UNICAMP

UNIVERSIDADE ESTADUAL DE CAMPINAS

Instituto de Matemática, Estatística e Computação Científica

VICTOR FREGUGLIA SOUZA

# Modeling Textile Images with Markov Random Fields

# Modelagem de Imagens Têxteis com Campos Aleatórios Markovianos

Campinas

2018

Victor Freguglia Souza

# Modeling Textile Images with Markov Random Fields

# Modelagem de Imagens Têxteis com Campos Aleatórios Markovianos

Dissertação apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre em Estatística.

Dissertation presented to the Institute of Mathematics, Statistics and Scientific Computing of the University of Campinas in partial fulfillment of the requirements for the degree of Master in Statistics.

Supervisor: Nancy Lopes Garcia

Este exemplar corresponde à versão final da Dissertação defendida pelo aluno Victor Freguglia Souza e orientada pela Profa. Dra. Nancy Lopes Garcia.

Campinas

2018

**Agência(s) de fomento e nº(s) de processo(s):** CAPES

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

So89m

Souza, Victor Freguglia, 1994-
Modeling textile images with Markov random fields / Victor Freguglia Souza. – Campinas, SP : [s.n.], 2018.

Orientador: Nancy Lopes Garcia.
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

1. Markov, Campos aleatórios de. 2. Markov, Modelos escondidos de. 3. Fourier, Transformadas de. 4. Imagens digitais. I. Garcia, Nancy Lopes, 1964-. II. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. III. Título.

Informações para Biblioteca Digital

**Título em outro idioma:** Modelagem de imagens têxteis com campos aleatórios markovianos
**Palavras-chave em inglês:**
Markov random fields
Hidden Markov models
Fourier transformations
Digital images
**Área de concentração:** Estatística
**Titulação:** Mestre em Estatística
**Banca examinadora:**
Nancy Lopes Garcia [Orientador]
Mariana Rodrigues Motta
Florencia Graciela Leonardi
**Data de defesa:** 19-02-2018
**Programa de Pós-Graduação:** Estatística

**Dissertação de Mestrado defendida em 19 de fevereiro de 2018 e aprovada pela banca examinadora composta pelos Profs. Drs.**

**Prof(a). Dr(a). NANCY LOPES GARCIA**

**Prof(a). Dr(a). MARIANA RODRIGUES MOTTA**

**Prof(a). Dr(a). FLORENCIA GRACIELA LEONARDI**

As respectivas assinaturas dos membros encontram-se na Ata de defesa

# Acknowledgements

I am very grateful to my advisor, Professor Nancy Lopes Garcia, for her patience and continuous support along this work, as well as the great opportunity to learn and improve.

I also thank my family and friends for all the encouragement and aid over these years.

Finally, I acknowledge the support of professors and staff from IMECC-UNICAMP, as well as financial support from CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior).

# Resumo

Quando novas tecnologias para tingimento de tecidos são desenvolvidas, avaliar a qualidade dessas técnicas envolve a medição da homogeneidade de cores resultante através de imagens. A presença da textura do tecido cria uma estrutura de dependência sofisticada para as cores dos pixels. A transformada de Fourier é utilizada juntamente com técnicas de regularização para separar o efeito da textura das imagens. As imagens são então modeladas por campos aleatórios de Markov ocultos e bases de Fourier como covariáveis possibilitando, assim, uma avaliação da homogeneidade de cores baseada em entropia, utilizando apenas a parte correspondente ao tingimento.

**Palavras-chave**: Campos aleatórios de Markov, campos aleatórios ocultos, transformada de Fourier, seleção de variáveis, regularização.

# Abstract

When new textile dyeing technologies are developed, evaluating the quality of these techniques involves measuring the resulting color homogeneity using digital images. The presence of a texture effect caused by the fabric creates a sophisticated dependence structure in pixels coloring. Fourier transform is used with regularization techniques to remove the texture signal from the image. Images are then modeled as a hidden Markov random field and Fourier basis as covariates, allowing an entropy-based evaluation of color homogeneity using the filtered signal that corresponds to the dyeing process only.

**Keywords**: Markov random fields, hidden random fields, Fourier transform, variable selection, regularization.

# List of Figures

# List of Tables

# List of abbreviations and acronyms

| | |
|---|---|
| MRF | Markov Random Field |
| HMRF | Hidden Markov Random Field |
| MLE | Maximum Likelihood Estimate |
| MPLE | Maximum Pseudo-Likelihood Estimate |
| PLS | Penalized Least-Squares |
| OLS | Ordinary Least-Squares |
| EM | Expectation-Maximization |
| MAP | Maximum a Posteriori |

# List of symbols

| | |
|---|---|
| $\delta(a, b)$ | Kronecker delta, equals 1 if $a = b$ and 0 otherwise. |
| $\mathbb{1}_E$ | Indicator function, equals 1 if $E$ is true, 0 otherwise. |
| $\|\cdot\|_2$ | 2-norm for vectors |

# Contents

# Introduction

Most of the colorants currently used for textile dyeing are synthetic, originated from non-renewable sources. These products are usually preferred due to technical and economic advantages, but they are also associated to environmental and toxic issues. Therefore, more sustainable processes have been a great demand in our society, see Alihosseini et al. (2008); Santis et al. (2005). In this sense, the interest for natural colorants for textile dyeing has increased in recent years. However, the regular natural dyes (from plant and animal sources) yields only a few grams per kg of dry material, increasing their prices (c.a. US$ 1/g) and limiting their applications, see Velmurugan et al. (2010). In order to overcome this problem, many researchers have investigated the use of microbial pigments as an alternative for textile dyeing, for example, Nagia and El-Mohamedy (2007) and Kasiri and Safapour (2014), and the number of publications in this field is increasing in the last years.

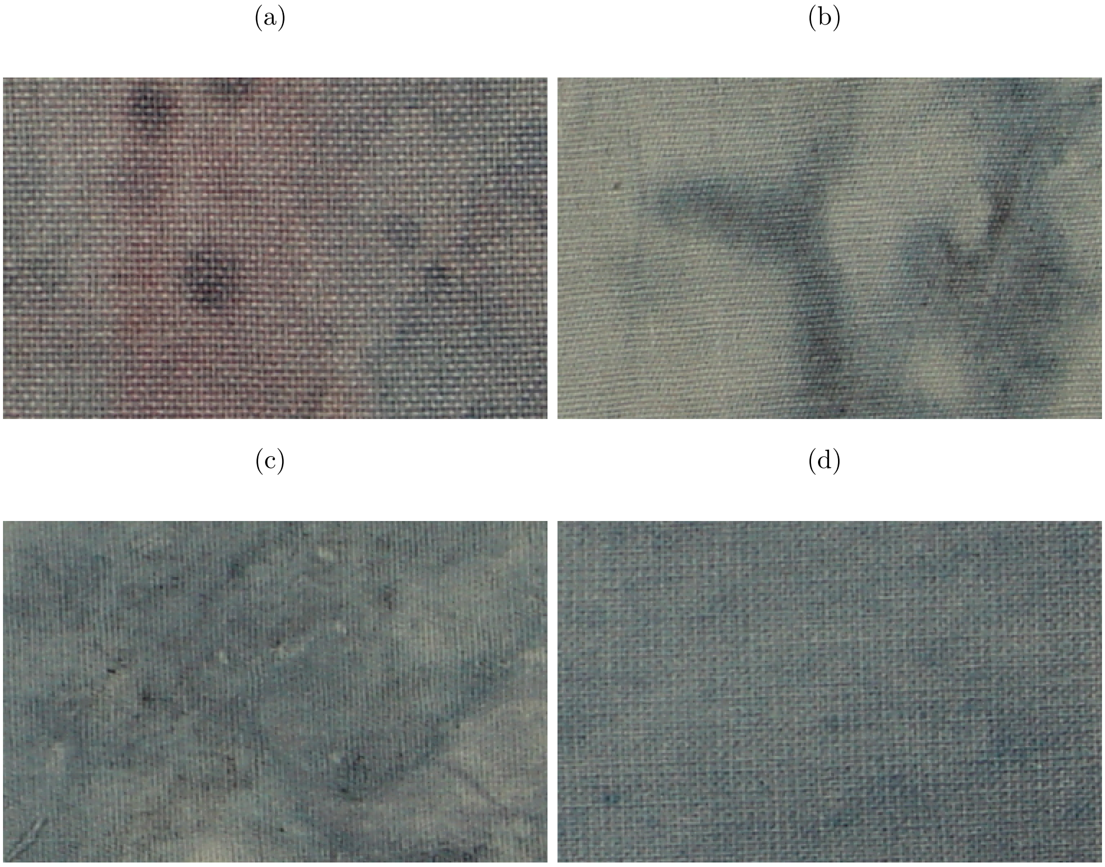
Although there are still few articles dealing with the use of microbial pigments for dyeing textile materials, the reports demonstrate that fungi, especially molds, have great potential for this purpose. In this context, based on the observation that filamentous fungi grow in wet tissues (e.g. black spots in bath towel), causing extremely persistent stains, Prof. Juliano Lemos Bicas from Faculdade de Engenharia de Alimentos/Unicamp and his team idealized a textile “biodyeing” procedure consisting of growing a colored filamentous fungi directly into the fabric to make a “controlled stain” in such material (Patent Application – BR 10 2013 027036 9). In principle, they expect that this “biodyeing” strategy can be applicable for other pigment-producing molds, such as Monascus sp.

One of the greatest challenges in such approach involves color homogeneity, which is difficult to achieve due to the heterogeneity of fungal growth. Thus, a method for measuring color distribution and color homogeneity in textiles has emerged as an essential tool to evaluate this process. But when regular image analysis software (e.g. ImageJ) is used, we observed that the inclination of the fibers and the thickness of wart and weft had a great (negative) impact on the image analysis. Consequently, a new method for image analysis would have to be developed in order to ignore the fibers during image treatment.

# 1 Materials and Methods

## 1.1 Dataset

The data used in this work consist of four digital images of 378 × 237 pixels in the Portable Network Graphics (.png) format provided by Prof. Juliano Lemos Bicas from Faculdade de Engenharia de Alimentos/Unicamp, shown in Figure 1. Originally, the data for each pixel consisted of three integer values, representing red, green and blue channels, ranging from 0 to 255 each.



(a)

(b)

(c)

(d)

Figure 1 – Original digital images.

These images were transformed to gray scale by using the *imager* package from R, which takes a specific linear combination of the 3-channel color values that corresponds to a lightness index, ranging from 0 to 1. Despite this index being discrete, the space of possible values is very dense, making it a nearly continuous variable. The transformed images are shown in Figure 2.

Mathematically, the gray scale data corresponds to matrices $Y$, with

$$0 \leqslant Y(x,y) \leqslant 1, (x,y) \in \{0,\ldots,377\} \times \{0,\ldots,236\}. \tag{1.1}$$

Note that when representing these matrices, the gray-level in each pixel is relative to the range of values present in that specific matrix, the highest value (which may not be 1) representing the white color and the lowest (not necessarily 0) black color. That is, for any real $b$ and positive $a$, $aY + b$ have the same visual representation as $Y$.

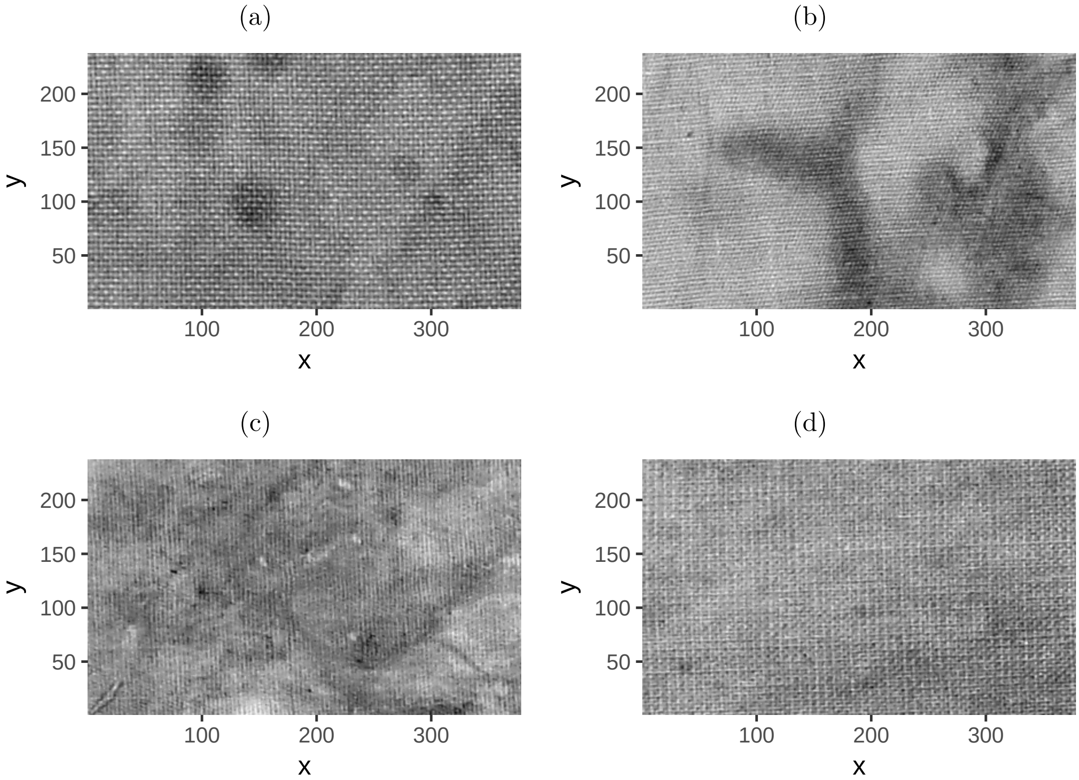


Figure 2 – Gray-scale transformed images.

## 1.2 Entropy-based Measure of Color Homogeneity

The concept of color homogeneity is vague, with no rigorous definition available and, therefore, defining measures of color homogeneity is not a trivial task. Alemaskin, Manas-Zloczower and Kaufman (2005) defines a measure of color homogeneity for both 3-channel colored and gray scale images based on entropy. Particularly for gray-scale images, it is assumed that on a very high resolution, images consist of only independent "black" (0) and "white"(1) particles uniformly spread along the image region. Observed gray-scale value (in a $[0, 1]$ scale) of a particular pixel is an average of the values from all the particles in a corresponding region, which is called a bin.

Under these assumptions, considering bins corresponding to individual pixels $j = (x, y)$, $Y(x, y)$, the average number of "white" particles in bin $j$, is an estimate for the conditional probability of a particle being "white" given the $j$-th bin, denoted $p_j$. With $J$ bins, the color homogeneity index of an observed image $\mathbf{Y}$ defined by the authors is

$$C_{\mathbf{Y}}^{J} = -\frac{\sum_{j=1}^{J} p_j \log(p_j) + (1-p_j)\log(1-p_j)}{J\log(2)}, \tag{1.2}$$

which varies from 0 to 1, and only assumes the maximum value if every pixel in the image has the same value of 0.5. We assume none of the bins has all its particles of the same color, i.e., $0 < p_j < 1$.

The index defined in Equation 1.2 is constructed based on the concept of entropy. While entropy itself measures how heterogeneous a probability measure is, the homogeneity index $C_{\mathbf{Y}}^{J}$ gets larger as the estimated probability for each bin gets closer to 0.5. This means that, while having all probabilities close to 0.5 indicates a very heterogeneous distribution for particles within a bin, the distributions are homogeneous between bins.

It is important to note that, when defining color homogeneity for textile dyeing, the measure should be shift invariant, *i.e.* adding the same constant value to each pixel should result in similar homogeneity measure, which is not the case for (1.2), as for example, an image with all pixels assuming the value 0.8 has a smaller index than a constant image with value 0.5. The authors define a transformation to the data to accommodate what they call an "ideal gray-level" $g^*$ $(0 < g^* < 1)$ that allows the index to achieve large values when bin colors are concentrated around $g^*$ rather than 0.5. This is done by substituting the estimated conditional probability for each bin $p_j$ from (1.2) by

$$p_j^* = \frac{\frac{p_j}{g^*}}{\frac{p_j}{g^*} + \frac{1-p_j}{1-g^*}}. \tag{1.3}$$

We use this transformation with $g^* = \bar{\mathbf{Y}}$(the average color of the image) to obtain a "fair" homogeneity measure, in the sense that large values for the index will be achieved when colors for each bin are concentrated around $\bar{\mathbf{Y}}$, rather than 0.5.

Another important aspect of $C_{\mathbf{Y}}^{J}$ is the number of bins $J$ and its relation with the image resolution. Despite the impossibility of observing a number of bins greater than the number of pixels in our original image, we can group multiple pixels in the same bin and treat them as a single pixel, obtaining a lower resolution image, by averaging gray-levels in each bin i.e. we compute $p_j$ as the average of all particles in the (larger than a pixel) bin, and since the particles are assumed equally distributed on the image region, the estimate is given by the average of grouped values. Particularly, we will consider bins formed by square regions of pixels ($2 \times 2$, $3 \times 3$, and so on) and when a dimension of the original image is not a multiple of these increasing factors, we just group the remaining pixels, causing the borders bins to be possibly a rectangular region.

Increasing the size of bins causes a reduction on the image resolution, which results in a loss of details, like the visual effect of the weft in textile images. Because the color variation that comes from the texture fades as the resolution is reduced, color homogeneity index naturally increases for larger bins sizes. Figure 3 shows values of $C^J_{\mathbf{Y}}$ for images of fungal dyed fabric from Figure 2 and an additional image of a synthetic colorant dyed textile, with bin sizes of 1 × 1 up to 20 × 20.

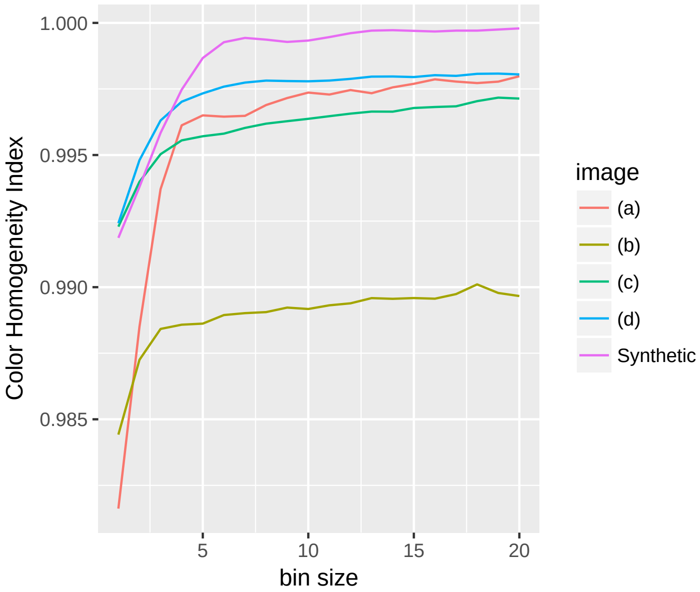


Figure 3 – $C^J_{\mathbf{Y}}$ computed for five different images and 20 bin sizes.

The same trend can be observed for every analyzed image, where there is a large increase in the color homogeneity for the first few bin sizes until the details are lost, then, we see an almost flat relation between bin size and the index. While the general trend is the same in each image, relations between resolution and color homogeneity depends directly on characteristics of "background" textures, as the bin size increase necessary to remove the details varies. To exemplify this effect, the synthetic dye image has practically the same color homogeneity index as images (c) and (d) when looking at the maximum resolution, but as the bin sizes increase, the homogeneity difference can be clearly perceived. Image (a) has the lowest index for high resolution, while it ranks third (very close to second) at lower resolutions.

The presence of texture hinders the usage of $C^J_{\mathbf{Y}}$, as using the highest resolution introduces an effect that should not be accounted to the dyeing process, even causing a change in index rankings when compared to lower resolutions and as this effect is not the same for every image (different weft structures), appropriate bin sizes may not be the same for every case. Due to these facts, before computing the color homogeneity index proposed, we need to make sure we can remove the texture effect and take into account only the color variation that comes from the dyeing process itself.

While this is not an optimal measure of color homogeneity because it depends only on pixel colors regardless of their position, this index successfully illustrates the additional challenges for measuring the concept color homogeneity in textile images brought by the weft structure and we won't extend the discussion on color homogeneity definitions and possible measures in this work.

## 1.3 Objectives and Methodology

With the negative effect texture causes when measuring color homogeneity in mind, our objective is to use models that are capable of identifying and isolating the signal corresponding to dyeing, obtaining a procedure that can evaluate color homogeneity resulting from the dyeing process exclusively. To achieve this, we consider that the observed images can be decomposed into three different components

$$\mathbf{Y} = \mathbf{S} + \mathbf{T} + \epsilon, \tag{1.4}$$

with **S** being a fixed effect, the smooth component, corresponding to dyeing, **T** is a random effect that corresponds to the main structure of the weft and $\epsilon$ is a random measurement error.

The methodology used to achieve these objectives consists of:

- To isolate the signal corresponding to the fungal dyeing process, we use Fourier transforms and regularization techniques to identify important low-frequency basis, as presented in Chapter 2.
- We model the random effect of the weft structure as a discrete Markov Random Field with pairwise interactions, presented in Chapter 3.
- To include the observational error, the theory of Hidden Markov Random Fields, presented in Chapter 4 is used.
- Finally, we recompute the same color homogeneity index defined for **S** and compare to commercially dyed samples.

# 2 Bivariate Fourier Basis and Filtering

The analysis of this dataset is challenging because not only the presence of the texture will bring a change in the color of neighboring pixels, but also, any inclination of the fibers and the thickness of wart and weft makes it difficult to measure homogeneity directly from the data.Therefore, it is crucial to separate the signal that comes from the texture and the one that comes from the dyeing process.

Note that, from the production method, the fabric texture signal presents a periodic pattern, what leads to the idea of using sines and cosines functions to model it. Separating the high-frequency signals from very low-frequency ones should isolate the texture and give us an exploratory tool to start analyzing the images.

In the following sections, we introduce the two-dimensional Fourier basis expansion, in its trigonometric (real) version, and some frequency-based filters that are useful for grouping signals according to their periodicity and strength, giving us a first idea of how to isolate the texture.

## 2.1 Two-dimensional Fourier Basis

Consider a rectangular grid of real values (an image) $Y \in R^{NM} = Y(x, y)$, $0 \leqslant x \leqslant N-1$, $0 \leqslant y \leqslant M-1$. We can assume $Y$ corresponds to observations of a function $f(x, y)$ and write this surface as a combination of basis functions

$$Y(x, y) = f(x, y) = \sum_{k=1}^{p} \beta_k \phi_k(x, y). \tag{2.1}$$

A matrix representation $Y = W\beta$ can be obtained by considering $Y$, a $NM \times 1$ vector, where each row corresponds to the observed value in a coordinate $Y(x, y)$, $W$ a $NM \times p$ matrix, with rows corresponding to coordinates $(x, y)$ and columns to basis functions $\phi_k(x, y)$ and $\beta$ the $p \times 1$ vector of coefficients $\beta_k$. Note that, the number of basis $p$ required to interpolate $f(x, y)$ at each point of $Y$ is, at most, $NM$.

One particular choice of basis functions is Fourier basis. These functions have the form

$$\phi_k(x, y) = \frac{1}{c_k} \phi_k^{(1)} \left( \frac{2\pi n_k x}{N} \right) \phi_k^{(2)} \left( \frac{2\pi m_k y}{M} \right), \tag{2.2}$$

where $\phi_k^{(1)}(.)$ and $\phi_k^{(2)}(.)$ are trigonometric functions sin(.) or cos(.) and integers $n_k$ and $m_k$ corresponding, respectively, to $x$-axis and $y$-axis directions frequencies, and $c_k$ is a constant.

Some of these basis functions are illustrated at Figures 4 to 7 on 100 × 100 pixels images ($N = M = 100$). The colors of the plots are relative to the function value, so the constant doesn't interfere. Notice how combining these basis can create repetitive patterns in specific directions, accommodating different possibles orientations of the image.

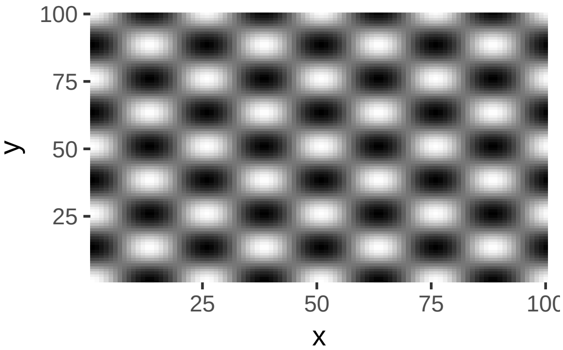


Figure 4 – $\phi_{k_1} = \cos\left(\frac{2\pi 4x}{100}\right)\cos\left(\frac{2\pi 4y}{100}\right)$

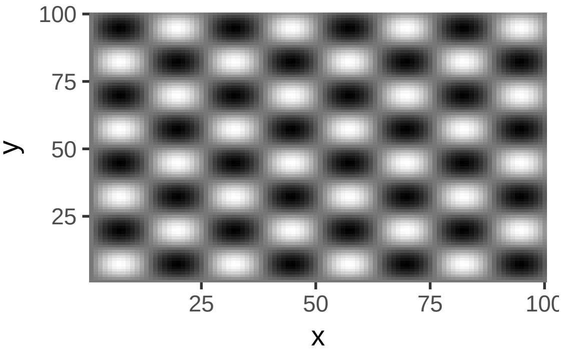


Figure 5 – $\phi_{k_2} = \sin\left(\frac{2\pi 4x}{100}\right)\sin\left(\frac{2\pi 4y}{100}\right)$

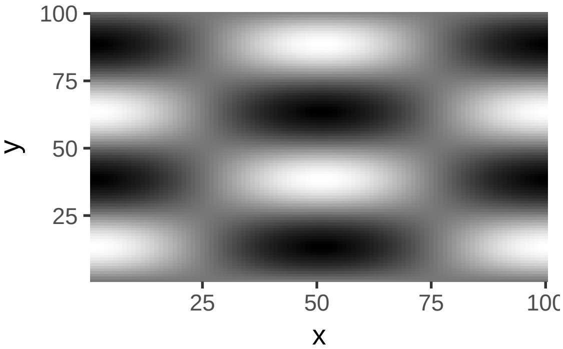


Figure 6 – $\phi_{k_3} = \cos\left(\frac{2\pi 1x}{100}\right)\sin\left(\frac{2\pi 2y}{100}\right)$

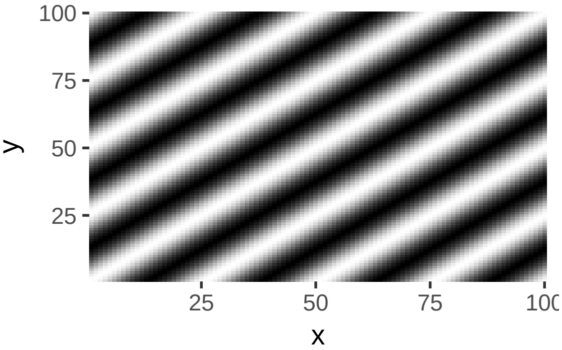


Figure 7 – $\phi_{k_1} + \phi_{k_2}$

Many combinations of $n$, $m$, $\phi^{(1)}$ and $\phi^{(2)}$ lead to linear dependent columns of matrix $W$, due to trigonometric identities. In this work, we select a set of $NM$ columns for $W$ that are linearly independent to form our basis functions. These columns are orthogonal by construction and we also choose $c$ to normalize each column of $W$, obtaining an orthonormal transformation matrix.

For the coefficients $\beta$ computation, we use the fast Fourier transform algorithm, which is very efficient for the complex version of Fourier basis and is available in most softwares, like R, and obtain those coefficients by the relations between the complex and the trigonometric forms of the Fourier transform, see Alexander and Poularikas (1998).

## 2.2 Filtering Frequencies

To obtain a particular type of signal, for example, the non-periodic part of the image, we can try to select a subset $\mathcal{B}$ of basis functions $\phi_k$ that corresponds to it

$$Y^*(x, y) = \sum_{k \in \mathcal{B}} \beta_k \phi_k(x, y), \tag{2.3}$$

particularly for textures, we observe a very repetitive pattern. For this reason, we expect functions $\phi_k$ with at least one high frequency $n_k$ or $m_k$ to be important for texture signal, while functions with low values for both frequencies (what we will refer as low-frequency) should be sufficient to describe blurs in the image.

Another important relationship between the Fourier representation and the image features is the absolute value of coefficients $\beta_k$, as values close to zero indicate that these functions can be discarded in order to describe the original image. Note that these coefficients are directly comparable because we used an orthonormal basis.

Consider the 378 by 237 pixels image in Figure 8 and its 89586 corresponding Fourier basis coefficients in Figure 9 as an example. For illustrating the effects of considering different subsets of basis functions, we will consider four groups of basis:

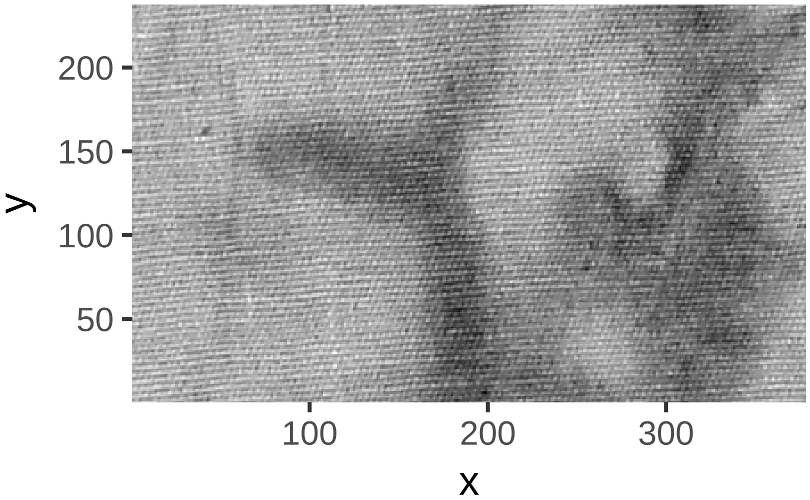


Figure 8 – Original image $Y$.

- (a) Low frequencies: functions corresponding to coefficients at left side of the black dashed line, expected to capture the blurry signal.
- (b) Thresholding frequencies: functions corresponding to coefficients above the white dashed line, expected to recover the original image with very small error.
- (c) Thresholded high frequencies: above the white dashed line and to the right the black dashed line, expected to capture the texture.

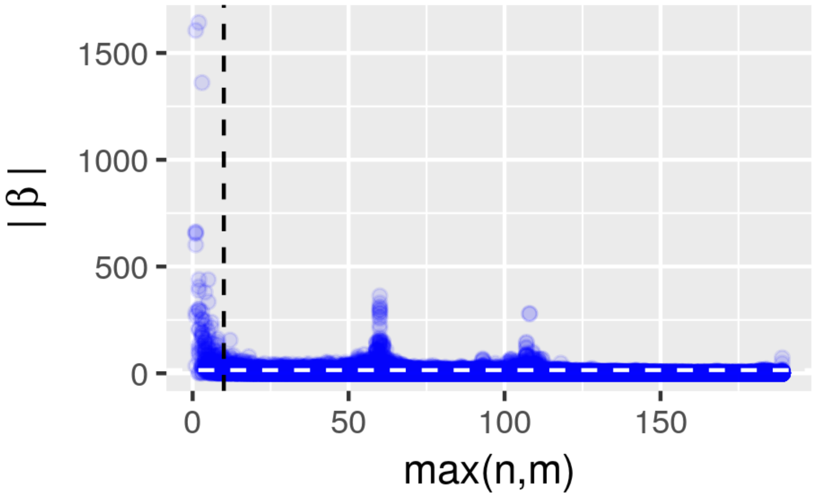


Figure 9 – Absolute coefficients of basis in the $y$-axis and maximum of $n_k$ and $m_k$ for each basis $\phi_k$.

- (d) Least important frequencies: below the white dashed line. It should have no pattern, corresponding to noise.

We arbitrarily chose the values used for frequency selection (black dashed line) and coefficient thresholding (white dashed line) as 10 and 15, respectively. With these values, 247(0.02%) basis were selected by (a), 3007(3.37%) by (b) and 2779(3.12%) by (c).

The resulting images are shown in Figures 10 to 13. A small number of periodic functions is enough to describe each type of signal in our image and all the remaining (the noise) have no apparent structure.

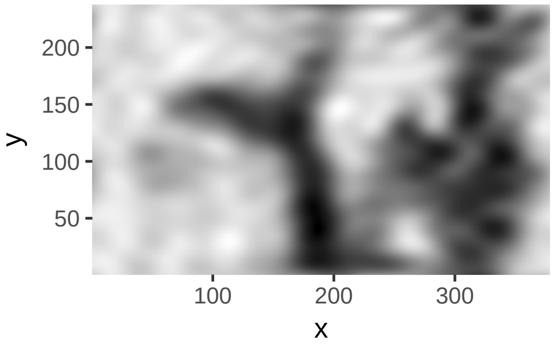


Figure 10 – Image reconstructed with (a).

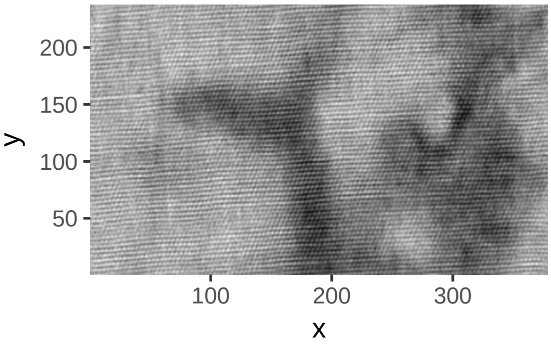


Figure 11 – Image reconstructed with (b).

In conclusion, we can capture the effect of the dyeing process with low-frequency functions and we can ignore a large number of higher frequency basis that have small coefficients and still recover a very similar image, as these basis corresponds to noise. In the example above,the values 10 and 15 were chosen arbitrarily, however we need a statistical method for answering questions such as: How to select which frequencies are

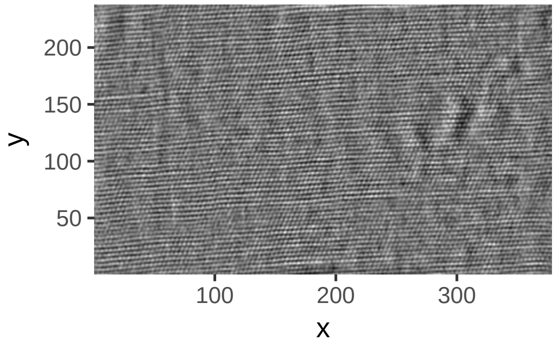


Figure 12 – Image reconstructed with (c).

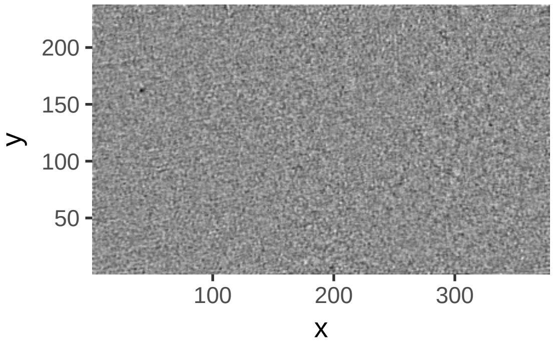


Figure 13 – Image reconstructed with (d).

considered low and how to determine whether a coefficient is small enough to remove the corresponding basis.

## 2.3 Penalized Least-Squares

Penalized least-squares is a modification of the usual ordinary least-squares used in regression problems that is able to achieve regularization, for example, by producing sparse coefficients estimates. This type of variable selection is done by introducing a penalty function to the objective function in the optimization problem. In a regression model with $k$ covariates, $Y = W\beta + \epsilon$, where $\epsilon$ is a vector of independent Gaussian distributed errors with zero mean, and $W$ a $n \times k$ design matrix, we can write the penalized least-squares $(\hat{\beta}_{PLS})$ estimate, given penalty functions $p_i(\cdot), 1 \leqslant i \leqslant k$, as

$$\hat{\beta}_{PLS} = \arg\min_{\beta} ||Y - W\beta||_2^2 + \sum_{i=1}^{k} p_i(|\beta_i|). \tag{2.4}$$

Different choices of penalty functions lead to different characteristics of the estimate. Some particular cases of penalty functions include $p_i(|\beta_i|) = \lambda|\beta_i|$, which leads to the Least Absolute Shrinkage and Selection Operator (LASSO) from Tibshirani (1996), $p_i(|\beta_i|) = \lambda(1 - \delta(|\beta_i|, 0))$, where $\delta(a, b) = 1$, if $a = b$ and 0 otherwise, which is a best subset selection method (can be seen as a modification of selection based on AIC) and $p_i(|\beta_i|) = \lambda|\beta_i|^2$ that corresponds to Ridge Regression (Hoerl and Kennard (1970)).

While the optimization in (2.4) can be quite difficult depending on the choice of penalty function, for orthogonal $W$ matrix, this estimation process can be done componentwise and has a closed form for many types of penalties. Particularly, for LASSO and best subset methods, the penalized least-squares estimators are obtained by soft-thresholding and hard-thresholding, respectively, as shown in Fan and Li (2006). Unlike Ridge Regression, these methods of regularization lead to sparsity of the estimates, which we will be interested on.

Table 1 and Figure 14 illustrate how penalized least-squares can be obtained from ordinary least-squares, denoted by $\hat{\beta}$ when the design matrix $W$ is orthogonal. $a_\lambda$ and $b_\lambda$ are thresholding constants that depend on the penalizing constant $\lambda$.

| Method | Penalty | Estimator |
|---|---|---|
| LASSO | $p_i(\beta_i) = \lambda\|\beta_i\|$ | $\hat{\beta}_{PLS} = (\hat{\beta} - a_\lambda sign(\hat{\beta}))\mathbb{1}_{\|\hat{\beta}\| \geqslant a_\lambda}$ |
| Best Subset | $p_i(\beta_i) = \lambda(1 - \delta(\|\beta_i\|, 0))$ | $\hat{\beta}_{PLS} = \hat{\beta}\mathbb{1}_{\|\hat{\beta}\| \geqslant b_\lambda}$ |

Table 1 – Penalized least-squares estimators for LASSO and Best Subset.

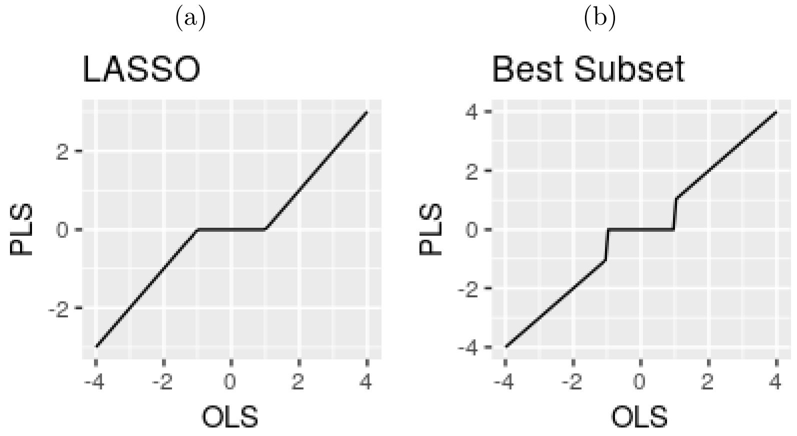


Figure 14 – Relation between Ordinary Least-Squares (OLS) and Penalized Least-Squares estimates for the two mentioned cases. Penalty constant $\lambda$ was chosen to have $a_\lambda = b_\lambda = 1$.

## 2.4 Frequency Filtering via Penalized Least-Squares

As mentioned in Section 2.1, we can write $Y = W\beta$, where $W$ is an orthonormal $n \times n$ matrix with each column corresponding to a certain Fourier basis function. We can see the problem of filtering a smaller set of basis functions to represent an image (and consider all the remaining signal as noise) as a variable selection problem. Note how this is a similar problem to choosing the order of a polynomial to fit a regression problem.

Directly applying a variable selection method like LASSO to the image data will produce a sparse representation of the image like done in Figure 11, but it won't be able to isolate low-frequency signal. In order to obtain a method for extracting low-frequency signal only, we make a slight modification of the previously presented methods by introducing a varying penalty value. Let $\phi_i(\cdot)$ be the Fourier basis corresponding to the $i$-th column of $W$, and $n_i, m_i$ its corresponding pair of frequencies. We can then define new penalty functions as $p_i(|\beta_i|) = \lambda_i|\beta_i|$ for LASSO and $p_i(|\beta_i|) = \lambda_i(1 - \delta(|\beta_i|, 0))$ for best subset, with

$$\lambda_i = g(\max(n_i, m_i)). \tag{2.5}$$

In this case, we can increase the penalty for basis associated high-frequency components, by choosing $g(\cdot)$ as a non-negative strictly increasing function. In this case, considering the orthogonal design matrix $W$, each coefficient $\beta_i$ in the solution of 2.4 will be given by hard-thresholding or soft-thresholding, depending on the type of penalty, and with varying constants $a_{\lambda_i}$ and $b_{\lambda_i}$ rather than a fixed value.

Note that there is a unique relation between the thresholding constants $a_{\lambda_i}, b_{\lambda_i}$ and the penalty function constant $\lambda_i$, and due to easier interpretation, we won't describe the constants used in the optimization process, but the corresponding constants in the solution. In summary, there is a relationship between a penalized least-squares regularization of the linear model with Fourier basis and thresholding frequencies.

To exemplify the usage of penalized least-squares for isolating the low-frequency signal consider $Y$ the image from Figure 8 again. We start by computing the Fourier basis representation of the image $W\beta$, then we hard-threshold the coefficients depending on the corresponding frequency, by using $b_{\lambda_i} = \exp\left(\dfrac{\max(n_i, m_i)}{2}\right)$, obtaining the low-frequency signal $Y_{low}$. Results are shown in Figure 15.

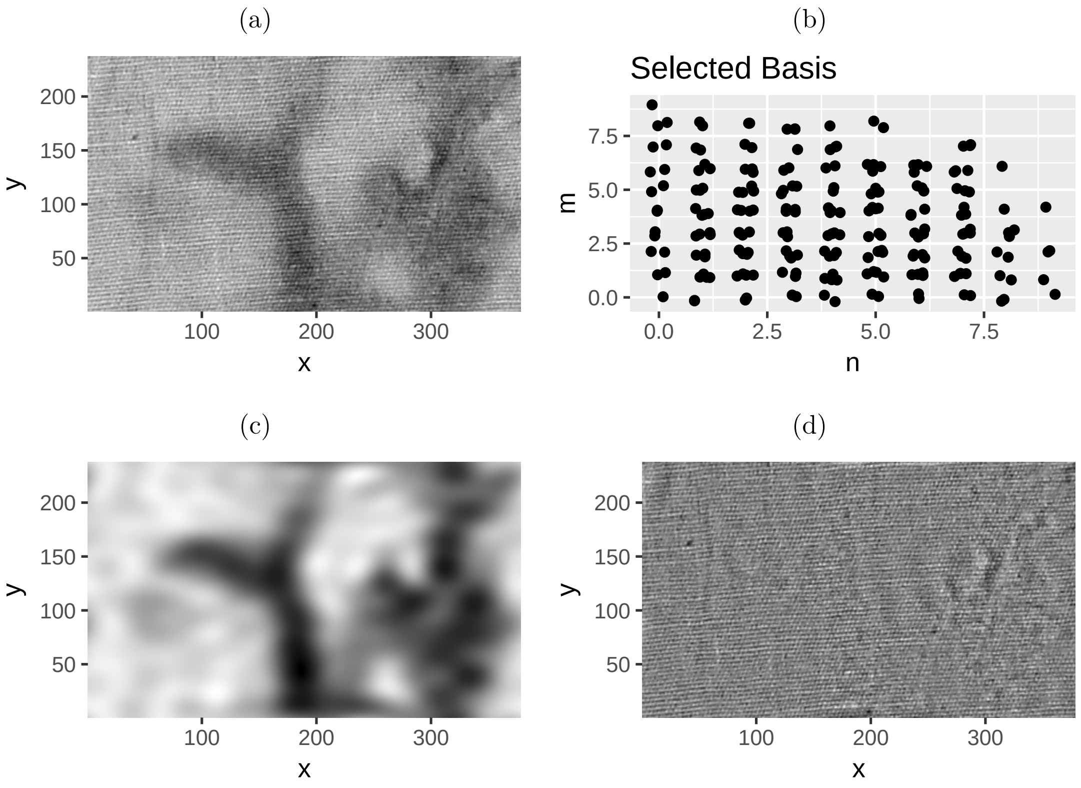


Figure 15 – Penalized Least-Squares based low-frequency extraction. (a) is the original image $Y$, (b) has the frequencies which basis were selected (with some noise to visualize multiple functions with the same frequencies), (c) is the reconstructed image from the selected basis functions $Y_{low}$ and (d) is the residual $Y - Y_{low}$.

Note how the large penalty for high-frequency functions prevented them from being included in the set of selected basis, the highest frequency among the included basis is 9. Since the high-frequency signal isn't present in $Y_{low}$, all the periodic pattern from the texture is included in the residual. As a second step, we repeat this process with a constant thresholding value $b_\lambda = 15$ to the image in (d) to separate the high-frequency signal from the noise, extracting the high-frequency part $Y_{high}$. In the end, we are able to separate the signal in the image as a low-frequency signal $Y_{low}$, a high-frequency signal $Y_{high}$ and a random noise $\hat{\epsilon}$ with

$$Y = Y_{low} + Y_{high} + \hat{\epsilon}. \tag{2.6}$$

Results from the second step are shown in Figure 16. The image in Figure 16(c) successfully captures the texture and $\hat{\epsilon}$ seems to correspond purely to random noise. Similar quality results are obtained when using the same procedure with soft-thresholding instead. We prefer to use hard-thresholding because it results in unbiased estimates.

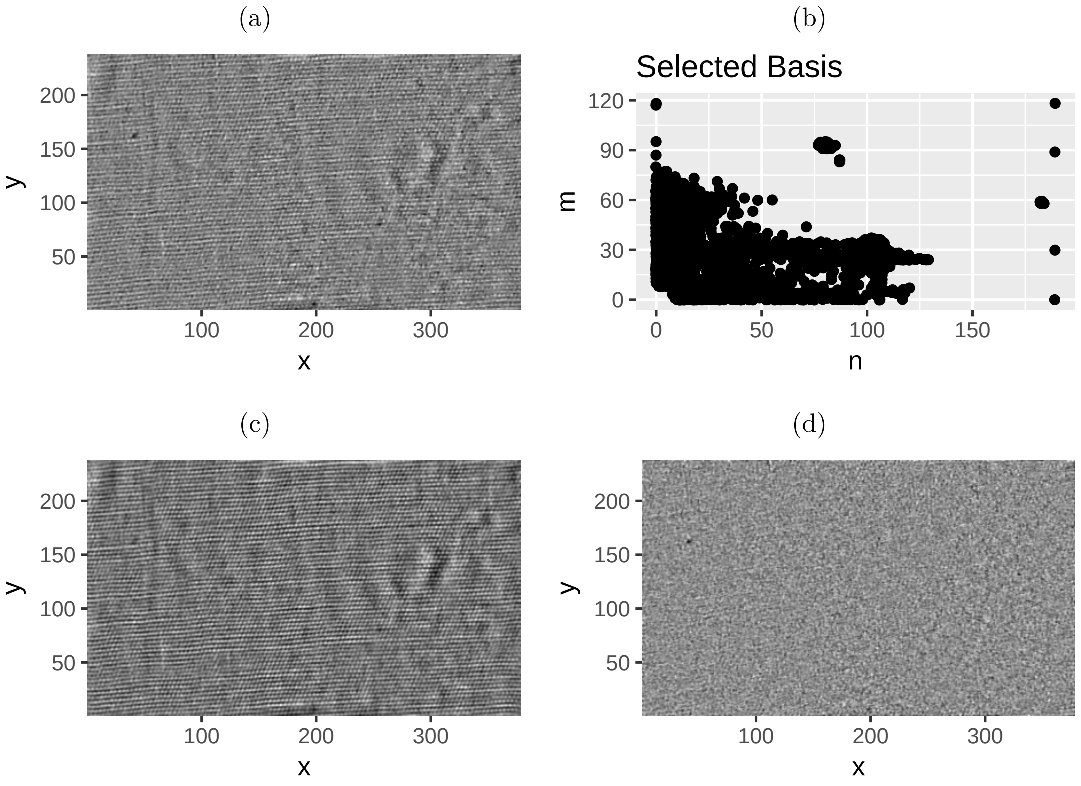


Figure 16 – Extraction of the high-frequency signal. (a) is the starting image $Y - Y_{low}$, (b) has the frequencies which basis were selected (with a small noise to visualize multiple functions with the same frequencies), (c) is the reconstructed image from the selected basis functions $Y_{high}$ and (d) is the final residual $\hat{\epsilon}$.

# 3 Markov Random Fields

## 3.1 Introduction to Markov Random Fields

Markov Random Fields are a generalization of Markov Chains for processes with a local dependence feature but no natural sequence-like structure. In general, processes with local dependence can be described by a graph, with edges representing the dependence structure.

Let $\mathbf{X} = \{X_1, \dots, X_n\}$ be a set of discrete and finite random variables and $\mathcal{G} = (\mathbf{X}, \mathcal{N})$ a graph with $\mathbf{X}$ as vertices and neighborhood system $\mathcal{N}$. Denote $X_{\mathcal{N}_i}$ the set of neighbors of $X_i$ according to $\mathcal{N}$ and $X_{-i} = \{X_j, j \neq i\}$.

**Definition 1** (Markov Random Field)**.** *The set of finite discrete-valued random variables* $\mathbf{X}$ *is a Markov Random Field (with respect to the neighborhood system* $\mathcal{N}$*) if it satisfies:*

$$P(X_i = a | X_{-i}) = P(X_i = a | X_{\mathcal{N}_i}), \quad i = 1, \dots, n, \quad \forall a \tag{3.1}$$

**Definition 2** (Clique)**.** *A subset* $X_{\{k\}} = \{X_{k_1}, \dots, X_{k_{l(k)}}\}$ *of* $\mathbf{X}$ *is a clique if* $\forall i, j \in \{k\}$, $X_i \in X_{N_j}$ *and* $X_j \in X_{N_i}$ *(all of its elements are neighbors). A single element or the empty set are also a clique.*

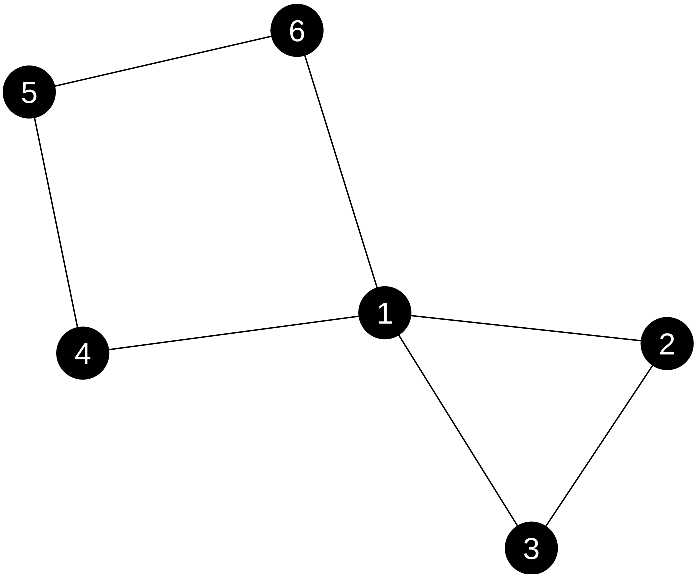


Figure 17 – Example of graph.

Note that, the number of elements $l(k)$ in a clique varies and any subset of a clique is also a clique. Figure 17 is an example of graph with cliques $\{1, 2, 3\}$, $\{1, 4\}$, $\{1, 6\}$, $\{5, 6\}$, $\{4, 5\}$, as well as each vertex individually.

Hammersley-Clifford theorem, see Besag (1974), states that, under the positivity condition (every configuration $\mathbf{x}$ for the random field $\mathbf{X}$ has positive probability), if $X$ is a Markov Random Field with cliques set of $\mathcal{K}$, its probability function can be written

$$P(\mathbf{X} = \mathbf{x}) = \frac{1}{Z_V} \exp\left[\sum_{k \in \mathcal{K}} V_k(x_k)\right], \tag{3.2}$$

for some:

$$V_k : \{0, 1, \ldots, G\}^{l(k)} \to \mathbb{R},$$

$$Z_V = \sum_{z \in \mathcal{X}} \exp\left[\sum_{k \in \mathcal{K}} V_k(z_k)\right],$$

$\mathcal{X} = \{0, 1, \ldots, G\}^n$ is the set of all possible configurations of X.

The probability mass function in (3.2) is referred as a Gibbs distribution (or Gibbs measure) and, for this reason, Markov Random Fields are also referred as Gibbs Random Fields. The functions $V_k$ are called clique potentials and $Z_V$ is referred as the partition function (in practice, it's a normalizing constant that depends on potentials).

## 3.2 Gibbs Distributions with Pairwise Interactions

Given a specific neighborhood system $\mathcal{N}$, the expression in (3.2) gives us a general form for the probability of possible configurations of the field $X$, but for moderate sized problems, the number of cliques $|\mathcal{K}|$ leads to a large number of parameters if no structure is assumed for clique potentials $V_k$, therefore, it is convenient to consider some class of Gibbs distributions that respects some structure with desired properties and can be represented with a reduced number of parameters.

Consider a finite rectangular lattice of dimension $N \times M$, $X = \{X_{(i_1,i_2)}, 0 \leqslant i_1 \leqslant N-1, 0 \leqslant i_2 \leqslant M-1\}$. For notation simplicity, we will write $X_{(i_1,i_2)}$ as $X_i$, but now $i$ represents a coordinate rather than just an index. We also refer to the variables $X_{(i_1,i_2)}$ as pixels because our field will represent an image. We also assume each pixel's color is in the set $\mathcal{Y} = 0, 1, \ldots, G$.

In the context of image processing, different types of analysis can be done using MRFs with nearest-neighbor interaction, see Besag (1986) and Geman and Geman (1984), but particularly for images with textures, a family of Gibbs distributions with more sophisticated dependence structure is necessary. In this case, we will consider a neighborhood system that has edges connecting pixels on a set of relative positions and assume the potentials for each pair of pixels in a specific relative position to be the same in the whole image what implies a spatial stationarity property. We will consider a model with these properties similar to Gimel'farb (1996) and a feature function for each interacting pair.

**Definition 3** (Feature Function)**.** *A feature function maps a pair in $\mathcal{Y}^2$ to a finite set $\mathcal{D}$. $h : \mathcal{Y}^2 \to \mathcal{D}$, $|\mathcal{D}| = D$. Some examples of feature functions are $h(a,b) = b - a$, $h(a,b) = |b - a|$ and $h(a,b) = 1 - \delta(a,b)$, where $\delta(a,b) = \mathbb{1}_{a=b}$.*

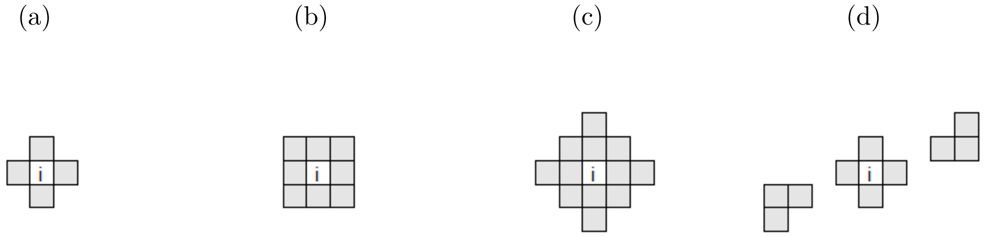
Let $C$ be a set of relative positions where pixels interact. We will work with (3.2) as

$$P(X = x) = \frac{1}{Z_V} \exp\left[\sum_i \left(V_0(x_i) + \sum_{c \in C} \sum_{j=i+c} V_c(h(x_i, x_j))\right)\right]. \tag{3.3}$$

Note that the last sum has, at most, one element because there can only be one pixel in a specific relative position of another specific pixel, but including this sum accommodates the irregularities in the borders where the neighbor, in some relative position, might not exist ($i + c$ doesn't belong to the lattice).

Figure 18 shows examples of interacting structures. The nearest-neighbor structure in Figure 18(a) is the simplest, with $C = \{(0,1),(1,0)\}$ (note that we don't need opposite directions of the same relative position in the model). Interaction structures shown in Figures 18(c) and 18(d), due to increased complexity, can produce a richer variety of patterns.

Figure 18 – Examples of dependence structures.



Let $g \in \mathcal{G}$ and $d \in \mathcal{D}$. Define $n_0(g) = \sum_i \delta(x_i, g)$, the number of occurrences of $g$ in an image, $n_c(d) = \sum_i \sum_{j=i+c} \delta(d, f(x_i, x_j))$, the number of occurrences of feature $d$ in pairs with relative position $c$, and $n_c = \sum_{d \in \mathcal{D}} n_c(d)$, total number of pairs with relative position $c$. Define parameters $\theta_c^d = V_c(d)$ and $\theta_0^w = V_0(w)$ (note that the knowledge of $V$ and $\theta$ are equivalent, we use the most appropriate notation in each case), we can rewrite (3.3) in terms of the occurrence counts

$$P(X = x) = \frac{1}{Z_\theta} \exp\left[\sum_{w \in \mathcal{Y}} \theta_0^w n_0(w) + \sum_{c \in C} \sum_{d \in \mathcal{D}} \theta_c^d n_c(d)\right]. \tag{3.4}$$

This allows us to notice two things: First, the statistic $\{n_c(d), n_0(g), c \in C, d \in \mathcal{D}, g \in \mathcal{Y}\}$ is a sufficient statistic for the vector of parameters $\theta$. And second, this representation of the probability distribution is not unique unless we add restrictions to the parametric space, as shown below.

Consider a different parameter vector $\psi^*$ which differs from the original $\theta$ only for a specific relative position $c$ by a constant, $\psi_c^d = \theta_c^d + u$ for some constant $u$. Then

$$\begin{aligned}
P_\psi(X = x) &= \frac{1}{Z_\psi} \exp\left[\sum_{w\in\mathcal{Y}} \psi_0^w n_0(w) + \sum_{c\in C}\sum_{d\in\mathcal{D}} \psi_c^d n_c(d)\right] \\
&= \frac{1}{Z_\psi} \exp\left[\sum_{w\in\mathcal{Y}} \theta_0^w n_0(w) + \sum_{c\in C}\sum_{d\in\mathcal{D}} \theta_c^d n_c(d) + n_c u\right] \\
&= \frac{\exp \sum_{d\in\mathcal{D}}(n_c^d v)}{Z_\psi} \exp\left[\sum_{w\in\mathcal{Y}} \theta_0^w n_0(w) + \sum_{c\in C}\sum_{d\in\mathcal{D}} \theta_c^d n_c(d)\right] \\
&= \frac{1}{Z_\theta} \exp\left[\sum_{w\in\mathcal{Y}} \theta_0^w n_0(w) + \sum_{c\in C}\sum_{d\in\mathcal{D}} \theta_c^d n_c(d)\right] = P_\theta(X = x),
\end{aligned}$$

this shows that by adding a constant value to all potentials of specific relative position (similar can be done for the single site potentials), we obtain a new set of parameters leading to the exact same distribution (non identifiability of the model). In order to obtain a unique representation of the distribution function in terms of $\theta$, the following linear restrictions are included

$$\sum_{d\in\mathcal{D}} \theta_c^d = 0, \forall c \qquad \text{and} \qquad \sum_{w\in\mathcal{Y}} \theta_0^w = 0. \tag{3.5}$$

Figures 19 – 23 illustrate how different sets of relative positions and potentials can create a variety of patterns on 100 × 100 lattices. They were simulated from binary Gibbs distributions, with specified interacting positions and potentials in Tables 2 – 6, respectively. Single site potentials were all set to zero for better visualization of the effect of relative positions selection and the feature function selected was $h(a, b) = |a - b|$. As can be seen, these models are particularly good for describing textures as they can accommodate many different stationary patterns.

The field in Figure 19 has positive parameters for equal pixel values ($\theta_c^0 > 0$) on both $(0, 1)$ and $(1, 0)$ relative positions. This causes pixels of the same value to cluster and form big regions of the same value along the image. The same is valid to Figure 20, except there are more isolated pixels with different value from it's neighbors, due to the parameters being closer to zero than the ones used in 19. This specification lead to a distribution equivalent to the Ising Model. On Figure 21 we see the effect of adding a repulsive effect in the diagonal $(1, 1)$ direction. This creates a texture-like effect. Figure 22 has a repulsive effect on relative position $(5, 0)$, what creates this pattern of 5 pixels wide stripes. Figure 23 has the most complex interaction structure and is the same as 21 with an additional repulsive effect on $(1,-1)$, what leads configurations of parallel lines alternating values.

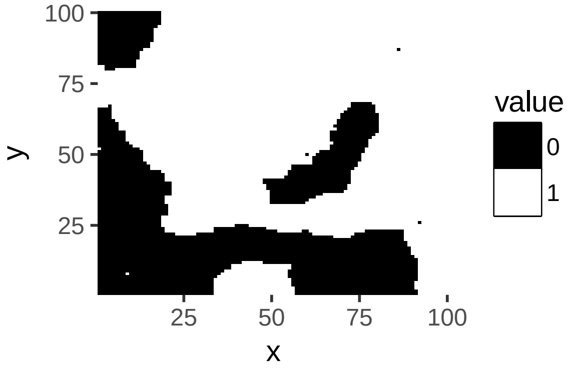


Figure 19 – Simulated Gibbs random field.

| $c$ | $\theta_c^0$ | $\theta_c^1$ |
|---|---|---|
| $(1,0)$ | 1 | $-1$ |
| $(0,1)$ | 1 | $-1$ |

Table 2 – Parameters used in Figure 19

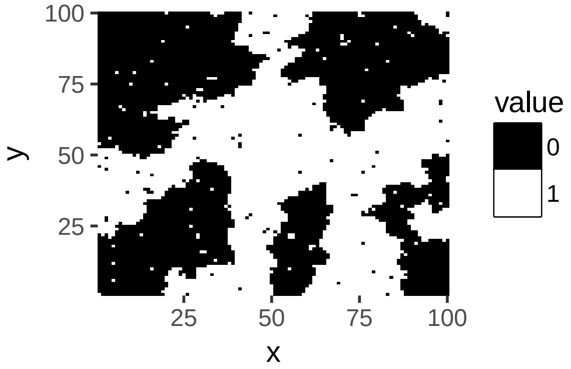


Figure 20 – Simulated Gibbs random field.

| $c$ | $\theta_c^0$ | $\theta_c^1$ |
|---|---|---|
| $(1,0)$ | 0.6 | $-0.6$ |
| $(0,1)$ | 0.6 | $-0.6$ |

Table 3 – Parameters used in Figure 20.

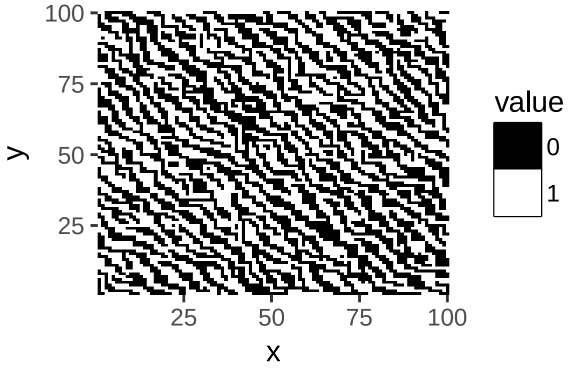


Figure 21 – Simulated Gibbs random field.

| $c$ | $\theta_c^0$ | $\theta_c^1$ |
|---|---|---|
| $(1,0)$ | 1 | $-1$ |
| $(0,1)$ | 1 | $-1$ |
| $(1,1)$ | $-1$ | 1 |

Table 4 – Parameters used in Figure 21.

Different specifications of feature function also leads to different types of patterns, as well as number of parameters. Table 7 presents notation and properties for some feature functions to be used in this work.

The main difference between these functions is that $h_1$ can have some directional effects, since it allows $h(a,b) \neq h(b,a)$. $h_2$ cannot detect different interactions on opposite directions, but the parametric space has its dimension reduced almost in half when

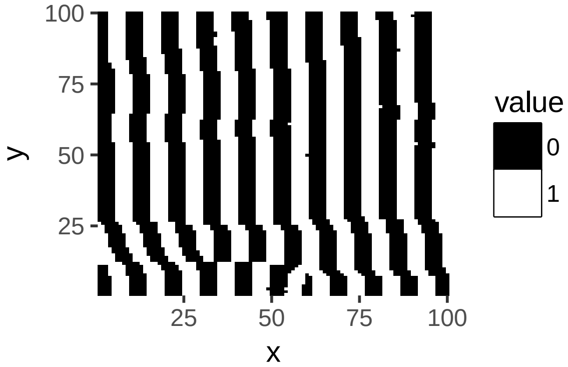


Figure 22 – Simulated Gibbs random field.

| $c$ | $\theta_c^0$ | $\theta_c^1$ |
|---|---|---|
| $(1, 0)$ | $1$ | $-1$ |
| $(0, 1)$ | $1$ | $-1$ |
| $(5, 0)$ | $-1$ | $1$ |

Table 5 – Parameters used in Figure 22.

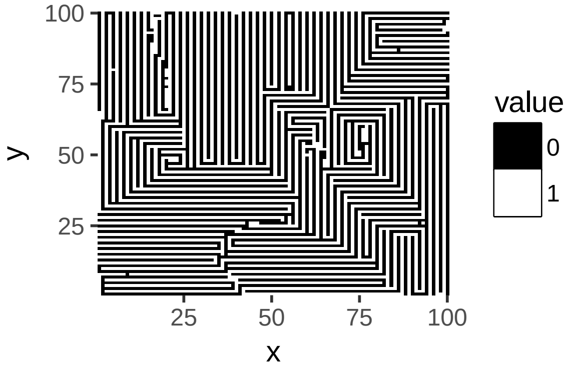


Figure 23 – Simulated Gibbs random field.

| $c$ | $\theta_c^0$ | $\theta_c^1$ |
|---|---|---|
| $(1, 0)$ | $1$ | $-1$ |
| $(0, 1)$ | $1$ | $-1$ |
| $(1, 1)$ | $-1$ | $1$ |
| $(1, -1)$ | $-1$ | $1$ |

Table 6 – Parameters used in Figure 23.

Table 7 – Feature functions and properties.

| Symbol | function | $\lvert\mathcal{D}\rvert$ | $\dim(\theta)$ |
|---|---|---|---|
| $f_1$ | $h_1(a, b) = b - a$ | $2G + 1$ | $G(2\lvert C\rvert + 1)$ |
| $f_2$ | $h_2(a, b) = \lvert b - a\rvert$ | $G + 1$ | $G(\lvert C\rvert + 1)$ |
| $f_3$ | $h_3(a, b) = 1 - \delta(a, b)$ | $2$ | $\lvert C\rvert + G$ |

compared to $h_1$. $h_3$ leads to a significant reduction on the number of free parameters and is mainly used in situations where the measured variable doesn't have a numeric meaning (for example if the values of $\mathbf{X}$ represents a field of different, unordered colors). Note the model with $h_2$ can be obtained from $h_1$ by forcing $\theta_c^a = \theta_c^{-a}, \forall c, a$ and $h_3$ can be considered the same as $h_2$ with the restriction $\theta_c^a = \theta_c^1\ \forall c, a > 1$.

Figures 24 – 26 show patterns that illustrate properties of each feature function. Figure 24 has an unitary increasing from left to right, and then a 4 units decrease. This directional asymmetric effect can only be obtained in models with $h_3$. Figures 25 and 26 demonstrates how $h_2$ and $h_3$ differ in terms of how the "sequence" of colors in horizontal direction following a more structured pattern or a completely random one.

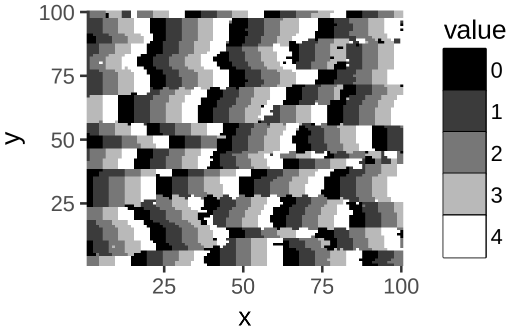


Figure 24 – Simulated Gibbs random field.

| $c$ | $\theta_c^{-4}$ | $\theta_c^{-3}$ | $\theta_c^{-2}$ | $\theta_c^{-1}$ |
|---|---|---|---|---|
| $(1,0)$ | $-1/8$ | $-1/8$ | $-1/8$ | $-1/8$ |
| $(0,1)$ | $-1/8$ | $-1/8$ | $-1/8$ | $-1/8$ |
| $(5,0)$ | $1$ | $0$ | $0$ | $-4$ |
| $\theta_c^{0}$ | $\theta_c^{1}$ | $\theta_c^{2}$ | $\theta_c^{3}$ | $\theta_c^{4}$ |
| $1$ | $-1/8$ | $-1/8$ | $-1/8$ | $-1/8$ |
| $1$ | $-1/8$ | $-1/8$ | $-1/8$ | $-1/8$ |
| $0$ | $3$ | $0$ | $0$ | $0$ |

Table 8 – Parameters used in Figure 24. Feature function $h_1 = b - a$ was used.

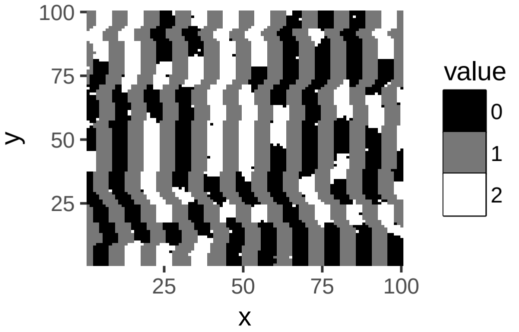


Figure 25 – Simulated Gibbs random field.

| $c$ | $\theta_c^{0}$ | $\theta_c^{1}$ | $\theta_c^{2}$ |
|---|---|---|---|
| $(1,0)$ | $1$ | $-1/2$ | $-1/2$ |
| $(0,1)$ | $1$ | $-1/2$ | $-1/2$ |
| $(5,0)$ | $-1$ | $2$ | $-1$ |

Table 9 – Parameters used in Figure 25. Feature function $h_2(a,b) = |b - a|$ was used.

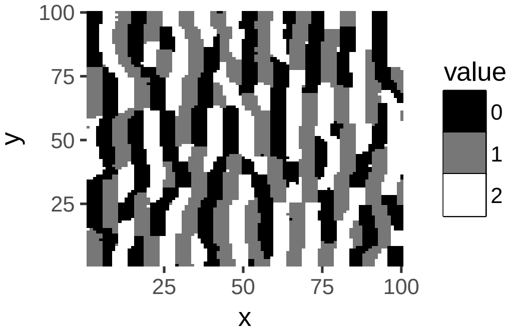


Figure 26 – Simulated Gibbs random field.

| $c$ | $\theta_c^{0}$ | $\theta_c^{1}$ |
|---|---|---|
| $(1,0)$ | $1$ | $-1$ |
| $(0,1)$ | $1$ | $-1$ |
| $(5,0)$ | $-1$ | $1$ |

Table 10 – Parameters used in Figure 26. Feature function $h_3(a,b) = 1 - \delta(a,b)$ was used.

## 3.3 Simulating from Markov Random Fields

Simulating realizations from a Markov Random Field plays an important role in studying and understanding these processes, but the normalizing constant $Z_\theta$ cannot be computed except for trivial cases (either in really small graphs/lattices or when all potentials are equal to zero) making it impossible to take samples from $\mathcal{X}$ directly. Fortunately, we can compute ratios of probabilities for different configurations (the intractable constant cancels out) and we can easily compute conditional distributions due Markovian dependence. These two facts allow to sample from Gibbs Random Fields via well-known methods like Metropolis-Hastings algorithm and Gibbs Sampler.

Consider two field configurations $\mathbf{x}$ and $\mathbf{y}$. For the family of Gibbs distributions described in previous section, the probability ratio, given a model specification is

$$\frac{P_\theta(\mathbf{X}=\mathbf{x})}{P_\theta(\mathbf{X}=\mathbf{y})} = \frac{\exp\left[\sum_i \left(V_0(x_i) + \sum_{c\in C}\sum_{j=i+c} V_c(h(x_i,x_j))\right)\right]}{\exp\left[\sum_i \left(V_0(y_i) + \sum_{c\in C}\sum_{j=i+c} V_c(h(y_i,y_j))\right)\right]}$$

$$= \exp\left[\sum_i \left(V_0(x_i) - V_0(y_i) + \sum_{c\in C}\sum_{j=i+c} V_c(h(x_i,x_j)) - V_c(h(y_i,y_j))\right)\right]. \tag{3.6}$$

Note that when $x$ and $y$ differ only in one pixel, this expression simplifies due to all terms in the sums not involving the differing pixel being zero.

And for conditional distribution of a specific pixel $X_i$,

$$P_\theta(X_i = a|X_{-i}) \propto \exp\left[V_0(a) + \sum_{c\in C}\left(\sum_{j=i+c} V_c(h(a,x_j)) + \sum_{k=i-c} V_c(h(x_k,a))\right)\right], \tag{3.7}$$

$\forall a \in \mathcal{Y}$. In this expression, we also have to include the relative positions in the opposite direction of the ones listed in the set $C$.

Equations (3.6) and (3.7) are sufficient to implement both Gibbs Sampler and Metropolis-Hastings algorithm for simulating Gibbs Random Fields. Gimel'farb (1999) implements a Gibbs Sampler algorithm for this model as follows:

**Algorithm 1** – Gibbs Sampler algorithm for Markov Random Field simulation.

```
begin
    Sample $\mathbf{x}^{(0)}$ from an independent discrete uniform distribution.;
    Set $j \leftarrow 0$;
    while $j < m$ do
        Sample a random sequence of pixels $S$ to visit where every pixel appears
          exactly one time.;
        for $S_i \in S$ do
            update pixel $x_{S_i}$ by sampling from $P_\theta(X_{S_i}|X_{-S_i} = x_{-S_i})$.
        end
        $j \leftarrow j + 1$
    end
end
```

Steps 2 and 3 together are called a macrostep, and according to general Gibbs Sampler theory, if the number of macrosteps $m$ is sufficiently large, the last field sampled is a sample from $P_\theta$. This procedure produces a single sample from the random field and due to the possibility of using parallel computing with recent technology and (empirically observed) poor mixing properties for some parameters specifications, when sampling

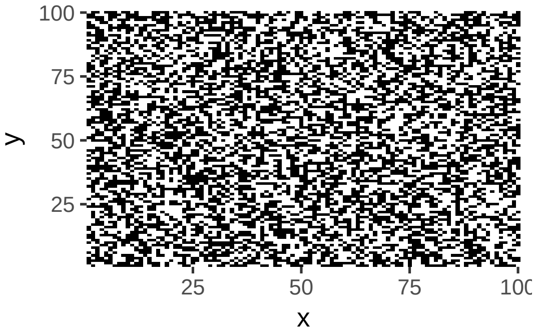


Figure 27 – Initial field configuration.

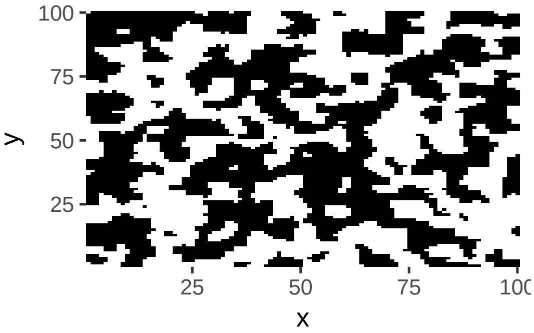


Figure 28 – Field after 5 macrosteps.

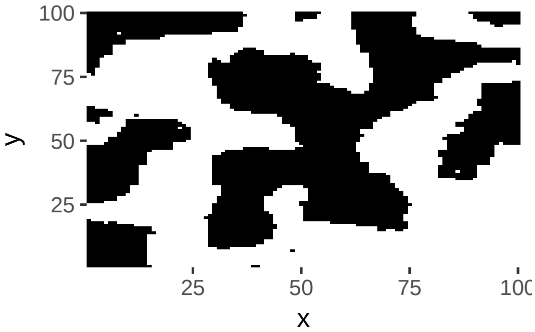


Figure 29 – Field after 50 macrosteps.

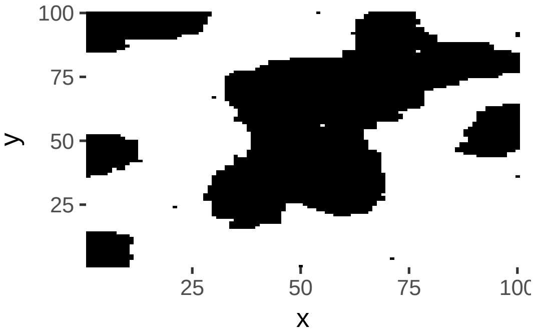


Figure 30 – Field after 150 macrosteps.

multiple random field configurations we use multiple independent chains, rather than taking spaced samples from a single chain.

Figures 27 – 30 presents steps of Gibbs Sampler algorithm for the model specified in Table 2 after a specific number of macrosteps. Our experiments suggests that both Gibbs Sampler and Metropolis-Hastings algorithm perform very similar. For this work we chose to use Gibbs Sampler because the computation of conditional probabilities can be reused when working with Hidden Markov Random Fields.

## 3.4 Parameter Estimation in Markov Random Fields

As mentioned in the previous section, the likelihood (or probability mass) function for MRFs cannot be computed due to the intractable constant $Z_\theta$, and therefore, maximum likelihood estimation for parameters cannot be done directly. Many different approaches have been proposed to obtain consistent estimates in this kind of model. Some of these methods include using an alternative, easier to compute, objective function called Pseudo-Likelihood, searching for estimates using approximations of the gradient of the (concave) Likelihood Function, as in Stochastic Approximation methods or attempts to find Monte-Carlo approximations of the likelihood function itself. Some of these methods are briefly presented in following subsections.

### 3.4.1 Maximum Pseudo-Likelihood

An alternative to find estimates for parameters in models with complicated interactions, that leads to intractable normalizing constants in the joint distribution (likelihood function), is by taking the product of conditional distributions of each variable instead. The pseudo-likelihood function is defined as

**Definition 4** (Pseudo-Likelihood function)**.** *The Pseudo-Likelihood function for a vector of parameters $\theta$ and a fixed observed field $\mathbf{x} \in \mathcal{X}$ is defined as*

$$PL(\theta; \mathbf{x}) = \prod_{i=1}^{n} P_\theta(X_i = x_i | X_{-i} = x_{-i}). \tag{3.8}$$

In MRFs particularly, these conditional distributions are simplified due to the local dependence property. Recall from Equation (3.7) how conditional probabilities are easily computed for the pairwise interactions model, except for a proportionality constant, which in this case can be computed, since it requires summing $|\mathcal{X}|$ terms rather than $|\mathcal{X}|^{NM}$ terms in the joint distribution. Maximum Pseudo-Likelihood Estimator ($\hat{\theta}_{MPLE}$) is defined as

$$\hat{\theta}_{MPLE} = \arg\max_\theta PL(\theta; \mathbf{x}), \tag{3.9}$$

which is a naive, but easier to compute, estimate for $\theta$. Discussion and results for the MPLE can be found at Besag (1975).

### 3.4.2 Monte-Carlo Maximum Likelihood

In order to introduce the Monte-Carlo Likelihood, recall the parametrization of Gibbs Distributions in Equation (3.4). Consider a vector $t(\mathbf{x})$ that contains the counts of pixel values $n_0(w)$ and counts for each feature $d$ in every relative position $n_c(d)$ and $\theta$ the vector of parameters associated to each respective component of $t(\mathbf{x})$. The equation can be rewritten with an inner-product as

$$P_\theta(X = x) = \frac{1}{Z_\theta} \exp \langle t(\mathbf{x}), \theta \rangle, \tag{3.10}$$

and $Z_\theta$ in terms of the sufficient statistics

$$Z_\theta = \sum_{\mathbf{z} \in \mathcal{X}} \exp \langle t(\mathbf{z}), \theta \rangle.$$

Consider a fixed and known set of parameters $\theta_0$. If we look at the ratio

$$\frac{Z_\theta}{Z_{\theta_0}} = \frac{\sum_{\mathbf{z} \in \mathcal{X}} \exp \langle t(\mathbf{z}), \theta \rangle}{Z_{\theta_0}} =$$
$$\frac{\sum_{\mathbf{z} \in \mathcal{X}} \exp \langle t(\mathbf{z}), \theta - \theta_0 \rangle \exp \langle t(\mathbf{z}), \theta_0 \rangle}{Z_{\theta_0}} =$$
$$\sum_{\mathbf{z} \in \mathcal{X}} \exp \langle t(\mathbf{z}), \theta - \theta_0 \rangle P_{\theta_0}(\mathbf{X} = \mathbf{z}) = E_{\theta_0}(\exp \langle t(\mathbf{X}), \theta - \theta_0 \rangle),$$

we obtain the expected value of a random variable. We can combine this result with the fact that it is possible to simulate from the Gibbs Distribution with parameters $\theta_0$ to obtain a Monte-Carlo approximation to the normalizing constants ratio. Let $\mathbf{W}_1, \mathbf{W}_2, \ldots \mathbf{W}_K$ be a sample of random fields from a Gibbs Distribution with parameters $\theta_0$,

$$\hat{R}_K(\theta) = \sum_{i=1}^{K} \frac{1}{K} \exp \langle t(\mathbf{W}_i), \theta - \theta_0 \rangle \xrightarrow[K\to\infty]{a.s.} \frac{Z_\theta}{Z_{\theta_0}}. \tag{3.11}$$

Note that multiplying our likelihood function by a constant factor (in this case, $Z_{\theta_0}$) doesn't affect maximum-likelihood estimation, therefore it is possible to define the MLE $\hat{\theta}_{MLE}$, given an observed field $\mathbf{x}$ as

$$\hat{\theta}_{MLE} = \arg\max_{\theta} \langle t(\mathbf{x}), \theta \rangle - \log \frac{Z_\theta}{Z_{\theta_0}}. \tag{3.12}$$

By combining the results in (3.11) and (3.12), we use the Monte-Carlo approximation of the ratio $\dfrac{Z_\theta}{Z_{\theta_0}}$ in the latter.

**Definition 5** (Monte-Carlo MLE)**.** *The Monte-Carlo MLE of* $\theta$*, with a size* $K$ *Monte-Carlo sample approximation to the normalizing constants ratio,* $\hat{\theta}^K_{MCE}$

$$\hat{\theta}^K_{MCE} = \arg\max_{\theta} \langle t(\mathbf{x}), \theta \rangle - \log \hat{R}_K(\theta). \tag{3.13}$$

A detailed description and discussion of this method can be found at Geyer (1991) and Geyer and Thompson (1992), with examples and comparisons to maximum pseudo-likelihood methods. In general, maximum pseudo-likelihood estimates are not always good for finite samples, specially when the dependence is too strong. On the other hand, Monte-Calor MLE requires sampling from a Markov Random Field a possibly large number of times ($K$), the averaging of large exponentials and the optimization of a function that has considerable computational cost to evaluate.

This method is also sensible to the arbitrary choice of $\theta_0$. Although the asymptotic convergence in (3.11) is valid for any choice of $\theta_0$, if the proposed value is not close enough to the actual MLE, the number of Monte-Carlo samples $K$ to achieve good results becomes unreasonable.

### 3.4.3 Stochastic Approximation

If we take the first derivative of $\log(P_\theta(\mathbf{X} = \mathbf{x}))$, we get

$$\frac{\partial \log P(\mathbf{X} = \mathbf{x})}{\partial \theta} = \frac{\partial}{\partial \theta} \left( \langle t((\mathbf{x}), \theta) \rangle - \log(Z_\theta) \right),$$

and

$$\frac{\partial}{\partial \theta} \log(Z_\theta) = \frac{\partial}{\partial \theta} \log(\sum_{\mathbf{z}\in\mathcal{X}} e^{\langle t(\mathbf{z}), \theta \rangle}) = \frac{1}{Z_\theta} \sum_{\mathbf{z}\in\mathcal{X}} t(\mathbf{z}) e^{\langle t(\mathbf{z}), \theta \rangle} = \mathbb{E}_\theta(t(\mathbf{X})).$$

Note that, both $t(\mathbf{x})$ and $\theta$ have the same dimension, because the same number of linear restrictions on $\theta$ (0 sum for each relative position and the single-pixel parameters) also exist for $t(\mathbf{x})$ because $\sum_{d \in \mathcal{D}} n_c(d)$ is also fixed for each relative position.

Due to exponential family properties, found for example in Barndorff-Nielsen (2014), the likelihood function $P_\theta(\mathbf{X} = \mathbf{x})$ is a log-concave function with respect to $\theta$, therefore, the MLE $\hat{\theta}$ is the point that satisfies the system of equations

$$t(\mathbf{x}) - E_\theta(t(\mathbf{X})) = 0. \tag{3.14}$$

The stochastic approximation algorithm proposed in Robbins and Monro (1951) consists of an iterative algorithm to solve (3.14), by obtaining estimates of $E_\theta(t(\mathbf{X}))$ and successively moving the current solution in the (estimated) gradient direction with predefined step sizes,

$$\theta^{[i+1]} = \theta^{[i]} + a^{[i]}(t(\mathbf{x}_0) - t(\mathbf{x}^{[i]})), \tag{3.15}$$

where $\mathbf{x}_0$ is the observed (learning) field, $\mathbf{x}^{[i]}$ is a field sampled with $\theta^{[i]}$ and $a^{[i]} > 0$ is the $i$-th step size, satisfying

$$\sum_{i=1}^{\infty} a^{[i]} = \infty, \qquad \sum_{i=1}^{\infty} (a^{[i]})^2 < \infty. \tag{3.16}$$

Note that the estimates $\theta^{[i]}$ are random variables (particularly, a Markov Chain) as they depend on the simulated field from the previous value and the conditions in (3.16) are necessary for convergence $\theta^{[i]} \to \hat{\theta}$ in probability. This method resembles Newton-Raphson algorithm, but with an estimate for the gradient and a constant vector instead of the Hessian matrix at each step.

For this work, when iterating (3.15), we take advantage of parallel computing to simulate multiple samples with $\theta^{[i]}$ and average the sufficient statistics instead of taking a single sample $x^{[i]}$, what results in a more precise gradient estimate at each step and faster convergence.

For the step size sequence, we use an approach similar to Gimel'farb (1996), and consider the sequence $a^{[i]} = \dfrac{c_0}{n(c_1 + c_2 i)}$, that allows to control the rate of convergence with the constants $c_0, c_1, c_2$ and make the relation between chosen constant constants and the convergence characteristics not depend on the dimension ($n$) of $\mathbf{X}$.

When using this algorithm, we will always start with $\theta^{[0]} = \mathbf{0}$. Defining reasonable stopping condition would involve considering image size and the dependence structure, so we fix the large number of iterations like 2000.

Stochastic Approximation produced better results when estimating parameters for high complexity interaction structures in simulations, as can be seen in Appendix A, and since we are interested on complex high dependence models necessary to describe textures, we use Stochastic Approximation when estimating parameters for MRFs.

## 3.5 Interaction Structure Specification

While there are multiple approaches for estimating the parameters of Markov Random Fields, all these methods requires prior knowledge of the interacting structure, in this model, specified by the set of relative positions $C$.

Csiszár and Talata (2006) proposes the estimation of the smallest region neighborhood based on a pseudo-Bayesian information criterion, which is an adaptation of the usual BIC but considering the pseudo-likelihood instead of the likelihood function. Particularly for our textures, we suppose the interaction structure can be sparse with spread relative positions, therefore, it is not possible to define these interaction structures in a set of naturally nested models. For this reason, we can't define a sequence of candidate models to compare, requiring the comparison of pseudo-likelihoods for all $2^{\mathcal{H}}$, where $\mathcal{H}$ is the number of candidate relative positions, leading to high computational cost. Galves, Orlandi and Takahashi (2010) presents estimation of the dependence structure for general families of interacting pairs, but requires multiple observations and considers only the case of binary data (Ising model).

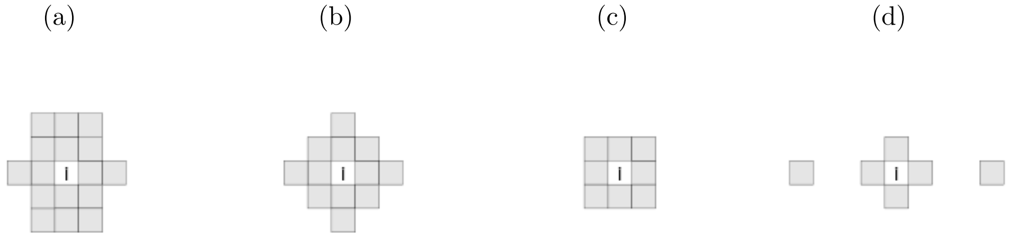


Figure 31 – Nested neighborhood regions and sparse structure.

Figure 31 is an example of how some sets of relative positions allowed cannot be described by nested neighborhood regions. While Figures 31(a) – 31(c) can be seen as a sequence of complexity reduction (nested models), this form of nesting won't be able to include the position $(3, 0)$ without $(2, 0)$, as in 31(d), which is a structure less complex than the others, but is not a sub-model from none of them.

We follow the search algorithm based on interaction maps proposed in Gimel'farb (1999). We consider a sufficiently large window of relative positions and for each of these positions, the distance between normalized sufficient statistics $n_c(d)/n_c$ ($n_c$ is the total number of pairs with relative position $c$) and the expected values of these statistics in an independent random field (triangular distribution) is computed. The interaction structure is then chosen by thresholding these distances. When constructing interaction maps, we will always consider the most general feature function $h(a, b) = b - a$.

Algorithm 2 shows the complete algorithm used to choose an interaction

**Algorithm 2 –** Algorithm for construction of an interaction map and dependence structure selection.

```
begin
    Define a window $W = \{w \in \mathbb{Z}^2 : \max(w) \leqslant c_{max}\}$ of relative positions.;
    for $w \in W$ do
        $d(w) = \sum_{u=-G}^{G} \frac{\left(\frac{n_w(u)}{n_w} - E(u)\right)^2}{E(u)}$;
    end
    Select the set $\hat{C} = \{c : d(c) \geqslant c_{thr}\}$
end
```

structure. $c_{max}$ controls the size of the initial window and $c_{thr}$ is a constant for thresholding, the author suggests using $\bar{d} + k\sigma(d)$, with $\bar{d}$ the mean of the computed distances and $\sigma(d)$ their standard deviation. $E(u)$ is the probability of observing the difference $u \in \{-G, \ldots, G\}$ in a pair of independent discrete uniform (in $\mathcal{Y} = \{0, \ldots, G\}$) random variables

$$E(u) = \frac{(1 + G - |u|)}{(1 + G)^2}.$$

Note that, for positions in the actual dependence structure $C$, the distances $d(w)$ should be large, naturally, because this is related to the (spatial) auto-correlation for those positions. On the other hand, for some $w$ that don't belong in the interaction structure, particularly, those that are linear combination of others in $C$, we observe a decaying of the computed distances, similar to what happens in auto-regressive models.

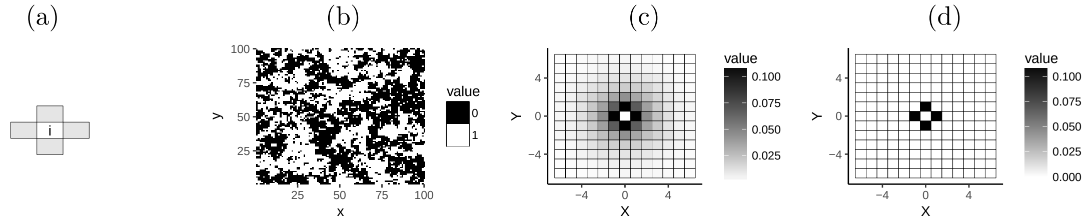


Figure 32 – Interaction Map example. (a) is the dependence structure used, (b) is a simulated image with that structure, (c) and (d) are interaction maps before and after thresholding with $k = 4$.

Figures 32 and 33 are examples of interaction maps. Due to this type of persistent correlation on combinations of conditionally dependent relative positions, interaction maps tend to miss consecutive relative positions with opposite interactions $\theta^0_{c_1} > 0$ and $\theta^0_{c_2} < 0$ with $c_2 = 2c_1$, for example. Figure 34 is an example where the interaction map fails to recover the correct structure due to consecutive interacting positions with opposite signals.

In time series, an important tool for determining the order an auto-regressive process is the partial auto-correlation function, because it removes the decaying correlations

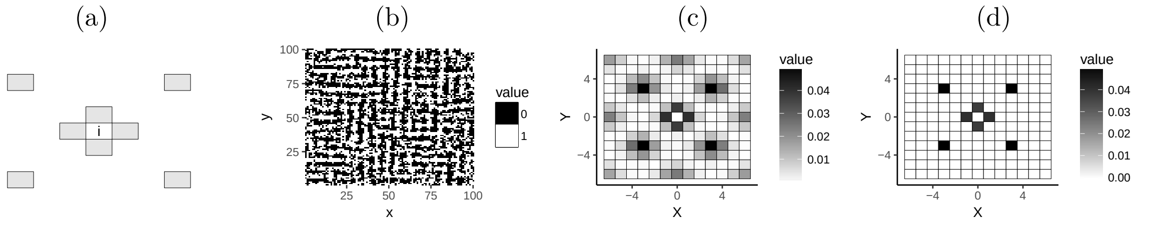


Figure 33 – Interaction Map example. (a) is the dependence structure used, (b) is a simulated image with that structure, (c) and (d) are interaction maps before and after thresholding with $k = 2.5$.

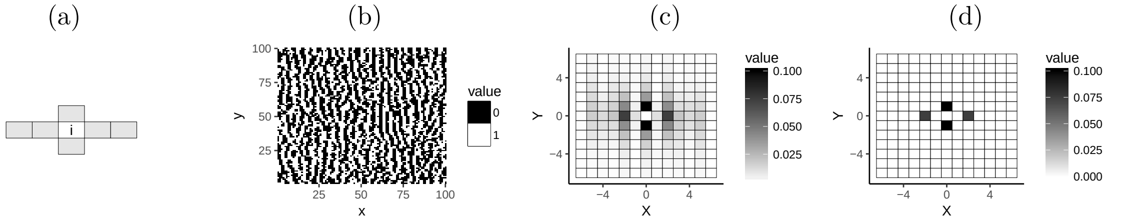


Figure 34 – Interaction Map example. (a) is the dependence structure used, (b) is a simulated image with that structure, (c) and (d) are interaction maps before and after thresholding with $k = 2.5$.

from previous lags. Since the distance computed for constructing interaction maps is directly related to the (absolute) correlation of pixels in that relative position, we can naturally construct an auto-correlation map that is very similar to the interaction map proposed. On the other hand, we can construct partial auto-correlation maps that remove the correlation from a subset of positions iteratively. Therefore, we propose the following Algorithm 3 to heuristically choose the interaction structure, depending a constant thresholding $v$.

Our experiments suggests that Algorithm 3 is able to correctly capture dependence structures like in Figure 34 and is less sensible to the choice of the arbitrary constant. We chose to use $v = 0.08$ because it worked for most cases, while no choice of $k$ could consistently lead to correct structures when using Algorithm 2. While there are no theoretical results to show that these methods work, Gimel'farb (1996) have achieved acceptable results using interaction maps, and we will use the proposed algorithm for the slight advantage that can be observed empirically.

**Algorithm 3 –** Algorithm for choosing interaction structure based on conditional auto-correlation maps.

```
begin
    Define a window $W = \{w \in \mathbb{Z}^2 : \max(w) \leqslant c_{max}\}$ of relative positions;
    Start with empty structure $\hat{C} = \varnothing$;
    for $w \in W$ do
        $\rho(w) = |cor(x_i - P_{\hat{C}}(x_i), x_{i+w} - P_{\hat{C}}(x_{i+w}))|$;
        where $P_C(x_i)$ is the projection of $x_i$ on the space spanned by columns
          $x_{i+c}, c \in C$.
    end
    Select the relative position with highest absolute auto-correlation $c$.;
    if $\rho(c) \geqslant v$ then
        $\hat{C} \leftarrow \hat{C} \cup c$.;
        Return to auto-correlations computation.
    else
        Select $\hat{C}$ as the interaction structure.
    end
end
```

# 4 Hidden Markov Random Fields

While MRFs have been successfully used to model textures in images with a moderate number of colors, in recent years, the quality of digital images have increased considerably and one consequence is a more dense color scale. For example, if we take a 256 gray levels image ($G = 255$), the number of parameters in the previous models would increase drastically and the relationships between colors (i.e. the feature function) could become a lot more complex. In cases like that or our transformation of the original 3-channel color values to a $[0, 1]$ interval, it is appropriate to model pixels as continuous variables, but we also want to preserve the texture characteristics from MRFs.

Hidden Markov Random Fields (HMRFs) are an extension of the idea of hidden Markov chains. In many problems, an underlying process with local dependence characteristics exists but it cannot be observed directly. Instead, we observe a random variable which distribution depends on that hidden process.

Let $\mathbf{T} = \{T_i, i \in \mathcal{J}\}$ be an observed field and $\mathbf{X} = \{X_i, i \in \mathcal{J}\}$ the underlying random field, for some set of indexes $\mathcal{J}$. Because we need knowledge about $\mathbf{X}$ to estimate for $f(\mathbf{t}|\mathbf{X})$ and we need information about $f(\mathbf{t}|\mathbf{X})$ to obtain estimates of $\mathbf{X}$, likelihood optimization is a challenge, specially because, just like in hidden Markov chains, the observed field $\mathbf{T}$ loses the local dependence property.

A random field $\mathbf{T}$ is said a Gaussian hidden Markov random field if $\mathbf{T}|\mathbf{X}$ is Normal distributed. Particularly, we will also assume that, $T_i|\mathbf{X}$ is independent from $T_j|\mathbf{X}$ if $i \neq j$ and the distribution of $T_i|\mathbf{X}$ only depends on $X_i$. We can then write

$$f(\mathbf{t}|\mathbf{X} = \mathbf{x}) = \prod_{i \in \mathcal{J}} \frac{1}{\sqrt{2\pi\sigma^2_{x_i}}} \exp\left(\frac{-(t_i - \mu_{x_i})^2}{2\sigma^2_{x_i}}\right), \tag{4.1}$$

what leads to the marginal distribution of $\mathbf{T}$

$$f(\mathbf{t}) = \sum_{\mathbf{x} \in \mathcal{X}} \prod_{i \in \mathcal{J}} \frac{1}{\sqrt{2\pi\sigma^2_{x_i}}} \exp\left(\frac{-(t_i - \mu_{x_i})^2}{2\sigma^2_{x_i}}\right) P(\mathbf{X} = \mathbf{x}). \tag{4.2}$$

In Equation (4.2) we see that, marginally, $\mathbf{T}$ is a mixture of multivariate Gaussian distributions, but instead each class in each index having the same probability in the whole set $\mathcal{J}$, the probability of the underlying variables is measured for a complete configuration $\mathbf{x}$, thus, falling under the same type of problem from Markov random fields with an exponential number of cases to sum in order to evaluate the likelihood function.

## 4.1 An Expectation-Maximization algorithm for Gaussian Hidden Markov Random Fields

The EM algorithm is an iterative method for finding maximum-likelihood estimates in problems involving unobserved variables due to the often intractable form of the likelihood function. Let $\theta$ be a vector of parameters, $\mathbf{T} = \mathbf{t}$ observed and $\mathbf{X}$ latent variables. Maximum-likelihood estimation via EM-algorithm consists of finding the value $\hat{\theta}$ that satisfies

$$\hat{\theta} = \arg\max_{\theta} L(\theta, \mathbf{t}) = \arg\max_{\theta} \int f_\theta(\mathbf{t}, \mathbf{x}) d\mathbf{x}, \tag{4.3}$$

by, given an initial value $\theta^{(0)}$, iteratively computing the function $Q(\theta|\theta^{(j)})$ (E-step) as

$$Q(\theta|\theta^{(j)}) = \mathbb{E}_{\mathbf{X}|\mathbf{T}=\mathbf{t},\theta^{(j)}} \left[\log f_\theta(\mathbf{t}, \mathbf{X})\right], \tag{4.4}$$

and updating the solution $\theta^{(j+1)}$ (M-step) as

$$\theta^{(j+1)} = \arg\max_{\theta} Q(\theta|\theta^{(j)}). \tag{4.5}$$

The sequence $\{\theta^{(j)}, i = 0, 1, 2, \dots\}$ converges to the maximum-likelihood estimator $\hat{\theta}$ under some conditions. An overview on the convergence and other properties of this method can be found at Dempster, Laird and Rubin (1977).

Consider a Gaussian HMRF model, with known (hyper-)parameters for the underlying Markov random field, let $\theta = \{\mu_k, \sigma_k^2, k = 0, 1, ..., G\}$ be the vector of parameters with $t_i \in \{0, 1, ..., G\}, \forall i \in \mathcal{J}$ and denote $N_k(w)$ the density of a Normal distribution with mean $\mu_k$ and variance $\sigma_k^2$ evaluated at the point $w$ and $N_k^{(j)}(w)$ the same with parameters $\mu_k^{(j)}$ and $\sigma_k^{2(j)}$. In this scenario, the joint distribution of $\mathbf{T}$ and $\mathbf{X}$ (also called the complete likelihood) is given by

$$f_\theta(\mathbf{t}, \mathbf{x}) = P(\mathbf{X} = \mathbf{x}) \prod_{i \in \mathcal{J}} N_{x_i}(t_i), \tag{4.6}$$

what makes the $Q$ function intractable due to the demand of summing over every possible configuration to compute the expected value, as mentioned in previous section

$$Q(\theta|\theta^{(j)}) = \sum_{\mathbf{x} \in \mathcal{X}} P(\mathbf{X} = \mathbf{x}|\mathbf{T} = \mathbf{t}, \theta^{(\mathbf{j})}) \log \left[P(\mathbf{X} = \mathbf{x}) \prod_{i \in \mathcal{J}} N_{x_i}(t_i)\right]. \tag{4.7}$$

We use the version of EM-algorithm presented in Zhang, Brady and Smith (2001) which maximizes pseudo-likelihood instead, to obtain a tractable version of the $Q$ function by substituting $P(\mathbf{X} = \mathbf{x})$ by $\prod_{i \in \mathcal{J}} P(X_i = x_i|X_{-i})$ in Equation (4.7),

$$Q^*(\theta|\theta^{(j)}) = \sum_{i \in \mathcal{J}} \sum_{k=0}^{G} P(X_i = k|T_i = t_i, X_{-i}, \theta^{(j)}) \left[\log P(X_i = k|X_{-i}) + \log N_k(t_i)\right] \tag{4.8}$$

$$= \sum_{i \in \mathcal{J}} \sum_{k=0}^{G} P(X_i = k | T_i = t_i, X_{\mathcal{N}_i}, \theta^{(j)}) \left[ \log P(X_i = k | X_{\mathcal{N}_i}) - \frac{1}{2} \left( \log \sigma_k^2 + \log 2\pi \right) - \frac{(t_i - \mu_k)^2}{2\sigma_k^2} \right].$$

From Equation (4.8), $P(X_i = k | T_i = t_i, X_{\mathcal{N}_i}, \theta^{(j)})$ doesn't depend on $\theta$, but varies depending on the conditioning configuration $X_{\mathcal{N}_i}$. Therefore, to perform the M-step, we first need to find the configuration $\hat{\mathbf{x}}^{(j)}$ which maximizes the conditional probabilities (which is the same as the maximum a posteriori estimate of $\mathbf{X}$). Once $\hat{\mathbf{x}}^{(j)}$ is computed, the M-step iteration has closed form as

$$\mu_k^{(j+1)} = \frac{\sum_{i \in \mathcal{J}} P(X_i = k | T_i = t_i, X_{\mathcal{N}_i} = \hat{x}_{\mathcal{N}_i}^{(j)}, \theta^{(j)}) t_i}{\sum_{i \in \mathcal{J}} P(X_i = k | T_i = t_i, X_{\mathcal{N}_i} = \hat{x}_{\mathcal{N}_i}^{(j)}, \theta^{(j)})}, \tag{4.9}$$

$$\sigma_k^{2(j+1)} = \frac{\sum_{i \in \mathcal{J}} P(X_i = k | T_i = t_i, X_{\mathcal{N}_i} = \hat{x}_{\mathcal{N}_i}^{(j)}, \theta^{(j)}) (t_i - \mu_k^{(j+1)})^2}{\sum_{i \in \mathcal{J}} P(X_i = k | T_i = t_i, X_{\mathcal{N}_i} = \hat{x}_{\mathcal{N}_i}^{(j)}, \theta^{(j)})}. \tag{4.10}$$

Note that, from results in Chapter 3, $P(X_i = k | X_{\mathcal{N}_i} = x_{\mathcal{N}_i}) \propto \exp(g(k, x_{\mathcal{N}_i}))$ where $g$ is some functions that depend on the parameters of the underlying Markov random field, thus, except for a (tractable) normalizing constant, we have

$$P(X_i = k | T_i = t_i, X_{\mathcal{N}_i} = x_{\mathcal{N}_i}, \theta^{(j)}) \propto \exp(g(k, x_{\mathcal{N}_i})) N_k^{(j)}(t_i). \tag{4.11}$$

To obtain the MAP estimate of $\mathbf{X}$, Zhang, Brady and Smith (2001) use the Iterated Conditional Modes algorithm from Besag (1986). This algorithm consists of starting with a configuration $\mathbf{x}^0$ and perform pixel-wise updates

$$x_i \leftarrow \arg\max_k P(X_i = k | T_i = t_i, X_{\mathcal{N}_i} = x_{\mathcal{N}_i}, \theta)$$

for the complete sequence of indexes $i$. After going through the whole image several times, the current set of labels $\mathbf{t}$ coincides with the MAP estimate $\hat{\mathbf{x}}$ which satisfies

$$\hat{\mathbf{x}} = \arg\max_{\hat{\mathbf{x}}} P(\mathbf{X} = \mathbf{t} | \mathbf{T} = \mathbf{t}, \theta).$$

We select the initial labels $\mathbf{x}^0$ and parameter values $\theta^{(0)}$ as the estimates from an usual EM-algorithm for independent mixture of Gaussian distributions.

The complete algorithm is shown below. Convergence criterion adopted usually depends on the distance between consecutive parameter vector estimates and number of iterations.

## 4.2 Hidden Markov Random Fields with covariates

A fixed effect can be included to HMRFs by adding an extra $\mathbf{w}_i\beta$ term to conditional expected value in each point, where $\mathbf{w}_i$ is a $m$-dimensional row vector of

**Algorithm 4** – Complete EM-algorithm for Gaussian HMRFs estimation.

```
begin
    Set initial values for $\theta^{(0)}$ and initial labels $\hat{\mathbf{x}}^{(0)}$;
    set $j \leftarrow 0$;
    while Convergence criterion not met do
        Update $\hat{\mathbf{x}}^{(j)}$ using the Iterated Conditional Modes algorithm, starting with
          the previous estimated labels $\hat{\mathbf{x}}^{(j-1)}$;
        Update $\theta^{(j+1)}$ according to (4.9) and (4.10).;
        $j \leftarrow j + 1$;
    end
    $\theta^{(j)}$ and $\hat{\mathbf{x}}^{(j)}$ are estimates for the vector of parameters and the underlying field
      labels, respectively.
end
```

covariates and $\beta$ is vector of coefficients. Since the only difference from the previous model presented is that $\mathbb{E}(T_i|X_i = x_i) = \mu_{x_i} + \mathbf{w}_i\beta$, rather than $\mu_{x_i}$, conditional distribution can be written as

$$f(\mathbf{t}|\mathbf{X} = \mathbf{x}) = \prod_{i\in\mathcal{J}} \frac{1}{\sqrt{2\pi\sigma^2_{x_i}}} \exp\left(\frac{-(t_i - \mu_{x_i} - \mathbf{w}_i\beta)^2}{2\sigma^2_{x_i}}\right). \tag{4.12}$$

An EM algorithm very similar to the case with no covariates can be done, concatenating $\beta$ to the vector of parameters $\theta$, with two differences: First, for the $E$ step, for $Q^*(\theta|\theta^{(j)})$ we get

$$\sum_{i\in\mathcal{J}}\sum_{k=0}^{G} P(X_i = k|T_i = t_i, X_{\mathcal{N}_i}, \theta^{(j)}) \left[\log P(X_i = k|X_{\mathcal{N}_i}) - \frac{1}{2}\left(\log\sigma^2_k + \log 2\pi\right) - \frac{(t_i - \mu_k - \mathbf{w}_i\beta)^2}{2\sigma^2_k}\right], \tag{4.13}$$

with

$$P(X_i = k|T_i = t_i, X_{\mathcal{N}_i} = x_{\mathcal{N}_i}, \theta^{(j)}) \propto \exp(g(k, x_{\mathcal{N}_i}))N^{(j)}_k(t_i - \mathbf{w}_i\beta^{(j)}), \tag{4.14}$$

and for the M step, we also need to maximize $Q^*(\theta|\theta^{(j)})$ with respect to $\beta$, requiring a numerical optimization instead of a closed form estimator as in previous case. We use a slightly modified version of the EM algorithm for HMRF with a "bias field" from Zhang, Brady and Smith (2001), and update $\theta$ by a sequence of component-wise optimizations, i.e., we fix $\mu$'s and $\sigma$'s most recent values to update $\beta$, then use this new value to compute the remaining estimates.

$$\beta^{(j+1)} = \left[\sum_{i\in\mathcal{J}} \mathbf{w}_i^T \left(\sum_{k=0}^{G} \frac{P(X_i = k|t_i, \hat{x}^{(j)}_{\mathcal{N}_i})}{\sigma_k^{2(j)}}\right)\mathbf{w}_i\right]^{-1} \left[\sum_{i\in\mathcal{J}} \mathbf{w}_i^T \left(\sum_{k=0}^{G} \frac{P(X_i = k|t_i, \hat{x}^{(j)}_{\mathcal{N}_i}, \theta^{(j)})}{\sigma_k^{2(j)}}(t_i - \mu^{(j)}_k)\right)\right] \tag{4.15}$$

$$\mu^{(j+1)}_k = \frac{\sum_{i\in\mathcal{J}} P(X_i = k|T_i = t_i, X_{\mathcal{N}_i} = \hat{x}^{(j)}_{\mathcal{N}_i}, \theta^{(j)})(t_i - \mathbf{w}_i\beta^{(j+1)})}{\sum_{i\in\mathcal{J}} P(X_i = k|T_i = t_i, X_{\mathcal{N}_i} = \hat{x}^{(j)}_{\mathcal{N}_i}, \theta^{(j)})}, \tag{4.16}$$

$$\sigma_k^{2(j+1)} = \frac{\sum_{i\in\mathcal{J}} P(X_i = k|T_i = t_i, X_{\mathcal{N}_i} = \hat{t}_{\mathcal{N}_i}^{(j)}, \theta^{(j)})(t_i - \mu_k^{(j+1)} - \mathbf{w}_i\beta^{(j+1)})^2}{\sum_{i\in\mathcal{J}} P(X_i = k|T_i = t_i, X_{\mathcal{N}_i} = \hat{t}_{\mathcal{N}_i}^{(j)}, \theta^{(j)})}. \tag{4.17}$$

The complete EM algorithm is the same as Algorithm 4, with estimates updates as in (4.15), (4.16) and (4.17) instead. Note that if the covariates include an intercept (any column of $\mathbf{X}$ is a constant), there would be an identifiability problem, i.e., adding a value to the coefficient associated with this intercept and subtracting it from each of the random effect means $\mu_k$ could result in the same likelihood and, therefore, the maximization step would not have a unique solution. To avoid identifiability problems, when fitting HMRF's with covariates, we add the restriction $\sum_k \mu_k^{[0]} = 0$, and due to the sequential updates, (4.16) results in estimates with the same restriction at each step.

## 4.3 Bayesian estimation for Gaussian Hidden Markov Random Fields

A second approach for estimating labels and parameters on hidden Markov random fields is to treat $\theta$ as random variables with prior distribution $\pi(\theta)$ and sample the interest quantities from their a posteriori distribution $\pi(\theta, \mathbf{t}|\mathbf{t})$, which is given by

$$\pi(\theta, \mathbf{t}|\mathbf{t}) \propto P(\mathbf{X} = \mathbf{t})\pi(\theta) \prod_{i\in\mathcal{J}} \frac{1}{\sqrt{\sigma_{x_i}^2}} \exp\left(\frac{(t_i - \mu_{x_i})^2}{2\sigma_{x_i}^2}\right), \tag{4.18}$$

If we that consider the components of $\theta$ are independent a priori and they are, for example, uniformly distributed $\pi(\mu_k) = \frac{1}{2l}\mathbb{1}_{\{-l\leqslant\mu_k\leqslant l\}}$ and $\pi(\sigma_k^2) = \frac{1}{L}\mathbb{1}_{\{0\leqslant\sigma_k^2\leqslant L\}}$ for some positive constants $l$ and $L$, we can obtain very simple forms for the full conditional distribution of each component in $\theta$ and $\mathbf{t}$ as follows

$$\pi(t_i|\mathbf{t}, \theta, X_{-i} = t_{-i}) \propto P(X_i = t_i|X_{-i} = t_{-i}) \frac{1}{\sqrt{\sigma_{x_i}^2}} \exp -\left(\frac{(t_i - \mu_{x_i})^2}{2\sigma_{x_i}^2}\right), \tag{4.19}$$

$$\pi(\mu_k|\mathbf{X} = \mathbf{x}, \mathbf{t}, \sigma_k^2) \propto \exp\left(\frac{(\bar{y}_k - \mu_k)^2}{2\sigma_k^2/n_k}\right)\mathbb{1}_{\{-l\leqslant\mu_k\leqslant l\}}, \tag{4.20}$$

$$\pi(\sigma_k^2|\mathbf{X} = \mathbf{x}, \mathbf{t}, \mu_k) \propto \sigma_k^{2(-n_k/2)} \exp\left(\frac{-\sum_{i:t_i=k}(t_i - \mu_k)^2}{2\sigma_k^2/n_k}\right)\mathbb{1}_{\{0\leqslant\sigma_k^2\leqslant L\}}, \tag{4.21}$$

where $n_k = \sum_{i\in\mathcal{J}} \mathbb{1}_{\{x_i=k\}}$ and $\bar{t}_k = \sum_{i:x_i=k} t_i/n_k$.

In each of these full conditionals we can recognize kernels of known distributions with $\pi(t_i|\cdots)$ being a multinomial with weights as specified in (4.19), $\pi(\mu_k)$ a truncated ($[-l, l]$) Normal distribution with mean $\bar{y}_k$ and variance $\sigma_k^2/n_k$ and $\pi(\sigma_k^2|\cdots)$ as a truncated

($[0, L]$) inverse gamma distribution with shape parameter $\dfrac{n_k - 2}{2}$ and scale parameter $\dfrac{\sum_{i:t_i=k}(t_i - \mu_k)^2}{2}$.

Since we can sample from these known conditional distributions with little computational cost, a Gibbs Sampler scheme appropriate to sample from (4.18) is presented in Algorithm 5. In this work, we use Uniform a priori distributions, but similar approach can be done, for example, by using conjugate priors or any other distribution that imply the full conditionals to have known forms.

For the Gibbs sampling scheme, we sample initial $\theta$ from its a priori distribution and $\mathbf{t}^0$ from independent uniform multinomial distributions. $\mu_k$ and $\sigma_k^2$ are estimated by either mean or median of the a posteriori sample. On the other hand, for the underlying field, these statistics do not have the same meaning, since the labels may not have a numerical interpretation, hindering the use of mean or median. We adopt a similar approach to the EM-algorithm version and estimate $\mathbf{t}$ as the maximum a posteriori value, when conditioning on the estimated parameters.

**Algorithm 5** – Gibbs Sampling scheme for (4.18).

```
begin
    Set initial values for $\theta^{(0)}$ and initial labels $\mathbf{t}^{(0)}$;
    set $j \leftarrow 1$;
    while Convergence criterion not met do
        Sample $\mathbf{t}^{(j)}$ component-wise by sampling from (4.19);
        Sample each $\mu_k^{(j)}$ according to $\pi(\mu_k|\mathbf{X} = \mathbf{x}^{(j)}, \mathbf{t}, \sigma_k^{2(j-1)})$;
        Sample each $\sigma_k^{2(j)}$ according to $\pi(\sigma_k^2|\mathbf{X} = \mathbf{x}^{(j)}, \mathbf{t}, \mu_k^{(j)})$;
        $j \leftarrow j + 1$;
    end
    Burn-in the first $B$ samples to suppress the effect of choosing the initial
      configuration and then select $H$ spaced samples to obtain low correlation.
end
```

## 4.4 Simulation Example

In order to illustrate and compare the described methods of estimation for HMRFs, we consider a binary Markov Random Field with pairwise interactions and a moderate dependence structure as presented in Table 12. We define two Gaussian mixture densities, with the same means, but different variances, as specified in Table 11. Figure 35 shows how the variance increase implies the resulting density losing its two modes characteristic.

We sample 10 binary fields $\mathbf{X}$ from the specified Markov Random Field with

| Density | $k$ | 0 | 1 |
|---|---|---|---|
| 1 | $\mu_k$ | $-1.0$ | $1.0$ |
|  | $\sigma_k$ | $0.7$ | $0.6$ |
| 2 | $\mu_k$ | $-1.0$ | $1.0$ |
|  | $\sigma_k$ | $1.4$ | $1.2$ |

Table 11 – Parameters for Gaussian mixtures.

(a)

| $c$ | $\theta_c^{(0)}$ | $\theta_c^{(1)}$ |
|---|---|---|
| Single | 0.0 | 0.0 |
| (1,0) | 0.6 | -0.6 |
| (0,1) | 0.6 | -0.6 |
| (3,3) | -0.3 | 0.3 |

Table 12 – Parameters specification for the underlying field of HMRFs simulations.

(b)

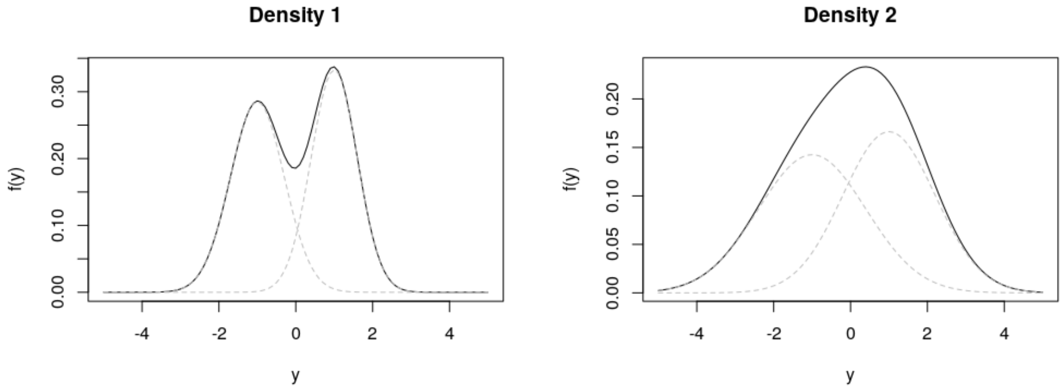


Figure 35 – Gaussian mixtures densities with parameters from Table 11. Dashed gray lines represent individual components of the mixture.

dimensions 150 × 100 and for each of these binary fields, we simulate pixels independently according to parameters of a density in Table 11 given its class $f(t_i|X_i)$, resulting in a sample from a Hidden Markov Random Field $\mathbf{T}$. For each $\mathbf{T}$, we estimate $\mu_k$ and $\sigma_k$, $k = 0, 1$ and class labels via the EM algorithm presented, an EM algorithm considering an independent field and Gibbs sampler with Uniform a priori distributions (1000 a posteriori samples after burn-in and thinning). Figure 36 shows examples of fields sampled for each mixture density.

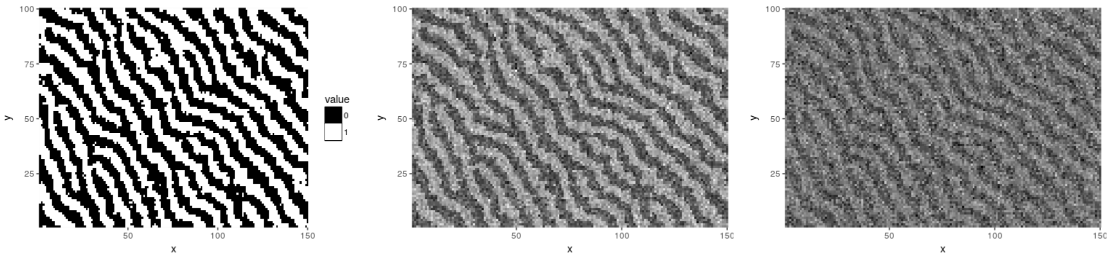

(a) (b) (c)

Figure 36 – An example of simulated HMRF: (a) is the underlying field, (b) was sampled with Density 1 and (c) with Density 2.

| Density | Estimate | Real | EM | ind. EM | Gibbs |
|---|---|---|---|---|---|
| 1 | $\hat{\mu}_0$ | -1 | -1.005(0.0106) | -1.002(0.0156) | -1.004(0.0110) |
| | $\hat{\mu}_1$ | 1 | 0.9944(0.0069) | 0.9970(0.0078) | 0.9954(0.0070) |
| | $\hat{\sigma}_0$ | 0.7 | 0.6968(0.0072) | 0.6988(0.0098) | 0.6975(0.0075) |
| | $\hat{\sigma}_1$ | 0.6 | 0.6001(0.0059)) | 0.5988(0.0053) | 0.5997(0.0058) |
| | Misclassification | – | 0.019 | 0.061 | 0.025 |
| 2 | $\hat{\mu}_0$ | -1 | -1.037(0.0130) | -1.005(0.0203) | -0.9970(0.0151) |
| | $\hat{\mu}_1$ | 1 | 0.9897(0.0179) | 1.002(0.01502) | 0.9962(0.0136) |
| | $\hat{\sigma}_0$ | 1.4 | 1.382(0.0098) | 1.395(0.0115) | 1.400(0.0093) |
| | $\hat{\sigma}_1$ | 1.2 | 1.202(0.0119) | 1.199(0.0157) | 1.204(0.0116) |
| | Misclassification | – | 0.093 | 0.220 | 0.073 |

Table 13 – Results for 10 simulations of each mixture density. Values in each cell represent the average of estimates for a particular combination of parameter and method and their sample deviation in parenthesis.

Resulting estimates are shown in Table 13. All three estimation methods had a very similar performance for $\mu_k$ and $\sigma_k$, $k = 0, 1$, but for the estimation of class labels, the EM algorithm considering an independent mixture had the highest misclassification ratio in both cases, what was expected since it ignores all spatial information. Figures 37 and 38 shows estimated labels for each method in a simulation.

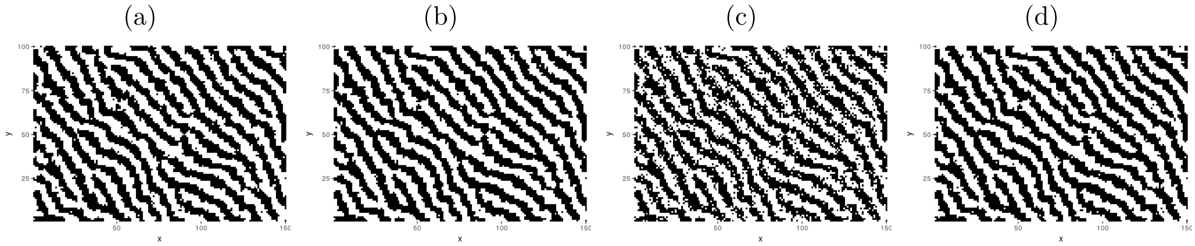


Figure 37 – Estimated class labels for a simulation using Density 1. (a) Real labels, (b) EM, (c) Independent EM, (d) Gibbs Sampler.

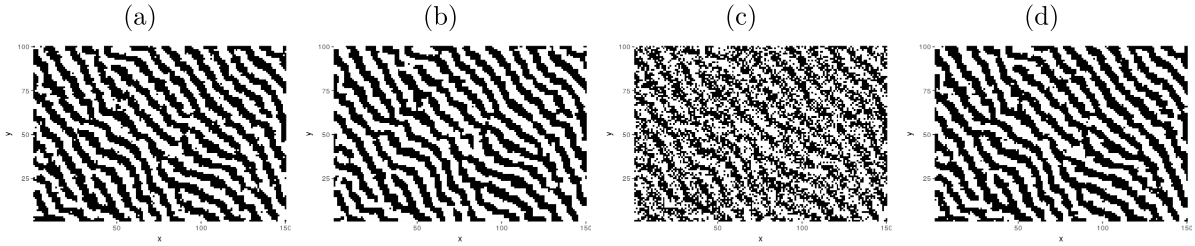


Figure 38 – Estimated class labels for a simulation using Density 2. (a) Real labels, (b) EM, (c) Independent EM, (d) Gibbs Sampler.

# 5 Analyzing dyed textile images

We analyzed the fungal dyed textile images by applying sequentially some of the presented methods. For each of the gray-scale images shown in Figure 39, we start by extracting the low-frequency signal using Penalized Least-Squares methods and Fourier basis from Chapter 2. The penalty function used was $p_i(\beta_i) = \lambda_{b_i} \mathbb{1}_{\{\beta_i \neq 0\}}$, which results in a hard-thresholding of the OLS estimates solution. We selected penalty constants $(\lambda_{b_i})$ to have thresholding constants of the form $b_i = \lambda \exp\left(\frac{\max(n_i, m_i)}{\alpha}\right)$. We compared the results from several choices of $\lambda$, which controls the base thresholding value related to scale of the gray levels and $\alpha$ that controls how the penalty relates to the frequency. $\lambda = 1$ and $\alpha = 2$, as in the example from Figure 15, produced appropriate results and were selected for extracting the low-frequency signal, denoted as $Y_{low}$. Resulting images from applying this hard-thresholding to Fourier basis are shown in Figures 40 ($Y_{low}$) and 41 ($Y - Y_{low}$).

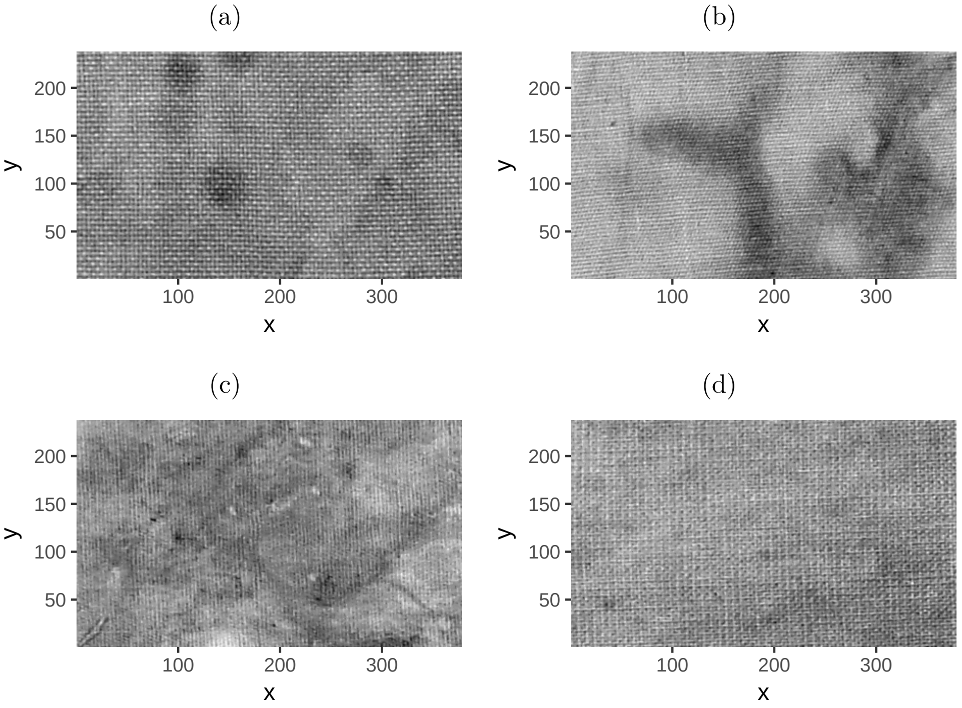


Figure 39 – Gray-scale images.

Due to the general structure of weft, we can identify a small number of main gray-scale levels forming a texture, with small variations in these structures. Due to these characteristics, a Hidden Markov Random Field model should be appropriate to describe the fabric structure seen after removing the color variation resulting from the dyeing process. We use Fourier basis selected by the Penalizes Least-Squares procedure as covariates to fit a Hidden Markov Random Field model that can describe the two main

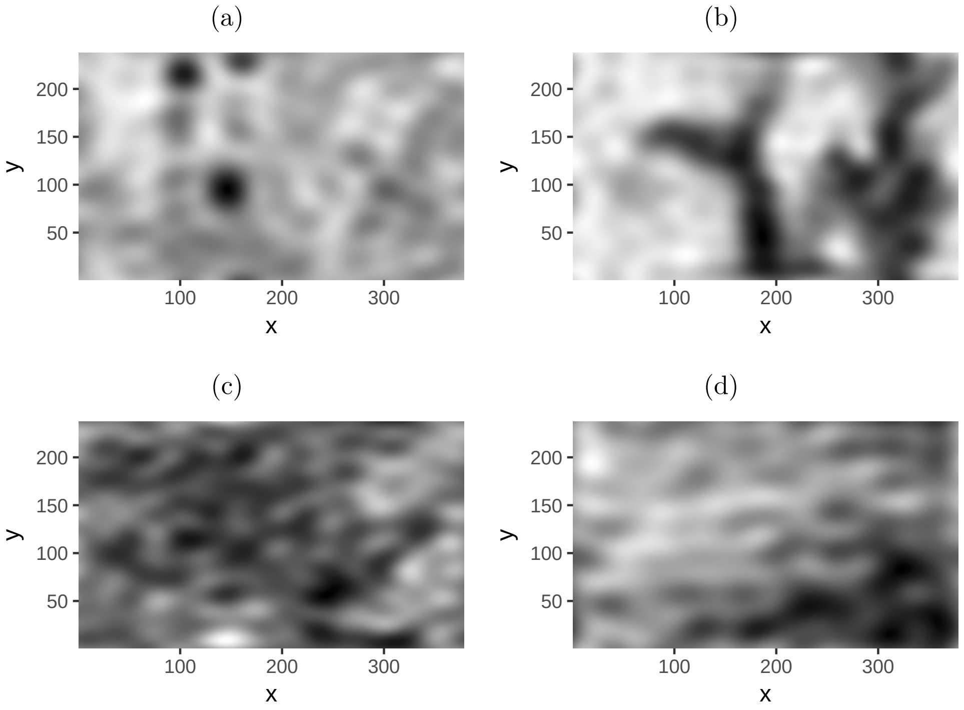


Figure 40 – Low-frequency filtered images $Y_{low}$.

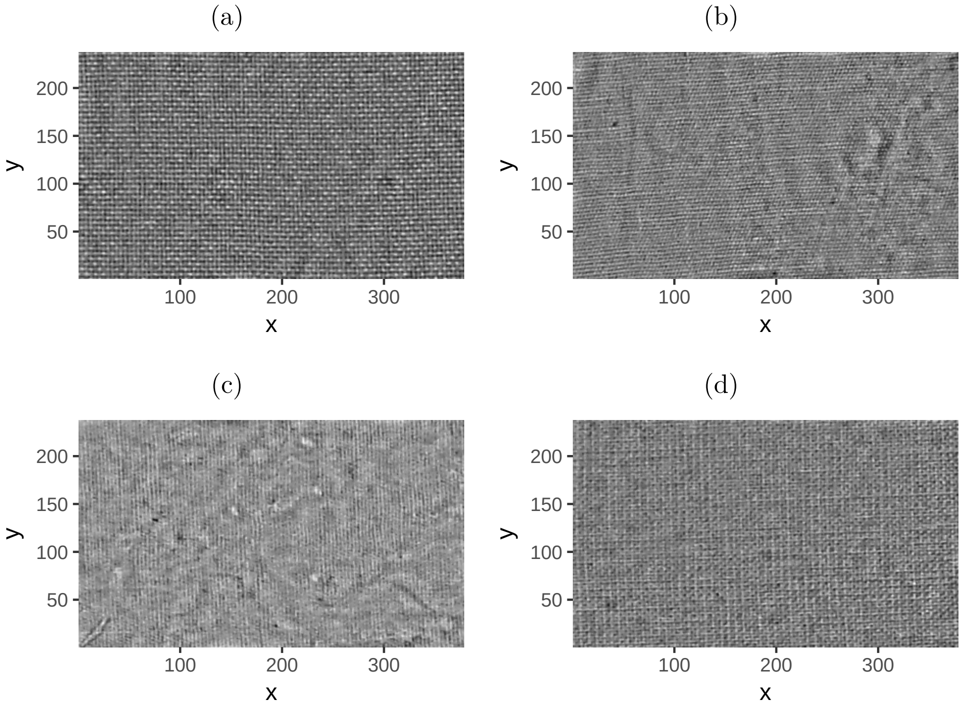


Figure 41 – Remaining signal after low-frequency filtering $Y - Y_{low}$.

parts of the image, texture and dyeing, simultaneously.

The estimation methods for Hidden Markov Random Fields shown in Chapter 4 require knowledge of the random effect distribution $P(\mathbf{X} = \mathbf{x})$, therefore, we first need to estimate these distributions for each texture with some method from Chapter 3. Since we cannot observe the underlying field **x** directly to estimate its distribution, we need a proxy sample of the texture **X**.

We fit an independent Gaussian mixture model to the data (after removing the low-frequency signal) with 5 classes and obtain maximum a posteriori estimates for labels in the underlying field. Since there is a certain symmetry for gray-levels histograms in each image, an odd number of classes is required to fit the data appropriately, and three classes wasn't enough to capture some details of the weft. We use these MAP class labels as a proxy learning sample of the underlying MRF to estimate the parameters for each texture and simulate multiple observations from the estimated distribution to check whether the sampled fields had patterns similar to the training sample. Figure 42 shows the regions of class labels used for the estimation process.

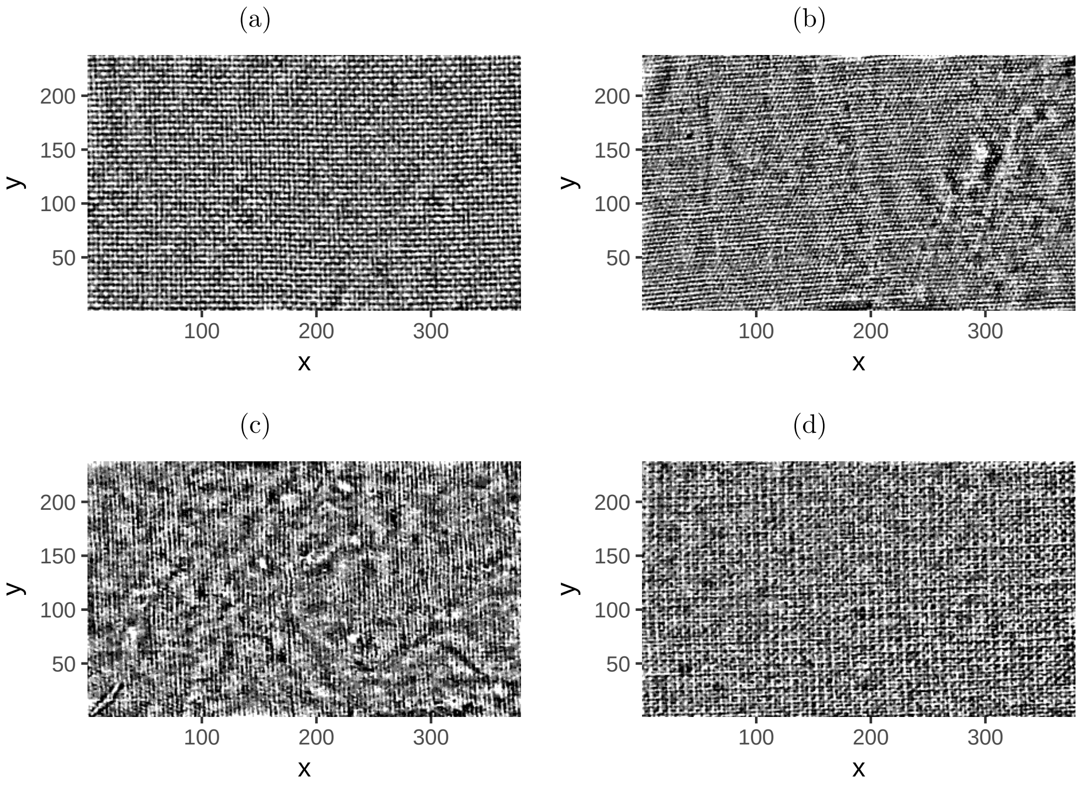


Figure 42 – Proxy samples from the underlying Markov Random Field used in estimation.

For each of these proxy samples, we used both Algorithm 3 and Algorithm 2 with a (partial) auto-correlation thresholding of 0.075 and $k = 3$, respectively, with initial window size 25. While none of the interaction structures could produce simulated samples with similar patterns (after estimating parameters), by gathering relative positions resulting from both algorithms and including nearest-neighbor interaction, similar simulated textures were achieved. Figure 43 shows interaction structures used for each image, 18, 18, 14 and 14 relative positions were selected for images (a) to (d), respectively. Note how the set of relative position reflects features of the texture, for example, the inclination of weft in the image.

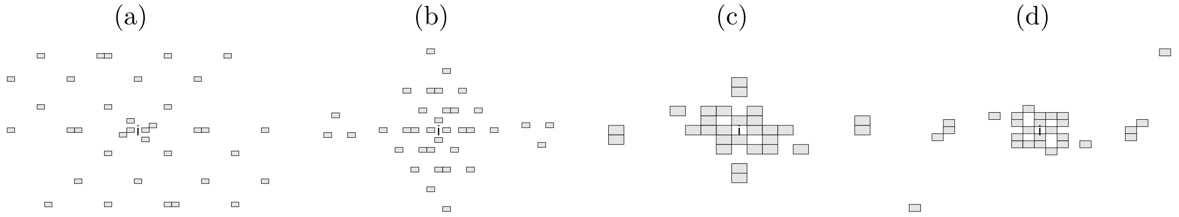


Figure 43 – Selected interaction structures $C$ for each texture. $i$ denotes the origin and gray squares corresponds to positions in $C$.

Given these interaction structures, parameters for each underlying MRF were estimated by using Stochastic Approximation, with feature function $h_2(a, b) = |b - a|$. This method was preferred over Maximum Pseudo-Likelihood, due to the high complexity of interactions, and Monte-Carlo Maximum Likelihood, due to an excessive number of parameters that would require a precise initial guess to estimate normalizing constant ratios with a reasonable number of sample, while stochastic approximation, despiting being a slower algorithm, deals with high complexity models in a more efficient manner. We use $a^{[i]} = \dfrac{1}{n(3 + i/20)}$, leading to small steps and a slow monotone convergence, with $\theta^{[0]} = \mathbf{0}$ (independent field) and 2000 steps. Estimated parameters are shown in Annex A.

With the estimated interaction structures and parameters for the underlying fields, we fitted a Hidden Markov Random Field with previously selected low-frequency Fourier basis as covariates. Initial parameters and class labels were chosen as the estimates from independent Gaussian mixture fitted to obtain proxy learning samples. Table 14 presents estimates for parameters in each class. Figure 44 shows the estimated fixed effect $\mathbf{X}\hat{\beta}$, which, naturally, looks very similar to $Y_{low}$ and Figure 45 are estimated means for each pixel, using MAP estimate for class labels.

| | $k$ | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| (a) | $\mu_k$ | -0.12 | -0.05 | -0.00 | 0.05 | 0.14 |
| | $\sigma_k$ | 0.03 | 0.02 | 0.02 | 0.02 | 0.04 |
| (b) | $\mu_k$ | -0.10 | -0.06 | -0.00 | 0.06 | 0.11 |
| | $\sigma_k$ | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 |
| (c) | $\mu_k$ | -0.09 | -0.05 | 0.00 | 0.04 | 0.08 |
| | $\sigma_k$ | 0.03 | 0.02 | 0.02 | 0.02 | 0.02 |
| (d) | $\mu_k$ | -0.11 | -0.06 | -0.00 | 0.06 | 0.10 |
| | $\sigma_k$ | 0.02 | 0.02 | 0.02 | 0.02 | 0.02 |

Table 14 – Estimated parameters for Hidden Markov Random Field classes.

Finally, with an estimate for the fixed effect that corresponds to dyeing, we compute the color homogeneity index presented for images formed by $\mathbf{W}\hat{\beta}$, which doesn't take into account color variation caused by the weft. Computed indexes for $1 \times 1$ to $20 \times 20$

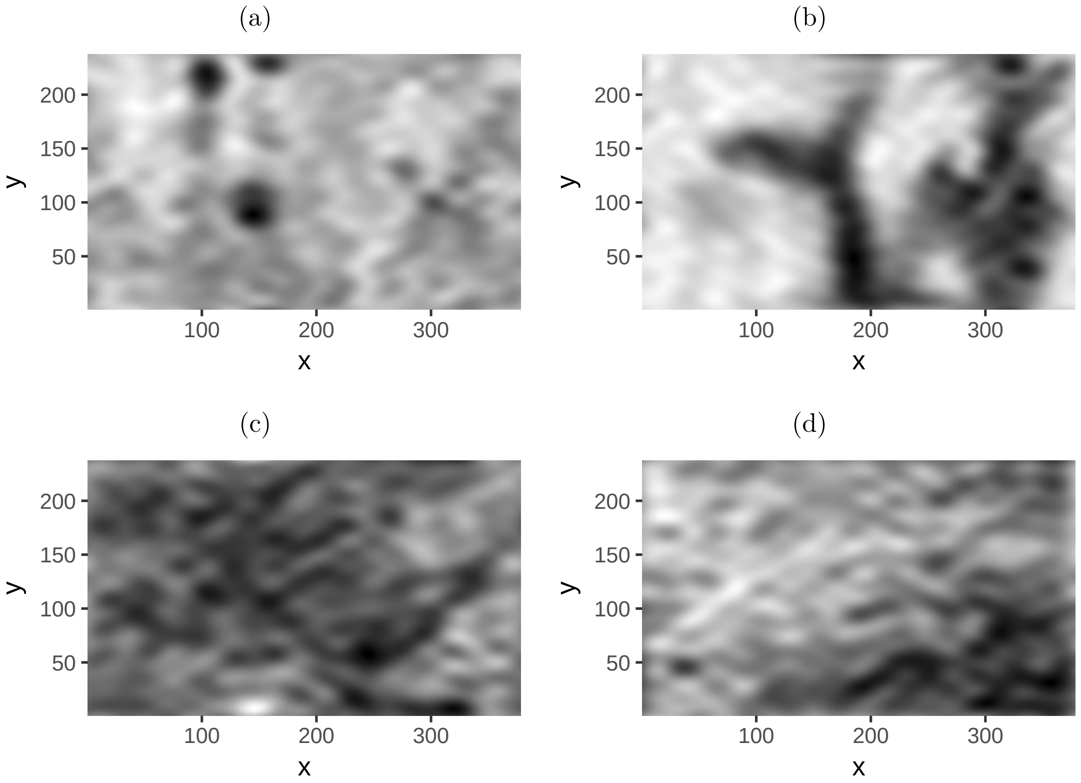


Figure 44 – Estimated fixed effect $\mathbf{X}\hat{\beta}$.

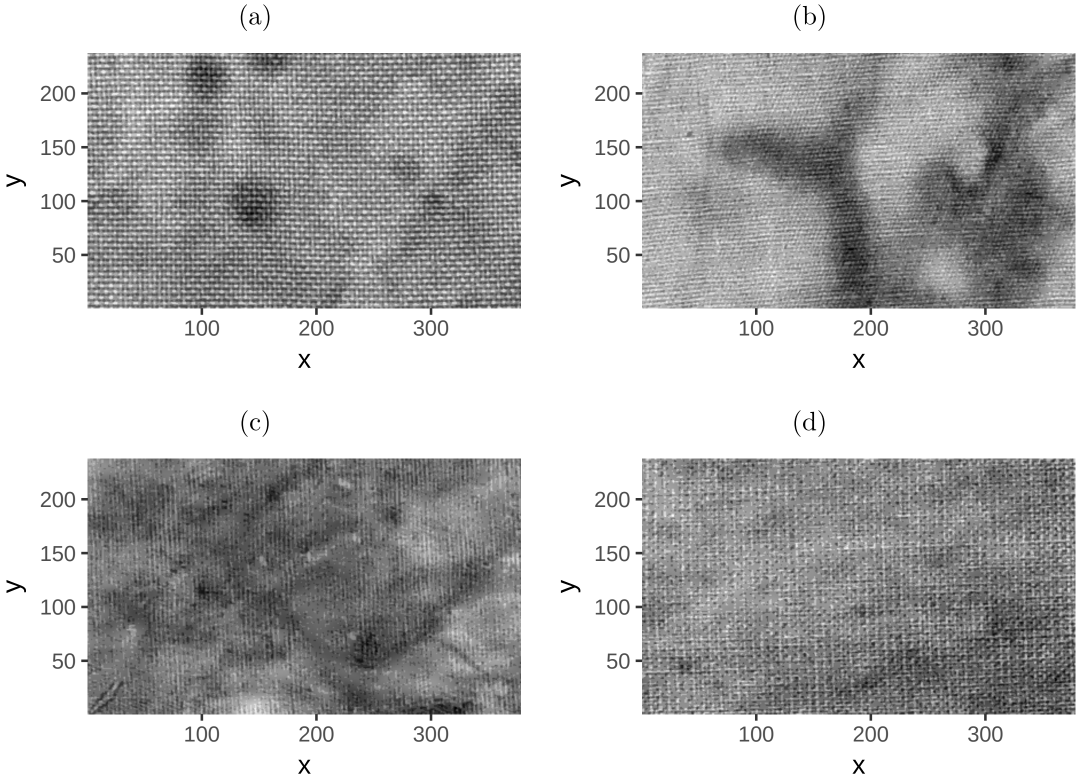


Figure 45 – MAP estimated means $\mathbf{X}\hat{\beta} + \hat{\mu}_{\hat{k}}$.

bins are shown in Figure 46. The same estimation procedure was done for a synthetic dyed fabric image for comparison. While for original images there was an increase of color homogeneity as bin sizes also increase, we see a practically flat relationship between bin size and homogeneity for filtered images.

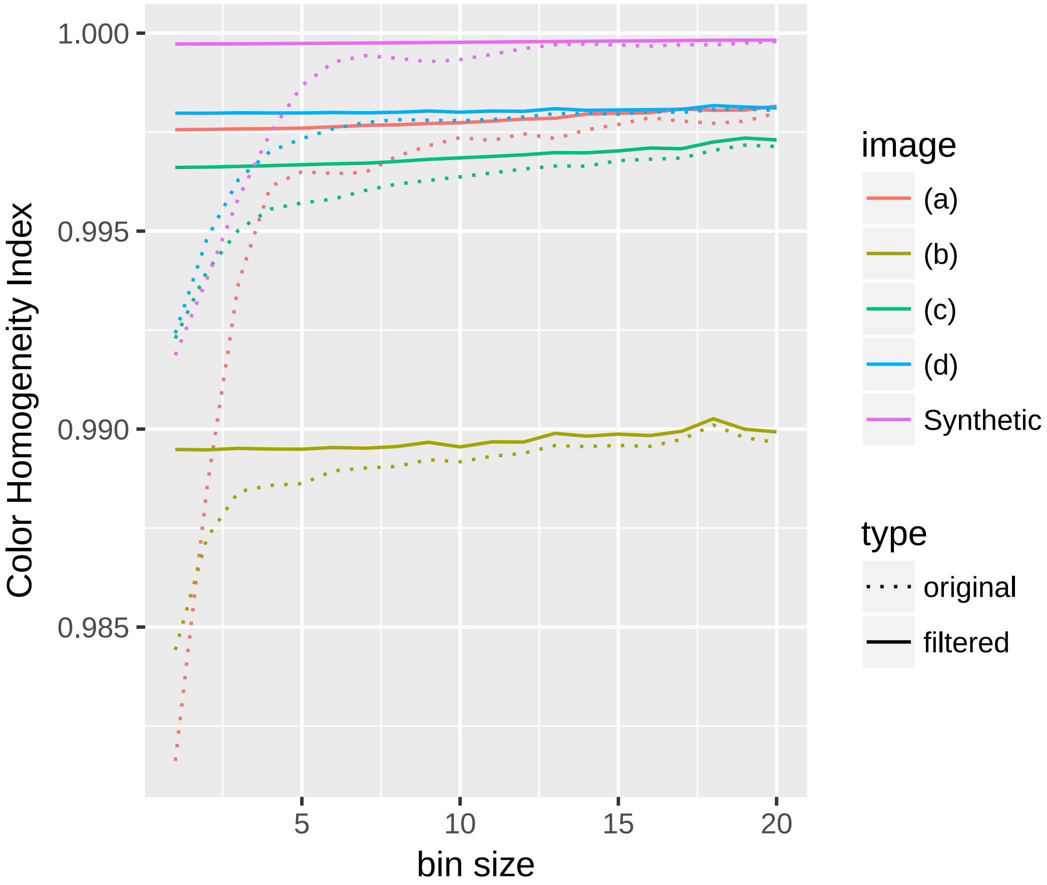


Figure 46 – Color Homogeneity Index for each image, in bin sizes varying from 1 (original resolution) to 20, for original image $Y$ and HMRF filtered image $\mathbf{W}\hat{\beta}$.

# 6 Final Considerations

Markov Random Fields play an important role in probabilistic modeling of some types image data. For most problems, a simple interaction structure like nearest-neighbor is enough to describe dependence, allowing a variety of proposed methods in the literature to solve computational challenges imposed by the unknown normalizing constant from Gibbs distributions. However, this is not the case for complex dependence structures necessary to describe textures, as the high dimension optimization may lead to unreasonable results for finite samples.

These challenges are even bigger when it is not possible to obtain an actual sample field, what is the case for Hidden MRFs. Our strategy of obtaining a proxy sample from estimates of independent Gaussian mixture model produced reasonable results for this type of model.

The negative effect from textures caused by the texture in color homogeneity index was successfully solved by using low-frequency Fourier basis. Simple Penalized Least-Squares fit with specific penalty functions were enough to extract the dyeing effect from images, in fact, results obtained by applying this method were very close to the final estimated fixed effect for HMRF, but they are essentially a numerical feature extraction procedure, as we do not make (or check) any strong assumptions to stochastic characteristics of the data at this step. On the other hand, assuming a probabilistic model describing the data, despite the high complexity and extra computational costs, allows future development of inferential methods, like hypothesis testing.

All source codes used in this work were included in the R package available at https://github.com/Freguglia/GibbsRF (complete documentation is still under construction) and can be installed via *'install_github'* function from *'devtools'* package.

Possible topics for future work include:

- Inferential estimation of the interaction structure for Markov Random Fields.
- Bayesian estimation of parameters for Markov Random Fields.
- Simultaneous estimation of underlying field parameters and mixture densities for Hidden Markov Random Fields.
- Variable selection (for fixed effects) in a HMRF.
- Strategies for defining the number of classes in a HMRF.
- HMRFs with non-gaussian mixture.

# Bibliography

ALEMASKIN, K.; MANAS-ZLOCZOWER, I.; KAUFMAN, M. Entropic analysis of color homogeneity. *Polymer Engineering & Science*, Wiley Online Library, v. 45, n. 7, p. 1031–1038, 2005.

ALEXANDER, E.; POULARIKAS, D. The handbook of formulas and tables for signal processing. *Boca Raton, FL, USA: CRC Press*, v. 73, 1998.

ALIHOSSEINI, F.; JU, K.-S.; LANGO, J.; HAMMOCK, B. D.; SUN, G. Antibacterial colorants: characterization of prodiginines and their applications on textile materials. *Biotechnology progress*, Wiley Online Library, v. 24, n. 3, p. 742–747, 2008.

BARNDORFF-NIELSEN, O. *Information and exponential families in statistical theory.* [S.l.]: John Wiley & Sons, 2014.

BESAG, J. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, JSTOR, p. 192–236, 1974.

______. Statistical analysis of non-lattice data. *The statistician*, JSTOR, p. 179–195, 1975.

______. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, JSTOR, p. 259–302, 1986.

CSISZÁR, I.; TALATA, Z. Consistent estimation of the basic neighborhood of markov random fields. *The Annals of Statistics*, v. 34, n. 1, p. 123–145, 2006.

DEMPSTER, A. P.; LAIRD, N. M.; RUBIN, D. B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, JSTOR, p. 1–38, 1977.

FAN, J.; LI, R. Statistical challenges with high dimensionality: Feature selection in knowledge discovery. *arXiv preprint math/0602133*, 2006.

GALVES, A.; ORLANDI, E.; TAKAHASHI, D. Y. Identifying interacting pairs of sites in infinite range ising models. *Preprint, http://arxiv. org/abs/1006.0272*, 2010.

GEMAN, S.; GEMAN, D. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, IEEE, n. 6, p. 721–741, 1984.

GEYER, C. J. Markov chain monte carlo maximum likelihood. Interface Foundation of North America, 1991.

GEYER, C. J.; THOMPSON, E. A. Constrained monte carlo maximum likelihood for dependent data. *Journal of the Royal Statistical Society. Series B (Methodological)*, JSTOR, p. 657–699, 1992.

GIMEL'FARB, G. L. Texture modeling by multiple pairwise pixel interactions. *IEEE Transactions on pattern analysis and machine intelligence*, IEEE, v. 18, n. 11, p. 1110–1114, 1996.

______. *Image textures and Gibbs random fields.* [S.l.]: Springer Science & Business Media, 1999.

HOERL, A. E.; KENNARD, R. W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, Taylor & Francis Group, v. 12, n. 1, p. 55–67, 1970.

KASIRI, M. B.; SAFAPOUR, S. Natural dyes and antimicrobials for green treatment of textiles. *Environmental chemistry letters*, Springer, v. 12, n. 1, p. 1–13, 2014.

NAGIA, F.; EL-MOHAMEDY, R. Dyeing of wool with natural anthraquinone dyes from fusarium oxysporum. *Dyes and pigments*, Elsevier, v. 75, n. 3, p. 550–555, 2007.

ROBBINS, H.; MONRO, S. A stochastic approximation method. *The annals of mathematical statistics*, JSTOR, p. 400–407, 1951.

SANTIS, D. D.; MORESI, M.; GALLO, A. M.; PETRUCCIOLI, M. Assessment of the dyeing properties of pigments from monascus purpureus. *Journal of chemical technology and biotechnology*, Wiley Online Library, v. 80, n. 9, p. 1072–1079, 2005.

TIBSHIRANI, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, JSTOR, p. 267–288, 1996.

VELMURUGAN, P.; KIM, M.-J.; PARK, J.-S.; KARTHIKEYAN, K.; LAKSHMANAPERUMALSAMY, P.; LEE, K.-J.; PARK, Y.-J.; OH, B.-T. Dyeing of cotton yarn with five water soluble fungal pigments obtained from five fungi. *Fibers and Polymers*, Springer, v. 11, n. 4, p. 598–605, 2010.

ZHANG, Y.; BRADY, M.; SMITH, S. Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm. *IEEE transactions on medical imaging*, IEEE, v. 20, n. 1, p. 45–57, 2001.

# Appendix

# APPENDIX A – A Simulation Study for parameter estimation in MRFs

To compare the described parameter estimation methods for MRFs, we consider three different models, shown in Tables 15 – 17 to simulate from and compare estimators in each scenario: Model 1 and 2 are binary fields with nearest neighbor interaction structures with low and high dependence, respectively, while Model 3 is a 4-color field with a more complex interaction structure.

| $c$ | $\theta_c^{(0)}$ | $\theta_c^{(1)}$ |
|---|---|---|
| Single | 0 | 0 |
| (1,0) | 0.4 | -0.4 |
| (0,1) | 0.4 | -0.4 |

Table 15 – Parameter configuration for Model 1.

| $c$ | $\theta_c^{(0)}$ | $\theta_c^{(1)}$ |
|---|---|---|
| Single | 0 | 0 |
| (1,0) | 1 | -1 |
| (0,1) | 1 | -1 |

Table 16 – Parameter configuration for Model 2.

| $c$ | $\theta_c^{(0)}$ | $\theta_c^{(1)}$ | $\theta_c^{(2)}$ | $\theta_c^{(3)}$ |
|---|---|---|---|---|
| Single | 0 | 0 | 0 | 0 |
| (1,0) | 0.50 | -0.17 | -0.17 | -0.17 |
| (0,1) | 0.50 | -0.17 | -0.17 | -0.17 |
| (3,0) | -0.60 | 0.00 | 0.30 | 0.30 |
| (5,3) | 0.60 | -0.20 | -0.20 | -0.20 |
| (2,0) | -0.10 | 0.30 | -0.10 | -0.10 |

Table 17 – Parameter configuration for Model 3.

We simulate 10 samples of $100 \times 100$ images for each model and compute the following estimators

- MPLE: Maximum Pseudo-Likelihood estimate.
- MCMLE(R): Monte-Carlo Maximum Likelihood estimate with the real parameters as reference $\theta_0$ and $K = 1000$.
- MCMLE(P): Monte-Carlo Maximum Likelihood estimate with the MPLE estimates as reference $\theta_0$ and $K = 1000$.
- SA: Stochastic Approximation estimate with $c_0 = 1$, $c_1 = 3$, $c_2 = 1/20$ and 200 iterations.

For one of the images in each model, we also compute MCMLE(P)* with $K = 15000$ to check the improvement caused by a larger Monte-Carlo sample size.

Simulation results are shown in Tables 18 – 20. We omit one of the parameters for each relative position due to the zero sum condition.

Our results suggests that MPLE, as mentioned in Geyer and Thompson (1992), overestimates de dependence, i.e., the absolute value of parameters is larger than it should be. This implies an inefficient reference parameter $\theta_0$ for MCMLE(P), which results in bad estimates for $K = 1000$, with large standard deviations and means with opposite sign. Increasing the Monte-Carlo sample size to 15000 resulted in an apparent improvement only for a simple interaction structure with high dependence (Model 2).

On the other hand, using MCMLE(R) with appropriate reference $\theta_0$ resulted in far better results, as expected, specially in Model 2. For the more complex interaction structure, these estimates were reasonable on average, but had a higher standard deviation. Except for MCMLE(P) in Model 2, Stochastic Approximation method had the most consistent results, achieving acceptable estimates in terms of both mean and sample deviation for every model.

Since Stochastic Approximation, even using only 200 iterations, is apparently the best option for more complex interaction structures, like the ones we use to describe textures in our problem, we select it as the estimation method for MRFs parameters in this work.

| | Real | MPLE | MCMLE(R) | MCMLE(P) | SA | MCMPLE(P)* |
|---|---|---|---|---|---|---|
| $\theta_0^{(1)}$ | 0 | 0.008(0.036) | 0.036(0.113) | −0.125(0.340) | 0.001(0.005) | 0.034 |
| $\theta_{(1,0)}^{(1)}$ | −0.4 | −0.505(0.015) | −0.310(0.291) | 0.270(1.306) | −0.399(0.013) | 0.309 |
| $\theta_{(0,1)}^{(1)}$ | −0.4 | −0.505(0.015) | −0.478(0.247) | 0.575(0.799) | −0.400(0.014) | 0.098 |

Table 18 – Model 1 simulation results: Estimates mean and its respective standard deviation in parenthesis.

| | Real | MPLE | MCMLE(R) | MCMLE(P) | SA | MCMPLE(P)* |
|---|---|---|---|---|---|---|
| $\theta_0^{(1)}$ | 0 | 0.000(0.024) | 0.000(0.0005) | 0.015(0.048) | 0.005(0.010) | 0.050 |
| $\theta_{(1,0)}^{(1)}$ | −1 | −1.125(0.055) | −0.995(0.022) | −1.026(0.251) | −1.053(0.119) | −1.124 |
| $\theta_{(0,1)}^{(1)}$ | −1 | −1.120(0.046) | −0.999(0.012) | −0.950(0.536) | −1.043(0.122) | −1.190 |

Table 19 – Model 2 simulation results: Estimates mean and its respective standard deviation in parenthesis.

| | Real | MPLE | MCMLE(R) | MCMLE(P) | SA | MCMPLE(P)* |
|---|---|---|---|---|---|---|
| $\theta_0^{(1)}$ | 0 | 0.041(0.055) | −0.048(0.302) | 0.098(0.142) | 0.003(0.022) | −0.032 |
| $\theta_0^{(2)}$ | 0 | 0.070(0.079) | 0.119(0.459) | 0.115(0.213) | 0.004(0.015) | −0.177 |
| $\theta_0^{(3)}$ | 0 | −0.017(0.068) | −0.031(0.152) | −0.053(0.139) | −0.005(0.019) | −0.098 |
| $\theta_{(1,0)}^{(1)}$ | −0.17 | −0.118(0.044) | −0.060(0.317) | −0.112(0.131) | −0.163(0.020) | 0.190 |
| $\theta_{(1,0)}^{(2)}$ | −0.17 | −0.278(0.044) | −0.11(0.770) | −0.232(0.348) | −0.171(0.028) | −0.240 |
| $\theta_{(1,0)}^{(3)}$ | −0.17 | −0.099(0.067) | −0.149(1.044) | −0.058(0.197) | −0.165(0.042) | −0.132 |
| $\theta_{(0,1)}^{(1)}$ | −0.17 | −0.223(0.045) | −0.252(0.235) | 0.013(0.200) | −0.170(0.016) | −0.012 |
| $\theta_{(0,1)}^{(2)}$ | −0.17 | −0.179(0.062) | −0.115(0.320) | 0.022(0.253) | −0.165(0.020) | 0.091 |
| $\theta_{(0,1)}^{(3)}$ | −0.17 | −0.226(0.089) | 0.052(0.416) | 0.025(0.220) | −0.163(0.035) | 0.030 |
| $\theta_{(3,0)}^{(1)}$ | 0.00 | 0.072(0.034) | 0.128(0.301) | 0.081(0.220) | 0.006(0.012) | −0.306 |
| $\theta_{(3,0)}^{(2)}$ | 0.30 | 0.436(0.043) | 0.474(0.377) | −0.193(0.205) | 0.304(0.015) | −0.061 |
| $\theta_{(3,0)}^{(3)}$ | 0.30 | 0.365(0.130) | 0.465(0.267) | 0.137(0.187) | 0.304(0.032) | 0.023 |
| $\theta_{(5,3)}^{(1)}$ | −0.20 | −0.257(0.064) | −0.175(0.428) | 0.086(0.142) | −0.202(0.016) | −0.005 |
| $\theta_{(5,3)}^{(2)}$ | −0.20 | −0.266(0.060) | −0.277(0.599) | −0.001(0.109) | −0.200(0.031) | −0.235 |
| $\theta_{(5,3)}^{(3)}$ | −0.20 | −0.223(0.088) | −0.298(0.673) | −0.060(0.174) | −0.192(0.040) | 0.017 |
| $\theta_{(2,0)}^{(1)}$ | 0.30 | 0.282(0.063) | 0.323(0.209) | −0.027(0.237) | 0.298(0.021) | 0.217 |
| $\theta_{(2,0)}^{(2)}$ | −0.10 | 0.035(0.043) | −0.139(0.303) | −0.122(0.307) | −0.102(0.018) | −0.109 |
| $\theta_{(2,0)}^{(3)}$ | −0.10 | 0.071(0.080) | 0.181(0.387) | −0.068(0.195) | −0.084(0.036) | −0.079 |

Table 20 – Model 3 simulation results: Estimates mean and its respective standard deviation in parenthesis.

# Annex

# ANNEX A – Underlying MRF estimates in Textile Images

| $c$ | $\theta_c^{(0)}$ | $\theta_c^{(1)}$ | $\theta_c^{(2)}$ | $\theta_c^{(3)}$ | $\theta_c^{(4)}$ |
|---|---|---|---|---|---|
| Single | 0.15 | -0.09 | -0.11 | -0.08 | 0.13 |
| (1,0) | 0.50 | 0.24 | -0.11 | -0.41 | -0.22 |
| (8,0) | 0.25 | 0.15 | 0.02 | -0.13 | -0.28 |
| (9,0) | -0.04 | 0.01 | 0.15 | -0.04 | -0.08 |
| (17,0) | 0.06 | 0.03 | 0.06 | -0.06 | -0.08 |
| (-13,5) | 0.24 | 0.08 | -0.06 | -0.15 | -0.11 |
| (-4,5) | 0.25 | 0.12 | -0.04 | -0.21 | -0.12 |
| (4,5) | 0.35 | 0.18 | -0.09 | -0.29 | -0.15 |
| (-17,11) | 0.13 | 0.03 | -0.02 | -0.05 | -0.09 |
| (-9,11) | 0.06 | 0.01 | 0.04 | -0.03 | -0.08 |
| (0,11) | 0.37 | 0.19 | -0.12 | -0.30 | -0.14 |
| (8,11) | 0.16 | 0.08 | 0.02 | -0.16 | -0.10 |
| (-13,16) | 0.06 | 0.01 | 0.07 | -0.06 | -0.07 |
| (-5,16) | 0.07 | 0.04 | 0.08 | -0.06 | -0.14 |
| (-4,16) | -0.01 | -0.02 | 0.08 | -0.01 | -0.04 |
| (4,16) | 0.18 | 0.05 | -0.02 | -0.13 | -0.09 |
| (12,16) | 0.07 | 0.00 | 0.02 | -0.02 | -0.08 |
| (-1,2) | -0.09 | -0.05 | 0.02 | 0.06 | 0.06 |
| (2,1) | -0.24 | -0.13 | 0.02 | 0.16 | 0.19 |

Table 21 – Estimated parameters for the underlying MRF of (a).

| $c$ | $\theta_c^{(0)}$ | $\theta_c^{(1)}$ | $\theta_c^{(2)}$ | $\theta_c^{(3)}$ | $\theta_c^{(4)}$ |
|---|---|---|---|---|---|
| Single | 0.20 | -0.14 | -0.12 | -0.15 | 0.21 |
| (1,0) | 0.61 | 0.21 | -0.13 | -0.44 | -0.25 |
| (3,0) | 0.13 | 0.13 | 0.08 | -0.13 | -0.21 |
| (4,0) | 0.28 | 0.16 | -0.04 | -0.20 | -0.20 |
| (7,0) | 0.07 | 0.03 | 0.02 | -0.01 | -0.12 |
| (11,1) | 0.24 | 0.14 | -0.01 | -0.17 | -0.19 |
| (14,1) | -0.05 | -0.05 | 0.03 | 0.12 | -0.04 |
| (-13,3) | 0.03 | -0.01 | 0.01 | 0.02 | -0.05 |
| (-2,4) | 0.30 | 0.17 | -0.03 | -0.24 | -0.20 |
| (1,4) | 0.01 | 0.05 | 0.08 | -0.01 | -0.13 |
| (2,4) | 0.13 | 0.08 | 0.02 | -0.08 | -0.15 |
| (5,4) | 0.04 | 0.02 | 0.06 | 0.02 | -0.14 |
| (-4,8) | -0.01 | -0.05 | 0.03 | 0.07 | -0.04 |
| (-1,8) | 0.03 | 0.01 | 0.03 | 0.01 | -0.07 |
| (0,8) | 0.11 | 0.06 | 0.00 | -0.06 | -0.11 |
| (3,8) | 0.10 | 0.08 | 0.06 | 0.02 | -0.26 |
| (1,12) | -0.00 | -0.02 | 0.01 | 0.05 | -0.05 |
| (-1,16) | 0.05 | 0.00 | -0.02 | 0.01 | -0.04 |
| (0,2) | -0.06 | -0.04 | -0.02 | 0.03 | 0.08 |

Table 22 – Estimated parameters for the underlying MRF of (b).

| $c$ | $\theta_c^{(0)}$ | $\theta_c^{(1)}$ | $\theta_c^{(2)}$ | $\theta_c^{(3)}$ | $\theta_c^{(4)}$ |
|---|---|---|---|---|---|
| Single | 0.15 | -0.10 | -0.11 | -0.11 | 0.17 |
| (0,1) | 1.36 | 0.59 | -0.67 | -0.84 | -0.43 |
| (1,0) | 0.77 | 0.43 | -0.07 | -0.51 | -0.62 |
| (2,-2) | -0.24 | -0.22 | -0.11 | 0.14 | 0.44 |
| (1,2) | -0.27 | -0.18 | 0.07 | 0.28 | 0.10 |
| (-4,2) | 0.14 | 0.09 | 0.04 | -0.11 | -0.17 |
| (0,-4) | 0.20 | 0.13 | 0.01 | -0.15 | -0.19 |
| (0,-5) | 0.01 | 0.02 | 0.08 | 0.02 | -0.13 |
| (-1,2) | -0.04 | 0.01 | 0.13 | 0.08 | -0.18 |
| (-1,-1) | 0.17 | 0.11 | 0.07 | -0.05 | -0.30 |
| (-2,1) | -0.08 | -0.05 | -0.03 | 0.04 | 0.12 |
| (-8,0) | 0.13 | 0.07 | -0.01 | -0.07 | -0.12 |
| (-8,-1) | -0.07 | -0.05 | 0.02 | 0.08 | 0.02 |
| (2,0) | -0.02 | -0.03 | -0.04 | 0.01 | 0.09 |
| (3,0) | 0.14 | 0.12 | 0.05 | -0.11 | -0.20 |

Table 23 – Estimated parameters for the underlying MRF of (c).

| $c$ | $\theta_c^{(0)}$ | $\theta_c^{(1)}$ | $\theta_c^{(2)}$ | $\theta_c^{(3)}$ | $\theta_c^{(4)}$ |
|---|---|---|---|---|---|
| Single | 0.04 | -0.06 | 0.03 | -0.04 | 0.03 |
| (1,0) | 1.16 | 0.64 | -0.08 | -0.84 | -0.88 |
| (0,1) | 1.01 | 0.51 | -0.11 | -0.71 | -0.71 |
| (-2,2) | -0.23 | -0.23 | -0.21 | 0.04 | 0.63 |
| (-2,-2) | -0.12 | -0.12 | -0.11 | 0.05 | 0.30 |
| (-8,0) | 0.61 | 0.38 | -0.02 | -0.50 | -0.46 |
| (-11,-11) | 0.10 | 0.02 | -0.01 | 0.00 | -0.12 |
| (-2,1) | -0.29 | -0.22 | -0.03 | 0.20 | 0.34 |
| (1,2) | -0.05 | -0.01 | 0.02 | 0.02 | 0.02 |
| (9,1) | -0.03 | 0.00 | 0.10 | 0.09 | -0.17 |
| (-8,1) | -0.25 | -0.15 | 0.08 | 0.27 | 0.04 |
| (2,1) | -0.05 | 0.00 | 0.08 | 0.09 | -0.12 |
| (0,2) | -0.01 | 0.00 | 0.06 | 0.03 | -0.07 |
| (-1,3) | 0.07 | 0.09 | 0.08 | -0.04 | -0.20 |
| (-4,2) | 0.05 | 0.09 | 0.05 | -0.05 | -0.14 |

Table 24 – Estimated parameters for the underlying MRF of (d).