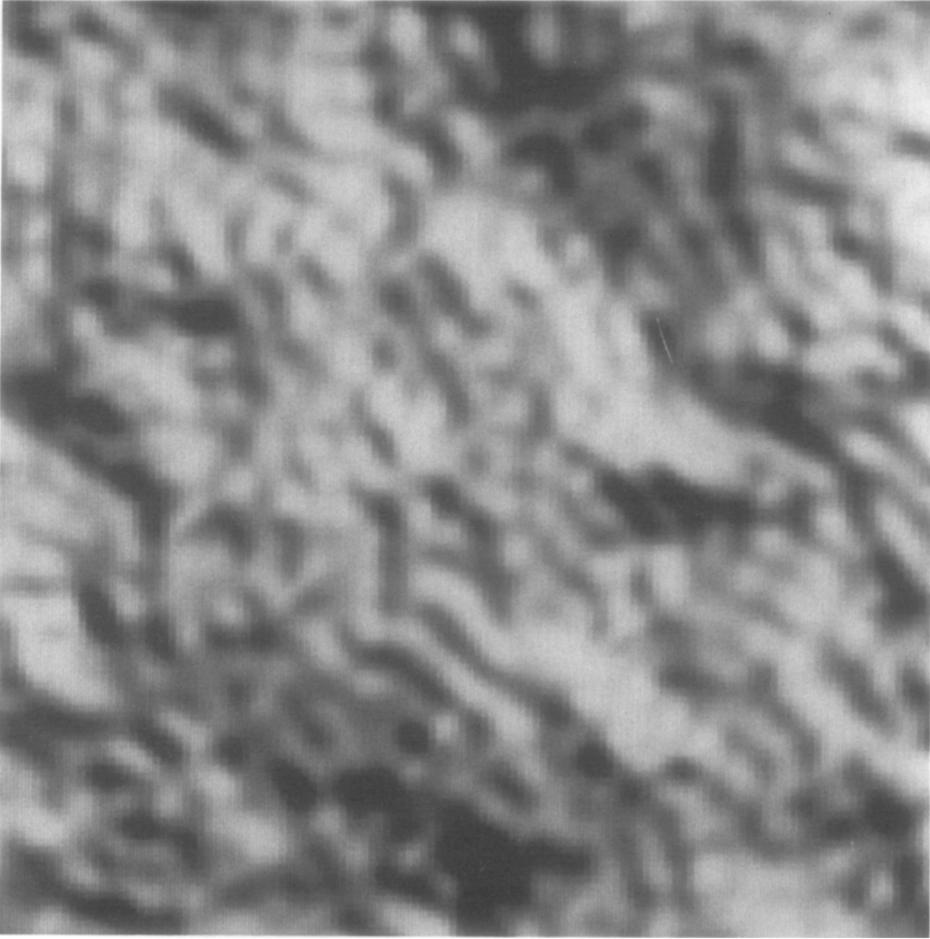


FIG. 1. Shaded image of a fractal surface. The surface has a fractal dimension of 2.15 and is low-pass filtered with an upper cutoff frequency of 24 cycles per surface (see Section 6 for description of fractal surfaces).

be statistically independent of shape and illuminant away from contours and thus may be derived independently.

In a coordinate system,  $(x, y, z)$ , where  $z$  is the surface height and is taken positive in the direction of the viewer, we can represent  $\mathbf{N}$  as the vector,  $(n_x, n_y, n_z)^T$ , whose components are given by

$$n_x = \frac{-p}{\sqrt{p^2 + q^2 + 1}}, \quad (2)$$

$$n_y = \frac{-q}{\sqrt{p^2 + q^2 + 1}}, \quad (3)$$

$$n_z = \frac{1}{\sqrt{p^2 + q^2 + 1}} = \sqrt{1 - n_x^2 - n_y^2}. \quad (4)$$

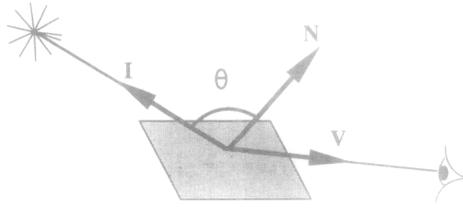


FIG. 2. Schematic of the image formation process. For a Lambertian surface, reflected luminance is proportional to the cosine between surface normal and illuminant direction vectors.

where

$$p = \frac{\partial z}{\partial x}, \quad q = \frac{\partial z}{\partial y}. \quad (5)$$

We have assumed an orientation for surfaces such that the surface normals point toward the viewer; thus,  $n_z$  is always positive. The illuminant vector,  $\mathbf{I}$ , is written as  $(i_x, i_y, i_z)^T$ . Expanding (1), we obtain for the shading equation,

$$L = \rho\lambda \left( i_x n_x + i_y n_y + i_z \sqrt{1 - n_x^2 - n_y^2} \right). \quad (6)$$

Figure 2 summarizes the imaging geometry. Our goal is to invert (6) and estimate  $n_x$  and  $n_y$  from the data  $L$  at each point in the image.

## 2.2. Previous Work

The models which have been developed for solving shape from shading fall into two general categories; those which employ variational techniques and those which use assumptions about the local geometry of surfaces to derive a closed-form shape estimator. In the variational approach [1, 13], solution of the problem consists of finding the surface which minimizes some integral error functional. The functional consists of an error term, which reflects how well the surface matches a given image, and one or more penalty terms which embody some constraints on the solution surface. The penalty term measures how far a surface departs from the given constraints. The functional takes the form of an integral evaluated in a bounded region,  $R$ , of the image

$$\iint_R (L(x, y) - f(\mathbf{S}(x, y)))^2 + \lambda P(\mathbf{S}(x, y))^2 dx dy, \quad (7)$$

where  $\mathbf{S}$  is a representation of local surface shape,  $f(\cdot)$  is the shading function (in our case, the lambertian equation given in (1)), and  $P(\cdot)$  is a penalty function. The representation used for  $\mathbf{S}$  is usually selected for mathematical convenience in the formulation of  $P(\cdot)$ . The constant  $\lambda$  is a Lagrange multiplier which weights the relative contribution of the penalty and error terms.

Typical penalty functions are those which embody an integrability constraint and those which embody some form of heuristic smoothness constraint. The integrability

constraint is an example of a “hard” constraint, in that it is satisfied by all differentiable  $C^2$  surfaces. A specific definition of the constraint is that the second mixed derivatives of surface depth taken in different order, are equal; that is,

$$\frac{\partial^2 z}{\partial x \partial y} = \frac{\partial^2 z}{\partial y \partial x}, \quad p_y = q_x. \quad (8)$$

Horn and Brooks [13] proposed a penalty term for this constraint given by

$$P = (p_y - q_x)^2. \quad (9)$$

An example of a smoothness is the “thin-plate” surface model originally developed by Terzopoulos [14] for surface interpolation between sparse depth values. This is given by

$$P = \left( \frac{\partial^2 z}{\partial x^2} \right)^2 + 2 \left( \frac{\partial^2 z}{\partial x \partial y} \right)^2 + \left( \frac{\partial^2 z}{\partial y^2} \right)^2. \quad (10)$$

Minimizing (10) is approximately equivalent to minimizing the tensile energy in a thin metal plate.

The variational approach ultimately leads to a set of non-linear differential equations which require boundary conditions for their solution. These conditions are generally taken from the known surface orientations at self-occluding contours in an image, though, theoretically, they could be known orientations at any point on the surface; for example, those derived at positions with specular highlights. The models are, however, limited by the requirement that such boundary conditions be derivable from the image. The clear perception of shape in the image of Fig. 1, in which there exist no self-occlusions or specular highlights, indicates that human perception of shape from shading does not share this limitation.

A second approach to solving shape from shading has been to make assumptions about the local geometry of surfaces and to derive closed-form solutions for local surface shape. Pentland [15], and later Lee and Rosenfeld [16], constrain the solution space by assuming that surface patches are umbilical; that is, they have equal principal curvatures. This is akin to assuming that surface patches may be approximated as being locally spherical. Using this constraint, they derive local estimators of surface orientation based on the directional derivatives of image intensity. These models are not limited by the need for boundary conditions; however, they do rely on an assumption of surface structure which is certainly not realistic, though it may sometimes lead to reasonable solutions.

### 3. STATISTICAL ESTIMATION

In this paper, we pose the problem of shape from shading as one in statistical estimation, and describe the constraints on the solution space as existing in the global statistical structure of surfaces. We can incorporate these constraints into the solution of the problem using Bayesian estimation techniques [2, 17]. This formulation has several advantages. First, it provides a general framework within which performance of models and biological systems can be evaluated and compared.

Second one can theoretically measure the statistical structure of natural surfaces, so that any model we use of that structure can be verified. Finally, for the application of a learning system, a statistical model of natural surfaces provides a means for computer synthesis of naturalistic surfaces for use as examples in training the system.

We model image formation as a two-dimensional stochastic process, in which the image is a deterministic function of the scene with additive white Gaussian noise. The actual data available to an observer is a vector of discrete samples of this process. The general equation for the observed image is

$$\mathbf{L} = f(\mathbf{S}) + \mathbf{M}, \quad (11)$$

where  $\mathbf{L}$  is a vector of image intensities,  $\mathbf{S}$  is a vector representing local scene characteristics,<sup>1</sup>  $f(\cdot)$  is the deterministic imaging function, and  $\mathbf{M}$  is a vector of independent noise samples. In general, the noise may be non-white or it may be combined with the image in some non-linear way, such as multiplicatively; however, we will only deal here with this simplified case.

Using Baye's theorem, we can formulate the posterior probability of the scene, given an image, as

$$p(\mathbf{S}|\mathbf{L}) = \frac{p(\mathbf{L}|\mathbf{S})p(\mathbf{S})}{p(\mathbf{L})}. \quad (12)$$

From (11), the forward probability of the image, given the scene,  $p(\mathbf{L}|\mathbf{S})$ , is equivalent to  $p(\mathbf{M} = \mathbf{L} - f(\mathbf{S}))$ . As the elements of  $\mathbf{M}$  are uncorrelated Gaussian random variables, this is given by

$$p(\mathbf{L}|\mathbf{S}) = p(\mathbf{M} = \mathbf{L} - f(\mathbf{S})), \quad (13)$$

$$p(\mathbf{L} = \mathbf{L}|\mathbf{S} = \mathbf{S}) = k \times \exp\left[\frac{-1}{2\sigma^2}(\mathbf{L} - f(\mathbf{S}))^T(\mathbf{L} - f(\mathbf{S}))\right]. \quad (14)$$

$p(\mathbf{S})$  is the prior probability of the scene and embodies the constraints on natural scenes. In effect, it represents our prior knowledge about the structure of scenes.  $p(\mathbf{L})$  is the prior probability of the image; however, for a given image, this is constant and so does not affect the estimation.

To estimate a set of scene characteristics from an image, we must specify some criterion on the posterior distribution,  $p(\mathbf{S}|\mathbf{L})$ , which defines an optimal solution. The two most common are the mode and the mean of the distribution. In the first case, the optimal solution is defined as the most probable scene given an image. This is referred to as maximum a posterior (MAP) estimation. In the second case, the optimal solution is the one which minimizes the mean squared error of the estimates. It is referred to as minimum mean squared error (MMSE) estimation.

Let us represent  $p(\mathbf{S})$  using a Gibbs distribution,

$$p(\mathbf{S} = \mathbf{S}) = k_1 \times \exp[-U(\mathbf{S})], \quad (15)$$

<sup>1</sup>For shape,  $\mathbf{S}$  is a tensor of rank two, corresponding to a set of sub-vectors representing surface orientation, indexed by spatial position. One can, however, represent it as a one-dimensional vector by concatenation of each of these subvectors.

where  $U(\mathbf{S})$  is referred to as the energy or Gibbs potential of the distribution. The constant  $k_1$  is set to

$$k_1 = \frac{1}{\int_{s_1} \dots \int_{s_n} \exp[-U(\mathbf{S})] ds_1 \dots ds_n} \quad (16)$$

in order to normalize  $p(\mathbf{S})$ . The posterior distribution reduces to

$$p(\mathbf{S} = \mathbf{S} | \mathbf{L} = \mathbf{L}) = k_2 \times \exp \left[ -\frac{1}{2\sigma^2} (\mathbf{L} - f(\mathbf{S}))^T (\mathbf{L} - f(\mathbf{S})) - U(\mathbf{S}) \right]. \quad (17)$$

MAP estimation requires the minimization of the function in the exponent of the distribution. This function is equivalent to a discrete formulation of the integral of the error functional given in (7), with  $P(\mathbf{S})^2 = U(\mathbf{S})$ , and  $\lambda = 2\sigma^2$ . A natural interpretation, then, of the penalty function used in variational approaches is that it parameterizes a probability distribution for  $\mathbf{S}$ . Values of  $\mathbf{S}$  for which  $P(\mathbf{S})^2$  is small are more likely than values for which  $P(\mathbf{S})^2$  is large.

In the simple case that  $\mathbf{S}$  is Gaussian, the posterior probability reduces to

$$p(\mathbf{S} = \mathbf{S} | \mathbf{L} = \mathbf{L}) = k_2 \times \exp \left[ \frac{-1}{2\sigma^2} (\mathbf{L} - f(\mathbf{S}))^T (\mathbf{L} - f(\mathbf{S})) - \mathbf{S}^T \mathbf{P}^{-1} \mathbf{S} \right], \quad (18)$$

with  $\mathbf{P}$  being the covariance matrix for  $\mathbf{S}$ . Both MAP and MMSE estimation, in this case involve the minimization of the exponent of the distribution. The corresponding penalty function is linear and given by the inverse covariance matrix,  $\mathbf{P}^{-1}$ .

The previous discussion suggests a four-step strategy for the application of a Bayesian approach to the solution of inverse imaging problems.

- (1) Specification of the imaging function. This defines the forward distribution,  $p(\mathbf{L} | \mathbf{S})$ .
- (2) Specification of the prior distribution of surfaces,  $p(\mathbf{S})$ .
- (3) Definition of criterion on  $p(\mathbf{S} | \mathbf{L})$  for an optimal solution.
- (4) An algorithm for finding the solution which matches the defined criterion.

For the problem of shape from shading, the imaging function is given in the form of Lambert's law. Combining Eqs. (1) and (11) and substituting the surface normal vector for  $\mathbf{S}$ , we obtain

$$\mathbf{L}_{i,j} = \lambda \mathbf{N}_{i,j} \cdot \mathbf{I} + \mathbf{M}_{i,j}. \quad (19)$$

We will leave a discussion of the prior distribution,  $p(\mathbf{S})$ , until Section 4. The criterion we use to define an optimal solution is that we minimize the mean squared error over all estimates; thus, the model will estimate the mean of the posterior distribution. We use an associative learning algorithm to derive a near optimal mapping between images and surfaces. This is described in the next section.

## 4. LEARNING A NEAR OPTIMAL ESTIMATOR

We would like to derive a pseudoinverse mapping from images to surfaces,

$$\hat{\mathbf{S}} = f^*(\mathbf{L}), \quad (20)$$

such that the average error between the estimated surfaces and the real surfaces is minimized. If the imaging function were linear and the distribution of  $\mathbf{S}$  were Gaussian, the optimal mapping would be linear. Unfortunately, for shape from shading neither of these conditions hold. A linear mapping may, however, be near optimal and, given the availability of learning algorithms for deriving such mappings, it is worth investigating their performance. As we shall see, for the problem of shape from shading, the linear map may be implemented as a localized convolution operator (see Section 6). Such an operator may be efficiently implemented on standard signal processing chips and does not require the iterative schemes of non-linear models. If the model works reasonably well, the gain in efficiency may be worth the loss of accuracy. When greater accuracy is desired, the estimate obtained from application of a linear operator may be used as an initial condition for a more complex iterative algorithm, in order to speed up processing.

Given pairs of example surfaces and images, we can use the Widrow-Hoff error-correcting algorithm to derive a linear mapping between them which minimizes the mean squared error between estimated and real surfaces [18]. In the connectionist literature, this type of algorithm is termed linear associative learning [19, 20], and is often interpreted to be computing a generalized inverse. We will refer to the example surfaces and images as the training stimuli. The learning rule is given by

$$\Delta A = \rho(\mathbf{S}_k - A\mathbf{L}_k)\mathbf{L}_k^T. \quad (21)$$

where  $A$  is the mapping being learned and the term in parentheses is the error between estimated and real values of  $\mathbf{S}_k$ . Iterative application of this rule to example vectors,  $\mathbf{S}_k$  and  $\mathbf{L}_k$ , with appropriate relaxation of the learning constant,  $\rho$ , will result in a convergence of  $A$  to the desired mapping.

If the example surfaces are randomly drawn from the distribution  $p(\mathbf{S})$ , and the images are calculated using the imaging function in (1), the derived mapping may be interpreted as the best linear estimate of the mean of  $p(\mathbf{S}|\mathbf{L})$ . This requires that the best set of sample surfaces used during learning span the probability space defined by  $p(\mathbf{S})$ . Of course, the actual form of the distribution need not be known, as it is implicitly defined by the set of sample surfaces. An optimal mapping for a particular space of surfaces may thus be learned through appropriate selection of samples.

## 5. SURFACE MODEL

In order to apply the learning algorithm to derive a shape from shading estimator, we need to specify a set of example surfaces to use as training stimuli. We could use examples of real surfaces as our training set, as suggested in the previous section. This would, in some sense, be the optimal approach; however, it has two difficulties. It requires the collection of a set of example surfaces which appropriately span the space of real surfaces, and it requires the accurate measurement of the shapes of

these surfaces. The other possibility is to use a statistical model of surfaces to artificially synthesize a set of example surfaces for use in training. This avoids the data collection problems of using real surfaces and has the further advantage that it provides a tool for analyzing the performance of the shape from shading estimator.

In this paper, we take the second approach, using a simulation of a fractal Brownian process to generate surfaces for the training set. Fractal processes have been shown to accurately model a wide range of natural surfaces [21, 23] and so provide some hope of deriving an estimator of general applicability. Fractal Brownian processes are characterized by their statistical self-similarity<sup>2</sup> expressed in the relationship,

$$p\left(\frac{\mathbf{S}(x + \Delta x) - \mathbf{S}(x)}{|\Delta x|^H} < y\right) = f(y). \quad (22)$$

$\mathbf{S}$ , the random fractal process, and  $x$ , the index of spatial location, may be vector-valued. The exponent,  $H$ , is constrained to have a value between 0 and 1. The fractal dimension of the process,  $\mathbf{S}$ , is given by

$$D = E + 1 - H, \quad (23)$$

where  $E$  is the Euclidean dimension of the function. The relation in (22) expresses the invariance of the statistics of  $\mathbf{S}$  over changes in scale. For two-dimensional processes, the fractal dimension is constrained to lie somewhere between 2 and 3; the fractional part, in some sense, specifying how much of 3D space the process is filling.

We can apply this model to surfaces by letting  $\mathbf{S}$  represent surface depth on a two-dimensional lattice. This lattice corresponds to the region of a scene projected onto an image, with surface depth measured in the direction of the observer. Figure 3 shows two examples of fractal surfaces, the first with  $D = 2.1$ , and the second with  $D = 2.5$ . Notice that the surface with higher fractal dimension appears rougher than the one with lower dimension. A fractal dimension near 2.15 seems to accurately model many natural surfaces [21].

A fractal Brownian process is Gaussian; therefore, it can be fully specified by its power spectrum. The power spectrum of a one-dimensional walk on a two-dimensional fractal process is given by [22]

$$P(f) = \frac{k}{f^\beta}, \quad f > 0, \quad (24)$$

where

$$\beta = 7 - 2D. \quad (25)$$

<sup>2</sup>Fractal Brownian processes are technically self-affine. A self-similar process has the property that a piece of the process,  $1/n$ th the size of the original, when scaled by a factor  $n$ , is in all statistical respects the same as the whole process. For a fractal Brownian process, the scaling factor is not  $n$ , but  $n^H$ , with  $0 < H < 1$ .

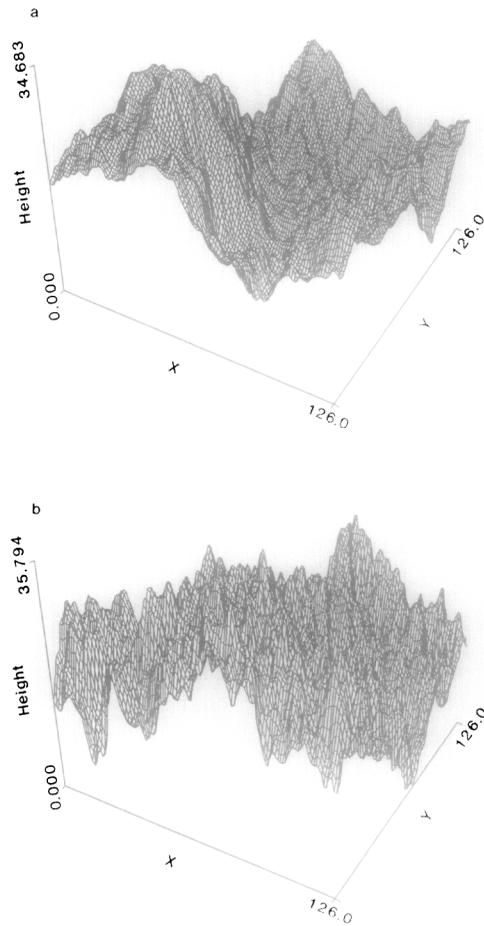


FIG. 3. 3D plots of two fractal surfaces with (a) fractal dimension of 2.15 and (b) fractal dimension of 2.5.

We assume that the surface process is spatially isotropic, so that its two-dimensional power spectrum is independent of orientation and is given as a function of radial spatial frequency,  $f_r$ , by

$$P(f_r) = \frac{k}{f_r^{\beta'}}, \quad f_r > 0. \quad (26)$$

The exponent is related to the fractal dimension by

$$\beta' = 8 - 2D. \quad (27)$$

The constant  $k$  is proportional to the variance of  $\mathbf{S}$ . We further assume that the process  $\mathbf{S}$  is stationary<sup>3</sup>; that is, its statistics are invariant to changes in spatial

<sup>3</sup>Idealized fractional Brownian processes are not stationary, as illustrated by the fact that the integral  $1/f_r^\beta$  is unbounded for all  $\beta$ . If we allow the power spectrum to have a low frequency cutoff, however, the integral becomes bounded for all  $\beta > 2$ ; that is, for  $D < 3$ . Such a process is stationary, and may be described as exhibiting fractal behavior at scales below that defined by the cutoff frequency.

position. Isotropy and stationarity are a natural result of the coordinate system for the surface process being viewer-centered. Since the ensemble of surfaces which we are considering consists of surfaces viewed from all possible positions and orientations (e.g., a ball is equally likely to appear in any position in an image), the statistics of the ensemble should be independent of position and orientation in this coordinate system.

The fractal model is related to the thin-plate potential function introduced by Terzopoulos to enforce surface smoothness. If we use the potential function in (10) to parameterize a Gibbs distribution, the resulting random process is Gaussian with a power spectrum which falls off proportionally to  $1/f_r^4$  [24]. Setting  $\beta' = 4$  in [27], we see that this is equivalent to a fractal Brownian process with  $D = 2$ . The fractal model typically generates complex surfaces which resemble natural landscapes; however, the correspondence between fractal model and the thin-plate smoothness constraint shows that the model also encompasses simpler surfaces such as spheres and ellipsoids (away from occluding boundaries).

The surface model has two free parameters to vary, the fractal dimension and the variance of surface depths, or equivalently, the variance of surface orientations,  $E[\mathbf{p}^2] = E[\mathbf{q}^2]$ . The orientation variance measures the degree of relief in the surfaces of an ensemble. For a given fractal dimension and orientation variance, we generate random surfaces by filtering Gaussian white noise through a filter with the appropriate power spectrum, generating a two-dimensional lattice of surface depths. Surface orientations and normal vectors are calculated using discrete differences between depth values at neighboring points in the lattice. Idealized fractals are not differentiable, so that one cannot define local orientation for a fractal surface. One can, however, define orientation on a coarse scale; correspondingly, we low-pass filtered the surfaces before calculating local surface orientations.

## 6. THE ESTIMATOR

Derivation of the estimator involves three major steps. The first step is the generation of a large number of sample fractal surfaces with a given fractal dimension and orientation variance. The second step is the synthesis of images of these surfaces using the Lambertian shading model for an assumed point light source at a fixed position. Finally, the Widrow–Hoff learning algorithm is applied to pairs of surfaces and corresponding images to derive the estimator. We represent surfaces using two vectors,  $\mathbf{N}_x$  and  $\mathbf{N}_y$ , containing the surface normal components  $n_x$  and  $n_y$  at discretely sampled points on the surface.<sup>4</sup> These vectors provide a discrete representation of the normal vector field of a surface. The representation fully specifies the shape of the surface down to its first derivative. Images are represented using a vector of luminance values at discretely sampled points, scaled by the mean luminance calculated over the whole image,

$$L'_{i,j} = \frac{L_{i,j}}{\hat{E}[\mathbf{L}]}, \quad (28)$$

<sup>4</sup>We also ran simulations using the first directional derivatives,  $\partial z/\partial x$  and  $\partial z/\partial y$ , to represent local surface shape. The performance of the resulting estimator, though fairly good, was not quite as good as the one derived for surface normals, hence our choice of the surface normal representation for analysis in the paper.

where  $\hat{E}[\mathbf{L}]$  is the sample mean luminance. This scaling eliminates the constant of proportionality,  $\rho\lambda$ , in the shading equation.<sup>5</sup>

Because the surface process is assumed to be stationary, the linear minimum mean squared error estimator for  $\mathbf{N}_x$  and  $\mathbf{N}_y$  will be a pair of convolution operators. Such an estimator can be learned by associating a vector containing  $n_x$  and  $n_y$  for a point on the surface with the image of the surrounding region. The two rows of the resulting matrix will contain discrete, two-dimensional, finite impulse response filters, which, when applied to whole images, will provide an estimate of the vectors,  $\mathbf{N}_x$  and  $\mathbf{N}_y$ . We refer to these as the SFS (for shape from shading) filters. If the filters are  $\mathbf{F}_x$  and  $\mathbf{F}_y$ , the estimated surface shape is given by

$$\mathbf{N}_x = \mathbf{F}_x * \mathbf{L}', \quad (29)$$

$$\mathbf{N}_y = \mathbf{F}_y * \mathbf{L}', \quad (30)$$

where  $*$  is the discrete form of the two-dimensional convolution operator, and the operation is performed away from the edge of the image (to within a distance from the edge of  $\frac{1}{2}$  the width of the filters).

### 6.1. Generalizing to Different Illuminant Directions

The estimator is derived for images generated using a fixed light source direction. We would like to be able to apply the estimator to images generated with different light sources. Generalizing the estimator to images generated with light sources at different tilts (orientation in the image plane) is fairly straightforward. If the illuminant is used for synthesizing images has a tilt of  $45^\circ$ , the SFS filters will be rotated versions of each other, due to the isotropy of the surface (the statistics of  $n_x$  are the same as those of  $n_y$  rotated  $90^\circ$ ). We can apply the SFS filters to images generated with light sources with tilts other than  $45^\circ$  in the following way. Consider a coordinate system defined by the tilt of the light source, so that

$$n'_x = n_x \cos(\theta - 45) - n_y \sin(\theta - 45), \quad (31)$$

$$n'_y = n_x \sin(\theta - 45) + n_y \cos(\theta - 45), \quad (32)$$

where  $\theta$  is the tilt of the light source away from  $0^\circ$ . For images generated with light sources tilted away from  $45^\circ$ , we can estimate surface normal components,  $n'_x$ ,  $n'_y$ , by rotating the filters by  $\theta - 45^\circ$  before applying them. The resulting surface normals are represented in a light source centered coordinate system tilted to match the orientation of the illuminant. A highly accurate estimate of illuminant tilt for use in this procedure is easily derived from the statistics of an image [25, 26].

We will show simulation results which indicate that the estimator generalizes near-perfectly to light sources of different slants (angle away from the viewer), as well.

### 6.2. Generalizing to Different Surfaces

The surfaces used in the derivation of the estimator are continuous surfaces drawn from a fractal ensemble with a fixed orientation variance. Because the fractal

<sup>5</sup>If a scene contains surfaces or regions of surfaces with different reflectances, an independently derived estimate of relative reflectances could be used to weight the scaling factor applied to different parts of the image.

process is ergodic, the sample orientation variances (degrees of relief), of these surfaces are roughly equal. We would like the estimator to work well for surfaces not explicitly drawn from this ensemble; that is, we would like the estimator to generalize over a wide variety of surfaces. These include surfaces with different degrees of relief than those of the training surfaces, surfaces not explicitly generated from the fractal model, such as spheres and ellipsoids, and surfaces which are only piece-wise continuous. In the case of piece-wise continuous surfaces, we would like the estimator to handle the discontinuities gracefully (e.g., by smoothing over them). Performance of the estimator for surfaces with all of the characteristics listed above will be tested in the simulations described in the next section.

## 7. SIMULATIONS

The basic strategy for the simulations is to derive a pair of SFS filters through application of the Widrow–Hoff learning algorithm to a training set of fractal surfaces and their images and to test the filters on images of novel surfaces. The test surfaces and images are computer generated, so that performance of the filters can be measured directly through comparison of the reconstructed surfaces with the original test surfaces. We use three quantitative measures of performance. The first two measure the similarity between the normal vector fields of the original surfaces and the estimated normal vector fields of these surfaces. One measure is the average cosine between the estimated vectors,  $\hat{\mathbf{N}}_x$  and  $\hat{\mathbf{N}}_y$ , and the original vectors,  $\mathbf{N}_x$  and  $\mathbf{N}_y$ . The other is the normalized mean squared error (NMSE) between the vectors. A cosine of 1 indicates perfect reconstruction up to a scaling factor, while a cosine of 0 indicates no correlation between the estimated and original test surfaces. The normalized mean squared error is given by

$$\text{NMSE}_x = \frac{E[(\hat{\mathbf{n}}_x - \mathbf{n}_x)^2]}{2E[\mathbf{n}_x^2]}, \quad (33)$$

$$\text{NMSE}_y = \frac{E[(\hat{\mathbf{n}}_y - \mathbf{n}_y)^2]}{2E[\mathbf{n}_y^2]}, \quad (34)$$

$$\text{NMSE} = \frac{1}{2}(\text{NMSE}_x + \text{NMSE}_y), \quad (35)$$

where  $E[\mathbf{n}_x^2]$  and  $E[\mathbf{n}_y^2]$  are the variances of the  $n_x$  and  $n_y$  components of the surface normals for the test surfaces. The NMSE measure provides an error measure similar to an inverse signal-to-noise ratio. Perfect reconstruction would result in an  $\text{NMSE} = 0$ , while a completely random reconstruction, with equivalent variances for the surface normals would result in an  $\text{NMSE} = 1$ .

The third performance measure provides an estimate of the extent to which the estimated surface normals violate the integrability constraint. As the two vectors of surface normal components are estimated independently, no explicit steps are taken to ensure that they are consistent with the constraint that they represent the normal vector field of a real  $C^2$  surface. We can express the integrability constraint using the relation

$$\oint_C \left( \frac{\partial z}{\partial x} \right) dx + \left( \frac{\partial z}{\partial y} \right) dy = 0; \quad (36)$$

that is, the integral of the surface gradient calculated along a closed contour is zero. A discrete approximation of the contour integral in (36) is the basis for the third performance measure. The approximation of the integral, computed over a one pixel wide cell on a surface, with lower, left-hand corner,  $i, j$  is

$$I_{ij} = p_{i,j} + q_{i,j+1} - p_{i+1,j} - q_{i,j}. \quad (37)$$

If  $p$  and  $q$  were originally computed using local differences between neighboring points on the graph of a discrete two-dimensional function (i.e., a surface), this term will equal zero for all cells. The error measure, which we call the normalized mean squared integral error (NMSIE), is given by

$$\text{NMSIE} = \frac{E[\hat{\mathbf{I}}^2]}{4E[\hat{\mathbf{p}}^2]}. \quad (38)$$

Values of  $\hat{p}$  and  $\hat{q}$  used to calculate  $I$  are computed from the estimated surface normals, and  $E[\hat{\mathbf{p}}^2]$  (or, equivalently,  $E[\hat{\mathbf{q}}]$ ) is the orientation variance of the estimated surfaces. The scaling factor normalizes the error measure, so that random estimates,  $\hat{N}_x$  and  $\hat{N}_y$ , have an NMSIE of 1, and estimates which perfectly match the integrability constraint have an NMSIE of 0.

We ran five sets of simulations to study different performance characteristics of the learned shape from shading estimators. The first set of simulations provides baseline performance data for a pair of filters applied to test images generated using the same set of surface and illuminant parameters as those used in the derivation of the filter. We look at the effect of filter size on performance in the second set of simulations. In the third set of simulations, we examine the performance of the filters on images of surfaces with different orientation variances than those used in the training set. In the fourth set of simulations we look at the performance of the filters on images generated with a range of illuminant slants. Finally, we apply the filters to non-fractal surfaces in the fifth set of simulations.

*Simulation 1.* Table 1 lists the training set parameters for the first set of simulations. We generated a set of 800  $29 \times 29$  pixel surfaces, with a fractal dimension of 2.15 and an orientation variance of 0.1. The light source had a tilt of  $45^\circ$  and a slant of  $35^\circ$  away from the viewer. These parameters effectively guaranteed that no points on the surface would be in shadow. We associated the directional components of the surface normal at the center point on the surface with the image of the entire surface, generating a  $2 \times 841$  association matrix. The rows of

TABLE 1  
Parameters Used to Generate Surfaces and Their Images

Training set	Surface parameters			Illuminant parameters	
	Fractal dim.	Bandwidth (cycles/surface)	Orientation variance	Tilt	Slant
1	2.15	0–24	0.1	$45^\circ$	$35^\circ$

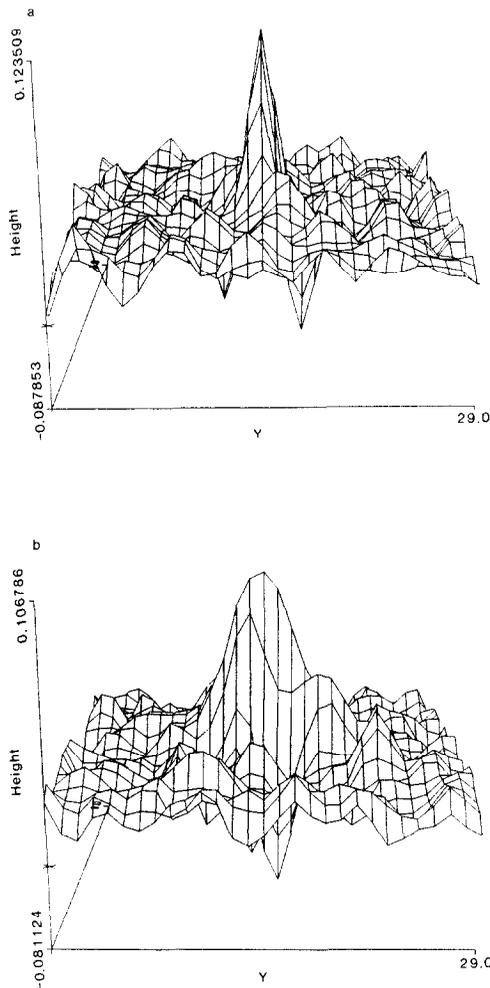


FIG. 4.  $29 \times 29$  pixel SFS filters generated in simulation 1: (a)  $F_x$  filter; (b)  $F_y$  filter.

this matrix are the convolution filters,  $F_x$  and  $F_y$ , which can be applied to larger images to estimate surface shape. These are shown in Fig. 4. Note that the filters appear qualitatively like second directional derivative filters, with a marked asymmetry in the direction of the light source.

We applied the filters shown in Fig. 4 to images of  $128 \times 128$  pixel surfaces generated with the parameters in Table 1 to estimate the surface normals at each point on the surface. Figure 5 shows an example of an estimated surface with the original surface from which the image was generated (see Appendix A for details of how the depths were derived from the estimated surface normal vectors). The cosine for this reconstruction was 0.81 and the normalized mean squared error was 0.17. As it is difficult to get a good feel for the performance of the model looking at the full 3D plots, we also show plots of small subsections of the surfaces. The reconstruction looks qualitatively poor when the entire surface is plotted, as in (a) and (b) of Fig. 5, but looks much better when only a small subsection is shown, as in (c) and (d) of

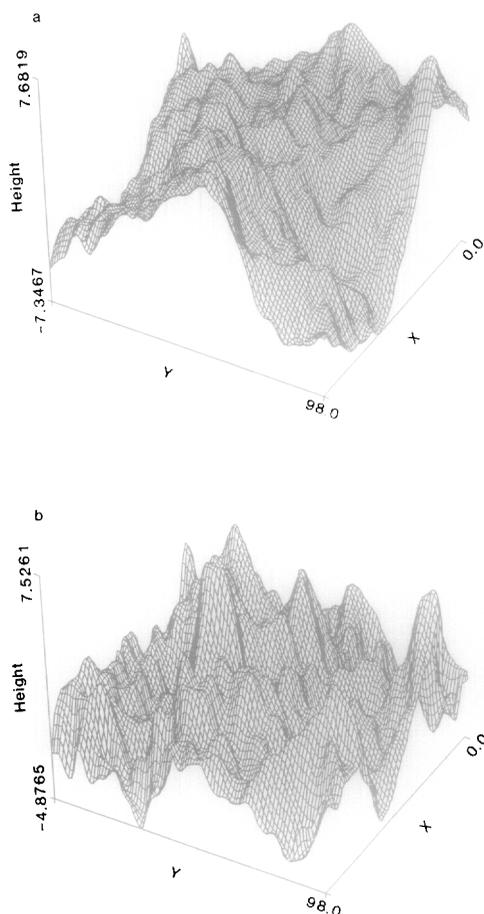


FIG. 5. 3D plots of test surface and surface reconstructed by application of SFS filter to its image: (a) test surface; (b) reconstructed surface; (c) center  $32 \times 32$  section of test surface; (d) center  $32 \times 32$  section of reconstructed surface.

the same figure. Figure 6 shows 2D plots of randomly selected slices through the surfaces. Both Figs. 5 and 6 indicate that the most notable difference between the estimated and original surfaces is in the low frequency components of shape. The filters clearly discard low frequencies in their estimates. The SFS filters shown in Fig. 4 are bandpass; however, fractal surfaces have much greater power at low frequencies than at high. The high frequency attenuation of the filters may be primarily due to the small amount of high frequency content in the surfaces; however, the low frequency attenuation of the filters represents a real loss of information about surface shape.

Application of these filters to 40 images of novel surfaces resulted in an average reconstruction cosine of 0.795 and an NMSE of 0.332. The NMSIE for the reconstructions was 0.025, indicating that the estimated surface normals do, in fact, match the integrability constraint as well.

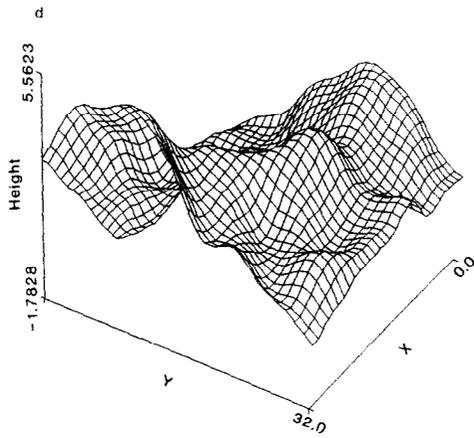
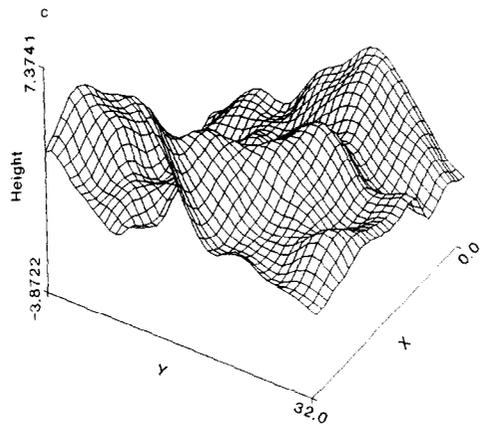


FIG. 5—Continued

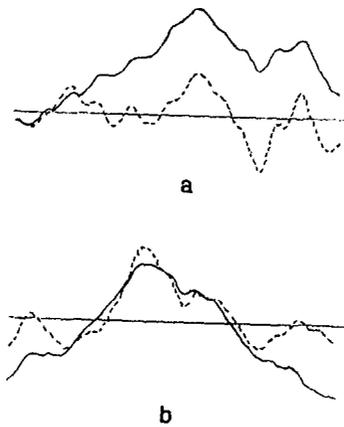


FIG. 6. 2D slices through test (solid line) and reconstructed (dashed line) surfaces shown in Fig. 6.

**Filter Performance vs. Filter Size**

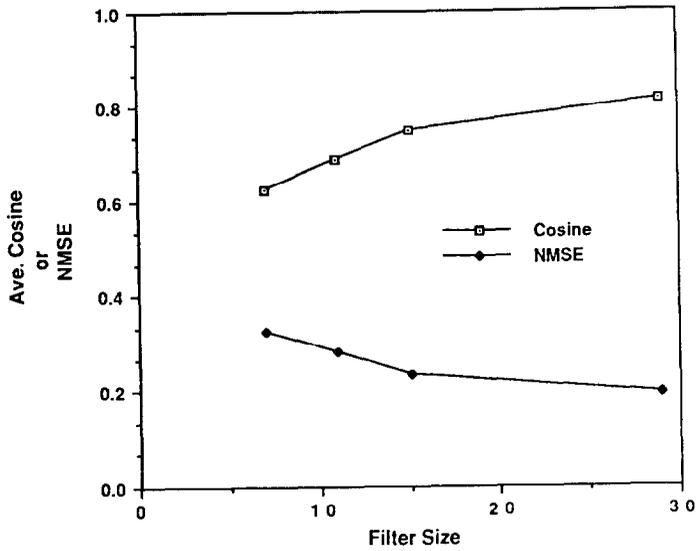


FIG. 7. Plots of average cosine and NMSE performance measures for SFS filters as a function of filter size.

**Filter Performance vs. Orientation Variance**

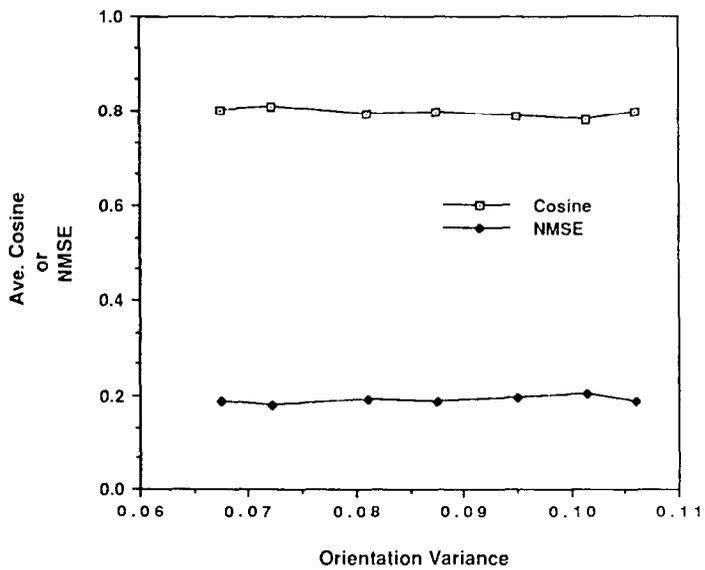


FIG. 8. Plots of average cosine and NMSE performance measures for  $29 \times 29$  SFS filters as a function of test surface orientation variance.

## Filter Performance vs. Illuminant Slant

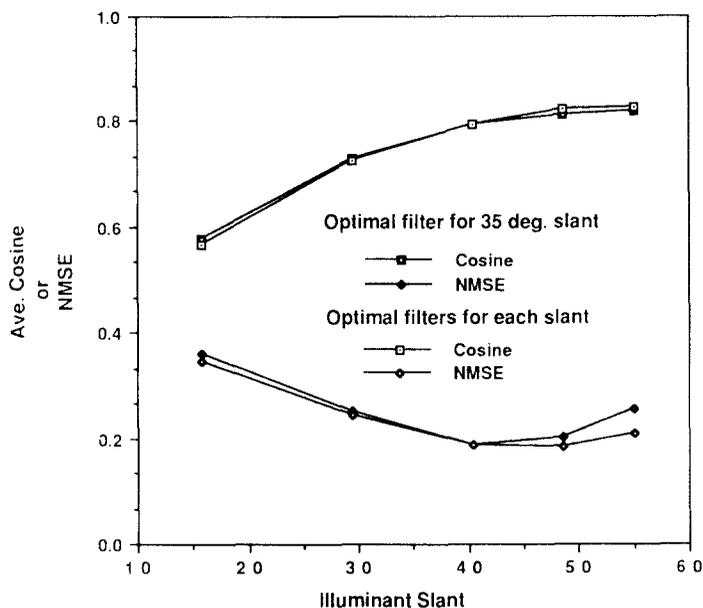


FIG. 9. Plots of cosine and NMSE performance measures for  $29 \times 29$  SFS filters as a function of illuminant slant. Open triangles are the performance of optimal filters derived for each individual slant. Closed triangles are the performance of SFS filters originally derived for a slant of  $35^\circ$ .

*Simulation 2.* The second set of simulations was designed to study the effect of filter size on the performance of the estimator. We derived filters of four different sizes using training sets with the parameters listed in Table 1. The four sizes used were  $7 \times 7$  pixels,  $11 \times 11$  pixels,  $15 \times 15$  pixels, and the original  $29 \times 29$  pixels. Figure 7 shows plots of the average cosine and NMSE measures for these filters tested on 40 images of novel surfaces. Asymptotic performance is reached near the  $29 \times 29$  pixel size originally used.

*Simulation 3.* We applied the SFS filters derived in the first simulation to images of surfaces with a range of orientation variances. The parameters used to generate the test surfaces were the same as those used in Simulation 1, with the exception that the orientation variance of the surfaces ranged from 0.06 to 0.11. The results of applying the filters to images of the test surfaces are shown in Fig. 8. Neither the average cosine nor the NMSE for the reconstructions changes significantly between orientation variances. As shown by the performance measures in Fig. 9, the filters generalize well to a range of orientation variances.

*Simulation 4.* It is clear from the formulation of the problem that a set of filters derived for one illuminant slant will generalize to work well on images with other slants. We derived different pairs of filters using each of five different illuminant slants. These represent optimal estimators for each slant. To study the generalizability of the estimator over different illuminant slants, we applied the original pair of filters to images generated with each of the six slants and compared

its performance with that of the optimal estimators for each slant. Note that we tested the filters for slants above and below that used to generate the images on which they were initially trained. Figure 9 shows the results. The performance of the pair of filters derived for one slant ( $35^\circ$ ) does not differ significantly from the performance of the optimal pair of filters derived for each individual slant. This indicates near perfect generalization, in the sense that a pair of filters derived for one slant work as well as possible for other slants, given the constraint that the filters be linear.

Performance does improve with increasing illuminant slant. For a  $0^\circ$  slant, the images are completely ambiguous, as two equally likely surfaces could give rise to the same image, one with opposite curvatures from the other. Learning an estimator for this condition by random sampling of surface is, therefore, impossible. As the light source moves away from the line of sight, the images become less ambiguous, giving an improvement in performance.

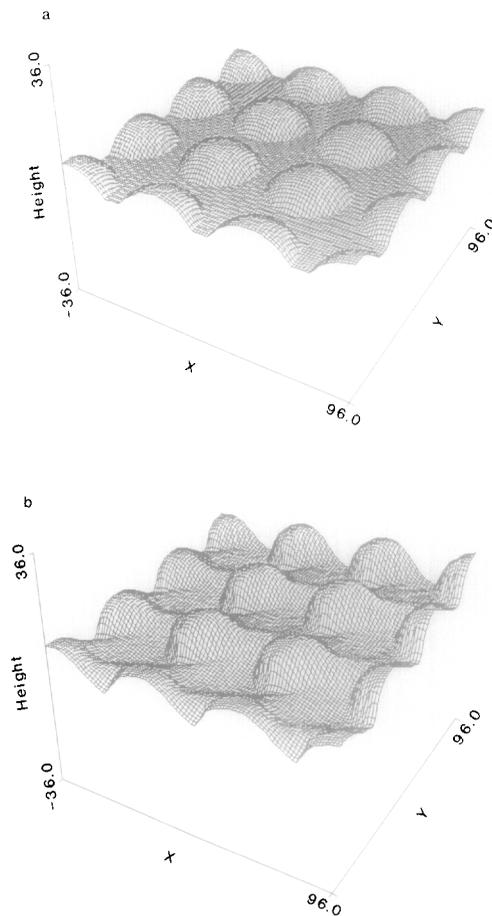


FIG. 10. 3D plots of field of spheres surfaces and surface reconstructed by application of SFS filters to the surface's image: (a) test surface; (b) reconstructed surface.

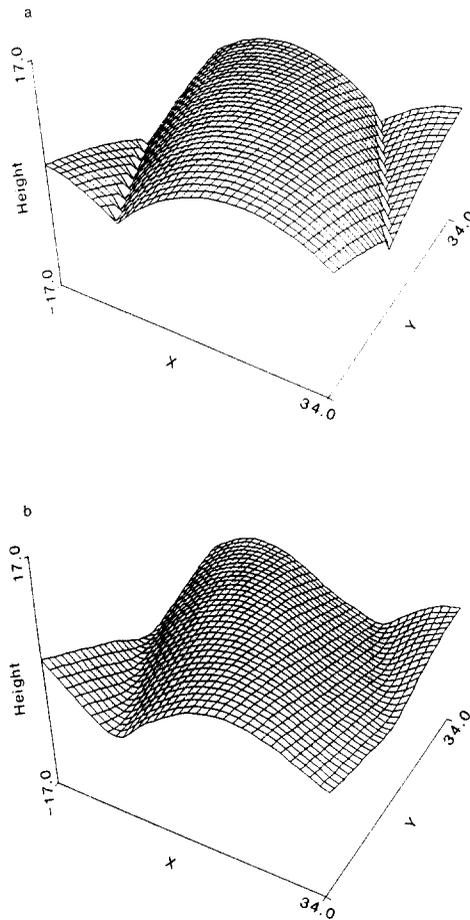


FIG. 11. 3D plots of test grating surface and surface reconstructed by application of SFS filters to the surface's image: (a) test surface; (b) reconstructed surface.

*Simulation 5.* We applied the SFS filters to images of non-fractal surfaces with generally good results. In one simulation we applied the filters to an image of spheres embedded in a plane, as shown in Fig. 10a. Such a surface has a simpler shape than the fractal test surfaces used in the previous simulations. It is also sharp edges around the borders of the spheres. Only the top two-thirds of the hemispheres of each sphere appear above the plane. This greatly reduces the amount of shadow in the image. Figure 10b shows a reconstruction of the surface using the SFS filters. The reconstructed surface is a good approximation to the original with the discontinuities smoothed out. The notable fault in the reconstruction is a small asymmetry in the direction of the illuminant. A similar asymmetry results from application of a non-linear iterative algorithm without boundary conditions [27]. Another example of a simple surface with sharp edges is shown in Fig. 11a. This surface has the further feature that it is highly anisotropic, being an oriented grating. Application of the SFS filters to an image of this surface resulted in the reconstructed surface shown in Fig. 11b. This is a near perfect reconstruction, with

the only difference between the two surfaces being that the sharp edges in the original surface have been smoothed out in the reconstruction.

## 8. DISCUSSION

The derived SFS filters provide a fairly accurate estimate of surface shape from shading. Through the problem is non-linear, application of linear filters seem sufficient to estimate local surface orientation for a broadly defined ensemble of surfaces. Furthermore, filters derived using a training set of sample fractal surfaces generalize to at least some nonfractal surfaces. A practical system could be built using arrays of SFS filters at different orientations. The output of those filters aligned with the illuminant tilt would provide the estimate of local surface orientation.

One way to see how well the performance of the SFS filters matches that of humans is to generate images of the reconstructed surfaces and compare their perceptual appearance with that of the test images. Figure 12 shows a test image and the image of the surface reconstructed from that image. Note the similarity in our perception of the shapes of the two surfaces. This does not result from an equality between the luminance patterns in the two images, as they are, in fact, different, but rather a similarity in the perceptually derived shapes of the two surfaces. Referring back to the observation that the SFS filters discard low frequency shape information, this would indicate that we have the same perceptual deficit. An intriguing conclusion is that there exist sets of surface metamers, the images of which lead to the same perceptual estimate of shape.

This work has some similarity to a model recently developed by Pentland [23]. Pentland approximated the Lambertian shading equation as a linear function of the first derivatives of surface depth and derived a linear estimation of surface depth from shading, without making recourse to prior constraints on the shapes of surfaces.<sup>6</sup> Though Pentland's model uses a different shape representation than that used here (depth vs. surface normals), his analysis sheds some light on our results. Pentland used a Taylor series expansion of the shading equation in  $p$  and  $q$  to show that linear terms dominate the function for points at which  $|p|, |q| \ll 1$  and the illuminant is oblique to the viewer; that is, the illuminant slant is much greater than 0. Relating these constraints to the shading equation in (6) expressed as a function of  $n_x$  and  $n_y$ , we see that the effect of the non-linear term is a monotonic increasing function of  $|n_x|, |n_y|$ , and  $i_z$ . Thus the linear terms of the equation, expressed as a function of  $n_x$  and  $n_y$ , dominant when  $|p|, |q| \ll 1$  and the illuminant slant is large. In the simulations presented here, the orientation variance was small, so that a great majority of the values of  $|p|$  and  $|q|$  were less than 1. Furthermore, the performance of the SFS filters improves with increasing illuminant slant. These results indicate that the accuracy of the estimator may, to some extent, be a function of the applicability of a linear reflectance model.

One other attempt has been made recently to solve the shape from shading problem using an associative learning algorithm. Lehky and Sejnowski [8] used the back-propagation algorithm [29] to train a multilayered neural network to estimate the orientation and direction of the principle curvatures of points on an elliptic

<sup>6</sup>By estimating surface depth instead of surface orientation, the model does, however, implicitly enforce the integrability constraint.

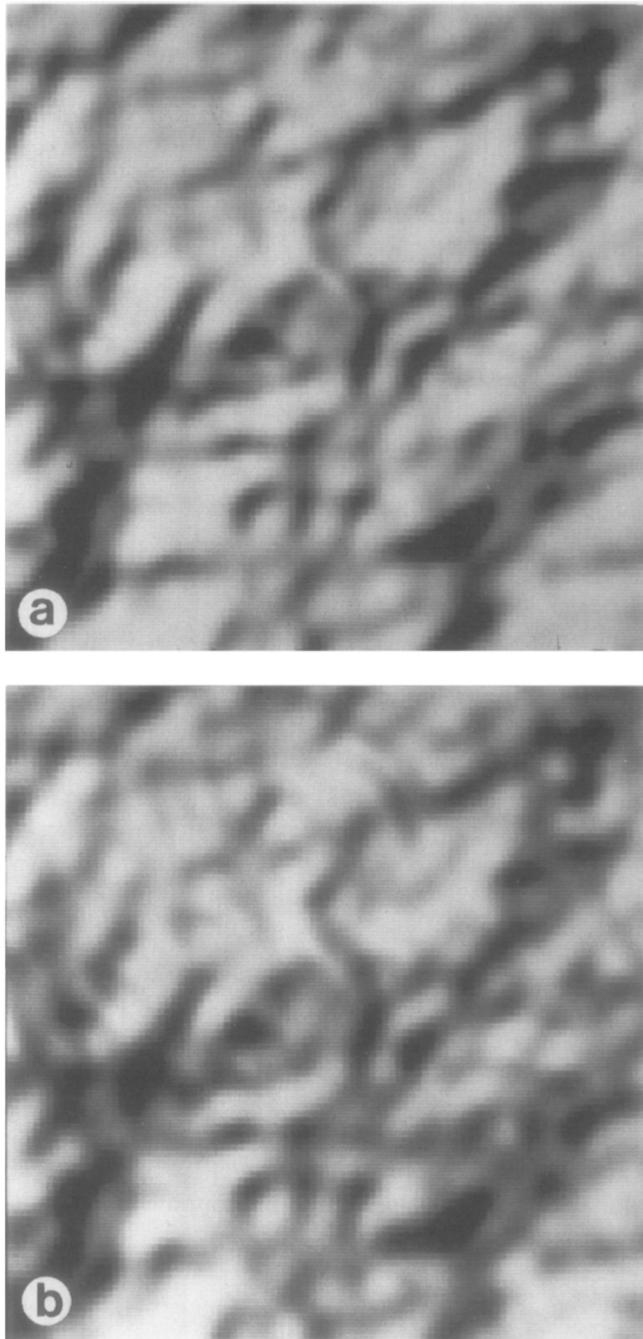


FIG. 12. Images of test surface (a) and surface reconstructed by application of SFS filters to test surface's image (b). The images of the original and reconstructed surfaces were generating using the same illuminant conditions ( $45^\circ$  tilt,  $35^\circ$  slant).

parabaloid. They trained the network to solve this problem independently of illuminant direction. The model generalizes to different parameters of the parabaloid surfaces and to different illuminant slants. Unfortunately, a quantitative comparison between the two models is made impossible by the difference in shape representations used. Lehky and Sejnowski use a distributed representation of principal curvature magnitude and direction, which is, in fact, degenerate (curvature magnitude is ambiguously represented). Though they report good results, their model has a limited domain of applicability, as it generalizes over only a small subclass of simple geometrical objects.

The results of Lehky and Sejnowski seem to suggest extending the learning procedure used in this paper to the learning of non-linear mappings between images and surface shape representations. Though the performance of a non-linear estimator would probably be somewhat better than that reported here for the linear model, it would not be qualitatively very different, as the linear model performs quite well for smooth surfaces. In related work on learning an estimator of surface reflectance, Hurlbert and Poggio [7] found only slight improvements of a linear model for both polynomial mappings and mappings through multiple layers of units with sigmoidal non-linearities (i.e., using back-propagation). Moreover, the principal weakness of the linear estimator is found not in its reconstruction of smooth surfaces, but in its treatment of surface discontinuities. It smooths over discontinuities and is not able to make use of shape information available at self-occluding contours. Currently existing non-linear algorithms share this weakness, in that they do not provide any explicit mechanisms for handling such discontinuities. Several researchers have recently developed techniques for finding discontinuities and incorporating them in surface reconstruction [31, 17, 14, 30]. Extension of learning models to incorporate these techniques is a necessary next step in the application of these models to problems in vision such as shape from shading.

## 9. SUMMARY

We have shown that the application of simple associative learning can be used to derive an accurate linear estimator of surface shape from shading. If the example surfaces used in learning the estimator are drawn from a prior statistical model of natural surfaces and the training images are generated using a well-defined imaging function, the estimator can be interpreted as being the best linear estimate of the mean of a posterior probability distribution,  $p(\mathbf{S}|\mathbf{L})$ . Similar techniques may be applied to solving other problems in vision if appropriate prior models and imaging functions can be specified.

## APPENDIX

The first step in reconstructing a surface from an estimate of the normal vector field of the surface is to convert the representation of the normal vector field to one of the gradient vector field; that is, the set of vectors  $(p_{i,j}, q_{i,j})^T$ . The simplest method of integrating the resultant gradient vectors is to first reconstruct a boundary of the surface and then integrate along perpendicular lines to this boundary using the appropriate element of the gradient vectors. Elements along the bottom

edge of a surface could be reconstructed using

$$z_{0,0} = 0, \quad (39)$$

$$z_{0,j} = z_{0,0} + \sum_{l=0}^{j-1} p_{0,l}, \quad j > 0. \quad (40)$$

The rest of the surface would be reconstructed using

$$z_{i,j} = z_{0,j} + \sum_{k=0}^{i-1} q_{k,j}, \quad i > 0. \quad (41)$$

When the gradient vectors are not exactly integrable this technique leads to a propagation of the errors in the direction in which the integration is done (in the case above, from the lower left corner to the upper right corner of a surface). The resulting surface will match the estimates of  $p$  exactly but will show an increasing error in  $q$  as one moves away from the lower left-hand corner.

We employ a modified version of this algorithm which seems to work quite well for the estimates obtained from the shape from shading estimator. The algorithm estimates the depth at surface points by averaging the results of integration in opposite directions around one pixel cell on the surface. The algorithm steps through the surface from left to right and from top to bottom, so that previously calculated depth values provided initial conditions for succeeding steps in the procedure. Depth values along the bottom and left edges of the surface are computed using

$$z_{0,0} = 0, \quad (42)$$

$$z_{0,j} = z_{0,0} + \sum_{l=0}^{j-1} p_{0,l}, \quad j > 0 \quad (43)$$

$$z_{i,0} = z_{0,0} + \sum_{k=0}^{i-1} q_{k,0}, \quad i > 0 \quad (44)$$

Depth values at interior points on the surface are computed using

$$z_{i,j} = \frac{1}{2}(z_{i-1,j} + q_{i-1,j} + z_{i,j-1} + p_{i,j-1}), \quad i > 0, \quad j > 0. \quad (45)$$

where depth values are computed moving from  $i = 1$  to  $N$  and  $j = 1$  to  $N$ . For each depth value computed, this procedure minimizes the error between the surface and the estimated gradient, given the previously estimated depth values. The procedure does not ensure that the global error is minimized; however, the quality of the reconstructions and the small integrability error shown in the first set of simulations indicates that such a simple algorithm is sufficient for our needs. More complex algorithms which do minimize the error between a reconstructed and a noisy gradient field are given in [13, 27].

## REFERENCES

1. K. Ikeuchi and B. K. P. Horn, Numerical shape from shading and occluding boundaries, *Artif. Intell.* **17**, 1981, 141–184.
2. J. L. Marroquin, *Probabilistic Solution of Inverse Problems*, MIT AI Memo 860, 1985.
3. T. Poggio, V. Torre, and C. Koch, Computational theory and regularization theory, *Nature (London)* **317**, 1985, 314–319.
4. D. Marr, and T. Poggio, Cooperative computation of stereo disparity, *Science* **194**, 1976, 283–287.
5. E. C. Hildreth, *The Measurement of Visual Motion*, MIT Press, Cambridge, Mass, 1983.
6. D. Kersten, A. J. O'Toole, M. E. Sereno, D. C. Knill and J. A. Anderson, Associative learning of scene parameters from images, *Appl. Opt.* **26**, No. 23, 1987, 4999–5006.
7. A. C. Hurlbert and T. A. Poggio, Synthesizing a color algorithm from examples, *Science* **239**, 1988, 482–485.
8. S. R. Lehky and T. J. Sejnowski, Network model of shape from shading: Neural function arises from both receptive and projective fields, *Nature (London)* **333**, 1988, 452–454.
9. E. Mingolla and J. T. Todd, Perception of solid shape from shading, *Biol. Cybern.* **53**, 1983, 137–151.
10. J. T. Todd and E. Mingolla, Perception of surface curvature and direction of illumination from patterns of shading, *J. Exp. Psychol. Human Perception and Performance* **9**, 1983, 583–595.
11. H. B. Bühlhoff and H. A. Mallot, Integration of depth modules: Stereo and shading, *J. Opt. Soc. Amer. A* **5**, 1988, 1749–1758.
12. B. K. P. Horn, *On Lightness*, MIT AI Memo 295, 1974.
13. B. K. P. Horn and M. J. Brooks, *The Variational Approach to Shape from Shading*, MIT AI Memo 813, 1985.
14. D. Terzopoulos, Regularization of inverse problems involving discontinuities, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-8**, No. 4, 1986, 413–424.
15. A. P. Pentland, *Local Shading Analysis*, SRI Technical Note 272, 1982.
16. C. H. Lee and A. Rosenfeld, Improved methods for estimating shape from shading using the light source coordinate system, *Artif. Intell.* **26**, 1985, 125–143.
17. S. Geman and D. Geman, Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-6**, 1984, 721–741.
18. B. Widrow and M. E. Hoff, Adaptive switching circuits, in *1960 IRE WESCON Convention Record*, pp. 96–104.
19. T. Kohonen, *Associative Memory: A System-Theoretical Approach*, Springer-Verlag, Berlin, 1977.
20. J. A. Anderson, Cognitive and psychological computation with neural models, *IEEE Trans. Systems Cybern.* **SMC-13**, 1983, 799–815.
21. B. B. Mandelbrot, *Fractals: Form, Chance and Dimension*, Freeman, San Francisco, 1977.
22. R. F. Voss, Random fractal forgeries, in *Fundamental Algorithms for Computer Science* (R. A. Earnshaw, Ed.), pp. 805–829, Springer-Verlag, Berlin, 1985.
23. A. P. Pentland, Fractal-based description of natural scenes, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-6**, 1984, 661–673.
24. R. Szeliski, Regularization uses fractal priors, in *Proceedings, AAAI-87, Seattle, Washington*, 1987, pp. 749–754.
25. A. P. Pentland, Finding the illuminant direction, *J. Opt. Soc. Amer. A* **72**, 1981, 448–455.
26. D. C. Knill, Estimating illuminant direction and degree of surface relief, *J. Opt. Soc. Am. A*, in press.
27. R. T. Frankot and R. Chellappa, A method for enforcing integrability in shape from shading algorithms, in *Proceedings 1st Int. Conference on Computer Vision, London, England*, pp. 118–127.
28. A. P. Pentland, *Shape Information from Shading: A Theory about Human Perception*, MIT Vision Sciences Technical Report 103, 1988.
29. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, Learning internal representations by error propagation, in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, (D. E. Rumelhart and J. L. McClelland, (Eds.)), Vol. 1 MIT Press, Cambridge, MA, 1986.
30. C. Koch, J. Marroquin, and A. Yuille, Analog neuronal networks in early vision, *Proc. Natl. Acad. Sci. USA* **83**, 1986, 4623–4267.
31. A. Blake and A. Zimmerman, *Visual Reconstruction*, MIT Press, Cambridge, MA., 1987.