Shape from Multiple, Transparent, and Occluded Textures

Michael J. Black and Ruth Rosenholtz

Xerox Palo Alto Research Center
3333 Coyote Hill Road, Palo Alto, CA 94304

Phone: (650)812–4745
FAX: (650)812-4334
Email: {black,rruth}@parc.xerox.com

Abstract

This paper explores the estimation of surface shape from texture when multiple textures are present in an image region due to texture discontinuities, occlusion, or pseudo-transparency (e.g. grass seen through a fence). We present both a theoretical analysis of the multiple texture problem and an algorithm that relaxes the *single texture assumption*. Using an analogy between shape-from-texture and motion, we exploit a robust mixture model to recover multiple surface shape estimates in the presence of multiple textured surfaces. The accuracy of the method is comparable that of shape-from-texture algorithms applied to a single texture.

Submitted as a **Short Paper.**

Index Terms: Shape from texture, multiple motions, layers, occlusion, transparency, robust mixture models.

Contact Information:

Michael J. Black
Xerox Palo Alto Research Center
3333 Coyote Hill Rd
Palo Alto, CA 94304

Phone: (650) 812–4745
FAX: (650) 812–4334
Email: black@parc.xerox.com

1 Introduction

Shape from texture methods have typically assumed a prior segmentation of the image into regions containing only a single texture. This "single texture" assumption is commonly violated in natural images. Segmentation into single texture regions may be difficult with current techniques prior to the recovery of surface shape particularly in cases where multiple textures are superimposed due to transparency or occlusion. In general, we would like shape-from-texture methods to be able to cope with the following violations of the single texture assumption (see Figure 1):

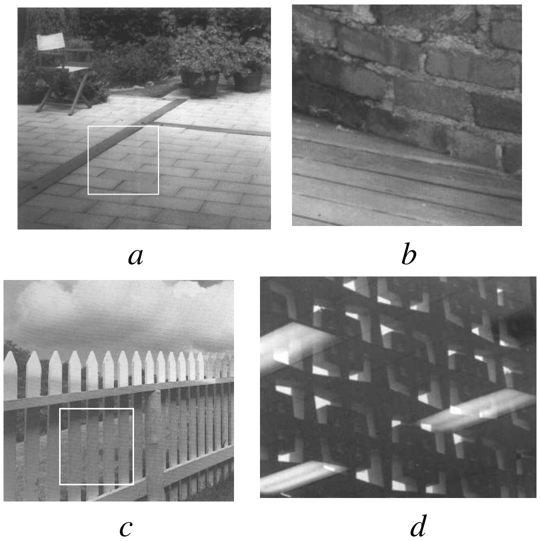


Figure 1: Multiple textured surfaces that violate the single texture assumption.

(a) Texture Outlier: An image patch contains a brightness structure that is not consistent with the dominant texture.

(b) Texture Discontinuity: Multiple textures within a single image region.

(c) Fragmented Occlusion or Pseudo-Transparency: One surface is viewed *through* another surface such as a fence.

(d) Transparent/Reflected Textures: eg. viewing a textured surface through glass which is also reflecting a textured surface. Without additional cues such as motion, images of this sort can be confusing to human observers. We leave this problem for future work.

We take as our basic model the shape-from-texture methods based on local distortions of the spatial frequency of the texture [8, 9, 10]. Given the spatial frequency representation of a texture in an image patch, these methods find the texture distortion, modeled as an affine transform, between one image patch and a neighboring image patch. We discuss Malik and Rosenholtz' work, upon which our algorithm is based, in more detail in Section 2.

Krumm and Shafer's work [9] did address the issue of segmentation using shape-from-texture. They first found shape from texture, and then used the shape estimates to undo the perspective distortion of the texture patches before performing segmentation on the undistorted textures. *However*, by assuming a single texture in each pair of image patches, they in effect assume that the scene had already been segmented.

In this paper we extend previous approaches by relaxing their "single texture assumption." We do this by making use of an analogy between shape-from-texture and structure from stereopsis and motion [10]: one can think of each patch of texture as a frame in a motion sequence, or as the image in one eye in binocular stereopsis.

Unlike shape-from-texture, recent work in motion has focused on recovering multiple motions due to motion discontinuities, occlusion, and transparency. A number of approaches represent the motion of an image region as a set of "layers," where the motion of each layer is described by a low-order parametric model (eg. affine) [1, 3, 5, 12]. Following the analogy between shape-from-texture and structure from motion, we exploit techniques from robust estimation of multiple motions to recover multiple surface shape estimates in the presence of multiple textured surfaces.

In Section 3 we will show that in certain multiple-texture situations, the spatial frequency representation approximates a (possibly weighted) sum of the significant frequency components corresponding to each of the different textures. We refer to multiple textures of this form as "additive" in the frequency domain.

In the additive case, a spatial frequency distribution containing peaks from multiple textures can be thought of as consisting of a number of layers where each layer corresponds to the spatial frequencies present in a single texture (see Figure 2). When textures are additive in the frequency domain we can apply tools developed for dealing with multiple motions to the new problem of multiple textures. We describe our algorithm in Section 4. Section 5 presents experimental results on real and synthetic images.

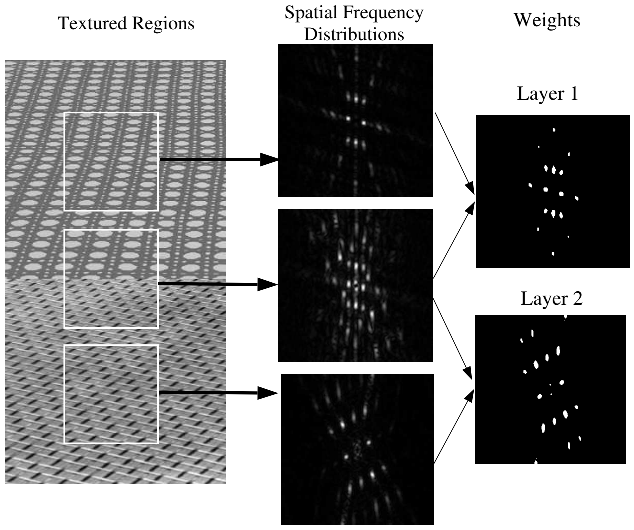


Figure 2: Additive texture split into layers.

2 Shape-from-Texture

We begin with the basic approach of Malik and Rosenholtz [10] which uses a modified version of windowed Fourier transform magnitude [10] as a local spectral estimation method, i.e. as a measure of the texture statistics within an image region. They model the local *texture distortion* as a set of affine transforms, A_i, between the spectral estimate of an image patch and of each neighboring patch. They derived the relationship between these affine transforms and the local shape and orientation of the surface. The surface parameters of interest are the slant, σ, the direction of tilt, $\mathbf{t}$, and the curvature parameters, $r\kappa_t$, $r\kappa_b$, and $r\tau$. Slant is the angle between the surface normal vector and the line of sight. Here κ_t and κ_b are the normal curvatures of the surface in the tilt direction $(\mathbf{t})$ and the direction perpendicular to the tilt direction $(\mathbf{b})$, respectively. The variable r is the distance from the center of the viewing sphere to the given point on the surface, and τ is the "geodesic torsion" of the surface in the tilt direction. For the mathematical relationship between the local affine transformations and the surface shape parameters see [10].

A differential method, resembling those used in optical flow estimation, is used to find the affine transforms A_i They assume that one spectrogram differs from another by only a 2×2 affine transformation $A = I + \Delta A$, so $F_2(\vec{\omega}) = F_1(A\vec{\omega})$, where F_i is the spectrogram for a patch centered about the ith point, and $\vec{\omega}$ is the frequency. Then, for small ΔA, one can write

$$F_2(\vec{\omega}) - F_1(\vec{\omega}) \approx \nabla \vec{F}_1 \circ \Delta A \vec{\omega} \tag{1}$$

where $\nabla \vec{F}_1$ is the gradient of the spectrogram at the given frequency. Malik and Rosenholtz solve for ΔA using least squares. We will replace their estimation technique with a robust technique that can simultaneously estimate multiple transformations.

Finally, Malik and Rosenholtz use non-linear minimization of the error between the n measured affine transformations and the theoretical affine transformations (for a given shape estimate) to solve for shape and orientation estimates:

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

where $\hat{A}_i(k,l)$ is the (k,l)th element of the theoretically predicted matrix $\hat{A}_i$ and $\bar{A}_i$ is the ith empirically measured affine transform matrix. $\hat{A}_i$ is a function of the shape parameters, the focal length of the camera, and the distance in the image between the ith pair of patches.

3 Theoretical Analysis

In this section we study the conditions under which two textures may be approximated as "additive" in the frequency domain. In these conditions, the problem of multiple textures may be treated like the problem of multiple motion layers.

The canonical example for shape from texture for multiple textures with occlusion is the fragmented occlusion example shown in Figure $1c$. We will refer to fragmented occlusion using terminology suggested by the example shown of a wood fence in front of a grassy field, though of course the analysis applies in general when one texture partially occludes another. In space domain, we have

$$\mathbf{image} \;=\; \mathbf{wood\ texture} \times \mathbf{where\ the\ fence\ is} + \mathbf{grass\ texture} \times \mathbf{where\ the\ fence\ isn't}$$

where **wood texture** is the texture actually on the boards of the fence and **where the fence is/isn't** are binary functions indicating the location of the boards of the fence and the spaces between them,

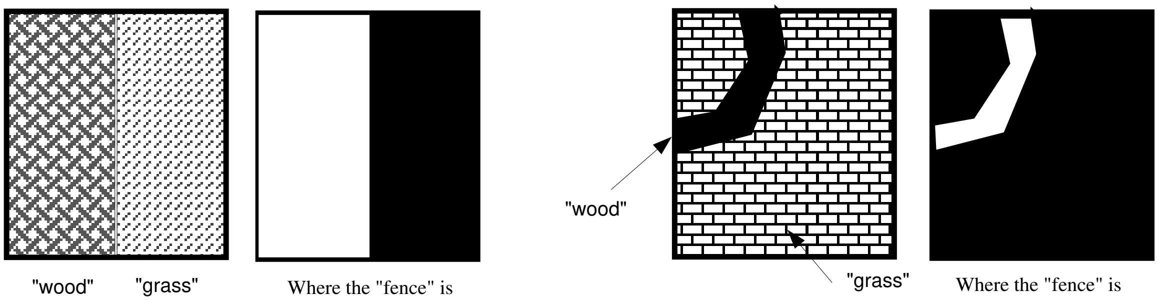


Figure 3: Texture with discontinuity (left) and with occlusion or outliers (right) can be thought of as special cases of the fragmented occlusion problem.

respectively. Transforming into frequency domain, and letting W, G, F and $\overline{\mathrm{F}}$ be the Fourier transforms of **wood texture**, **grass texture**, **where the fence is** and **where the fence isn't**, respectively, we get

$$F(\mathbf{image}) = \mathrm{W} * \mathrm{F} + \mathrm{G} * \overline{\mathrm{F}}. \tag{2}$$

At the scale shown in the figure, one really wants to know the orientation and shape of the fence and the orientation and shape of the grassy field, but not the shape and orientation of the individual boards that make up the fence. Therefore, we are interested in when Equation 2 can be approximated by the "additive" equation

$$F(\mathbf{image}) = k_1\mathrm{F} + k_2\mathrm{G} \tag{3}$$

for some constants, k_1 and k_2.

Note that the cases of texture discontinuity and shape from texture with occlusion are basically subsets of this canonical example of fragmented occlusion, as depicted in Figure 3.

3.1 When is the "Additive" Approximation Valid?

We need a model of the subsequent processing in the Fourier domain in order to define what we mean by a valid approximation. We will assume that all frequency components that have magnitude which is N or more times lower than the magnitude of the largest frequency component are suppressed, as in [10]. For this analysis, we will assume that the textures are roughly periodic, with well-defined peaks in frequency domain at frequencies $\vec{\omega}_i$, for some i.

3.1.1 Fragmented Occlusion.

To simplify the situation somewhat, we will assume that the wood texture, W, can be approximated by only a dc component. In other words, we assume that the boards of the fence are all untextured or at least that any "wood grain" texture on the boards is relatively low power.

Then, splitting $\overline{\mathrm{F}}$ into its dc component and the remaining frequency components, we can rewrite Equation 2 as

$$F(\mathbf{image}) = k_1\mathrm{F}(\vec{\omega}) + k_2\mathrm{G}(\vec{\omega}) + \sum_{\vec{\omega}_i \neq 0} k_i\mathrm{G}(\vec{\omega} - \vec{\omega}_i)$$

where k_1 and k_2 are constants representing the dc values of W and $\overline{\mathrm{F}}$, respectively. The first two terms are a weighted sum of F and G, as desired, but that we also have additional copies of the grass texture, G, convolved with the non-dc components of $\overline{\mathrm{F}}$. We have written these additional copies as $k_i\mathrm{G}(\vec{\omega} - \vec{\omega}_i)$. We not only have extra copies of the grass texture, but those extra copies effectively undergo different affine transforms because their affine transforms are centered about the frequencies $\vec{\omega}_i$.

Without loss of generality, we assume that the highest peak in frequency domain corresponds to the F texture. We refer to the dc component of any given texture, T, as $T(0)$. Then we write the highest peak in T, not counting the dc component, as $T(1)$, the next highest peak $T(2)$, and so on. Then the highest peak in the frequency domain (other than, perhaps, the dc component, which we ignore) has magnitude $|(\mathrm{W}(0) - \mathrm{G}(0))F(1)|$.

In order for the frequency components to be separable into two textures two conditions must be met. First the extra copies of the grass texture, G must be small enough that they are suppressed; i.e. the frequency components are lower than $1/N$ times the highest frequency component. Since the highest peak in all extra copies of the grass texture has magnitude $|\mathrm{G}(1)\mathrm{F}(1)|$, this can be expressed as the condition that

$$|\mathrm{W}(0) - \mathrm{G}(0)|/N > |\mathrm{G}(1)|. \tag{4}$$

Intuitively, this first condition states that the contrast between the wooden fence and the grass ($|W(0) - G(0)|$) must be high enough compared to the contrast of the grass texture ($|G(1)|$).

The second condition requires that we see enough frequency components in each texture to be able to estimate shape from texture. We must not suppress either the F texture or the first copy of the G texture and there must be at least n peaks in the first copy of both textures in order to be able to perform shape from texture on both textures. If we are not to suppress the first copy of the grass texture, we must have the following condition:

$$|[\mathrm{W}(0) - \mathrm{G}(0)]\mathrm{F}(1)|/N < |\mathrm{G}(n)\overline{\mathrm{F}}(0)|, \tag{5}$$

where $|\mathrm{G}(n)\overline{\mathrm{F}}(0)|$ is the magnitude of the nth highest peak associated with the first copy of the grass texture.

Intuitively, $\overline{\mathrm{F}}(0)$ is the percentage of pixels in the image in which one can see the grass texture. Therefore, for the fence example, this second constraint is satisfied if there is enough space between the boards of the fence for us to see "enough" of the grass to recover shape from the grass texture.

3.1.2 Texture Discontinuities and Outliers

The cases of shape from texture in the presence of a discontinuity of texture and in the presence of an occluder turn out to be much simpler. This is because both the discontinuity and the occluder tend to be low frequency; otherwise we would have the previous case of fragmented occlusion. Therefore, F and $\overline{\mathrm{F}}$ are each a single low frequency "blob" in frequency domain, and, Equation 2 becomes a blurred version of the desired $k_1\mathrm{W} + k_2\mathrm{G}$. When this additive approximation holds we should be able to separate the two textures in frequency domain and perform the shape from texture on both textured surfaces.

4 Estimating Multiple Affine Transformations

Consider an image patch, p, and some neighboring patch both containing n textured surfaces each having a different shape, surface orientation, or depth. Assume that the textures have distinct peaks in frequency domain. Let $m_{\vec{\omega},i}$ be the indicator function for the frequencies, $\vec{\omega}$, in patch p corresponding to textured surface i. For example, if $\vec{\omega}$ corresponds to texture i, then $m_{\vec{\omega},i}$ will be 1 and it will be 0 otherwise. Given different surface shapes, the frequencies corresponding to each texture will undergo a different affine transformation, ΔA, between patch p and its neighbor.

One can think of $m_{\vec{\omega},i}$ as assigning frequencies to one of a set of n layers. If we knew the assignment of frequencies to layers then we could estimate the affine transformation, ΔA_i, for layer i using Equation (1). The idea is to only consider those frequencies corresponding to layer i when estimating the the transformation ΔA_i. This is done using weighted least squares estimation in which we minimize

$$E(\Delta A_i) = \sum_{\vec{\omega}} m_{\vec{\omega},i} (\nabla \vec{F}_1(\vec{\omega})^T \circ \Delta A_i \vec{\omega} - (F_2(\vec{\omega}) - F_1(\vec{\omega})))^2 \tag{6}$$

with respect to the affine parameters ΔA_i for layer i. This can be repeated for each of the n layers.

The problem is then one of assigning frequencies to layers. Assume for the moment that the affine transformations of the layers are known. In this case the constraint

$$\nabla \vec{F}_1(\vec{\omega})^T \circ \Delta A_i \vec{\omega} - (F_2(\vec{\omega}) - F_1(\vec{\omega})) \tag{7}$$

will be close to zero for those frequencies that belong to layer i. We define a weight function $g(\cdot)$ that is maximized when the residual $\nabla \vec{F}_1(\vec{\omega})^T \circ \Delta A_i \vec{\omega} - (F_2(\vec{\omega}) - F_1(\vec{\omega}))$ is zero and goes to zero as the residual grows. We could take $g(\cdot)$ to be a normal distribution as in ([6, 11]) or a robust distribution as in ([2, 7]). We prefer a robust $g(\cdot)$ which falls off more rapidly than a normal distribution and hence we take

$$g(x, \sigma) = \frac{\sigma^2}{(\sigma^2 + x^2)^2} \tag{8}$$

where σ is a scale parameter. This weight function can be derived from the robust error norm used in [1].

We want the weights, $m_{\vec{\omega},i}$, for a particular frequency, to sum to one which means that every frequency is accounted for by some layer or combination of layers. Thus, given the affine transformations ΔA_i we compute the weights using the *mixture constraint*

$$m_{\vec{\omega},i} = \frac{g(\nabla \vec{F}_1(\vec{\omega})^T \circ \Delta A_i \vec{\omega} - (F_2(\vec{\omega}) - F_1(\vec{\omega})), \sigma)}{\sum_{i=1}^{n} g(\nabla \vec{F}_1(\vec{\omega})^T \circ \Delta A_i \vec{\omega} - (F_2(\vec{\omega}) - F_1(\vec{\omega})), \sigma)}. \tag{9}$$

Note that the weights assigning frequencies to layers have values between zero and one and that a frequency can be "shared" by multiple layers.

In practice we are not given $m_{\vec{\omega},i}$ or ΔA_i and must estimate them both. To do this we exploit the Expectation-Maximization (EM) algorithm which iterates between solving for the weights $m_{\vec{\omega},i}$ given an estimate of ΔA_i (the "E" step) and solving for the affine transformations ΔA_i given the weights (the "M" step) (see [11] for details).

Dealing with Outliers.

The mixture constraint assumes that every frequency belongs to some layer or is shared by multiple layers. It does not account for outliers which belong to no layer. To cope with outliers, we follow the approach in [4] and introduce a new *outlier layer* that does not correspond to any affine transformation and is not used in the minimization of Equation (6) but is used when enforcing the mixture constraint. If we are estimating n affine transformations then the normalization in Equation (9) is computed over the $n + 1$ layers

$$m_{\vec{\omega},i} = \frac{g(\nabla \vec{F}_1(\vec{\omega})^T \circ \Delta A_i \vec{\omega} - (F_2(\vec{\omega}) - F_1(\vec{\omega})), \sigma)}{\sum_{i=1}^{n} g(\nabla \vec{F}_1(\vec{\omega})^T \circ \Delta A_i \vec{\omega} - (F_2(\vec{\omega}) - F_1(\vec{\omega})), \sigma) + m_{\vec{\omega},n+1}}. \tag{10}$$

The initial value for the $m_{\vec{\omega},n+1}$ is taken to be a small value (eg. $g(2.5\sigma, \sigma)$).

The values of $m_{\vec{\omega},n+1}$ are updated only when normalizing the weights. If a frequency $\vec{\omega}$ does not correspond to any affine transformation ΔA_i then, the residual error will be high and it will receive low weights. The normalization step will automatically shift weight to the outlier layer and $m_{\vec{\omega},n+1}$ will increase.

Implementation.

Our implementation assumes that the number of layers is known. In all our experiments there were only two textures present at one time so a two-layer model (plus outliers) is sufficient. To estimate the affine transformation of each layer we first construct a Gaussian pyramid representation of the spatial frequency distributions under consideration. The affine transformation is computed at a coarse level and then, at the next finer level, the estimated transformation is used to register the two spatial frequency distributions by warping one towards the other. This process is repeated down to the finest level in the pyramid while the transformation is updated at each stage (see [1]).

Within each level of the pyramid the affine transformation is estimated using a gradient descent method embedded within a continuation strategy in which the value of σ begins at a high value and is gradually lowered [1]. The effect of this process is that initially all frequencies influence all layers and they are gradually "assigned" to different layers.

The algorithm begins with the weights, $m_{\vec{\omega},i}$ set to a small constant value. Using these initial weights, the first affine transformation is estimated using Equation (6). The weights are then updated using Equation (10) and then the second transformation is estimated. The weights are updated again and the process repeats for a fixed number of iterations or until a convergence criterion is met.

Recovering Multiple Shapes.

In the case of a single texture, Malik and Rosenholtz [10] recover surface shape by computing the transformations between a single patch and eight neighboring patches. In the case of multiple textures, each pair of image patches gives rise to n affine transformations. To estimate multiple shapes from these multiple transformations we need to know which transformations correspond to which of the n surfaces. We group the individual layers recovered from each of the eight neighboring patches into a single set of layers using a simple match metric that takes into account where the weights of the layers agree and where they disagree. Once the correspondence between the layers of the different patches is known, we use the shape estimation procedure described in Section 2 with the affine transformations corresponding to each set of layers to recover the multiple surface shapes.

5 Results

We present a number of results on natural and synthetic images that indicate the accuracy of the recovered surface shapes are comparable to those reported by Malik and Rosenholtz [10] for shape from texture with only one texture.

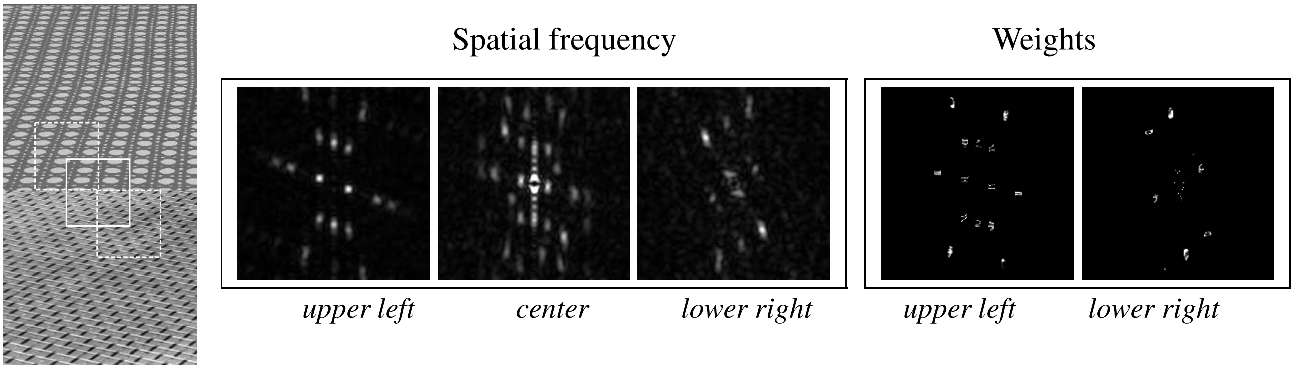


Figure 4: Texture Discontinuity Example. Left: Image showing example patch locations. Center: Spatial frequency distributions. Right: recovered weights (white indicates weights, $m_{\vec{\omega},i}$, near 1).

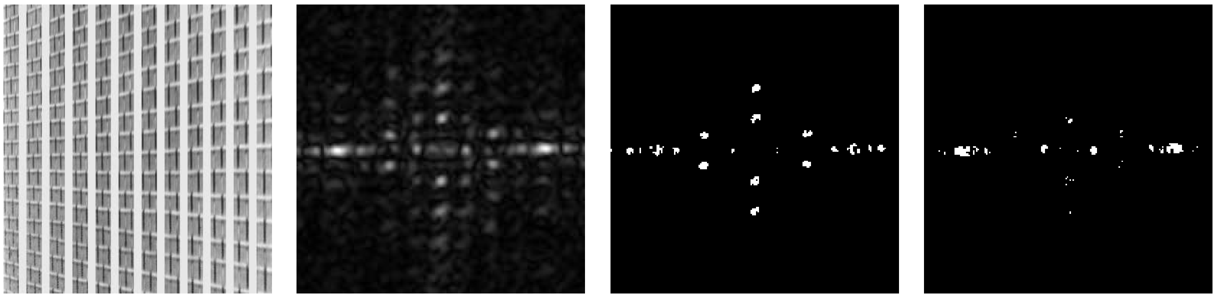
5.1 Texture Discontinuities

Figure 4 illustrates a synthetic texture discontinuity. Nine patches (128×128 pixels) are extracted with the central patch centered on the texture boundary. The neighboring patches are offset from the center by 64 pixels. A two level Gaussian pyramid was used with 40 iterations of the continuation method at each level. The value of σ began at 70.0 and was lowered by a factor of 0.9 after each iteration to a minimum of 35.0. Frequencies with magnitudes lower than 63.0 were ignored.

Figure 4 shows the spatial frequency distributions for the center patch and two neighboring patches. Notice how the center patch appears to be roughly the sum of the two neighboring patches. The algorthim described in the previous section estimates the affine transformations between this center patch and each of the neighbors and estimates the recovered weights, $m_{\vec{\omega},i}$, indicating which frequencies correspond to each transformation. Notice in Figure 4 how the recovered weights correspond to the different texture patterns. In this example the two neighboring patches shown have only a single texture pattern and hence the frequencies corresponding to the other pattern are treated as outliers.

Using the recovered affine transformations between the center patch and each of the eight neighboring patches we recover the following surface shape estimates for the top and bottom surfaces:

Surface	slant	tilt	rk_t	rk_b	$r\tau$
Top (True)	70	-90	0	0	0
Bottom (True)	70	-90	0	0	0
Top (Est.)	65.4	-86.7	-0.52	-0.16	-0.01
Bottom (Est.)	60.0	-93.7	-0.04	-0.06	-0.11



a *b* *c*

Figure 5: Fence Example. *(a)* Spatial frequency distribution of center patch. Weights corresponding to: *(b)* the background (grass). *(c)* the foreground (fence).

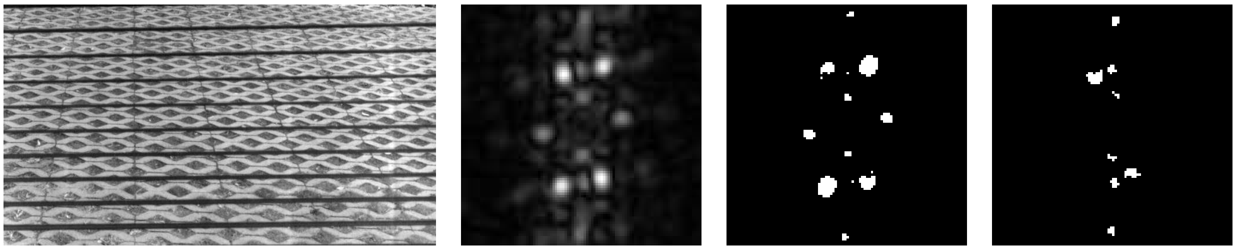
The slight underestimates of slant are due to the fact that the algorithm assumes that the offset in the image is the number of pixels (in this case 64) from the center of one patch to the center of another. The effective step, due to the discontinuity, is between the center of the *texture* in one patch to the center of the same texture in the other patch. The overestimate in the step size causes the algorithm to underestimate the rate of change of the texture, and thus the slant. From the Fourier transform phase and the segmentation in frequency domain of the two textures, one could conceivably both segment the textures in space domain and correct for these slant underestimates.

5.2 Synthetic Fence Example

The next experiment considers the fragmented occlusion case in Figure 5 in which we have a "fence" pattern in front of a textured surface (the "grass"). We use 128×128 pixel patches, with neighboring patches offset from the center patch by 32 pixels. A four level Gaussian pyramid was used with 20 iterations of the continuation method at each level. The value of σ began at 60.0 and was lowered by a factor of 0.95 after each iteration to a minimum of 20.0. Frequencies with magnitude lower than 90.0 were ignored.

Figure 5 shows the averaged weights from all the layers corresponding to one or the other surface. Notice that the power in the "fence" texture lies along a line in frequency domain. This means that the estimation of the affine transformations for the fence texture will be underconstrained. This has been referred to as the texture aperture effect [10].

The true and estimated parameters for the fence example are:



a *b* *c*

Figure 6: Venetian Blind Example. *(a)* Spatial fequency distribution of center patch. Weights corresponding to: *(b)* the background (ground plane). *(c)* the foreground (blinds).

Surface	slant	tilt	rk_t	rk_b	$r\tau$
"Grass" (True)	70	-180	0	0	0
"Fence" (True)	0	undef	0	0	0
"Grass" (Est.)	67.9	-185.4	-0.15	-0.05	-0.04
"Fence" (Est.)	8.7	38.6	-7.14	-4.54	16.7

Notice that the shape of the background textured surface has been recovered accurately. For the fence, the slant of 8.7 is fairly accurate while the true tilt is undefined for a slant of 0 degrees. The high curvature terms are due to the aperture effect. In general one should detect when the solution is ill-conditioned due to the texture aperture effect and indicate which results are reliable and which are not. A solution to this problem is beyond the scope of this paper.

5.3 Seeing Through Blinds

We now consider a natural scene exhibiting fragmented occlusion. Figure 6 shows a textured ground plane (the "grass") viewed through Venetian blinds (the "fence"). The patches are 128×128 pixels, with neighboring patches offset from the center patch by 64 pixels. Exactly the same parameters were used as were used in the previous synthetic example.

Figure 6 shows the spatial frequency distribution of the center patch and the recovered weights which, for display, are averaged across the corresponding layers of all the patches. Notice that the blinds, once again, suffer from the texture aperture problem while the peaks corresponding to the ground-plane texture are clearly recovered.

The recovered surface parameters are:

	Estimated				
Surface	slant	tilt	rk_t	rk_b	$r\tau$
"Grass"	58.3	-87.4	0.02	0.13	0.09
"Fence"	16.9	-147.7	2.28	2.38	10.99

While we do not have ground truth for this image, the "grass" texture results appear to accurately reflect the orientation of the ground plane. Likewise the slant of the blinds appears to be well recovered but due to the aperture problem we cannot recover the shape accurately without additional assumptions.

6 Conclusions

Previous work in shape-from-texture has assumed that, within the region of interest, there is only one texture present, and no occluders.

But natural scenes typically contain more than one textured object, and it is quite common to see one textured surface through another, as in viewing our canonical example of a grassy field through a fence. We have shown theoretically that for a large class of problems, the shape from texture problem for multiple textures and occlusion can be modeled using an additive layered representation. We have presented a robust shape from texture method based on a combination of Malik and Rosenholtz' [10] affine model of shape from texture and previous work in robust affine motion estimation [1, 5], and have demonstrated results on both synthetic and natural images which achieve the accuracy of methods that estimate shape from texture with only one texture.

On a final note, since the phase information is available, we can use the segmentation into layers in the frequency domain to reconstruct the original image patch in the spatial domain (though note that we do not know the DC value of the reconstructed texture). In our current work we are exploring the use of this layered segmentation in the frequency domain as a way of performing texture segmentation in the space domain.

References

[1] M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, January 1996.

[2] M. J. Black and R. Rosenholtz. Robust estimation of multiple surface shapes from occluded textures. In *International Symposium on Computer Vision*, pages 485–490, Miami, FL, November 1995.

[3] T. Darrell and A. Pentland. Cooperative robust estimation using layers of support. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):474–487, May 1995.

[4] S. Gold and A. Rangarajan. Graph matching by graduated assignment. In *Proc. Computer Vision and Pattern Recognition, CVPR-96*, pages 239–244, San Francisco, June 1996.

[5] A. Jepson and M. J. Black. Mixture models for optical flow computation. In *Proc. Computer Vision and Pattern Recognition, CVPR-93*, pages 760–761, New York, June 1993.

[6] A. Jepson and M. J. Black. Mixture models for optical flow computation. In Ingmer Cox, Pierre Hansen, and Bela Julesz, editors, Partitioning Data Sets: With Applications to Psychology, Vision and Target Tracking, pages 271–286, DIMACS Workshop, April 1993. AMS Pub., Providence, RI.

[7] S. X. Ju, M. J. Black, and A. D. Jepson. Skin and bones: Multi-layer, locally affine, optical flow and regularization with transparency. In *Proc. Computer Vision and Pattern Recognition, CVPR-96*, pages 307–314, San Francisco, June 1996.

[8] J. Krumm and S. Shafer. Shape from periodic texture using the spectrogram. In *Proc. Computer Vision and Pattern Recognition*, pages 284–301, Champaign-Urbana, Illinois, 1992.

[9] J. Krumm and S. Shafer. Segmenting textured 3d surfaces using the space/frequency representation. *Spatial Vision*, 8(2):281–308, 1994.

[10] J. Malik and R. Rosenholtz. Computing local surface orientation and shape from texture for curved surfaces. *International Journal of Computer Vision*, 23(2):149–168, 1997.

[11] G.J. McLachlan and K.E. Basford. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker Inc., N.Y., 1988.

[12] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing*, 3(5):625–638, September 1994.

Biographies

Michael J. Black

Michael Black received his B.Sc. in 1985 from the University of British Columbia, his M.S. in 1989 from Stanford University, and his Ph.D. in 1992 from Yale University. Between 1990 and 1992, he was a visiting researcher at the NASA Ames Research Center, and from 1992 to 1993 he was an Assistant Professor in the Department of Computer Science at the University of Toronto where he is currently an Adjunct Professor. In 1993 Dr. Black joined the Xerox Palo Alto Research Center and is currently the head of the image understanding research group. In 1991 he won the IEEE Computer Society Outstanding Paper Award (CVPR'91) for his work with P. Anandan on robust motion estimation. His research interests include optical flow estimation, applications of robust statistics in computer vision, and human motion and gesture understanding.

Ruth Rosenholtz

Ruth Rosenholtz received her B.S. from Swarthmore College in 1988, and her M.S. and Ph.D. from the University of California at Berkeley in 1991 and 1994, respectively. In 1995 she was a post-doctoral researcher at Utrecht University, the Netherlands, and in 1996 she held a post-doc at the NASA Ames Research Center. She joined the Xerox Palo Alto Research Center in 1997, in the image understanding group. Her research interests include texture, visual attention, and modeling of the human visual system.