

Recovering shape and irradiance maps from rich dense texton fields

Anthony Lobay and D.A. Forsyth

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

Abstract

We describe a method that recovers an estimate of surface shape and of the irradiance field for a textured surface. The method assumes the surface is viewed in scaled orthography, and we demonstrate the appropriateness of this assumption. Our method uses interest points to obtain the locations of putative texton instances, clusters the textons into types, and then uses an autocalibration method to recover the frontal appearance of each texton model. This yields (a) a dense set of normal estimates, each up to a two-fold ambiguity (b) a dense set of irradiance estimates and (c) whether each instance is, in fact, an instance of the relevant texton. Because we are able to obtain a very large number of instances of a large number of different textons, this information is obtained at sites very closely spaced in the image. As a result, we need only a simple smoothness constraint to reconstruct a surface model, using EM to resolve the normal ambiguity.

We show results on images of real scenes, comparing our reconstructions with those obtained using other methods and demonstrating the accuracy of both the recovered shape and the irradiance estimate. **Keywords:** Shape from texture, texture, computer vision, surface fitting, shading maps, textons, point features

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** [23] (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 [8], has been successfully demonstrated for a variety of surfaces by Malik and Rosenholtz [17, 19]; a reformula-

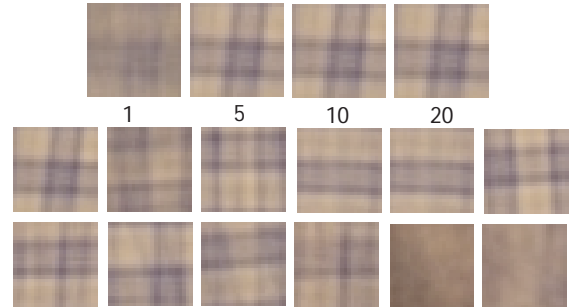


Figure 1: The **top row** shows estimates of the frontal appearance of a texture element for the image of the shirt depicted in figure 2 after 1, 5, 10 and 20 iterations of EM respectively. Initially the estimate is blurred, because the slant-tilt estimates are poor, but very quickly it becomes sharp. The **other rows** show the frontal appearance of each of the 12 texture elements found for this shirt. Note that the clustering could reasonably be criticized, but that it is not particularly important to identify the correct number of clusters. Each texton consists of a small patch centered on some part of the shirt pattern; the more such patches, the better, because this leads to a very dense set of surface orientation and irradiance estimates. The two elements on the bottom right are difficult to localize; this is detected automatically using the Hessian trick of section 3.1 and they are omitted from reconstruction.

tion in terms of wavelets is due to Clerc and Mallat [4, 5]. The method requires texture element coordinate frames to form a frame field that is locally parallel around the point in question (see [9] for this point; the assumption is known as **texture stationarity**). It is not known how widespread stationary textures are, but mechanisms of texture production such as surface damage or painting clearly are not biased toward stationary textures, though reaction-diffusion equations might be. As a result, these methods are not known to work on a large class of textured surfaces.

Perspective views are assumed for most shape from

texture methods. This is important for views of planes, because a view of a plane spanning a small visual angle can encompass a very large change in. One usually ignores the effects of perspective when the range of depths spanned by the observed scene is small compared to the average depth (1/10, say). Curved surfaces tend to meet this test. Pairs of equivalent texture elements that display appreciable perspective effects (i.e. two image instances are not within an affine map) are *a fortiori* far apart in space and so on the surface — it would be most unwise to use such pairs of elements to make local curvature estimates because they are far apart on the surface. This means that an orthographic model is sufficient to recover shape estimates for the vast majority of curved surfaces.

Surface interpolation methods have largely fallen out of fashion in computer vision, due to the uncertainty regarding the semantic status of surface patches in regions where data is absent. Shape from texture is a problem where an interpolate has an unquestionably useful role — it expresses the fact that, because one has a prior belief that surfaces are relatively slowly changing, incomplete local measurements of the surface normal can constrain one another and lead to good global estimates of the normal at some points.

Applications for shape from texture have been largely absent, explaining its status as a minority interest. However, we believe that image-based rendering of clothing is an application with substantial promise. Cloth is difficult to model for a variety of reasons. It is much more resistant to stretch than to bend: this means that dynamical models result in stiff differential equations (for example, see [21]) and that it buckles in fine scale, complex folds (for example, see [3]). However, rendering cloth is an important technical problem, because people are interesting to look at and most people wear clothing. A natural strategy for rendering objects that are intrinsically difficult to model satisfactorily is to rearrange existing pictures of the objects to yield a rendering. In particular, one would wish to be able to retexture and reshadde such images. This paper demonstrates methods that will make this possible.

Our shape from texture process uses a texture model and structure from motion lemma from [7]. We recapitulate this material briefly for the reader’s convenience in sections 1 and 2. The major new material in this paper involves practical applications of this method in a pipeline where we:

Recover image instances of multiple distinct texture elements, which we do using the interest point method of [15] (section 3).

Recover the frontal appearance of all elements, which we do using lemma 2 of [7]. By so doing we can exclude uninformative elements, obtain irradiance and normal esti-

mates, and (often) significantly enrich the field of elements (section 2).

Obtain a surface model and an irradiance map, using EM to resolve the two-fold ambiguity that results from our recovery method (section 4).

1 A Texture Model

We model a texture on a surface as a marked point process, of unknown spatial properties. A **point process** is some random procedure that results in points lying on a surface (exact definitions involve tedious measure theory [6]). A **marked point process** is one where each point carries a **mark**, drawn randomly according to some mark density from an available collection (for example, points might be red or blue; rendered as squares or circles; etc.); we assume that this collection is discrete.

In our model, the marks are texture elements (*texels* or *textons*, as one prefers; we use the term *instances* to refer to the marks that appear in the image) 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 and rotating randomly about the mark’s origin (according to the mark distribution). We assume that the texture elements do not occlude one another and are sufficiently small that they can be modelled as lying on a surface’s tangent plane at a point.

1.1 Surface Cues from 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 $f_y = q$.

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.



Figure 2: *On the **center left**, an image of a shirt with the position of each texton instance superimposed as a cross; there are so many it is difficult to resolve them, as the detail from the collar region (inset, **left**) shows. There are 350 instances in total, and instances are less dense in the area of darker shading near the arms. Instances from the area indicated do not result in much surface normal data, because the representation provided by Lowe’s method appears to be sensitive to relatively large changes in brightness. This means that the reconstruction using all instances (**top center**: textured, **bottom center**: untextured) has some problems that result from a large region without data. If one crops the image to the box shown on the **left**, the reconstruction, shown on the **right** is much better.*

The transformation from the model texture element to the i ’th image element is then

$$\mathcal{T}_{M \rightarrow i} = \mathcal{R}_{G(i)} \mathcal{F}_i \mathcal{R}_{S(i)}$$

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).

Lemma 2 of [7] says that, given a sufficient number of image instances — which number is three or more — of a small texture element in a scaled orthographic view, the element can be determined up to rotation in its own coordinate system. The process produces the texture imaging transformation at each image instance, and yields a factorisation that gives the slant-tilt frame (and so p and q) up to a two-fold ambiguity. Shape from texture is well known to have strong analogies with structure from motion [17], and this lemma can be restated as saying that, given a sufficient number of scaled orthographic views of a plane object, the object is known (as are the views). This is a self-calibration result, and should be compared with the known fact that five perspective views of a plane object yield camera calibration [22]. Furthermore, the process is not limited to a single texture element — there might be many different textons.

2 Frontal Textons, Irradiance and Normals

The development in this section assumes that there is a single texture element. However, we can deal with multiple texture elements by identifying and clustering instances separately (section 3). We then recover normal and irradiance information for each element separately, and finally reconstruct a surface and irradiance field by fusing all this information (section 4). We can work with each instance separately because we cluster texton instances with an affine-robust method.

2.1 Recovering Information for a Single Texton

For the moment, assume that all texture imaging transformations are known, but the element is not known. If the irradiance is unknown, we can assume it is constant over the texture element (elements are “small”). Write \mathcal{I}_μ for the estimate of the texture element, and \mathcal{I}_i for the patch obtained by applying the known texture imaging transformation \mathcal{T}_i^{-1} to the image texture element i . Then we must choose \mathcal{I}_μ and some set of constants λ_i to minimize

$$\sum_i \|\lambda_i \mathcal{I}_\mu - \mathcal{I}_i\|^2$$

and these constants represent the irradiance field.

Now assume that we have an estimate of the model texture element and the irradiance field; we can clearly recover the texture imaging transformations by transforming the lighted model texture element to look like an image patch. Finally, given all parameters, 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 and irradiance 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 observation is an instance or not. We now have a rather natural application of EM.

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 $\mathcal{T}_i = \mathcal{T}_i(\theta_{g_i}, \sigma_i, \theta_{s_i})$ 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}_μ 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}_μ is its domain of definition D — say a $n \times n$ 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 $\mathcal{I}_i(D)$. 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 use a prior that charges for λ_i different from one. We can now write the negative log-posterior

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

where L is some unknown normalizing constant of no further interest. The application of EM to this expression is straightforward. Computing expected values of the δ_i follows the usual pattern, but the continuous parameters require numerical minimization. This minimisation is unusual in being efficiently performed by coordinate descent. This is because, for fixed \mathcal{I}_μ , 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}_μ and minimize over each \mathcal{T}_i in turn; now fix all the \mathcal{T}_i and minimize over \mathcal{I}_μ .

This process produces **normal information** automatically, as each \mathcal{T}_i is an explicit function of rotation on the surface, the surface slant and p and q (section 1). However, there is a two-fold ambiguity, as a rotation on the surface of 180° can be absorbed by the map $(p, q) \rightarrow (-p, -q)$. Furthermore, the **EM Coefficients** encode the extent to which

an image pattern is, in fact, an instance of a texton. However, with many image elements the process could be slow. In fact, **increased efficiency** is possible because, although using all putative instances gives the best estimate of the frontal element, one runs into diminishing returns quite quickly. This suggests our strategy of using a subset of the instances to estimate the frontal element, then fixing the appearance of the element and using this to estimate configuration parameters, irradiance and δ 's for all other instances. Recovery of the frontal appearance of the texton is good; figure 1 shows all frontal textons from the shirt of figure 2. Recall that frontal appearances are estimated by backprojection and averaging: The relatively crisp images suggest the image instances have been well registered by the backprojection process.

3 Finding Instances of Textons

Assume we have a view of a surface textured with scattered instances of multiple elements. The results above indicate that if we can identify enough instances of elements we can recover normals and irradiance.

For the moment, assume that there is a single element. We must now find image patches that appear to be instances of the same texture element. There is some history of doing this successfully by clustering image patches (e.g. [13, 16]). A particularly simple and effective mechanism has been made available by recent work on representing image patches around interest points. Schmid and Mohr demonstrated that one could match objects by identifying interest points in an image and then building representations of the image around those points [20]. The key observation in this work is that an appropriately chosen representation can (a) distinctively identify image patches and (b) be robust to affine transformations. Such representations are now widely used in recognition (e.g. [14, 15, 20]; points are matched to points in images of models) and tracking (e.g. [15]; points are matched to points in the next frame). Furthermore, one can build a texture representation by identifying points that repeat and are good for matching [11, 12]. However, the emphasis in these last papers is on reducing the number of interest points by identifying patches that are uncommon within one scene and match well across views. Instead, we use interest points to obtain patches that match within a fixed scene and we want a dense set of texton instances.

A comparison of methods by Mikolajczyk and Schmid [18] is unequivocally in favour of the method of Lowe [15], which we use. We adopt this method. We obtain the descriptors for the scene by applying Lowe's program (which he has kindly made available at <http://www.cs.ubc.ca/~lowe/keypoints/>). These descriptors are then clustered using k-means to find

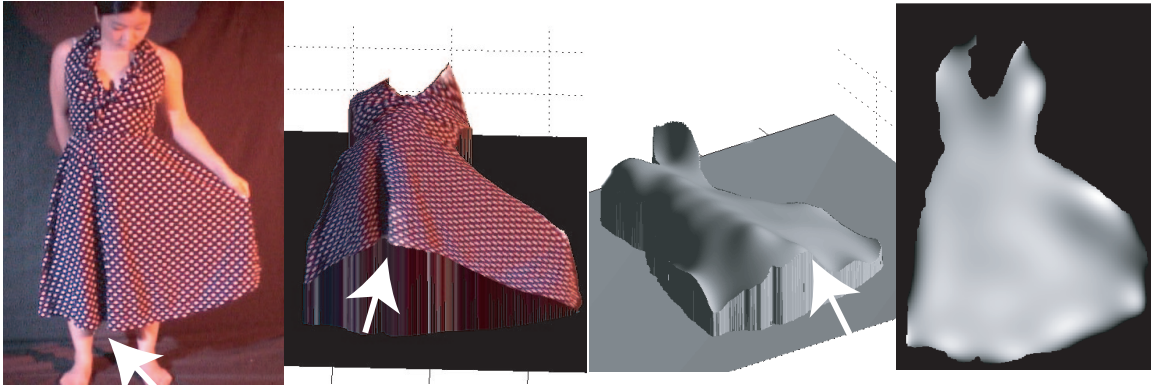


Figure 3: On the **left**, a view of a model in a spotted dress. In the **center left**, a textured view of the reconstruction obtained using our method. This reconstruction used 1200 texton instances, in 8 clusters. Note the relatively fine detail that was obtained by the reconstruction, including the two main folds in the skirt (indicated with arrows). Typically, rendering texture on top of the view produces a better looking surface, so we show the surface without texturing on the **center right**; arrows indicate the reconstructed folds in the geometry. Notice that the fold in the skirt is well represented. The smoothing term is generally good at resolving normal ambiguities, but patches of surface that are not well connected to the main body can be subjected to a concave-convex ambiguity, as has happened to part of the skirt’s bodice here. On the **right**, the irradiance map estimated using our method.

descriptors that appear to represent instances of the same texture element. Because the descriptors produced by Lowe’s program are invariant under rotation and translation and robust to quite substantial foreshortening, each cluster should represent instances of a potential texture element. There is little reason to attempt to extract heavily foreshortened instances, because *a fortiori* they must result in poor estimates of surface normal and of element appearance (there are few pixels on the element). We must now determine (a) which putative instances are, in fact, instances and (b) which textons are useful. This information emerges from the process of recovering frontal textons.

3.1 Handling Multiple Textons

It is relatively straightforward to deal with multiple textons (figure 1, figure 2). We first cluster putative instances using k-means; note that the value of k isn’t crucial here, as long as it is neither too small nor too large. This is because if k exceeds the number of texton classes, some elements will be represented by more than one cluster. The only consequence of processing these clusters independently is (in principle) a slight reduction in the accuracy with which the frontal appearance of the element can be estimated. This doesn’t appear in practice.

Each cluster is then processed independently, to produce independent frontal appearance, normal and irradiance estimates at the instance centers. The irradiance estimates for a given element are known up to a single missing scale factor. We can fix the scales for one element, and must now scale all others to be consistent with that element

(which we do by smoothness).

Bad elements are those that cannot produce reliable estimates of p and q ; for example, consider an element that has a constant grey level, or is a single point. We identify bad elements by looking at the Hessian of the fitting criterion. If this has small eigenvalues, then the estimates of p and q are unreliable. Generally, we expect this phenomenon to be a property of the texture element rather than of instances, and so we remove texture elements whose Hessian has too small a norm. Once we have done so, nothing further need be done to merge estimates of p and q obtained from different texture elements. Note that, in principle, one might extend this trick and use the Hessian as a guide to an appropriate weighting of the fitting error, but we see no practical advantage in doing so.

4 Fitting a Surface and an Irradiance Map

We now have a set of points (x_i, y_i) at which we know measurements of the gradient up to a two fold sign ambiguity: either $\mathbf{d}_i = (p, q)$ or $\mathbf{d}_i = (-p, -q)$. Furthermore, we have an estimate — from the expected value of the hidden variable in the previous section — of the reliability of these measurements. We accept only measurements for which these expected values exceed a threshold (0.8 for what follows). There are three possibilities at each point that has been accepted: First, $\mathbf{d}_i = (p, q)$; second, $\mathbf{d}_i = (-p, -q)$; third, the measurement does not derive from the surface (a bad texton match, say). We encode these states using a missing variable, and apply EM.

Assume for the moment that there is no sign ambiguity. We must now fit a surface to gradient data. We represent the surface with radial basis functions, a natural choice for scattered data interpolation. We use $\phi_j(x, y) = 1/((x - x_j)^2 + (y - y_j)^2 + \epsilon^2)$ as a basis function and we require that the normal measurement be orthogonal to the tangent of the fitted surface. If we write p_i for the measured x -derivative at $\mathbf{x}_i = (x_i, y_i)$, etc., we must minimize

$$\sum_{i \in \text{points}} \left(\frac{\partial h}{\partial x}(\mathbf{x}_i) - p_i \right)^2 + \left(\frac{\partial h}{\partial y}(\mathbf{x}_i) - q_i \right)^2$$

where h is a linear function of the vector of surface coefficients \mathbf{a} so that the error is quadratic in the surface coefficients. We should like to impose a smoothness constraint, and have found in practice that penalizing large coefficients is quite sufficient (for this method, see, for example [10]). Incorporating the hidden variables, the log-likelihood becomes

$$\sum_i \left[\begin{array}{l} \delta_i^1 \frac{1}{2\sigma_i^2} \left\{ \left(\frac{\partial h}{\partial x}(\mathbf{x}_i) - p_i \right)^2 + \left(\frac{\partial h}{\partial y}(\mathbf{x}_i) - q_i \right)^2 \right\} + \\ \delta_i^2 \frac{1}{2\sigma_i^2} \left\{ \left(\frac{\partial h}{\partial x}(\mathbf{x}_i) + p_i \right)^2 + \left(\frac{\partial h}{\partial y}(\mathbf{x}_i) + q_i \right)^2 \right\} + \\ \delta_i^3 K \end{array} \right] + \lambda \mathbf{a}^T \mathbf{a} + C$$

where C is a constant of no further interest and λ adjusts the weight of the smoothness term with respect to the error term. From this point, the application of EM is straightforward; the expression for the re-estimates of the hidden variables are the usual, and when known values of the hidden variables are substituted the minimization problem involves solving a linear system. As a result, the method is very much faster than that of [7]; it appears to produce much better surfaces, too. **A direct method** is possible, too. One uses the approximating surface to evaluate the slant and tilt at each texton instance, recovers the image instance to its frontal frame, and compares with the recovered appearance of the texton. In principle, this approach should lead to an improved representation because one can then couple the process of estimating whether an image pattern is a texton instance with that of estimating the approximating surface. In our experience, this does not materially change the recovered surface, probably because there are so many instances that are good that the accuracy added in principle is not significant in practice. **Starting the method** is straightforward. In all the examples shown, we start with a vertical cylinder.

4.1 Recovering an Irradiance Map

At each acceptable measurement is an estimate of irradiance *relative to other instances of that texton*. This estimate is available because we must shade the texton to get it to agree with the image. We can recover a relative irradiance field if we can scale each class of textons with respect

to a reference class. This, again, is a maximization problem. We approximate the scattered irradiance field with radial basis functions. Write L_i for the irradiance estimate at the data point $\mathbf{x}_i = (x_i, y_i)$, and $\phi_j(x, y)$ for the radial basis function $1/((x - x_j)^2 + (y - y_j)^2 + \epsilon^2)$. We wish to scale the irradiance value for each class of textons with respect to the first class (say), using a scaling value s_k for the k 'th class. We must then minimize

$$\sum_{i \in \text{class 1}} (L_i - \sum_j a_j \phi_j(\mathbf{x}_i))^2 + \sum_{k \in \text{other classes}} \left\{ \sum_{i \in \text{class } k} (s_k L_i - \sum_j a_j \phi_j(\mathbf{x}_i))^2 \right\} + \lambda \sum_j a_j^2$$

(where the last term is a smoothness term, as above) with respect to the s 's and the a 's. This is a straightforward linear system.

5 Results

It is always difficult to **evaluate** a reconstruction method, particularly if ground truth is not available. For most interesting cases of shape from texture, it is not; furthermore, synthetic images are an unreliable guide. Figure 2 demonstrates just how rich a set of feature points we obtain, suggesting that a competent reconstruction method should be able to obtain detail at quite a fine scale. In the lowest third of that image there are relatively few accurate orientation estimates because some dark patches mean that many instances are poorly rectified. We ascribe this phenomenon to strong shading differences following [18], who note that Lowe's method is affected by strong changes in illumination. If one reconstructs incorporating the scattered good measurements that remain in this area, the reconstruction is poorer than if one omits them (figure 2). This implies that a rich set of feature points is truly helpful.

Our reconstructions are qualitatively accurate, too. Figures 3 and 4 show reconstructions of different dresses. Note the reconstructions have been able to identify the visible folds in the dress, and the overall fall of the garment. Again, we attribute this to the dense set of texton instances meaning that we can reconstruct surface detail at quite a small scale. Furthermore, the irradiance maps in these reconstructions appear reasonable, offering some guide to the folds on the original garment (see figure 4).

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.

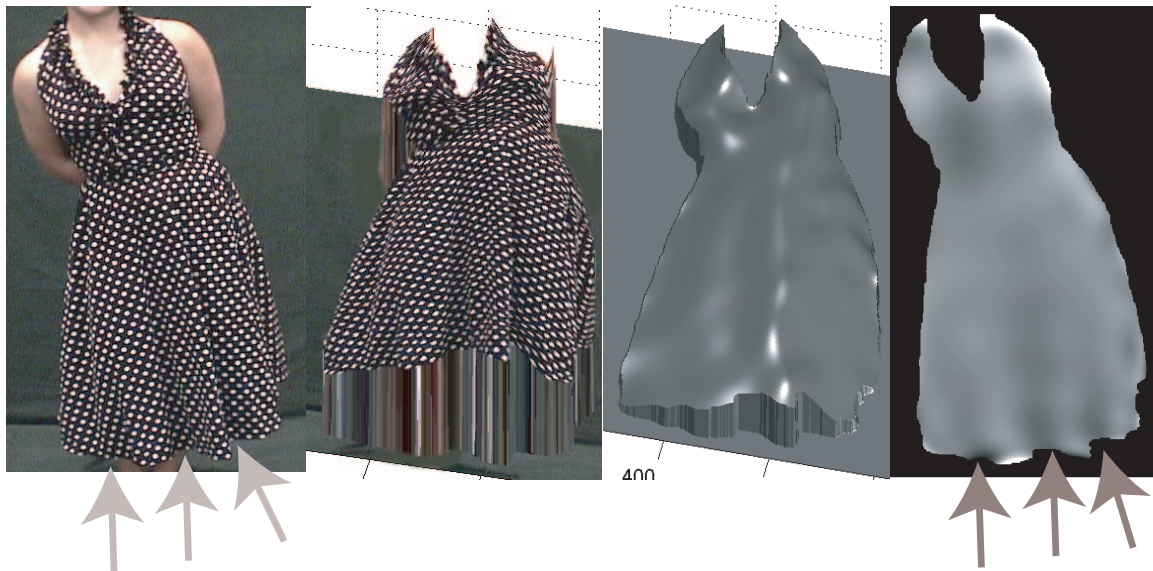


Figure 4: *On the left, another view of a model in a spotted dress. In the center left, a textured view of the reconstructed surface and on the center right, a view without texture. On the right, the irradiance map recovered using our method. The arrows point to folds in the original image of the skirt that are reproduced in the irradiance map; they should be observable there, because the change in orientation of the surface produces a change in irradiance.*

- [2] A. Blake and C. Marinos. Shape from texture: estimation, isotropy and moments. *Artificial Intelligence*, 45(3):323–80, 1990.
- [3] R. Bridson, R. Fedkiw, and J. Anderson. Robust treatment of collisions, contact and friction for cloth animation. *Computer Graphics*, (Annual Conference Series):594–603, 2002.
- [4] M. Clerc and S. Mallat. Shape from texture through deformations. In *Int. Conf. on Computer Vision*, pages 405–410, 1999.
- [5] M. Clerc and S. Mallat. The texture gradient equation for recovering shape from texture. *IEEE T. Pattern Analysis and Machine Intelligence*, 24(4):536–549, 2002.
- [6] D.J. Daley and D. Vere-Jones. *An Introduction to the theory of point processes*. Springer-Verlag, 1988.
- [7] D.A. Forsyth. Shape from texture without boundaries. In *Proc. ECCV*, volume 3, pages 225–239, 2002.
- [8] J. Garding. Shape from texture for smooth curved surfaces. In *European Conference on Computer Vision*, pages 630–638, 1992.
- [9] J. Garding. Surface orientation and curvature from differential texture distortion. In *Int. Conf. on Computer Vision*, pages 733–739, 1995.
- [10] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer Verlag, 2001.
- [11] S. Lazebnik, C. Schmid, and J. Ponce. Affine-invariant local descriptors and neighborhood statistics for texture recognition. In *Int. Conf. on Computer Vision*, 2003.
- [12] S. Lazebnik, C. Schmid, and J. Ponce. Sparse texture representation using affine-invariant neighborhoods. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2003.
- [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] David G. Lowe. Object recognition from local scale-invariant features. In *Int. Conf. on Computer Vision*, pages 1150–1157, 1999.
- [15] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Computer Vision*, 2003. submitted.
- [16] 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.
- [17] J. Malik and R. Rosenholtz. Computing local surface orientation and shape from texture for curved surfaces. *Int. J. Computer Vision*, pages 149–168, 1997.
- [18] K. Mikolajczk and C. Schmid. A performance evaluation of local descriptors. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2003.
- [19] R. Rosenholtz and J. Malik. Surface orientation from texture: isotropy or homogeneity (or both)? *Vision Research*, 37(16):2283–2293, 1997.
- [20] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):530–534, May 1997.
- [21] D. Terzopolous, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. *Computer Graphics (SIGGRAPH 87 Proceedings)*, pages 205–214, 1987.
- [22] B. Triggs. Autocalibration from planar scenes. In *Proc ECCV*, volume 1, pages 89–105, 1998.
- [23] A.P. Witkin. Recovering surface shape and orientation from texture. *Artificial Intelligence*, 17:17–45, 1981.