

# Texture Descriptors via Stable Distributions

Pattaraporn Khuwuthyakorn<sup>1</sup>   Antonio Robles-Kelly<sup>1,2</sup>   Jun Zhou<sup>1,2</sup>

<sup>1</sup>College of Engineering and Computer Science, ANU, Canberra ACT 0200, Australia

<sup>2</sup>National ICT Australia (NICTA)\*, Locked Bag 8001, Canberra ACT 2601, Australia

## Abstract

*In this paper, we present a texture descriptor which hinges in the use of the local image statistics so as to recover a compact representation of the texture under study. To this end, here, we make use of stable distributions and their link to Fourier analysis so as to provide a means to compute in a computationally efficient manner a local texture descriptor. This link between stochastic processes and Fourier analysis provides an efficient means to compute texture spectra which can be interpreted as a probability distribution for purposes of recognition and analysis. Making use of our local descriptor, we provide a metric between texture pairs that can be made devoid of rotations on the texture plane by recovering the optimal linear transformation via Procrustes analysis. We demonstrate the utility of our descriptor and its associated metric on a database of real-world textures.*

## 1 Introduction

The recovery of shape from texture information is a key problem in computer vision and pattern recognition. Indeed, texture has found application not only as a shape queue, i.e. shape-from-texture, but has also attracted broad attention due to its applications to recognition and classification [11, 12] tasks.

In the case of shape-from-texture, the problem has been approached in either a global or a local manner. The global approach hinges in the recovery of vanishing points from either texture-gradient or spectral information. In this manner, texture planes can be recovered via the structural analysis of predetermined texture primitives [4, 1, 5]. This treatment provides an intuitive geometrical meaning to the task of recovering the parameters governing the pose by making use of methods akin to 3D view geometry. Unfortunately, the complexity required to handle scenes with multiple tex-

ture planes renders these methods impractical in many real-world settings.

In the case of local approaches, these are usually based upon measures of spectral distortion. In these methods, the spectral distortion in the texture provides a means to estimate the pose parameters of the plane. Thus, the texture under study is transformed into the Fourier domain, where the perspective geometry can be treated in a local fashion so as to obtain measures of texture distortion [14, 7, 2, 10]. These methods are more suitable for the analysis of scenes containing multiple planes or curved surfaces. Nonetheless, they rely on numerical optimisation techniques that may be computationally intensive and, additionally, often do not provide closed-form solutions to the pose estimate recovery problem.

In this paper, we focus in the development of a descriptor for recognition tasks based upon texture. Thus, our aim of computation is a means to describe the texture under study so as to capture the information of the image patch in a compact form. This problem is reminiscent to that found in texture-plane segmentation, in which textures with similar perspective pose parameters, i.e. local spectral distortion, have to be separated into groups or clusters. Krumm and Shafer [8] have used spectral back-projection to estimate local surface orientation by recovering the parameters which minimise the sum-of-squares difference between local spectra. Their method employs the local power-spectrum, which is projected onto the front-parallel plane. Texture planes are segmented using a dendrogram-based clustering method. In a related development, Ribeiro and Hancock [13] showed that the slant and tilt of textured surfaces can be estimated without the need for iterative numerical optimisation by working in the frequency domain and measuring texture distortion making use of the affine transformation on the pattern of spectral peaks.

These methods make ample use of the Fourier transform for purposes of recovering the pose parameters of texture planes. In this paper, we present a texture descriptor which hinges in the link between the statistics of the texture under study and its local spectra. As a result, the descriptor can be viewed as a probability distribution function which can

\*NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

be computed efficiently making use of Fourier transforms. Moreover, we also present a metric between textures making use of our descriptor. This metric can be interpreted as the goodness of fit between two texture spectra subject to a linear transformation. In this manner, our descriptor is invariant to rotations on the texture plane. The rest of the paper is organized as follows. Section 2 presents stable distributions and provides a link between stochastic processes and Fourier transforms. In the section, we also present the metric between descriptors and elaborate on their computation. We report experimental results on a texture database in Section 3. Section 4 provides conclusions on the work presented here.

## 2 Texture Descriptor

Here, we view textures as stochastic processes whose distributions are “heavy-tailed”, i.e. there is a higher probability of extreme values within their distributions. In other words, we view the description of textures as the recovery of a distribution which corresponds to a high variability whose extreme values can be “clustered”. From this viewpoint, texture pixel values arise from a stochastic process such that there exist considerable amount of extreme colour values about the mean. As a result, in this section, we commence by introducing stable distributions and, later on, we employ them to provide a foundation for the use of Fourier transforms to build a texture descriptor. In this way, we can finalise the section by presenting a similarity measure which, at input, takes two texture descriptors given by 2D Fourier transforms and, at output, delivers a metric that can be used for recognition tasks.

### 2.1 Textures as Stochastic Processes

It is worth noting that, whereas normal and log-normal distributions are a reasonable choice when the data presents a clear tendency towards the mean, texture behaviour is such that results in rather common deviations of the pixel values from the colour or luminance mean. These “extreme” values in the distribution cannot be ignored since they are not outliers, but rather they arise from the stochastic process under study. As a result, and to take our analysis further, we focus in the use of stable distributions [9].

Thus, following our assumption of stable distributions, we view the colour values for each channel, i.e. red (R), green (G) and blue (B), at the pixel site  $u$  as random variables  $\mathcal{Y}_u$  whose inherent basis  $\mathcal{X}_u = \{\chi_u(1), \chi_u(2), \dots, \chi_u(|\mathcal{X}_u|)\}$  is such that

$$P(\mathcal{Y}_u) = \sum_{k=1}^{|\mathcal{X}_u|} P(\chi_u(k)) \quad (2.1)$$

where,  $\chi_u(k)$  are identically distributed variables and, as usual for probability distributions of real-valued variables, we have written  $P(\mathcal{Y}_u) = Pr[y \leq \mathcal{Y}_u]$  for all  $y \in \mathbb{R}$ .

In other words, we view the pixel values of the texture under study as arising from a family of distributions whose variance is not necessarily finite. It is worth noting in passing that, for finite variance, the formalism above implies that  $P(\mathcal{Y}_u)$  is normally distributed. Nonetheless, this treatment generalises the stochastic process to a number of independent influences, each of which is captured by the corresponding variable  $\chi_u(k)$ .

In practice, the Probability Density Function (PDF)  $f(\mathcal{Y}_u)$  is not available in close form. As a result, we can reparameterise the PDF recasting it as a function of the variable  $t$  making use of the characteristic function

$$\begin{aligned} \psi(t) &= \int_{-\infty}^{\infty} \exp(it\mathcal{Y}_u) f(\mathcal{Y}_u) d\mathcal{Y}_u \\ &= \exp\left(\mathbf{i}vt - \gamma |t|^\alpha (1 + \mathbf{i}\beta \operatorname{sign}(t)\varphi(t, \alpha))\right) \end{aligned} \quad (2.2)$$

where  $\mathbf{i} = \sqrt{-1}$ ,  $v \in \mathbb{R}$  and  $\gamma \in \mathbb{R}^+$  are function parameters,  $\beta \in [-1, 1]$  and  $\alpha \in (0, 2]$  are the skewness and characteristic exponent, respectively, and  $\varphi(\cdot)$  is defined as follows

$$\varphi(t, \alpha) = \begin{cases} \tan\left(\alpha \frac{\pi}{2}\right) & \text{if } \alpha \neq 1 \\ -\frac{2}{\pi} \log |t| & \text{if } \alpha = 1 \end{cases} \quad (2.3)$$

### 2.2 Fourier Transforms and Stable Distributions

Note that, the main advantage of the formalism above is that it allows us to model the PDF of the channel pixel-value  $\mathcal{Y}_u$  as a Fourier transform. To see this, we can write the log-characteristic function as follows

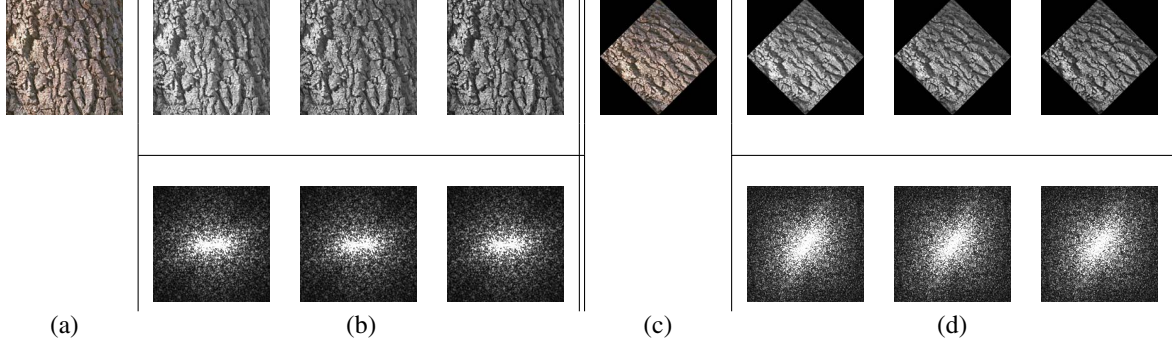
$$\begin{aligned} \log[\psi(t)] &= \mathbf{i}vt - \gamma |t|^\alpha (1 + \mathbf{i}\beta \operatorname{sign}(t)\varphi(t, \alpha)) \\ &= \mathbf{i}vt - |t|^\alpha \gamma^* \exp\left(-\mathbf{i}\beta^* \frac{\pi}{2} \vartheta \operatorname{sign}(t)\right) \end{aligned}$$

where  $\vartheta = 1 - |\beta|$  and the parameters  $\gamma^*$  and  $\beta^*$  are given by

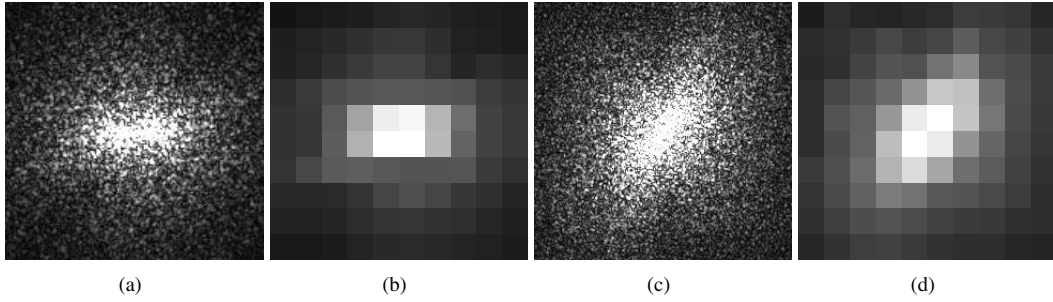
$$\begin{aligned} \beta^* &= \frac{2}{\pi\vartheta} \arccos\left(\frac{\cos\left(\alpha \frac{\pi}{2}\right)}{\sqrt{\Omega}}\right) \\ \gamma^* &= \left(\frac{\gamma\sqrt{\Omega}}{\cos\left(\alpha \frac{\pi}{2}\right)}\right)^{\frac{1}{\alpha}} \end{aligned} \quad (2.4)$$

and  $\Omega = \cos^2\left(\alpha \frac{\pi}{2}\right) + \beta^2 \sin^2\left(\alpha \frac{\pi}{2}\right)$ .

Recall that, in the previous section, we presented the characteristic function in Equation 2.2. We can use Fourier inversion on the characteristic function and, making use of



**Figure 1.** Example textures and associated FFTs. (a) and (c): Input texture images; (b) and (d), Top row: R, G and B channels for the textures in (a) and (c); (b) and (d), Bottom row: FFTs for the R, G and B channels in the top rows.



**Figure 2.** Left-hand panels: Mean (a) and bicubic fit (b) for the non-rotated transforms in Figure 1; Right-hand panel: Mean (c) and bicubic fit (d) for the rotated texture in Figure 1.

the shorthands defined above, it can be shown that the PDF may be computed via the equation

$$f(\mathcal{Y}_u; v, \beta^*, \gamma^*, \alpha) = \frac{1}{\pi \gamma^*} \int_0^\infty \cos \left( \frac{(u - \mathcal{Y}_u)s}{\gamma^*} + s^\alpha \sin(\phi) \right) \exp \left( -s^\alpha \sin(\phi) \right) ds \quad (2.5)$$

where  $\phi = \frac{\beta^* \pi \eta}{2}$ .

Note that the expression above is, in fact, a Fourier transform. This observation is not only theoretically important but practically useful since it provides a means of computing the PDF of a texture via Fourier analysis. Moreover, it substantiates the use of the Fourier transform as a PDF for the texture under study.

### 2.3 Descriptor Construction

As a result, we construct our descriptor as follows. We commence by computing the Fast Fourier Transform (FFT)

of the texture under study. Note that, since our focus of study are image textures, the FFTs are two-dimensional in nature. Our descriptor is then given by the set  $\Gamma$  of three-dimensional points whose coordinates are  $x, y, z$ . For the points in  $\Gamma$ , we constrain the coordinates  $x, y$  to be on a lattice and interpolate the  $z$ -coordinates to the values  $\mathcal{F}(x, y)$ , where  $\mathcal{F}(\cdot)$  is the Fourier transform corresponding to the image texture.

In Figure 1, we show the Fourier transforms for an example texture. Since our textures are colour ones, we use the mean value for the interpolations corresponding to the FFTs for each of the colour channels. In Figure 2, we show the mean FFT and the interpolated points  $\Gamma$  for the example textures in Figure 1. Here, we have used descriptors with 100 points on a  $10 \times 10$  lattice and used, for purposes of interpolation, a bicubic spline [6]. It is worth noting that, since the points are distributed on a lattice, the descriptors can be stored as vectors whose elements are the raster-scanned  $z$  coordinates of those points in  $\Gamma$ .



**Figure 3.** Sample Images for the four categories in the database.

| Angle | 0    | 10   | 20   | 30   | 40   | 50   | 60   | 70   | 80   | 90   |
|-------|------|------|------|------|------|------|------|------|------|------|
| 0     | 0.00 | 0.01 | 0.03 | 0.04 | 0.05 | 0.06 | 0.05 | 0.04 | 0.03 | 0.03 |
| 10    | 0.01 | 0.00 | 0.00 | 0.01 | 0.02 | 0.03 | 0.03 | 0.02 | 0.03 | 0.03 |
| 20    | 0.03 | 0.00 | 0.00 | 0.00 | 0.01 | 0.01 | 0.02 | 0.02 | 0.03 | 0.04 |
| 30    | 0.04 | 0.01 | 0.00 | 0.00 | 0.00 | 0.01 | 0.01 | 0.02 | 0.04 | 0.05 |
| 40    | 0.05 | 0.02 | 0.01 | 0.00 | 0.00 | 0.00 | 0.01 | 0.01 | 0.04 | 0.05 |
| 50    | 0.06 | 0.03 | 0.01 | 0.01 | 0.00 | 0.00 | 0.00 | 0.01 | 0.03 | 0.04 |
| 60    | 0.05 | 0.03 | 0.02 | 0.01 | 0.01 | 0.00 | 0.00 | 0.00 | 0.03 | 0.04 |
| 70    | 0.04 | 0.02 | 0.02 | 0.02 | 0.01 | 0.01 | 0.00 | 0.00 | 0.01 | 0.02 |
| 80    | 0.03 | 0.03 | 0.03 | 0.04 | 0.04 | 0.03 | 0.03 | 0.01 | 0.00 | 0.00 |
| 90    | 0.03 | 0.03 | 0.04 | 0.05 | 0.05 | 0.04 | 0.04 | 0.02 | 0.00 | 0.00 |

**Table 1.** Matrix of normalised errors  $\mathcal{E}$  for the texture in the left-hand panel of Figure 1 rotated in steps of  $10^\circ$  from  $0^\circ$  to  $90^\circ$ .

## 2.4 Metric Between Texture Descriptors

Note that, in case of texture rotations, the FFT will spin about its center. This is consistent with the work in [13], where the spectral distortion of the textures under study is used for purposes of recovering shape queues. It is worth stressing in passing, however, that the work presented in [13] is based upon Fourier analysis and involves the recovery of the spectral homographies between neighbouring local spectra. In our approach, the homography is not the aim of computation. This is as we seek to recover a metric between texture descriptors as compared to the slant and tilt angles of the perspective textured plane.

This spectral distortion as a result of perspective transformations on the texture can be appreciated in Figures 1 and 2, where we show two example textures and their respective FFTs. From the figures, it is clear that the transforms in the right-hand panels are the rotated analogues of their counterparts in the left-hand side.

As a result, if a metric between descriptors is to be computed, we require a means to recover the parameters that govern the linear, i.e. affine, transformation between two FFTs corresponding to a texture pair. In other words, in order to compute a metric between a pair of descriptors, we aim at recovering the linear transformation between two sets of 3D points. These 3D points correspond to the sets  $\Gamma$  for the two textures under comparison.

Thus, here, we employ Procrustes analysis so as to deter-

mine a linear transformation between two sets of 3D points. Let the centered coordinates of the data point indexed  $i$  be  $\tilde{p}_i = [x_i - \mu_x, y_i - \mu_y, z_i - \mu_z]^T$ , where  $\mu_x$ ,  $\mu_y$  and  $\mu_z$  are the mean data-coordinate values in the x, y and z axis. With these ingredients, the matrix of normalized 3D point-coordinates is given by  $\tilde{\mathbf{D}} = [\tilde{p}_1, \tilde{p}_2, \dots, \tilde{p}_N]^T$ . A Procrustes transformation of the matrix of normalized 3D point-coordinates  $\tilde{\mathbf{D}}$  is of the form  $\mathbf{Q} = \mathcal{R}\tilde{\mathbf{D}}$  which minimizes the normalized sum of squared errors

$$\mathcal{E} = \frac{\|\mathbf{M} - \mathbf{Q}\|^2}{\|\tilde{\mathbf{D}}\|^2} \quad (2.6)$$

where  $\mathbf{M}$  is a matrix whose  $i^{th}$  row corresponds to the coordinates of the ground truth point indexed  $i$  and  $\mathcal{R}$  is a transformation matrix. In the equation above, we have assumed that the point coordinates are centered, i.e. the centroid of the data point-cloud is at the origin.

It is known that minimizing  $\mathcal{E}$  is equivalent to maximizing  $\text{Tr}[\tilde{\mathbf{D}}\mathbf{M}^T\mathcal{R}]$  [3]. Let the singular value decomposition (SVD) of  $\tilde{\mathbf{D}}\mathbf{M}^T$  be  $\mathbf{USV}^T$ . The maximum of  $\text{Tr}[\tilde{\mathbf{D}}\mathbf{M}^T\mathcal{R}]$  is achieved when  $\mathbf{V}^T\mathcal{R}\mathbf{U} = \mathbf{I}$ . As a result, the optimal transformation matrix  $\mathcal{R}$  is given by

$$\mathcal{R} = \mathbf{V}\mathbf{U}^T \quad (2.7)$$

The goodness-of-fit for this transformation is determined using the sum of squared errors  $\mathcal{E}$  between the points in  $\Gamma$  as interpolated by the bicubic spline for the two textures under study. By using  $\tilde{\mathbf{D}}$  as an alternative to the non-centered



**Figure 4.** Example recognition results. (a) and (c): Sample test images; (b) and (d): Results ordered by relevance from left-to-right and top-to-bottom yielded by the  $k$ -nearest neighbour classifier making use of the goodness of fit between the bicubic interpolants for the  $z$  coordinates of the points in  $\Gamma$  used by our descriptor.

point-coordinates in  $\Gamma$ , the error  $\mathcal{E}$  is normalized by the sum of squares for the centered 3D point-coordinates. The use of the normalized error makes the quantity  $\mathcal{E}$  devoid of scaling and translational components in the descriptors under comparison.

### 3 Experiments

Having presented the theoretical basis for our descriptor, we now turn our attention to its utility for purposes of recognition. To this end, we have used the VisTex database<sup>1</sup>. This is a database of real-world textures which covers a number of materials, such as brick, bark, buildings, etc. In our experiments, we have used a subset of 108 images. These images correspond to four categories, i.e. bark, brick, buildings and paintings. Note that, in the case of the bark and brick textures, these are homogeneous ones, whereas the paintings and buildings images are textured, real-world scenes. In Figure 3, we show example images for the four categories in the data set.

We commence by illustrating the invariance to rotation of the normalised sum of squared errors  $\mathcal{E}$  between two sample textures. In Table 1, we show the matrix of errors  $\mathcal{E}$  as a function of rotation angle for the bark texture in Figure 1. In the matrix, we show the goodness of fit for rotations between  $0^\circ$  and  $90^\circ$  degrees in increments of  $10^\circ$ . From the matrix, we can conclude that the largest “deviation” in goodness of fit is never greater than 0.06. Moreover, the largest values of  $\mathcal{E}$  are those between textures that have been rotated  $30^\circ$ . This may be due to the bicubic interpolation step, which is more prone to errors for rotations between  $30^\circ$  and  $50^\circ$ .

For purposes of showing the utility of our descriptor for recognition tasks, we have used a subset of forty eight images, i.e. 12 for each category, as reference for the recog-

nition task in hand. The rest of the images have been used for testing by assigning them to the classes of reference, i.e. bark, brick, buildings or paintings, using a  $k$ -nearest neighbour approach. For purposes of  $k$ -nearest neighbour classification, in our experiments we have used the goodness of fit  $\mathcal{E}$  between the bicubic interpolations for each texture pair.

In Figure 4, we show two sample recognition results. In (a) and (c), we show the test texture images. In (b) and (d), we show the six “closest” images in the database corresponding to the test images in (a) and (c), respectively. In the panels, we show, from left-to-right and from top-to-bottom, the reference images ranked in terms of relevance, i.e. from the first to the sixth smallest  $\mathcal{E}$ . Note that, in both cases, the metric is able to capture the dissimilarity between the textures under study. Even for complex textured images, like the buildings in Figure 4(c) and (d), the nearest neighbour classifier delivers good results.

Now, we turn our attention to a more quantitative analysis of the results. In Table 2, we show the results for  $k = \{1, 2, 3, 4, 5, 6\}$ . Note that, from the table, it is clear that the descriptor and its associated metric performs best when applied to homogeneous textures. This is not surprising since the Fourier approach presented here is a frequency domain one which is expected to be most stable when applied to periodic textures.

### 4. Conclusions

In this paper, we have presented a local texture descriptor which hinges in the use of stable distributions and Fourier transforms for purposes of representing the spectra of a texture in a compact manner. In this way, here, we have viewed the texture under study as a set of random variables which arise from a stochastic process. This stochastic process responds to a “heavy-tailed” distribution which can be computed via FFTs. This permits the formulation of a local texture descriptor whose recovery is computationally effi-

<sup>1</sup>This database is publicly available. It can be downloaded from <http://vismod.media.mit.edu/pub/VisTex/VisTex.tar.gz>

| Category | 1-NN   | 2-NN   | 3-NN    | 4-NN    | 5-NN    | 6-NN    |
|----------|--------|--------|---------|---------|---------|---------|
| Bark     | 40.00% | 53.33% | 80.00%  | 80.00%  | 86.67%  | 86.67%  |
| Brick    | 80.00% | 93.33% | 100.00% | 100.00% | 100.00% | 100.00% |
| Building | 60.00% | 73.33% | 73.33%  | 73.33%  | 86.67%  | 86.67%  |
| Painting | 46.67% | 53.33% | 66.67%  | 66.67%  | 66.67%  | 66.67%  |

**Table 2.** Recognition results for the six-nearest neighbours of the test textured images in the reference data set.

ciently. Moreover, this treatment allows metrics between texture-pairs to be recovered via Procrustes analysis. We have illustrated the utility of our descriptor by performing recognition tasks on a real-world texture database.

## References

- [1] J. Aloimonos and M. Swain. Shape from texture. *Biological Cybernetics*, 58(5):345–360, 1988.
- [2] J. Garding. Shape from texture for smooth curved surfaces. In *European Conference on Computer Vision*, pages 630–638, 1992.
- [3] R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.
- [4] K. Ikeuchi. Shape from regular patterns. *Artificial Intelligence*, 22:49–75, 1984.
- [5] K. Kanatani and T. Chou. Shape from texture: General principle. *Artificial Intelligence*, 38:1–48, 1989.
- [6] R. Keys. Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(6).
- [7] J. Krumm and S. A. Shafer. Shape from periodic texture using spectrogram. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 284–289, 1992.
- [8] J. Krumm and S. A. Shafer. Texture segmentation and shape in the same image. In *IEEE International Conference on Computer Vision*, pages 121–127, 1995.
- [9] J. K. Lindsey. *Statistical Analysis of Stochastic Processes in Time*. Cambridge University Press, 2004.
- [10] J. Malik and R. Rosenholtz. Computing local surface orientation and shape from texture for curved surfaces. *IJCV*, 23(2):149–168, June 1997.
- [11] T. Ojala, M. Pietikainen, and D. Harwood. A comparative study of texture measures with classification based on featured distribution. *Pattern Recognition*, 1(29):51–59, 1996.
- [12] T. Randen and J. H. Husoy. Filtering for texture classification : A comparative study. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(4):291–310, 1999.
- [13] E. Ribeiro and E. R. Hancock. Shape from periodic texture using the eigenvectors of local affine distortion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(12):1459–1465, December 2001.
- [14] B. J. Super and A. C. Bovik. Planar surface orientation from texture spatial-frequencies. *Pattern Recognition*, 28(5):729–743, May 1995.