Shape from texture without boundaries

D.A. Forsyth

Computer Science Division
U.C. Berkeley
Berkeley, CA 94720
daf@cs.berkeley.edu

Abstract. *We describe a shape from texture method that constructs a maximum a posteriori estimate of surface coefficients using only the deformation of individual texture elements. Our method does not need to use either the boundary of the observed surface or any assumption about the overall distribution of elements. The method assumes that texture elements are of a limited number of types of fixed shape. We show that, with this assumption and assuming generic view and texture, each texture element yields the surface gradient unique up to a two-fold ambiguity. Furthermore, texture elements that are not from one of the types can be identified and ignored. An EM-like procedure yields a surface reconstruction from the data. The method is defined for othographic views — an extension to perspective views appears to be complex, but possible. Examples of reconstructions for synthetic images of surfaces are provided, and compared with ground truth. We also provide examples of reconstructions for images of real scenes. We show that our method for recovering local texture imaging transformations can be used to retexture objects in images of real scenes.* **Keywords:** *Shape from texture, texture, computer vision, surface fitting*

There are surprisingly few methods for recovering a surface model from a projection of a texture field that is assumed to lie on that surface. **Global methods** attempt to recover an entire surface model, using assumptions about the distribution of texture elements. Appropriate assumptions are **isotropy** [15] (the disadvantage of this method is that there are relatively few natural isotropic textures) or **homogeneity** [1, 2]. Current global methods do not use the deformation of individual texture elements.

Local methods recover some differential geometric parameters at a point on a surface (typically, normal and curvatures). This class of methods, which is due to Garding [5], has been successfully demonstrated for a variety of surfaces by Malik and Rosenholtz [9, 11]; a reformulation in terms of wavelets is due to Clerc [3]. The method has a crucial flaw; it is necessary either to know that texture element coordinate frames form a frame field that is locally parallel around the point in question, or to know the differential rotation of the frame field (see [6] for this point, which is emphasized by the choice of textures displayed in [11]; the assumption is known as **texture stationarity**).

There is a **mixed** method, due to [4]. As in the local methods, image projections of texture elements are compared yielding the cosine of slant of the surface at each texture element up to one continuous parameter. A surface is interpolated using an extremisation method, the missing continuous parameter being supplied by the assumption that the texture process is Poisson (as in global methods) — this means that quadrat counts of texture elements are multinomial. This method has several disadvantages: firstly, it requires the assumption that the texture is a homogenous Poisson process; secondly, it requires some information about surface boundaries; thirdly, one would expect to extract more than the cosine of slant from a texture element.

1 A Texture Model

We model a texture on a surface as a marked point process, of unknown spatial properties; definitions appear in [4]. In our model, the marks are texture elements (*texels* or *textons*, as one prefers; e.g. [7, 8] for automatic methods of determining appropriate marks) and the orientation of those texture elements with respect to some surface coordinate system. We assume that the marks are drawn from some known, finite set of classes of Euclidean equivalent texels. Each mark is defined in its own coordinate system; the surface is textured by taking a mark, placing it on the tangent plane of the surface at the point being marked, translating the mark's origin to lie on the surface point being marked, and rotating randomly about the mark's origin (according to the mark distribution). We assume that these texture elements are sufficiently small that they will, in general, not overlap, and that they can be isolated. Furthermore, we assume that they are sufficiently small that they can be modelled as lying on a surface's tangent plane at a point.

2 Surface Cues from Orthographic Viewing Geometry

We assume that we have an orthographic view of a compact smooth surface and the viewing direction is the z-axis. We write the surface in the form $(x, y, f(x, y))$, and adopt the usual convention of writing $f_x = p$ and $f_y = q$.

Texture imaging transformations for orthographic views: Now consider one class of texture element; each instance in the image of this class was obtained by a Euclidean transformation of the model texture element, followed by a foreshortening. The transformation from the model texture element to the particular image instance is affine. This means that we can use the center of gravity of the texture element as an origin; because the COG is covariant under affine transformations, we need not consider the translation component further.

Furthermore, *in an appropriate coordinate system on the surface and in the image*, the foreshortening can be written as

$$\mathcal{F}_i = \begin{pmatrix} 1 & 0 \\ 0 & \cos\sigma_i \end{pmatrix}$$

where σ_i is the angle between the surface normal at mark i and the z axis.

The transformation from the model texture element to the i'th image element is then $\mathcal{T}_{M\to i} = \mathcal{R}_{G(i)}\mathcal{F}_i\mathcal{R}_{S(i)}$ where $\mathcal{R}_S(i)$ rotates the texture element in the local surface frame, $\mathcal{F}_i$ foreshortens it, and $\mathcal{R}_G(i)$ rotates the element in the image frame. From elementary considerations, we have that

$$\mathcal{R}_G(i) = \frac{1}{\sqrt{p^2+q^2}}\begin{pmatrix} p & q \\ -q & p \end{pmatrix}$$

The transformation from the model texture element to the image element is not a general affine transformation (there are only three degrees of freedom). We have:

Lemma 1: *An affine transformation $\mathcal{T}$ can be written as $\mathcal{R}_G\mathcal{F}\mathcal{R}_S$, where $\mathcal{R}_G$, $\mathcal{R}_S$ are arbitrary rotations and $\mathcal{F}$ is a foreshortening (as above) if and only if*

$$det(\mathcal{T})^2 = tr(\mathcal{T}\mathcal{T}^T) - 1$$

and

$$0 \leq det(\mathcal{T})^2 \leq 1$$

If these conditions hold, we say that this transformation is a **local texture imaging transformation**.

Proof: If the conditions are true, then $\mathcal{T}\mathcal{T}^T$ has one eigenvalue 1 and the other between zero and one. By choice of eigenvectors, we can diagonalize $\mathcal{T}\mathcal{T}^T$ to be $\mathcal{R}_G\mathcal{F}^2\mathcal{R}_G^T$, meaning that $\mathcal{T} = \mathcal{R}_G\mathcal{F}\mathcal{R}_S$ for arbitrary $\mathcal{R}_S$. The other direction is obvious. □

Notice that, given an affine transformation $\mathcal{A}$ that is a texture imaging transformation, we know the factorization into components only up to a two-fold ambiguity. This is because

$$\mathcal{A} = \mathcal{R}_G\mathcal{F}\mathcal{R}_S = \mathcal{R}_G(-\mathcal{I})\mathcal{F}(-\mathcal{I})\mathcal{R}_S = \mathcal{A}$$

where $\mathcal{I}$ is the identity. The other square roots of the identity are ruled out by the requirement that $\cos\sigma_i$ be positive.

Now assume that the model texture element(s) are known. We can then recover all transformations from the model texture elements to the image texture elements. We now perform an eigenvalue decomposition of $\mathcal{T}_i\mathcal{T}_i^T$ to obtain $\mathcal{R}_G(i)$ and $\mathcal{F}_i$. From the equations above, it is obvious that these yield the value of p and q at the i'th point *up to a sign ambiguity* (i.e. (p,q) and $(-p,-q)$ are both solutions). **The Texture Element is Unambiguous in a Generic Orthographic View:** Generally, the model texture element is not known. However, an image texture element can be used in its place. We know that an image texture element is within some (unknown) affine transformation of the model texture element, but this transformation is unknown. Write the transformation from image element j to image element i as

$$\mathcal{T}_{j\to i}$$

and recall that this transformation can be measured relatively easily in principle (e.g. [4, 9, 11]). If we are using texture element j as a model, there is a unique (unknown) affine transformation $\mathcal{A}$ such that

$$\mathcal{T}_{M\to i} = \mathcal{T}_{j\to i}\mathcal{A}$$

for *every* image element i. The rotation component of $\mathcal{A}$ is of no interest. This because rotating a model texture element simply causes the rotation on the surface, $\mathcal{R}_S$, to change, and this term offers no shape information. Furthermore, $\mathcal{A}$ must have positive determinant, because we have excluded the possibility that the element is flipped by the texture imaging transformation. Finally, $\mathcal{A}$ must have a positive element in the top left hand corner, because we exclude the case where $\mathcal{A}$ is $-\mathcal{I}\lceil$ because this again simply causes the rotation on the surface, $\mathcal{R}_S$ to change without affecting the element shape. Assume that we determine $\mathcal{A}$ by searching over the lower diagonal affine transformations to find transformations such that

$$\mathcal{T}_{M\to i} = \mathcal{T}_{j\to i}\mathcal{A}$$

is a texture imaging transformation for every i. It turns out that there is no ambiguity.

Lemma 2: *Given $\mathcal{T}_{M\to i}$ for $i = 1, \ldots, N$ is a texture imaging transformation arising from a generic surface, then $\mathcal{T}_{M\to i}\mathcal{B}$ is a texture imaging transformation for $i = 1, \ldots, N$, for $\mathcal{B}$ a lower-diagonal matrix of positive determinant and with $\mathcal{B}_{00} > 0$, if and only if $\mathcal{B}$ is the identity.*

Proof: Recall that only lower diagonal $\mathcal{B}$ are of interest, because a rotation in place does not affect the shape of the texture element. Recall that $det(\mathcal{M}) = det(\mathcal{M}^T)$ and $trace(\mathcal{M}) = trace(\mathcal{M}^T)$. This means that, for $\mathcal{M}$ a texture imaging transformation, both $\mathcal{M}$ and $\mathcal{M}^T$ satisfy the constraints above. We can assume without loss of generality that $\mathcal{T}_{M\to 1} = \mathcal{R}_G\mathcal{F}_1$ (because we have choice of coordinate frame up to rotation on the model texture element). This means that $\mathcal{T}^T_{M\to 1}\mathcal{T}_{M\to 1}$ is diagonal — it is the square of the foreshortening — and so $\mathcal{B}^T\mathcal{T}^T_{M\to 1}\mathcal{T}_{M\to 1}\mathcal{B}$ must also be diagonal and have 1 in the top left hand corner. This means $\mathcal{B}$ must be diagonal, and that $\mathcal{B}_{00} = 1$ (the case $\mathcal{B}_{00} = -1$ is excluded by the constraint above). So the only element of interest is $\mathcal{B}_{11} = \lambda$. Now for some arbitrary j — representing the transform of *another element* — we can write

$$\mathcal{T}^T_{M\to j}\mathcal{T}_{M\to j} = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$$

where $(a+c-1) = (ac-b^2)$. Now if $\mathcal{T}_{M\to j}\mathcal{B}$ is a texture imaging transformation, then we have

$$\mathcal{B}^T\mathcal{T}^T_{M\to j}\mathcal{T}_{M\to j}\mathcal{B} = \begin{pmatrix} a & \lambda b \\ \lambda b & \lambda^2 c \end{pmatrix}$$

Now this matrix must also meet the constraints to be a texture imaging transformation, so that we must have that $(a + \lambda^2 c - 1) = \lambda^2(ac - b^2)$ as well as $(a+c-1) = (ac-b^2)$. Rearranging, we have the system $(a-1) = ((a-1)c-b^2)$

and $(a-1) = \lambda^2((a-1)c - b^2)$ which has solutions when $\lambda^2 = 1$ (or when $(a,b) = (1,0)$, which is not generic). If $\lambda = -1$, then the transformation's determinant is negative, so it is not a texture imaging transformation, so $\lambda = 1$. □

Notice the case where $\lambda = -1$ corresponds to flipping the model texture element in its frame, and incorporating a flip back into the texture imaging transformation. Lemma 2 is crucial, because it means that, *for orthographic views, we can recover the texture element independent of the surface geometry* (whether we should is another matter). We have not seen lemma 2 in the literature before, but assume that it is known — it doesn't appear in [10], which describes other repeated structure properties, in [12], which groups plane repeated structures, or in [13], which groups affine equivalent structures but doesn't recover normals. At heart, it is a structure from motion result. **Special cases:** The non-generic cases are interesting. Notice that if $(a,b) = (1,0)$ for all image texture elements, then $0 \leq \lambda \leq min(1/\cos\sigma_j)$, where the minimum is over all texture elements. The easiest way to understand this case is to study the surface gradient field. In particular, at each texture element there is an iso-depth direction, which is perpendicular to the normal. Now apply $\mathcal{T}_{M\to j}^{-1}$ to this direction for the j'th texture element, yielding a direction in the frame of the model texture element. This direction, and this alone, is not foreshortened. In the case that $(a,b) = (1,0)$, for all texture elements the *same* direction is not foreshortened.

There are two cases. In the first, this effect occurs as a result of an unfortunate coincidence between view and texture field. It is a view property, because the gradient of the surface (which is determined by the view), is aligned with the texture field; this case can be dismissed by the generic view assumption. The more interesting case occurs when the texture element is circular; this means that the effect of rotations in the model frame is null, so that texture imaging transformations for circular spots are determined only up to rotation in the model frame. In turn, any distribution of circular spots on the surface admits a set of texture imaging transformations such that $(a,b) = (1,0)$. This texture is ambiguous, because one cannot tell the difference between a texture of circular spots in a generic view and a texture of unfortunately placed ellipses; furthermore, the fact that λ is indeterminate means that the surface may consist of ellipses with a high aspect ratio viewed nearly frontally, or ones with a low aspect ratio heavily foreshortened. Again, a generic view assumption would allow only the first interpretation, so when a texture element appears like an ellipse, we fix its shape as a circle.

Reasoning about the iso-depth direction in the model texture element's frame allows us to understand lemma 2 in somewhat greater detail. In effect, the reason $\mathcal{B}$ is generically unique is that it must fix many different directions in the model texture element's frame. The only transformations that do this are the identity and a flip.

Recovering geometric data for orthographic views of unknown texture elements: Now assume that the texture element is unknown, but each texture imaging transformation is known. Then we have an excellent estimate of the texture element. In the simplest case, we assume the image represents the

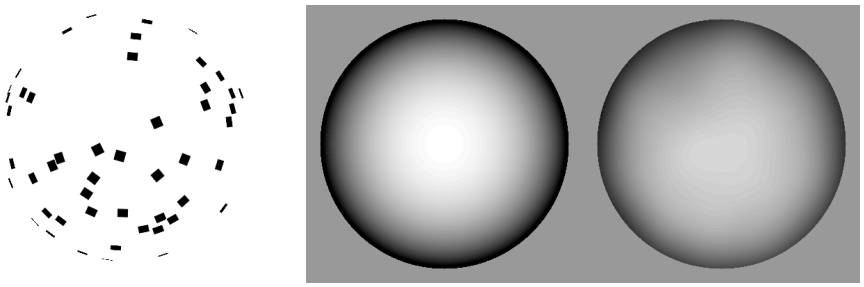

Fig. 1. *The reconstruction process, illustrated for a synthetic image of a sphere.* **Left**, *an image of a textured sphere, using a texture element that is not circular.* **Center**, *the height of the sphere, rendered as an intensity field, higher points being closer to the eye;* **right** *shows the reconstruction obtained using the EM algorithm using the same map of height to intensity.*

albedo (rather than the radiosity), and simply apply the inverse of each texture imaging transformation to its local patch and average the results over all patches. In the more interesting case, where we must account for shading variation, we assume that the irradiance is constant over the texture element. Write $\mathcal{I}_\mu$ for the estimate of the texture element, and $\mathcal{I}_i$ for the patch obtained by applying $\mathcal{T}_i^{-1}$ to the image texture element i. Then we must choose $\mathcal{I}_\mu$ and some set of constants λ_i to minimize

$$\Sigma_i \mid\mid \lambda_i \mathcal{I}_\mu - \mathcal{I}_i \mid\mid^2$$

Now assume that we have an estimate of the model texture element, and an estimate of the texture imaging transformation for each image texture element. In these circumstances, it is possible to tell whether an image texture element represents an instance of the model texture element or not — it will be an instance if, by applying the inverse texture imaging transformation to the image texture element, we obtain a pattern that looks like the model texture element. This suggests that we can insert a set of hidden variables, one for each image texture element, which encode whether the image texture element is an instance or not. We now have a rather natural application of EM. **Practical details:** For the i'th texture element, write θ_{g_i} for the rotation angle of the in-image rotation, σ_i for the foreshortening, θ_{s_i} for the rotation angle of the on-surface rotation and $\mathcal{T}_i$ for the texture imaging transformation encoded by these parameters. Write δ_i for the hidden variable that encodes whether the image texture element is an instance of the model texture element or not. Write $\mathcal{I}_\mu$ for the (unknown) model texture element.

To compare image and model texture elements, we must be careful about domains. Implicit in the definition of $\mathcal{I}_\mu$ is its domain of definition D— say a nxn pixel grid — and we can use this. Write $\mathcal{T}_i^{-1}\mathcal{I}$ for the pattern obtained by applying $\mathcal{T}_i^{-1}$ to the domain $\mathcal{T}_i(D)$. This is most easily computed by scanning D, and for each sample point $\mathbf{s} = (s_x, s_y)$ evaluating the image at $\mathcal{T}_i^{-1}\mathbf{s}$.

We assume that imaging noise is normally distributed with zero mean and standard deviation σ_{im}. We assume that image texture elements that are not instances of the model texture element arise with uniform probability. We have that $0 \leq \sigma_i \leq 1$ for all i, a property that can be enforced with a prior term. To avoid the meaningless symmetry where illumination is increased and albedo falls, we insert a prior term that encourages λ_i to be close to one. We can now write the negative log-posterior

$$\frac{1}{2\sigma_{im}^2}\sum_i \left(|| \lambda_i \mathcal{I}_\mu - \mathcal{T}_i^{-1}\mathcal{I} ||^2 \delta_i \right) + \sum_i (1-\delta_i)K + \frac{1}{2\sigma_{light}^2}(\lambda_i - 1)^2 + L$$

where L is some unknown normalizing constant of no further interest. The application of EM to this expression is straightforward, although it is important to note that most results regarding the behaviour of EM apply to maximum likelihood problems rather than maximum *a posteriori* problems. We are aware of no practical difficulties that result from this observation.

Minimisation of the Q function with respect to δ_i is straightforward, but the continuous parameters require numerical minization. This minimisation is unusual in being efficiently performed by coordinate descent. This is because, for fixed $\mathcal{I}_\mu$, each $\mathcal{T}_i$ can be obtained by independently minimizing a function of only three variables. We therefore minimize by iterating two sweeps: fix $\mathcal{I}_\mu$ and minimize over each $\mathcal{T}_i$ in turn; now fix all the $\mathcal{T}_i$ and minimize over $\mathcal{I}_\mu$.

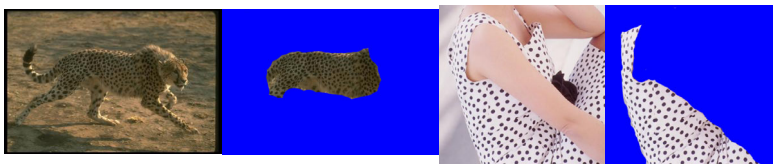

Fig. 2. *An image of a running cheetah, masked to suppress distractions, and of a spotted dress, ditto. These images were used to reconstruct surfaces shown in figures below.*

3 Surface Cues from Perspective Viewing Geometry

Shape from texture is substantially more difficult from generic perspective views than from generic orthographic views (unless, as we shall see, one uses the highly restrictive homogeneity assumption). We use spherical perspective for simplicity, imaging onto a sphere of unit radius.

Texture imaging transformations for perspective views: Because the texture element is small, the transformation from the model texture element to the particular image instance is affine. This means that we can again use the center of gravity of the texture element as an origin; because the COG is covariant under affine transformations, we need not consider the translation

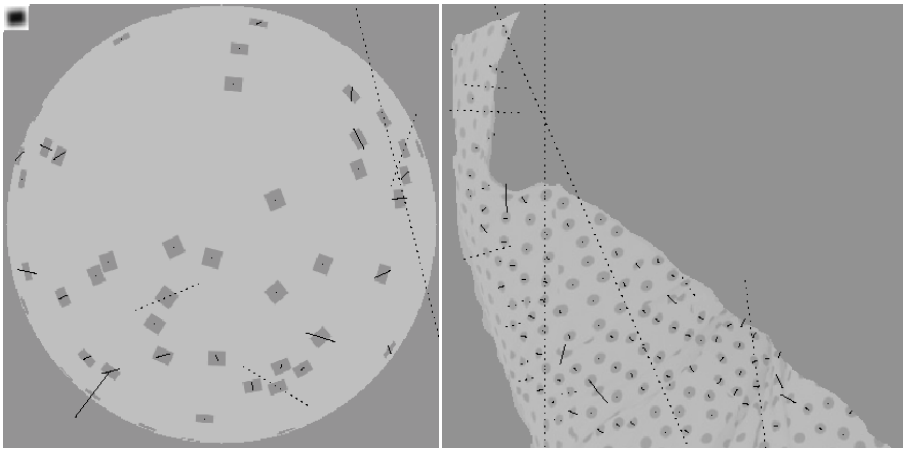

Fig. 3. Left, *the gradient information obtained by the process described in the text for a synthetic image of a textured sphere. Since the direction of the gradient is not known, we illustrate gradients as line elements, do not supply an arrow head and extend the element forward and backward. The gradient is relatively small and hard to see on the many nearly frontal texture elements — look for the black dot at the center of the element. Gradients are magnified for visibility, and overlaid on texture elements; gradients shown with full lines are associated with elements with a value of the hidden data flag (is this a texture element or not) greater than 0.5; the others are shown with dotted lines. The estimated element is shown in the top left hand corner of the image.* **Right**, *a similar plot for the image of the spotted dress.*

component further. The transformation from the model texture element to the i'th image element is then

$$\mathcal{T}^{(p)}_{M\to i} = \frac{1}{r_i}\mathcal{R}_G(i)\mathcal{F}_i\mathcal{R}_S(i)$$

where r_i is the distance from the focal point to the COG of the texture element, $\mathcal{R}_S(i)$ rotates the texture element in the local surface frame, $\mathcal{F}_i$ foreshortens it, and $\mathcal{R}_G(i)$ rotates the element in the image frame. We have superscripted this transformation with a (p) to indicate perspective. Again, $\mathcal{R}_G(i)$ is a function only of surface geometry (at this point, the form does not matter), and $\mathcal{R}_S(i)$ is a property of the texture field. This transformation is a scaled texture imaging transformation. This means it is a general affine transformation. This is most easily seen by noting that any affine transformation can be turned into a texture imaging transformation by dividing by the largest singular value.

The Texture Element is Ambiguous for Perspective Views: Recall we know the model texture element up to an affine transformation. If we have an orthographic view, the choice of affine basis is further restricted to a two-fold discrete ambiguity by lemma 2 — we need to choose a basis in which each $\mathcal{T}_{M\to i}$ is a texture imaging transformation, and there are generically two such bases. There is no result analogous to lemma 2 in the perspective case. This means

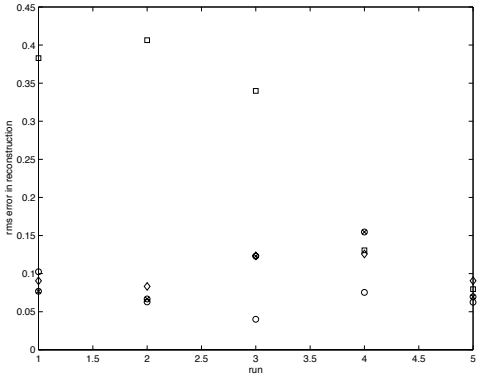

Fig. 4. *The root mean square error for sphere reconstruction for five examples each of four different cases, as a percentage of the sphere's radius. The horizontal axis gives the run, and the vertical axis gives the rms error. Note that in most cases the error is of the order of 10% of radius. Squares and diamonds correspond to the case where the texture element must be estimated (in each run, the image is the same, so we have five different such images), and circles and crossed circles correspond to the case where it is known (ditto). Squares and circles correspond to the symmetric error metric, and diamonds and crossed circles correspond to EM based reconstruction. Note firstly that in all but three cases the RMS error is small. Accepting that the large error in the first three runs using the symmetric error metric for the estimated texture element may be due to a local minimum, there is very little to choose between the cases. This suggests that (1) the process works well (2) estimating the texture element is successful (because knowing it doesn't seem to make much difference to the reconstruction process) and (3) either interpolation mechanism is probably acceptable.*

that any choice of basis is legal, and so, *for perspective views, we cannot recover the texture element independent of the surface geometry.*

This does not mean that shape from texture in perspective views is necessarily ambiguous. It does mean that, *for a generic texture*, we cannot disconnect the process of determining the texture element (and so measuring a set of geometric data about the surface) from the process of surface reconstruction. This implies that reconstruction will involve a fairly complicated minimisation process. We demonstrate shape from texture for only orthographic views here; this is because the process of estimating texture imaging transformations and the fitting process can be decoupled.

An alternative approach is to "eliminate" the shape of the model texture element from the problem. This was done by Garding [5], Malik and Rosenholtz [9, 11] and Clerc [3]; it can be done for the orthographic case, too [4], but there is less point. This can be done only under the assumption of local homogeneity.

4 Recovering Surface Shape from Transform Data

We assume that the texture imaging transformation is known uniquely at each of a set of scattered points. Now this means that at each point we know p and q *up*

to sign, leaving us with an unpleasant interpolation problem. We expect typical surface fitting procedures to be able to produce a surface for any assignment of signs, meaning that we require some method to choose between them — essentially, a prior.

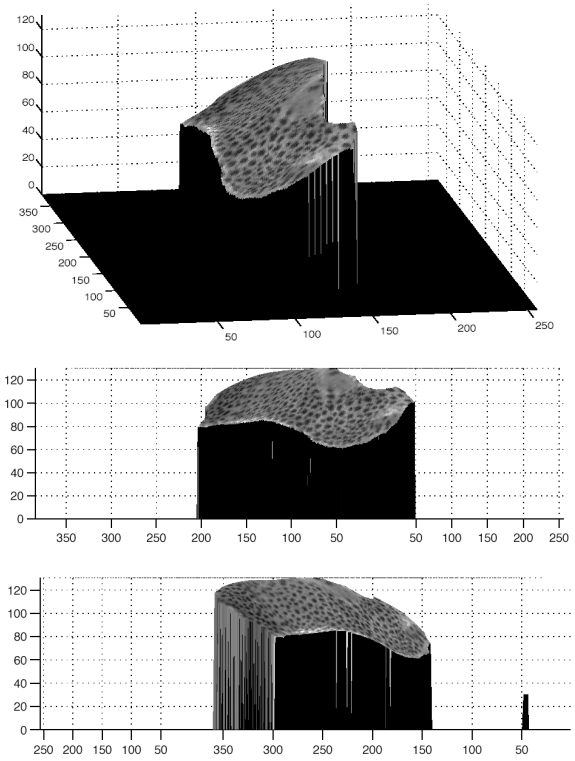

Fig. 5. *The reconstructed surface for the cheetah of figure 2 is shown in three different textured views; note that the curve of the barrel chest, turning in toward the underside of the animal and toward the flank is represented; the surface pulls in toward the belly, and then swells at the flank again. Qualitatively, it is a satisfactory representation of the cheetah.*

The surface prior: The question of surface interpolation has been somewhat controversial in the vision community in the past — in particular, why would one represent data that doesn't exist? (the surface components that lie between data points). One role for such an interpolate is to incorporate spatial constraints — the orientation of texture elements with respect to the view is unlikely to change arbitrarily, because we expect the scale of surface wiggles to be greater than the inter-element spacing. It is traditional to use a second derivative approximation to curvature as a prior; in our experience, this is unwise, because it is very

badly behaved at the boundary (where the surface is nearly vertical). Instead, we follow [4] and compute the norm of the shape operator and sum over the surface. This yields

$$\pi(\theta) \propto \exp\left\{-(\frac{1}{2\sigma_k^2})\int_R (\kappa_1^2 + \kappa_2^2)dA\right\}$$

where the κ_i are the extremal values of the normal curvatures.

Robustness: We expect significant issues with outliers in this problem. In practice, the recovery process for texture imaging transformations tends to be unreliable near boundaries, because elements are heavily foreshortened and so have lost some detail. As figure 3 indicates, there are occasional large gradient estimates at or near the boundary. It turns out to be a good idea to use a robust estimator in this problem. In particular, our log-likelihoods all use $\phi(x;\epsilon) = x/(x+\epsilon)$. We usually compose this function with a square, resulting in something proportional to the square for small argument and close to constant for large x.

The data approximation problem: Even with a prior, we have a difficult approximation problem. The data points are scattered and so it is natural to use radial basis functions. However, we have only the gradient of the depth, but, which is worse, we do not know the sign of the gradient. We could either fit using only to $p^2 + q^2$ — which yields a problem rather like shape from shading, *which requires boundary information*, which is often not available — or use the orientation of the gradient as well. To exploit the orientation of the gradient, we have two options.

Option 1: The symmetric method We can fit using only p^2 and q^2 — in which case, we are ignoring information, because this data implies a four-fold ambiguity and we have only a two-fold ambiguity. A natural choice of fitting error (or negative log-posterior, in modern language) is

$$\frac{1}{2\sigma_f^2}\sum_i \left(\phi(\left[p_i^2 - (z_x(x_i,y_i;\theta))^2\right]^2 + \left[q_i^2 - (z_y(x_i,y_i;\theta))^2\right]^2;\epsilon)\right) + \pi(\theta)$$

We call this the symmetric method because the negative log-posterior is invariant to the transformation $(p_i, q_i) \to (-p_i, -q_i)$

Option 2: The EM method The alternative is to approach the problem as a missing data problem, where the missing data is the sign of the gradient. It is natural to try and attack this problem with EM, too. The mechanics are straightforward. Write the depth function as $z(x, y; \theta)$, and the Gaussian curvature of the resulting surface as $K(\theta)$. The log-posterior is

$$\frac{1}{2\sigma_f^2}\sum_i \left(\begin{array}{c}\left(\phi([p_i - z_x(x_i,y_i;\theta)]^2 + [q_i - z_y(x_i,y_i;\theta)]^2)\right)(1-\delta_i) + \\ \left(\phi([p_i + z_x(x_i,y_i;\theta)]^2 + [q_i + z_y(x_i,y_i;\theta)]^2)\right)(\delta_i)\end{array}\right) + \pi(\theta)$$

where δ_i (the missing variables) are one or zero according as the sign of the data is correct or not. The log-posterior is linear in the hidden variables, so the Q function is simple to compute, but the maximization must be done numerically,

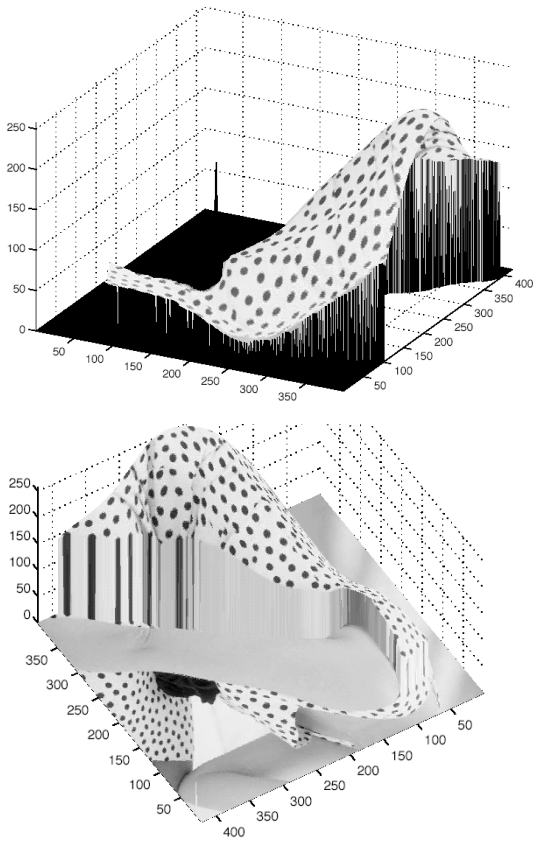

Fig. 6. *The reconstructed surface for the dress of figure 2 is shown in two different textured views. In one view, the surface is superimposed on the image to give some notion of the relation between surface and image. Again, the surface appears to give a fair qualitative impression of the shape of the dress.*

and is expensive because the Gaussian curvature must be integrated for each function evaluation (meaning that gradient evaluation is particularly expensive).

Approximating functions: In the examples, we used a radial basis function approximation with basis functions placed at each data point. We therefore have

$$z(x, y; \theta) = \sum_i \frac{a_i}{(x - x_i)^2 + (y - y_i)^2 + \nu}$$

where ν sets a scale for the approximating surfaces and a_i are the parameters. The main disadvantage with this approach is that the number of basis elements — and therefore, the general difficulty of the fitting problem — goes up with the number of texture elements. There are schemes for reducing the number of basis elements, but we have not experimented with their application to this problem.

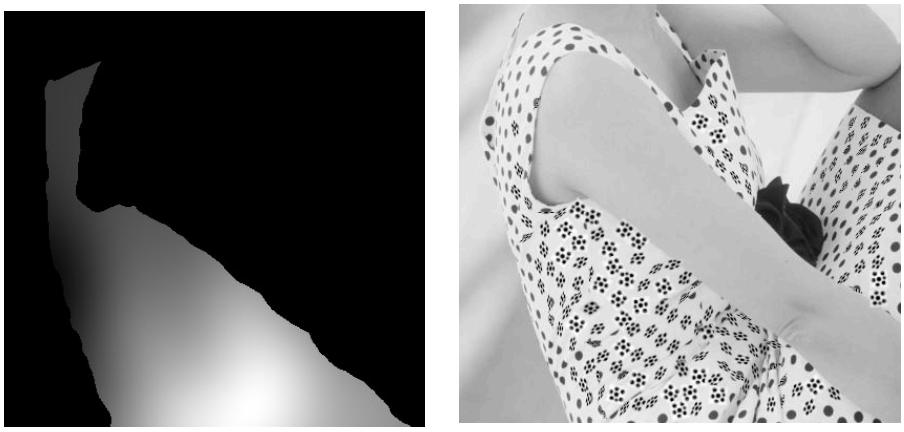

Fig. 7. *The texture on the surfaces of figure 6 implies that the surface can follow scale detail at a scale smaller than the elements on the dress; the figure on the* **left***, which is a height map with lighter elements being closer to the eye, indicates that this is not the case — the reconstruction smooths the dress to about the scale of the inter-element spacing, as one would expect. On the* **right***, texture remapping; because we know the texture imaging transformation for each detected element, we can remap the image with different texture elements. Note that a reconstruction is not required, and the ambiguity in gradient need not be resolved. Here we have replaced spots with rosettes. Missing elements, etc. are due to the relatively crude element detection scheme.*

4.1 Experimental results

We implemented this method for orthographic views in Matlab on an 867Mhz Macintosh G4. We used a crude template matcher to identify texture elements In the examples where the texture element is a circular spot, the element was fixed at a circular spot; for other examples, the element was estimated along with the texture imaging transformations. Determining texture imaging transformations for of the order of 60 elements takes of the order of tens of minutes. Reconstruction is achingly slow, taking of the order of hours for each of the examples shown; this is entirely because of the expense of computing the prior term, a computation that in principle is easily parallelised.

Estimating transformations: Figure 1 shows input images with gradient orientations estimated from texture imaging transformations superimposed. Notice that there is no arrow-head on these gradient orientations, because we don't know which direction the gradient is pointing. Typically, elements at or near the rim lead to poor estimates which are discarded by the EM — a poor transformation estimate leads to a rectified element that doesn't agree with most others, and so causes the probability that the image element is not an instance of the model element to rise.

Reconstructions compared with ground truth: In figure 4, we compare a series of runs of our process under different conditions. In each case, we used

synthetic images of a textured sphere, so that we could compute the root mean square error of the radius of the reconstructed surface. We did five runs each of four cases — estimated (square) texture element vs. known circular texture element and symmetric reconstruction vs EM reconstruction. In each run of the known (resp. unknown) texture element case, we used the same image for symmetric vs EM reconstruction, so that we could check the quality of the gradient information.

In all but three of the 20 runs, the RMS error is about 10% of radius. This suggests that reconstruction is successful. In the other three runs, the RMS error is large, probably due to a local minimum. All three runs are from the symmetric reconstruction algorithm applied to an estimated element. We know that the gradient recovery is not at fault in these cases, because the EM reconstruction algorithm recovered a good fit in these cases. This means that estimating the texture element is not significantly worse than knowing it. Furthermore, it suggests that the reconstruction algorithms are roughly equivalent.

Reconstructions from Images of Real Scenes: Figures 5 and 6 show surfaces recovered from images of real textured scenes. In these cases, lacking ground truth, we can only argue qualitatively, but the reconstructed surface appears to have a satisfactory structure. Typically, mapping the texture back onto the surface makes the reconstruction look as though it has fine-scale structure. This effect is most notable in the case of the dress, where the reconstructed surface looks as though it has the narrow folds typical of cloth; this is an illusion caused by the shading, as figure 7 illustrates.

Texture Remapping: One amusing application is that, knowing the texture imaging transformation and an estimate of shading, we can remap textures, replacing the model element in the image with some other element. This does not require a surface estimate. Quite satisfactory results are obtainable (figure 7).

5 Comments

Applications: Shape from texture has tended to be a core vision problem — i.e. interesting to people who care about vision, but without immediate practical application. It has one potential application in image based rendering — shape from texture appears to be the method with the most practical potential for recovering detailed deformation estimates for moving, deformable surfaces such as clothing and skin. This is because no point correspondence is required for a reconstruction, meaning that shape estimates are available from spotty surfaces relatively cheaply — these estimates can then be used to condition point tracking, etc., as in [14]. However, shape from texture has the potential advantage over Torresani *et al.*'s method that it does not require feature correspondences or constrained deformation models.

SFT=SFM: There is an analogy between shape from texture and structure from motion that appears in the literature — see, for example, [5, 9–11, 13] — but it hasn't received the attention it deserves. In essence, shape from texture is about one view of multiple instances of a pattern, and structure from motion is (currently) about multiple views of one instance of a set of points. Lemma

2 is, essentially, a structure from motion result, and if it isn't known, this is because it treats cases that haven't arisen much in practice in that domain. However, the analogy has the great virtue that it offers attacks on problems that are currently inaccessible from within either domain. For example, one might consider attempting to reconstruct a textured surface which does *not* contain repeated elements by having several views; lemma 2 then applies in structure from motion mode, yielding estimates of *each* texture element; these, in turn, yield normals, and a surface results.

Acknowledgments
Much of the material in this paper is in response to several extremely helpful and stimulating conversations with Andrew Zisserman.

References

1. Y. Aloimonos. Detection of surface orientation from texture. i. the case of planes. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 584–593, 1986.
2. A. Blake and C. Marinos. Shape from texture: estimation, isotropy and moments. *Artificial Intelligence*, 45(3):323–80, 1990.
3. M. Clerc and S. Mallat. Shape from texture through deformations. In *Int. Conf. on Computer Vision*, pages 405–410, 1999.
4. D.A. Forsyth. Shape from texture and integrability. In *Int. Conf. on Computer Vision*, pages 447–452, 2001.
5. J. Garding. Shape from texture for smooth curved surfaces. In *European Conference on Computer Vision*, pages 630–8, 1992.
6. J. Garding. Surface orientation and curvature from differential texture distortion. In *Int. Conf. on Computer Vision*, pages 733–9, 1995.
7. T. Leung and J. Malik. Detecting, localizing and grouping repeated scene elements from an image. In *European Conference on Computer Vision*, pages 546–555, 1996.
8. J. Malik, S. Belongie, J. Shi, and T. Leung. Textons, contours and regions: cue integration in image segmentation. In *Int. Conf. on Computer Vision*, pages 918–925, 1999.
9. J. Malik and R. Rosenholtz. Computing local surface orientation and shape from texture for curved surfaces. *Int. J. Computer Vision*, pages 149–168, 1997.
10. J.L. Mundy and A. Zisserman. Repeated structures: image correspondence constraints and 3d structure recovery. In J.L. Mundy, A. Zisserman, and D.A. Forsyth, editors, *Applications of invariance in computer vision*, pages 89–107, 1994.
11. R. Rosenholtz and J. Malik. Surface orientation from texture: isotropy or homogeneity (or both)? *Vision Research*, 37(16):2283–2293, 1997.
12. F. Schaffalitzky and A. Zisserman. Geometric grouping of repeated elements within images. In D.A. Forsyth, J.L. Mundy, V. diGesu, and R. Cipolla, editors, *Shape, contour and grouping in computer vision*, pages 165–181, 1999.
13. T.Leung and J. Malik. Detecting, localizing and grouping repeated scene elements from an image. In *European Conference on Computer Vision*, pages 546–555, 1996.
14. L. Torresani, D. Yang, G. Alexander, and C. Bregler. Tracking and modelling non-rigid objects with rank constraints. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2001. to appear.
15. A.P. Witkin. Recovering surface shape and orientation from texture. *Artificial Intelligence*, 17:17–45, 1981.