Rotationally Invariant 3-D Texture Classification

R. Azencott, S. Jain, M. Papadakis

December 2, 2009

1 Introduction

Textures appear in most natural images, but there is no precise mathematical definition of a texture. However, there are various descriptions of textures available in the literature. These approaches can be broadly classified as structural or deterministic, and stochastic [5, 7, 12, 19, 29]. The structural approach is more suitable for describing textures that can be formed by repeating a unit pattern also known as a primitive. Examples of such textures are shown in Figure 1. Textures in which the absence of a faithful repetition of a certain pattern is the norm are better described by statistical models. Some examples of non-deterministic textures are shown in Figure 2.

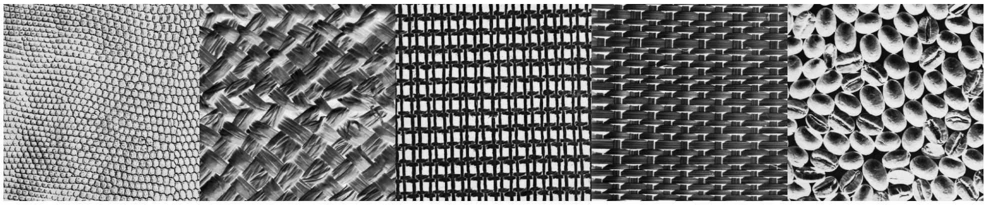

Figure 1: Examples of structural 2-D textures

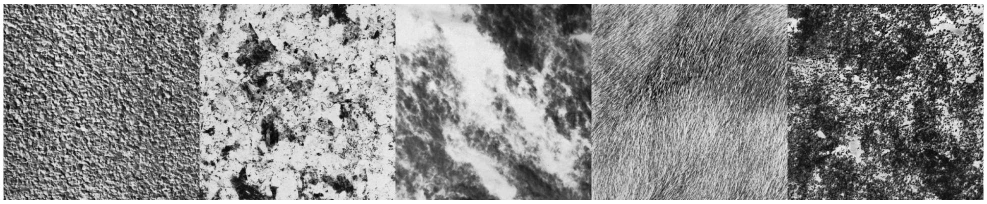

Figure 2: Examples of stochastic 2-D textures

Textures appear naturally in many classes of imaging, among them medical images used for diagnostic and research purposes. In imaging modalities such as X-ray CT and MRI, different tissues give rise to different textures, but these textures are now 3-D. Thus, efficient texture classification and segmentation routines can be used for the automatic or semi-automatic detection of anomalies. Textures arising from tissues can not be treated as those in Figure 1, where the precise perpetual repetition of the primitive determines the texture. Natural tissue variation should be taken into account. In addition, various types of noise join forces to enhance the variability in textures of this sort. Thus, stochastic models are better suited for modeling textures that appear in medical imaging. These textures are the focus of this chapter.

The first step in a texture analysis problem is to identify a good set of features or texture signatures. A feature is a certain attribute of the model. To be useful, a feature should be easily computable otherwise it can not be easily utilized. A **texture signature** is a collection of features identifying a texture. Feature vector is an alternative term that can be used instead of texture signature. In fact, the use of the noun 'vector' underscores the importance of vector-valued texture signatures versus scalar-valued signatures. The complexity of the nature of a texture becomes feasible with a multidimensional approach. The dimensionality of a texture signature is very critical in enabling texture discrimination. The feature vectors or texture signatures corresponding to two different textures should be distant enough from each other with respect to a suitable metric. Various kinds of features have been considered in the literature such as those arising from spatial frequency based techniques such as Gabor filters [2, 24, 28] and wavelets [3, 8, 30], or those arising from spatial interaction and autoregressive models such as Markov random fields [4, 13, 21]. The autoregressive and spatial interaction models are particularly attractive because they take into account the local statistical interactions of an image which capture a lot of information about textures. Of these, Gaussian Markov Random Fields (GMRF) have been studied most extensively for 2-D texture analysis [4, 7, 14, 24]. More recently, GMRFs have been used for the study of 3-D textures in [25]. The literature on 3-D texture analysis in general is extremely limited due to the tremendous computational challenges one encounters in this kind of image analysis. A few recent generalizations of the 2-D methods to 3-D can be found in [16, 17, 20, 26, 31].

We are interested in rotationally invariant classification of 3-D textures because, in applications such as medical imaging, a tissue type must be classified regardless of the position or orientation of the subject. Various authors have considered different approaches to obtain a rotationally invariant classification scheme, mostly in 2-D. In [15], Kashyap and Khotanazad propose a so-called circular symmetric autoregressive model. They fit two traditional simultaneous autoregressive models, one for the nearest neighbors and one for the diagonal nearest neighbors. By the diagonal nearest neighbors they mean points on the diagonal at a unit distance. These points do not lie on the original lattice so the gray scale values here are obtained via linear interpolation. The parameters for these models are used as rotation invariant features. This kind of circularly symmetric models are also considered in [24], where higher order neighborhoods beyond first order ones are also employed. In [24], the authors also propose circularly symmetric Gabor filter-based features. The main shortcoming of these models is that they impose an isotropic structure on possibly non-isotropic textures. A mathematically rigorous treatment of isotropic textures can be found in [22]. However, when we are dealing with textures that show directional characteristics, isotropic models limit the flexibility in texture classification. This problem is also recognized in [7], where the authors model the textures using a continuous stationary Gaussian random field. Rotations are defined via a continuous counterpart of the discrete power spectral density associated to a discrete model defined on a finite lattice. This is similar to the approach that we describe for rotation of textures in Section 4.2. Using the continuous power spectral density, they derive a likelihood function for the rotation parameter. The classification is then carried out in two steps. First, they obtain a maximum likelihood estimate for the rotation angle and then they compare against each texture in the training set for classification. This approach is shown to work well in 2-D but a 3-D version would be computationally extremely expensive since the likelihood function they obtain constitutes a product over all the lattice nodes. A 3-D rotation invariant approach for texture classification using local binary patterns is considered in [9]. To the best of our knowledge there has been no previous attempt of a 3-D rotation invariant texture classification using GMRF.

Instead of using a rotationally invariant probabilistic model for a texture, we aim at defining a rotationally invariant distance between two texture signatures which might themselves be non-isotropic. Two different non-isotropic textures might not be distinguishable if isotropy of the texture signatures is imposed. Thus, using non-isotropic signatures, and implementing the rotational invariance via a distance function, makes the models more flexible.

We model discrete textures, i.e. textures defined on a discrete lattice, using GMRF following the approach of Rama Chellappa described for 2-D textures in [4], and extended to 3-D volumes in [25]. GMRFs are particularly useful for the following reasons:

- These stochastic models have a very well understood mathematical structure,
- The computability of Kullback-Leibler distance (see Section 4.1),
- Relatively low computational cost of parameter estimation and texture synthesis.

The texture signature is obtained by fitting a GMRF model to each of a finite set of rotations of the texture. This finite set of rotations is obtained by a uniform sampling of $SO(3)$. We define a rotationally invariant distance between these texture signatures in Section 4. Since the rotation of a discrete texture is not well-defined, we propose a continuous counterpart of the discrete texture that we wish to rotate. We consider a continuous texture to be a realization of a stationary Gaussian process on $\mathbb{R}^3$. *Then the concept of rotating a texture takes a natural form and is shown to be (Equation (20)) equivalent to rotating the corresponding autocovariance function.* The conversion between discrete and continuous textures is achieved via Isotropic Multiresolution Analysis developed in [27]. This is discussed in detail in Section 4.2.

2 Isotropic Multiresolution Analysis and Fast Algorithms

Example 2.1

3 Description of the GMRF Model

To describe the GMRF model, we follow the notation in [25]. The image/volume is defined on a 2/3-D lattice denoted by $\mathbf{\Lambda}$. By a lattice, we mean the following:

$$\mathbf{\Lambda}^{2D} = \{\boldsymbol{k} = (i,j) \mid 1 \le i \le M, 1 \le j \le N\},$$

$$\mathbf{\Lambda}^{3D} = \{\boldsymbol{k} = (i,j,k) \mid 1 \le i \le M, 1 \le j \le N, 1 \le k \le P\}.$$

where, i, j, k are integers and M, N, P denote the size of the lattice in each dimension. The total number of sites or nodes on the lattice are denoted by N_T. In the 2-D case, $N_T = MN$ and for 3-D, $N_T = MNP$. In the rest of this chapter, we assume that we work with a 3-D lattice though most of the discussion in this section applies verbatim to 2-D.

A **neighborhood** $\eta_{\boldsymbol{k}}$ of node $\boldsymbol{k}$ is a subset of the lattice, $\eta_{\boldsymbol{k}} \subset \boldsymbol{\Lambda}$ such that,

- $\boldsymbol{k} \notin \eta_{\boldsymbol{k}}$,
- $\boldsymbol{l} \in \eta_{\boldsymbol{k}} \Rightarrow \boldsymbol{k} \in \eta_{\boldsymbol{l}}$.

A **neighborhood system** $\boldsymbol{\eta}$, is the collection of neighborhoods at all nodes, $\boldsymbol{\eta} = \{\eta_{\boldsymbol{k}}, \boldsymbol{k} \in \boldsymbol{\Lambda}\}$.

A **clique** C defined with respect to $\boldsymbol{\eta}$, is a subset of $\boldsymbol{\Lambda}$ such that C either contains a single node, or all nodes in C are neighbors. The set of all cliques in a neighborhood is denoted by $\mathcal{C}$ and $\mathcal{C}_n$, for $n \in \mathbb{N}^+$, denotes the set of cliques with n nodes.

Let $\mathbf{X}$ be a family of random variables defined on the lattice $\boldsymbol{\Lambda}$ via $\mathbf{X} := \{X_1, \ldots, X_{N_T}\}$. We refer to $\mathbf{X}$ as a **random field** and use the notation $\mathbf{X} = \mathbf{x}$ for the joint event $\{X_i = x_i, i = 1, \ldots, N_T\}$, where $\mathbf{x} = \{x_1, \ldots, x_{N_T}\}$ is called a **configuration** of $\mathbf{X}$. An image/volume is a configuration or realization of $\mathbf{X}$. Each x_i belongs to a (usually finite) set $\mathcal{A}$, referred to as the **alphabet** (or gray-levels in image processing terminology).

3.1 Gibbs and Markov Random Fields

In this subsection, we assume that the alphabet $\mathcal{A}$ is discrete to keep the discussion simple. The technical details required to describe random fields are not needed for our discussion on GMRF in Section 3.2, even though the alphabet in that case is $\mathbb{R}$. For the general theory of Gibbs and Markov fields with a continuous alphabet, the reader is referred to Chapter 2 in [11]. A random field $\mathbf{X}$ is referred to as a **Gibbs Random Field (GRF)** if it satisfies the following probability distribution:

$$\mathbb{P}(\mathbf{X} = \mathbf{x}) = \frac{1}{Z} \exp\left(\frac{U(\mathbf{x})}{\mathcal{T}}\right).$$

where Z is a positive normalizing constant known as the **partition function**, $\mathcal{T}$ is the temperature and U is referred to as the **Gibbs energy**.

For a finite lattice $\boldsymbol{\Lambda}$ with a symmetric neighborhood structure $\boldsymbol{\eta}$, an example of Gibbs energy function is defined as:

$$U(\mathbf{x}) = \sum_{\boldsymbol{k} \in \boldsymbol{\Lambda}, C \in \mathcal{C}_1} V_C(x_{\boldsymbol{k}}) + \sum_{\boldsymbol{k} \in \boldsymbol{\Lambda}} \sum_{\boldsymbol{l} \in \eta_{\boldsymbol{k}}, C \in \mathcal{C}_2} V_C(x_{\boldsymbol{k}}, x_{\boldsymbol{l}}), \tag{1}$$

where V_C are known as the **clique potentials**.

A random field $\mathbf{X}$ is called a **Markov Random Field (MRF)** on the lattice $\boldsymbol{\Lambda}$ with respect to a neighborhood system $\boldsymbol{\eta}$ if it satisfies the following conditions:

- $\mathbb{P}(\mathbf{X} = \mathbf{x}) > 0$, for all $\mathbf{x} \in \mathcal{A}^{N_T}$, and
- $\mathbb{P}(X_{\boldsymbol{k}} = x_{\boldsymbol{k}} \mid X_{\boldsymbol{l}} = x_{\boldsymbol{l}},\ \forall \boldsymbol{l} \in \boldsymbol{\Lambda} \backslash \boldsymbol{k}) = \mathbb{P}(X_{\boldsymbol{k}} = x_{\boldsymbol{k}} \mid X_{\boldsymbol{l}} = x_{\boldsymbol{l}},\ \forall \boldsymbol{l} \in \eta_{\boldsymbol{k}})$.

The **local characteristic** at node $\boldsymbol{k}$ is the function $\pi^{\boldsymbol{k}} : \mathcal{A}^{N_T} \to [0,1]$ defined by

$$\pi^{\boldsymbol{k}}(\mathbf{x}) = \mathbb{P}(X_{\boldsymbol{k}} = x_{\boldsymbol{k}} \mid X_{\boldsymbol{l}} = x_{\boldsymbol{l}} \forall \boldsymbol{l} \in \eta_{\boldsymbol{k}}).$$

The family $\{\pi^{\boldsymbol{k}}\}_{\boldsymbol{k}\in\boldsymbol{\Lambda}}$ is called the **local specification** of the MRF.

An MRF is characterized by its local property, (the conditional probability density at a node given its neighbors) while a GRF is characterized by its global property (the Gibbs distribution). The **Hammersley-Clifford theorem** (see e.g. [10]) states that a random field is a GRF if and only if it is an MRF. Thus, it gives us the flexibility of designing an MRF using local or global properties depending on their availability for a specific application.

3.2 Gaussian Markov Random Field

A special class of MRF arises from the following clique potentials:

$$V_C(x_{\boldsymbol{k}}) = \frac{(x_{\boldsymbol{k}} - \mu_{\boldsymbol{k}})^2}{2\sigma^2}, \forall C \in \mathcal{C}_1,$$

and

$$V_C(x_{\boldsymbol{k}}, x_{\boldsymbol{l}}) = -\theta_{\boldsymbol{k},\boldsymbol{l}} \frac{(x_{\boldsymbol{k}} - \mu_{\boldsymbol{k}})(x_{\boldsymbol{l}} - \mu_{\boldsymbol{l}})}{\sigma^2}, \boldsymbol{l} \in \eta_{\boldsymbol{k}}, \forall C \in \mathcal{C}_2,$$

where $\mu_{\boldsymbol{k}}$ determines the mean at node $\boldsymbol{k}$, and as we shall see shortly, $\theta_{\boldsymbol{k},\boldsymbol{l}}$ are parameters related to the covariance matrix of the random field. Substituting these clique potentials in (1) we obtain,

$$U(\mathbf{x}) = \frac{1}{2\sigma^2} \sum_{\boldsymbol{k}\in\boldsymbol{\Lambda}} (x_{\boldsymbol{k}} - \mu_{\boldsymbol{k}})^2 - \frac{1}{\sigma^2} \sum_{\boldsymbol{k}\in\boldsymbol{\Lambda}} \sum \boldsymbol{l} \in \eta_{\boldsymbol{k}} (x_{\boldsymbol{k}} - \mu_{\boldsymbol{k}}) \theta_{\boldsymbol{k},\boldsymbol{l}} (x_{\boldsymbol{l}} - \mu_{\boldsymbol{l}}).$$

We now assume that the alphabet is $\mathbb{R}$. Thus, the joint probability density of all N_T nodes in $\boldsymbol{\Lambda}$ is given by:

$$p(\mathbf{x}) = \frac{\sqrt{\det(\mathbf{B})}}{\sqrt{(2\pi\sigma^2)^{N_T}}} \exp\left[-\frac{(\mathbf{x} - \boldsymbol{\mu})^T \mathbf{B} (\mathbf{x} - \boldsymbol{\mu})}{2\sigma^2}\right], \tag{2}$$

where $\boldsymbol{\mu}$ is an $N_T \times 1$ vector of means and $\mathbf{B} = [b_{\boldsymbol{l}\boldsymbol{k}}]$ is the following $N_T \times N_T$ matrix:

$$b_{\boldsymbol{l}\boldsymbol{k}} = \begin{cases} 1, & \text{if } \boldsymbol{l} = \boldsymbol{k}, \\ -\theta_{\boldsymbol{l},\boldsymbol{k}}, & \text{if } \boldsymbol{l} \in \eta_{\boldsymbol{k}}, \\ 0, & \text{else.} \end{cases} \tag{3}$$

The function in Equation (2) is the joint probability density function of a multivariate Gaussian distribution with covariance matrix $\Sigma = \sigma^2 \mathbf{B}^{-1}$ and mean vector $\boldsymbol{\mu}$. Hence, this random field is referred to as a Gaussian Markov Random Field. The necessary and sufficient condition for p defined in Equation (2) to be a density function is that $\mathbf{B}$ is a positive matrix. The Markov property now reads

$$\mathbb{P}(X_{\boldsymbol{k}} \in A \mid X_{\boldsymbol{l}},\ \forall \boldsymbol{l} \in \boldsymbol{\Lambda} \backslash \boldsymbol{k}) = \mathbb{P}(X_{\boldsymbol{k}} \in A \mid X_{\boldsymbol{l}},\ \forall \boldsymbol{l} \in \eta_{\boldsymbol{k}}),\ \forall A \subset \mathbb{R}.$$

This is true because the conditional probability density $p(x_{\boldsymbol{k}} \mid x_{\boldsymbol{l}}, \boldsymbol{l} \in \boldsymbol{\Lambda} \backslash \boldsymbol{k})$ is given by [25]:

$$\begin{aligned} p(x_{\boldsymbol{k}} \mid x_{\boldsymbol{l}}, \boldsymbol{l} \in \boldsymbol{\Lambda} \backslash \boldsymbol{k}) &= p(x_{\boldsymbol{k}} \mid x_{\boldsymbol{l}}, \boldsymbol{l} \in \eta_{\boldsymbol{k}}) \\ &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2\sigma^2} \left(x_{\boldsymbol{k}} - \mu_{\boldsymbol{k}} - \sum_{\boldsymbol{l} \in \eta_{\boldsymbol{k}}} \theta_{\boldsymbol{k},\boldsymbol{l}} (x_{\boldsymbol{l}} - \mu_{\boldsymbol{l}}) \right)^2 \right]. \qquad (4) \end{aligned}$$

The conditional distribution corresponding to this density is referred to as the local characteristic at node $\boldsymbol{k}$ for the continuous alphabet case. This conditional distribution is defined in the sense of regular conditional probability (see Theorem V.8.1 in [23]). Now, the gray level at the node $\boldsymbol{k}$ can be expressed as a linear combination of the gray levels at the neighboring nodes:

$$x_{\boldsymbol{k}} = \mu_{\boldsymbol{k}} + \sum_{\boldsymbol{l} \in \eta_{\boldsymbol{k}}} \theta_{\boldsymbol{k},\boldsymbol{l}} (x_{\boldsymbol{l}} - \mu_{\boldsymbol{l}}) + e_{\boldsymbol{k}}, \qquad (5)$$

where the correlated Gaussian noise, $\mathbf{e} = (e_1, \ldots, e_{N_T})$, has the following structure:

$$\mathbb{E}[e_{\boldsymbol{k}} e_{\boldsymbol{l}}] = \begin{cases} \sigma^2, & \boldsymbol{k} = \boldsymbol{l}, \\ -\theta_{\boldsymbol{k},\boldsymbol{l}} \sigma^2, & \boldsymbol{l} \in \eta_{\boldsymbol{k}}, \\ 0, & \text{else}. \end{cases} \qquad (6)$$

In [4, 14], Equations (5) and (6) are used to define GMRF. This model is referred to as Conditional Markov (CM) model in [14] where the various parameter estimation schemes for this model are discussed. Also contained in [14] is a discussion on the choice of neighbors.

For models that we use, we assume the following spatial symmetry for the parameters θ:

$$\theta_{\boldsymbol{k},\boldsymbol{l}} = \theta_{\boldsymbol{k},-\boldsymbol{l}}. \qquad (7)$$

Hence, the neighborhoods must be defined so that $\boldsymbol{l} \in \eta_{\boldsymbol{k}} \Rightarrow -\boldsymbol{l} \in \eta_{\boldsymbol{k}}$. We also assume the stationarity of the model. Hence, $\theta_{\boldsymbol{k},\boldsymbol{l}}$ only depends on $\boldsymbol{k} - \boldsymbol{l}$ and the neighborhood $\eta_{\boldsymbol{k}}$ has the same structure at each $\boldsymbol{k}$. By the same structure we mean that the set $\{\boldsymbol{k} - \boldsymbol{l}\}_{\boldsymbol{l} \in \eta_{\boldsymbol{k}}}$ is the same for all $\boldsymbol{k}$. This set is denoted by η. Similarly, the mean, $\mu_{\boldsymbol{k}}$ is also constant across all the nodes and hence we drop the subscript $\boldsymbol{k}$ and denote the mean by μ. Due to the symmetry assumed in Equation (7), if $\boldsymbol{r} \in \eta$ then $-\boldsymbol{r}$ is also in η. We use η^+ to denote half of the elements of η such that only one of $\boldsymbol{r}$ or $-\boldsymbol{r}$ is in η^+. For instance, in the 2-D case, if $\eta = \{(1,0), (0,1), (-1,0), (0,-1)\}$ then $\eta^+ = \{(1,0), (0,1)\}$. Thus, for the stationary case, Equation (5) takes the following form

$$x_{\boldsymbol{k}} = \mu + \sum_{\boldsymbol{r} \in \eta^+} \theta_{\boldsymbol{r}} (x_{\boldsymbol{k}-\boldsymbol{r}} + x_{\boldsymbol{k}+\boldsymbol{r}} - 2\mu) + e_{\boldsymbol{k}}. \qquad (8)$$

The vector of parameters $[\theta_{\boldsymbol{r}}, \boldsymbol{r} \in \eta^+]$ is denoted by $\boldsymbol{\theta}$.

In the rest of the discussion we assume that $\mu = 0$, unless otherwise mentioned.

3.3 Parameter Estimation

Chellappa and Kashyap discuss various parameter estimation schemes in [4, 14]. These, amongst others, include a Maximum likelihood scheme, a coding scheme due to Besag and

Least squares (LS). The ML-estimates have good statistical properties but are expensive to compute because they require numerical optimization techniques. We follow the LS estimation scheme given in Section 4.3 of [4]. The statistical properties of this scheme are analyzed in [14]. For a given realization $\mathbf{x}$, the estimates are given by the following statistics:

$$\widehat{\boldsymbol{\theta}}(\mathbf{x}) = (\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{x}, \tag{9}$$

$$\widehat{\sigma^2}(\mathbf{x}) = \frac{1}{N_T}\left(\mathbf{x} - \mathbf{Y}\widehat{\boldsymbol{\theta}}\right)^T\left(\mathbf{x} - \mathbf{Y}\widehat{\boldsymbol{\theta}}\right), \tag{10}$$

where $\mathbf{Y} = [\mathbf{y}_{\boldsymbol{r}}], \boldsymbol{r} \in \boldsymbol{\Lambda}$ and $\mathbf{y}_{\boldsymbol{r}} = [x_{\boldsymbol{l}} + x_{-\boldsymbol{l}}, \boldsymbol{l} \in \eta^+_{\boldsymbol{r}}]$. Thus, if m is the size of the half-neighborhood $\eta^+_{\boldsymbol{r}}$ then $\mathbf{Y}$ is a $N_T \times m$ matrix. For a derivation of these equations see Appendix B of [25]. Using the estimate, $\widehat{\boldsymbol{\theta}}$, from (9), the expression for estimate of σ^2 reduces to

$$\widehat{\sigma^2}(\mathbf{x}) = \frac{1}{N_T}\left(\mathbf{x}^T\mathbf{x} - \widehat{\boldsymbol{\theta}}^T\mathbf{Y}^T\mathbf{x}\right). \tag{11}$$

A sufficient condition on $\boldsymbol{\theta}$ for the corresponding $\mathbf{B}(\boldsymbol{\theta})$ (see Equation (3)) to be a positive matrix is $|\boldsymbol{\theta}| < 0.5$ [18]. Here, $|\boldsymbol{\theta}|$ denotes the ℓ^1-norm of the vector $\boldsymbol{\theta}$. As shown in [18], in general, this sufficient condition only represents a subset of the valid parameter space for $\boldsymbol{\theta}$. But, in the case of order one neighborhood, this is also a necessary condition.

In the light of Equation (9) and the constraint on $\boldsymbol{\theta}$ for order one neighborhood, we have to solve the following constrained optimization problem to estimate $\boldsymbol{\theta}$:

Find

$$\min_{\boldsymbol{\theta}} \|\mathbf{Y}^T\mathbf{Y}\boldsymbol{\theta} - \mathbf{Y}^T\mathbf{x}\|_2,$$

subject to,

$$\sum_{i=1}^{3} |\theta_i| < 0.5.$$

The entries of the matrix $\mathbf{Y}^T\mathbf{Y}$ and the vector $\mathbf{Y}^T\mathbf{x}$ can be calculated from the auto-covariance function of $\mathbf{x}$. This will facilitate fast calculation of the rotationally invariant distance defined in Section 4. For a stationary random process $\mathbf{X}$ on $\mathbb{Z}^3$, the auto-covariance function is given by

$$\rho(\boldsymbol{l}) = \mathbb{E}[\mathbf{X}(\boldsymbol{l})\mathbf{X}(0)].$$

In particular, for a infinite extent GMRF whose local specifications are given by (4), the auto-covariance function ρ decays to zero as $|\boldsymbol{l}|$ goes to ∞. This is due to the fact that the power spectral density $\widehat{\rho}$ is the inverse of a positive trigonometric polynomial.

Due to the ergodicity (implicit in the model, see Theorem III.4.4 in [1]), the auto-covariance function, ρ, can be approximated by

$$\rho_0(\boldsymbol{l}) = \frac{1}{N_T}\sum_{\boldsymbol{r}\in\boldsymbol{\Lambda}} x_{\boldsymbol{r}}x_{\boldsymbol{r}+\boldsymbol{l}}, \quad \text{for all} \quad \boldsymbol{l} \in \boldsymbol{\Lambda}, \tag{12}$$

for a sufficiently large N_T. Using the Discrete Fourier Transform, we obtain

$$\widehat{\rho_0}(\boldsymbol{k}) = \widehat{\mathbf{x}}(\boldsymbol{k})\overline{\widehat{\mathbf{x}}(\boldsymbol{k})} = |\widehat{\mathbf{x}}(\boldsymbol{k})|^2 \quad \text{for all} \quad \boldsymbol{k} \in \boldsymbol{\Lambda}', \tag{13}$$

where $\mathbf{\Lambda}'$ is a grid of $\mathbb{T}^3$, similar to $\mathbf{\Lambda}$. Note that those values of $\widehat{\rho_0}$ give an approximation of the power spectral density $\widehat{\rho}$ defined on $\mathbb{T}^3$, at the grid points $\mathbf{\Lambda}'$. Equation (13) facilitates efficient calculation of ρ_0 via FFT. Now, the entries of the vector $\mathbf{Y}^T\mathbf{x}$ are expressed in terms of ρ_0 as follows:

$$\begin{aligned}(\mathbf{Y}^T\mathbf{x})_{\boldsymbol{r}} &= \sum_{\boldsymbol{l}\in\mathbf{\Lambda}} x_{\boldsymbol{l}}(x_{\boldsymbol{l}+\boldsymbol{r}} + x_{\boldsymbol{l}-\boldsymbol{r}}) \\ &= N_T(\rho_0(\boldsymbol{r}) + \rho_0(-\boldsymbol{r})),\end{aligned}$$

for each $\boldsymbol{r} \in \eta^+$. Similarly, the entries of the matrix $\mathbf{Y}^T\mathbf{Y}$ are given by

$$\begin{aligned}(\mathbf{Y}^T\mathbf{Y})_{(\boldsymbol{k},\boldsymbol{r})} &= \sum_{\boldsymbol{l}\in\mathbf{\Lambda}} (x_{\boldsymbol{l}+\boldsymbol{k}} + x_{\boldsymbol{l}-\boldsymbol{k}})(x_{\boldsymbol{l}+\boldsymbol{r}} + x_{\boldsymbol{l}-\boldsymbol{r}}) \\ &= N_T(\rho_0(\boldsymbol{r}-\boldsymbol{k}) + \rho_0(\boldsymbol{r}+\boldsymbol{k}) + \rho_0(-\boldsymbol{r}-\boldsymbol{k}) + \rho_0(-\boldsymbol{r}+\boldsymbol{k})),\end{aligned}$$

for each $(\boldsymbol{k}, \boldsymbol{r}) \in \eta^+ \times \eta^+$. Thus, for any given zero mean discrete texture $\mathbf{x}$, that is not necessarily a GMRF, we can calculate the ρ_0 using (12). Then we can estimate the parameters for a GMRF with neighborhood of order one using the above equations. *We refer to the model with these parameters as the GMRF fitted to the texture* $\mathbf{x}$.

3.4 Synthesizing a GMRF

Next, we briefly discuss the algorithm for sampling a stationary GMRF with toroidal boundaries. For a fixed neighborhood structure $\boldsymbol{\eta}$ and known parameters, $\theta_{\boldsymbol{k},\boldsymbol{l}}, \boldsymbol{\mu}, \sigma$, we can write Equation (5) as:

$$\mathbf{B}(\mathbf{x} - \boldsymbol{\mu}) = \mathbf{e}.$$

where $\mathbf{e}$ is a zero mean Gaussian noise sequence with the covariance structure defined in (6). This noise sequence can be written as

$$\mathbf{e} = \sqrt{\mathbf{B}}\mathbf{e}_0,$$

where $\mathbf{e}_0$ is a zero mean Gaussian noise sequence with covariance matrix given by σ^2 times the identity. This is easily generated with a standard random number generator in MATLAB. Hence, a realization of $\mathbf{X}$ is obtained via

$$\mathbf{x} = \mathbf{B}^{-1}\sqrt{\mathbf{B}}\mathbf{e}_0 + \boldsymbol{\mu} = \mathbf{B}^{\frac{-1}{2}}\mathbf{e}_0 + \boldsymbol{\mu}.$$

The stationarity assumption implies that $\mu_{\boldsymbol{k}} = \mu, \forall \boldsymbol{k} \in \mathbf{\Lambda}$ and that $\theta_{\boldsymbol{k},\boldsymbol{l}}$ only depends on the difference $\boldsymbol{k} - \boldsymbol{l}$. This along with the toroidal boundary condition makes $\mathbf{B}$ a block circulant matrix which can be inverted efficiently using FFT. For details of this algorithm and block circulant matrices, see [25], Chapter 4.

4 Rotationally Invariant Distance

Now, to obtain a rotationally invariant texture distance, we begin by defining a texture signature. We use a neighborhood of order one, $\eta^+ = \{(1,0,0),(0,1,0),(0,0,1)\}$. Thus, $\boldsymbol{\theta}$

is a three-dimensional vector. Recall that for a given texture $\mathbf{x}$, $\widehat{\boldsymbol{\theta}}(\mathbf{x})$ and $\widehat{\sigma^2}(\mathbf{x})$, denote the parameters of the GMRF fitted to $\mathbf{x}$. We define the texture signature $\Gamma_{\mathbf{x}}$, via

$$\Gamma_{\mathbf{x}}(\boldsymbol{\alpha}) = \left[\widehat{\boldsymbol{\theta}}(\mathcal{R}_{\boldsymbol{\alpha}}\mathbf{x}), \widehat{\sigma^2}(\mathcal{R}_{\boldsymbol{\alpha}}\mathbf{x})\right], \tag{14}$$

for all $R_{\boldsymbol{\alpha}} \in SO(3)$, where $\boldsymbol{\alpha} = (\alpha, \beta, \gamma)$ is the Euler angle parametrization for the rotation $R_{\boldsymbol{\alpha}}$ (see Equation (22) below), and $\mathcal{R}_{\boldsymbol{\alpha}}$ is the rotation operator induced by $R_{\boldsymbol{\alpha}}$ on $L^2(\mathbb{R}^3)$. Notice the abuse of notation when we write $\mathcal{R}_{\boldsymbol{\alpha}}\mathbf{x}$, since $\mathbf{x}$ is not defined on $\mathbb{R}^3$ but on $\boldsymbol{\Lambda}$, which is a finite sub-lattice of $\mathbb{Z}^3$. For now, just think of $\mathbf{x}$ as samples of some continuous infinite extent texture, $\mathbf{x}_{cont}$, and then $\mathcal{R}_{\boldsymbol{\alpha}}\mathbf{x}$ denotes the samples of $\mathcal{R}_{\boldsymbol{\alpha}}\mathbf{x}_{cont}$ on $\boldsymbol{\Lambda}$. This idea of rotating and resampling the texture will be made more precise in Section 4.2.

Now, we define a distance between two textures by the following expression:

$$\min_{\boldsymbol{\alpha}_0 \in SO(3)} \int_{SO(3)} \text{KLdist}\left(\Gamma_{\mathbf{x}_1}(\boldsymbol{\alpha}), \Gamma_{\mathbf{x}_2}(\boldsymbol{\alpha}\boldsymbol{\alpha}_0)\right) d\boldsymbol{\alpha}, \tag{15}$$

where KLdist(.,.) is the KL-distance between two Gaussian densities defined in Equation (19), and the product $\boldsymbol{\alpha}\boldsymbol{\alpha}_0$ denotes the Euler angles corresponding to the rotation operator $\mathcal{R}_{\boldsymbol{\alpha}}\mathcal{R}_{\boldsymbol{\alpha}_0}$. The integration is carried out with respect to the Haar measure on $SO(3)$. For the Euler angle parametrization of $SO(3)$ with the ZYZ-convention (see Equation (22) below), the Haar measure is given by (See Chapter 5 of [6]),

$$d\boldsymbol{\alpha} = \sin(\beta) d\alpha d\beta d\gamma, \tag{16}$$

where $d\alpha$, $d\beta$ and $d\gamma$ stand for the Lebesgue measure. The Haar measure is usually normalized by $\frac{1}{8\pi^2}$.

4.1 KL-distance Between two Gaussian Markov Random Fields

The Kullback-Leibler distance between two N-dimensional probability distributions with joint probability density functions p_1 and p_2 is given by

$$D(p_2 || p_1) = \int_{\mathbb{R}^N} p_2(\mathbf{x}) \log_e \left(\frac{p_2(\mathbf{x})}{p_1(\mathbf{x})}\right) d\mathbf{x}. \tag{17}$$

The joint probability density function (p.d.f) of a multivariate Gaussian distribution with covariance matrix Σ and mean vector $\boldsymbol{\mu}$ is given by

$$p(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^N \det(\Sigma)}} \exp\left[-\frac{(\mathbf{x}-\boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x}-\boldsymbol{\mu})}{2}\right]. \tag{18}$$

Now suppose, we have two N-dimensional Gaussian probability density functions, p_1 and p_2, with means $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$, and covariance matrices Σ_1 and Σ_2 respectively. Then substituting the formulae for p_1 and p_2 in (17) yields the following expression for the KL-distance between two Gaussian distributions:

$$\frac{1}{2}\log_e\left(\frac{\det \Sigma_1}{\det \Sigma_2}\right) + \frac{1}{2}\left(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\right)^T \Sigma_1^{-1} \left(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1\right) + \frac{1}{2}\text{Trace}(\Sigma_1^{-1}\Sigma_2) - \frac{N}{2}.$$

Note that the distance defined above is not symmetric. Hence, if we exchange Σ_1 and Σ_2, we get another expression:

$$\frac{1}{2}\log_e\left(\frac{\det\Sigma_2}{\det\Sigma_1}\right)+\frac{1}{2}\left(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2\right)^T\Sigma_2^{-1}\left(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2\right)+\frac{1}{2}\mathrm{Trace}(\Sigma_2^{-1}\Sigma_1)-\frac{N}{2}.$$

Adding these two expressions yields the following expression, which is symmetric:

$$\frac{1}{2}\left(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2\right)^T\left(\Sigma_2^{-1}+\Sigma_1^{-1}\right)\left(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2\right)+\frac{1}{2}\mathrm{Trace}(\Sigma_2^{-1}\Sigma_1+\Sigma_1^{-1}\Sigma_2)-N.$$

Assuming zero means for both the p.d.fs, we conclude that the KL-distance between two Gaussian distributions is:

$$\frac{1}{2}\mathrm{Trace}(\Sigma_2^{-1}\Sigma_1+\Sigma_1^{-1}\Sigma_2-2I_{N\times N}). \tag{19}$$

Since in our case, Σ_1 and Σ_2 are block circulant, (19) can be calculated efficiently using FFT.

4.2 Rotation of Textures

As pointed out in the introduction of this chapter, rotating a discrete texture is not well-defined. Hence, we define a continuous counterpart of the discrete texture we wish to rotate. We model a continuous texture as a realization of a stationary Gaussian random field, $\mathbf{X}_{cont}$, on $\mathbb{R}^3$ with a square integrable autocovariance function. We assume that the texture remains invariant under rotation by π about any line passing through the origin. This is consistent with the symmetry assumptions for the discrete texture. We further assume that the autocovariance function of a texture, ρ_{cont}, belongs to the zero resolution space, V_0, of the IMRA described in Example 2.1. Recall that V_0 is the closed linear span of the set $\{T_{\boldsymbol{k}}\phi\}_{\boldsymbol{k}\in\mathbb{Z}^3}$. This assumption implies that the power spectral density is compactly supported, inside a radial set. Thus, its support remains invariant under all rotations. Hence, the autocovariance function remains in the same resolution space after rotation and can therefore be sampled at the same sampling rate. This is the advantage of assuming that ρ_{cont} belongs to V_0.

The sequence of coefficients $\{\langle\rho_{cont},T_{\boldsymbol{k}}\phi\rangle\}_{\boldsymbol{k}\in\mathbb{Z}^3}$ is denoted by ρ. This sequence can be considered as the samples of ρ_{cont} on $\mathbb{Z}^3$. In fact, if we assume that the power spectral density $\widehat{\rho_{cont}}$ is supported on the ball $B(0,2b_1)$ where $\widehat{\phi}$ is equal to one, then $\langle\rho_{cont},T_{\boldsymbol{k}}\phi\rangle=\rho_{cont}(\boldsymbol{k})$ and $\rho_{cont}=\sum_{\boldsymbol{k}\in\mathbb{Z}^3}\rho_{cont}(\boldsymbol{k})T_{\boldsymbol{k}}\phi$.

Next, observe that the autocovariance function of $\mathcal{R}_{\boldsymbol{\alpha}}\mathbf{X}_{cont}$ is given by $\mathcal{R}_{\boldsymbol{\alpha}}\rho_{cont}$:

$$\begin{aligned}\mathbb{E}[\mathcal{R}_{\boldsymbol{\alpha}}\mathbf{X}_{cont}(\boldsymbol{s})\mathcal{R}_{\boldsymbol{\alpha}}\mathbf{X}_{cont}(0)] &= \mathbb{E}[\mathbf{X}_{cont}(R_{\boldsymbol{\alpha}}^T\boldsymbol{s})\mathbf{X}_{cont}((R_{\boldsymbol{\alpha}}^T0)]\\ &= \rho_{cont}(R_{\boldsymbol{\alpha}}^T\boldsymbol{s})=\mathcal{R}_{\boldsymbol{\alpha}}\rho_{cont}(\boldsymbol{s}).\end{aligned} \tag{20}$$

Now, since we assumed that $\widehat{\rho_{cont}}$ is supported on the ball $B(0,2b_1)$, $\mathcal{R}_{\boldsymbol{\alpha}}\rho_{cont}$ is also supported on the same ball. With a slight abuse of notation, the sequence of samples, $\{\langle\mathcal{R}_{\boldsymbol{\alpha}}\rho_{cont},T_{\boldsymbol{k}}\phi\rangle\}_{\boldsymbol{k}\in\mathbb{Z}^3}$ is denoted by $\mathcal{R}_{\boldsymbol{\alpha}}\rho$. Also note that

$$\rho_{cont}(\boldsymbol{k})=\mathbb{E}[\mathbf{X}_{cont}(\boldsymbol{k})\mathbf{X}_{cont}(0)]=\mathbb{E}[\mathbf{X}(\boldsymbol{k})\mathbf{X}(0)],$$

i.e., the autocovariance function of $\mathbf{X}$, the samples of $\mathbf{X}_{cont}$ on $\mathbb{Z}^3$, is ρ.

Given a realization $\mathbf{x}$ of a discrete texture $\mathbf{X}$ on the finite lattice $\mathbf{\Lambda}$, we can calculate ρ_0 corresponding to $\mathbf{x}$ using (12). As pointed out in Section 3.3 this is an approximation for ρ on $\mathbf{\Lambda}$ if $\mathbf{\Lambda}$ is sufficiently big. Assuming that the values of ρ are negligible on $\mathbb{Z}^3 \setminus \mathbf{\Lambda}$, the parameters of the GMRF model fitted to the 'rotated texture' denoted by $\mathcal{R}_{\boldsymbol{\alpha}}\mathbf{x}$ can be calculated using $\mathcal{R}_{\boldsymbol{\alpha}}\rho$. This formalizes the idea of a 'discrete' rotation that we had alluded to in Section 4. This is an excellent example of how MRAs in general and IMRA in particular act as a bridge between the digital or discrete, and the analog or continuous domains. We make sense of a discrete rotation by first converting to the analog domain via the IMRA, rotating in the analog domain, where rotations are well defined, and finally, converting back to the discrete domain via sampling.

In principle, we can calculate the sequence $\{\langle \mathcal{R}_{\boldsymbol{\alpha}}\rho_{cont}, T_{\boldsymbol{k}}\phi\rangle\}_{\boldsymbol{k}\in\mathbb{Z}^3}$ exactly, but that is computationally very expensive. Instead, we resample ρ on a finer grid using the IMRA. This works by performing one or more steps of the reconstruction algorithm described in Section 2, assuming that the detail or high-pass component is zero at each reconstruction step. If we use dyadic dilation, for example, then one step of the reconstruction gives the coefficients $\left\{\left\langle \rho_{cont}, T_{\frac{\boldsymbol{k}}{2}} D\phi \right\rangle\right\}_{\boldsymbol{k}\in\mathbb{Z}^3}$. These can be considered as the samples of the covariance function on the denser grid $\frac{\mathbb{Z}^3}{2}$. We then rotate this covariance function defined on a denser grid by using linear interpolation to get the values at grid points that do not lie on the integer lattice after rotation. This is not exact but the error can be made smaller by resampling on a finer grid. In practice we see that just one or two steps of the reconstruction algorithm are enough to give good results. This is demonstrated in Section 5, where we discuss the results from our experimental study.

4.3 Practical Implementation of the Distance

For the practical implementation of the distance defined in Equation (15), we must discretize the integral over $\boldsymbol{\alpha}$. To this end, we discretize the ZYZ-Euler angles. First, note that, due the symmetries in the model, it is enough to restrict the Euler angles to the following domains:

$$0 \leq \alpha \leq \pi, \qquad 0 \leq \beta \leq \frac{\pi}{2}, \qquad 0 \leq \gamma \leq \pi. \tag{21}$$

To prove this, recall that our model is invariant under a rotation by π about any line passing through the origin. Using the ZYZ-convention, the rotation $R_{\boldsymbol{\alpha}}$ is defined by:

$$R_{\boldsymbol{\alpha}} = R_Z(\gamma)R_Y(\beta)R_Z(\alpha), \tag{22}$$

where $R_Z(\alpha)$ and $R_Y(\alpha)$ denote rotation by α about the Z-axis and Y-axis respectively,

$$R_Z(\alpha) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad R_Y(\alpha) = \begin{bmatrix} \cos(\alpha) & 0 & -\sin(\alpha) \\ 0 & 1 & 0 \\ \sin(\alpha) & 0 & \cos(\alpha) \end{bmatrix}.$$

Now, suppose $\alpha = a + \pi$, where $0 < a < \pi$ and $0 \leq \beta \leq \frac{\pi}{2}$, $\quad 0 \leq \gamma \leq \pi$, then the corresponding rotation, $R_{\boldsymbol{\alpha}}$, is given by

$$R_{\boldsymbol{\alpha}} = R_{\boldsymbol{\alpha}_1}R_Z(\pi),$$

where $\boldsymbol{\alpha}_1 = (a, \beta, \gamma)$. Since, our model is invariant under rotation by $R_Z(\pi)$, we infer that it is enough to restrict α to $[0, \pi]$. Similar calculations show that γ can be restricted to $[0, \pi]$ as well. To see that it is enough to restrict β between $\left[0, \frac{\pi}{2}\right]$, consider $\boldsymbol{\alpha} = (\alpha, a + \pi/2, \gamma)$, where $0 < a < \frac{\pi}{2}$. Then, we have,

$$R_{\boldsymbol{\alpha}} = R_Z(\gamma) R_Y \left(a + \frac{\pi}{2}\right) R_Z(\alpha).$$

Again, because of the symmetry properties of our texture, $R_Y\left(a + \frac{\pi}{2}\right)$ can be replaced by $R_Y\left(a + \frac{3\pi}{2}\right)$, because this corresponds to a rotation by π about the new Y-axis after the texture has been rotated by $R_Z(\alpha)$. It is easy to check that

$$R_Y\left(a + \frac{3\pi}{2}\right) = R_Z(\pi) R_Y\left(\frac{\pi}{2} - a\right) R_Z(\pi).$$

Since, $\frac{\pi}{2} - a$ lies in $\left[0, \frac{\pi}{2}\right]$, we conclude that it is enough to restrict β in $\left[0, \frac{\pi}{2}\right]$.

Now, we are ready to discretize the integral in (15). We take points spaced uniformly with respect to the Haar measure defined in Equation (16). This is done by taking equally spaced points on the interval $[0, \pi]$ for both α and γ. Hence, the discrete sets of parameters are given by $\alpha_i = \{\frac{i\pi}{N_\alpha}\}$ for $i = 0, 1, \ldots, N_\alpha - 1$, and $\gamma_i = \{\frac{i\pi}{N_\gamma}\}$ for $i = 0, 1, \ldots, N_\gamma - 1$. For β, we take the discrete set $\beta_i = \{\arccos(1 - \frac{i+0.5}{N_\beta})\}$, for $i = 0, 1, \ldots, \mathbb{N}_\beta - 1$. Notice that we take the discrete values of β starting with 0.5, this is done to avoid β_0 from being equal to zero. If $\beta_0 = 0$, we get into a situation referred to as *gimbal lock*. In that case, only $\alpha + \gamma$ is uniquely determined. Hence, we offset by 0.5 to avoid duplicating rotations in our discrete set. For more information, and a nice illustration of why this situation is called gimbal lock, visit the following web-page.
`http://en.wikipedia.org/wiki/Euler_angles`

Using the above discrete set of rotations, a practically implementable version of the distance defined in Equation (15) is given by

$$\min_{\boldsymbol{\alpha}_0 \in SO(3)} \frac{1}{N_\alpha N_\beta N_\gamma} \sum_{i=0}^{N_\alpha - 1} \sum_{j=0}^{N_\beta - 1} \sum_{k=0}^{N_\gamma - 1} \mathrm{KLdist}(\Gamma(\boldsymbol{\alpha}_{(i,j,k)}), \Gamma(\boldsymbol{\alpha}_{(i,j,k)} \boldsymbol{\alpha}_0)), \tag{23}$$

where $\boldsymbol{\alpha}_{i,j,k} = (\alpha_i, \beta_j, \gamma_k)$ and as before, the product $\boldsymbol{\alpha}_{i,j,k}\boldsymbol{\alpha}_0$ represents the Euler angles corresponding the rotation operator $\mathcal{R}_{\boldsymbol{\alpha}_{i,j,k}} \mathcal{R}_{\boldsymbol{\alpha}_0}$. For actual experiments, the minimization can be carried out on a subset of $SO(3)$ corresponding to the Euler angles described by (21) because of the symmetries present in the model.

5 Experimental Results

To test the ideas developed so far, we generate synthetic volumes arising from 3-D GMRF models which we synthesize using the algorithm described in Section 3.4. We use the MATLAB function `patternsearch` to solve the minimization problem which is required to calculate the distance defined in Equation (23). The number of discrete angles are taken to be $N_\alpha = N_\beta = N_\gamma = 5$. This gives a total of 125 rotations sampled uniformly with respect to the Haar measure on $SO(3)$. Thus, each evaluation of the function to be minimized in

Equation (23) requires the calculation of 125 sets of parameters and the calculation of 125 K-L distances. On a 2.8 GHz machine each evaluation of the function takes about three-quarters of a second. The MATLAB routine mentioned above is particularly useful because it does not calculate derivatives and thus, each iteration of the algorithm requires very few evaluations of the function to be minimized. The details of this routine can be found in the MATLAB documentation on the web-page of Mathworks. We do not intend to study this optimization algorithm here. We just use it as a black-box and it works well for the calculation of the rotationally invariant distance. It takes about one minute to calculate the distance between two texture signatures.

For our first set of experiments, we generate two distinct synthetic textures, denoted by $\mathcal{T}_1$ and $\mathcal{T}_2$, that have the same conditional variance, $\sigma^2 = 1$, and mean, $\mu = 0$. We use the first order neighborhood to generate these textures using the parameters $\boldsymbol{\theta} = (\theta_x, \theta_y, \theta_z)$ shown in Table 1. We make this choice to test the real potential of this method. If we generate synthetic textures with neighborhoods larger than those of order one, then we do not have control over the conditional variance of the GMRF of order one fitted to those textures. In fact, as we show in another set of experiments described below, a few textures that we generated with higher order neighborhoods could be discriminated based on the value of σ^2 corresponding to the order one GMRF fitted to these textures. If the estimate for σ^2 itself is enough to discriminate between textures then we do not have to look at $\boldsymbol{\theta}$. Hence, for our first set of experiments we stick with the synthetic textures obtained from order one GMRFs with equal conditional variance. We denote $\mathcal{T}_1$ by $\mathcal{T}_{1,0}$ while $\mathcal{T}_{1,\frac{\pi}{2}}$ denotes the texture $\mathcal{T}_{1,0}$ rotated about the X-axis by $\frac{\pi}{2}$. Similarly, $\mathcal{T}_{2,0}$ denotes the texture $\mathcal{T}_2$ while $\mathcal{T}_{2,\frac{\pi}{2}}$ denotes the texture $\mathcal{T}_{2,0}$ rotated about the Y-axis by $\frac{\pi}{2}$. Thus, we expect the distances between $\mathcal{T}_{1,0}$ and $\mathcal{T}_{1,\frac{\pi}{2}}$, and between $\mathcal{T}_{2,0}$ and $\mathcal{T}_{2,\frac{\pi}{2}}$ to be small, while we want both $\mathcal{T}_{1,0}$ and $\mathcal{T}_{1,\frac{\pi}{2}}$ to have a large distance from both $\mathcal{T}_{2,0}$ and $\mathcal{T}_{2,\frac{\pi}{2}}$.

	$\mathcal{T}_{1,0}$	$\mathcal{T}_{1,\frac{\pi}{2}}$	$\mathcal{T}_{2,0}$	$\mathcal{T}_{2,\frac{\pi}{2}}$
θ_x	0.1	0.1	0.05	0.20
θ_y	0.1	0.25	0.15	0.15
θ_z	0.25	0.1	0.20	0.05
σ^2	1.0	1.0	1.0	1.0

Table 1: Parameters for synthetic textures used for experimental study.

In Tables 2, 3 and 4, we tabulate the distance of a realization of each of the four textures (two rotations of two distinct textures) given in Table 1 from another realization of each of these textures, for different upsampling factors. The diagonal entries in the tables correspond to the distance between two realizations of the same texture while the off-diagonal entries correspond to the distance between realizations of two different textures. We notice that in Tables 2 and 3, the distance between $\mathcal{T}_{1,0}$ and $\mathcal{T}_{1,\frac{\pi}{2}}$ is of the order of the distance between two realizations of $\mathcal{T}_{1,0}$ or $\mathcal{T}_{1,\frac{\pi}{2}}$. Same is true for $\mathcal{T}_{2,0}$ and $\mathcal{T}_{2,\frac{\pi}{2}}$. We further observe that *the distance between any realization of $\mathcal{T}_1$ and any realization $\mathcal{T}_2$ is an order of magnitude higher than the distance between two realizations of either $\mathcal{T}_1$ or $\mathcal{T}_2$, or their rotated versions.* Thus, we see that the distance is able to discriminate textures up to rotations.

The results in Table 2 are obtained by resampling the auto covariance on the grid $\frac{\mathbb{Z}^3}{4}$ while the results in Table 3 are obtained using the auto covariance resampled on $\frac{\mathbb{Z}^3}{2}$. The finer grid does not give any significant improvement. Hence, in this case, resampling on $\frac{\mathbb{Z}^3}{2}$ is enough. Table 4 shows the results obtained with the autocovariance defined on the original grid $\mathbb{Z}^3$. Here $\mathcal{T}_{1,0}$ and $\mathcal{T}_{1,\frac{\pi}{2}}$ are at a much larger distance from each other than two realizations of $\mathcal{T}_{1,0}$ or $\mathcal{T}_{1,\frac{\pi}{2}}$. Same is true for $\mathcal{T}_{2,0}$ and $\mathcal{T}_{2,\frac{\pi}{2}}$ as well. Since we have not resampled the autocovariance function on a finer grid in this case, we end up rotating by a simple linear interpolation. Hence, we conclude that rotation by simple linear interpolation is not enough. The data must be resampled on a grid that is fine enough to keep the error due to linear interpolation under control.

Finally, observe that the Euler angles corresponding to the minimizer for each case are listed below each distance. They have been normalized by π to make it easier to read. Hence, in each case we list the angles $(\alpha^*/\pi, \beta^*/\pi, \gamma^*/\pi)$, where $(\alpha^*, \beta^*, \gamma^*)$ is the minimizer for the optimization problem required to calculate the distance. This minimizer corresponds to the rotation that must be applied on one texture to get the other in case one is a rotated version of the other. As we can observe in Tables 2 and 3, , all these minimizers are close to the expected values modulo the symmetries assumed, i.e. each can be off by $\pm\pi$.

	$\mathcal{T}_{1,0}$	$\mathcal{T}_{1,\frac{\pi}{2}}$	$\mathcal{T}_{2,0}$	$\mathcal{T}_{2,\frac{\pi}{2}}$
$\mathcal{T}_{1,0}$	0.0007 (0.5871,0,0.3333)	0.0005 (0.4925,0.4515,0.5274)	0.0072 (0.1592,0.0348,0)	0.0137 (0.0199,0.4913)
$\mathcal{T}_{1,\frac{\pi}{2}}$	0.0010 (0.6070,0.4913,0.5274)	0.0007 (0,0,0)	0.0101 (0.9103,0.4714,0.5274)	0.0182 (0.5025,0,0)
$\mathcal{T}_{2,0}$	0.0123 (0.7512,0.0050,0)	0.0128 (0.4925,0.4814,0.9947)	0.0006 (0.0050,0.0298,0)	0.0004 (0.9900,0.4963,0)
$\mathcal{T}_{2,\frac{\pi}{2}}$	0.0093 (0.7263,0.4913,0.0398)	0.0101 (0,0.0039,0.5025)	0.0012 (0.0099,0.4963,0)	0.0009 (0.0050,0,0)

Table 2: Distances between two rotations of two distinct textures using the rotationally invariant distance defined in Equation (23) and autocovariance resampled on $\frac{\mathbb{Z}^3}{4}$. Below each distance we list the angles $(\alpha^*/\pi, \beta^*/\pi, \gamma^*/\pi)$, where $(\alpha^*, \beta^*, \gamma^*)$ is the minimizer.

	$\mathcal{T}_{1,0}$	$\mathcal{T}_{1,\frac{\pi}{2}}$	$\mathcal{T}_{2,0}$	$\mathcal{T}_{2,\frac{\pi}{2}}$
$\mathcal{T}_{1,0}$	0.0006 (0.6716,0,0.3333)	0.0006 (0.4974,0.4814,0.5075)	0.0073 (0.6319,0,0.3333)	0.0136 (0,0.4913,0)
$\mathcal{T}_{1,\frac{\pi}{2}}$	0.0013 (0.4825,0.4913,0.0050)	0.0007 (0,0.0050,0)	0.0100 (0.4975,0.4814,0.0050)	0.0164 (0.8507,0,0.6667)
$\mathcal{T}_{2,0}$	0.0125 (0.8208,0,0.6667)	0.0203 (0.0099,0,0)	0.0010 (0,0,0)	0.0004 (0.0050,0.4963,0)
$\mathcal{T}_{2,\frac{\pi}{2}}$	0.0119 (0,0.4863,0.0249)	0.0082 (0.5024,0,0)	0.0007 (0,0.4814,0.0050)	0.0008 (0,0.0050,0)

Table 3: Distances between two rotations of two distinct textures using the rotationally invariant distance defined in Equation (23) and autocovariance resampled on $\frac{\mathbb{Z}^3}{2}$. Below each distance we list the angles $(\alpha^*/\pi, \beta^*/\pi, \gamma^*/\pi)$, where $(\alpha^*, \beta^*, \gamma^*)$ is the minimizer.

	$\mathcal{T}_{1,0}$	$\mathcal{T}_{1,\frac{\pi}{2}}$	$\mathcal{T}_{2,0}$	$\mathcal{T}_{2,\frac{\pi}{2}}$
$\mathcal{T}_{1,0}$	0.0026 (0,0,0)	0.0812 (0.0050,0.0050,0)	0.0330 (0.0099,0,0)	0.1750 (0.0050,0,0)
$\mathcal{T}_{1,\frac{\pi}{2}}$	0.1118 (0,0,0)	0.0010 (0,0,0)	0.0852 (0.0099,0,0)	0.0562 (0,0,0)
$\mathcal{T}_{2,0}$	0.0454 (0,0.0050,0)	0.0694 (0.0050,0.0050,0)	0.0016 (0,0,0)	0.0108 (0,0.4963,0)
$\mathcal{T}_{2,\frac{\pi}{2}}$	0.0607 (0,0.4963,0)	0.0473 (0,0,0)	0.0246 (0,0.4963,0)	0.0018 (0,0,0)

Table 4: Distances between two rotations of two distinct textures using the rotationally invariant distance defined in Equation (23) and autocovariance sampled on the original grid $\mathbb{Z}^3$. Below each distance we list the angles $(\alpha^*/\pi, \beta^*/\pi, \gamma^*/\pi)$, where $(\alpha^*, \beta^*, \gamma^*)$ is the minimizer.

For the experiments described above we only took two distinct textures to test the rotational invariance of the our distance function. Next, we study five distinct textures, two of which are the ones presented above ($\mathcal{T}_1$ and $\mathcal{T}_2$) while the other three, denoted by $\mathcal{T}_3, \mathcal{T}_4$ and $\mathcal{T}_5$, are produced using GMRF models with higher order neighborhoods. Now, since we still use order one neighborhood to get the texture signatures, the estimate for σ^2 may not be the same as, or even close to, the σ^2 used to produce these textures. In fact, for the examples we use here, even though we used the same σ^2 (but different $\boldsymbol{\theta}$s) to produce $\mathcal{T}_3, \mathcal{T}_4$ and $\mathcal{T}_5$, the value of σ^2 estimated for a order one GMRF fitted to each of these three textures is different. Thus, in this case, the estimate of σ^2 might be enough to discriminate textures. It is interesting to note that this difference in the value of σ^2 is not an impediment for our distance function. This is evident from the results tabulated in Table 5. For each of the five textures, we see that the distance between two of its realizations is still relatively much smaller than its distance from other textures. Thus, we see that we can even discriminate textures arising from higher order GMRFs using our scheme. This shows that this classification scheme is not limited to synthetic textures produced using GMRFs with order one neighborhoods, and may also be applied to a larger class of textures.

	$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_3$	$\mathcal{T}_4$	$\mathcal{T}_5$
$\mathcal{T}_1$	0.0006	0.0073	0.4232	2.3180	1.7724
$\mathcal{T}_2$	0.0125	0.0010	0.4894	2.5227	1.8381
$\mathcal{T}_3$	0.4466	0.5134	0.0004	0.5208	0.4563
$\mathcal{T}_4$	2.4314	2.6315	0.5605	0.0021	0.3533
$\mathcal{T}_5$	1.8200	1.9227	0.4318	0.2540	0.0043

Table 5: Distances between five distinct textures using the rotationally invariant distance defined in Equation (23) and autocovariance resampled on the grid $\frac{\mathbb{Z}^3}{2}$.

6 Conclusion and Future Work

In this chapter we have presented a novel approach to rotationally invariant 3-D texture classification. We define a rotationally invariant distance on GMRF-based texture signatures. Rotation of a texture is achieved via rotation of the autocovariance function corresponding to the texture. The practical implementation of the rotationally invariant distance is shown to work well on experimental data. Moving forward, we want to test if these GMRF-based texture signatures, along with the rotationally invariant distance, can be used to separate natural 3-D textures arising, for example, in medical imaging. We have used a 2-D version of this scheme to separate natural textures (taken from the Brodatz library for instance) that are not generated using a GMRF. The positive results in 2-D together with the fact that we could discriminate textures obtained from higher order GMRFs in 3-D, makes our scheme a promising candidate for rotationally invariant classification of natural textures in 3-D.

References

[1] R. Azencott and D. Dacunha-Castelle. *Series of Irregular Observations.* Springer-Verlag, 1986.

[2] R. Azencott, J.P. Wang, and L. Younes. Texture classification using Gabor filters. *IEEE Trans. Pattern Anal. Machine Intell.*, 19(2):148–152, 1997.

[3] T. Chang and C. Kuo. Texture analysis and classification with tree-structured wavelet transform. *IEEE Transaction on Image Processing*, 2(4):429–441, 1995.

[4] R. Chellappa. Two-dimensional discrete Gaussian Markov random field models for image processing. *Progress in Pattern Recognition 2*, pages 79–112, 1985.

[5] R. Chin and C. Harlow. Automated visual inspection: A survey. *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-4(6):557–573, 1982.

[6] G.S. Chirikjian and A. B. Kyatkin. *Engineering Applications of Noncommutative Harmonic Analysis: With Emphasis on Rotation and Motion Groups.* CRC Press, 2000. electronic resource.

[7] F.S. Cohen, Z. Fan, and M.A. Patel. Classification of rotated and scaled textured images using Gaussian Markov Random Field models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(2):192–202, 1991.

[8] K. Etemad and R. Chellappa. Separability based tree-structured local basis selection for texture classification. In *ICIP 1994*, volume 3, Austin, TX, 1994.

[9] J. Fehr and H. Burkhardt. 3d rotation invariant local binary patterns. In *ICPR 2008*, Tampa, FL, 2008.

[10] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Anal. Machine Intell.*, 6:721–741, 1984.

[11] X. Guyon. *Random Fields on a Network.* Probability and its Applications. Springer Verlag, 1995.

[12] R. Haralick. Statistical and structural approaches to texture. *Proceedings of IEEE*, 67(5):610–621, 1979.

[13] R.L. Kashyap. Analysis and synthesis of image patterns by spatial interaction models. *Progress in Pattern Recognition*, pages 149–186, 1981.

[14] R.L. Kashyap and R. Chellappa. Estimation and choice of neighbors in spatial-interaction models of images. *IEEE Transactions on Information Theory*, 1:60–72, 1983.

[15] R.L. Kashyap and A. Khotanazad. A model-based method for rotation invariant texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(4):472–481, 1986.

[16] V.A. Kovalev, F. Kruggel, H.J. Gertz, and D.Y. von Cramon. Three-dimensional texture analysis of MRI brain datasets. *IEEE Trans.on Medical Imaging*, 20(5):424–433, 2001.

[17] A.S. Kurani, D.H. Xu, J.D. Furst, and D.S. Raicu. Co-occurance matrices for volumetric data. In *7th IASTED International Conference on Computer Graphics and Imaging*, 2004.

[18] S. Lakshmanan and H. Derin. Valid parameter space for 2-D Gaussian Markov Random Fields. *IEEE Transactions on Information Theory*, 39(2):703–709, 1993.

[19] S.Y. Lu and K.S. Fu. A syntactic approach to texture analysis. *Computer Graphics and Image Processing*, 7:303–330, 1978.

[20] A. Madabhushi, M. Feldman, D. Metaxas, D. Chute, and J. Tomaszewski. A novel stochastic combination of 3d texture features for automated segmentation of prostatic adenocarcinoma from high resolution MRI. *Medical Image Computing and Computer-Assisted Intervention*, 2878:581–591, 2003.

[21] J. Mao and A.K. Jain. Texture classification and segmentation using multiresolution simultaneous autoregressive models. *Pattern Recognition*, 25(2):173–188, 1992.

[22] M. Papadakis, B.G. Bodmann, S.K. Alexander, D. Vela MD, S. Baid, A.A. Gittens, D.J. Kouri, S.D. Gertz MD, S. Jain, J.R. Romero, X. Li, P. Cherukuri, D.D. Cody, G. W. Gladish MD, I. Aboshady MD, J.L. Conyers, and S.W. Casscells. Texture-based tissue characterization for high-resolution CT-scans of coronary arteries. *Comm. in Numer. Methods in Engineering*, 25:597–613, 2009.

[23] K. R. Parthasarathy. *Probability Measures on Metric Spaces.* Probability and Mathematical Statistics. Academic Press, 1967.

[24] R. Porter and N. Canagarajah. Robust rotation-invariant texture classification: wavelet, Gabor filter and GMRF based schemes. *IEE Proc.- Vis. Image Signal Process.*, 144(3):180–188, 1997.

[25] Elena Boykova Ranguelova. Segmentation of textured images on three-dimensional lattices. *PhD Thesis, University of Dublin, Trinity College*, 2002.

[26] C.C. Reyes-Aldasoro and A. Bhalerao. Volumetric feature selection for MRI. *Information Processing in Medical Imaging*, 2732:282–293, 2003.

[27] J.R. Romero, S. Alexander, S. Baid, S. Jain, and M. Papadakis. The geometry and the analytic properties of Isotropic Multiresolution Analysis. *Advances in Comp. Math.*, 31:283–328, 2009.

[28] A. Teuner, O. Pichler, and B.J. Hosticka. Unsupervised texture segmentation of images using tuned Gabor filters. *IEEE Transaction on Image Processing*, 4(6):863–870, 1995.

[29] F. Tomita and S. Tsuji. *Computer Analysis of Visual Textures.* Kluwer Academic Publishers, 1990.

[30] M. Unser. Texture classification and segmentation using wavelet frames. *IEEE Transaction on Image Processing*, 2(4):429–441, 1995.

[31] D.H. Xu, A.S. Kurani, J.D. Furst, and D.S. Raicu. Run-length encoding for volumetric texture. In *4th IASTED International Conference on Visualization, Imaging and Image Processing*, 2004.