Shape From Texture: Textural Invariance and the Problem of Scale In Perspective Images of Textured Surfaces

James V. Stone

Cognitive and Computing Studies,
University of Sussex, Brighton
E. Sussex BN1 9QY United Kingdom
JANET: jims@uk.ac.sussex.cogs

We introduce adaptive multi-scale filtering, a general method for deriving shape from texture under perspective projection without recourse to prior segmentation of the image into geometric texture elements (texels), and without any form of thresholding of filtered images.

If texels on a given surface can be identified in an image then the orientation of that surface can be obtained [1]. Unfortunately there is no known procedure for identifying texels for arbitrary textures. Even if the size and shape of texels on the surface is invariant with regard to position, perspective projection ensures that the size and shape of the corresponding image texels will vary by orders of magnitude.

Commencing with an initial set, F_0, of identical image filters, iterative filtering derives an ordered set, F_N, which contains a unique filter for each image position. Each element of F_N is tuned to the three-dimensional structure of the surface; that is, each filter projects to an identical shape on the surface. Thus image texels of various sizes, but associated with a single spatial scale on the surface, can be identified in different parts of the image. When combined with a conventional shape from texture method F_N provides accurate estimates of surface orientation. Results for planar surfaces are presented.

The problem of shape from texture necessarily involves establishing a correspondence between similar surface entities and their counterparts in different parts of the image. These image entities will vary in orientation and scale as a function of surface orientation and image position. Thus the problem of setting up such a correspondence and the problem of estimating surface orientation are inextricably linked.

Earlier workers have underestimated, or ignored, the problem of scale involved establishing a surface-image correspondence. Kanatani and Chou [2] suggest several schemes for overcoming the problem of "resolution threshold and sub-texture", but implemented none, and admit to potential problems. Other workers [1] restrict analysis to synthetic images that can be segmented into geometric texels. Still others [3] restrict analysis to images of surfaces which either have a population of high-contrast surface edges that can be detected with sophisticated filtering techniques, and/or to surfaces with relatively low slant/focal length ratio (see [4]), thereby facilitating detection of image objects. Only Blostein and Ahuja [5] have explicitly attempted to address the problem of scale, for the case of approximately circular image texels.

Defining Texels: Conventionally a texel is deemed to be a pattern element on a surface. However, surface textures with no texels (e.g. wood grain), and textures that do not produce identifiable image texels (e.g. grass), are capable of providing measurable image gradients, which can then be used to estimate surface orientation. In order to make use of textures such as grass and wood grain the conventional notion of texel can be extended beyond patterned, or geometric, texels to include any type of image object that can be used to infer surface orientation.

The problem then, is not only how to identify texels in an image, but how to construct a generic definition of "amount of texture" that is sufficiently robust as to allow analysis of images of arbitrary textures. An example of one such definition was given in [3] who defined texture in terms of "line length". Line length provides a simple index of "amount of texture" for both grass and wood textures. The "amount of texture" can still be measured by line length, as detected via the use of a band-pass filter, even though no geometric texels can be identified on the surface. The output of such a filter may reflect activity at spatial frequencies much lower than that associated with a single surface texel. The resultant image lines are analagous to those seen when looking a page of print at a distance; the form of such lines is determined by the arrangement of text lines and paragraphs on the page, and has little to do with the shape of individual characters. In this case lines in the image do not have a meaningful interpretation in terms of individual surface texels (characters), but (provided these image lines are associated with a single type of surface object) can nevertheless be used to estimate the orientation of the textured surface.

The Problem of Scale: A line consists of a series of edges, and each of these edges can be associated with a zero-crossing in the second spatial derivative of a *small band of spatial frequencies of the image luminance function.* Thus the process of identifying an edge in the image depends not only on the variation in image grey level,

BMVC 1990 doi:10.5244/C.4.33

but also on the scale at which the image is filtered to detect edges.

It follows that, in practice, the definition of surface texture in terms of image "line length" is actually a definition that is specified in terms of those image line lengths which are associated with a small band of spatial frequencies on the surface. For the purpose of recovering surface orientation, *it doesn't matter which band of spatial frequencies is chosen*, provided surface texels (e.g. edges) which are associated with this band of spatial frequencies on the surface can be identified in different parts of the image. ('Shape from texture' methods utilise image data that are assumed to be derived from a single small band of spatial frequencies on the surface).

Even if we restrict our definition of texel to edges, the problem of identifying image edges that correspond to a single small band of spatial frequencies on the surface has to be addressed. To return briefly to the printed page analogy given above, it doesn't matter whether edges identified in the image correspond to text lines, paragraphs or even individual characters, provided all of the image lines are associated with only one type of surface object. Use of a fixed sized image filter would result in edges associated with different spatial scales on the surface being detected in different parts of the image. Therefore, conventional shape from texture methods [1, 2, 3] will be tend to be inaccurate for surfaces with large values of slant, where the variation (due to projection) of image texel size and orientation across the image is large.

ADAPTIVE MULTI-SCALE FILTERING

Adaptive multi-scale filtering is a method for computing an ordered set, F_N, of filters (one for each image position) such that each filter projects to an identical circle on the imaged surface. Convolving each image point only with its corresponding filter in F_N is approximately equivalent to filtering the imaged surface with a single, fixed-sized filter. The resultant image data is therefore derived from a small bandwidth of spatial frequencies on the surface; this type of data is ideal for 'shape from texure' methods. (Indeed, the set F_N actually implies a particular surface, see Parallel Multi-Scale Filtering below).

Initially an image of a textured surface is convolved using identical circular (difference of Gaussian) filters of the set F_0. Image edges are obtained from the unthresholded zero-crossings in the filtered image. These edges can then be used to provide an estimate of the surface orientation, $(T = (p,q) = (-\partial Z/\partial X, -\partial Z/\partial Y,))$. This estimate can be obtained by one of three 'shape from texture' methods described in [2, 3, 4]. This initial estimate, T_1, will be inaccurate because 'shape from texture' methods rely on the assumption that objects detected in a given image correspond to a set of similar surface objects. Such a set is unlikely to be realised using the identical filters of F_0.

Using this initial estimate of surface orientation, T_1, a new set of filters, F_1, is constructed. The set F_1 consists of filters, one for each point in the image, such that the size and orientation of each filter maps to an identical circle on the estimated surface. The rate of change of size and orientation of filters in F_1 which are used throughout the image is determined by T_1 (see below). Next, the image is convolved with the filters in F_1. That is, each image point is filtered with its corresponding unique filter from F_1. The set of edges resulting from this convolution is used to re-estimate $T(= T_2)$, from which a new ordered set of filters, F_2 can be computed. This procedure is repeated, and has been found to converge in most cases (see Results). The result is a set of image filters, F_N, that will detect all and only events from a particular spatial scale on the surface. As an example, Figure 2 depicts a textured surface, and Figure 1 depicts the theoretical shape of filters in F_N.

Adaptive multi-scale filtering has thus far been implemented for texels that are defined as edges on a planar surface. However, the method could be implemented for other types of texel (e.g. lines, corners, geometric texels), and for non-planar surfaces.

Calculating the Parameters of an Image Filter

We require that set of image filters which would be obtained by projecting a set of filters from an estimated surface into the image.

Consider a small circular filter on a non-fronto-parallel surface. This circular surface filter projects to an image filter which may assumed to be an ellipse. In this section a method for computing the lengths (s_M and s_m) and orientations (β_M and β_m) of the major and minor axes (M and m) of such an ellipse is given.

In order to compute the orientations and lengths of the major and minor axes of an image ellipse it is first necessary to derive an expression for the line compression function, lcf. This function maps line elements at orientation α from a surface with orientation $T = (p,q)$ to image line elements with orientation β.

The equations for perspective projection are, $x = X/Z, y = Y/Z$, where x and y are image parameters, and X, Y, and Z are 'world' parameters, and the focal length, f, is set to unity. Given that the equation of a plane is defined as $AX + BY + CZ - D = 0$, the differential of length, dS, on a surface is given by:

$$dS = (dX^2 + dY^2 + dZ^2)^{1/2} \quad (1)$$

Expressions for the total differentials dX, dY, dZ can be derived in terms of image parameters:

$$\begin{aligned} dX &= (\partial X/\partial x)dx + (\partial Y/\partial y)\, dy \\ &= K[qxdy - dx(1+qy)]/W \end{aligned} \quad (2)$$

$K = D/C, W = (1 + px + qy)^2$

This can be re-written in terms of the orientation, β, of a line element, ds. Substituting $dx = cos(\beta).ds$, $dy = sin(\beta).ds$ into (2), and into the corresponding equations for the total differentials dY and dZ, yields:

$$\begin{aligned} dS = \; & (K/W).(qx.sin(\beta) - cos(\beta).(1+qy))^2 + \\ & (py.cos(\beta) - sin(\beta)(1+px))^2 + \\ & (p.cos(\beta) + q.sin(\beta))^2)^{1/2}\, ds \end{aligned} \quad (3)$$

Where x, y is the 'position' of image line element, ds is the differential of line length at orientation β in the image.

In order to simplify notation we can re-write (3) in terms of a line compression function, lcf, such that:

$$dS = lcf(x, y, p, q, \beta)\ ds \quad (4)$$

This will be used to derive expressions for the elliptical filter parameters.

The Orientation of an Image Filter: The orientations, β_M and β_m, of the major and minor axes of an image ellipse, which projects to a circle on a surface with local orientation T, can be obtained as follows.

The value of lcf is minimal for $\beta = \beta_M$, and maximal for $\beta = \beta_m$. Thus if β equals β_M or β_m then the derivative of lcf with respect to β is equal to zero. The derivative of lcf can be shown to be [4]:

$$\partial(lcf)/\partial\beta = a.tan^2(\beta) + b.tan(\beta) + c \quad (5)$$

$a = -(1 + pq)$, $b = p^2(x^2 - y^2 - 1) + q^2(x^2 - y^2 + 1) + 2(px - qy)$, $c = 1 + (p + q)(x + y) + pq$

The values of β_M and β_m may then be obtained by setting $\partial(lcf)/\partial\beta = 0$ and solving (5) for $tan(\beta)$.

The Length of the Axes of an Elliptical Image Filter: Consider an image ellipse, with axis lengths s_M and s_m, which is derived from a surface circle with diameter of length S_1. Note that both of the axes of this ellipse are associated with a single circle on the surface. It follows that m is the projection of a circle diameter at one orientation on the surface, and M is the projection of another diameter of the same circle (but at a different orientation) on the surface.

Substituting $x = x_0 + s.cos(\beta)$, $y = y_0 + s.sin(\beta)$ into (5), the length, S_1, of a line on the surface which corresponds to an image length s_1 at orientation β is given by:

$$S_1 = \int_0^{s_1} (K/W).(qx_0.sin(\beta) - cos(\beta).(1 + qy_0))^2 + (py_0.cos(\beta) - sin(\beta)(1 + px_0))^2 + (p.cos(\beta) + q.sin(\beta))^2)^{1/2}\ ds \quad (6)$$

Evaluating (6) and re-writing the resultant equation in terms of s_1 yields a function, g, such that:

$$s_1 = g(S_1, x_0, y_0, \beta) \quad (7)$$

Thus, given a surface line of length S_1 which projects to an image line at, x_0, y_0, with orientation, β, the length of that image line, s_1, can be computed from (7).

To summarise. For a circle on a surface with known 3-space orientation the image orientations, β_m and β_M, of the minor and major axes of the corresponding image ellipse can be obtained from (5). These angles, along with the known surface circle diameter, S_1, can then be used to derive the image lengths s_M and s_m, respectively, using (7). Note that the value of the surface diameter is the chosen value of S_1, and defines the scale at which the surface is to be filtered.

RESULTS

Adaptive multi-scale filtering has been tested on synthetic images of planar surfaces. Each 512x512 image was filtered with a difference of Guassian (DOG) with ratio of large to small Gaussian set to 1.6. For the synthetic images tested here the standard deviation of the larger of the two Gaussians used to construct the DOG filter is 6 pixel units. Each image has 255 grey-levels. *All zero-crossings in the filtered image were labelled as edges.* The particular 'shape from texture' method used to re-estimate $T = (p, q)$ after each set F_i has been convolved with the image is described in [2]; however, other methods [3, 4] could have be used for this purpose. The largest filter in each iteration has a major axis whose length is the same as the radius of the identical circular filters of the initial filter set F_0. This means that the size of filters in F_0 acts as a reference size, with other image filters varying with respect to filters in F_0 according to the estimated surface orientation.

Due to the expense of using filters whose major and minor axes are not parallel to the x and y image axes, the actual asymmetric filter dimensions used were computed as follows. The value of p was set to zero, and the size and orientation of all filters at a given height, y, in the image were assumed to be identical to the (vertically oriented) elliptical filter computed for $x = 0, y$.

Figures 2 and 5 depict an image of a textured plane with $T = (0.0, -0.839)$ and focal length of 512 pixel units. Figure 1 depicts an idealised set of image filters; these represent the image filters required to detect events at a single spatial scale on the surface in Figures 1 and 5. Each edge map is labelled with the (estimated) value of T_i used to construct the filter set, F_i, and this, in turn, is used to construct the edge map shown. This edge map is used to derive a new (re-estimated) value of $T = T_{i+1}$.

After filtering with a set, F_0, of identical filters, Figure 3 shows that the resultant image edges reflect activity across a range of spatial scales on the surface. The estimate of surface orientation, T_1, based on this edge map, is consequently in considerable error (T_1=0.0,-0.236); the difference, δ, between the actual and estimated surface orientation being 31.1 degrees. A new set of image filters, F_1, based on the value of T_1 is constructed and convolved with the original image. This new set of image filters is constructed according to the method outlined in the previous section, yielding a new filter for each image position. This process of filtering and re-estimation of T is repeated and converges at $T = (0.0, -0.760)$. This represents an error in the surface normal of 2.76 degrees. Results were also obtained for Figure 5, where the surface texels are not well defined. The final estimate after 6 iterations was $T = (0.0, -0.831)$, an error of 0.270 degrees.

A plot of estimated orientation versus iteration for both Figures 2 and 5 is given in Figure 8. In order to test that the estimates will converge on the correct value, Figs 2 and 5 were filtered with a set of filters appropriate to $T = (0.0, -0.839)$, the actual surface orientation. The resultant edge map produced new estimates of $T = (0.0, -0.794)$ and $T = (0.0, -.822)$; this represents an error of $\delta = 1.55$ and $\delta = 0.576$ degrees, respectively.

For both of the tested images, edges associated with many spatial scales on the surface were initially detected. Normally a proportion of such edges would be discarded on the grounds that their corresponding surface entities (e.g. patterned texels) cannot be identified in different parts of the image (and cannot therefore be used to estimate T). In contrast each edge in all of the edge maps of both images tested here was given equal weighting in estimating T. Thus even the ill-defined surface objects corresponding to the image edges of Figs 3,4,6 and 7 contribute to the estimation of surface orientation. Preliminary results suggest this method also works well on images for which no patterned texels can be identified.

This method has been found not to converge in a small number of cases. This occurs when the computed direction of tilt ($= p/q$) associated with T_1 has the wrong sign. This forces the subsequent filter set to be 'tuned' to a surface even further from the actual surface normal than was the initial filter set, F_0 (for which the associated value of $T = (0.0, 0.0)$). Non-convergence appears to be caused by textures whose spatial frequency spectrum approximates that of 2-D white noise. However, this type of texture might be expected to cause problems for *any shape from texture method that relies upon the detection of similar-sized surface objects in the image.* In the case of a pure 'white-noise' texture it is certainly unreasonable to expect a 'shape from texture' method to estimate surface orientation, because it is difficult to differentiate between image frequency components that are derived from similar surface frequency components.

Certain 'white noise' textures will be more amenable to a parallel method, from which the serial, adaptive multi-scale filtering technique was originally derived. This method, parallel multi-scale filtering [4], has not been implemented.

Parallel Multi-Scale Filtering

In terms of Marr's[6] three levels of analysis (computational, algorithmic, implementational), the method of adaptive multi-scale filtering is pitched at the algorithmic level; it specifies a method for excecuting a particular computational task. The task consists of deriving a set of image filters, F_N, appropriate to a particular surface orientation, T, where the values of F_N and T are initially unknown. There are many algorithmic level descriptions for executing this task, one of which is the method of adaptive multi-scale filtering described above. Other algorithms, plus possible neurophysiological implementations are considered here.

In order to estimate surface orientation, T, a distribution of image filters, F_N, which can identify events derived from a small band of spatial scales on the surface must be established. In fact, surface orientation, T, is implicit in the distribution of filters specified by F_N. In short, F_N and T are co-determined. When two variables are co-determined it makes sense to compute the value of the one which is easier to evaluate, and, if required, from this to obtain the value of the other variable. Thus we might choose to find T, and from this compute F_N; alternatively we might find F_N and from this compute T. Note that the fact that F_N and T are co-determined does not imply that they cannot be evaluated by independent means (e.g. T could be obtained from stereo information). In fact, it will be shown that it is a relatively simple matter to evaluate F_N independently of T, and then to use F_N to obtain the value of T.

Consider a one-dimensional image of a one-dimensional textured surface. In the case of a 1-D image p=0, the length, δy, of texture elements is proportional to $1/Z^2 = ((1 + qy)/K)^2$ [4]. The corresponding distribution of image filters, F_N, can be obtained by use of stacked arrays of filters, where each array consists of a set of filters of a single size, σ. Now each image point is analysed at many scales, σ, (i.e. by each array in the bank of filters). For each image position, y, there will be a corresponding filter length, $\sigma_{max} = f(y)$, which will represent a peak, or local maximum, of activity in the filter bank, B. If we trace a curve through the set of peaks in the filter bank (where each peak corresponds to a local maximum of filter activity) it would look something like the curve drawn in Figure 9. Moreover *the distribution of filters specified by such a curve is precisely the distribution, F_N, of filters previously derived using adaptive multi-scale filtering.* As stated above F_N and T are co-determined, so that F_N may be used to derive T. In the case of a planar surface the curve defined by F_N is of the form $\sigma_{max} = H.(1+qy)^2$ (H=scale constant); that is, the filters in B that are associated with maximal activity will be matched to the scale of the strongest Fourier component of the surface texture. Textures with many strong frequency components will generate many parallel 'peak-activity' contours in the filter bank, one for each component.

Each component in the Fourier transform of the surface texture gives rise to a peak-activity contour in B; and each contour defines a filter set F_N. Any one of these sets may be used to estimate the surface orientation, by finding that value of q which best describes the associated parameterized (in the case of a planar surface) peak-curve in B. Finally it should be noted that, just as a planar 1-D surface can be recovered from its peak-actitvity contours in a filter bank, so the local orientations of a curved 1-D surface can, in principle, be recovered from its peak-activity contours.

The rationale given here for 1-D surfaces can be extended to 2-D surfaces, but the relation between peak-actitvity contours and surface orientation is more complex. This is due to the fact that 2-D filters in F_N vary not only in size, σ, they also vary in orientation. In the 1-D case only one parameter, σ, is required to specify a filter, whereas in the 2-D case three parameters are required. These are the lengths, s_M and s_m, of its major and minor axes, and the orientation, $\beta = \beta_M$, of the image filter. Thus the corresponding filter bank for a 2-D image has 5 dimensions, x, y, β, s_M, and s_m. The corresponding peak-actitvity contours (which are actually 4-D hyper-surfaces) would be drawn in this 5-D space. However, if the set of filters is restricted so that all image filters are circular then the filter bank for a 2-D image would be 3-dimensional, the dimensions being x, y, and a(=area of image filter). In the case of a planar surface the area of the filters in a peak-activity contour is of the form $a_{max} = H.(1 + px + qy)^{-3}$ [4] (H=scale constant). As with the 1-D case the surface orientation can be obtained by finding that value of T which best describes the parameterized curve described by contiguous values of a_{max} in B. Given that there exists a one-to-one mapping between the shape of a peak-activity contour and surface orientation, the problem of shape from texture can be reformulated in terms of detecting such contours in contiguous arrays of filters.

Finally, it has recently been proposed [7] that the known distribution of receptive field sizes and densities may facilitate visual interpretation of 3-D surfaces. This approach suggests that it is not the outcome of the computation of surface orientation that enables detection of similar surface objects, but that it is the detection of similar surface objects at many spatial scales in the image that implies a particular surface. The method of adaptive multi-scale filtering is a fusion of these two approaches, integrating the process of estimating surface orientation and detection of surface objects.

CONCLUSION

Adaptive multi-scale filtering provides a general method for deriving shape from texture without recourse to prior segmentation of the image into discrete texture elements, and without any form of thresholding of filtered images.

The problem of scale is an integral part of the problem of shape from texture. The process of adaptive multi-scale filtering treats it as such, yielding accurate estimates of surface orientation even for images of surfaces with large values of slant.

Future work will extend the method of iterative to deal with curved surfaces.

References

[1] **Ohta Y, Maenobu K, Sakai T** "Obtaining surface orientation from texels under perspective projection" *Proc. 7th IJCAI* , Vancouver, Canada, (1981) pp 746-751.

[2] **Kanatani, K and Chou, T** "Shape From Texture: General Principle", *Artificial Intelligence*, 38, 1, (1989), pp 1-49.

[3] **Aloimonas J,** "Shape from Texture" *Biological Cybernetics* 58, (1988) pp 345-360.

[4] **Stone, J V** *Shape From Texture: A Computational Analysis*, PhD Thesis(in preparation), Experimental Psychology, University of Sussex, England.

[5] **Blostein D, and Ahuja N** "Shape From Texture: Inegrating Surface Element Extraction and Surface Estimation", *PAMI*, 1, 12, (1989), pp 1233-1251.

[6] **Marr, D,** in *Vision* WH Freeman, San Fracisco(1981)

[7] **Johnstone A,** "The Geometry of the Striate Topographic Map: Functional Implications" in *Models of Brain Function*(in press)

Acknowledgements: Thanks to Steve Isard for help in developing ideas presented here, and to Raymond Lister for comments on drafts of this paper.

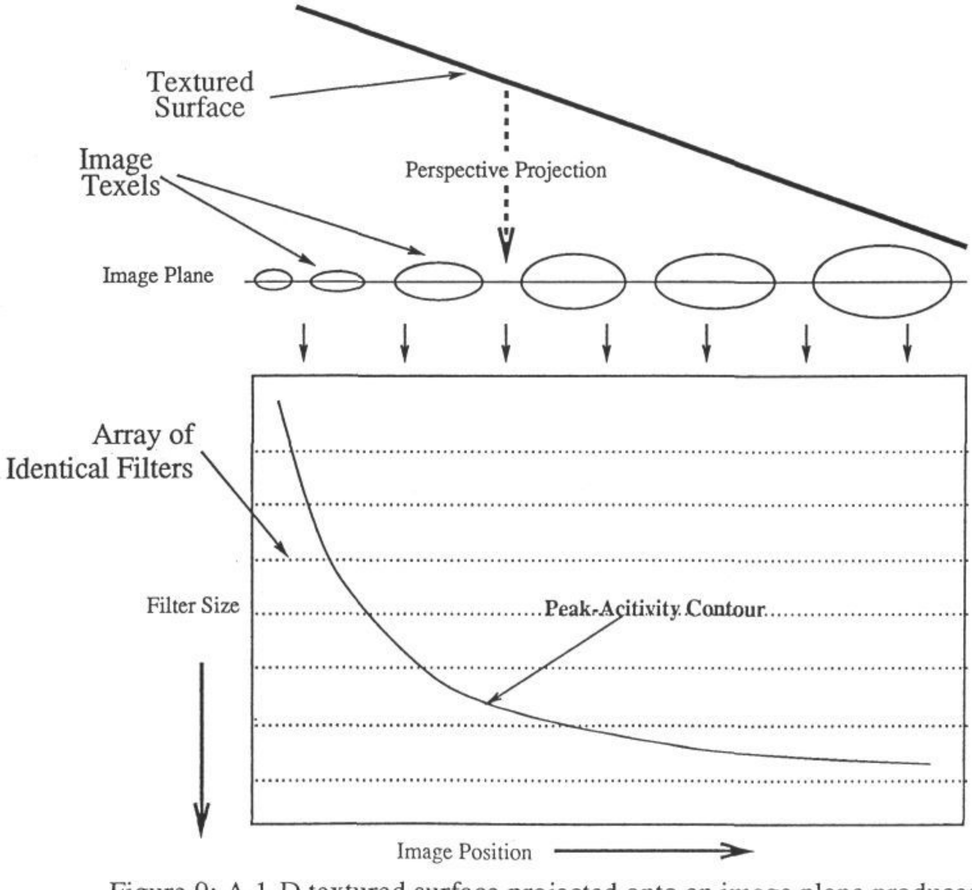
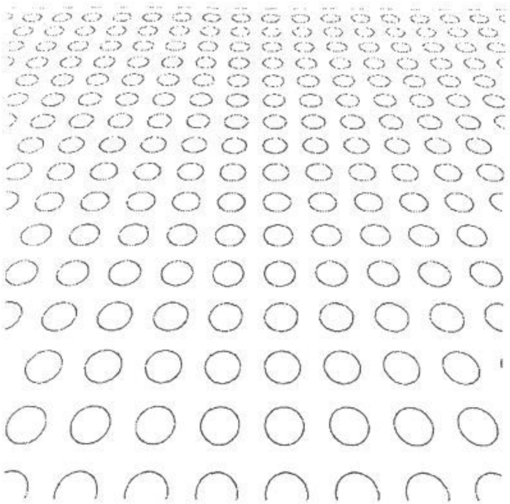


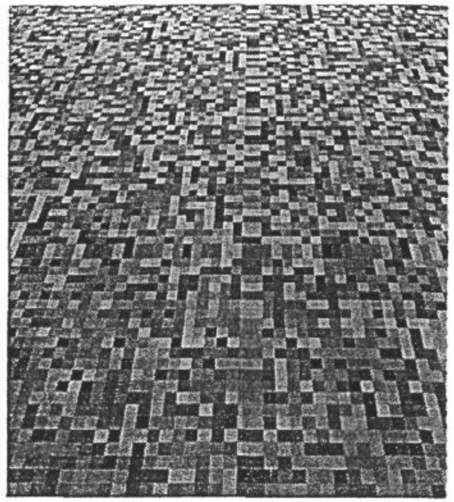
Figure 9: A 1-D textured surface projected onto an image plane produces peak-activity contours in stacked arrays of filters.

Figure 1:



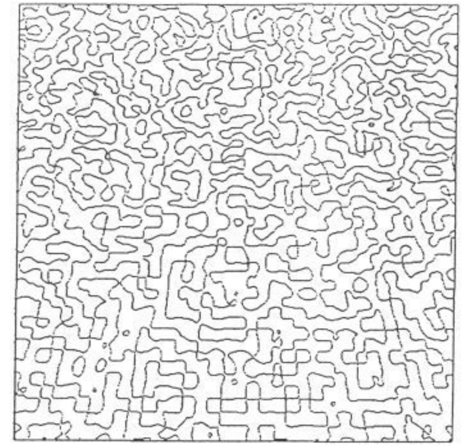
Synthetic image of textured plane. p=0, q=-0.839, f=512

Figure 2:



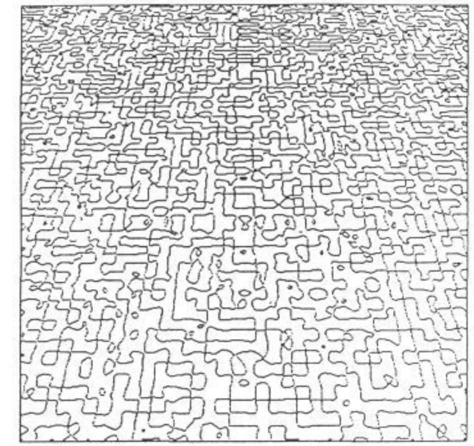
Synthetic image (512x512 pixels) of textured plane
p=0.0, q=-0.839, focal length=512 pixels

Figure 3:



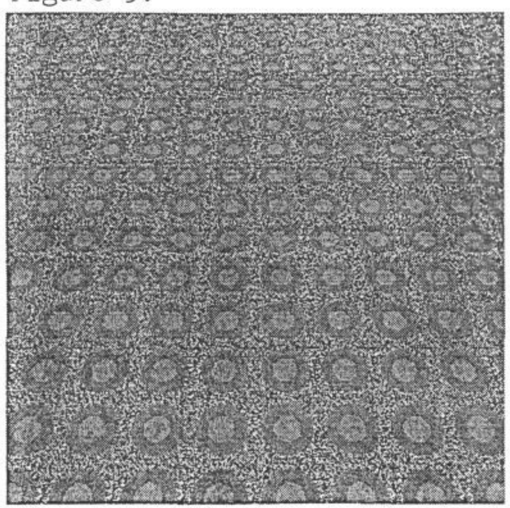
Zero-cossings in image: p=0.0, q=-0.839, f=512
Filtering based on estimated p=0.0,q=-0.0
Re-estimated p=0.0, q=-0.236

Figure 4:



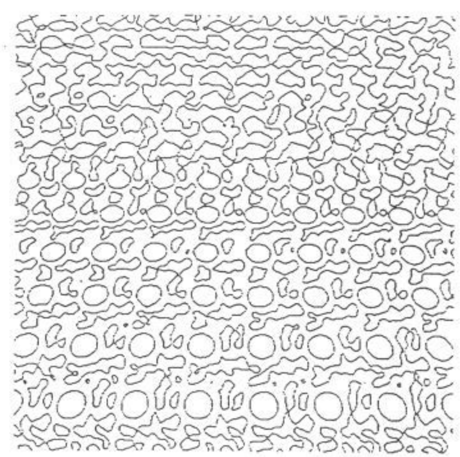
Zero-crossings in image: p=0, q=-0.839, f=512
Filtering based on estimated p=0.0, q=-0.738
Re-estimated p=0.0, q=-0.760

Figure 5:



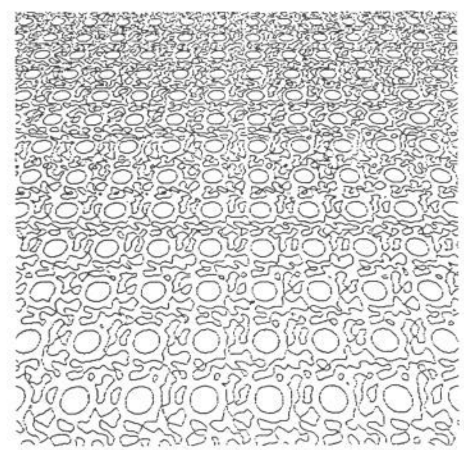
Synthetic image of plane: p=0.00, q=-0.839, f=512

Figure 6:



Zero-crossings in image
Filtering based on estimated p=0.0, q=0.0
Re-estimated p=0.014, q=-0.113

Figure 7:



Zero-crossings in image
Filtering based on estimated p=0.0, q=-0.814
Re-estimated p=0.033, q=-0.831

Figure 8: