Shape from non-homogeneous, non-stationary, anisotropic, perspective texture

Angeline M. Loh[†] and Richard Hartley[‡]
[†]The University of Western Australia,
[‡]Australian National University, and National ICT Australia
angie@csse.uwa.edu.au, Richard.Hartley@anu.edu.au

Abstract

We present a method for Shape-from-Texture in one of its most general forms. Previous Shape-from-Texture papers assume that the texture is constrained by one or more of the following properties: homogeneity, isotropy, stationarity, or viewed orthographically. We make none of these assumptions. We do not presume that the frontal texture is known a priori, or from a known set, or even present in the image. Instead, surface smoothness is assumed, and the surface is recovered via a consistency constraint. The key idea is that the frontal texture is estimated, and a correct estimation leads to the most consistent surface. In addition to surface shape, a frontal view of the texture is also recovered. Results are given for synthetic and real examples.

1 Introduction

Shape-from-Texture aims to estimate the shape of a surface based on cues from markings on the surface, or its texture. The problem is ill-posed unless some assumptions are made. This work avoids making some of the usual assumptions regarding texture, by only assuming surface smoothness and performing an efficient search for the frontal texture.

1.1 Basic Idea

Suppose that a texture is composed of individual texels, and that one texel is known to be viewed frontally in an image. Other texels may be slanted away from the camera, and lie at a different distance from the camera than the frontal texel. Slanting the texel will cause foreshortening in the tilt direction[1], and changing distance from the camera will result in a change of scale. The appearance of any texel will be related to the appearance of the frontal texel by a geometric transform, which locally can be modeled as an affine transformation. By measuring at each texel the affine transformation that relates that local texel to the known frontal texel, we may determine the orientation of the surface at that point in the image. From this information it is possible to reconstruct the surface.

Sometimes however, no frontal texel is visible, or identifiable in the image. Our approach is to search over all possible texels to determine which is frontal. Each hypothesized frontal texel leads to an estimate of the orientation across the surface. However,

[1] Slant is defined as the angle between the line of sight and the surface normal. Tilt is the direction in which the surface is slanted.

BMVC 2005 doi:10.5244/C.19.8

the key observation of this paper is that an incorrect hypothesis of the frontal texel leads to estimates of surface orientation that are inconsistent, and in fact can not be realized by a reconstructed surface. Therefore, a search over all possible frontal texels leads to a unique consistent estimate of frontal texel, and hence surface shape. We demonstrate that this is feasible using the orientation cue. The search is carried out using the Levenberg-Marquardt method, and does not require that the frontal texel be visible in the image.

It has been shown by Forsyth [5], that under orthographic projection, the frontal texel can be determined regardless of the shape for the surface. However, his approach cannot be applied in the case where individual texels may be scaled due to perspective views or for any other reason. The contribution of this work is to provide a Shape-from-Texture method that does not assume that the texel is homogeneous, isotropic, or stationary, or that we are using an orthographic view. To our knowledge, no algorithm exists for this level of generality.

1.2 Defining Assumptions

In this work, the assumptions regarding texture are defined as such: **Homogeneity** means that the distance between texels and their pattern of placement is consistent across the surface. The term refers to the *location* of texels, without regard to the rotation of each texel. **Stationarity** means that the texels differ by a translation on the surface but not a rotation. **Isotropy** means that the texel has a constant inertia about every axis.

2 Relevant Work

Interest in Shape-from-Texture is said to have started with Gibson's '*The Perception of the Visual World*' in which he proposes that texture provides an important visual shape cue. Despite the large number of papers in Shape-from-Texture produced since Gibson's work, there are curiously few methods that actually estimate the shape of a surface based on its texture [5].

There are still fewer methods that may be commonly applied to images of *real* surfaces. This is due to restrictive assumptions made about either the texture or camera model. All Shape-from-Texture algorithms begin with some assumption about the texture. Witkin [19] and Brown [1] assume that the texture is isotropic, although this is in reality rarely the case. Texture homogeneity is more frequently seen in the literature, for example in the work of Kanatani [9], Lee [11] and Stone [17]. Work by Gårding [6], Malik [13] and Clerc [3] assume stationarity, so these algorithms cannot be used when the texture is allowed to rotate on the surface.

Assumptions about the camera model can also limit the images to which the algorithm may apply. An orthographic camera model requires that the visual angle is small compared to the distance of the object from the camera. Choe [2] and Super [18] for example make this assumption. When perspective effects, such as the shrinking of a texture as it moves away from the camera, are non-negligible an appropriate assumption is the perspective or pin-hole camera model, as used by Clerc [4] and Jau [8].

The method described in this paper solves the Shape-from-Texture problem for perspective views. The mathematical formulation in Section 3.1 is related to that found in Forsyth's paper [5] except that Forsyth uses an orthographic camera model. An additional contribution of this work is the surface consistency measure, which allows us to estimate

the frontal texture even when individual texels are scaled with respect to the frontal texel — a property not accounted for in the orthographic case.

Surface consistency measures have been used to constrain object shape in areas other than Shape-from-Texture. For example see the work of Yuille et al. [20] who use a different cost function to the one described in this paper.

3 Overview of the New Method

We begin with the image of a smooth textured surface. The shape of the surface is found by firstly searching for the frontal texel, as described in Section 3.1. The search aims to find the texel that leads to the best surface consistency measure, described in Section 3.2. Surface shape is then estimated using the procedure in Section 3.3.

3.1 Searching for the Frontal Texel

Write $T_{f\to i}$ as the affine transformation from the frontal texel f to the texel i. The aim is to determine $T_{f\to i}$ for all i, as these matrices may be decomposed to give the orientation and distance of each i.

Since we do not know the appearance of the frontal texel, we can not compute the matrices $T_{f\to i}$ directly from the image. Instead, we can write

$$T_{f\to i} = T_{j\to i} T_{f\to j} \tag{1}$$

where $T_{j\to i}$ is the transformation from a reference texel j, arbitrarily chosen as a texel from the image (or some affine transform of it), to texel i, and $T_{f\to j}$ gives the affine transformation from f to j. The matrices $T_{j\to i}$ may be computed from the image, and it is

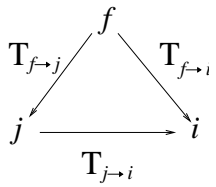


Figure 1: Transformations between texels

then a matter of searching for the affine transformation $T_{f\to j}$ such that the most consistent surface is achieved from the matrices $T_{j\to i}T_{f\to j}$. Our measure of surface consistency is given in the next section. The matrix $T_{f\to j}$ may be decomposed via RQ decomposition [7] to give $T_{f\to j} = RQ$ where R is a lower triangular matrix and Q is orthogonal. We are free to replace $T_{f\to j}$ with $T_{f\to j}P$ where P is a rotation matrix, as an initial rotation does not alter our interpretation of the surface orientation, and hence does not affect our measure of surface consistency. We choose $P = Q^T$ so that we need only search lower triangular matrices as candidates for $T_{f\to j}P$. We can further reduce our search space by the fact that our surface consistency measure does not depend on scale, so that we only need to search for matrices of the form $\begin{pmatrix} 1 & 0 \\ c & b \end{pmatrix}$. Using a Levenberg-Marquardt search, we determine

the values of c and b that give the most consistent surface, c_f and b_f. The transformations $T_{f\to i}$ are then given by $T_{j\to i}\begin{pmatrix} 1 & 0 \\ c_f & b_f \end{pmatrix}$.

3.2 Surface Consistency Measure

We can test whether our transformations $\hat{T}_{f\to i}$ (where the hat denotes the fact that $\hat{T}_{f\to i}$ is only a candidate) give a consistent surface by doing the following: first, calculate the surface normals that arise from the transformations $\hat{T}_{f\to i}$. This is done by decomposing $\hat{T}_{f\to i}$ via singular value decomposition such that

$$\begin{aligned} \hat{T}_{f\to i} &= UDV^T \\ &= UDU^T(UV^T) \end{aligned} \tag{2}$$

where D is a diagonal matrix satisfying $D_{11} < D_{22}$, and U and V are orthogonal matrices. The decomposition is interpreted as follows.

1. The texel is rotated by the rotation UV^T.
2. The image is now rotated by U^T, stretched in the coordinate directions by D and then rotated back by the rotation U. The nett effect of this is to scale it along two axis directions whose orientation is determined by U.
3. Thus, U gives the tilt direction.
4. The amount of scaling (given by the entries of D) indicates the degree of slant.

Note that the degree of tilt or slant of the image does not depend on V. Hence we may ignore it. Alternatively, we can multiply on the right by any rotation, without changing the result of the slant or tilt calculation. If the slant and tilt of the surface patch are given by σ and τ respectively, then

$$\begin{aligned} \cos(\tau) &= U_{11}, \\ \sin(\tau) &= U_{12}, \end{aligned} \tag{3}$$

$$\begin{aligned} \cos(\sigma) &= D_{11}/D_{22}, \\ \sin(\sigma) &= \sqrt{1-\cos^2(\sigma)}. \end{aligned} \tag{4}$$

Equation 3 comes from the fact that

$$U = \begin{pmatrix} \cos(\tau) & \sin(\tau) \\ -\sin(\tau) & \cos(\tau) \end{pmatrix} \tag{5}$$

and Equation 4 arises from

$$D = \begin{pmatrix} \cos(\sigma)r & 0 \\ 0 & r \end{pmatrix} \tag{6}$$

where r is the scaling factor. The surface normal is then given by

$$\begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix} = \begin{pmatrix} \pm\cos(\tau)\sin(\sigma) \\ \pm\sin(\tau)\sin(\sigma) \\ \cos(\sigma) \end{pmatrix}. \tag{7}$$

From the surface normals, the gradients are calculated for each patch:

$$f_x = -n_x/n_z \text{ and } f_y = -n_y/n_z$$

The Fundamental Theorem of Line Integrals states that

$$\| \int_C \nabla z.dr \| = 0 \tag{8}$$

for all closed curves C parameterized by r and a differentiable function z whose gradient ∇z is continuous on C. Using our calculated surface gradients as ∇z in Equation 8 and setting C to be any closed curve, we would expect that the left-hand-side of Equation 8 would be close to zero if we had correctly estimated the frontal texel, and much larger if we had not. Therefore, a measure of surface inconsistency is obtained by summing the left-hand-side of Equation 8 around various loops. The loops used in our method are triangles obtained by choosing sets of three neighboring texels and using their coordinates as the corners of the triangles. The cost term is then a discrete approximation to a continuous integral, given by

$$\begin{aligned} cost \quad &= \sum_{i=1}^{N} \quad (f_{x,i,1}, f_{y,i,1}).(x_{i,2} - x_{i,1}, y_{i,2} - y_{i,1}) \\ &+ \qquad (f_{x,i,2}, f_{y,i,2}).(x_{i,3} - x_{i,2}, y_{i,3} - y_{i,2}) \\ &+ \qquad (f_{x,i,3}, f_{y,i,3}).(x_{i,1} - x_{i,3}, y_{i,1} - y_{i,3}) \end{aligned} \tag{9}$$

where N is the number of loops, $(x_{i,n}, y_{i,n})$ is the location of the nth texel in the ith loop, and $(f_{x,i,n}, f_{y,i,n})$ are the corresponding gradients in the x and y directions.

We use this cost term with the Levenberg-Marquardt method [14] to search for the values of c and b that result in the most consistent surface.

3.3 Estimating Shape Using the Frontal Texel

Once our frontal texel has been determined, we use it to estimate the surface shape. This is done by decomposing the transformations $T_{f \to i} = T_{j \to i} \begin{pmatrix} 1 & 0 \\ c_f & b_f \end{pmatrix}$ via equations 2, 3 and 4. Then the orientation at each texel is specified by

$$\begin{aligned} \tau &= \arctan(U_{12}/U_{11}), \text{ and} \\ \sigma &= \arccos(D_{11}/D_{22}) \end{aligned} \tag{10}$$

4 Results

The algorithm was firstly tested on a synthetic image of a sphere. The texel used to cover the surface is shown in Figure 3(a). This particular texel was chosen to demonstrate that the algorithm works even if the texture is anisotropic. The texel also has many right angles which clearly expose any potential error in the estimate of the frontal texel.

In order to place the texels, the frontal version was randomly rotated about its normal axis, and placed at random locations on the surface such that its surface normal matched that of the surface under it. The texels were also scaled to account for the shrinking effect as texels move away from the camera. The textured surface shown in Figure 2 is a non-homogeneous, anisotropic and non-stationary texture, with simulated perspective effects. The only input into our Shape-from-Texture algorithm was the image of the textured surface. Individual texels were found using the Maximally Stable Extremal Regions (MSER) detector by Matas et al [15]. True texels were distinguished from other detected regions using SIFT features [12] to initially find the "average" region. Regions were then rejected

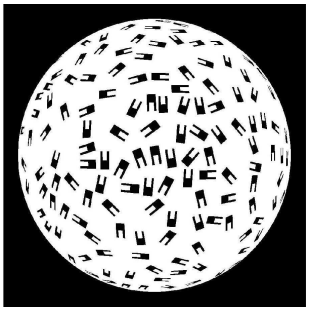


Figure 2: The textured surface

as true texels if their distance function from the average exceeded a threshold. While a more sophisticated method of pruning false texels (such as clustering) is clearly better, the method described was found to be sufficient for our purposes.

In order to search for the frontal texel, we first needed to choose a reference texel j. This texel j is later transformed by $\begin{pmatrix} 1 & 0 \\ c_f & b_f \end{pmatrix}^{-1}$ to give the true frontal texel. We chose j to be an arbitrary texel that has been transformed to make it isotropic, that is, it has constant second moment about every axis. This is a natural choice for j since *all* visible texels need only be foreshortened to achieve a rotated and scaled version of j. The texel j is shown in Figure 3(b).

The transformation $T_{j\to i}$ for every texel i is found as follows: i is first foreshortened to make it isotropic, and this is rotated and scaled to become j. The inverse of this process gives $T_{j\to i}$. This method of computing transformations was influenced by Schaffalitzky and Zisserman's work [16] using isotropic textures as a common point of reference.

Then to find the values of c_f and b_f an initial coarse search in (c,b) space was performed to find some near-optimal values. This was followed by a Levenberg-Marquardt optimization starting from these values to find the optimal solution. The estimated frontal texel which resulted is shown in Figure 3(c). The estimate looks good, given that its rotation and scale compared to the true frontal in Figure 3(a) are of no consequence, since these factors play no role in determining surface orientation.

The surface orientation at each texel can then be calculated. This is demonstrated with the needle diagram shown in Figure 4(a). Note that the well-known tilt ambiguity can be resolved here since during the placement of texels, scaling occurred to account for the distance of each texel from the theoretical camera. The scale at each texel can be deduced from the transformation $T_{f\to i}$. The tilt ambiguity can be resolved by choosing the candidate for tilt which is in the direction of decreasing scale. The slant and tilt values were interpolated at values in a regular grid and the surface was reconstructed using the algorithm by Kovesi [10]. This surface is shown in Figure 4(b).

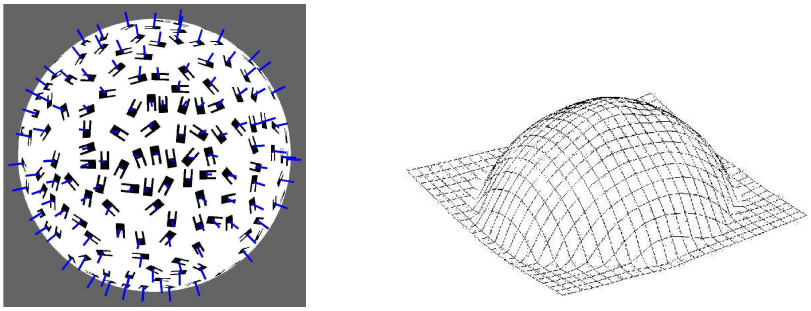
Next the algorithm was tested on a real image. Fabric was draped over the back of a chair to form a the curved textured surface seen in Figure 5. This particular fabric was chosen because it demonstrates that the algorithm works on non-stationary and anisotropic textures. The image has clear perspective effects, seen by the decreasing scale of the leaves as they move away from the camera. Again, the reference texel j was chosen to be an arbitrary texel that has been transformed to make it isotropic, as shown in Figure 6(a).

(a) True frontal

(b) Reference texel chosen arbitrarily

(c) Frontal estimated by algorithm

Figure 3: The estimated frontal shown in (c) is a good match with (a), since the scale and rotation of (c) compared to (a) make no difference to the recovered surface and hence the surface consistency measure.



(a) (b)

Figure 4: (a) Needle diagram showing the calculated surface orientation at each texel. (b) Calculated mesh surface.

The transformations $T_{j\to i}$ are found as before: each visible texel (such the one shown in Figure 6(b)) is firstly made isotropic (Figure 6(c)), and this is rotated by the angle that makes it most similar to the reference texel (Figure 6(d)). Therefore the sequence (b)–(d) shows the transformation $T_{i\to j}$; the inverse transformation gives $T_{j\to i}$. As before, an initial coarse search in (c,b) space yields near optimal values for surface consistency, which are then refined with a Levenberg-Marquardt method. Figure 7(a) displays the frontal texel as estimated by the algorithm. It is a transformed version of the reference texel shown in Figure 6(a). A real image of the fabric when viewed frontally is shown in Figure 7(b). For comparison, and considering that initial rotations play no role in shape information, we have rotated the image in (b) so that it matches the orientation in (a).

The estimated frontal texel is used to calculate the orientation at each texel. This is shown in a needle diagram in Figure 8(a). Kovesi's method [10] is used to construct the surface seen in Figure 8(b). A mesh surface viewed from the side is shown in Figure 8(c).

Note that in the example above, there is no ambiguity when calculating the transformations $T_{j\to i}$. However ambiguities can occur if the texel has rotational symmetry; for example, if the frontal texel is a rectangle (as in the case of bricks) then our reference texel

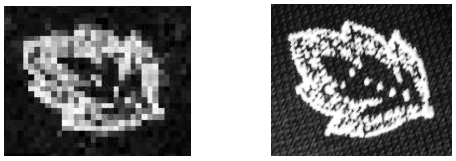
Figure 5: Real image: fabric with a non-stationary, anisotropic texture.

(a) Reference texel (b) Some visible texel (c) Isotropic version of (b) (d) Rotated version of (c) to match (a)

Figure 6: Calculating the transformations $T_{i\to j}$. The sequence (b)–(d) shows how some visible texel is transformed into the reference texel shown in (a).

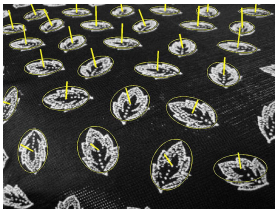
j and all visible texels i are trapeziums. It is impossible to know which of the four sides of j correspond to the four sides of each i. Furthermore, a circular frontal texel presents infinite possibilities when calculating the transformations $T_{j\to i}$ because an appropriate scaling factor can be used to compensate for any rotation component. This problem is intrinsic to all perspective views when the transformation between textures is calculated, however it is only a problem when the textures exhibit rotational symmetry.

So far in this work, we have described textures in terms of distinct texels, but in principle our algorithm will work for textures where one cannot distinguish individual texels; interest point locators such as Lowe's [12] might be used to detect corresponding points in the repeated pattern, and a second moment matrix assigned to each point. Transformations between textures at different points would be determined by the transformations between their second moment matrices.

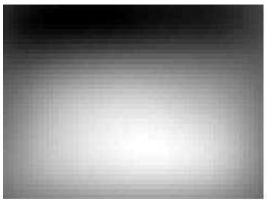


(a) Frontal estimated by algorithm (b) Real image of fabric viewed frontally

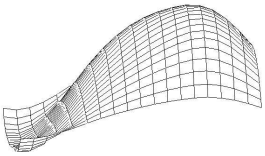
Figure 7: Estimated and true frontal texels.



(a) Needle diagram also showing detected texels



(b) Calculated surface height shown in gray value



(c) Mesh surface seen from side view

Figure 8: Estimated surface shape.

5 Conclusion and Future Work

The real example given in this paper demonstrates the robustness of our algorithm; a single image of a non-stationary, anisotropic texture is the only input to the method. The frontal texture is unknown, and we do not even know if any frontal texture appears in the image. There are also considerable perspective effects. The algorithm firstly estimates the frontal texture using a smoothness constraint. Our results show an accurate estimate of the frontal texture. Then, the surface shape can be estimated. The transformations from the frontal texture to all other viewed textures can be calculated, and these transformations interpreted to give shape information. Our real example gives a qualitatively good surface reconstruction. This work contributes a Shape-from-Texture method that is not restricted to homogeneous, isotropic or stationary textures, or an orthographic view. To our knowledge no other method works with this level of generality. Future work will explore the effect of noise in $T_{j\to i}$ (the affine transformations between texels), as well as the effects of surface smoothness, choice of reference texel and reducing the texel density.

6 Acknowledgments

Many thanks to Andrew Zisserman for his useful feedback and conversation, especially concerning the estimation of affine transformations. This work was funded by an Australian Postgraduate Award and a Western Australian IVEC Doctoral Scholarships top-up. The work was largely carried out at the National ICT Australia (NICTA) site, Canberra. NICTA is funded by the Australian Government's Backing Australia's Ability initiative, in part through the Australian Research Council. Many thanks to the Visual Geometry Group at Oxford University who provided helpful ideas. The authors acknowledge the assistance of the University of Western Australia Graduates Association for partially funding research undertaken in the UK through the Bankwest Travel Award.

References

[1] L.G. Brown and H. Shvaytser. Surface orientation from projective foreshortening of isotropic texture autocorrelation. *PAMI*, 12(6):584–588, June 1990.

[2] Y. Choe and R.L. Kashyap. Shape from texture and shaded surface. In *ICPR90*, pages Vol–I 294–296, 1990.

[3] M. Clerc and S. Mallat. Shape from texture through deformations. In *ICCV99*, pages 405–410, 1999.

[4] M. Clerc and S. Mallat. The texture gradient equation for recovering shape from texture. *PAMI*, 24(4):536–549, April 2002.

[5] D.A. Forsyth. Shape from texture without boundaries. In *ECCV02*, page III: 225 ff., 2002.

[6] J. Gårding. Shape from texture for smooth curved surfaces. In *ECCV92*, pages 630–638, 1992.

[7] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[8] J.Y. Jau and R.T. Chin. Shape from texture using the wigner distribution. *CVGIP*, 52(2):248–263, November 1990.

[9] K.I. Kanatani and T.C. Chou. Shape from texture: General principle. *AI*, 38(1):1–48, February 1989.

[10] P. Kovesi. Surface normals to surfaces via shapelets. In *Proceedings Australia-Japan Advanced Workshop on Computer Vision*, pages 101–108, 2003.

[11] K.M. Lee and C.C.J. Kuo. Direct shape from texture using a parametric surface model and an adaptive filtering technique. In *CVPR98*, pages 402–407, 1998.

[12] D. Lowe. Distinctive image features from scale invariant keypoints. In *IJCV04*, pages 91–110, 2004.

[13] J. Malik and R. Rosenholtz. Computing local surface orientation and shape from texture for curved surfaces. *IJCV*, 23(2):149–168, June 1997.

[14] D. W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *JSIAM*, 11(2):431–441, June 1963.

[15] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *BMVC02*, pages 384–393, 2002.

[16] F. Schaffalitzky and A. Zisserman. Viewpoint invariant texture matching and wide baseline stereo. In *ICCV01*, pages 636–643, 2001.

[17] J.V. Stone and S.D. Isard. Adaptive scale filtering: A general-method for obtaining shape from texture. *PAMI*, 17(7):713–718, July 1995.

[18] B.J. Super and A.C. Bovik. Shape from texture using local spectral moments. *PAMI*, 17(4):333–343, April 1995.

[19] A.P. Witkin. Recovering surface shape and orientation from texture. *AI*, 17(1-3):17–45, August 1981.

[20] A.L. Yuille, D. Snow, R. Epstein, and P.N. Belhumeur. Determining generative models of objects under varying illumination. *IJCV*, 35(3):203–222, December 1999.