Computing Local Surface Orientation and Shape From Texture for Curved Surfaces

Jitendra Malik and Ruth Rosenholtz

Department of Electrical Engineering and Computer Science
University of California at Berkeley, Berkeley, CA 94720
email: malik@cs.berkeley.edu, rruth@robotics.eecs.berkeley.edu

Abstract

Shape from texture is best analyzed in two stages, analogous to stereopsis and structure from motion: (a) Computing the 'texture distortion' from the image, and (b) Interpreting the 'texture distortion' to infer the orientation and shape of the surface in the scene. We model the texture distortion for a given point and direction on the image plane as an affine transformation and derive the relationship between the parameters of this transformation and the shape parameters. We have developed a technique for estimating affine transforms between nearby image patches which is based on solving a system of linear constraints derived from a differential analysis. One need not explicitly identify texels or make restrictive assumptions about the nature of the texture such as isotropy. We use non-linear minimization of a least squares error criterion to recover the surface orientation (slant and tilt) and shape (principal curvatures and directions) based on the estimated affine transforms in a number of different directions. A simple linear algorithm based on singular value decomposition of the linear parts of the affine transforms provides the initial guess for the minimization procedure. Experimental results on both planar and curved surfaces under perspective projection demonstrate good estimates for both orientation and shape. A sensitivity analysis yields predictions for both computer vision algorithms and human perception of shape from texture.

1 Introduction

In its geometric essence, shape from texture is a cue to 3D shape very similar to binocular stereopsis and structure from motion. All of these cues are based on the information available in multiple perspective views of the same surface in the scene. In binocular stereopsis, the two eyes get slightly different views of the same surface; in structure from motion, the relative motion of the observer and the surface generates the different views. To put shape from texture in this framework, consider two nearby patches on a surface in the scene with same (or sufficiently similar) texture. The appearances of the two patches in a single monocular image will be slightly different because of the slightly different geometrical relationships that they have with respect to the observer's eye or camera. We thus get multiple views in a *single* image.

This naturally suggests a two stage framework (1) Computing the 'texture distortion' from the image, and (2) Interpreting the 'texture distortion' to infer the orientation and shape of the scene surface in 3D. The 'texture distortion' is the counterpart in texture analysis of binocular disparity in stereopsis or optical flow in structure from motion. We believe that the natural way to model the texture distortion *locally* is as a 2-D affine transformation between neighboring image patches. This affine transformation will depend on the direction and magnitude of the vector displacement between the two patches in the image. We will call the map which associates to each direction in the image plane an affine transformation, the *texture distortion map*. For each point on a smoothly curved textured surface this map is well defined and can be related to surface shape and orientation with respect to the viewer.

Traditionally, researchers have formalized the notion of texture distortion as that of finding the gradient of certain scalar-valued functions such as foreshortening, area, density, compression (minor axis) or scaling (major axis). We critically examine these and other approaches to the shape from texture problem in Section 2, where we will also explain our assumption of textural homogeneity.

In Section 3 we derive the relationship between the texture distortion map and surface orientation (slant and tilt) and shape (principal curvatures and directions) of the surface. Tilt is the direction, in the image, of the gradient of the distance to the surface. We refer to tilt as an angle from the positive x-axis. Slant is the angle between the surface normal at a point and the vector from the focal point to the point on the surface. We will explain the remaining shape parameters in Section 3. Our derivation of the relationship between the texture distortion and the surface parameters makes use of previous results due to Gårding [10].

A major motivation for using the texture distortion map formalism becomes evident in Section 4. It has proven difficult to develop algorithms for estimating the individual texture gradients which do not rely on either explicit texel identification (which is difficult for most natural textures) or on restrictive assumptions about the nature of the surface texture such as isotropy. On the other hand, we are able to develop an accurate and robust algorithm for estimating affine transformations between two nearby image patches which relies neither on texel identification nor on restrictive assumptions about the texture. The method uses least squares to solve a system of linear constraints on the parameters of the affine transformation. We derive these constraints from an assumption of stationarity under translations for the scene texture. The method bears strong resemblances to differential techniques for estimating optical flow [26].

In Section 5 of the paper, we develop a new algorithm for recovering surface orientation and shape based on the estimated affine transforms in a number of different directions. The method uses nonlinear minimization of a least squares error criterion to estimate the shape parameters. We use a simple linear algorithm based on singular value decomposition of the linear parts of the affine transforms to find the initial conditions for the minimization procedure. In Section 6 we present results on a number of examples of planar and curved surfaces. We know of no previous researchers who have demonstrated direct estimation of surface curvature parameters using shape from texture.

Our shape estimation algorithm is arguably optimal in a maximum likelihood sense if the measurement errors in the affine parameters can be assumed to be independent and normally distributed. By studying the Hessian of the error function at the minimum point, one can characterize the confidence intervals of the shape estimates. This also gives us an ideal observer which enables one to answer questions like: When the slant of a surface is increased, what happens to the uncertainty in slant and tilt? What is the effect of curvature? Field of view? In Section 7 we determine the confidence intervals on the shape estimates in Section 6 and present ideal observer predictions for shape from texture.

2 Relationship to previous work

Modern developments in shape from texture originate with the work of Gibson[12]. He coined the term *texture gradient* to describe the phenomenon in which neighboring surface patches which have identical, or sufficiently similar, texture in the scene project in the image plane to patches with different appearances because of the differences in distance, orientation, and shape of the surface patches with respect to the viewer. Gibson used the term 'gradient' to suggest a differential process. If we can measure this 'gradient,' we can use it to derive the local shape and orientation of the surface in the scene.

Subsequent research on shape from texture has followed two main lines: (1) Measurement and use of texture gradients (2) Statistical inference based on a probabilistic model of the texture.

Researchers have formalized the notion of texture gradient as that of finding the gradient of certain scalar-valued functions such as foreshortening, area, density, compression or scaling. The mathematical relationship between these gradients and scene geometry has been developed both for planar surfaces [24] and curved surfaces [10]. There are two difficulties in the use of these gradients:

1. Gårding has shown that these simple distortion gradients do not contain enough information for measurement of complete local surface curvature e.g. sign of Gaussian curvature.

2. It is not clear how these gradients could be measured in the image. Explicit texel identification as used by Blostein and Ahuja[5] is not feasible in many or most contexts. Lindeberg and Gårding[18] present a technique applicable for measuring area gradient, but that is inadequate by itself.

The other major family of approaches to shape from texture in the computer vision literature is based on starting with a probabilistic model of the texture. The problem of determining surface shape and orientation is treated as one of statistical inference–estimating the parameters of the model given sample measurements in the image–and can be approached in either a Bayesian or maximum likelihood framework. The model most often used is that of isotropy or weak isotropy of the texture [27, 8, 6, 4, 11]. Under projection, the texture will not generally appear isotropic, and thus they use the deviation from isotropy in the projection to infer 3D shape and orientation. There are two major weaknesses of such an approach:

1. It cannot deal with directional textures such as grass, fabrics etc. Furthermore, we cannot easily determine the appropriateness of the isotropy assumption (though see [4]).
2. It makes only partial use of available information, e.g. it does not exploit the change in size of the projected texture.

The other assumption which has been used in the literature is that of first order homogeneity, i.e. that the texture pattern has constant area or density[13, 1, 15, 25, 19]. This is a reasonable assumption for natural textures, and our first criticism does not apply. However, this assumption is too weak–it fails to exploit the systematic change in shape of the texture elements. For this reason, the use of a maximum likelihood estimator does not guarantee that we are making the best possible use of the data–the model has ignored an important and informative constraint in the physical situation. In some work [13, 1] the shape from texture problem is treated as locally underdetermined and a regularization assumption introduced to arrive at a shape estimate. We regard the use of a regularization term as inappropriate until the well-founded physical constraints have been fully exploited.

Our work is broadly in the texture gradient tradition initiated by Gibson. There are two significant points of departure:

1. Instead of measuring gradients of certain scalar functions, we model and measure the texture distortion *locally* as 2-D affine transformations between neighboring image patches. In the Section 3 we will relate the parameters of such an affine transformation to the five local shape and orientation parameters: slant, tilt, and three curvature parameters. In contrast, the traditional formulation using texture gradients lacks representational adequacy–Gårding showed that only two of the three curvature parameters can be recovered from the traditional texture gradients.

2. We introduce a new assumption about the surface texture, which is more powerful than the traditional first order homogeneity assumption, and yet appears to have broad physical validity (unlike isotropy). This assumption is basic to our being able to measure the affine transformations mentioned above.

We want to model the fact that the texture is the "same" at different points on the surface in the scene–not just the areas but also the shapes of the texels are approximately constant. While this implies periodicity for a deterministic pattern, for a texture which is best thought of as a realization of a stochastic process we can formalize this as stationarity under translations. The term *homogeneity* is used in the probability and statistics literature as equivalent to stationarity under translations. We can assume that not only the first but also the second (and higher) order statistics are translation-invariant. The notion of translation-invariance extends naturally from planar surfaces to surfaces of constant Gaussian curvature. This is intuitively obvious–one can slide around spherical triangles on a sphere without having to distort their shapes, just as one can slide around planar triangles on a planar surface. Technical justification may be found in Appendix A. Our model for shape from texture is based on a local analysis, and we will assume that over the scale of analysis, the Gaussian curvature of the surface does not change too much. This will enable us to bypass technical difficulties in extending the notion of textural homogeneity to surfaces of non-constant Gaussian curvature.

Previous use of homogeneity in the computer vision literature has made the weaker assumption of translation invariance for only the first order statistics (e.g. the fraction of surface area occupied by texels). We are able to exploit changes in the shape of the texture elements as well.

3 Relationship between the texture distortion map and 3D shape

We argued previously that instead of studying texture gradients, one should model texture distortion in a particular direction on the image plane as an affine transformation between a pair of image patches. This section will develop the relationship between the parameters of this affine transformation and the surface shape and pose.

We use perspective projection to a spherical image surface instead of to a planar surface. While there is a 1-1 mapping which relates the two kinds of perspective projection the relations which follow turn out to be simpler in the spherical case [13, 10]. Appendix B explains how to convert between the two forms of perspective projection.

This section has two parts. In the first part we review the formalism developed by Gårding[10]–essentially he defines an orthonormal frame field on the image sphere with one of the vectors in the tilt direction. The backprojection map takes on a particularly simple form, and Gårding obtains expressions for the different texture gradients for the general situation of smooth curved surfaces under perspective projection. In order to find an expression for the affine transformation

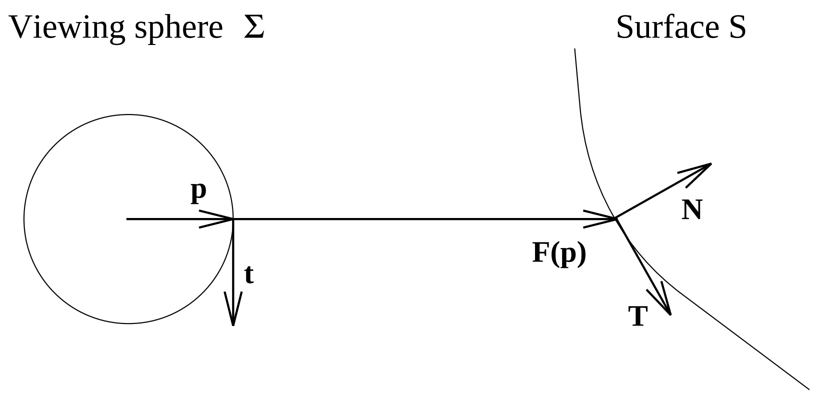


Figure 1: **Local surface geometry.**

between a pair of image patches, we need an expression for the change in the tilt angle between two neighboring points on the image sphere in terms of the shape parameters. We also need an expression for the change in the vector on the surface formed by backprojecting the tilt vector. We also derive these expressions in the first part of this section.

In the second part of the section the motivation for the work done in the first part will become more clear, as we exploit the frame field defined by Gårding to derive an expression for the affine transformation on the image sphere which relates two neighboring image patches.

3.1 The slant-tilt frame field

This section is based on Gårding [10], to which the reader is referred for proofs of the various assertions. Relevant differential geometry concepts may be found in O'Neill [21] or Koenderink [16].

The basic geometry is illustrated in Figure 1. A smooth surface S is mapped by central projection to a unit sphere Σ centered at the focal point. The backprojection map F from Σ to S is defined as $F(\mathbf{p}) = \mathbf{r}(\mathbf{p}) = r(\mathbf{p})\mathbf{p}$ where $\mathbf{p}$ is a unit vector from the focal point to a point on the image sphere, and $r(\mathbf{p})$ is the distance along the visual ray from the focal point through $\mathbf{p}$ to the corresponding point $\mathbf{r} = F(\mathbf{p})$ on the surface S. We consider this map for regions of the surface where the map is not singular by excluding neighborhoods containing the occluding contour. The differential of the backprojection map F_* maps tangent vectors of Σ at $\mathbf{p}$ to tangent vectors of S at $F(\mathbf{p})$.

Define the tilt direction $\mathbf{t}$ in $T_p(\Sigma)$, the tangent plane of the viewing sphere at $\mathbf{p}$, to be a unit vector in the direction of the gradient of the distance function $r(\mathbf{p})$, and the auxiliary vector $\mathbf{b} = \mathbf{p} \times \mathbf{t}$. Then $(\mathbf{t}, \mathbf{b})$ form an orthonormal basis for the tangent plane to the image sphere Σ and together with $\mathbf{p}$ constitute an orthonormal frame field on Σ. Garding shows that $\mathbf{t}$ and $\mathbf{b}$ backproject to orthogonal vectors $F_*(\mathbf{t}) = r'\mathbf{p} + r\mathbf{t}$ and $F_*(\mathbf{b}) = r\mathbf{b}$ in the tangent space $T_{F(p)}(S)$. Dividing these vectors by their lengths gives us an orthonormal basis $(\mathbf{T}, \mathbf{B})$ of the tangent space of the surface at $F(\mathbf{p})$. The vectors $\mathbf{T}, \mathbf{B}$ along with the unit normal to the surface $\mathbf{N} = \mathbf{T} \times \mathbf{B}$ constitute an orthonormal frame field on the surface. The slant angle σ is defined to be the angle between the surface normal $\mathbf{N}$ and the viewing direction $\mathbf{p}$, so that $\cos\sigma = \mathbf{N}.\mathbf{p}$. $F_*(\mathbf{p})$ can be expressed very conveniently in terms of the $(\mathbf{t}, \mathbf{b})$ and $(\mathbf{T}, \mathbf{B})$ bases as

$$F_*(\mathbf{p}) = \begin{bmatrix} r/\cos\sigma & 0 \\ 0 & r \end{bmatrix} = \begin{bmatrix} \frac{1}{m_p} & 0 \\ 0 & \frac{1}{M_p} \end{bmatrix} \tag{1}$$

where r is the distance to the object from the center of the viewing sphere, and σ is the slant angle. We see that $m_p = \cos\sigma/r$ is the scaling of the texture pattern in the "minor axis," i.e. in the tilt direction, due to projection, and $M_p = 1/r$ is the scaling in the "major axis," i.e. in the orthogonal direction. The use of the terms "major axis" and "minor axis" is intended as a mnemonic – a texture composed of circles on a planar surface would project to ellipses with minor axes aligned with the tilt direction.

The shape of the surface is captured in the *shape operator*, which measures how the surface normal $\mathbf{N}$ changes as one moves in various directions in the tangent space of the surface $T_{F(p)}(S)$. One can represent the shape operator in the $(\mathbf{T}, \mathbf{B})$ basis as

$$\begin{bmatrix} -\nabla_T \mathbf{N} \\ -\nabla_B \mathbf{N} \end{bmatrix} (\mathbf{p}) = \begin{bmatrix} \kappa_t & \tau \\ \tau & \kappa_b \end{bmatrix} \begin{bmatrix} \mathbf{T} \\ \mathbf{B} \end{bmatrix} \tag{2}$$

where κ_t is the normal curvature in the $\mathbf{T}$ direction, κ_b the normal curvature in the $\mathbf{B}$ direction and τ is the geodesic torsion. The determinant of the operator gives the Gaussian curvature $K = \kappa_t \kappa_b - \tau^2$, and half the trace is the mean curvature $H = (\kappa_t + \kappa_b)/2$.

Gårding goes on to obtain expressions for the derivatives of the $(\mathbf{p}, \mathbf{t}, \mathbf{b})$ frame field on Σ expressed in terms of the frame field itself. One can also define the derivatives of the frame field $(\mathbf{N}, \mathbf{T}, \mathbf{B})$ on S with respect to $(\mathbf{p}, \mathbf{t}, \mathbf{b})$ by first pulling back these fields from the surface S to Σ. We reproduce here the most relevant formulas:

$$\begin{aligned}
\nabla_{\mathbf{t}} \mathbf{N} &= -(\frac{r}{\cos\sigma})(\kappa_t \sin\sigma \mathbf{p} + \kappa_t \cos\sigma \mathbf{t} + \tau \mathbf{b}) && (3) \\
\nabla_{\mathbf{b}} \mathbf{N} &= -r(\tau \sin\sigma \mathbf{p} + \tau \cos\sigma \mathbf{t} + \kappa_b \mathbf{b}) && (4) \\
\nabla_{\mathbf{t}} \mathbf{t} &= -\mathbf{p} + \frac{r\tau}{\cos\sigma \sin\sigma} \mathbf{b} && (5) \\
\nabla_{\mathbf{b}} \mathbf{t} &= (\frac{1}{\sin\sigma})(\cos\sigma + r\kappa_b) \mathbf{b} && (6) \\
\nabla_{\mathbf{t}} \mathbf{b} &= -\frac{r\tau}{\cos\sigma \sin\sigma} \mathbf{t} && (7) \\
\nabla_{\mathbf{b}} \mathbf{b} &= -\mathbf{p} - \frac{1}{\sin\sigma}(\cos\sigma + r\kappa_b) \mathbf{t} && (8)
\end{aligned}$$

Using the linearity of the derivative, we can compute $\nabla_{\mathbf{v}} \mathbf{t}$ for an arbitrary vector $\mathbf{v} = \Delta t \mathbf{t} + \Delta b \mathbf{b}$ in the tangent space.

$$\nabla_{\mathbf{v}} \mathbf{t} = -\Delta t\, \mathbf{p} + \frac{r\tau}{\cos\sigma \sin\sigma} \Delta t\, \mathbf{b} + (\frac{1}{\sin\sigma})(\cos\sigma + r\kappa_b) \Delta b\, \mathbf{b} \tag{9}$$

In the next section, we will also need the derivative of the $\mathbf{T}$ vector field. Since Gårding does not derive any formula for this, we shall do so here.

Since $\mathbf{T} = \mathbf{B} \times \mathbf{N} = \mathbf{b} \times \mathbf{N}$, we have

$$\nabla_{\mathbf{v}}\mathbf{T} = \nabla_{\mathbf{v}}(\mathbf{b} \times \mathbf{N}) = (\nabla_{\mathbf{v}}\mathbf{b}) \times \mathbf{N} + \mathbf{b} \times (\nabla_{\mathbf{v}}\mathbf{N}) \tag{10}$$

Letting $\mathbf{v} = \mathbf{t}$ in the above expression and substituting from equations 7 and 3, we get

$$\begin{aligned}
\nabla_{\mathbf{t}}\mathbf{T} &= (\nabla_{\mathbf{t}}\mathbf{b}) \times \mathbf{N} + \mathbf{b} \times (\nabla_{\mathbf{t}}\mathbf{N}) \\
&= (-\frac{r\tau}{\cos\sigma \sin\sigma}\mathbf{t}) \times (\cos\sigma\mathbf{p} - \sin\sigma\mathbf{t}) - \mathbf{b} \times (\frac{r}{\cos\sigma})(\kappa_t \sin\sigma\mathbf{p} + \kappa_t \cos\sigma\mathbf{t} + \tau\mathbf{b}) \\
&= r\kappa_t\mathbf{p} - r\kappa_t \tan\sigma\mathbf{t} + \frac{r\tau}{\sin\sigma}\mathbf{b} \\
&= \frac{r\kappa_t}{\cos\sigma}(\cos\sigma\mathbf{p} - \sin\sigma\mathbf{t}) + \frac{r\tau}{\sin\sigma}\mathbf{b} \\
&= \frac{r\kappa_t}{\cos\sigma}\mathbf{N} + \frac{r\tau}{\sin\sigma}\mathbf{B}
\end{aligned}$$

Similarly, letting $\mathbf{v} = \mathbf{b}$ in expression 10 and substituting from equations 8 and 4, we get

$$\begin{aligned}
\nabla_{\mathbf{b}}\mathbf{T} &= (\nabla_{\mathbf{b}}\mathbf{b}) \times \mathbf{N} + \mathbf{b} \times (\nabla_{\mathbf{b}}\mathbf{N}) \\
&= (-\mathbf{p} - \frac{(\cos\sigma + r\kappa_b)}{\sin\sigma}\mathbf{t}) \times (\cos\sigma\mathbf{p} - \sin\sigma\mathbf{t}) - \mathbf{b} \times r(\tau\sin\sigma\mathbf{p} + \tau\cos\sigma\mathbf{t} + \kappa_b\mathbf{b}) \\
&= r\tau\cos\sigma\mathbf{p} - r\tau\sin\sigma\mathbf{t} + (\sin\sigma + \cos\sigma\cot\sigma + r\kappa_b\cot\sigma)\mathbf{b} \\
&= r\tau(\cos\sigma\mathbf{p} - \sin\sigma\mathbf{t}) + (\sin\sigma + \cos\sigma\cot\sigma + r\kappa_b\cot\sigma)\mathbf{b} \\
&= r\tau\mathbf{N} + (\sin\sigma + \cos\sigma\cot\sigma + r\kappa_b\cot\sigma)\mathbf{B}
\end{aligned}$$

Using the linearity of the derivative, we can compute $\nabla_{\mathbf{v}}\mathbf{T}$ for an arbitrary vector $\mathbf{v} = \Delta t\mathbf{t} + \Delta b\mathbf{b}$ in the tangent space.

$$\nabla_{\mathbf{v}}\mathbf{T} = (\frac{r\kappa_t}{\cos\sigma}\Delta t + r\tau\Delta b)\mathbf{N} + (\frac{r\tau}{\sin\sigma})\Delta t\,\mathbf{B} + (\sin\sigma + \cos\sigma\cot\sigma + r\kappa_b\cot\sigma)\Delta b\,\mathbf{B} \tag{11}$$

3.2 Affine transformations on the image sphere

Figure 2 depicts the situation. We wish to find the matrix, A, which represents the affine transformation between the spherically projected texture at point $\mathbf{p}_1$ and the projected texture at a nearby point $\mathbf{p}_2$. This matrix will be a function of the local orientation and shape parameters. The orientation parameters are σ, the slant of the surface and $\mathbf{t}$, the direction of tilt of the surface. The shape parameters are $r\kappa_t$, $r\kappa_b$, and $r\tau$. The variable r is the distance from the center of the viewing sphere to the given point on the surface. Note that the inseparability of the distance, r, and the curvature parameters is inherent to the problem. The image of a surface S at distance r is indistinguishable from that of a k scaled copy (for which the curvatures will be $1/k$ of S) at a distance of kr.

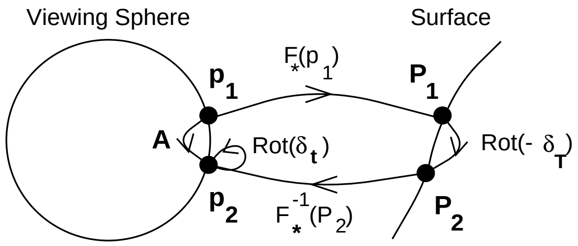


Figure 2: **Determining the affine transformation, A, between the texture at point $\mathbf{p}_1$ and the texture at point $\mathbf{p}_2$.** $F_*(\cdot)$ is the backprojection map from the viewsphere to the surface. The texture is constant over the image, so the map between $\mathbf{P}_1$ and $\mathbf{P}_2$, the corresponding points on the surface, is just a rotation between the $(\mathbf{T}, \mathbf{B})$ bases at the two points. $Rot(\delta_t)$ rotates between the $(\mathbf{t}, \mathbf{b})$ bases at the points $\mathbf{p}_1$ and $\mathbf{p}_2$. $A = \mathrm{Rot}(\delta_t)F_*^{-1}(\mathbf{P}_2)\mathrm{Rot}(-\delta_T)F_*(\mathbf{p}_1)$.

Our analysis is a differential analysis. We will freely assume that $\mathbf{p_2} - \mathbf{p_1}$ can be modelled as a vector $\mathbf{v} = \Delta t\, \mathbf{t} + \Delta b\, \mathbf{b}$ in the tangent space at the point $\mathbf{p_1}$ and the expressions derived will be true in the limit as $\Delta t, \Delta b \rightarrow 0$.

To find the affine transformation, we first backproject from the point $\mathbf{p_1}$ on the viewsphere to the corresponding point $\mathbf{P_1}$ on the surface, using the map $\mathbf{F}_*(\mathbf{p_1})$. Using the basis $(\mathbf{t_1}, \mathbf{b_1})$ on the tangent plane of the image sphere Σ, and $(\mathbf{T_1}, \mathbf{B_1})$ on the tangent plane on the surface S, this map can be represented as

$$F_*(\mathbf{p}) = \begin{bmatrix} r/\cos\sigma & 0 \\ 0 & r \end{bmatrix} = \begin{bmatrix} \frac{1}{m_1} & 0 \\ 0 & \frac{1}{M_1} \end{bmatrix} \tag{12}$$

Recall that m_1 is the scaling of the texture pattern in the "minor axis," i.e. in the tilt direction, due to projection, and M_1 is the scaling in the "major axis."

Our assumption of textural homogeneity enables us (see Appendix A) to model the transformation between the tangent spaces at points $\mathbf{P_1}$ and $\mathbf{P_2}$ as a rotation, by some angle δ_T, between the two bases $(\mathbf{T}_1, \mathbf{B}_1)$ and $(\mathbf{T}_2, \mathbf{B}_2)$.

To find the rotation angle between the bases at points $\mathbf{P_1}$ and $\mathbf{P_2}$, we begin by noting that $\mathbf{T_2} = \mathbf{T_1} + \nabla_\mathbf{v}\mathbf{T}$ to first order, and hence from equation 11

$$\mathbf{T_2} = \mathbf{T_1} + (\frac{r\kappa_t}{\cos\sigma}\Delta t + r\tau\Delta b)\mathbf{N_1} + (\frac{r\tau}{\sin\sigma})\Delta t\,\mathbf{B_1} + (\sin\sigma + \cos\sigma\cot\sigma + r\kappa_b\cot\sigma)\Delta b\,\mathbf{B_1} \tag{13}$$

In the right hand side of this equation, the term in the direction of the surface normal $\mathbf{N_1}$ represents a change of the plane of the frame as a whole, and the terms in the direction of $\mathbf{B_1}$ represent a rotation about the surface normal on S. Since $\mathbf{T}, \mathbf{B}$ are unit vectors, we see that

$$\delta_T = (\frac{r\tau}{\sin\sigma})\Delta t + (\sin\sigma + \cos\sigma\cot\sigma + r\kappa_b\cot\sigma)\Delta b \tag{14}$$

as $\Delta t, \Delta b \rightarrow 0$. Note that if the $\mathbf{T}, \mathbf{B}$ basis vectors undergo counterclockwise rotation by δ_T, the texture on the surface undergoes rotation by $-\delta_T$.

Next, we project back onto the viewing sphere, using the matrix $F_*^{-1}(\mathbf{P_2})$. This puts us back on the viewing sphere, but in the $(\mathbf{t_2}, \mathbf{b_2})$ basis, not the original $(\mathbf{t_1}, \mathbf{b_1})$ basis. We must convert between these bases by rotating by the angle between the tilt vectors, δ_t. As in the case of the $\mathbf{T}$ vector field, to find this angle we begin by noting that $\mathbf{t_2} = \mathbf{t_1} + \nabla_{\mathbf{v}}\mathbf{t}$ to first order. Hence from equation 9 we get

$$\mathbf{t_2} = \mathbf{t_1} - \Delta t\,\mathbf{p_1} + \frac{r\tau}{\cos\sigma\sin\sigma}\Delta t\,\mathbf{b_1} + (\frac{1}{\sin\sigma})(\cos\sigma + r\kappa_b)\Delta b\,\mathbf{b_1} \tag{15}$$

As before, the term in the direction of $\mathbf{p_1}$ represents a change of the plane of the frame as a whole, and the term in the direction of $\mathbf{b_1}$ represents a rotation of the frame about the normal. We obtain

$$\delta_t = \frac{r\tau}{\cos\sigma\sin\sigma}\Delta t + (\frac{1}{\sin\sigma})(\cos\sigma + r\kappa_b)\Delta b \tag{16}$$

as $\Delta t, \Delta b \to 0$. Thus we have

$$\begin{aligned} A &= \mathrm{Rot}(\delta_t)F_*^{-1}(\mathbf{P}_2)\mathrm{Rot}(-\delta_T)F_*(\mathbf{p}_1) & (17)\\ &= \mathrm{Rot}(\delta_t)\cdot\begin{bmatrix} \cos\delta_T\frac{m_2}{m_1} & \sin\delta_T\frac{m_2}{M_1} \\ -\sin\delta_T\frac{M_2}{m_1} & \cos\delta_T\frac{M_2}{M_1}\end{bmatrix} & (18)\end{aligned}$$

Gårding showed that the normalized gradients of the minor and major axis scale factors, in the $(\mathbf{t}, \mathbf{b})$ basis, are

$$\begin{aligned} \frac{\nabla m}{m} &= -\tan\sigma\begin{bmatrix} 2 + r\kappa_t/\cos\sigma \\ r\tau\end{bmatrix} & (19)\\ \frac{\nabla M}{M} &= -\tan\sigma\begin{bmatrix} 1 \\ 0\end{bmatrix} & (20)\end{aligned}$$

We note that

$$\begin{aligned} m_2 &= m_1 + \nabla m \circ (\Delta t \quad \Delta b)^T \\ M_2 &= M_1 + \nabla M \circ (\Delta t \quad \Delta b)^T \end{aligned}$$

to first order where $\mathbf{p_2} - \mathbf{p_1} = (\Delta t \quad \Delta b)^T$ is the step between the points $\mathbf{p}_1$ and $\mathbf{p}_2$ in the image. Using this equation and equations 18 and 1, we get

$$A = \mathrm{Rot}(\delta_t)\cdot\begin{bmatrix} k_m\cos\delta_T & k_m\sin\delta_T\cos\sigma \\ -k_M\frac{\sin\delta_T}{\cos\sigma} & k_M\cos\delta_T\end{bmatrix}$$

where

$$\begin{aligned} k_m &= 1 + \frac{\nabla m}{m}\circ\begin{bmatrix}\Delta t \\ \Delta b\end{bmatrix} \\ k_M &= 1 + \frac{\nabla M}{M}\circ\begin{bmatrix}\Delta t \\ \Delta b\end{bmatrix} & (21)\end{aligned}$$

The actual affine transformation we find between the two points will not, in general, be in terms of the $(\mathbf{t}, \mathbf{b})$ basis. Instead, our matrix will be related by a change of basis: $\hat{A} = UAU^{-1}$. The change of basis matrix, U, rotates the standard basis on the image sphere to the $(\mathbf{t}, \mathbf{b})$ basis and is given by

$$\mathrm{Rot}(\theta_t) = \begin{bmatrix} \cos\theta_t & -\sin\theta_t \\ \sin\theta_t & \cos\theta_t \end{bmatrix}$$

where θ_t is the tilt angle at point $\mathbf{p}_1$. The standard basis on the image sphere is given by the E_1, E_2 frame field (see Appendix B).

The analysis above assumes that we have spherical projection. In reality, cameras use planar projection. Appendix B converts between the two types of projection.

4 Estimating affine transforms

In order to estimate the shape and orientation of a textured surface, we must first estimate the texture distortion in a number of directions in the image plane, modeled as a set of affine transforms. In this section, we present a method for finding the affine transform between a pair of image patches.

Rather than estimating the transform in the space domain, we estimate it in the frequency domain. We can do this because [20] if G is the Fourier transform of g, which we write as $G = \mathcal{F}(g)$, then

$$\mathcal{F}[g(A\mathbf{x})] = \frac{1}{\mid A \mid} G(A^{-T}\omega)$$

Thus if we can find the affine transformation in the frequency domain, we can find the affine tranformation in the space domain. The advantage of working in the frequency domain for texture analysis was first noted by Bajcsy and Leiberman [2]: If we keep only the magnitude of the frequency response, our algorithm will be insensitive to small changes of position, since a small change in position causes only a change in phase. Otherwise, we would have to make sure that the patches were centered over 'corresponding points', a notion which is not even properly defined for stochastic textures. We will demonstrate this invariance to small changes in position with an example in Section 6.

We create spectrograms of a pair of image patches in the following way:

1. **Extract image patches:** Extract a square patch from the image, centered at each of the chosen points.
2. **Preprocess the patches:** First normalize each patch, and subtract its mean. This makes the spectrograms largely invariant to shading and contrast differences, and invariant

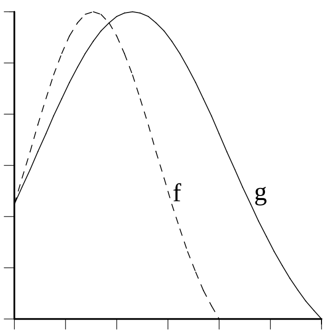


Figure 3: **Finding the affine transformation in 1D:** Suppose $g(\omega) = f(a\omega)$. Then for small $a = 1 + \Delta a$, we can write $g(\omega) - f(\omega) \approx \frac{df}{d\omega}\Delta a\omega$ and solve for the unknown, a.

to foreground/background reversals. Next window the patch using a Welch windowing function[22]. Finally, remove the best fitting plane from the patch (known as *detrending*). This reduces the effect of shading variations across the patch.

3. **Fourier transform:** Pad the patch with zeros, and take the Fourier Transform Magnitude (FTM) of the patch. Padding with zeros interpolates between samples in the Fourier transform.

4. **Extract low frequency components:** Extract a low frequency region from the FTM and use it as the spectrogram. Most of the information tends to be in the low frequencies in the FTM. It is important that, whatever criteria we use to window the frequencies in the FTM, the spectrograms all look qualitatively the same, i.e. have the same number of peaks. We may use any such reasonable criterion. For this paper, we chose a window size such that all of the spectrograms have magnitude less than 50% of their maximum for all points outside of the window.

5. **Normalize the spectrograms.**

We use a differential method to solve for the affine transformation between the spectrograms of two patches. To illustrate the concept, we first consider the 1-D case, shown in Figure 3. In 1-D, the differential method works as follows. Suppose we have two curves, f and g, such that they are related by an affine transformation. In 1-D, one curve is a scaled version of the other:

$$g(\omega) = f(a\omega)$$

Then suppose we write the scaling factor, a, as $1 + \Delta a$, where we assume that Δa is small. We know from calculus that

$$f(\omega + \Delta\omega) \approx f(\omega) + \frac{df}{d\omega}(\omega) \cdot \Delta\omega$$

In this case, $\Delta\omega$ equals $\Delta a \cdot \omega$, so we write

$$f(a\omega) \approx f(\omega) + \frac{df}{d\omega}(\omega) \cdot \Delta a\omega$$

Substituting $g(\omega) = f(a\omega)$, we get

$$g(\omega) - f(\omega) \approx \frac{df}{d\omega}(\omega) \cdot \Delta a \omega \tag{22}$$

If we are given f and g, we can solve for the affine transformation between the two curves by directly solving for Δa.

To extend the idea to two dimensions, we assume that one spectrogram differs from another by only a 2×2 affine transformation $A = I + \Delta A$. Then the 2-D analog of equation 22 is

$$F_2(\vec{\omega}) - F_1(\vec{\omega}) \approx \vec{\nabla} F_1 \circ \Delta A \vec{\omega}$$

where F_i is the spectrogram for the ith point, $\vec{\omega}$ is frequency, and $\vec{\nabla} F_1$ is the gradient of the spectrogram, at that frequency. The only unknown in this equation is ΔA, thus, for a 2×2 transformation matrix, this gives us one linear equation in four unknowns. If we write the elements of ΔA as $a_{i,j}$, we can write the equation as:

$$\begin{bmatrix} \frac{\partial F_1}{\partial \omega_x}\omega_x & \frac{\partial F_1}{\partial \omega_x}\omega_y & \frac{\partial F_1}{\partial \omega_y}\omega_x & \frac{\partial F_1}{\partial \omega_y}\omega_y \end{bmatrix} \begin{bmatrix} a_{1,1} \\ a_{1,2} \\ a_{2,1} \\ a_{2,2} \end{bmatrix} = F_2(\vec{\omega}) - F_1(\vec{\omega})$$

where each of the partial derivatives is evaluated at $\vec{\omega}$. Each frequency sample point in the spectrogram gives us a new equation, resulting in the matrix equation: $\mathbf{D}\vec{a} = \vec{b}$. We can solve this overdetermined system of linear equations to obtain the best (in the least-squares sense) estimate for the elements of ΔA , (see e.g. [22]. Note that we are assuming that the texture is 'rich' enough so that we have enough independent constraints on the parameters of the affine transformation. If we have four significant Fourier components with different orientations, this will be true. Otherwise, we have the equivalent of the 'aperture problem' in the motion domain [26].

The differential method is in essence one step of the familiar Newton-Raphson iterative procedure for minimizing nonlinear functions, where each step is taken along the gradient of the function. This suggests that we can refine our estimate of A by warping the first spectrogram using the current estimate, and then finding the new, (hopefully smaller) ΔA. We exploit this in our implementation–the iterations are terminated as soon as the error between the transformed first spectrogram and the second spectrogram stops decreasing. In our experiments, we find that the number of iterations is small, typically 4 or 5. Usually, the result after the first iteration is good enough–in a pinch, we could live with this non-iterative algorithm.

We used two heuristics for reducing the effect of noise and rejecting outliers in the spectrogram:

1. Reject points in the spectrogram with magnitude smaller than 10% of the maximum. These points typically correspond to noise. The choice of this threshold should be based on the signal-to-noise ratio.

2. Reject outliers, using a reweighted least squares technique from robust statistics [23]. In this method, we obtain an initial estimate, $\vec{a_0}$, for $\vec{a}$, and calculate the resulting residual, $\vec{r} = \mathbf{D}\vec{a_0} - \vec{b}$. We expect a higher residual at the peaks of the spectrogram than in the low areas, so we weight the residual by $\frac{1}{F_1}$. Then we reject as outliers any points with weighted residual, $\hat{r}$, greater than 2.5 times $\hat{\sigma} = 1.4826\sqrt{\mathrm{median}(\hat{r_i}^2)}$.

Some previous researchers have also attempted to estimate affine transforms or shape parameters in Fourier domain. Bajcsy and Lieberman [2] attempted to estimate the affine transformation between f and g by finding the peak in each function, and taking the ratio of the locations of those peaks. Lindeberg and Gårding [18], and Super and Bovik [25] attempt to find the surface parameters by finding second moments in the Fourier domain, and then comparing those moments. By contrast, our method uses the entire spectrogram, rather than only the peaks or the moments, to estimate the affine transformation and ultimately the shape parameters. Krumm and Shafer [17] also find the affine transformation by using the entire spectrogram, but they find it by testing every possible combination of slant and tilt in a discrete set, and choosing the one which gives them the smallest error. By contrast, the differential method allows us to solve for the affine transformation directly.

It may be noted that our affine transform estimation procedure can be applied to other problems in vision, e.g. in the context of stereopsis; see Jones and Malik[14].

5 Shape Recovery Algorithms

To estimate the *texture distortion map* at a point p, we find the spectrograms for that point and for neighboring points in a number of different directions, $\vec{v_i} = (\Delta t_i \quad \Delta b_i)^T$, around p and get estimates, $\bar{A_i}$, of the affine transforms, $\hat{A_i}$, for each of these directions using the algorithm developed in the previous section. In this section, we develop two algorithms for recovering surface orientation (slant and tilt) and shape (principal curvatures and directions). In Section 6 we will present experimental results on a number of images of planar and curved surfaces. In Section 7 we will examine the sensitivity analysis associated with our second algorithm, and discuss predictions, based on this sensitivity analysis, for the performance of other shape from texture methods, including human perception of shape from texture.

Recall (Section 3) that $\hat{A_i} = UA_iU^{-1}$, where $U = \mathrm{Rot}(\theta_t)$ is the matrix for transforming coordinates from the standard basis to $(\mathbf{t}, \mathbf{b})$, and

$$A_i = \mathrm{Rot}(\delta_t^i) \begin{bmatrix} k_m^i \cos\delta_T^i & k_m^i \sin\delta_T^i \cos\sigma \\ -k_M^i \frac{\sin\delta_T^i}{\cos\sigma} & k_M^i \cos\delta_T^i \end{bmatrix}$$

where

$$
\begin{aligned}
k_m^i &= 1 + \tfrac{\nabla m}{m} \circ \begin{bmatrix} \Delta t_i \\ \Delta b_i \end{bmatrix} &= 1 - \tan\sigma \begin{bmatrix} 2 + r\kappa_t/\cos\sigma \\ r\tau \end{bmatrix}^T \begin{bmatrix} \Delta t_i \\ \Delta b_i \end{bmatrix} \\
k_M^i &= 1 + \tfrac{\nabla M}{M} \circ \begin{bmatrix} \Delta t_i \\ \Delta b_i \end{bmatrix} &= 1 - \tan\sigma \begin{bmatrix} 1 \\ 0 \end{bmatrix}^T \begin{bmatrix} \Delta t_i \\ \Delta b_i \end{bmatrix}
\end{aligned}
$$

and

$$
\begin{aligned}
\delta_T^i &= (\frac{r\tau}{\sin\sigma})\Delta t_i + (\sin\sigma + \cos\sigma\cot\sigma + r\kappa_b\cot\sigma)\Delta b_i \\
\delta_t^i &= \frac{r\tau}{\cos\sigma\sin\sigma}\Delta t_i + (\frac{1}{\sin\sigma})(\cos\sigma + r\kappa_b)\Delta b_i
\end{aligned}
$$

There are five unknowns, the slant σ, the tilt direction specified by θ_t, and the three shape parameters $(r\kappa_t, r\kappa_b, r\tau)$. Each estimation of an affine transform in an image direction $\vec{v_i} = (\Delta t_i \quad \Delta b_i)^T$ yields us four nonlinear equations. Simple equation counting tells us that one direction is not enough, and generically two directions ought to be sufficient.

We present here two shape recovery algorithms. The first algorithm is a linear algorithm based on singular value decomposition (SVD) of the $\bar{A}_i$ matrices. It requires knowledge of the affine transforms in a minimum of two directions, exploiting more when they are available, to give us estimates of $(\sigma, \theta_t, r\kappa_t, r\tau)$. The second algorithm is based on a least squares formulation of an error criterion. It permits recovery of $r\kappa_b$ as well. Since the error function is nonlinear in the parameters, it has to be minimized by gradient descent or some variation thereof. The first algorithm gives us a good initial guess. The second algorithm has an associated sensitivity analysis, so we can provide confidence intervals for the orientation and shape estimates.

5.1 Algorithm based on SVD of the A matrices

The three stages of the algorithm are

1. Estimate k_m^i, k_M^i as the singular values of $\bar{A}_i$ where s_1^i, s_2^i are the singular values of $\bar{A}_i$. To justify this, we begin by noting that the singular values s_1, s_2 of matrix $\hat{A}_i$ are related to the eigenvalues λ_1, λ_2 of the matrix $\hat{A}_i^T\hat{A}_i$ by the relationship $\lambda_1 = s_1^2$ and $\lambda_2 = s_2^2$. Let us now compute expressions for trace$(\hat{A}_i^T\hat{A}_i) =$ trace$(A_i^T A_i)$ and det$(\hat{A}_i^T\hat{A}_i) =$ det$(A_i^T A_i)$. Dropping the sub- and superscripts i to avoid unnecessary clutter,

$$
s_1^2 + s_2^2 = \mathrm{trace}(A^T A) = k_m^2\cos^2\delta_T + k_M^2\frac{\sin^2\delta_T}{\cos^2\sigma} + k_m^2\sin^2\delta_T\cos^2\sigma + k_M^2\cos^2\delta_T \qquad (23)
$$

and

$$
s_1^2 s_2^2 = \det(A_i^T A_i) = k_m^2 k_M^2 \qquad (24)
$$

from which we see that

$$
\begin{aligned}
(s_1+s_2)^2 &= \mathrm{trace}(A^TA) + 2\sqrt{\det(A^TA)} \\
&= k_m^2\cos^2\delta_T + k_M^2\frac{\sin^2\delta_T}{\cos^2\sigma} + k_m^2\sin^2\delta_T\cos^2\sigma + k_M^2\cos^2\delta_T + 2k_mk_M(\cos^2\delta_T+\sin^2\delta_T) \\
&= (k_m+k_M)^2 - k_m^2\sin^2\delta_T(1-\cos^2\sigma) + +k_M^2\sin^2\delta_T(\frac{1}{\cos^2\sigma}-1) \\
&= (k_m+k_M)^2 - (k_m\sin\delta_T\sin\sigma)^2 + (k_M\sin\delta_T\tan\sigma)^2
\end{aligned}
$$

As the step in the image plane $\Delta t_i, \Delta b_i \to 0$, $k_m, k_M \to 1$ while $\sin\delta_t \approx \delta_t \to 0$. This suggests that from the above expression, we can approximate $s_1+s_2 \approx k_m + k_M$. In conjunction with the expression $s_1s_2 = k_mk_M$, this implies that we can estimate k_m and k_M as the singular values of $\bar{A}$. We must decide which singular value corresponds to k_m and which to k_M. We initially make the heuristic assumption that k_M is the singular value closer to one. This amounts to assuming that the magnitude of change along the major axis is smaller than the change along the minor axis, and that δ_T is small. This assumption will be true for any planar surface, and for many curved surfaces with positive curvature and torsion. More generally, we can try both cases and choose the one which gives us shape parameters which best agree with our affine transformations.

2. Estimate the tilt direction and $\tan\sigma$ from the major axis gradient. To do this, note from Eqn 20 and Eqn 21 that $k_M^i - 1 = -\tan\sigma\Delta t_i$. We don't know Δt_i, because we do not yet know the tilt direction. Let $(t_x \quad t_y)^T$ be the normalized major axis gradient vector, $\frac{\nabla M}{M}$, in the standard basis, (x,y). This vector is in the tilt direction, and has length $|-\tan\sigma|$. Then
$$
\begin{bmatrix} \Delta x_1 & \Delta y_1 \\ \Delta x_2 & \Delta y_2 \\ . & . \\ \Delta x_n & \Delta y_n \end{bmatrix} \cdot \begin{bmatrix} t_x \\ t_y \end{bmatrix} = \begin{bmatrix} k_M^1 - 1 \\ k_M^2 - 1 \\ . \\ k_M^n - 1 \end{bmatrix}
$$
We can solve this overdetermined linear system for the least-squares estimate of $(t_x \quad t_y)^T$, giving us both the tilt direction and the slant σ.

3. Estimate the surface curvature from the minor axis gradient. To do this, note from Eqn 19 and Eqn 21 that
$$
k_m^i - 1 \quad = \quad -\tan\sigma \begin{bmatrix} 2 + r\kappa_t/\cos\sigma \\ r\tau \end{bmatrix} \circ \begin{bmatrix} \Delta t_i \\ \Delta b_i \end{bmatrix}
$$
Since the tilt direction has been previously estimated we now know the steps, $(\Delta t_i \quad \Delta b_i)$, taken in the different directions. So we have an overdetermined linear system which can be solved to estimate $r\kappa_t$ and $r\tau$.

Note that for the systems of linear equations in steps 2 and 3 to be solvable, two independent directions are necessary and sufficient.

Complete information about local surface shape requires knowledge of three parameters, and here we have only found two: $r\kappa_t$ and $r\tau$; the third parameter, $r\kappa_b$, is left undetermined. This was to be expected as this algorithm is essentially based on factorizing the affine transform matrices to obtain the major and minor axis gradients–we know from Gårding that these underspecify shape.

5.2 Algorithm based on least squares formulation

The second algorithm is based on finding the orientation and shape parameters which minimize the sum of squared errors between the predicted and empirically measured entries of the affine transformation matrices, i.e., we wish to minimize the following error function:

$$\chi^2(\sigma, \theta_t, r\kappa_t, r\kappa_b, r\tau) = \sum_{i=1}^{n} \sum_{k=1}^{2} \sum_{l=1}^{2} (\hat{A}_i(k,l) - \bar{A}_i(k,l))^2$$

where $\hat{A}_i(k,l)$ the (k,l)th element of the theoretically predicted matrix $\hat{A}_i$ and is a function of the shape parameters, and $\bar{A}_i(k,l)$ is the (k,l)th element of the empirically measured affine transform matrix $\bar{A}_i$. Ideally, each term in this error sum should be weighted by the inverse of the standard deviation of the measurement error of that particular entry. A specific characterization of the probability distribution of the measurement errors in the entries of the affine transform matrices A_i is not yet available. We expect it to depend on the particular algorithm used for the estimation of the affine transforms. In the absence of a particularly appropriate model, we will proceed on the (convenient!) assumption that these errors are independent and normally distributed with standard deviation Δa.

For minimizing the error function, we just used the gradient descent routine in the *Mathematica* package–there are any of a number of variants such as conjugate gradient that could have been used equivalently. The starting point is provided by the orientation and shape estimates returned by the first algorithm. Since the first algorithm does not estimate $r\kappa_b$, we arbitrarily set $r\kappa_b = r\kappa_t$.

6 Experimental Results

In this section we show the results of our algorithms on a number of synthetic and real images. Synthetic images are useful because, since we know the ground truth, they allow us to better judge the accuracy of our methods. Table 1 shows the results of the first algorithm on nine synthetic images. For each of these images, we found the affine transform in eight different directions around a given point of the image. Each of these images was created by mapping Brodatz textures on various surfaces. The image marked "jitter" is the same image as that marked "wire2", but we have jittered the positions of our "corresponding points," by an average

Image	*True*				*Estimated*			
	σ	tilt	$r\kappa_t$	$r\tau$	σ	tilt	$r\kappa_t$	$r\tau$
wire1	69	180	0	0	66	169	-.05	.12
wire2	64	-25	0	0	56	-36	.79	1.2
jitter	64	-25	0	0	55	-38	.86	1.4
noise	64	-25	0	0	57	-33	.76	.95
cane	69	-90	0	0	58	-91	.95	-.04
straw	64	-90	0	0	49	-106	-.14	1.7
cy[illegible]	[illegible]	[illegible]	[illegible]	0	[illegible]	[illegible]	[illegible]	[illegible]3
sp[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]6
sp[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]3

Table 1: **T[illegible] [illegible]ethod 1**

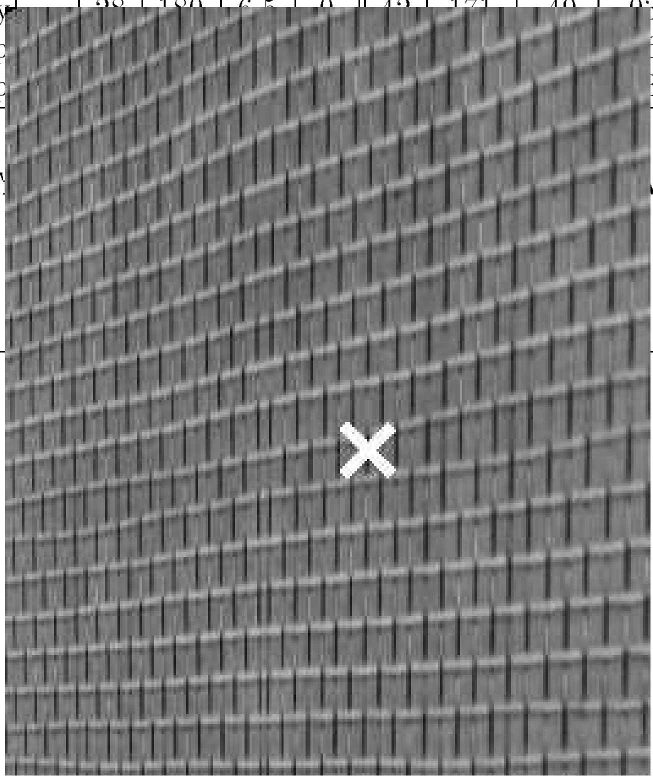


Figure 4: **Planar wire texture, slant=**70°**, tilt=**180°**, "wire1".**

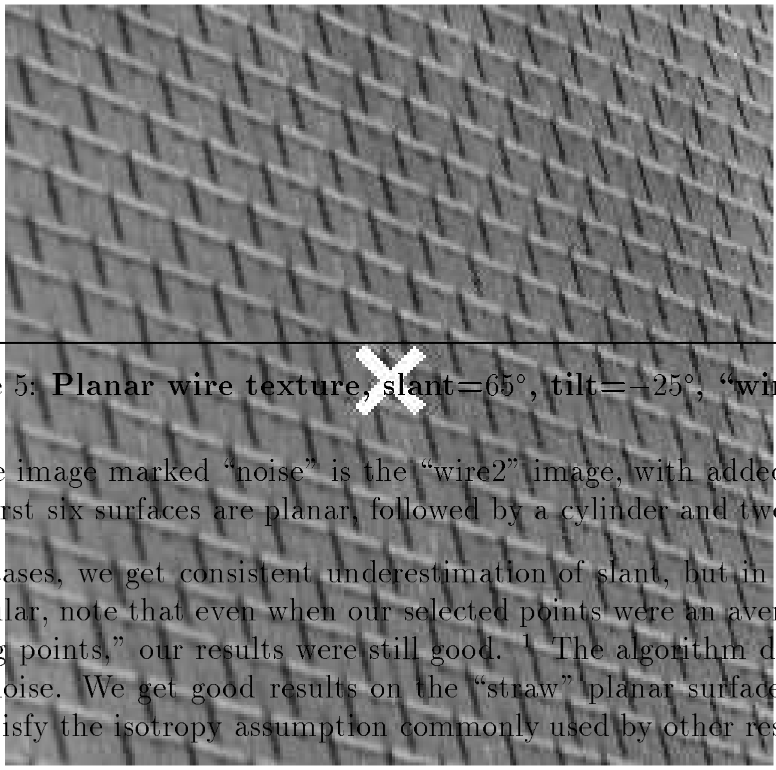


Figure 5: **Planar wire texture, slant=65°, tilt=−25°, "wire2".**

of seven pixels. The image marked "noise" is the "wire2" image, with added noise of standard deviation 30. The first six surfaces are planar, followed by a cylinder and two spheres.

For the planar cases, we get consistent underestimation of slant, but in general the results are good. In particular, note that even when our selected points were an average of seven pixels from "corresponding points," our results were still good. [1] The algorithm does quite well even in the presence of noise. We get good results on the "straw" planar surface, even though this texture does not satisfy the isotropy assumption commonly used by other researchers.

The tilt and some of the slant estimates for the curved surfaces were also reasonable, but the curvature estimates were not accurate enough to be usable. If this algorithm were to be used for shape estimation by itself, the recommended strategy would be to obtain slant and tilt estimates at a large number of points and then fit a smooth surface consistent with these estimates. Differentiating this fitted surface yields the desired shape parameters.

However a better approach is to use our second algorithm which gives much more accurate orientation and shape estimates *locally* without any need for a surface fitting stage.

Table 2 shows the results of our second algorithm on a number of synthetic examples. We get improved slant and tilt estimates, and also significantly better estimates of the curvature

[1] We in no way mean to imply that seven pixels is the limit of our shift invariance. For this particular image, a shift of much more than seven pixels simply puts one closer to a different texel.

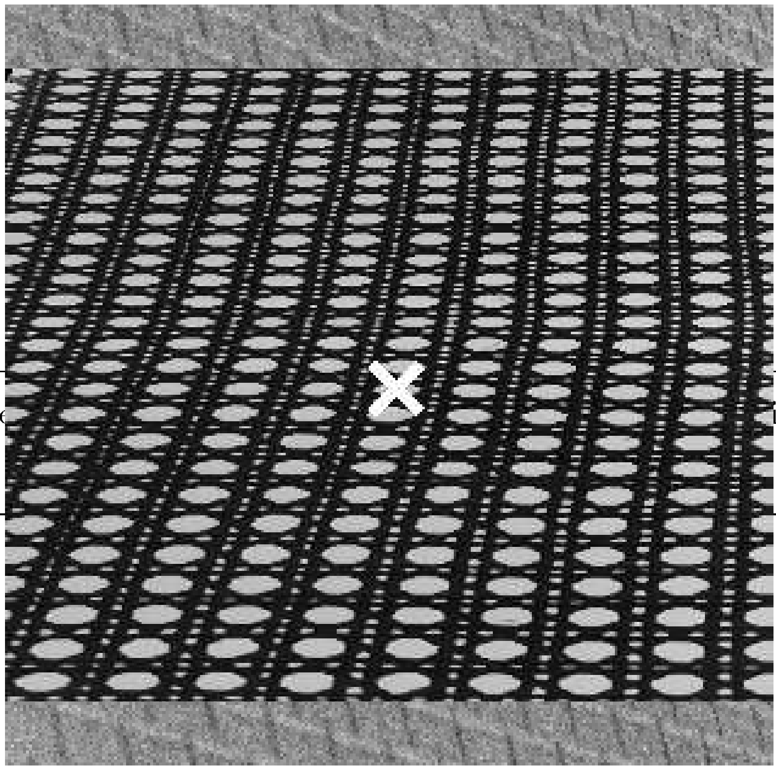


Figure [illegible] **ise”.**

Figure 7: **Planar cane texture, slant=65°, tilt=−90°, “cane”.**

Figure **raw”.**

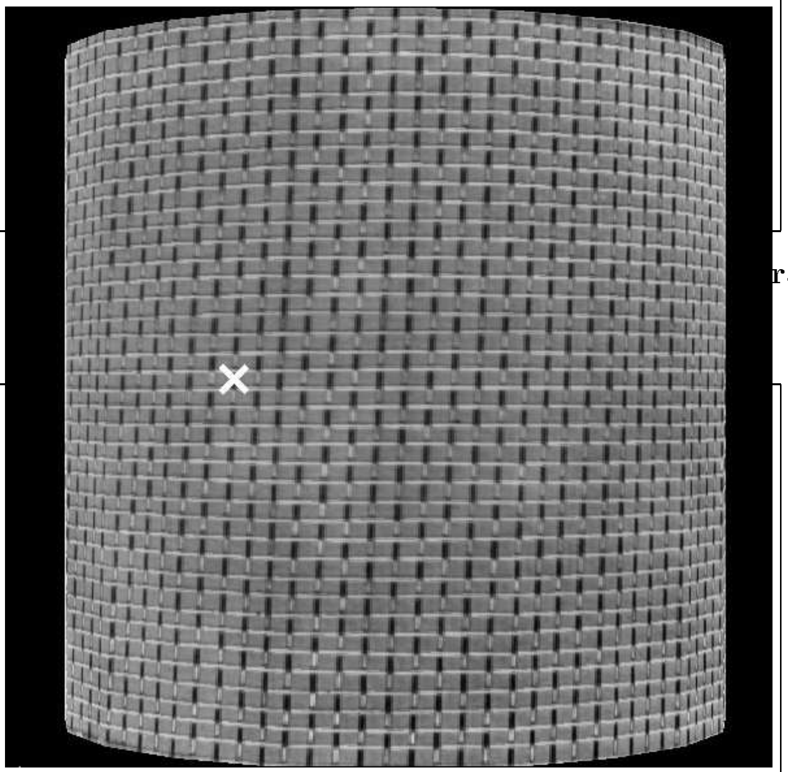


Figure 9: **Wire texture on cylinder, “cyl”.**

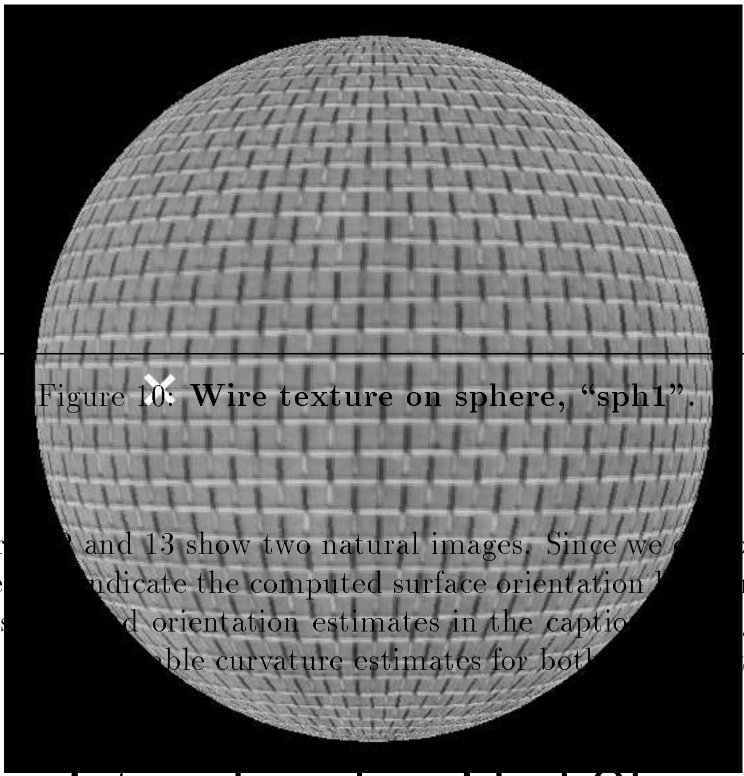


Figure 10: **Wire texture on sphere, “sph1”.**

parameters.

In addition, Figur[illegible] and 13 show two natural images. Since we [illegible]t have ground truth for the natural image[illegible]ndicate the computed surface orientation [illegible]rojected circle in the image, and give the s[illegible]d orientation estimates in the captio[illegible]get quite reasonable orientation estimates [illegible]ble curvature estimates for bot[illegible]ural images.

7 Confidence Intervals and an Ideal Observer

Our second algorithm allows us to easily determine confidence intervals on the orientation and shape estimates. For more details on the methodology that we follow, the reader is referred to [22], Chapters 14.4 and 14.5. In the latter part of this section we will demonstrate how one may use confidence intervals to predict the performance of either a computer algorithm or a human observer at the shape from texture task.

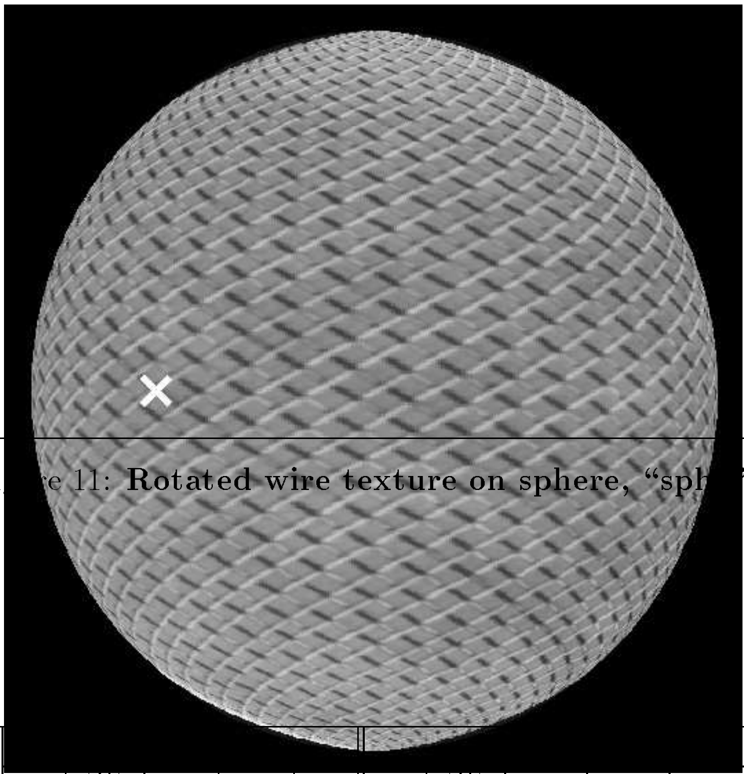
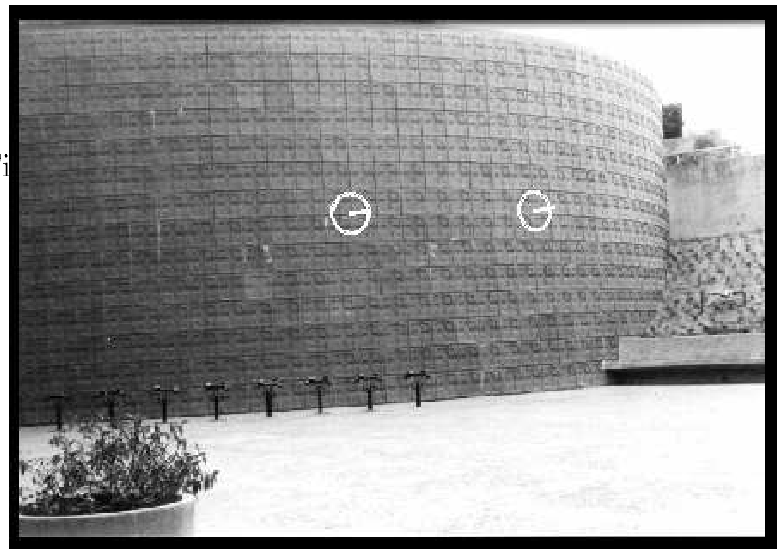


Figure 11: **Rotated wire texture on sphere, "sph[illegible]".**

Image	σ	tilt	$r\kappa_t$	$r\kappa_b$	$r\tau$	σ	tilt	$r\kappa_t$	$r\kappa_b$	$r\tau$	χ^2_{min}
cane	69	-90	0	0	0	71	-92	-.06	.00	-.03	.031
wire2	64	-25	0	0	0	63	-20	.22	.00	-.03	.027
noise	64	-25	0	0	0	61	-18	.32	-.01	-.10	.027
straw	64	-90	0	0	0	60	-73	-.13	.29	-.27	.010
cyl	28	180	6.5	0	0	26	180	3.2	-.13	-.15	.003
sph1	39	180	6.6	6.6	0	40	179	6.0	5.8	.67	.002
sph2	40	180	6.6	6.6	0	35	182	8.0	5.6	.95	.002

Table 2: **True and Estimated Surface Parameters: Method 2**



Left pt: $\sigma = 19$, $\theta_t = -8$, $r\kappa_t = 0.53$,$r\kappa_b = -0.12$, $r\tau = 0.00$.
Right pt: $\sigma = 40$, $\theta_t = -11$, $r\kappa_t = 2.9$, $r\kappa_b = -0.03$, $r\tau = 0.58$

Figure 13: Lecture hall image, with empirical results shown for two points.

7.1 Confidence Intervals on the Shape Estimates

Let us abbreviate the five geometrical parameters $(\sigma, \theta_t, r\kappa_t, r\kappa_b, r\tau)$ as $g_i, i = 1, 2, \ldots 5$. The gradient of the error function, χ^2, with respect to the parameters $\mathbf{g}$ will be zero at the χ^2 minimum. To obtain confidence intervals on the parameters, one computes the so-called *curvature matrix* $[\alpha]$ which is defined as half of the Hessian of the χ^2 function

$$\alpha_{kl} = \frac{1}{2\Delta a^2} \frac{\partial^2 \chi^2}{\partial g_k \partial g_l}$$

Since we have five geometrical shape parameters g_i, this is a 5×5 symmetric matrix. The inverse of the curvature matrix is the covariance matrix, C, of the fit, on the assumption of normally distributed errors. In that case the standard errors in the parameters are given by $\sqrt{C_{ii}}$. The confidence interval for parameter g_i, $\pm\delta g_i$ is $\pm\sqrt{C_{ii}}$ for 68 percent confidence, $\pm 2\sqrt{C_{ii}}$ for 95 percent confidence.

As mentioned above, if we assume that the errors in the entries of the affine transformation matrices A_i are independent and identically normally distributed, and if we have an estimate for the standard deviation of these errors, we can give confidence intervals for the estimates of the shape parameters. In Figures 14-18 we show the 68 percent confidence intervals for the parameters, assuming[2] a measurement error Δa of standard deviation 0.0323. Using this value for the standard deviation, 66 percent of the estimated parameters fall within the 68 percent confidence intervals.

7.2 Ideal observer predictions

In addition to obtaining confidence intervals for our empirical estimates of shape parameters, we can use the theory outlined in the previous section to develop an ideal observer model for shape from texture. Roughly speaking, an ideal observer gives us a prediction for the best performance one expects out of any estimator, given the visual information and the measurement error. As such it is of interest both for computer vision shape from texture algorithms and for predicting the performance of the human visual system. In the context of shape from texture models based on discrete texels using isotropy and first order homogeneity assumptions, such ideal observers have been developed by Blake et al[3].

The uncertainty in the shape estimates depends upon the measurement errors in the earlier stages of processing; here, in the estimation of the affine transforms. The errors in the affine transforms will of course depend on the texture being viewed. For example, if the texture is a realization of a Poisson process then we expect that for sparser textures it will be more difficult to estimate the affine transforms accurately. A second example would be that of a sinewave grating texture–clearly we can expect to recover only some components of the affine

[2] We computed this value as the standard deviation of the errors in the affine transformation matrices for the examples used in this paper. The errors are known because we know the ground truth in these synthetic examples.

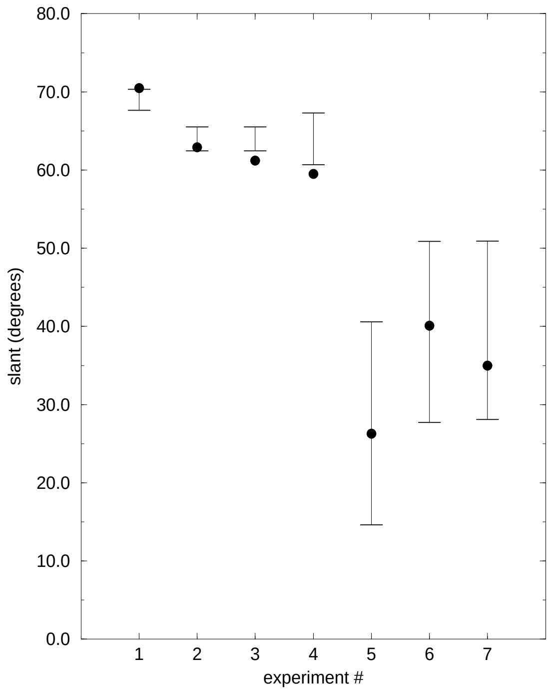


Figure 14: **Slant estimates with confidence intervals.**

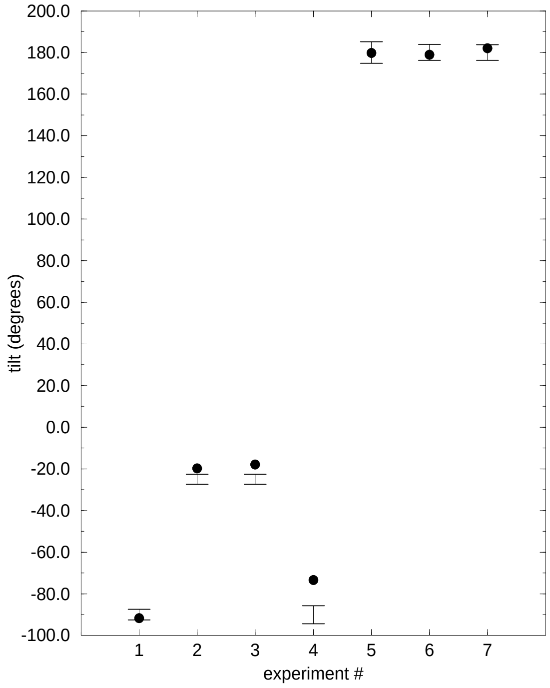


Figure 15: **Tilt estimates with confidence intervals.**

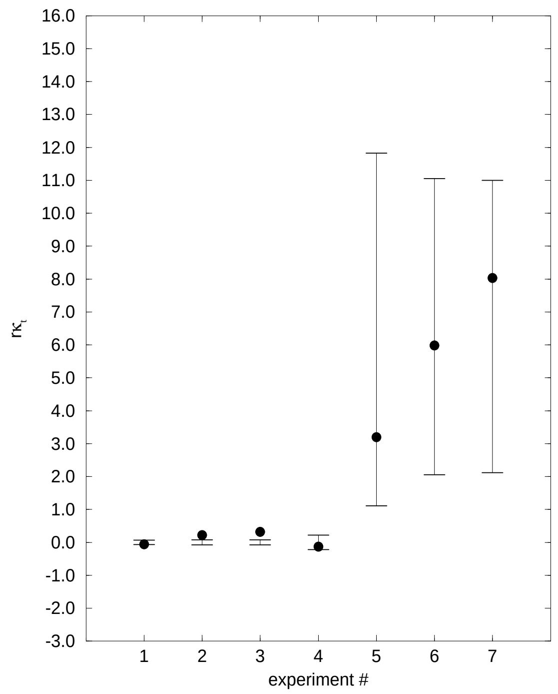


Figure 16: $r\kappa_t$ **estimates with confidence intervals.**

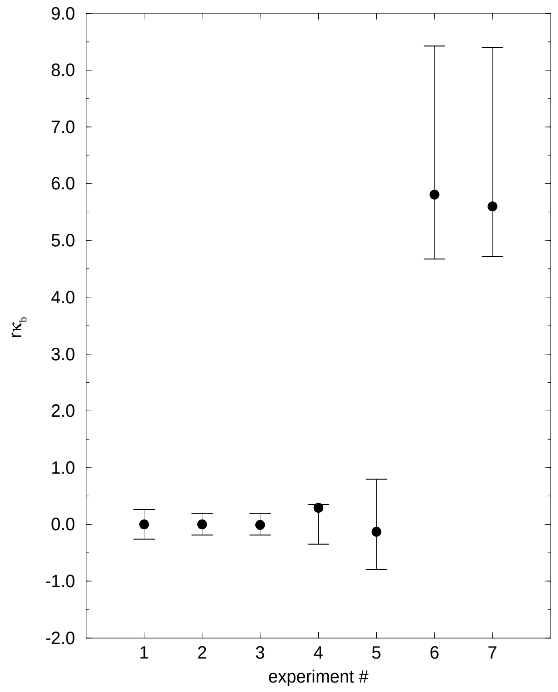


Figure 17: $r\kappa_b$ **estimates with confidence intervals.**

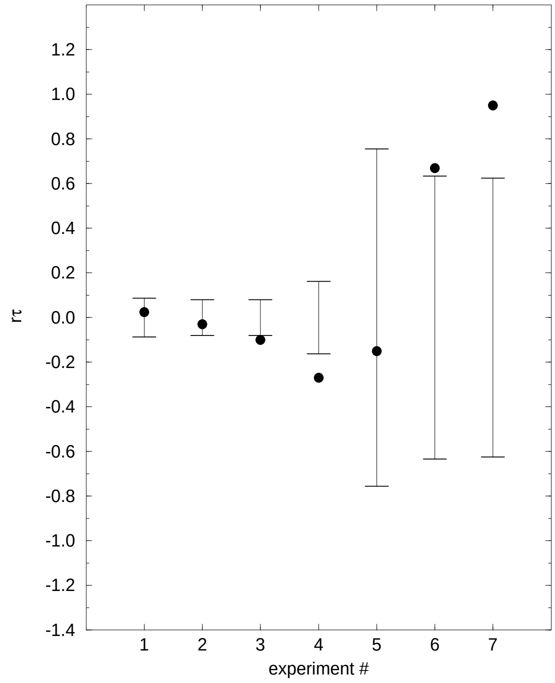


Figure 18: $r\tau$ **estimates with confidence intervals.**

transforms. If we know the prior distribution of the texture on the surface (except perhaps for a few unknown parameters), we can expect to analytically characterize the uncertainties in the affine transform estimates. This would be in the spirit of the work of Blake et al. We however favor a different approach–one should try to estimate the uncertainty of the affine transform estimates as a byproduct of the affine transform estimation process itself, and avoid making strong assumptions about the texture distribution. This approach is similar to what would be considered 'natural' for stereopsis or optical flow.

Without prior knowledge of the distribution of the texture, what can we say about the distribution of errors in the affine transforms? A simple choice would be to assume, as we did in the last section, that the errors in the affine transform parameters are independent and identically distributed normal random variables. While clearly this assumption would be incorrect for, e.g., highly anisotropic texture, it gave us very reasonable results for the range of textures in the previous section. One may think of the situation as follows: we have a number of surfaces of different shapes and orientations, all with similar textures which satisfy the above assumption, and we wish to know whether shape estimation is more difficult for some of these shapes than others. As above, we will find confidence intervals for the shape parameters for a number of these hypothetical shapes. The confidence intervals give us a measure of the uncertainty in the shape estimates.

We present here the results of several ideal observer "experiments." Since the confidence intervals will depend on the measurement error in the elements of the affine transformations, we give the confidence intervals in terms of *relative units*. To obtain the actual expected error one

$r\kappa_b$	0	1	2	3	4	5	6
$r\kappa_t$	6	5	4	3	2	1	0
$r\tau$	0	$\sqrt{5}$	$\sqrt{8}$	$\sqrt{9}$	$\sqrt{8}$	$\sqrt{5}$	0

Table 3: **Relationships between the curvature parameters for the experiment in Figure 22.**

would multiply these values by the standard deviation of the measurement error. [3]

In the first experiment in Figure 19, we varied the slant of a planar surface, and plot the confidence intervals for slant and tilt estimates. We see that we expect improved tilt estimates, and somewhat improved slant estimates, for higher slants. Blake, et al[3] report similar results for their ideal observer based on compression gradient.

In the second experiment, in Figure 20, we varied the field of view *(f.o.v.)* for both a plane and a sphere, and again present the confidence intervals for slant and tilt estimates. We see in these plots that one expects better slant and tilt estimates for larger f.o.v.

In Figure 21 we show the confidence intervals for surfaces with varying $r\kappa_t$ and $r\kappa_b$, for which tilt is aligned along one of the principal directions, i.e. $r\tau = 0$. We see that slant estimates are best for planar or near planar surfaces, whereas tilt estimates are worst for near planar surfaces. There also appear to be some asymmetries: slant estimates are worse for positive $r\kappa_t$ than negative, and tilt estimates are worse for negative $r\kappa_b$ than positive, at least for small $r\kappa_t$.

In the fourth experiment, presented in Figure 22, we rotate a cylindrical surface about its surface normal at a point, thus keeping the shape (i.e. principal curvatures) constant, but varying the direction of tilt with respect to the surface, and thus varying $r\kappa_t$, $r\kappa_b$, and $r\tau$. We show the relationship between the values of $r\kappa_t$, $r\kappa_b$, and $r\tau$ in Table 3. We see, roughly, that estimates are worst for larger $r\tau$ i.e. when the tilt direction is in between the two principal directions.

In the final experiment, we varied the slant of a cylindrical surface, keeping the tilt perpendicular to the axis of the cylinder. This amounts to computing shape estimates for a series of points along the circumference of the cylinder. In Figure 23 we plot the confidence intervals for the estimate of $r\kappa_t$. Note that for smaller slants we expect more error in the curvature parameter. This may explain much of the error in $r\kappa_t$ for both our synthetic cylinder example and for the first point in the real cylinder example.

These observations suggest several lines of psychophysical investigation.

[3] Unless otherwise specified, the field of view used was quite small to insure that our differential assumptions held: 0.15 degrees. We found no qualitative difference for larger fields of view, though of course the confidence intervals are larger for smaller fields of view.

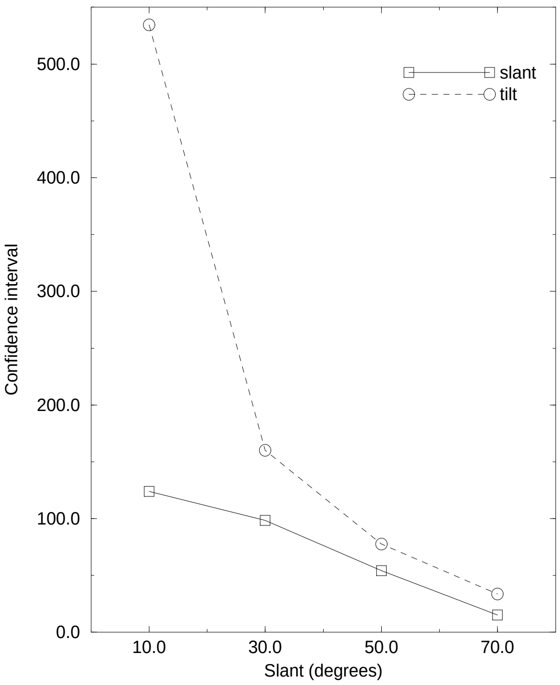
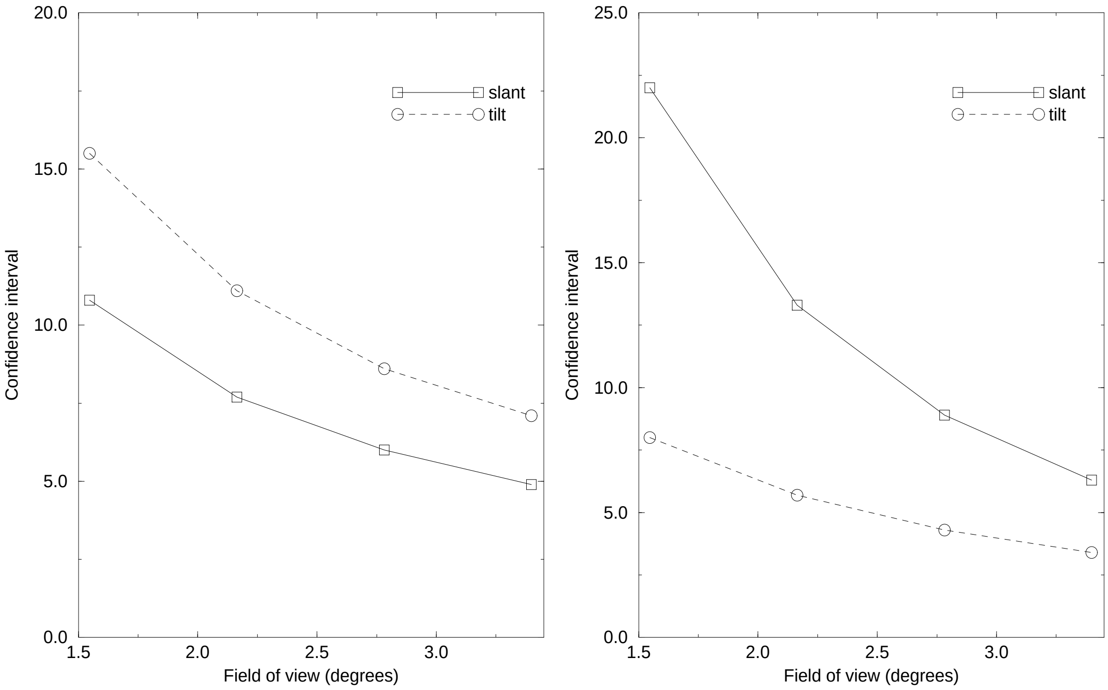
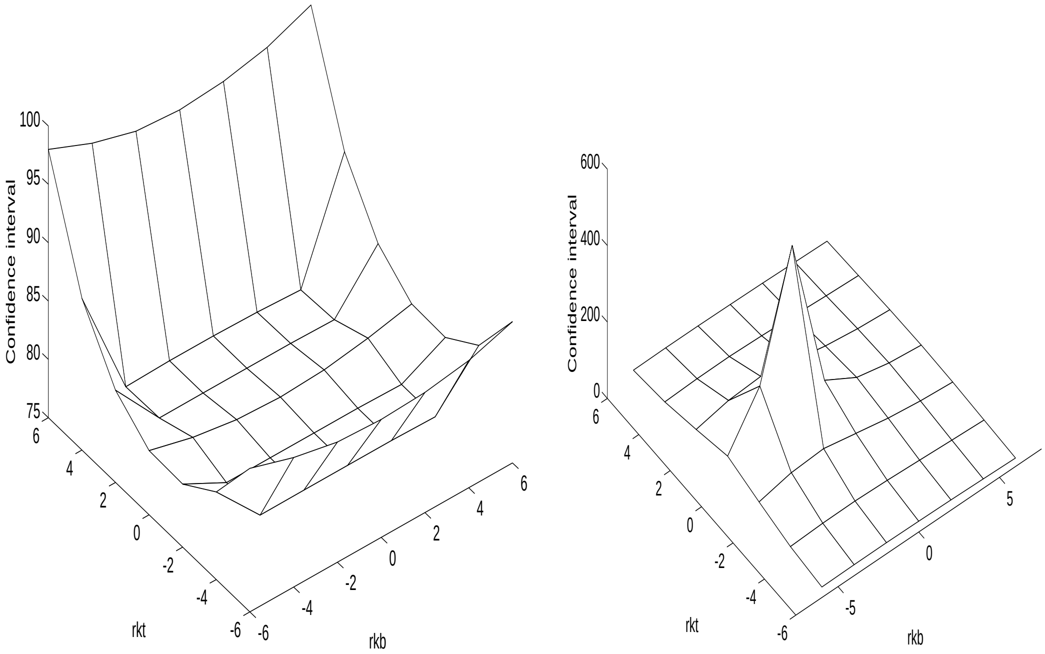


Figure 19: **Confidence intervals for slant and tilt as a function of slant for a planar surface.**



Planar surface Spherical surface

Figure 20: **Confidence intervals for slant and tilt as a function of f .o.v. angle.**



Slant Tilt

Figure 21: **Confidence intervals for slant and tilt as a function of $r\kappa_t$ and $r\kappa_b$ $(r\tau = 0)$.**

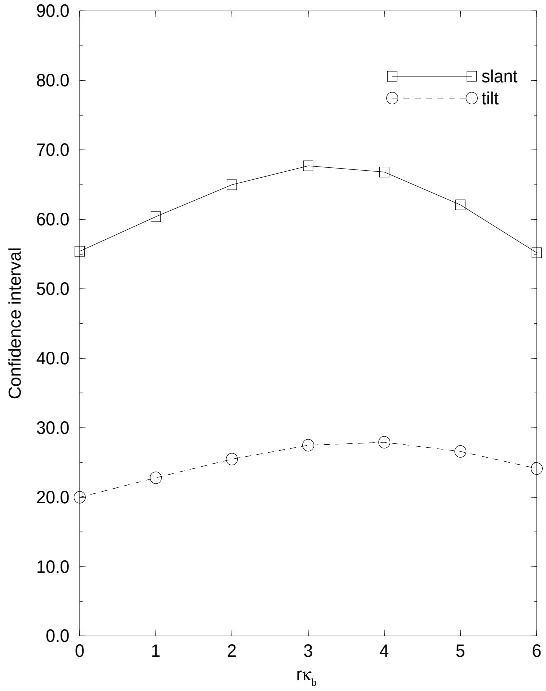


Figure 22: **Confidence intervals for slant and tilt as $r\kappa_t$, $r\kappa_b$, and $r\tau$ vary, keeping principal curvatures constant.**

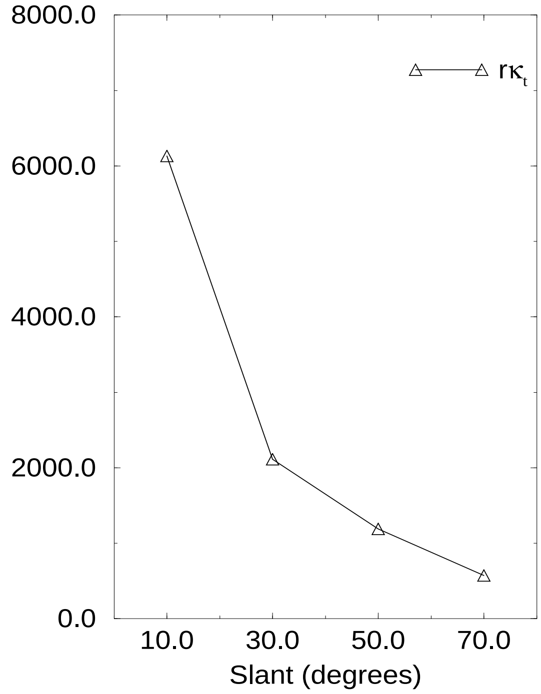


Figure 23: **Confidence intervals for $r\kappa_t$ as a function of slant, for a cylinder.**

Conclusion

In conclusion, we have presented a method for finding the shape of surfaces locally from texture distortion, modeled as a set of affine transforms in different directions in the image. The advantage of this representation is that it captures *all* the information available locally and does so without any restrictive assumptions. We develop a differential technique for estimating the affine transforms, which can be applied to a number of other vision problems. Our results demonstrate that local shape-from-texture without any *a priori* assumptions on the texture is a viable module for early vision. Our second shape recovery algorithm has an associated sensitivity analysis, which both allows us to find confidence intervals for our estimated parameters and serves as an ideal observer model that can make predictions about human estimates of shape from texture.

This research was supported by NSF PYI grant IRI-8957274, Xerox, JSEP contract F49620-93-C-0014, and Rosenbaum fellowship from the Newton Institute. We thank Phil Lapsley, Pietro Perona and Sanjay Tiwari for useful discussions, and John Oliensis for catching an error in an earlier version.

A Textural homogeneity for curved surfaces

As was pointed out by Helmholtz back in the last century, the notion of free mobility of figures–sliding figures around on a surface without distorting the shapes of figures–is equally valid for surfaces of zero Gaussian curvature e.g. planes, cyliders, surfaces of constant positive Gaussian curvature e.g. spheres and surfaces of constant negative Gaussian curvature. Rigorous results were obtained by Cartan who showed that in some sense, the Riemannian metric is determined locally by the curvature. We refer the reader to [9], pp. 156-159 for the proof and the following corollary;

> **Corollary 2.3** *Let M be a space of constant gaussian curvature and let p and q be any two points of M. Let $\{e_j\}$ and $\{f_j\}$ be arbitrary orthonormal bases of the tangent spaces $T_p(M)$ and $T_q(M)$, respectively. Then there exist neighborhoods U of p and V of q, and an isometry $g : U \to V$ such that $dg_p(e_j) = f_j$, where dg_p is the differential of the map g at p.*

The isometry g can be used to slide figures around on a surface without having to distort their shapes or having to leave the surface. In this setting the notion of texture homogeneity for a patch of surface is quite naturally defined. The derivative of this isometry gives us the mapping between the orthonormal bases $(\mathbf{T}_1, \mathbf{B}_1)$ and $(\mathbf{T}_2, \mathbf{B}_2)$ of the tangent spaces at points $\mathbf{P_1}$ and $\mathbf{P_2}$ respectively that we use in Section 3. This is just a rotation by angle δ_T.

For surfaces which do not have constant Gaussian curvature, the concept of a texture pattern which is homogeneous over the surface becomes somewhat ill-defined. The pattern cannot be isometrically related at two distinct points $\mathbf{P_1}$ and $\mathbf{P_2}$. To proceed further, we believe that some additional assumptions about the physical processes which generated the texture are necessary. We do not pursue this line of inquiry in this paper.

While our model is rigorously valid only for surfaces of constant Gaussian curvature, we can extend it heuristically to surfaces where over the local scale of analysis i.e. the patches over which affine transforms are measured, the Gaussian curvature does not vary 'too much'.

B Relation Between Spherical and Planar Perspective Projection

The geometry described in Section 3 assumes that we have spherical projection. In reality, our images are projected using planar perspective projection. Therefore we must convert between the two kinds of projection. This involves both rewriting the step $(\Delta x, \Delta y)$ on the viewing sphere and converting our empirically measured affine transform, $\bar{A}$, to the corresponding transformation on the viewing sphere.

Figure 24 depicts the situation. Let Σ be the viewsphere of unit radius centered at the origin of $\mathbf{R}^3$. Longitude and latitude on the earth suggest a convenient parametrization. We have the Cartesian coordinate system oriented as follows–the x-axis points due east, the y-axis points due north and the z-axis is oriented to make a right-handed coordinate system. The point $\mathbf{p}(\theta, \phi)$

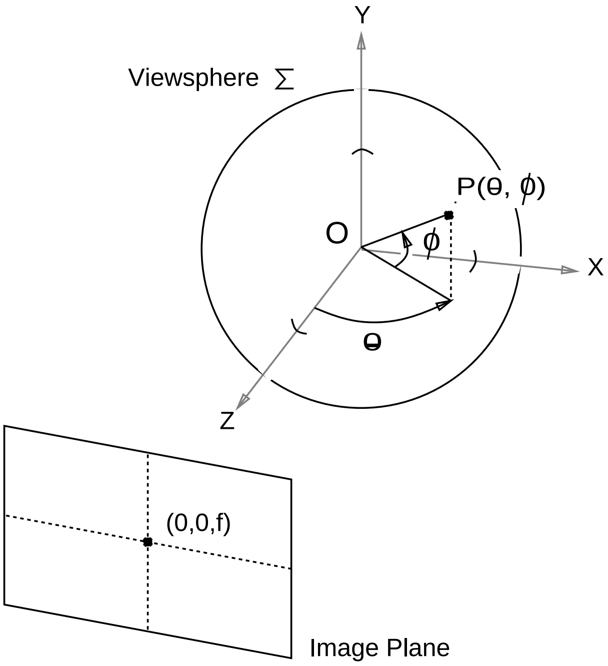


Figure 24: **Transforming image plane coordinates into longitude/lattitude spherical coordinates.** The optical center is at the origin, O, of the coordinate system. The Z-axis intersects the image plane $Z = f$ at the point $(0, 0, f)$. Point $P = P(\theta, \phi)$ lies on the surface of the unit viewsphere Σ. Point P corresponds to a point $Q = (X, Y, f)$ (not shown) in the image plane, i.e. point P is the intersection of ray $\overline{OQ}$ with the surface of the unit viewsphere. The projection of P onto the Z-X plane makes a counterclockwise angle of θ with respect to the Z-axis.

of Σ with longitude θ ($-\pi/2 \leq \theta \leq \pi/2$) and latitude ϕ ($-\pi/2 \leq \phi \leq \pi/2$) has Euclidean coordinates

$$\mathbf{p}(\theta,\phi) = (\cos\phi\sin\theta, \sin\phi, \cos\phi\cos\theta)$$

We can measure angles and distances on the viewing sphere using the (orthonormal) spherical frame field

$$\begin{aligned}
\mathbf{E}_1 &= \cos\theta\,\mathbf{i} - \sin\theta\,\mathbf{k} \\
\mathbf{E}_2 &= -\sin\phi\sin\theta\,\mathbf{i} + \cos\phi\,\mathbf{j} - \sin\phi\cos\theta\,\mathbf{k} \\
\mathbf{E}_3 &= \cos\phi\sin\theta\,\mathbf{i} + \sin\phi\,\mathbf{j} + \cos\phi\cos\theta\,\mathbf{k}
\end{aligned}$$

where $\mathbf{i}, \mathbf{j}$, and $\mathbf{k}$ are the unit vectors along the x, y, and z axes respectively. $\mathbf{E}_1$ points due east along a circle of latitude, $\mathbf{E}_2$ points due north along a circle of longitude. At a point (θ,ϕ) on the viewsphere, the vectors $\mathbf{E}_1$, $\mathbf{E}_2$ provide an orthonormal basis for the tangent space at the point. We define θ_t, the tilt angle, as the counterclockwise rotation needed to align the $\mathbf{E}_1$, $\mathbf{E}_2$ vectors with the $\mathbf{t}, \mathbf{b}$ vectors.

We next introduce the dual 1-forms of the $(\mathbf{E}_1, \mathbf{E}_2)$ basis, $D_1 = \cos\phi\, d\theta$ and $D_2 = d\phi$. If at a point (θ_0, ϕ_0), one takes a small step $(\Delta\theta, \Delta\phi)$, then the dual 1-forms tell us that the component of this vector in the $\mathbf{E}_1$ direction, Δx, is $\cos\phi_0\,\Delta\theta$ and the component in the $\mathbf{E}_2$ direction, Δy, is $\Delta\phi$.

We can identify the point $\mathbf{p}(\theta,\phi)$ of the viewsphere with the direction (X, Y, f) where X, Y are the coordinates of the projection on an image plane at distance f from the optical center. We then get the following simple relationship between θ, ϕ, and the image plane coordinates:

$$\theta = \tan^{-1}\frac{X}{f} \tag{25}$$

$$\phi = \sin^{-1}\frac{Y}{R} \tag{26}$$

where $R = \sqrt{X^2+Y^2+f^2}$. This defines a mapping from (X, Y) to (θ, ϕ) . We compute its Jacobian which can be used to approximate the step $(\Delta\theta, \Delta\phi)$, associated with the step in the image plane, $(\Delta X, \Delta Y)$.

$$\begin{bmatrix} \Delta\theta \\ \Delta\phi \end{bmatrix} \approx \begin{bmatrix} \frac{f}{X^2+f^2} & 0 \\ -\frac{XY}{R^2\sqrt{X^2+f^2}} & \frac{\sqrt{X^2+f^2}}{R^2} \end{bmatrix} \begin{bmatrix} \Delta X \\ \Delta Y \end{bmatrix}$$

We next need to find the step $(\Delta x, \Delta y)$ on the viewsphere measured in the orthonormal basis $(\mathbf{E}_1, \mathbf{E}_2)$, associated with the step $(\Delta\theta, \Delta\phi)$. For this we use the 1-forms above. Thus the complete transformation between $(\Delta X, \Delta Y)$ on the image plane and $(\Delta x, \Delta y)$ on the viewing sphere is:

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \approx \begin{bmatrix} \cos\phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{f}{X^2+f^2} & 0 \\ -\frac{XY}{R^2\sqrt{X^2+f^2}} & \frac{\sqrt{X^2+f^2}}{R^2} \end{bmatrix} \begin{bmatrix} \Delta X \\ \Delta Y \end{bmatrix} \tag{27}$$

$$
\begin{aligned}
&= \begin{bmatrix} \frac{f\cos\phi}{X^2+f^2} & 0 \\ -\frac{XY}{R^2\sqrt{X^2+f^2}} & \frac{\sqrt{X^2+f^2}}{R^2} \end{bmatrix} \begin{bmatrix} \Delta X \\ \Delta Y \end{bmatrix} \\
&= \begin{bmatrix} \frac{f}{R\sqrt{X^2+f^2}} & 0 \\ -\frac{XY}{R^2\sqrt{X^2+f^2}} & \frac{\sqrt{X^2+f^2}}{R^2} \end{bmatrix} \begin{bmatrix} \Delta X \\ \Delta Y \end{bmatrix} \\
&= J \begin{bmatrix} \Delta X \\ \Delta Y \end{bmatrix}
\end{aligned}
$$

Note that for the point $(0,0)$ where the optical axis intersects the image plane, J is just $1/f$ times the identity matrix and this transformation reduces to simply scaling $(\Delta X, \Delta Y)$ by a factor of $1/f$. For other points on the image plane, we get a distortion which becomes significant for large fields of view. Leonardo da Vinci was the first to point out this effect of position on the image plane. He drew a distinction between natural perspective (onto a sphere) and artificial perspective (onto a plane) as used in constructing a painting.

Converting from affine transforms $\overline{A}$ on the image plane to affine transforms $\overline{\overline{A}}$ measured with respect to the bases $(\mathbf{E}_{i1}, \mathbf{E}_{i2})$, $i = 1, 2$ on the image sphere is simple:

$$\overline{\overline{A}} = J_2 \bar{A} J_1^{-1} = G(\bar{A})$$

where J_i is as defined in equation 28.

To give an idea of the significance of the transformation, G, between the affine transform matrices on the plane and the sphere, we work it out numerically for one of our examples. Note that in general the effect of G will be greater for larger fields of view and for starting points which are further from the center of the image plane. The Lecture Hall image (Figure 13) was taken using a lens with a field of view of 45 degrees, which is greater than that for any of our other images. The focal length for this image is roughly 908 pixels. The effect of G will be more significant for the second point in this image, which is at $(122, -49)$ relative to the center of the image. Thus we expect G to have a greater effect at the second point in the Lecture Hall image than for any other points in our examples, so we will demonstrate the "worst case" effect of G for the affine transform between the second point and a point 25 pixels to the right. Our affine transform on the plane between these two points was

$$\bar{A} = \begin{bmatrix} 0.8433 & 0.0092 \\ 0.0178 & 0.9725 \end{bmatrix}$$

On the sphere, the transform is

$$\overline{\overline{A}} = \begin{bmatrix} 0.8366 & 0.0091 \\ 0.0180 & 0.9689 \end{bmatrix}$$

Note that the largest difference between the two matrices is 0.0067, in the $(1,1)$ entry. To get a feel for the order of magnitude of the error, note (Section 7) that the standard deviation of the errors in the elements of the affine transformation matrices for our synthetic examples was roughly twice this value. Since this was a worst case scenario for our examples, we assumed throughout that the transformation between the two forms of projection was insignificant, and ignored it.

References

[1] J. Aloimonos. "Shape from texture." *Biological Cybernetics*, Vol. 58, pp. 345-360, 1988.

[2] R. Bajcsy and L. Lieberman. "Texture gradient as a depth cue." *CGIP*, 5:52-67, 1976.

[3] A. Blake, H. Bulthoff, and D. Sheinberg. "Shape from texture: ideal observers and human psychophysics." *Vision Research*, 33(12):1723-37, Aug. 1993.

[4] A. Blake and C. Marinos. "Shape from texture: estimation, isotropy and moments." *Artificial Intelligence*, 45(1990):323-380.

[5] D.Blostein and N. Ahuja. "Shape from texture: integrating texture-element extraction and surface estimation." *IEEE Trans. on PAMI*, 11(12):1233–1251, December 1989.

[6] L.G. Brown and H. Shvaytser. "Surface orientation from projective foreshortening of isotropic texture autocorrelation." *IEEE Trans. on PAMI*, 12(6), June 1990, pp. 584-588.

[7] J. Cutting and R. Millard. "Three gradients and the perception of flat and curved surfaces." *Journal of Experimental Psychology: General*, 113(2):198–216, 1984.

[8] L.S. Davis, L. Janos, and S.M. Dunn. "Efficient recovery of shape from texture." *IEEE Trans. on PAMI*, 5(5), 1983.

[9] M.P. Do Carmo. "Riemannian Geometry." Boston: Birkhauser, 1992.

[10] J. Gårding. "Shape from texture for smooth curved surfaces in perspective projection." *Journal of Mathematical Imaging and Vision*, 2(4):327–50, Dec. 1992.

[11] J. Gårding. "Shape from texture and contour by weak isotropy." *J. of Artificial Intelligence*, 64(2):243-97, Dec. 1993.

[12] J. Gibson. "The Perception of the Visual World." Boston: Houghton Mifflin, 1950.

[13] K. Ikeuchi. "Shape From Regular Patterns." *J. of Artificial Intelligence*, 22:49-75, 1984.

[14] D. Jones and J. Malik. "Determining three-dimensional shape from orientation and spatial frequency disparities." *Proc. of the Second ECCV*, 1992.

[15] K. Kanatani and T.C. Chou. "Shape from texture: General Principle." J. of Artificial Intelligence, Vol. 38, pp. 1-48, 1989.

[16] J. Koenderink. "Solid Shape." Cambridge, Mass: MIT Press, 1990.

[17] J. Krumm and S. Shafer. "Shape from Periodic Texture Using the Spectrogram." Proc. CVPR, Champaign-Urbana, Illinois, 1992, 284-301.

[18] T. Lindeberg and J. Gårding. "Shape from Texture from a Multi-Scale Perspective" Proc. ICCV, Berlin, Germany, 1993, pp. 683-691.

[19] C. Marinos and A. Blake. "Shape from texture: the homogeneity hypothesis." Proc. ICCV, Osaka, Japan, 1990, pp.350-353.

[20] R.J. Marks. "Introduction to Shannon Sampling and Interpolation Theory." Springer-Verlag, 1991.

[21] B. O'Neill. "Elementary Differential Geometry." New York: Academic Press, 1966.

[22] W.H. Press, B.P. Flannery, S.A. Teukolski, W.T. Vetterling. "Numerical Recipes in C." Cambridge University Press, 1988.

[23] P.J. Rousseeuw and A.M. Leroy. "Robust Regression and Outlier Detection." Wiley, 1987.

[24] K.A. Stevens. "The information content of texture gradients." *Biological Cybernetics*, Vol. 42, pp. 95-105, 1981.

[25] B. Super and A. Bovik. "Shape-from-Texture by Wavelet-Based Measurement of Local Spectral Moments." Proc. CVPR, Champaign-Urbana, Illinois, 1992, 296-301.

[26] A. Verri, F. Girosi, and V. Torre. "Differential techniques for optical flow." *Journal of the Optical Society of America A*, 5:912–922, 1990.

[27] A.P. Witkin. "Recovering surface shape and orientation from texture." J. of Artificial Intelligence, Vol. 17, pp. 17-45, 1981.