TECHNISCHE UNIVERSITÄT WIEN

VIENNA UNIVERSITY OF TECHNOLOGY

D I P L O M A R B E I T

Displacement Rank Approach for Texture Classification

Ausgeführt am Institut für
Wirtschaftsmathematik
der Technischen Universität Wien

unter der Anleitung von
o.Univ.Prof. Dipl.-Ing. Dr.techn. Manfred Deistler
und
Prof. Dr. Jelena Kovačević
Prof. Dr. Markus Püschel

durch

Clemens Tummeltshammer
Hochwassergasse 16
1230 Wien

Wien, Oktober 2009 _____

Abstract

As texture is one of the main characteristics of images, texture classification has become an emergent field within image analysis over the last decades. This thesis proposes two novel approaches for classifying textures using the theory of displacement operators.

In the first method, a Stein or Sylvester displacement operator with two sparse operator matrices is utilized to transform a texture into a displacement with low rank. Two textures are assumed to be of the same kind if they have low rank with respect to the same displacement operator. New operator matrices and the corresponding classes of textures, or rather classes of structured matrices, are introduced.

The second method makes use of the assumption that textures are generated by a Gaussian Markov Random Field. Thereby the inverse of the covariance matrix, the precision matrix, which represents the field, is a compound of structured matrices with low displacement rank. A basic framework of only vertical and horizontal field interactions is presented and then expanded to a more general framework, which is less restrictive to the structured matrices.

Since textures are often subject to different maps, such as Wavelet transform or FIR filters, it is examined how the displacement operator and its operator matrices have to change so that the transformed texture still has low displacement rank. The discussed transformations are linear maps and upsampling.

Empirical data is given and shows the possible value of these approaches in the future of texture classification.

Kurzfassung

In den letzten Jahrzehnten entwickelte sich die Texturklassifikation zu einem wichtigen Bestandteil der Bildanalyse. Diese Diplomarbeit stellt zwei neuartige Methoden zur Texturerkennung vor und verwendet dabei die Theorie der Displacement-Operatoren.

Die erste Methode benutzt einen Stein oder Sylvester Displacement-Operator mit dünn besetzten Operatormatrizen, um eine Textur in ein Displacement mit geringem Rang zu transformieren. Es wird angenommen, dass zwei Texturen derselben Klasse angehören, falls beide geringen Rang bezüglich des gleichen Displacement-Operators haben. Neue Operatormatrizen und die entsprechenden Texturklassen mit ihren repräsentativen Texturen bzw. strukturierten Matrizen werden vorgestellt.

Bei der zweiten Methode wird angenommen, dass die Textur von einem Gaussian Markov Random Field (GMRF) generiert wurde. Dabei wird unterstellt, dass die Inverse der Kovarianzmatrix, die Präzisionsmatrix, eine Zusammensetzung von strukturierten Matrizen mit geringem Displacement-Rang ist. Ein Basismodell mit vertikalen und horizontalen Feldinteraktionen wird vorgestellt und im weiteren Verlauf weiterentwickelt, um die Bedingungen an die strukturierten Matrizen aufzulockern.

Schließlich wird untersucht, wie die Operatormatrizen angepasst werden müssen, um einen geringen Displacement-Rang beizubehalten, falls die Textur linear oder durch Upsampling transformiert wird.

Empirische Daten in der Form von Texturen werden präsentiert und demonstrieren das mögliche Potenzial dieser Methoden zur Texturklassifikation.

Acknowledgment

I would like to thank my advisors Jelena Kovačević and Markus Püschel from Carnegie Mellon University, where most of the research was conducted. While I was staying in Pittsburgh, they were a great support and considerably extended my knowledge in various areas such as image processing and numerical analysis.

I am very grateful to my supervisor, Manfred Deistler, who gave me the opportunity to write my final thesis about the work I did in Pittsburgh. He also introduced me through inspiring lectures to the areas of econometrics and signal processing.

Special thanks to Gunnar Karlsson from Kungliga Tekniska Högskolan, who made my internship at Carnegie Mellon University possible by putting me in contact with Jelena Kovačević.

Moreover I would like to thank Kristen Swan, Christian Schmid, and Markus Waser for proofreading and very valuable advice on my thesis.

Finally I especially would like to thank my parents for their endless support during my studies in Vienna and abroad in Stockholm and in Pittsburgh.

Contents

List of Figures

List of Tables

Special Symbols

Notation	Description
$\mathbb{R}$	real numbers
$\mathbb{C}$	complex numbers
$\mathbb{R}^{m\times n}$	$m \times n$ real matrices
$\mathbb{C}^{m\times n}$	$m \times n$ complex matrices
$P_{(i,j)}$	entry in the i-th row and the j-th column of the matrix P
$P_{(k:i,l:j)}$	submatrix formed by the rows k to i and the columns l to j of the matrix P
$P_{(i,:)}$	i-th row of the matrix P
$P_{(:,j)}$	j-th column of the matrix P
$P_{M\|(i,j)}$	entry in the i-th row and the j-th column of the matrix P_M
$P_{M\|(k:i,l:j)}$	submatrix formed by the rows k to i and the columns l to j of the matrix P_M
$\mathbf{e}_i$	i-th unit vector
$\mathbf{t} = (t_i)_{i=1}^n$	column-vector of dimension n with t_i being the i-th coordinate
$I = I_r$	$r \times r$ identity matrix
$\mathbf{0}$	null-matrix
P^T	transpose of P
$\bar{P}$	complex conjugate of P
P^*	Hermitian transpose of P
Z_f	$n \times n$ lower shift matrix with $Z_{f\|(1,n)} = f$
Y_{00}	$Z_0 + Z_0^T$
Y_{11}	$Y_{00} + \mathbf{e}_1\mathbf{e}_1^T + \mathbf{e}_n\mathbf{e}_n^T$
$\text{rank}(P)$	rank of the matrix P
$\det(P)$	determinant of the square matrix P
$[P]$	column space of the matrix P
$\text{diag}(\mathbf{t})$	diagonal matrix with entries $(t_1, t_2, \cdots, t_n)$ along the diagonal
$P^{\frac{1}{2}}$	square root of the matrix P

log	logarithm with base e
mod	modulo
abs	absolute value
$\otimes$	Kronecker product
$\oplus$	Kronecker sum
$\subseteq$	a subset of
$\subsetneq$	a proper subset of
$\Delta_{A,B}$	Stein type displacement operator
$\nabla_{A,B}$	Sylvester type displacement operator
$\Delta_{A,B}(M)$	displacement of M with respect to the Stein type displacement operator, $M - AMB$
$\nabla_{A,B}(M)$	displacement of M with respect to the Sylvester type displacement operator, $AM - MB$
$r_{A,B}(M)$	displacement rank of M with respect to the operator matrices A and B
$\overrightarrow{M}$	column vector $\begin{pmatrix} \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_n \end{pmatrix}$ derived from the matrix $M = (\mathbf{m}_1, \cdots, \mathbf{m}_n)$
Ω	$m \times n$ lattice with sites (k, l)
$W \setminus V$	$\{(k,l) \mid (k,l) \in W \text{ and } (k,l) \notin V\}$
$-W$	$\Omega \setminus W$
f_X	density function of the random variable X
$f_{X\|Y}$	conditional density function of the random variable X, given a realization of the random variable Y
$\perp\!\!\!\perp$	probabilistic independence
$\propto$	is proportional to

1 Introduction

In this thesis, two new proposals to texture classification are introduced. Both cases are based on the displacement rank approach. The first assumes that the texture itself has structure with respect to a displacement operator with sparse operator matrices. The second assumes that the texture is generated by a Gaussian Markov Random Field and the inverse of the respective covariance matrix is a composition of structured matrices.

1.1 Motivation

Texture is one of the main features of an image, such as color, shape, etc. Hence different techniques for texture classification have been applied in multiple areas such as automated inspection, medical image processing, document processing, and remote sensing. The field of automated inspection involves detecting defects in images of textiles and controlling automaticly the quality of carpet wear and car paint. Texture classification is, and will be even more in the future, an important part of medical technology such as detecting diseased lungs and leukemia, distinguishing between dissimilar white blood cells, and analyzing a heart's ultrasound image. Already quite common in many applications is character recognition, which is part of document processing. Remote sensing incorporates the identification of images taken from airplanes or satellites [36].

Though several promising approaches to characterize textures have been proposed in the last decades, researchers of different fields are still facing difficulties and are eager to find more sophisticated, precise, and less computationally expensive solutions to classify textures.

1.2 Texture

Although the classification of textures has been intensively studied, there is no single formal definition of texture. However, it is often assumed that

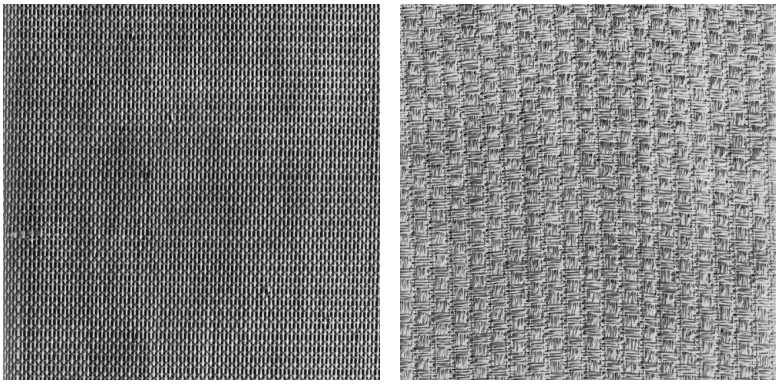
within a texture, one or more patterns appear repeatedly [1]. The elements of these patterns and their spatial organization create the following properties of a texture [15]:

- Fineness
- Coarseness
- Smoothness
- Granulation
- Randomness
- Lineation

Common examples of textural images are pictures of fabrics, grassland, walls, soil, etc. To represent textures, grayscale images are typically used. The value $I(k, l)$ denotes the intensity of the texture I at site (k, l). In general, this intensity ranges from 0 to 1, black to white. Values in-between are mapped to 2^N shades of gray, where N is the number of bits available for every site. Figure 1 shows two textures from the Brodatz album [5].

Figure 1: Brodatz textures [5]

(a) D53 (b) D82



1.3 Classification of Texture

The characterization of an unknown texture employs a classifier to decide which class of textures the unknown texture belongs to. The textural features that represent each class and the computed textural features of the unknown texture are the parameters of this optimization process. Four main categories of methods to compute textural features were designated by Tuceryan and Jain [36]:

- Statistical
- Geometrical
- Model based
- Signal processing

Statistical methods try to describe texture through the spatial distribution of its gray values. One of the earliest approaches was published by Haralick [15]. He utilized a gray level co-occurrence matrix which stores the frequency of two gray values appearing seperated by a given offset. The co-occurrence matrix enables the computation of certain features of the texture such as energy, entropy, contrast, homogeneity and correlation [36].

Geometrical methods assume that a texture is a set of primitives. After the type of primitive is determined, one can either compute statistical features of the found primitives or derive the placement rule of the primitives to characterize the texture [36].

Model based classification of textures uses a certain image model, and the parameters of the model are the features of the texture. Commonly utilized models are Markov Random Fields or Gibbs Random Fields [36]. For example, Chellappa and Chatterjee [8] assume that the texture is generated by a Gauss Markov Random Field and use the least square method to estimate the model parameters. Cohen, Fan and Patel [10] introduce a Gauss

Markov Random Field model, which is able to describe even rotated and scaled textures.

In *Signal processing* methods, the texture is transformed by a linear map, a filter or a filter bank. Energy computations of the output are then used as textural features. Commonly applied transforms are Laws masks, dyadic Gabor filter banks, optimized Gabor filters, Wavelet transforms, Wavelet packets, Wavelet frames, Fourier transform, discrete cosine transform, eigenfilters and optimized FIR filters [30]. For example, Chang and Kuo [6] use the tree-structured wavelet transform, since they claim that the middle frequency channels are dominant in textures. Therefore the pyramid-structured wavelet transform is suboptimal as only low frequency channels get further decomposed. Jain and Farrokhnia [18] utilize Gabor filters for texture classification and segmentation.

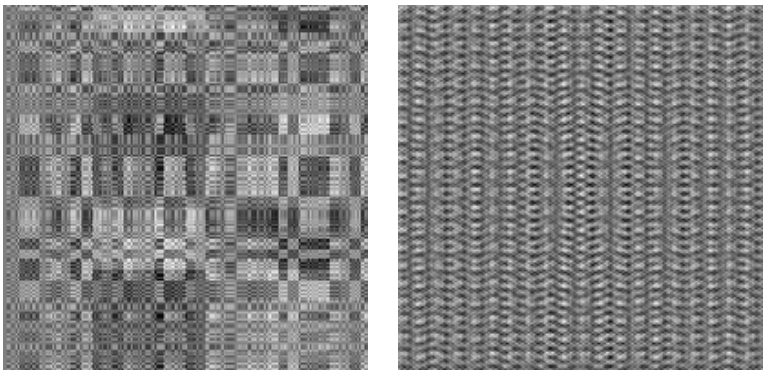
A good overview of the topic of textures and their classification can be found in [15, 36]. For a comparative study of different signal processing methods, I refer the reader to [9, 30]. An examination of invariant texture classification methods is presented in [40].

1.4 Contribution of this Thesis

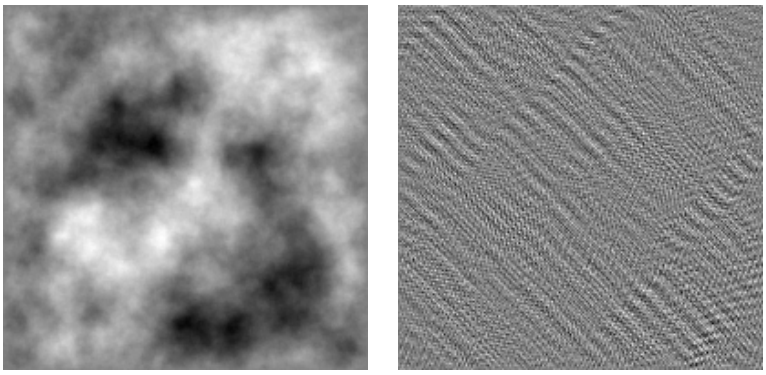
While various methods can be found in the literature of texture classification, the usage of the displacement operator is an entirely novel attempt. Although an incomplete model for texture characterization is presented, this thesis gives an introduction to the connections between the displacement rank approach and textures, serving as a foundation for further research. Two methods are presented and their potency is demonstrated empirically by generating several textures that correspond to different classes of operators. A selection of this empirical data is displayed in sections 4.1.3 and 4.2.3. A preview is given in Figure 2. Both methods may provide a strong basis for an advanced texture classification model in the future.

Figure 2: Sample of textures

(a) Exact structure (b) Exact structure



(c) Random field (d) Random field



1.5 Structure of this Thesis

Section 2 provides an overview of the two types of displacement operators and the displacement rank approach. In section 3, an introduction to Gaussian Markov Random Fields is presented. After the necessary theory is demonstrated, three equivalent representations of a Gaussian Markov Random Field are described. The main section 4 covers the two new proposals of texture classification. The case where the texture itself is assumed to have structure is shown in section 4.1, whereas section 4.2 details the case where the texture is assumed to follow a Gaussian Markov Random Field and the inverse of the

respective covariance matrix has structure. Section 5 demonstrates how the displacement operator changes once the structured matrix is transformed. Finally a conclusion is given, mentioning the potential of these approaches and suggestions for further research.

2 The Displacement Operator

The following section is meant to provide a general description of displacement operators and their use in computations with structured matrices. The displacement operator will be defined, its origins presented and its use in the displacement rank approach explained. Texts from Kailath and Sayed [20] and Pan [28] were consulted in preparation of this overview and should be examined by the reader for further clarification.

2.1 Introduction

Academics in various fields of science and engineering frequently encounter structured matrices and computations that contain them, such as matrix-vector multiplication or solving $n \times n$ linear systems of equations. The fact that fundamental operations with polynomials and rational functions can be represented through structured matrices, as both share many characteristics, gave researchers further incentive to simplify calculations with structured matrices. For further insights on the correlation between polynomials and rational functions and structured matrices, I refer the reader to [20, 28] and references therein.

While searching for a way to make computations with structured matrices more effective and faster, the displacement rank approach was derived. The idea is that the structured matrices get compressed and can then be represented by their displacements, which are specified by just a small number of parameters. Through decompression, one can regain the matrices from their displacements.

The big advantage of using the displacements instead of the original structured matrix is the reduction in time and memory space needed when conducting certain basic computations. One just has to consider that a $n \times n$

matrix with displacement rank r has n^2 parameters, while its displacement only needs from rn to $2rn$ parameters. Furthermore, solving a non-singular structured linear system of n equations ($Ax = b$) with the classical Gaussian elimination algorithm uses time of the order n^3. Improved versions, which partly use the structure of matrices, use time of the order n^2. The displacement rank approach enables a further acceleration to acquire a solution in linear arithmetic time, with logarithmic or polylogarithmic factors possible.

2.2 History

The beginnings of the displacement rank approach can be found in the seminal paper [19], in which the authors wrote about their achievements with Toeplitz-like matrices. Later it was shown that the displacement rank approach can be used for further structured matrices [13, 14, 17]. Pan [27] showed that there is a relationship between the four most important classes of structured matrices, Toeplitz, Hankel, Vandermonde and Cauchy. It was demonstrated that a representative of one of these four types can be transformed into a matrix of another class by simply changing the related displacement operators and displacements.

2.3 Structured Matrices

Commonly used matrices with structure are Toeplitz, Hankel, Vandermonde, Cauchy, Pick, Bézout, Loewner, circulant Toeplitz and block matrices with structured blocks. The most frequent ones used are shown in Table 1.

Table 2 displays the number of parameters and flops needed for describing M and calculating Mv, where M is the regarding structured matrix and v is a vector. The flop count is achieved if the multiplication is implemented efficiently, specifically if the structure of the matrix was used. For further reading on efficient algorithms, please consult chapters 2 and 3 in [28].

Table 1: Most common classes of structured matrices [28]

Toeplitz matrices $(t_{i-j})_{i,j=0}^{n-1}$

$$\begin{pmatrix} t_0 & t_{-1} & \cdots & t_{1-n} \\ t_1 & t_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & t_{-1} \\ t_{n-1} & \cdots & t_1 & t_0 \end{pmatrix}$$

Hankel matrices $(h_{i+j})_{i,j=0}^{n-1}$

$$\begin{pmatrix} h_0 & h_1 & \cdots & h_{n-1} \\ h_1 & h_2 & \ddots & h_n \\ \vdots & \ddots & \ddots & \vdots \\ h_{n-1} & h_n & \cdots & h_{2n-2} \end{pmatrix}$$

Vandermonde matrices $(t_i^j)_{i,j=0}^{n-1}$

$$\begin{pmatrix} 1 & t_0 & \cdots & t_0^{n-1} \\ 1 & t_1 & \cdots & t_1^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & t_{n-1} & \cdots & t_{n-1}^{n-1} \end{pmatrix}$$

Cauchy matrices $(\frac{1}{s_i - t_j})_{i,j=0}^{n-1}$

$$\begin{pmatrix} \frac{1}{s_0-t_0} & \cdots & \frac{1}{s_0-t_{n-1}} \\ \frac{1}{s_1-t_0} & \cdots & \frac{1}{s_1-t_{n-1}} \\ \vdots & & \vdots \\ \frac{1}{s_{n-1}-t_0} & \cdots & \frac{1}{s_{n-1}-t_{n-1}} \end{pmatrix}$$

Table 2: Required amount of parameters and flops to typify M and compute Mv [28]

Matrices M	Parameter count for $m \times n$ matrix M	Flop count for calculating Mv
general	mn	$2mn - n$
Toeplitz	$m + n - 1$	$O((m+n)log(m+n))$
Hankel	$m + n - 1$	$O((m+n)log(m+n))$
Vandermonde	m	$O((m+n)log^2(m+n))$
Cauchy	$m + n$	$O((m+n)log^2(m+n))$

In [28], Pan states four common features structured matrices have:

1. They can be described by only a few parameters.
2. Structured matrix-vector multiplication can be achieved in almost linear time.
3. They have a tight correlation to some calculations with polynomials and rational functions.

4. There exists a displacement operator L which transforms the structured matrix into a matrix with low rank r, namely the displacement. One can use the displacement operator L and the displacement to retrieve the original matrix.

Characteristic 1 is obviously true for the matrices of Table 1. Toeplitz matrices are entirely described by the vector $(t_i)_{i=1-n}^{n-1}$, Hankel matrices by $(h_i)_{i=0}^{2n-2}$, Vandermonde matrices by $(t_i)_{i=0}^{n-1}$, and Cauchy by $(s_i)_{i=0}^{n-1}$ and $(t_i)_{i=0}^{n-1}$. Table 2 and [28] demonstrate characteristic 2. For more information on characteristic 3, I refer the interested reader to [28]. Characteristic 4 is the most important conclusion for this thesis as it is assumed that two textures are equivalent if they have the same displacement operator L.

2.4 The Displacement Rank Approach

2.4.1 Basic Definitions

To achieve these low flop counts shown in Table 2 when computing with structured matrices, the displacement rank approach is needed. When it was first introduced by Kailath, Kung and Morf [19], the goal was to examine if a matrix was Toeplitz-like or not. Nowadays the displacement rank approach covers a wider range of structured matrices and has become a stronger tool than probably expected. Originated in [19], the structure of the linear displacement operator L is given by

$$\Delta_{A,B}(M) = M - AMB, \tag{1}$$

where $A \in \mathbb{C}^{m\times m}$ and $B \in \mathbb{C}^{n\times n}$ are the fixed operator matrices and $M \in \mathbb{C}^{m\times n}$ is the structured matrix. This function $\Delta_{A,B}(\cdot)$ is called the *Stein displacement operator* $L = \Delta_{A,B}$. Initially in [19], M was a Hermitian matrix, A was equal to Z_0 and B was equal to Z_0^* where $\cdot^*$ signifies the Hermitian

conjugation and Z_f a lower shift matrix with first row $[0 \ \cdots \ 0 \ f]$

$$Z_f = \begin{pmatrix} 0 & \cdots & \cdots & 0 & f \\ 1 & \ddots & & 0 & 0 \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix}. \tag{2}$$

A second design of displacement structure, the *Sylvester displacement operator* $L = \nabla_{A,B}$, was first proposed in [17]:

$$\nabla_{A,B}(M) = AM - MB. \tag{3}$$

The most popular operator matrices are Z_f, Z_f^T and $D(\mathbf{u})$, where $\cdot^T$ denotes the transpose and $D(\mathbf{u})$ is a diagonal matrix with entries $(u_i)_{i=1}^n$,

$$D(\mathbf{u}) = \begin{pmatrix} u_1 & & & & \\ & u_2 & & & \\ & & \ddots & & \\ & & & u_{n-1} & \\ & & & & u_n \end{pmatrix}.$$

In case there exists a displacement operator $L = \nabla_{A,B}$ or $L = \Delta_{A,B}$ so that the *displacement* $L(M)$ has low rank r and r is independent of m and n or at least relatively small compared to m and n, then $r_{A,B}$ (or just r) is named the *displacement rank* of M, and M is considered as structured matrix concerning L [19].

2.4.2 Generator Matrices

As the displacement $L(M)$ has low rank r, it can be represented by only a small number of parameters. These parameters are stored in two *Generator*

Matrices $G \in \mathbb{C}^{m\times r}$ and $H \in \mathbb{C}^{n\times r}$ such that

$$L(M) = GH^T. \tag{4}$$

To derive the generator matrices G and H, one can either use the Singular Value Decomposition (SVD) of the displacement $L(M)$ or the Eigenvalue Decomposition, in case $L(M)$ is Hermitian or real symmetric [28].

Definition. The SVD of a matrix M. Let $M \in \mathbb{C}^{m\times n}$, $r = \text{rank}(M)$, and assume that M is nonzero. In this case, there exists a rectangular diagonal matrix $S \in \mathbb{R}^{m\times n}$ and unitary matrices $U \in \mathbb{C}^{m\times m}$ and $V \in \mathbb{C}^{n\times n}$ such that

$$M = USV^*, \tag{5}$$
$$U^*U = I_m \quad \text{and} \quad V^*V = I_n, \tag{6}$$
$$S_{(i,i)} = \sigma_i(M), \quad \text{for } i \in \{1, \cdots, min\{m,n\}\}, \tag{7}$$
$$\sigma_1(M) \geq \sigma_2(M) \geq \cdots \sigma_r(M) > \sigma_{r+1}(M) = \cdots = \sigma_{min\{m,n\}}(M) = 0, \tag{8}$$

where $(\sigma_i)_{i=1}^{min\{m,n\}}$ are the nonnegative singular values of M.

When $L(M)$ is a displacement and $L(M) = USV^*$ its SVD, then a possible generator of minimum length is given by [28]

$$G = U_{(:,1:r)}S^{\frac{1}{2}}_{(1:r,1:r)} \quad \text{and} \quad H = V_{(:,1:r)}S^{\frac{1}{2}}_{(1:r,1:r)}, \tag{9}$$

where $S^{\frac{1}{2}}_{(1:r,1:r)} = \text{diag}(\sigma_1(M)^{\frac{1}{2}}, \sigma_2(M)^{\frac{1}{2}}, \cdots, \sigma_r(M)^{\frac{1}{2}})$.

Both generator matrices G and H together have $(m+n)r$ entries compared to mn entries of the displacement $L(M)$.

2.4.3 Examples of Displacement Operators

The following examples display particular displacement operators for some of the fundamental categories of structured matrices.

Example. Toeplitz matrices. Let T be a Toeplitz matrix defined by the vector $\mathbf{t} = (t_i)_{i=1-n}^{n-1}$. Hence,

$$\begin{aligned}\Delta_{Z_0,Z_0^T}(T) &= T - Z_0 T Z_0^T \\ &= \begin{pmatrix} t_0 & t_{-1} & \cdots & t_{1-n} \\ t_1 & t_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & t_{-1} \\ t_{n-1} & \cdots & t_1 & t_0 \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & t_0 & \cdots & t_{2-n} \\ 0 & \vdots & \ddots & \vdots \\ 0 & t_{n-2} & \cdots & t_0 \end{pmatrix} \\ &= \begin{pmatrix} t_0 & t_{-1} & \cdots & t_{1-n} \\ t_1 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ t_{n-1} & 0 & \cdots & 0 \end{pmatrix}.\end{aligned}$$

As the displacement $L(T)$ has rank 2, the corresponding generator matrices G and H, derived from the SVD of $L(T)$, are of size $n \times 2$.

Definition. Pick matrix. A Pick matrix $P \in \mathbb{C}^{n\times n}$ is given by

$$P_{(i,j)} = \frac{U_{(i,:)}U_{(i,:)}^* - V_{(j,:)}V_{(j,:)}^*}{1 - f_i f_j}, \quad \text{for } i,j \in \{1,\cdots,n\},$$

where $U \in \mathbb{C}^{n\times p}$, $V \in \mathbb{C}^{n\times q}$, and $\mathbf{f} \in \mathbb{C}^{n\times 1}$ is a vector with $|f_i| < 1 \ \ \forall i$.

Example. Pick matrices. Let P be a Pick matrix defined by the vector $\mathbf{f}$ and the matrices U and V, $F = \text{diag}(\mathbf{f})$, and define

$$UV_{(i,j)} := U_{(i,:)}U_{(i,:)}^* - V_{(j,:)}V_{(j,:)}^*, \quad \text{for } i,j \in \{1,\cdots,n\}.$$

Hence,

$$\Delta_{F,F}(P) = P - FPF = \begin{pmatrix} \frac{UV_{(1,1)}}{1-f_1f_1} & \frac{UV_{(1,2)}}{1-f_1f_2} & \cdots & \frac{UV_{(1,n)}}{1-f_1f_n} \\ \frac{UV_{(2,1)}}{1-f_2f_1} & \frac{UV_{(2,2)}}{1-f_2f_2} & & \vdots \\ \vdots & & \ddots & \vdots \\ \frac{UV_{(n,1)}}{1-f_nf_1} & \cdots & \cdots & \frac{UV_{(n,n)}}{1-f_nf_n} \end{pmatrix}$$
$$- \begin{pmatrix} \frac{f_1UV_{(1,1)}f_1}{1-f_1f_1} & \frac{f_1UV_{(1,2)}f_2}{1-f_1f_2} & \cdots & \frac{f_1UV_{(1,n)}f_n}{1-f_1f_n} \\ \frac{f_2UV_{(2,1)}f_1}{1-f_2f_1} & \frac{f_2UV_{(2,2)}f_2}{1-f_2f_2} & & \vdots \\ \vdots & & \ddots & \vdots \\ \frac{f_nUV_{(n,1)}f_1}{1-f_nf_1} & \cdots & \cdots & \frac{f_nUV_{(n,n)}f_n}{1-f_nf_n} \end{pmatrix}$$
$$= \begin{pmatrix} UV_{(1,1)} & UV_{(1,2)} & \cdots & UV_{(1,n)} \\ UV_{(2,1)} & UV_{(2,2)} & & \vdots \\ \vdots & & \ddots & \vdots \\ UV_{(n,1)} & \cdots & \cdots & UV_{(n,n)} \end{pmatrix}.$$

As before, the displacement $L(P)$ has rank 2. Therefore the corresponding generator matrices G and H, computed from the SVD of $L(P)$, are of size $n \times 2$.

Example. Hankel matrices. Let H be a Hankel matrix defined by the vector $\mathbf{h} = (t_i)_{i=0}^{2n-2}$. Hence,

$$\nabla_{Z_1,Z_0^T}(H) = Z_1 H - H Z_0^T$$
$$= \begin{pmatrix} h_{n-1} & h_n & \cdots & h_{2n-2} \\ h_0 & h_1 & \cdots & h_{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ h_{n-2} & h_{n-1} & \cdots & h_{2n-3} \end{pmatrix} - \begin{pmatrix} 0 & h_0 & \cdots & h_{n-2} \\ 0 & h_1 & \cdots & h_{n-1} \\ 0 & \vdots & \ddots & \vdots \\ 0 & h_{n-1} & \cdots & h_{2n-3} \end{pmatrix}$$
$$= \begin{pmatrix} h_{n-1} & h_n - h_0 & \cdots & h_{2n-2} - h_{n-2} \\ h_0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ h_{n-2} & 0 & \cdots & 0 \end{pmatrix}.$$

Similar to the previous examples, the displacement $L(H)$ has rank 2, and therefore the corresponding generator matrices G and H, derived from the SVD of $L(T)$, are of size $n \times 2$.

As one can see from the above examples, the type of operator and operator matrices change based on the kind of structure the matrix has. Furthermore, the types and operator matrices shown above are not unique. For a Hankel matrix, one may also use Δ_{Z_0,Z_0}, which also yields a displacement with rank 2. Apart from the common operator matrices, others like $Z_0 + Z_0^T$ for Hankel + Toeplitz matrices can be used, depending on the structure.

Definition. Supposing that a matrix M has a low displacement rank with respect to a displacement operator, which is affiliated with Toeplitz, Hankel, etc., we call this matrix M *Toeplitz-like, Hankel-like,* etc.

The displacements are then used to compute matrix-vector multiplications, solve linear systems of equations or other basic calculations. To learn more about this topic, I refer the reader to [20] and [28].

2.4.4 Inversion of Displacement Operators

The concluding process is the retrieval of the original matrix from its compressed displacement. This retrieval corresponds to the inversion of the displacement operator L. The following theorem and proof from [29] states a necessary and sufficient condition for the invertibility of a displacement operator L.

Theorem 2.1 *Let* $\{\mathfrak{a}_1, \cdots, \mathfrak{a}_m\}$ *be the eigenvalues of* (A)*, and* $\{\mathfrak{b}_1, \cdots, \mathfrak{b}_n\}$ *the eigenvalues of* (B)*. The Stein type displacement operator* $\Delta_{A,B}$ *is non-singular if and only if* $\mathfrak{a}_i\mathfrak{b}_j \neq 1$ *for all combinations* (i, j)*, and the Sylvester type displacement operator* $\nabla_{A,B}$ *is non-singular if and only if* $\mathfrak{a}_i \neq \mathfrak{b}_j$ *for all combinations* (i, j)*.*

For the proof of Theorem 2.1, three definitions and two propositions are needed [2].

Definition. Kronecker Product. Let $C \in \mathbb{R}^{m\times n}$ and $D \in \mathbb{R}^{k\times l}$. The Kronecker product $(C \otimes D) \in \mathbb{R}^{mk\times nl}$ is given by

$$C \otimes D = \begin{pmatrix} C_{(1,1)}D & C_{(1,2)}D & \cdots & C_{(1,n)}D \\ \vdots & \vdots & \ddots & \vdots \\ C_{(m,1)}D & C_{(m,2)}D & \cdots & C_{(m,n)}D. \end{pmatrix}. \tag{10}$$

Definition. Kronecker Sum. Let $C \in \mathbb{R}^{n\times n}$ and $D \in \mathbb{R}^{m\times m}$. The Kronecker sum $(C \oplus D) \in \mathbb{R}^{nm\times nm}$ is given by

$$C \oplus D = C \otimes I_m + I_n \otimes D. \tag{11}$$

The two propositions show how the eigenvalues and eigenvectors of a Kronecker product or a Kronecker sum of 2 matrices look like.

Proposition 2.2 *Let* $C \in \mathbb{R}^{n\times n}$*,* $D \in \mathbb{R}^{m\times m}$*, and* $\{\mathfrak{c}_1, \cdots, \mathfrak{c}_n\}$ *and* $\{\mathfrak{d}_1, \cdots, \mathfrak{d}_m\}$ *be the eigenvalues of* C *and* D*, respectively. Then the eigenvalues of* $C \otimes D$

are equal to

$$\{\mathfrak{c}_i\mathfrak{d}_j | 1 \leq i \leq n, 1 \leq j \leq m\}. \tag{12}$$

If x is an eigenvector of C corresponding to the eigenvalue $\mathfrak{c}_k$ and y is an eigenvector of D corresponding to the eigenvalue $\mathfrak{d}_l$, then $x \otimes y$ is an eigenvector of $C \otimes D$ corresponding to the eigenvector $\mathfrak{c}_k\mathfrak{d}_l$.

Proof.

$$(C \otimes D)(x \otimes y) = (Cx) \otimes (Dy) = (\mathfrak{c}_k x) \otimes (\mathfrak{d}_l y) = \mathfrak{c}_k\mathfrak{d}_l(x \otimes y)$$

□

Proposition 2.3 *Let $C \in \mathbb{R}^{n\times n}$, $D \in \mathbb{R}^{m\times m}$, and $\{\mathfrak{c}_1, \cdots, \mathfrak{c}_n\}$ and $\{\mathfrak{d}_1, \cdots, \mathfrak{d}_m\}$ be the eigenvalues of C and D, respectively. Then the eigenvalues of $C \oplus D$ are equal to*

$$\{\mathfrak{c}_i + \mathfrak{d}_j | 1 \leq i \leq n, 1 \leq j \leq m\}. \tag{13}$$

If x is an eigenvector of C corresponding to the eigenvalue $\mathfrak{c}_k$ and y is an eigenvector of D corresponding to the eigenvalue $\mathfrak{d}_l$, then $x \otimes y$ is an eigenvector of $C \oplus D$ corresponding to the eigenvector $\mathfrak{c}_k + \mathfrak{d}_l$.

Proof.

$$\begin{aligned}(C \oplus D)(x \otimes y) &= (C \otimes I_m)(x \otimes y) + (I_n \otimes D)(x \otimes y) \\ &= (Cx \otimes y) + (x \otimes Dy) = (\mathfrak{c}_k x \otimes y) + (x \otimes \mathfrak{d}_l y) \\ &= \mathfrak{c}_k(x \otimes y) + \mathfrak{d}_l(x \otimes y) = (\mathfrak{c}_k + \mathfrak{d}_l)(x \otimes y)\end{aligned}$$

□

The last definition is frequently needed when the Kronecker product or the Kronecker sum are used. It symbolizes the transformation of a matrix to a vector.

Definition. Vectorization. The column vector $\overrightarrow{M} = \begin{pmatrix} \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_n \end{pmatrix}$ is obtained by vectorizing the matrix $M = (\mathbf{m}_1, \cdots, \mathbf{m}_n)$ columnwise.

Proof. [Theorem 2.1] From

$$\overrightarrow{\Delta_{A,B}(M)} = (I - B^T \otimes A)\overrightarrow{M} \quad \text{and}$$
$$\overrightarrow{\nabla_{A,B}(M)} = (I \otimes A - B^T \otimes I)\overrightarrow{M},$$

we can see that $\Delta_{A,B}$ is non-singular if and only if the matrix $(I - B^T \otimes A)$ is also non-singular and $\nabla_{A,B}$ is non-singular if and only if the matrix $(I \otimes A - B^T \otimes I)$ is also non-singular. Propositions 2.2 and 2.3 show that the eigenvalues of the matrix $(I - B^T \otimes A)$ are equal to $1 - \mathfrak{a}_i\mathfrak{b}_j$ for all combinations (i, j) and the eigenvalues of the matrix $(I \otimes A - B^T \otimes I)$ are equal to $\mathfrak{a}_i - \mathfrak{b}_j$ for all combinations (i, j), which proves the theorem. □

3 Gauss Markov Random Fields

Rue and Held's book [34] was consulted in preparation of the following section.

3.1 Introduction

In the past years, Gauss Markov Random Fields (GMRFs) have been an increasingly important topic. Especially in the areas of spatial statistics and image analysis, researchers make use of GMRFs; however one can also find applications of them in structural time-series analysis, analysis of longitudinal and survival data, spatiotemporal statistics, graphical models and semiparametric statistics. GMRFs are employed to characterize the spatial and temporal dynamics of nature and real systems. A GMRF is a random vector, which has a *multivariate normal* (also called Gaussian) distribution. To justify the term Markov, it also has to fulfill supplementary *conditional independence* conditions. These conditional independences cause the *precision matrix*, which is the inverse of the covariance matrix, to be highly sparse. Therefore the GMRF is preferably represented by its precision matrix, rather than by its covariance matrix, as the latter is in general not sparse.

3.2 History

In [7] Chellappa gives a brief overview of the most important developments in the literature of GMRFs:
In 1954, Whittle [37] published a paper about non-causal autoregressive time series. Furthermore he discussed the weaknesses of the ordinary least square method when it comes to estimating the parameters of a 2-D non-causal autoregressive (NCAR) model and hypothesis testing processes to choose a model structure. An abstract concept of GMRFs was suggested by Rosanov [33] in 1967. Around that time, Levy [21, 22] and Wong [38] examined continuous random fields and thereby stimulated Woods [39], who was

unfamiliar of Rosanov's previous work, to produce equivalent results in 1972. In addition, Woods demonstrated that GMRFs can be used for spectral estimation. In the seventies, many papers were published about GMRFs or Markov Random Fields (MRFs); for instance, Moran [24, 25] explored on his own representations of basic MRF models, and together with Besag [4], the maximum likelihood estimation for rectangular lattice autonormal schemes. But it was especially the celebrated paper by Besag [3], which motivated multiple researchers to further examine the applications of GMRFs or MRFs in general. Besag's seminal paper not only includes an illustration of the Hammersely Clifford theorem, but also various representations of MRF models, coding estimates, and connections between NCAR and GMRF models. Since then, applications of GMRFs have been utilized in various areas, such as texture classification. For published papers on this topic, I refer the reader to [8, 10, 16, 31, 32].

3.3 Theory

This section will provide the necessary background for understanding GMRFs, including notations, and explanations of conditional independence, undirected graphs, positive-definiteness of matrices and the multivariate normal distribution.

3.3.1 Some Definitions

Let Ω denote a $m \times n$ lattice with sites (k, l) such that $\Omega = \{(k, l)|1 \leq k \leq m, 1 \leq l \leq n\}$. If $\mathbf{X}$ takes values on Ω, then $X_{(k,l)}$ indicates the value of $\mathbf{X}$ at site (k, l). In case $W \subseteq \Omega$ is a subset, then $\mathbf{X}_W = \{X_{(k,l)}|(k, l) \in W\}$ are the values of $\mathbf{X}$ at the sites (k, l), which are part of the set W. Let $W \subseteq \Omega$ and $V \subseteq \Omega$ be two sets; thus $W \setminus V = \{(k, l)|(k, l) \in W \text{ and } (k, l) \notin V\}$. Furthermore $-W$ denotes the set $\Omega \setminus W$.

The density of a random variable X is denoted by f_X. The conditional

density of a random variable X, when a realization y_0 of the random variable Y is known, is given by $f_{X|Y}$. To calculate the conditional density, the following formula is used

$$f_{X|Y}(x, y_0) = \frac{f_{X,Y}(x, y_0)}{f_Y(y_0)}, \tag{14}$$

for $f_Y(y_0) > 0$. $\mathbf{E}(\cdot)$, $\mathrm{Cov}(\cdot, \cdot)$, $\mathrm{Prec}(\cdot)$ and $\mathrm{Var}(\cdot)$ symbolize the expected value, the covariance, the precision and the variance respectively.

3.3.2 Conditional Independence

Assume that $\mathbf{X} = (X_1, X_2, X_3)^T$ is a random vector consisting of three random variables and x_3 is the known realization of X_3. If the revealment of x_2 doesn't provide new information about the distribution of X_1, then X_1 and X_2 are conditionally independent. Therefore the joint density can be represented as follows

$$f_{\mathbf{X}} = f_{X_1|X_3} f_{X_2|X_3} f_{X_3},$$

instead of the general formula for the joint density

$$f_{\mathbf{X}} = f_{X_1|X_2,X_3} f_{X_2|X_3} f_{X_3}.$$

As a result, two random variables X and Y are conditionally independent given the random variable Z if and only if $f_{X,Y|Z} = f_{X|Z} f_{Y|Z}$. This conditional independence is indicated by [12]

$$X \perp\!\!\!\perp Y \mid Z.$$

Be aware that, although X and Y are conditionally independent, they don't have to be marginally independent.

The following *factorization theorem* by Dawid [12] shows a simple way to test if two random variables/vectors are conditionally independent.

Theorem 3.1 *$X \perp\!\!\!\perp Y \mid Z$ if and only if there exist some functions g and h such that for all z with $f_Z(z) > 0$*

$$f_{X,Y,Z}(x, y, z) = g(x, z)h(y, z).$$

3.3.3 Undirected Graphs

A convenient way of representing a GMRF is a system of undirected graphs, which practically *visualizes the conditional dependences* between the different random variables. An undirected graph $\mathcal{G}$ consists of a set of Nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$. For $k, l \in \mathcal{V}$ and $k \neq l$, the pair $\{k, l\}$ is an element of $\mathcal{E}$ if and only if node k and node l are connected by an undirected graph. Describing Markov fields with undirected graphs was first introduced by Darroch, Lauritzen and Speed [11].

Every node that shares an edge with node k is considered to be in the *neighborhood* of node k [11]:

$$ne(k) = \{l \in \mathcal{V} | \{k, l\} \in \mathcal{E}\}.$$

Additionally, if k and l are neighbors, it is denoted by $k \sim l$.

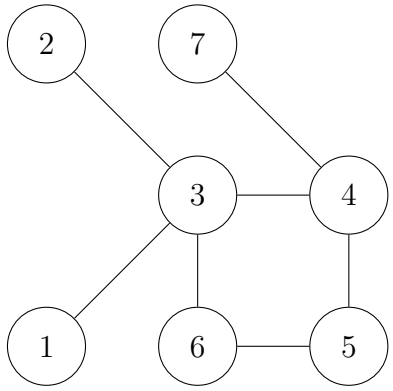
If $\{k_i, k_{i+1}\} \in \mathcal{E}$ for $i \in \{1, \cdots, j-1\}$, then the sequence $k_1, \cdots, k_j$ of distinct nodes is called a *path*. In case every path from node $k \notin W$ to node $l \notin W$ includes at least one node from a subset $W \subseteq \mathcal{V}$, then W *seperates* k and l. A subset $W \subseteq \mathcal{V}$ seperates two disjoint subsets $V \subseteq \mathcal{V} \setminus W$ and $U \subseteq \mathcal{V} \setminus W$, if W seperates the nodes k and l for all $k \in V$ and $l \in U$.

A basic model of an undirected graph is displayed in Figure 3. One can verify that the sets of nodes and edges and some examples of neighborhoods

are given by

$$\mathcal{V} = \{1, 2, \cdots, 7\},$$
$$\mathcal{E} = \{\{1,3\},\{2,3\},\{3,4\},\{3,6\},\{4,5\},\{5,6\},\{4,7\}\},$$
$$ne(1) = 3, \quad ne(3) = \{1,2,4,6\}, \quad ne(5) = \{4,6\}, \quad \text{etc.}$$

Figure 3: Basic model of an undirected graph



This system can be easily extended to a 2-dimensional lattice. The nodes are then the sites of the lattice, and the elements of the sets and the neighbourhoods look as follows

$$\{k,l\} \in \mathcal{V}, \quad \text{for } 1 \le k \le m \text{ and } 1 \le l \le n,$$
$$\{\{k,l\},\{i,j\}\} \in \mathcal{E} <=> \text{ there is an edge between } \{k,l\} \text{ and } \{i,j\},$$
$$ne(\{k,l\}) = \{\{i,j\} \in \mathcal{V} | \{\{k,l\},\{i,j\}\} \in \mathcal{E}\}.$$

With a 2-dimensional lattice, normally the Euclidean distance is used to determine the order of the neighborhood [16].

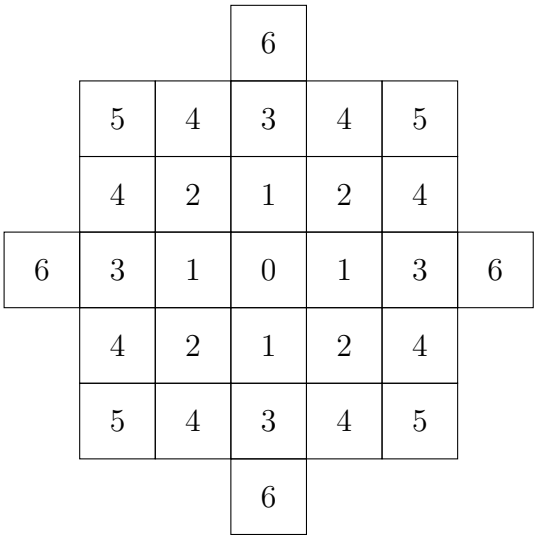
Definition. Let Ω be a $m \times n$ lattice. A P-th order neighborhood $ne^P(\{k,l\})$ of site (k,l) can be written as

$$ne^P(\{k,l\}) = \{\{i,j\} | 0 < (k-i)^2 + (l-j)^2 \le D^P\}, \tag{15}$$

with D^P being the square of the Euclidean distance between the corresponding site and its furthest neighbor [26].

Common values for D^P are $1, 2, 4, 5, 8, 9$ for $P = 1, 2, 3, 4, 5, 6$. Be aware that a P-th order neighborhood set also contains the neighbors of the neighborhood sets with lower orders. Figure 4 displays neighborhoods of different orders following the previous definition.

Figure 4: Hierarchical neighborhood system [16]



3.3.4 Symmetric Positive-Definite Matrices

To ensure that a GMRF $\mathbf{X}$ defines a proper density, the precision matrix must be positive-definite. The case where the precision matrix is positive-semidefinite and the corresponding GMRF $\mathbf{X}$ is therefore improper is not discussed in this thesis.

Let $A \in \mathbb{R}^{n \times n}$ be a symmetric matrix, $\{\lambda_1, \cdots, \lambda_n\}$ its real eigenvalues and $\lambda_{min} = min\{\lambda_1, \cdots, \lambda_n\}$. A is positive-definite (positive-semidefinite) if and only if $\lambda_{min} > 0$ ($\lambda_{min} \geq 0$). A positive-definite matrix is symbolized by $A > 0$ and a positive-semidefinite matrix by $A \geq 0$ [2].

Some fundamental attributes of a symmetric positive-definite (SPD) matrix A are:

- A^{-1} is SPD.
- $\text{rank}(A) = n$.
- $\det(A) > 0$.
- Every principal submatrix of A is positive definite.
- There exists a unique lower triangular matrix C such that $A = CC^T$, whereas the diagonal elements of C are strictly positive. This factorization is denoted by *Cholesky decomposition*.

It is sufficient for a symmetric matrix to be positive definite if the matrix is diagonally dominant:

$$A_{(i,i)} - \sum_{j:j\neq i} |A_{(i,j)}| > 0 \quad \forall i. \tag{16}$$

Note that this condition is not necessary for positive definiteness.

3.3.5 Multivariate Normal/Gaussian Distribution

This section will recapitulate the most fundamental characteristics of the multivariate normal distribution. Let $\mathbf{X} = (X_1, \cdots, X_n)^T$ be a normal random vector with mean $\boldsymbol{\mu}$ and SPD covariance matrix Σ. The distribution of $\mathbf{X}$ is given by the density function

$$f_{\mathbf{X}}(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right), \quad \mathbf{x} \in \mathbb{R}, \tag{17}$$

where $\mu_k = \mathbf{E}(X_k)$, $\Sigma_{(k,l)} = \text{Cov}(X_k, X_l)$ and $\Sigma_{(k,k)} = \text{Var}(X_k) > 0$. This distribution of $\mathbf{X}$ is also denoted by $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$.

3.4 Characterization of GMRF's

3.4.1 Basic Definition

If the random vector $\mathbf{X} = (X_1, \cdots, X_n)^T$ is normally distributed with parameters $\boldsymbol{\mu}$ and Σ and the undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is given such that $\mathcal{V} = \{1, \cdots, n\}$ and the edge $\{k, l\}$ is not an element of $\mathcal{E}$ if and only if $X_k \perp\!\!\!\perp X_l \mid X_{-\{k,l\}}$, then $\mathbf{X}$ is a GMRF concerning $\mathcal{G}$.

As we know, changes in $\boldsymbol{\mu}$ don't affect the pairwise conditional independences of $\mathbf{X}$. Therefore all the information about these independences must be contained in the second parameter, the covariance matrix Σ. However, it is more convenient to use its inverse, the precision matrix Q, to represent a GMRF. This follows from the fact that the *precision matrix reflects the conditional independences* of $\mathbf{X}$. The next theorem from [34] proves this link.

Theorem 3.2 *If* $\mathbf{X}$ *is normally distributed with mean* $\boldsymbol{\mu}$ *and precision matrix* $Q > 0$*, then*

$$X_k \perp\!\!\!\perp X_l \mid \mathbf{X}_{-\{k,l\}} \iff Q_{k,l} = 0, \quad \text{for } k \neq l.$$

Proof. Partitioning $\mathbf{X}$ as $(X_k, X_l, \mathbf{X}_{-\{k,l\}})$, fixing $k \neq l$, assuming $\boldsymbol{\mu} = \mathbf{0}$ without loss of generality, and using the formula for the multivariate normal distribution (17) yields

$$\begin{aligned} f_{X_k, X_l, \mathbf{X}_{-\{k,l\}}} &\propto \exp\left(-\frac{1}{2} \sum_{i,j} x_i Q_{(i,j)} x_j \right) \\ &\propto \exp\left(-\frac{1}{2} x_k x_l (Q_{(k,l)} + Q_{(l,k)}) - \frac{1}{2} \sum_{\{i,j\} \neq \{k,l\}} x_i Q_{(i,j)} x_j \right), \end{aligned}$$

where the sign $\propto$ denotes "is proportional to".

Clearly $x_k x_l$ is not included in the second term. The first term includes $x_k x_l$ if and only if $Q_{(k,l)} \neq 0$. As a result, the density function $f_{X_k, X_l, \mathbf{X}_{-\{k,l\}}}$

can be factorized as follows

$$f_{X_k,X_l,\mathbf{X}_{-\{k,l\}}}(x_k, x_l, \mathbf{x}_{-\{k,l\}}) = g(x_k, \mathbf{x}_{-\{k,l\}})h(x_l, \mathbf{x}_{-\{k,l\}}),$$

for some functions g and h if and only if $Q_{(k,l)} = 0$. Applying Theorem 3.1 concludes the proof. □

Consequently it is possible to deduce from Q whether or not X_k and X_l are conditionally independent and the graph $\mathcal{G}$ is fully defined by Q. In contrast, a graph $\mathcal{G}$ specifies which entries of Q are nonzero.

Remark. Some authors use the term *potential matrix* instead of *precision matrix* for the inverse of the covariance matrix. For consistency, the term precision matrix is used throughout this thesis.

The following definition is the first of three characterizations for a GMRF stated in this thesis.

Definition. GMRF. Let $\mathbf{X} = (X_1, \cdots, X_n)^T$ be a random vector with mean $\boldsymbol{\mu}$ and precision matrix $Q > 0$. $\mathbf{X}$ is a GMRF regarding the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ if and only if its density is given by

$$f_{\mathbf{X}} = (2\pi)^{-\frac{n}{2}}|Q|^{\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T Q(\mathbf{x}-\boldsymbol{\mu})\right) \tag{18}$$

and

$$Q_{(k,l)} \neq 0 \iff \{k,l\} \in \mathcal{E}, \quad \text{for all } k \neq l.$$

For a completely dense matrix Q, the graph $\mathcal{G}$ is completely connected. Therefore every Gaussian distribution with SPD covariance matrix Σ is also a GMRF. At the same time, every GMRF is a Gaussian distribution with SPD covariance matrix; but in general, only GMRFs with a sparse precision matrix are used.

Certain values such as the conditional expected value can be easily determined from the precision matrix Q [34].

Theorem 3.3 *If* $\mathbf{X}$ *is a GMRF with respect to the graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ *with parameters* $\boldsymbol{\mu}$ *and* $Q > 0$*, then*

$$\mathbf{E}(X_k|\mathbf{X}_{-k}) = \mu_k - \frac{1}{Q_{(k,k)}} \sum_{l:l\sim k} Q_{(k,l)}(x_l - \mu_l), \tag{19}$$

$$\text{Prec}(X_k|\mathbf{X}_{-k}) = Q_{(k,k)}, \tag{20}$$

$$\text{Corr}(X_k, X_l|\mathbf{X}_{-\{k,l\}}) = -\frac{Q_{(k,l)}}{\sqrt{Q_{(k,k)}Q_{(l,l)}}}, \quad \textit{for } k \neq l. \tag{21}$$

Consequently the conditional precision of X_k given $\mathbf{x}_{-k}$ for $k \in \{1, \cdots, n\}$ can be found along the diagonal of Q, whereas the conditional correlation between X_k and X_l given $\mathbf{x}_{-\{k,l\}}$ for $k \neq l$ can be extracted by scaling the off-diagonal elements of Q. By contrast, the entries of the covariance matrix Σ reflect the marginal dependencies: $\text{Var}(X_k) = \Sigma_{(k,k)}$ and $\text{Corr}(X_k, X_l) = \frac{\Sigma_{(k,l)}}{\sqrt{\Sigma_{(k,k)}\Sigma_{(l,l)}}}$.

3.4.2 Markov Properties

Up to this point, only the pairwise Markov property has been incorporated. Additionally there exist the local Markov property and the global Markov property, which can be used to describe a GMRF.

Theorem 3.4 *In case* $\mathbf{X}$ *is a GMRF concerning the graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$*, the following three Markov properties are equivalent:*

- *The pairwise Markov property,*

$$X_k \perp\!\!\!\perp X_l \mid \mathbf{X}_{-\{k,l\}}, \quad \textit{if } \{k,l\} \notin \mathcal{E} \textit{ and } k \neq l.$$

- *The local Markov property,*

$$X_k \perp\!\!\!\perp \mathbf{X}_{-\{k,ne(k)\}} \mid \mathbf{X}_{ne(k)}, \quad \forall k \in \mathcal{V}.$$

- *The global Markov property,*

$$\mathbf{X}_A \perp\!\!\!\perp \mathbf{X}_B \mid \mathbf{X}_C, \quad \textit{for all disjoint sets } A,B \textit{ and } C,$$

with C seperating A and B and $A, B \neq \emptyset$.

A proof by Speed and Kiiveri can be found in [35].

3.4.3 Full Conditionals

A second approach to representing a GMRF is full conditionals $f_{X_k|\mathbf{X}_{-k}}$. This approach was introduced by Besag [3,4] under the name of *Conditional Autoregressions* (CAR).

The normal full conditionals are given by

$$\mathbf{E}(X_k|\mathbf{X}_{-k}) = \mu_k - \sum_{l|l\sim k} \beta_{k,l}(x_l - \mu_l) \quad \text{and} \tag{22}$$

$$\text{Prec}(X_k|\mathbf{X}_{-k}) = \kappa_k > 0, \tag{23}$$

with $k \in \{1, \cdots, n\}$, $\{\beta_{k,l}\}$ for $k \neq l$ and vectors $\boldsymbol{\kappa}$ and $\boldsymbol{\mu}$. In case $\beta_{k,l}$ is not zero, the node l is in the neighborhood of node k ($k \sim l$). Apparently $\beta_{k,l}$ is nonzero if $\beta_{l,k}$ is nonzero, as $\sim$ is a symmetrical relation. Considering the formulas (19) and (20), as well as the fact that a precision matrix has to be symmetric, yields the following theorem:

Theorem 3.5 *Let* $\mathbf{X}$ *be a random vector defined by the n normal full conditionals as in* (22) *and* (23). $\mathbf{X}$ *is a GMRF with respect to the graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ *with the parameters* $\boldsymbol{\mu}$ *and* Q *where*

$$Q_{(k,k)} = \kappa_k \quad \textit{and} \quad Q_{(k,l)} = \kappa_k \beta_{k,l},$$

if $\kappa_k\beta_{k,l} = \kappa_l\beta_{l,k}$ *and* $Q > 0$.

For a proof of this theorem, see [34].

The precision matrix Q is given by

$$Q = \text{diag}(\boldsymbol{\kappa})(I + B),$$

where $B \in \mathbb{R}^{n\times n}$ and $B_{(k,l)} = \beta_{k,l}$, with $\beta_{k,k}$ set to zero. Therefore,

$$Q > 0 \Longleftrightarrow (I + B) > 0.$$

3.4.4 Noncausal Autoregressive Field Representation

A third possibility for symbolizing a GMRF is a noncausal autoregressive (AR) model, which has a *correlated field as input*. It was introduced by Woods [39] as the *minimum mean-square representation*.

Let Ω be a $m \times n$ lattice, $\mathbf{X}$ a random field defined on Ω, and $\overrightarrow{\mathbf{X}}$ the vectorial representation of $\mathbf{X}$, obtained by stacking every coloumn on top of each other. In case $\mathbf{X}$ is a zero-mean GMRF, then it can be written as

$$Q_\sigma \overrightarrow{\mathbf{X}} = \overrightarrow{e}, \tag{24}$$

where $Q_\sigma \in \mathbb{R}^{mn\times mn}$ is a scaled version of the inverse of the covariance matrix

$$\mathbf{E}(\overrightarrow{\mathbf{X}}\overrightarrow{\mathbf{X}}^T) = \left(\frac{1}{\sigma^2}Q_\sigma\right)^{-1}, \tag{25}$$

$\overrightarrow{e} \in \mathbb{R}^{mn\times 1}$ is a zero-mean Gaussian noise with

$$\mathbf{E}(\overrightarrow{e}\,\overrightarrow{e}^T) = \sigma^2 Q_\sigma, \tag{26}$$

and

$$\mathbf{E}(\overrightarrow{\mathbf{X}}\,\overrightarrow{e}^T) = \sigma^2 I. \tag{27}$$

Note that the GMRF covariance is not affected by changing to a nonzero-mean.

3.4.5 One-Sided Representations for Noncausal Fields

In [26] Moura and Balram presented two equivalent one-sided representations of the field defined by (24). Unlike the original noncausal definition (24), where the field is driven by correlated noise, the two equivalent characterizations by Moura and Balram are *driven by white noise*.

The *backward representation* uses the lower-upper Cholesky decomposition of the symmetric and positive definite precision matrix Q_σ:

$$Q_\sigma = U^T U, \tag{28}$$

with U being upper triangular. Revision of the field representation (24) yields

$$U\overrightarrow{\mathbf{X}} = \overrightarrow{w}, \tag{29}$$

where $\overrightarrow{w}$ is white noise

$$\overrightarrow{w} = (U^T)^{-1}\overrightarrow{e}, \tag{30}$$

$$\mathbf{E}(\overrightarrow{w}\,\overrightarrow{w}^T) = \sigma^2 I. \tag{31}$$

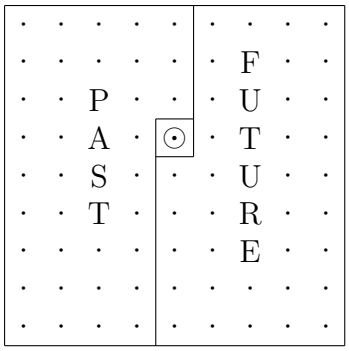
From those equations, it follows that

$$\mathbf{E}(\overrightarrow{\mathbf{X}}\,\overrightarrow{w}^T) = \sigma^2 U^{-1}. \tag{32}$$

The field defined by (29) is equivalent to (24), but, unlike the latter, it is *one-sided* and any entry in the random vector $\overrightarrow{\mathbf{X}}$ depends just on the random variables X_k, which are situated in its "future". Considering the not vectorized version $\mathbf{X}$, which is defined on a 2-D lattice, the "past" is every column

to the left of the "present" and every node above the present in the same column (see Figure 5). Be aware that the partition of "past" and "future" depends on the ordering of the vectorized version $\overrightarrow{\mathbf{X}}$ of the field $\mathbf{X}$.

Figure 5: Division of the 2-D lattice: $\odot$ symbolizes the "present" [26]



In addition to (28), there exists the upper/lower Cholesky decomposition of Q_σ, which is applied in the *forward representation* model.

$$Q_\sigma = L^T L, \tag{33}$$

with L being lower triangular. Using the factorization in (24) generates

$$L\overrightarrow{\mathbf{X}} = \overrightarrow{z}, \tag{34}$$

where $\overrightarrow{z}$ is white noise

$$\overrightarrow{z} = (L^T)^{-1}\overrightarrow{e}, \tag{35}$$

$$\mathbf{E}(\overrightarrow{z}\,\overrightarrow{z}^T) = \sigma^2 I. \tag{36}$$

From those equations, it follows that

$$\mathbf{E}(\overrightarrow{\mathbf{X}}\,\overrightarrow{z}^T) = \sigma^2 L^{-1}. \tag{37}$$

In the "forward" regressor model (34), the "present" depends only on those values that lie in its "past".

3.4.6 The Structure of the Precision Matrix

Moura and Balram [26] also explored the form of the precision matrix Q_σ for zero-mean, nondegenerated GMRFs of first and higher order. Considering that sites at the border of a finite 2-D lattice can have neighbours from the outside of the lattice, the introduction of *boundary rules* is required. Moura and Balram give some examples of suitable boundary conditions, such as setting the values of the neighbours outside of the lattice to zero or supposing that the lattice is a torus. For further reading on this topic, I refer to [26].

Moura and Balram claim that a SPD matrix Q_σ is the precision matrix of a first-order GMRF if and only if it can be written as

$$Q_\sigma = Q_c + Q_{b.c.}, \tag{38}$$

where $Q_{b.c.}$ contains all the information about the chosen boundary conditions and Q_c is unaffected by them. Q_c is structured as follows

$$Q_c = \begin{pmatrix} A_1 & \tilde{A}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \tilde{A}_1 & A_2 & \tilde{A}_2 & \ddots & \vdots \\ \mathbf{0} & \tilde{A}_2 & A_3 & \ddots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \tilde{A}_{n-1} \\ \mathbf{0} & \cdots & \mathbf{0} & \tilde{A}_{n-1} & A_n \end{pmatrix}, \tag{39}$$

where

$$A_i = \begin{pmatrix} q^{i,1} & -\alpha_v^{i,1} & \mathbf{0} & \cdots & \mathbf{0} \\ -\alpha_v^{i,1} & q^{i,2} & -\alpha_v^{i,2} & \ddots & \vdots \\ \mathbf{0} & -\alpha_v^{i,2} & q^{i,3} & \ddots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & -\alpha_v^{i,m-1} \\ \mathbf{0} & \cdots & \mathbf{0} & -\alpha_v^{i,m-1} & q^{i,m} \end{pmatrix} \tag{40}$$

and

$$\tilde{A}_i = -\operatorname{diag}(\alpha_h^{i,1}, \alpha_h^{i,2}, \cdots, \alpha_h^{i,m}). \tag{41}$$

The parameters $\alpha_v^{i,j}$ denote the vertical and $\alpha_h^{i,j}$ the horizontal field interactions between neighboring sites.

The following characteristics of the matrix Q_c are apparent from the description [26]:

- It is highly sparse, symmetric and block tridiagonal.
- The blocks A_i are also highly sparse, symmetric and tridiagonal while the blocks $\tilde{A}_i$ are diagonal matrices.
- In total, the matrix Q_c has a maximum of 5 nonzero diagonals.

For further reading on the structure of the boundary precision matrix $Q_{b.c.}$ in first-order fields and cases where the field is homogeneous or of higher order, I refer the reader to [26].

4 Texture Description through Displacement Operators

4.1 Exact Structure

The goal of the project is to demonstrate the potential of the displacement operator as an effective approach for classifying textures. The idea behind this approach is that two textures are equivalent or of the same type, if both have low displacement rank with respect to the same displacement operator.

One can think of having n classes of textures, each uniquely defined by a type of displacement operator and its sparse operator matrices, such as Δ_{Z_0,Z_0^T} for Toeplitz-like matrices. In addition, each class has a certain number of representative textures/structured matrices, which have low displacement rank with respect to the displacement operator of that class. To assign the class for an unknown texture, an optimization algorithm is used. The algorithm operates either by checking with which displacement operator the unknown texture has the lowest displacement rank or by verifying which representative texture is the "closest" to the unknown texture. In the latter case, it is necessary to define "close" in a mathematical way.

This thesis doesn't examine this optimization process but rather considers new classes of structured matrices and their representative textures. By reversing the process, first finding new displacement operator matrices and then working backwards, new textures/structured matrices can be found. This work is described in the following section.

4.1.1 Basic Setup

As mentioned above, to find representatives for a certain class, one has to begin by checking which textures M have a low displacement rank for given *sparse displacement operator matrices* A and B and given generator matrices G and H. A and B are either common displacement operator matrices

like shift matrices or diagonal matrices, modifications of these or completely different ones. G and H are matrices of size $n \times r$, where n is the dimension of the matrix M and r is the displacement rank. For simplicity reasons, we assume that the matrix M is a square matrix. The generator matrices G and H are either random matrices, 2-dimensional trigonometric, polynomic, logarithmic or exponential functions or a mixture of these. In MATLAB® the *Sylvester type displacement operator* $\nabla_{A,B}(M)$ is solved for M by

```
M = lyap(A,-B,-G*H');
```
(42)

which solves the continuous-time Lyapunov equation.

In a similar manner one can solve the *Stein type displacement operator* $\Delta_{A,B}(M)$ for M

```
M = dlyap(A,B,G*H');
```
(43)

which solves the discrete-time Lyapunov equation.

If necessary, a *logtransformed matrix* of the obtained M is generated to make patterns more visible.

```
if min(min(M)) < 0
  M_pos = M >= 0;
  M_neg = M < 0;
  M_check = M_pos - M_neg;
  M_log = (log(1+abs(M)));
  M_log = M_check.*M_log;
else
  M_log = (log(1+M));
end
```

in MATLAB® where `log` denotes the natural logarithm, `abs` the absolute value, and `.*` the elementwise matrix multiplication.

4.1.2 Visualization

The matrices M and M_log are then displayed by the function `imshow(M,[])` as a grayscale image. The rectangular brackets adjust the display range in such a way that the value `min(M(:))` is mapped to black and the value `max(M(:))` to white, where `min(M(:))` and `max(M(:))` denote the minimum and the maximum entries of the matrix M . The function linearly interpolates the values in-between and displays them as intermediate shades of gray using 8-bits of gray levels [23]. This is equivalent to applying the linear transformation (44) and using the command `imshow(M)`, the former of which maps the interval `[min(M(:)),max(M(:))]` on the interval `[0,1]` of each entry of the matrix M:

$$M^P_{(i,j)} = kM_{(i,j)} + d, \quad i,j \in \{1, \cdots, n\}, \tag{44}$$

where

$$k = \frac{1}{\max(M(:)) - \min(M(:))} \quad \text{and}$$
$$d = -k \min(M(:)).$$

The interval `[0,1]` is chosen because it is the default display range for a grayscale image when using the function `imshow(M)`. The case where $\max(M(:)) = \min(M(:))$ is not relevant, as it would mean that every pixel of the texture has the same tone of gray and can therefore be excluded.

Lemma 4.1 *Let M have low displacement rank $r_{A,B}(M)$ and M^P be defined as in equation* (44)*. Then $r_{A,B}(M^P) \leq r_{A,B}(M) + 2$.*

Proof. The transformed matrix M^P, which has only values in the intervall $[0, 1]$, is given by $M^P = kM + D$, where k is the scalar from above and D is a constant matrix of the same size as M with all entries equal to d from above. For the Stein displacement operator, this yields

$$\begin{aligned}\Delta_{A,B}(M^P) &= kM + D - A(kM + D)B \\ &= k(M - AMB) + D - ADB.\end{aligned}$$

$M - AMB$ has a rank of $r_{A,B}(M)$, D has a rank of 1 (unless it's the null matrix), and the maximum rank of ADB is 1. Consequently the rank of the sum of these three terms is bound by $r_{A,B}(M) + 2$. One can easily verify the same results for the Sylvester type displacement operator. □

Therefore, at least before the rounding process takes place (see Remark below), the matrix, which is represented by the image, also has a low displacement rank concerning the same displacement operator matrices.

Remark. One has to consider that the images created by `imshow(M,[])` or `imshow(M)` are rounded versions of the actual matrices with low displacement rank, as the computer uses only a finite number, like 256 or 65536, of shades of gray. In fact the formula specifying the colormap index of each data value of the original matrix, when using the function `imshow(M,[])`, is given by [23]

```
cm_index = fix((data-dmin)/(dmax-dmin)*cm_length)+1;
```

where `cm_index` stands for the colormap index, `data` denotes the value of the original matrix M, `dmax` and `dmin` represent $\max(M(:))$ and $\min(M(:))$ respectively, `cm_length` is the length of the colormap, and `fix` rounds the value in the brackets towards zero. One can easily verify that the linear transformation (44) is equivalent to the term in the outer brackets with `cm_length` equal to 1. The rounding process `fix` will, in many cases, destroy the structure of the matrix but will generate a matrix which is very close to a structured matrix.

4.1.3 Results

The following images will show a small sample of the structured textures that can be generated by solving the continuous-time Lyapunov equation (42) for the Sylvester type displacement operator or the discrete-time Lyapunov equation (43) for the Stein type displacement operator. The caption of each group of images will indicate which type of displacement operator and what kind of operator matrices were used. Be aware that a group of images may extend more than one page. The caption of each image of a structured matrix M denotes the generator matrices G and H of the corresponding structured matrix. The unknown operator matrices will be characterized throughout the text. One can find a rough description of the generator matrices and the corresponding displacement rank r in Table 3.

Table 3: Generator matrices G and H

Abbreviation	**G**	**H**	**r**
GH_1	Random	Random	1
GH_2	Random	Random	15
GH_3	Sinus	x	1
GH_4	Sinus	Cosinus	1
GH_5	Sinus(Random)	Cosinus(Random)	1
GH_6	Sinus(Random)	Cosinus(Random)	1
GH_7	Triangle	Triangle	5
GH_8	Triangle	Triangle	15
GH_9	Chess	Chess	5
GH_{10}	Chess	Chess	15
GH_{11}	Stripes	Stripes	1
GH_{12}	Exp(Random)+ +Log(Random)	Exp(Random)* *Log(Random)	1
GH_{13}	Ones	Ones	1
GH_{14}	Sinus*Cosinus	Sinus*Cosinus	10
GH_{15}	Random	Random	5

Random denotes a matrix of size $n \times r$ with random entries, where n stands for the dimension of the original matrix M. Exp and Log signify the exponential function and the natural logarithm respectively. Ones is a matrix of size $n \times r$ with all entries being equal to 1. The $n \times 1$ vector Stripes and the $n \times r$ matrices Chess and Triangle are given as follows

$$\text{Stripes} = [\overbrace{1 \cdots 1}^{5 \text{ times}} \ \overbrace{0 \cdots 0}^{5 \text{ times}} \ \overbrace{1 \cdots 1}^{5 \text{ times}} \ 0 \cdots]^T,$$

$$\text{Chess} = \begin{pmatrix} r_1 & 0 & r_2 & \cdots \\ 0 & r_k & 0 & \cdots \\ r_l & 0 & r_{l+1} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \text{ with } r_i \text{ random, and}$$

$$\text{Triangle} = \begin{pmatrix} 1 & & & \\ & \ddots & & \\ & & & 1 \\ & & \iddots & \\ 1 & & & \\ & \ddots & & \end{pmatrix}, \text{ with blank areas equal to zero.}$$

In each row, the first image represents the structured matrix M, which has low displacement rank with regard to the indicated type and operator matrices. Its displacement can be described by the denoted generator matrices. For example, if an image has the caption "(c) Struct. matrix M with GH_2" and the respective group of images has the caption "Figure 7: Stein with $A = Z_0$ and $B = Z_0$", then for the structured matrix M the image symbolizes the following is true

$$\Delta_{Z_0,Z_0}(M) = M - Z_0 M Z_0 = GH_2,$$

and M has displacement rank equal to 15 (see Table 3).

Only if beneficial, a logtransformed matrix M_{log} of M is added to make

patterns more apparent.

The last image in each row illustrates the absolute value of the 2-dimensional discrete Fourier transform (2-D DFT) of M. The origin of the transform is moved to the center of the picture. This means that the low frequencies can be found in the middle and the high frequencies at the border of the image. Before the Fourier transform was applied, the mean of the matrix was subtracted; hence the dc is equal to zero. The unitary discrete Fourier transform matrix was used to preserve the energy of the matrix.

```
dftmat = 1/sqrt(n)*dftmtx(n);
dft_M = dftmat*(M-mean(mean(M)))*dftmat;                    (45)
shift_dft_M = abs(fftshift(dft_M));
```

where n is equal to the dimension of the matrix M.

All matrices are displaced through the function `imshow(M,[])`.

Figure 6 shows some Toeplitz-like matrices. Therefore the Stein displacement operator with $A = Z_0$ and $B = Z_0^T$ is used.

Figure 6: Stein with $A = Z_0$ and $B = Z_0^T$

(a) Struct. matrix M with GH_1 (b) 2-D DFT of M

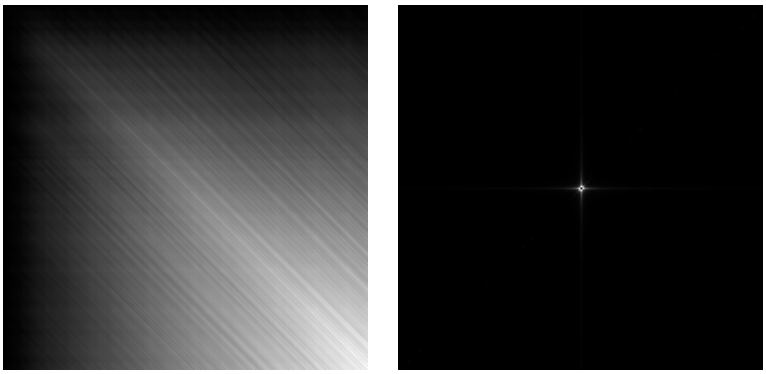
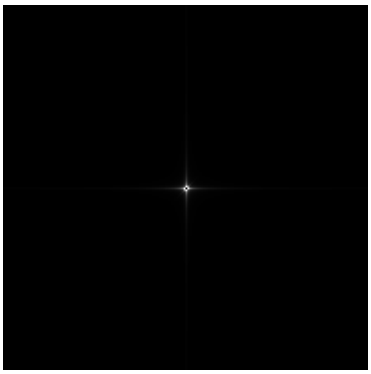


Figure 6: Stein with $A = Z_0$ and $B = Z_0^T$ (Continued)

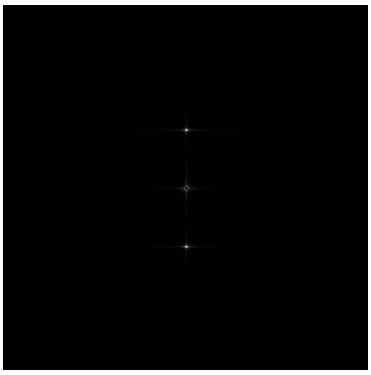
(c) Struct. matrix M with GH_2

(d) 2-D DFT of M

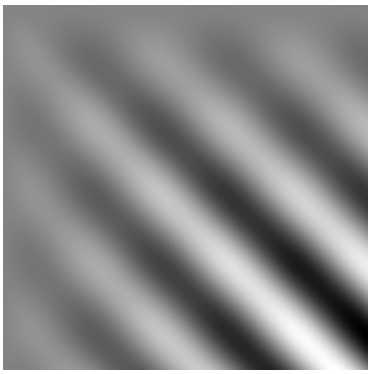


(e) Struct. matrix M with GH_3

(f) 2-D DFT of M



(g) Struct. matrix M with GH_4



(h) 2-D DFT of M

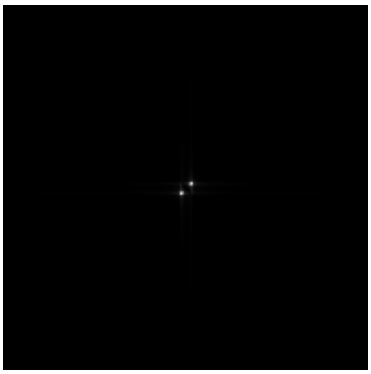
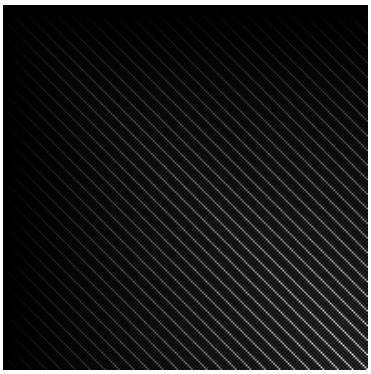
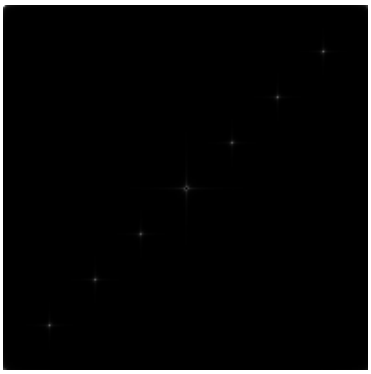


Figure 6: Stein with $A = Z_0$ and $B = Z_0^T$ (Continued)

(i) Struct. matrix M with GH_7

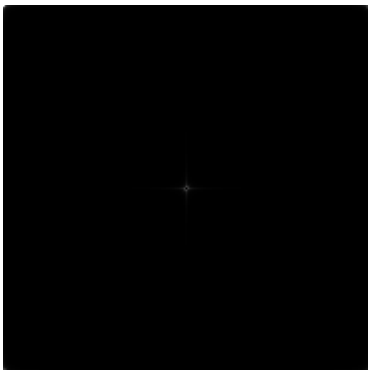


(j) 2-D DFT of M

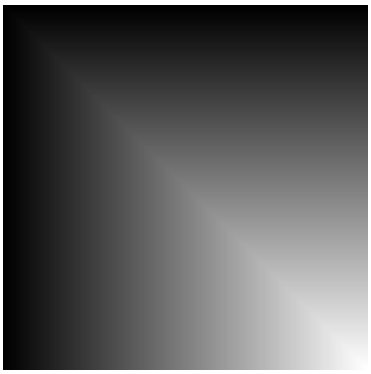


(k) Struct. matrix M with GH_{10}

(l) 2-D DFT of M



(m) Struct. matrix M with GH_{13}



(n) 2-D DFT of M

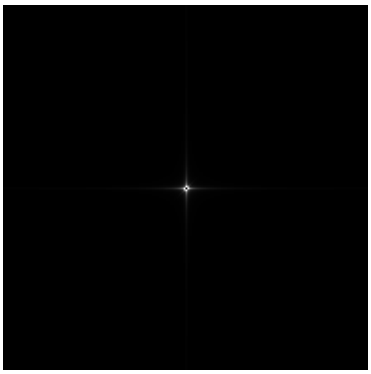
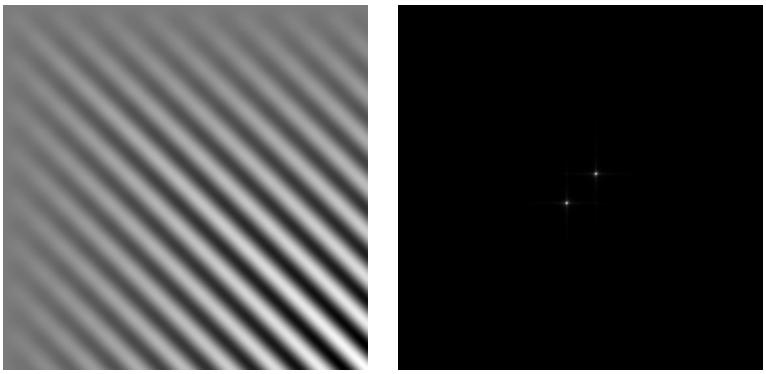


Figure 6: Stein with $A = Z_0$ and $B = Z_0^T$ (Continued)

(o) Struct. matrix M with GH_{14} (p) 2-D DFT of M



Since Hankel matrices are just horizontally flipped Toeplitz matrices, Hankel-like matrices are similar to Toeplitz-like matrices. A few examples of them are displayed in Figure 7.

Figure 7: Stein with $A = Z_0$ and $B = Z_0$

(a) Struct. matrix M with GH_3 (b) 2-D DFT of M

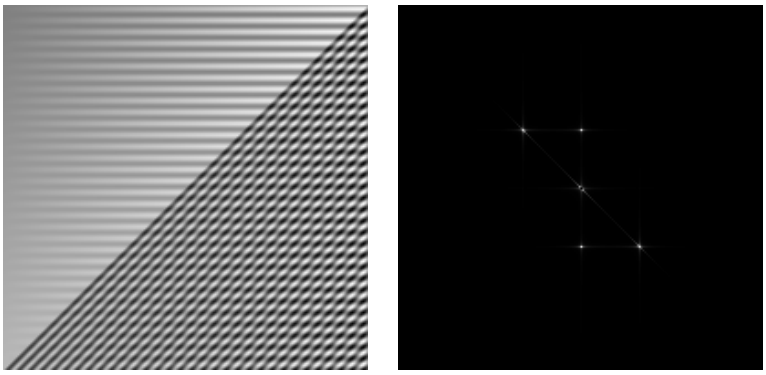
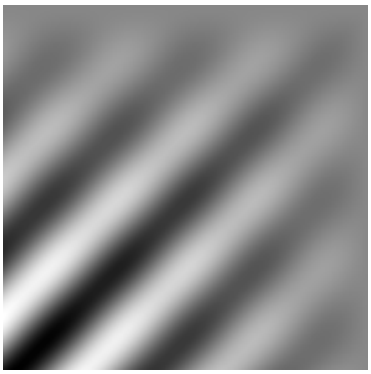
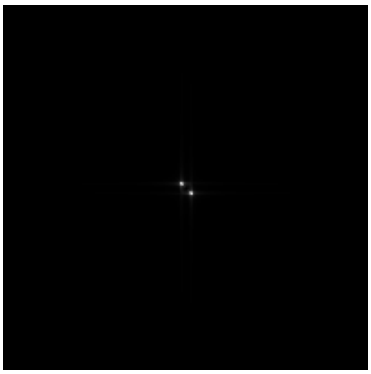


Figure 7: Stein with $A = Z_0$ and $B = Z_0$ (Continued)

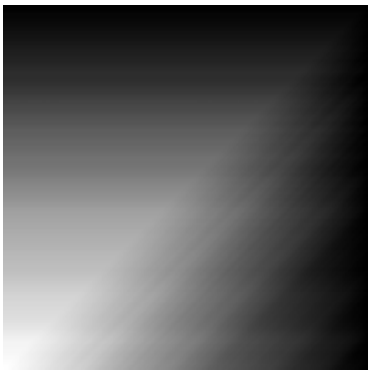
(c) Struct. matrix M with GH_4



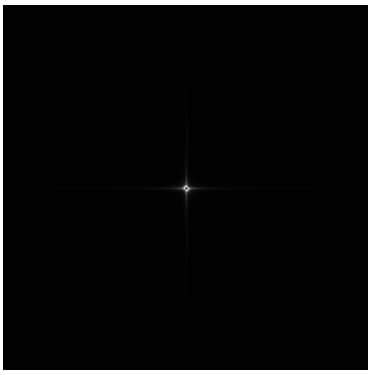
(d) 2-D DFT of M



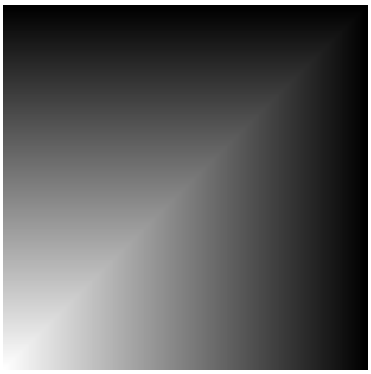
(e) Struct. matrix M with GH_6



(f) 2-D DFT of M



(g) Struct. matrix M with GH_{13}



(h) 2-D DFT of M

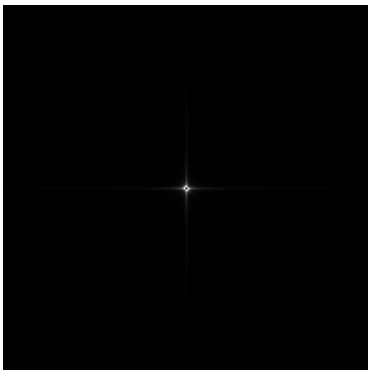
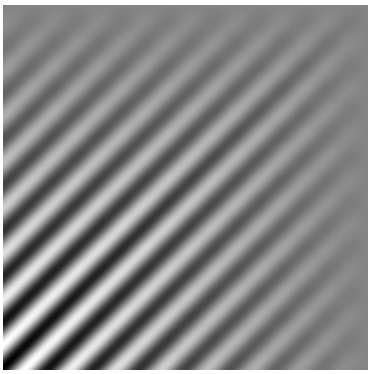
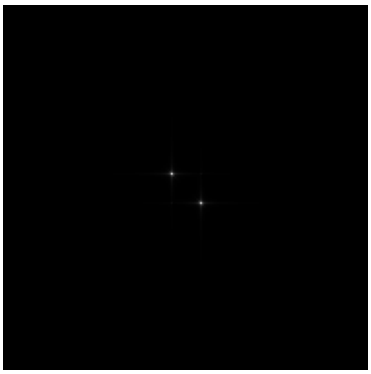


Figure 7: Stein with $A = Z_0$ and $B = Z_0$ (Continued)

(i) Struct. matrix M with GH_{14}

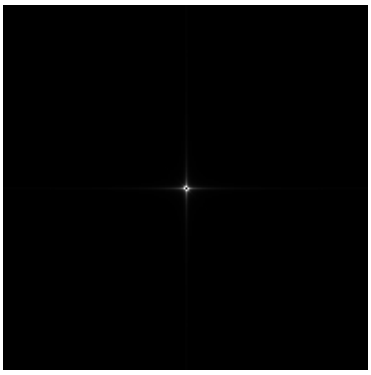


(j) 2-D DFT of M



(k) Struct. matrix M with GH_{15}

(l) 2-D DFT of M

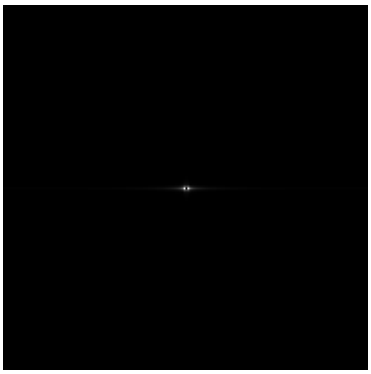


In Figure 8, the operator matrices $Y_{00} = Z_0 + Z_0^T$ and $Y_{11} = Y_{00} + \mathbf{e}_1\mathbf{e}_1^T + \mathbf{e}_n\mathbf{e}_n^T$ are used. They are used in a Sylvester type displacement operator for Toeplitz+Hankel-like matrices.

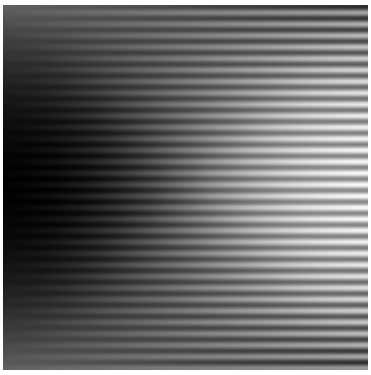
Figure 8: Sylvester with $A = Y_{00}$ and $B = Y_{11}$

(a) Struct. matrix M with GH_1

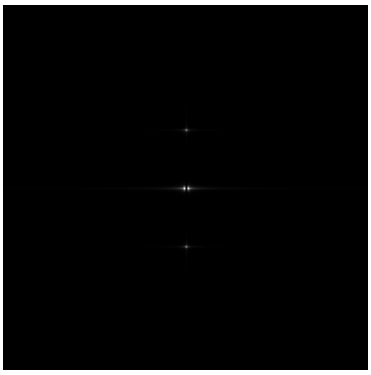
(b) 2-D DFT of M



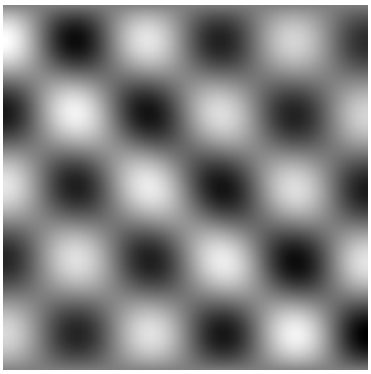
(c) Struct. matrix M with GH_3



(d) 2-D DFT of M



(e) Struct. matrix M with GH_4



(f) 2-D DFT of M

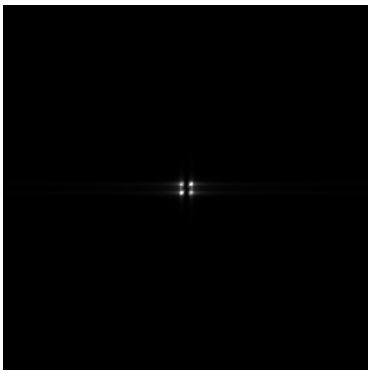
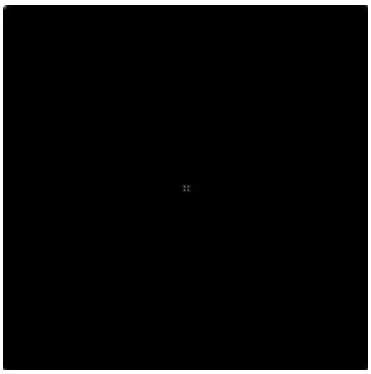


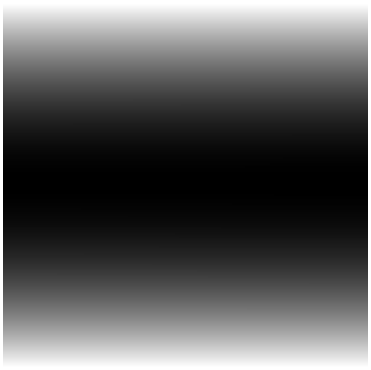
Figure 8: Sylvester with $A = Y_{00}$ and $B = Y_{11}$ (Continued)

(g) Struct. matrix M with GH_5

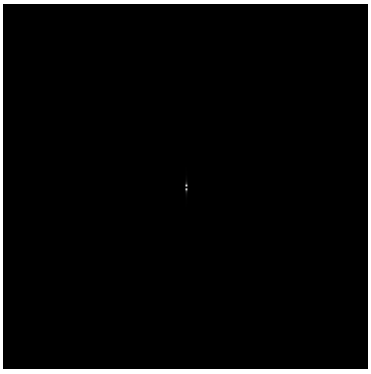
(h) 2-D DFT of M



(i) Struct. matrix M with GH_6



(j) 2-D DFT of M



(k) Struct. matrix M with GH_7

(l) 2-D DFT of M

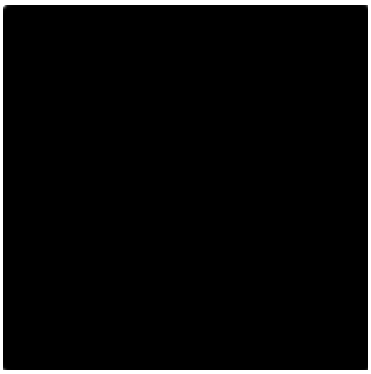


Figure 8: Sylvester with $A = Y_{00}$ and $B = Y_{11}$ (Continued)

(m) Struct. matrix M with GH_{14} (n) 2-D DFT of M

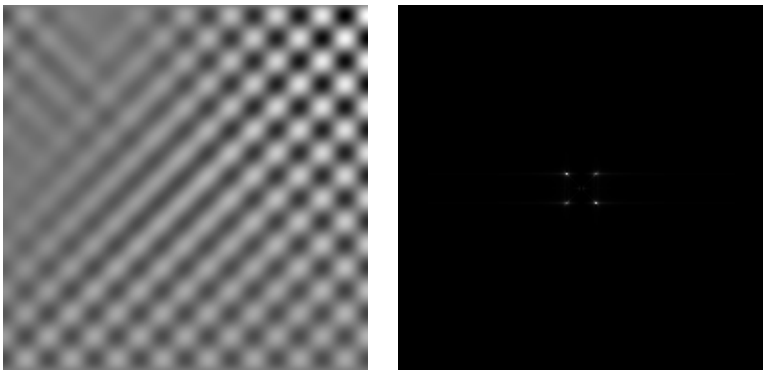


Figure 9 uses the operator matrices of another basic structured matrix, the Cauchy matrix. Both operator matrices are diagonal and have random entries along their diagonals. Hence, $A = \text{diag}(\mathbf{w})$ and $B = \text{diag}(\mathbf{v})$, where $\mathbf{w}$ and $\mathbf{v}$ are random vectors.

Figure 9: Sylvester with $A = \text{diag}(\mathbf{w})$, $B = \text{diag}(\mathbf{v})$, and $\mathbf{w}$ and $\mathbf{v}$ random

(a) Struct. matrix M with GH_2 (b) 2-D DFT of M

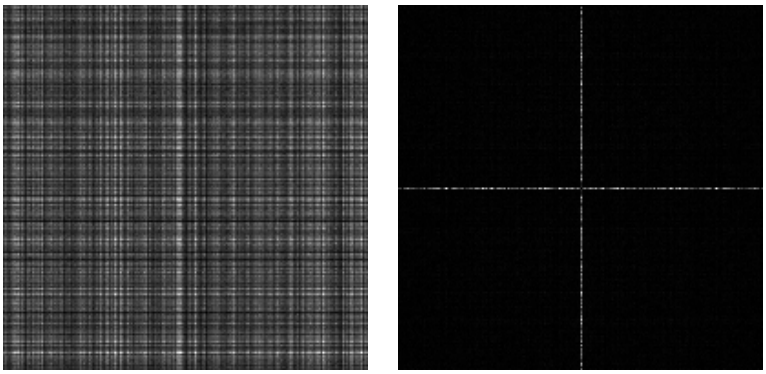
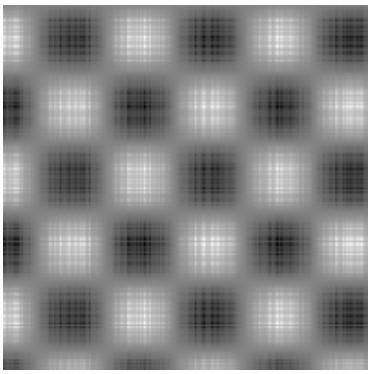
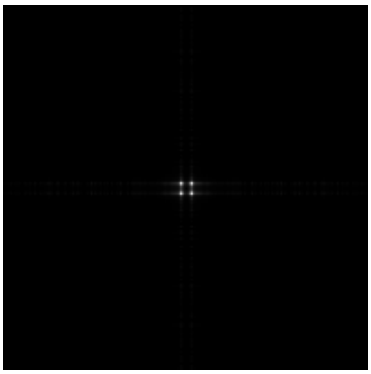


Figure 9: Sylvester with $A = \text{diag}(\mathbf{w})$, $B = \text{diag}(\mathbf{v})$, and $\mathbf{w}$ and $\mathbf{v}$ random (Continued)

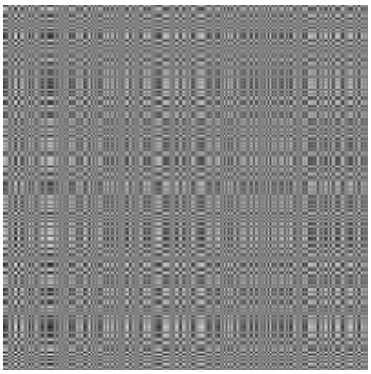
(c) Struct. matrix M with GH_4



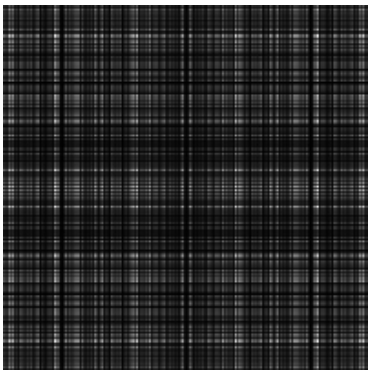
(d) 2-D DFT of M



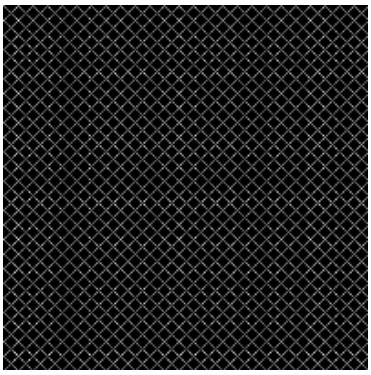
(e) Struct. matrix M with GH_5



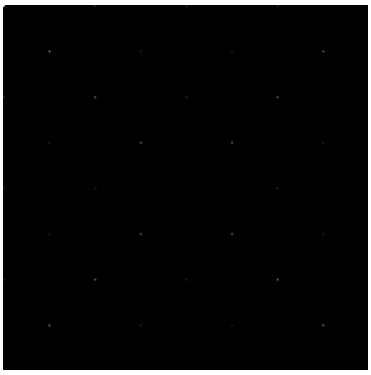
(f) 2-D DFT of M



(g) Struct. matrix M with GH_7



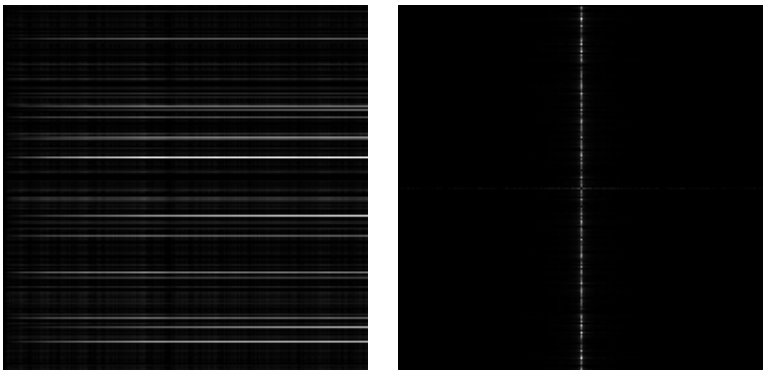
(h) 2-D DFT of M



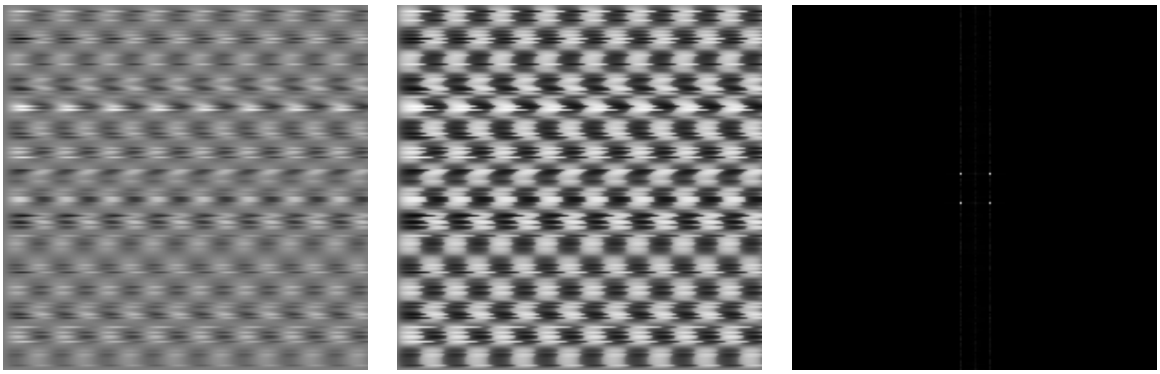
Vandermonde-like matrices can be seen in Figure 10. The Stein type displacement operator is utilized with a diagonal matrix with random entries and a shift matrix as generator matrices.

Figure 10: Stein with $A = \text{diag}(\mathbf{w})$, $B = Z_0^T$, and $\mathbf{w}$ random

(a) Struct. matrix M with GH_1 (b) 2-D DFT of M



(c) Struct. matrix M with GH_{14} (d) Logtransf. matrix of M (e) 2-D DFT of M



The next operator matrix has not yet appeared in the literature of displacement operators. It is similar to the shift matrix Z_0 but the line of ones is bent. Figure 11 illustrates the new operator matrix Y_C. The black area is all zeros and the curved white line represents the ones.

As operator matrices, Y_C and the transpose of Y_C are used in a Stein displacement operator in Figure 12.

Figure 11: Operator matrix Y_C

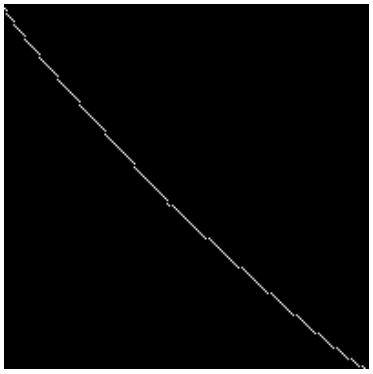
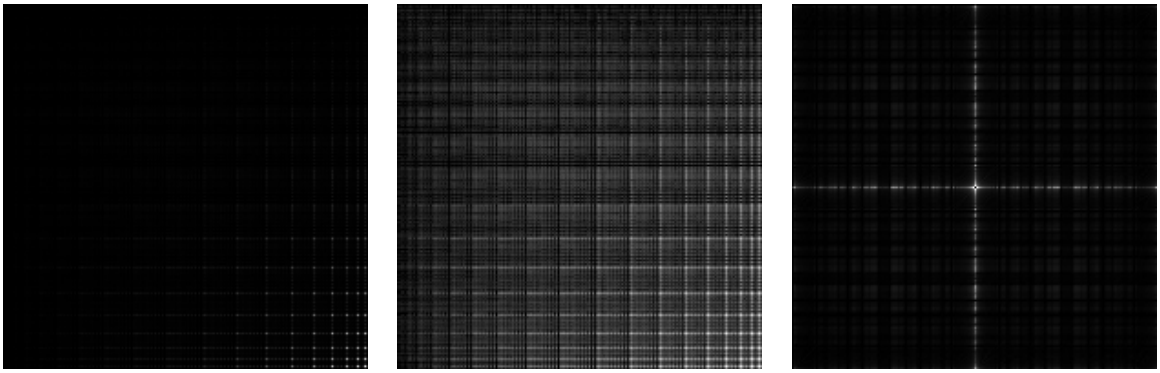


Figure 12: Stein with $A = Y_C$ and $B = Y_C^T$

(a) Struct. matrix M with GH_1 (b) Logtransf. matrix of M (c) 2-D DFT of M



(d) Struct. matrix M with GH_2 (e) Logtransf. matrix of M (f) 2-D DFT of M

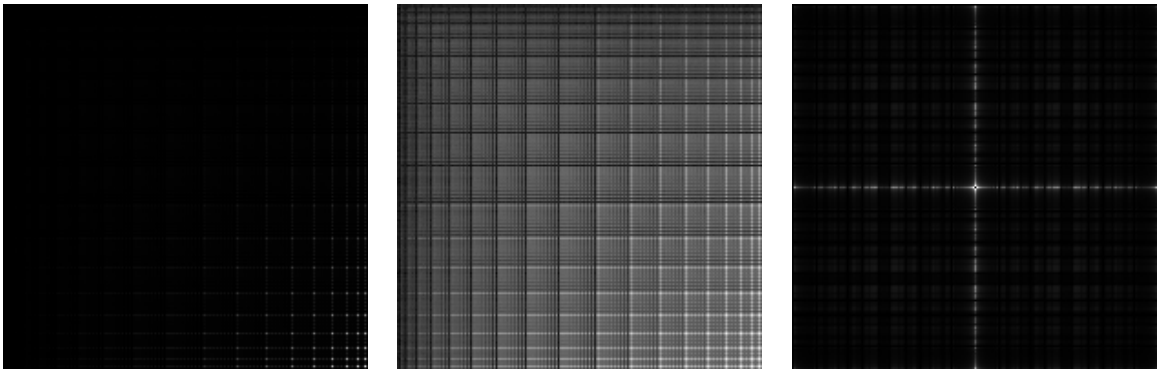
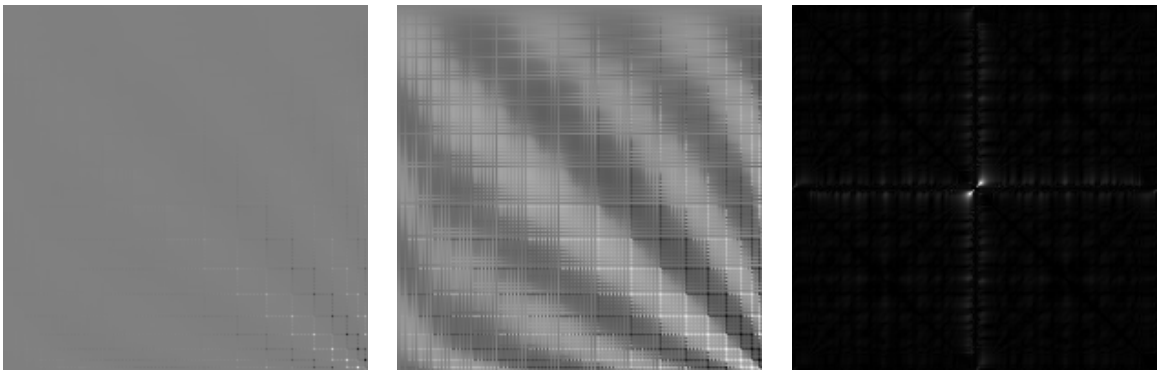
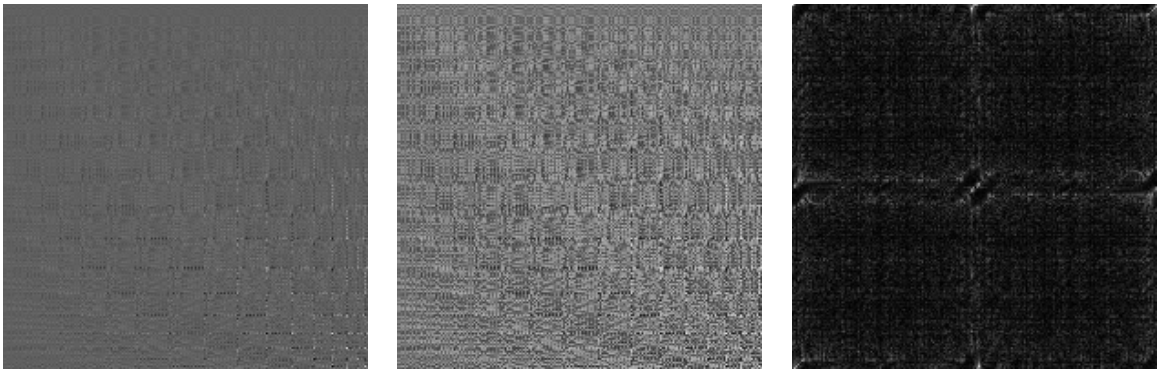


Figure 12: Stein with $A = Y_C$ and $B = Y_C^T$ (Continued)

(g) Struct. matrix M with GH_4 (h) Logtransf. matrix of M (i) 2-D DFT of M



(j) Struct. matrix M with GH_5 (k) Logtransf. matrix of M (l) 2-D DFT of M



(m) Struct. matrix M with GH_7 (n) Logtransf. matrix of M (o) 2-D DFT of M

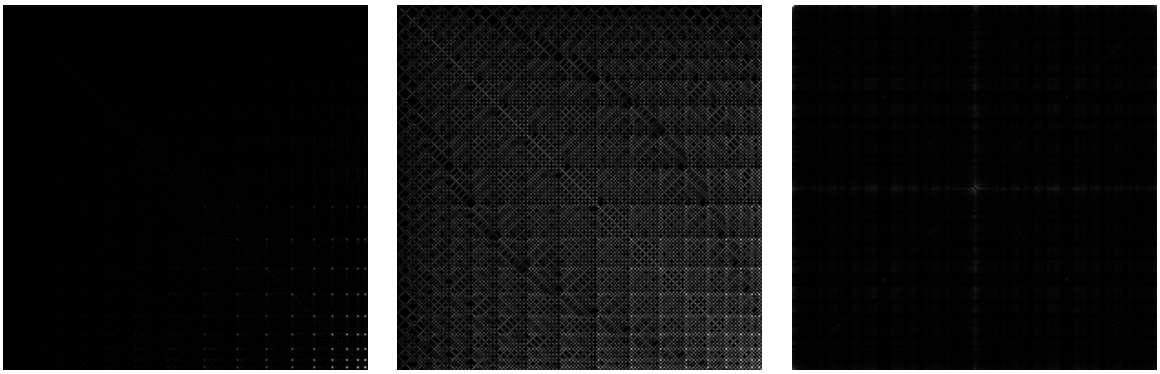
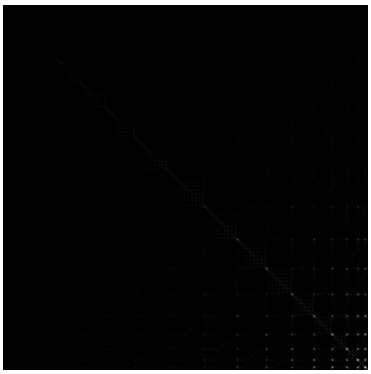
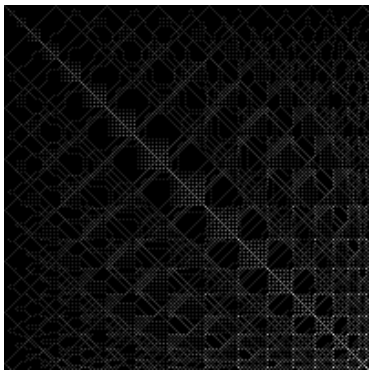


Figure 12: Stein with $A = Y_C$ and $B = Y_C^T$ (Continued)

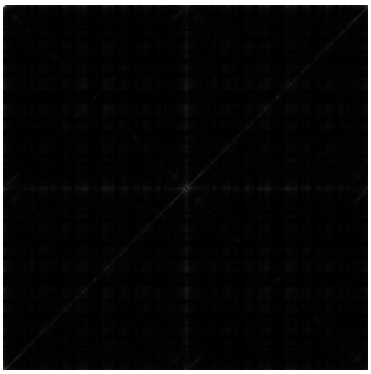
(p) Struct. matrix M with GH_8



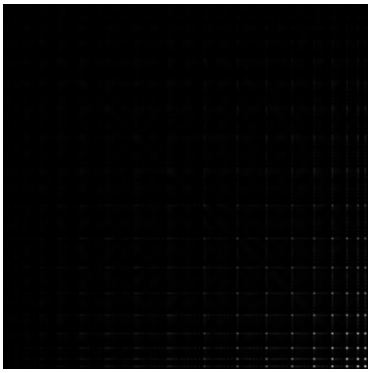
(q) Logtransf. matrix of M



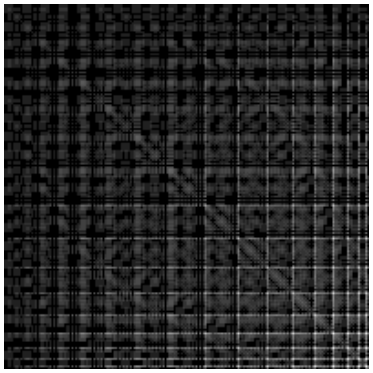
(r) 2-D DFT of M



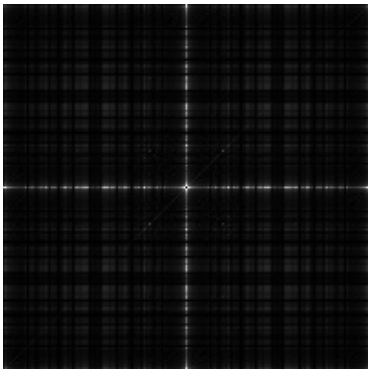
(s) Struct. matrix M with GH_{11}



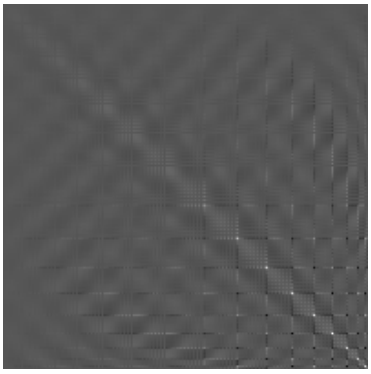
(t) Logtransf. matrix of M



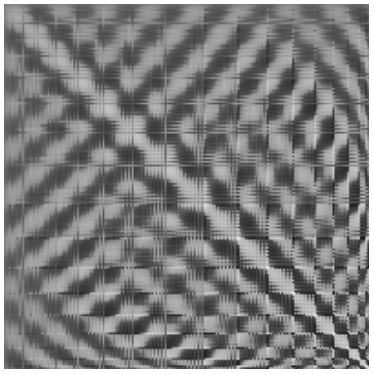
(u) 2-D DFT of M



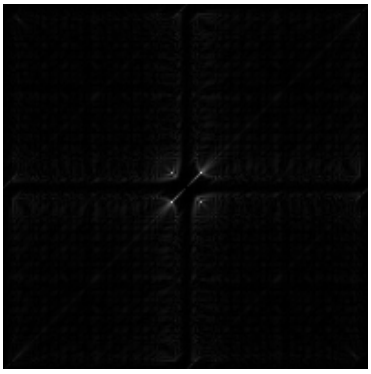
(v) Struct. matrix M with GH_{14}



(w) Logtransf. matrix of M



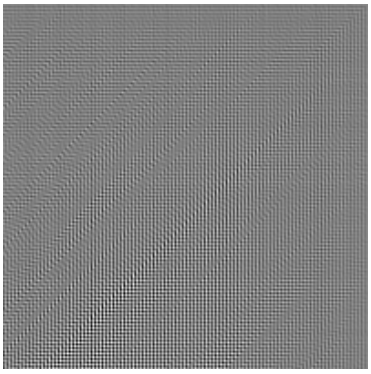
(x) 2-D DFT of M



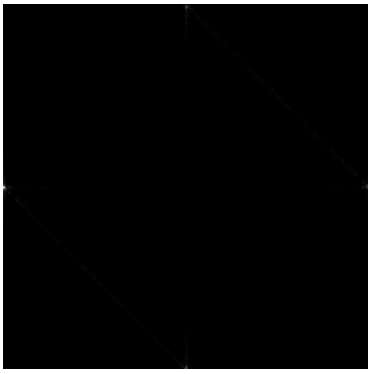
The applied operator matrices in Figure 13 are $-Z_0$ and Z_0.

Figure 13: Stein with $A = -Z_0$ and $B = Z_0$

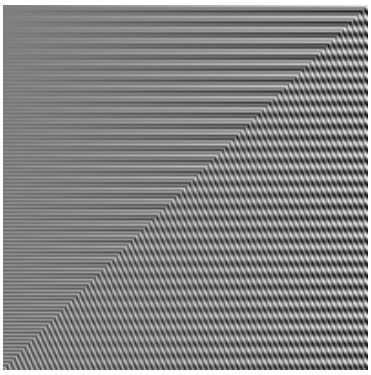
(a) Struct. matrix M with GH_2



(b) 2-D DFT of M



(c) Struct. matrix M with GH_3



(d) 2-D DFT of M

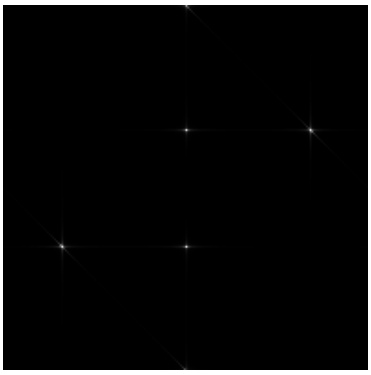
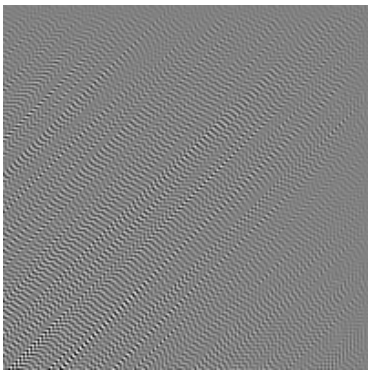
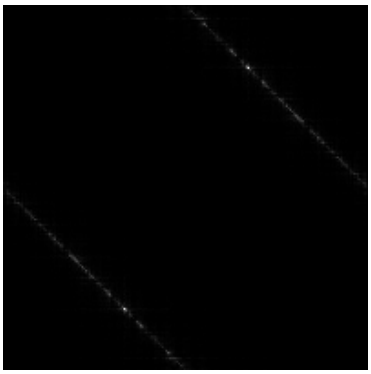


Figure 13: Stein with $A = -Z_0$ and $B = Z_0$ (Continued)

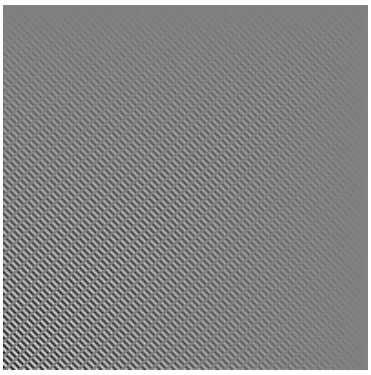
(e) Struct. matrix M with GH_5



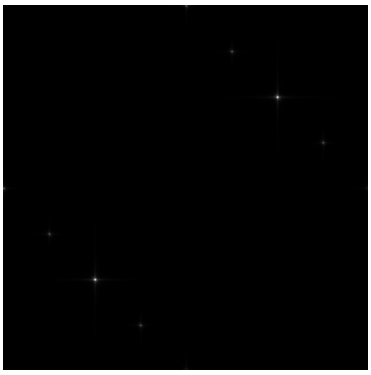
(f) 2-D DFT of M



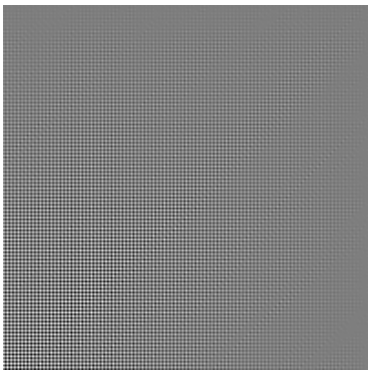
(g) Struct. matrix M with GH_7



(h) 2-D DFT of M



(i) Struct. matrix M with GH_9



(j) 2-D DFT of M

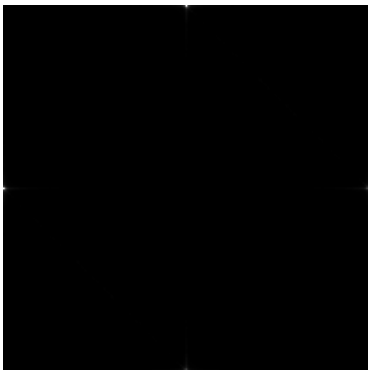
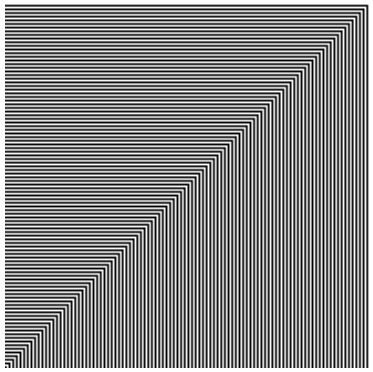
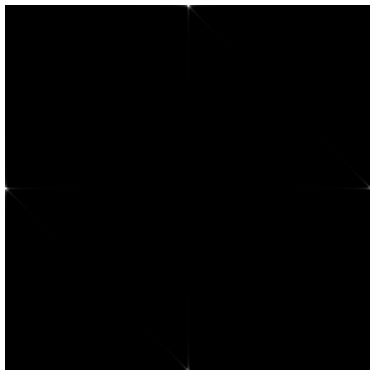


Figure 13: Stein with $A = -Z_0$ and $B = Z_0$ (Continued)

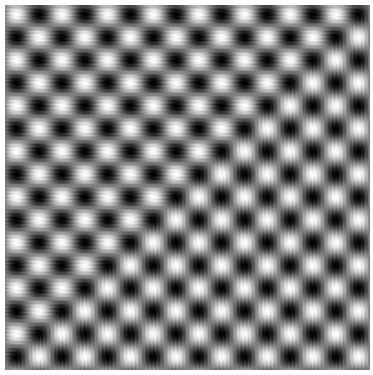
(k) Struct. matrix M with GH_{13}



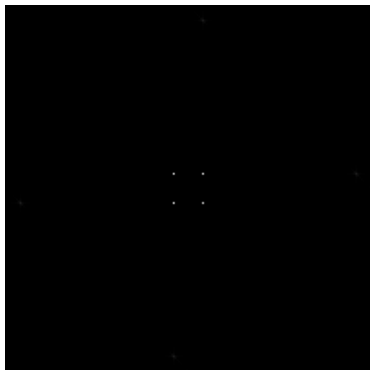
(l) 2-D DFT of M



(m) Struct. matrix M with GH_{14}



(n) 2-D DFT of M



Here a new operator matrix is introduced. It's again similar to a shift matrix, but this time the line of ones is rotated. As before, an image of the matrix best clarifies the structure of the new operator matrix X_f, where f is the angle between the upper border of the image and the line of ones. In Figure 14, the black area and the white line denote zeros and ones respectively.

Figure 15 demonstrates matrices which have a low displacement rank with respect to a Stein type displacement operator, with the operator matrices being equal to X_{80} and X_{80}^T.

Figure 14: Operator matrix X_f with $f = 80$

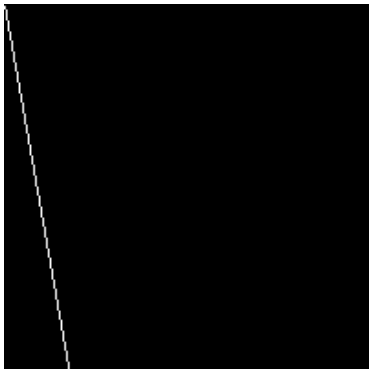
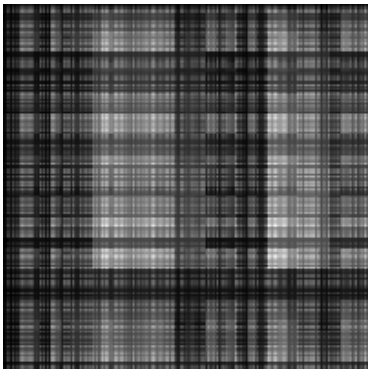
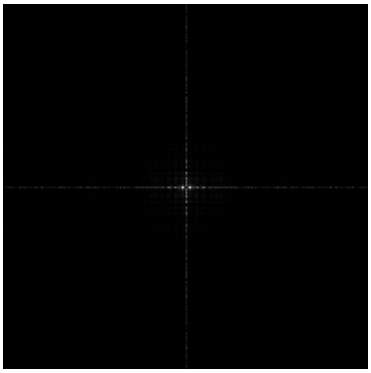


Figure 15: Stein with $A = X_{80}$ and $B = X_{80}^T$

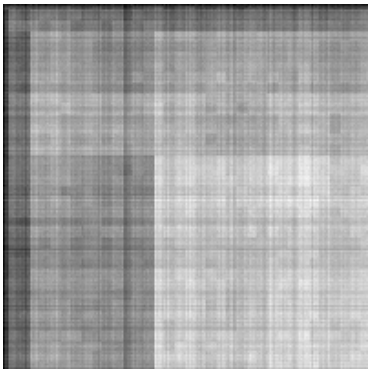
(a) Struct. matrix M with GH_1



(b) 2-D DFT of M



(c) Struct. matrix M with GH_2



(d) 2-D DFT of M

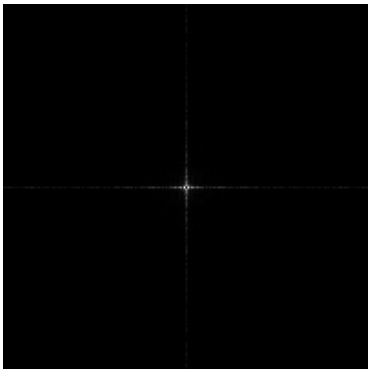
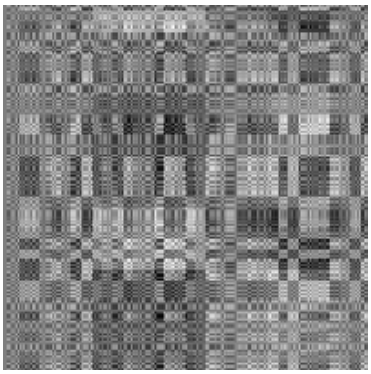
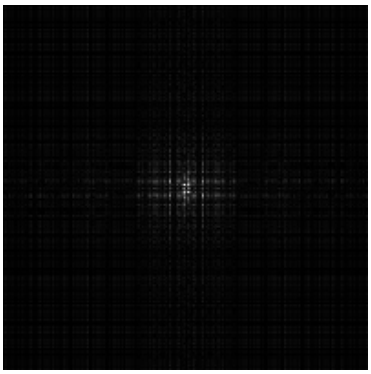


Figure 15: Stein with $A = X_{80}$ and $B = X_{80}^T$ (Continued)

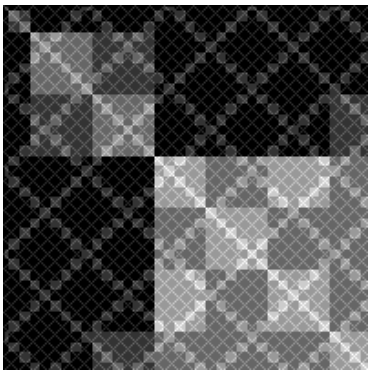
(e) Struct. matrix M with GH_5



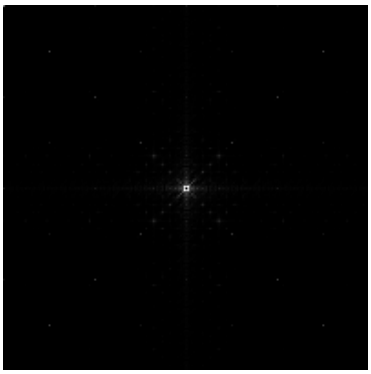
(f) 2-D DFT of M



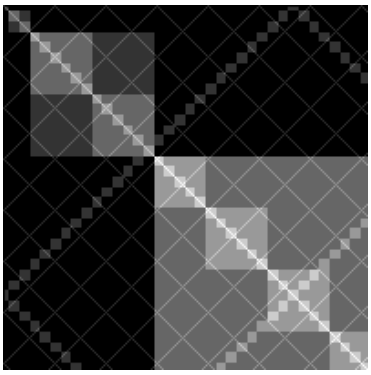
(g) Struct. matrix M with GH_7



(h) 2-D DFT of M



(i) Struct. matrix M with GH_8



(j) 2-D DFT of M

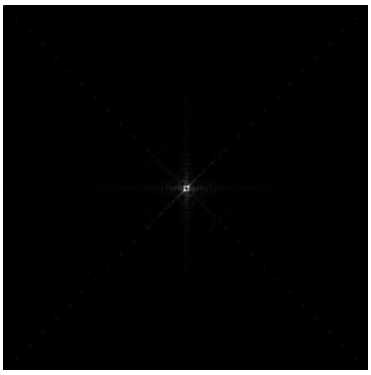
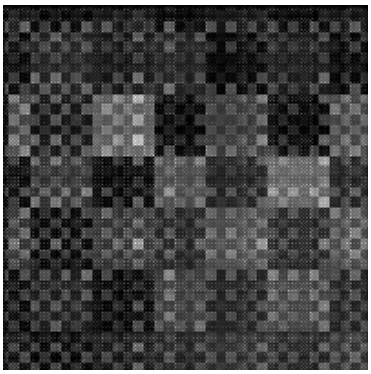
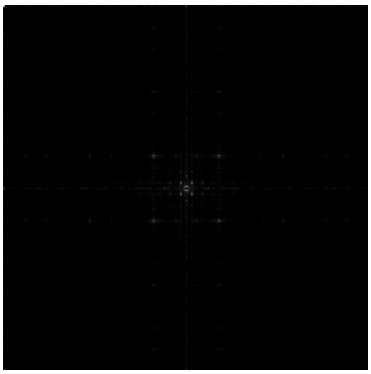


Figure 15: Stein with $A = X_{80}$ and $B = X_{80}^T$ (Continued)

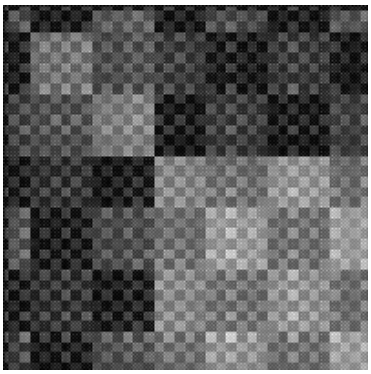
(k) Struct. matrix M with GH_9



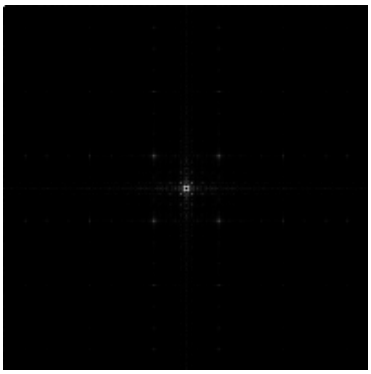
(l) 2-D DFT of M



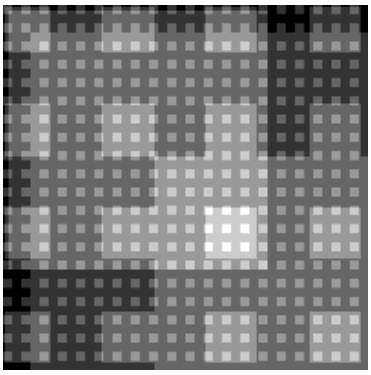
(m) Struct. matrix M with GH_{10}



(n) 2-D DFT of M



(o) Struct. matrix M with GH_{11}



(p) 2-D DFT of M

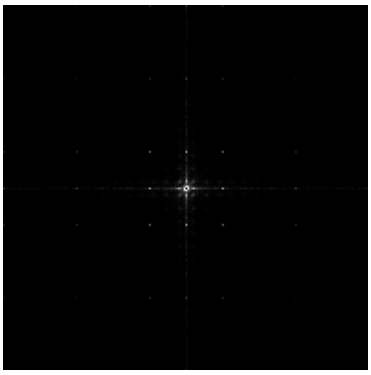
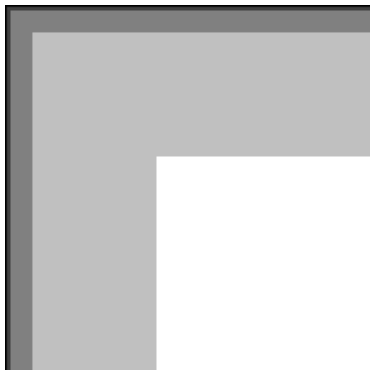
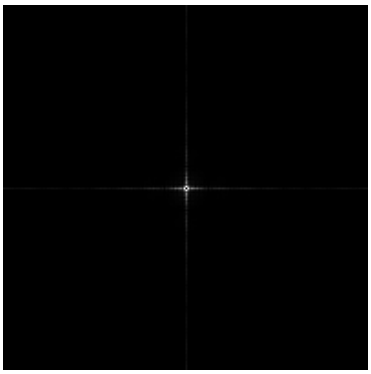


Figure 15: Stein with $A = X_{80}$ and $B = X_{80}^T$ (Continued)

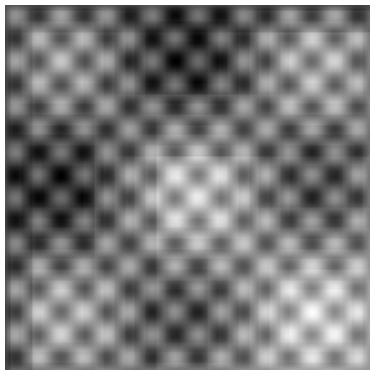
(q) Struct. matrix M with GH_{13}



(r) 2-D DFT of M



(s) Struct. matrix M with GH_{14}



(t) 2-D DFT of M

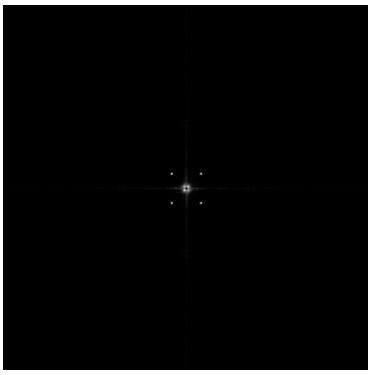


Figure 16 displays two more operator matrices. The first, W, is a wiggly line of ones along the main diagonal. The second, C, is a centered circle of ones.

Figure 16: Operator matrices W and C

(a) (b)

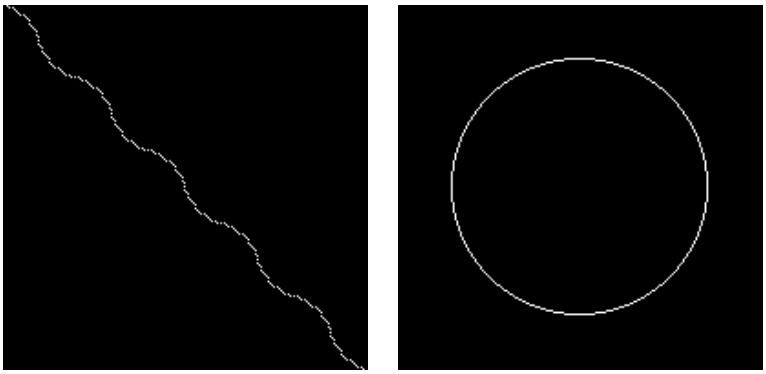


Figure 17 shows the structured matrices that solve the Stein type displacement operator with $A = W$ and $B = C$.

Figure 17: Stein with $A = W$ and $B = C$

(a) Struct. matrix M with GH_5 (b) Logtransf. matrix of M (c) 2-D DFT of M

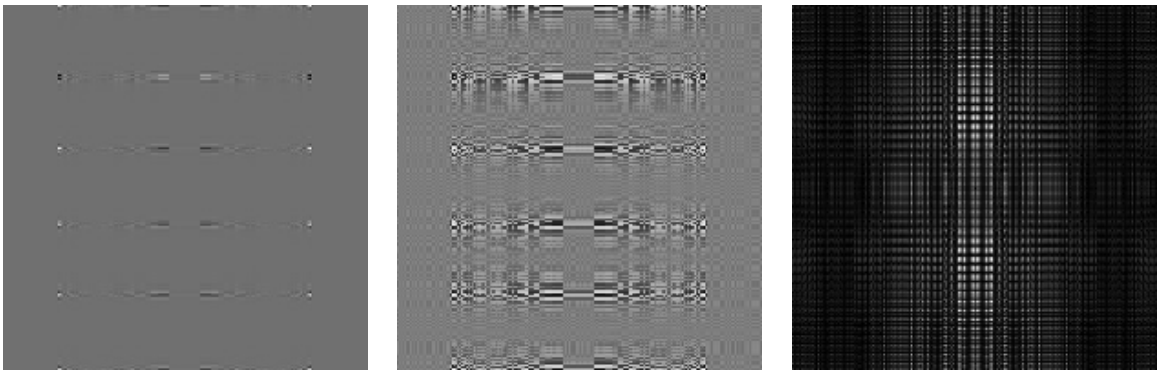
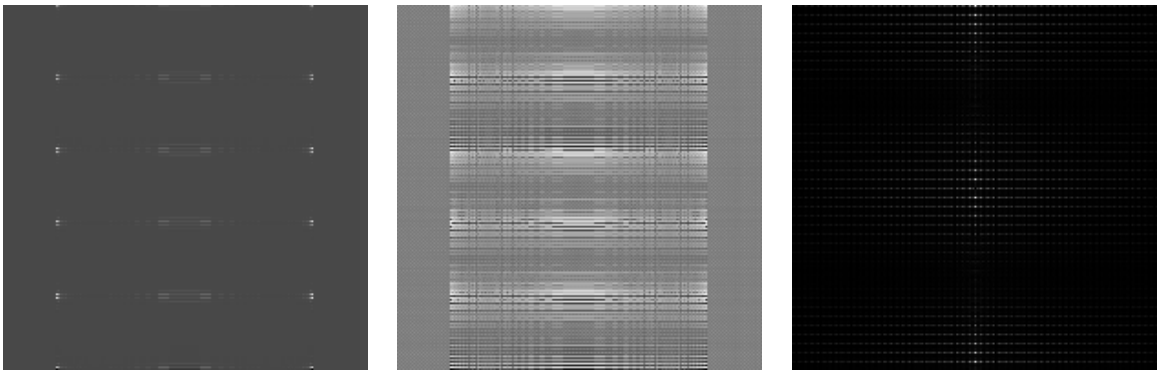


Figure 17: Stein with $A = W$ and $B = C$ (Continued)

(d) Struct. matrix M with GH_{10} (e) Logtransf. matrix of M (f) 2-D DFT of M



The next Stein type displacement operator in Figure 19 uses the already known operator matrix X_{80}^T and a new operator matrix V, which has vertical lines of ones, as shown in Figure 18.

Figure 18: Operator matrix V

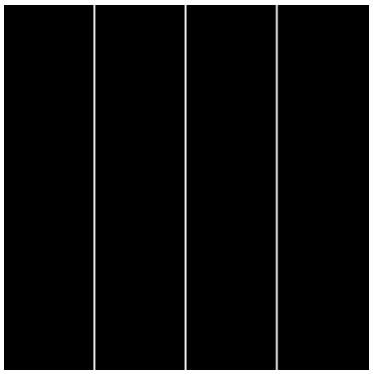
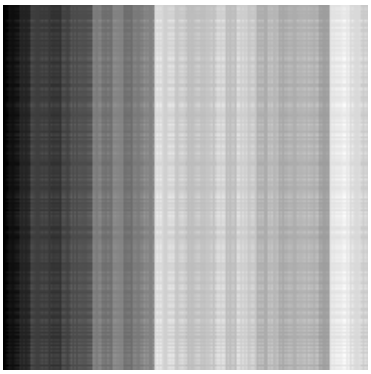
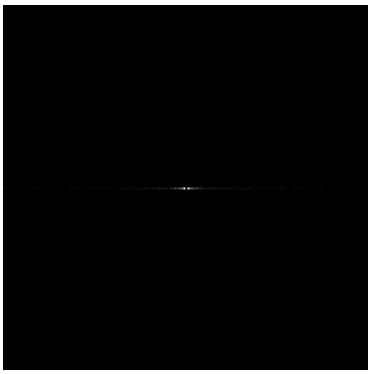


Figure 19: Stein with $A = V$ and $B = X_{80}^T$

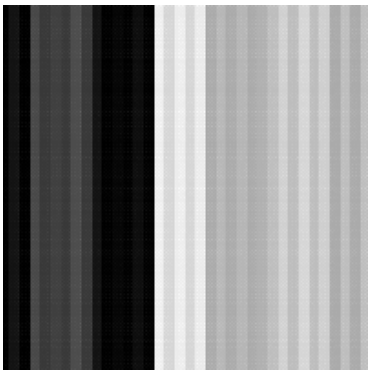
(a) Struct. matrix M with GH_1



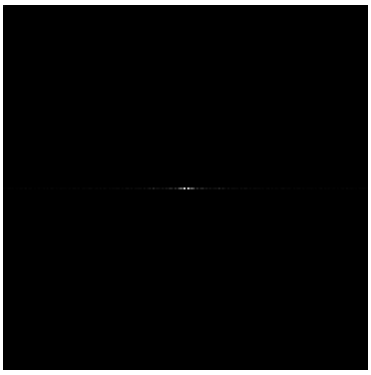
(b) 2-D DFT of M



(c) Struct. matrix M with GH_9



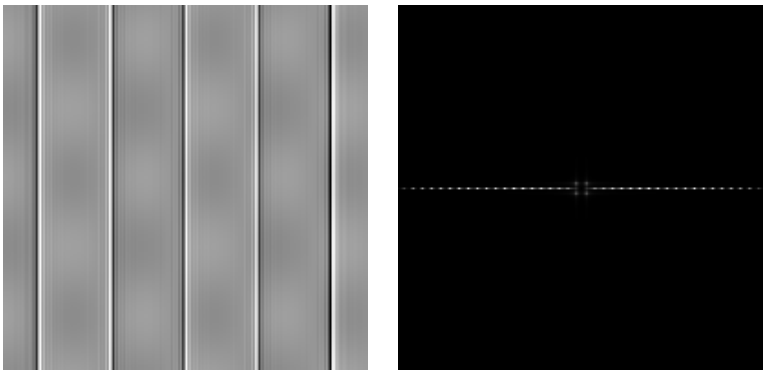
(d) 2-D DFT of M



The structured matrix M in Figure 20 was created by using the Stein type displacement operator, and V and W as operator matrices.

Figure 20: Stein with $A = V$ and $B = W$

(a) Struct. matrix M with GH_4 (b) 2-D DFT of M



A new operator matrix is introduced in Figure 21. The matrix X^f, with $f \in \{E, R\}$, has entries unequal to zero only along the main- and the anti-diagonal. The entries are either all equal to 1, X^E, or random, X^R. The structured matrices with respect to the Stein type displacement operator Δ_{X^E,X^R} can be seen in Figure 22.

Figure 21: Operator matrix X^E

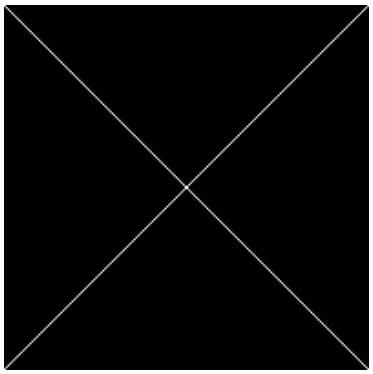
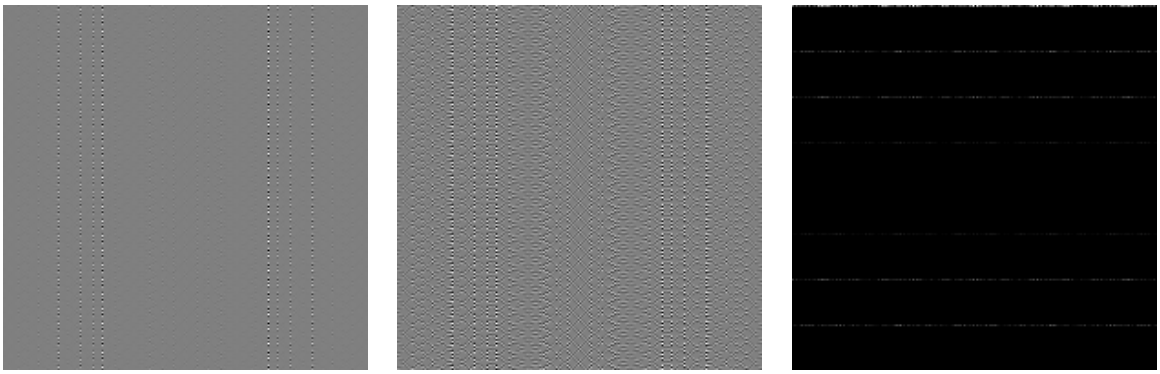
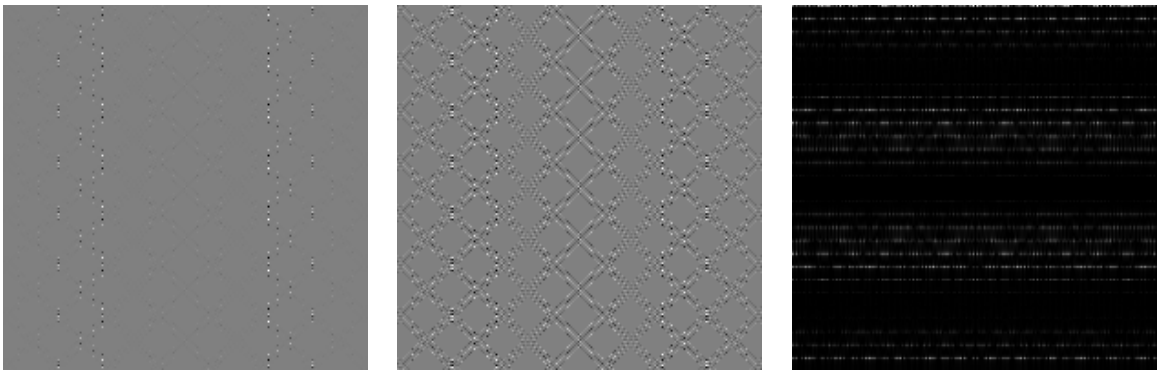


Figure 22: Stein with $A = X^E$ and $B = X^R$

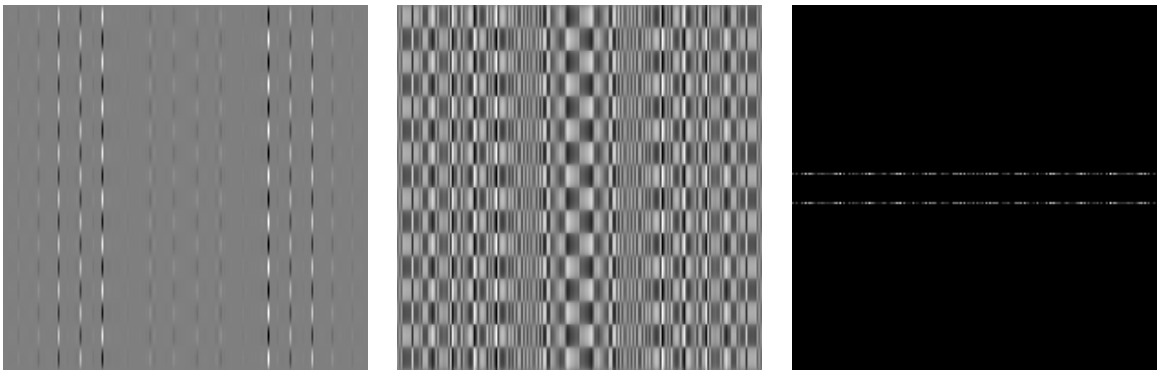
(a) Struct. matrix M with GH_7 (b) Logtransf. matrix of M (c) 2-D DFT of M



(d) Struct. matrix M with GH_8 (e) Logtransf. matrix of M (f) 2-D DFT of M



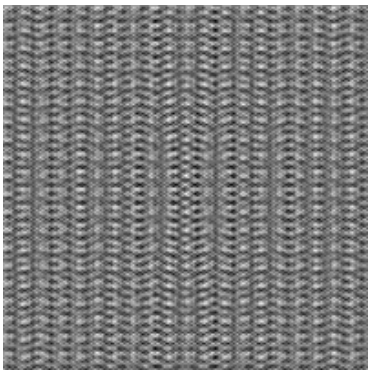
(g) Struct. matrix M with GH_{14} (h) Logtransf. matrix of M (i) 2-D DFT of M



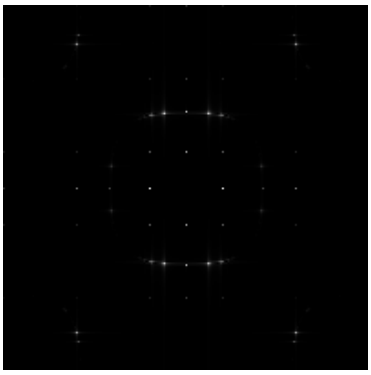
The subsequent operator matrix Y_X is a sum of already known operator matrices. The matrix Y_X is given by the downshift matrix Z_0 plus the horizontally flipped downshift matrix. Therefore it looks similar to the matrix X^E from Figure 21, except that it is shifted down one line. This operator matrix, its transposed, and the Stein type displacement operator yield the structured matrices in Figure 23.

Figure 23: Stein with $A = Y_X$ and $B = Y_X^T$

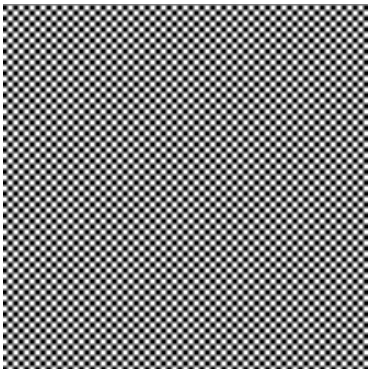
(a) Struct. matrix M with GH_{11}



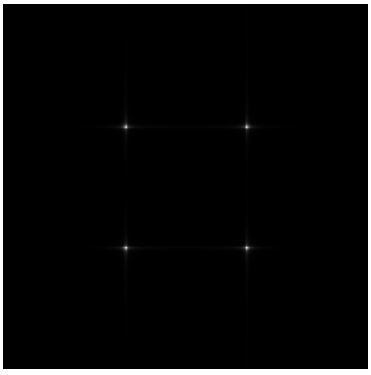
(b) 2-D DFT of M



(c) Struct. matrix M with GH_{15}



(d) 2-D DFT of M



4.1.4 Summary

As one can see, various textures or rather structured matrices can be created by solving the two Lyapunov equations. Every texture has structure with regard to the corresponding displacement operator. Due to the diversity of the results and intuition, one can speculate that for every texture, there exists a displacement operator with two sparse operator matrices that transforms the texture into a displacement with low rank r; however, that still remains to be seen.

One method would be to think of further operator matrices or generator matrices and enlarge the database of structured matrices by hand, but obviously it is impossible to cover every texture in this way. Especially to find a displacement operator for a certain texture, another approach is necessary. For instance, one could begin by using already existing displacement operators and changing certain parameters of the operator or generator matrices to see how the matrix M is impacted. Another method could be to write an algorithm, which captures the structure of the matrix and suggests an operator type and two sparse displacement operators.

Once enough displacement operators are found which cover a sufficiently large range of textures, the process has to be reversed. For a given texture, the ideal displacement operator must be found to indicate which "family" of textures it belongs to. This is a very complex optimization problem and won't be treated in this thesis.

If the structured matrix is slightly disturbed by a noise term, the matrix will most likely lose its structure with respect to the original displacement operator. Therefore it is necessary for the structure to be exact. This drawback leads us to the next section, where structure is not imposed on the matrix itself, but rather the matrix is a Gauss Markov Random Field and structure is imposed on the inverse of its covariance matrix.

4.2 Random Field

In the second approach to describing textures, it is assumed that the texture is generated by a GMRF model. In this case, the texture itself doesn't have a low displacement rank, but rather the set of matrices which make up the precision matrix have low displacement ranks with respect to certain displacement operators.

First a framework is introduced, in which there are only vertical and horizontal field interactions. Thus the precision matrix of size $n^2 \times n^2$ is highly sparse and can be defined by just 2 matrices of size $n \times n$, which are assumed to have a low displacement rank. The noncausal autoregressive field representation from section 3.4.4 is utilized to describe the GMRF, and the one-sided characterization from section 3.4.5 enables the generation of drawings from the GMRF.

Due to restrictions on the matrices that compose the precision matrix, a modification of the elementary model is presented. Thereby more structured matrices can be applied in the compound of the precision matrix. Multiple textures are shown at the end of this section to demonstrate the potential of this approach.

4.2.1 Basic Setup

As already indicated, it is supposed that the *precision matrix is a compound of matrices with low displacement rank*. This precision matrix is the main parameter of a GMRF, which takes values on a 2-D lattice and symbolizes the texture. Without loss of generality, it is assumed that the GMRF has zero mean and that the lattice is square. Furthermore the values at sites outside of the lattice are set to zero, simply meaning that sites close to the border of the lattice may have fewer neighbors.

The first option to define a GMRF through matrices with low displace-

ment rank is given by the following equation.

$$\mathbf{X} = M\mathbf{X} + \mathbf{X}N + \psi, \tag{46}$$

where $\mathbf{X}$ is a GMRF defined on a $n \times n$ lattice Ω with precision matrix $Q_\sigma \in \mathbb{R}^{n^2 \times n^2}$, $M \in \mathbb{R}^{n \times n}$ and $N \in \mathbb{R}^{n \times n}$ are matrices that have a low rank with respect to a known displacement operator, and $\psi \in \mathbb{R}^{n \times n}$ is zero-mean Gaussian noise with

$$\mathbf{E}(\overrightarrow{\psi}\,\overrightarrow{\psi}^T) = \sigma^2 Q_\sigma. \tag{47}$$

Transforming equation (46) yields

$$(I_{n^2} - I_n \otimes M - N^T \otimes I_n)\overrightarrow{\mathbf{X}} = \overrightarrow{\psi}. \tag{48}$$

Considering the *noncausal autoregressive field representation* (24), it follows that

$$Q_\sigma = (I_{n^2} - I_n \otimes M - N^T \otimes I_n), \tag{49}$$

which ensures that equation (46) is a valid representation for the GMRF $\mathbf{X}$.

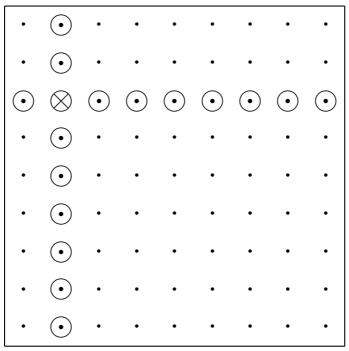
For dense matrices M and N, the neighbors of a site (k, l) are all the sites that are either in row k or column l. There is no diagonal interaction in this field (see Figure 24). Thus,

$$ne(\{k,l\}) = \{\{i,j\} | i = k \text{ and } N_{(j,l)} \neq 0 \;\; \text{or} \;\; j = l \text{ and } M_{(k,i)} \neq 0\}. \tag{50}$$

Since Q_σ is a precision matrix, it has to be symmetric and positive definite. The next Lemma shows which conditions the matrices M and N have to satisfy to ensure that Q_σ is SPD.

Lemma 4.2 *Let Q_σ be defined as in equation (49), and $\{\mathfrak{m}_1, \cdots, \mathfrak{m}_n\}$ and $\{\mathfrak{n}_1, \cdots, \mathfrak{n}_n\}$ be the eigenvalues of M and N, respectively. Q_σ is SPD if and*

Figure 24: Neighborhood scheme: Neighbours of site $\otimes$ are denoted by $\odot$



only if M and N are symmetric and the eigenvalues of Q_σ,

$$\{1 - \mathfrak{m}_i - \mathfrak{n}_j | i, j \in \{1, \cdots, n\}\}, \tag{51}$$

are strictly positive.

Proof. It is clear from equation (49) that Q_σ is symmetric if and only if M and N are symmetric. Proposition 2.3 shows that the eigenvalues of Q_σ are given by equation (51). The definition of a SPD matrix proves the lemma. □

These conditions are rather restrictive; only a few of the structured matrices introduced in sections 2.3 and 4.1.3 fulfill these requirements. For example, the Toeplitz matrix with all entries along the first off-diagonals being equal to $0,249$ and the rest set to zero

$$T_1 = \begin{pmatrix} 0 & 0.249 & 0 & \cdots & 0 \\ 0.249 & 0 & 0.249 & \ddots & \vdots \\ 0 & 0.249 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 & 0.249 \\ 0 & \cdots & 0 & 0.249 & 0 \end{pmatrix} \tag{52}$$

can be used to obtain a SPD precision matrix. Setting $M = T_1$ and $N = T_1$

in equation (49) yields a precision matrix Q_σ for a first order GMRF, which corresponds to the structure presented in section 3.4.6, except that the field is homogeneous. Therefore the GMRF can also be represented in the following way [39]:

$$\mathbf{X}_{(k,l)} = \sum_{\{i,j\}\in\eta} \phi_{\{i,j\}} \mathbf{X}_{(k+i,l+j)} + \psi_{(k,l)}, \tag{53}$$

where $\eta = \{\{1,0\},\{0,1\},\{-1,0\},\{0,-1\}\}$ and ψ is defined as in equation (47). In this particular case with $M = T_1$ and $N = T_1$, $\phi_{\{i,j\}}$ is equal to 0.249 for all $\{i,j\}$ in η.

As Q_σ is symmetric and positive definite, the Cholesky decomposition can be computed. The lower or upper triangular matrix then enables the generation of a GMRF with precision matrix Q_σ:

```
U = chol(Q_sigma);
vecPsi = randn(n*n,1);
vecX = U\vecPsi;
X = reshape(vecX,n,n);
```
(54)

in MATLAB® where `chol` denotes the Cholesky decomposition, `randn` generates pseudo-random values following the standard normal distribution, `U\vecPsi` is equivalent to $U^{-1}vecPsi$, and the `reshape` command transforms the vectorized version of $\mathbf{X}$ into a matrix of size $n \times n$.

An example of a GMRF with precision matrix Q_σ, as described in equation (49) with $M = T_1$ and $N = T_1$, is presented in Figure 25. Two drawings with the corresponding 2-D discrete Fourier transform are shown, whereas the computation of the drawing and of the 2D-DFT is given by equations (54) and equations (45), respectively. As described in section 4.1.2, the command `imshow(M,[])` is used to visualize the drawings of the GMRFs. One can see that, after the linear interpolation (44) is applied to every entry in the

matrix, the new mean and the new covariance matrix are given by

$$\mathbf{E}(\overrightarrow{\mathbf{X}^P}) = \overrightarrow{D} \tag{55}$$

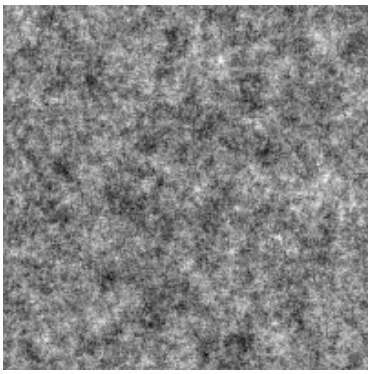
and

$$\mathbf{E}((\overrightarrow{\mathbf{X}^P} - \mathbf{E}(\overrightarrow{\mathbf{X}^P}))(\overrightarrow{\mathbf{X}^P} - \mathbf{E}(\overrightarrow{\mathbf{X}^P}))^T) = k^2\mathbf{E}(\overrightarrow{\mathbf{X}}\overrightarrow{\mathbf{X}}^T), \tag{56}$$

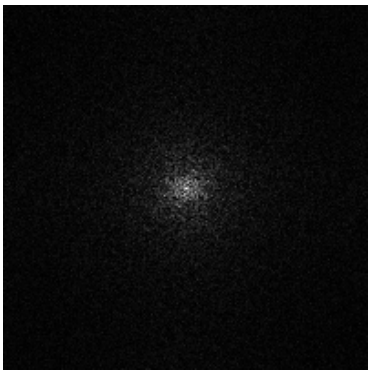
where $\mathbf{X}^P = k\mathbf{X} + D$, $k = \dfrac{1}{\max(\mathbf{X}(:)) - \min(\mathbf{X}(:))}$, $D \in \mathbb{R}^{n\times n}$, and $D_{(i,j)} = -k\min(\mathbf{X}(:))$ for all pairs (i, j).

Figure 25: Example of a GMRF (1)

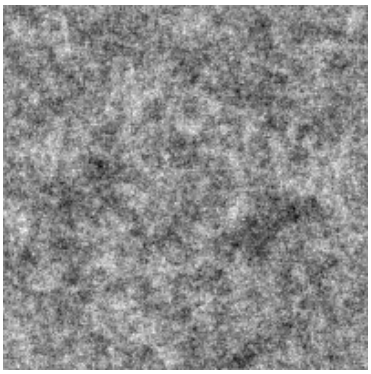
(a) First drawing



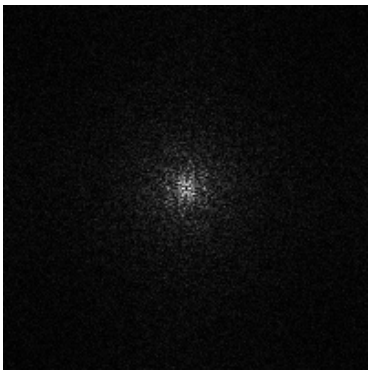
(b) 2-D DFT of first drawing



(c) Second drawing



(d) 2-D DFT of second drawing



4.2.2 Deviation of the Basic Setup

The restrictions on the structured matrices described in Lemma 4.2 give incentives to find a less limiting approach for the composition of the precision matrix. This is achieved by using the following formula for the precision matrix:

$$\mathcal{Q}_\sigma = Q_\sigma^T Q_\sigma = (I_{n^2} - I_n \otimes M - N^T \otimes I_n)^T (I_{n^2} - I_n \otimes M - N^T \otimes I_n). \quad (57)$$

Clearly $\mathcal{Q}_\sigma$ is symmetric for any matrices M and N. However, it is no longer sparse for most matrices M and N. Futhermore $\mathcal{Q}_\sigma$ is positive-definite if and only if $\det(Q_\sigma) \neq 0$; otherwise $\mathcal{Q}_\sigma$ is positive-semidefinite. The next Lemma indicates for which matrices M and N the precision matrix $\mathcal{Q}_\sigma$ is positive definite.

Lemma 4.3 *Let $\mathcal{Q}_\sigma$ be defined as in equation (57), and $\{\mathfrak{m}_1, \cdots, \mathfrak{m}_n\}$ and $\{\mathfrak{n}_1, \cdots, \mathfrak{n}_n\}$ be the eigenvalues of M and N, respectively. $\mathcal{Q}_\sigma$ is positive definite if and only if*

$$1 - \mathfrak{m}_i - \mathfrak{n}_j \neq 0, \qquad \forall i, j \in \{1, \cdots, n\}. \quad (58)$$

Proof. Follows from Lemma 4.2. □

Given the new precision matrix $\mathcal{Q}_\sigma$, the constraints on the matrices M and N are much looser than in the basic setup. Under these conditions, the structured matrices from sections 2.3 and 4.1.3 can also be used in the composition of the precision matrix, as they satisfy equation (58) but are not symmetric or don't comply with the restraints of Lemma 4.2.

Before the framework of a GMRF with precision matrix $\mathcal{Q}_\sigma$ is characterized, two propositions are needed.

Proposition 4.4 *Let $\mathcal{Q}_\sigma \in \mathbb{R}^{n^2 \times n^2}$ be symmetric and positive-definite, $\mathcal{Q}_\sigma = U^T U$ its Cholesky decomposition, and $B \in \mathbb{R}^{n^2 \times n^2}$ such that $\mathcal{Q}_\sigma = B^T B$.*

Then there exists a unique orthogonal matrix $O \in \mathbb{R}^{n^2 \times n^2}$*, which solves* $B = OU$*.*

Proof. As $\mathcal{Q}_\sigma$ is positive-definite, the upper triangular matrix U is unique and has strictly positive entries along the diagonal. Therefore U is regular, which proves the existence and the uniqueness of O. To be orthogonal, O has to fulfill $O^T O = OO^T = I$, as shown:

$$O^T O = U^{-1^T} B^T B U^{-1} = U^{T^{-1}} U^T U U^{-1} = I.$$

□

Note that in the following proposition the letter Q_σ, which is used in section 3.4.4, is substituted by the letter $\mathcal{Q}_\sigma$.

Proposition 4.5 *Let* $\mathcal{Q}_\sigma \in \mathbb{R}^{n^2 \times n^2}$ *be a SPD matrix,* $\mathcal{Q}_\sigma \overrightarrow{\mathbf{X}} = \overrightarrow{e}$ *a minimum mean square represenation of a GMRF as in equation* (24)*,* $\mathcal{Q}_\sigma = U^T U$ *the Cholesky decomposition of* $\mathcal{Q}_\sigma$*,* $O \in \mathbb{R}^{n^2 \times n^2}$ *an arbitrary orthogonal matrix, and* $B = OU$*. Then it is statistically equivalent to either use the decomposition* $\mathcal{Q}_\sigma = U^T U$ *or the decomposition* $\mathcal{Q}_\sigma = B^T B$ *to aquire a representation of the GMRF* $\mathbf{X}$*, which is driven by white noise.*

Proof. The statistically equivalent representation to equations (24), (29) and (34) is given by

$$B \overrightarrow{\mathbf{X}} = \overrightarrow{\epsilon}, \tag{59}$$

where

$$\overrightarrow{\epsilon} = B^{T^{-1}} \overrightarrow{e} \tag{60}$$

and therefore

$$\begin{aligned} \mathbf{E}(\overrightarrow{\epsilon}\,\overrightarrow{\epsilon}^T) &= \mathbf{E}(B^{T^{-1}} \overrightarrow{e}\,\overrightarrow{e}^T B^{-1}) \\ &= \sigma^2 O U^{T^{-1}} U^T O^T O U U^{-1} O^T \\ &= \sigma^2 I. \end{aligned} \tag{61}$$

Furthermore

$$\mathbf{E}(\overrightarrow{\mathbf{X}}\overrightarrow{\epsilon}^T) = \mathbf{E}(\overrightarrow{\mathbf{X}}\overrightarrow{e}^T B^{-1}) = \sigma^2 B^{-1}. \tag{62}$$

□

Remark. Although it is very convenient to be able to use any decomposition $\mathcal{Q}_\sigma = B^T B$, one should consider that, contrary to the representations from section 3.4.5, the GMRF defined by equation (59) is generally not one-sided.

As in the basic setup, an effective way to generate a GMRF $\mathbf{X}$ with precision matrix $\mathcal{Q}_\sigma$ makes use of the *noncausal autoregressive field representation* from section 3.4.4.

$$\mathcal{Q}_\sigma \overrightarrow{\mathbf{X}} = \overrightarrow{e}, \tag{63}$$

where

$$\begin{aligned} \mathbf{E}(\overrightarrow{\mathbf{X}}\overrightarrow{\mathbf{X}}^T) &= \left(\frac{1}{\sigma^2}\mathcal{Q}_\sigma\right)^{-1}, \\ \mathbf{E}(\overrightarrow{e}\overrightarrow{e}^T) &= \sigma^2 \mathcal{Q}_\sigma, \\ \mathbf{E}(\overrightarrow{\mathbf{X}}\overrightarrow{e}^T) &= \sigma^2 I, \end{aligned} \tag{64}$$

and $\mathcal{Q}_\sigma$ is given like in equation (57). It is known from Propositions 4.4 and 4.5 that the model given by equation (63) is equivalent to

$$Q_\sigma \overrightarrow{\mathbf{X}} = \overrightarrow{\zeta}, \tag{65}$$

where

$$\overrightarrow{\zeta} = Q_\sigma^{T^{-1}} \overrightarrow{e} \tag{66}$$

and therefore

$$\mathbf{E}(\overrightarrow{\zeta}\overrightarrow{\zeta}^T) = \sigma^2 I, \tag{67}$$

$$\mathbf{E}(\overrightarrow{\mathbf{X}}\overrightarrow{\zeta}^T) = \sigma^2 Q_\sigma^{-1}. \tag{68}$$

Equation (65) can be rewritten as

$$\begin{aligned} & (I_{n^2} - I_n \otimes M - N^T \otimes I_n)\overrightarrow{\mathbf{X}} = \overrightarrow{\zeta} \\ \iff & \mathbf{X} - M\mathbf{X} - \mathbf{X}N = \zeta \\ \iff & (I - M)\mathbf{X} - \mathbf{X}N = \zeta. \end{aligned} \tag{69}$$

Equation (69) has the same form as the continuous-time Lyapunov equation, which simplifies the generation of a GMRF in MATLAB®. The following code is necessary for given structured matrices M and N:

```
zeta = randn(n,n);
X = lyap(eye(n)-M,-N,-zeta);
```

For example, using the Toeplitz matrix T_1 (52), a precision matrix $\mathcal{Q}_\sigma$ of order 3 is aquired by setting $M = T_1$ and $N = T_1$ in equation (57). Two drawings from a GMRF with precision matrix $\mathcal{Q}_\sigma$ and their according 2D-DFTs are shown in Figure 26.

Figure 26: Example of a GMRF (2)

(a) First drawing (b) 2-D DFT of first drawing

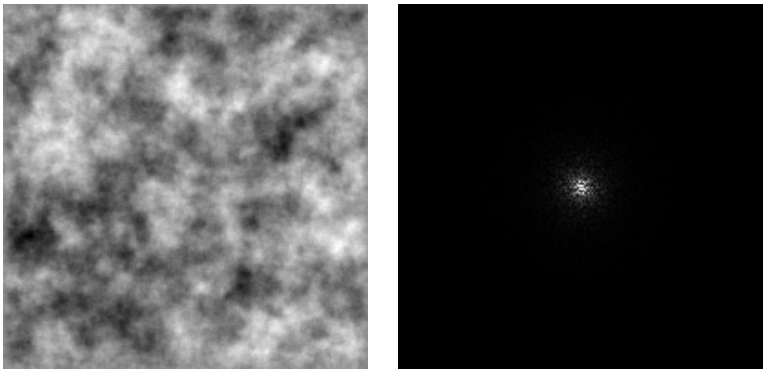
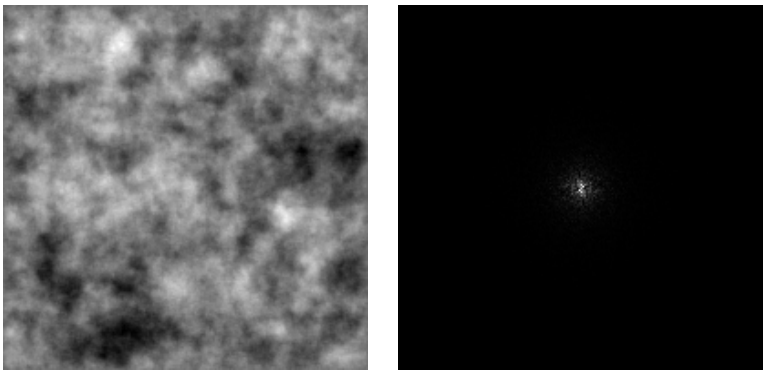


Figure 26: Example of a GMRF (2) (Continued)

(c) Second drawing (d) 2-D DFT of second drawing



Changing the values along the first off-diagonals of the matrix T_1 from 0.249 to 0.25 and keeping the rest of the setup from the prior example unaltered yields the drawings demonstrated in Figure 27.

Figure 27: Example of a GMRF (3)

(a) First drawing (b) 2-D DFT of first drawing

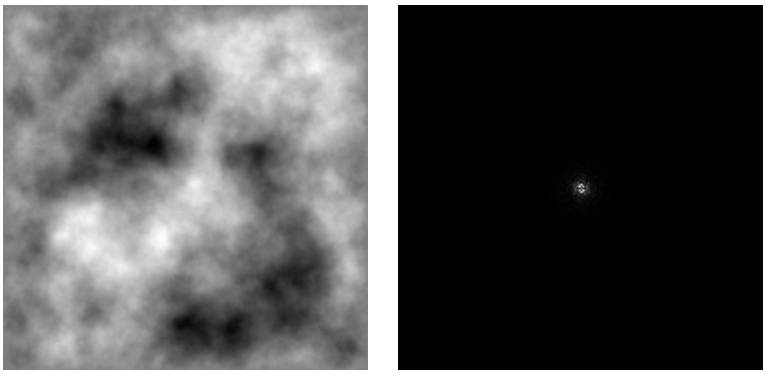
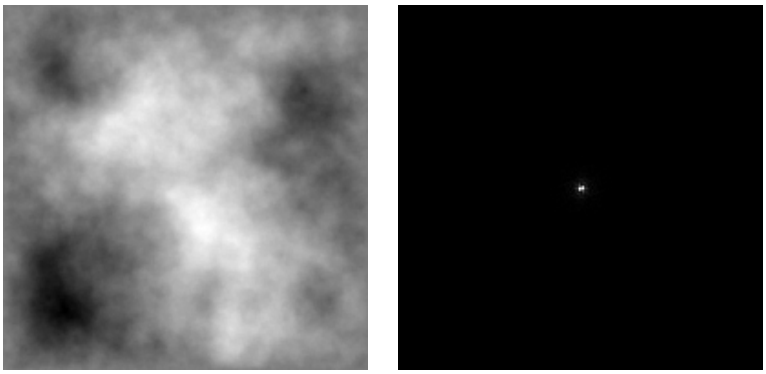


Figure 27: Example of a GMRF (3) (Continued)

(c) Second drawing (d) 2-D DFT of second drawing



4.2.3 Results

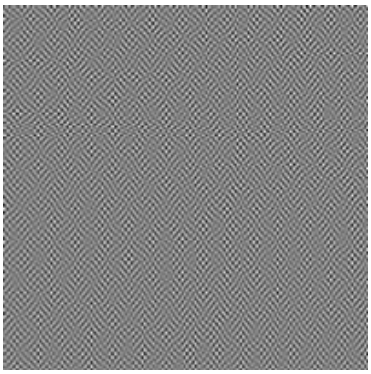
Now the textures/structured matrices generated in section 4.1.3 are used to compose the precision matrix of a GMRF $\mathbf{X}$. The caption of each figure indicates which texture/structured matrix is used for both M and N (see equation (57)). For example, "Figure 28: M and N are equal to Figure 6e" denotes that for the GMRF $\mathbf{X}$ and the matrix S, illustrated in Figure 6e, the following holds true:

$$\mathbf{E}(\overrightarrow{\mathbf{X}}\overrightarrow{\mathbf{X}}^T) = (I_{n^2} - I_n \otimes S - S^T \otimes I_n)^T (I_{n^2} - I_n \otimes S - S^T \otimes I_n).$$

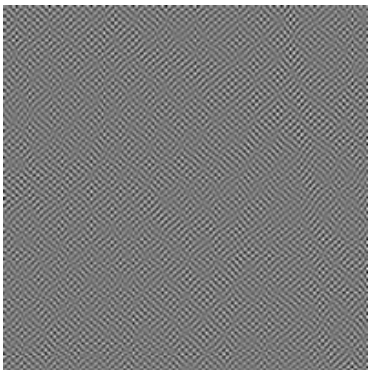
The pictures within one figure are organized in such a way that each column represents one drawing of the GMRF $\mathbf{X}$. The first row of each column contains the actual drawing, the second row its logtransform, and the third row its 2-D discrete Fourier transform.

Figure 28: M and N are equal to Figure 6e

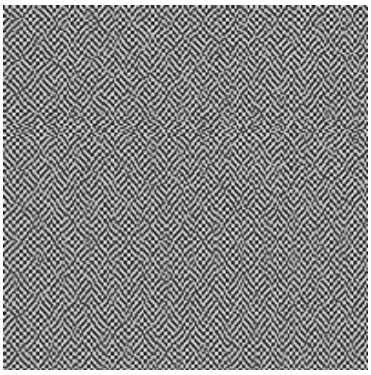
(a) First drawing



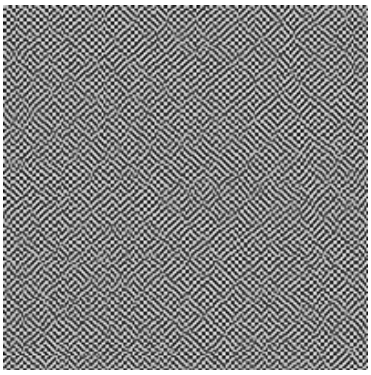
(b) Second drawing



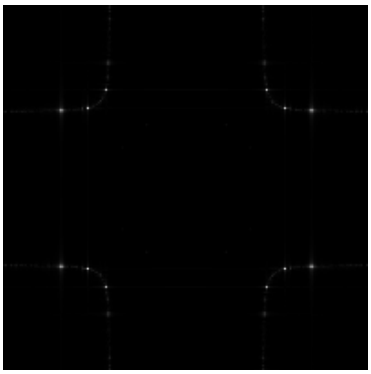
(c) Logtransf. of first drawing



(d) Logtransf. of second drawing



(e) 2-D DFT of first drawing



(f) 2-D DFT of second drawing

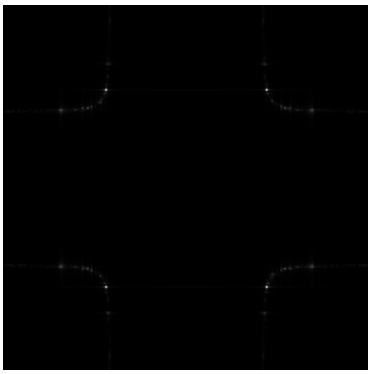
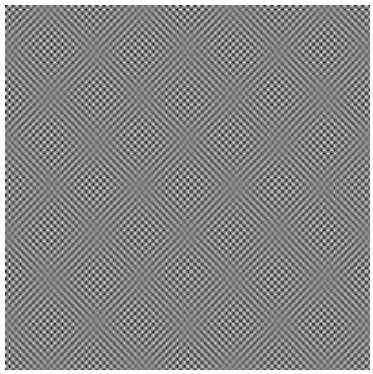
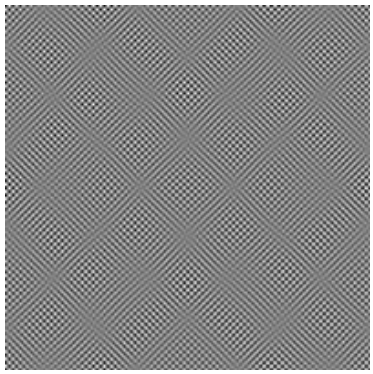


Figure 29: M and N are equal to Figure 6m

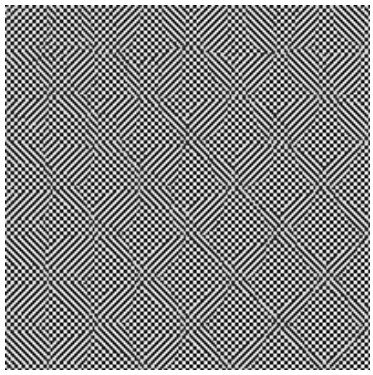
(a) First drawing



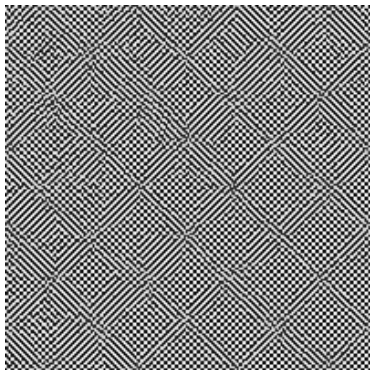
(b) Second drawing



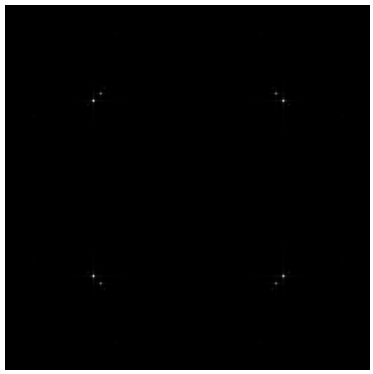
(c) Logtransf. of first drawing



(d) Logtransf. of second drawing



(e) 2-D DFT of first drawing



(f) 2-D DFT of second drawing

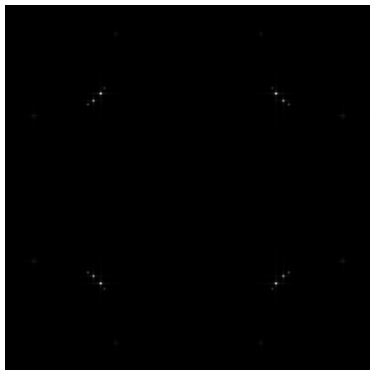
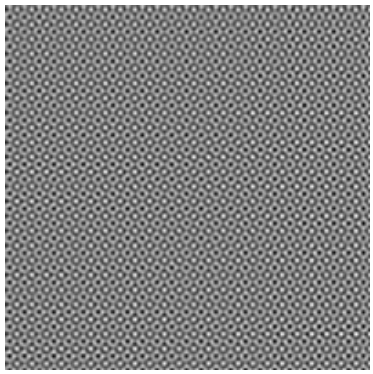
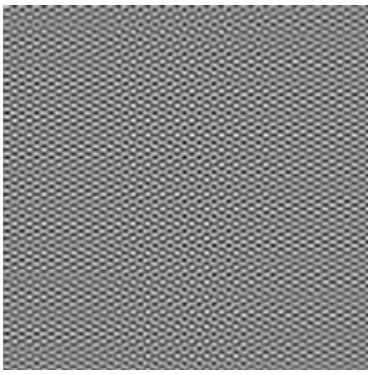


Figure 30: M and N are equal to Figure 6o

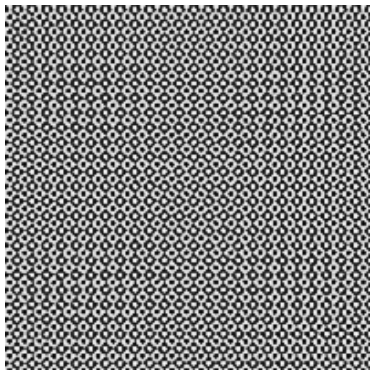
(a) First drawing



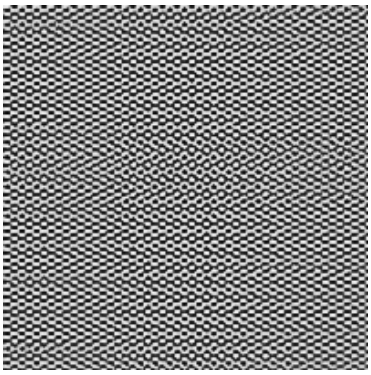
(b) Second drawing



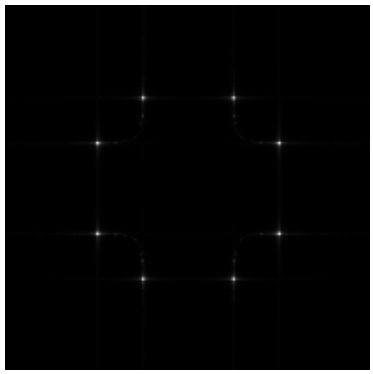
(c) Logtransf. of first drawing



(d) Logtransf. of second drawing



(e) 2-D DFT of first drawing



(f) 2-D DFT of second drawing

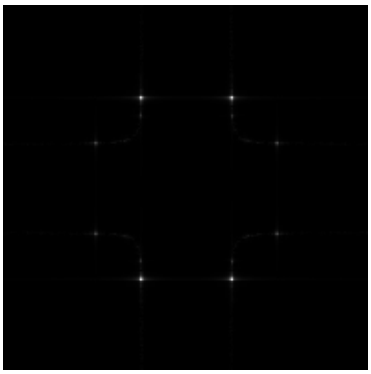
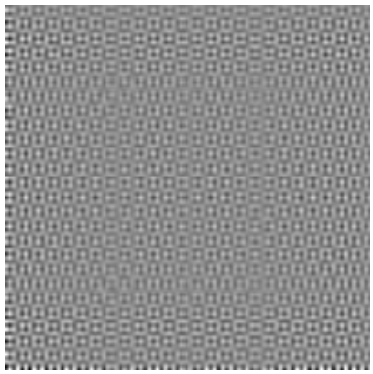
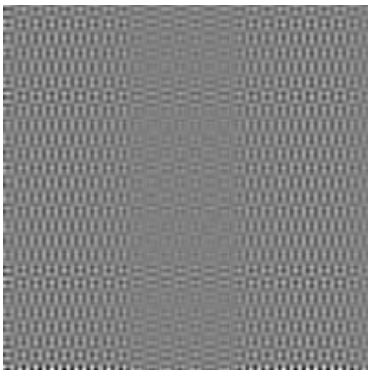


Figure 31: M and N are equal to Figure 7c

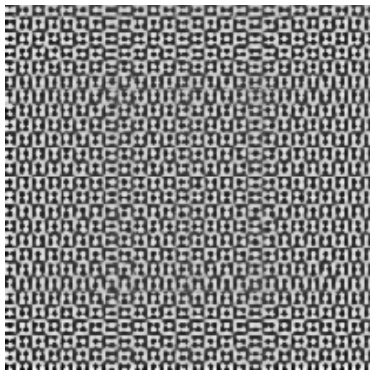
(a) First drawing



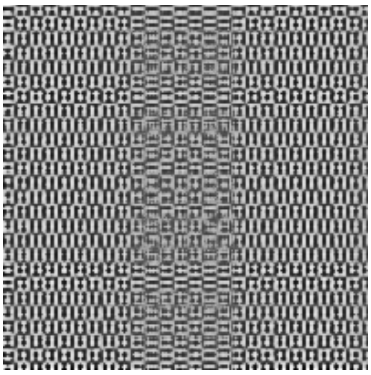
(b) Second drawing



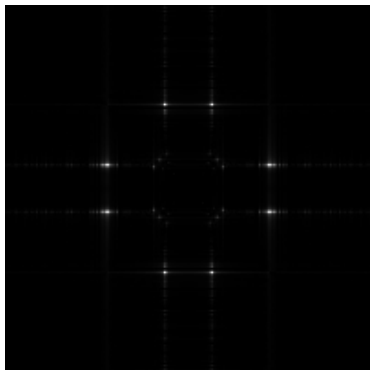
(c) Logtransf. of first drawing



(d) Logtransf. of second drawing



(e) 2-D DFT of first drawing



(f) 2-D DFT of second drawing

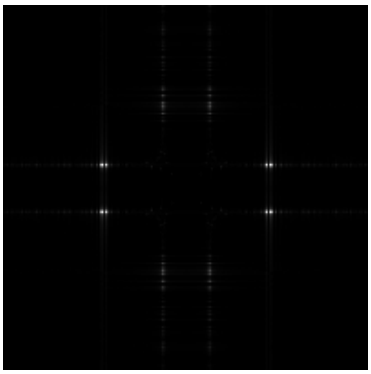
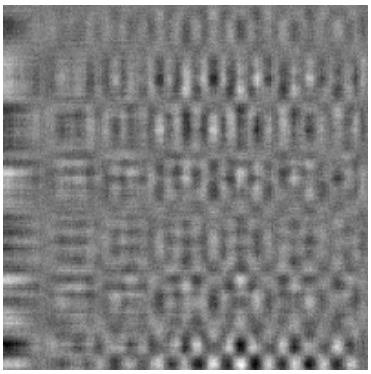
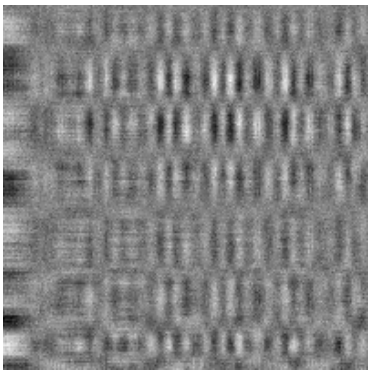


Figure 32: M and N are equal to Figure 7e

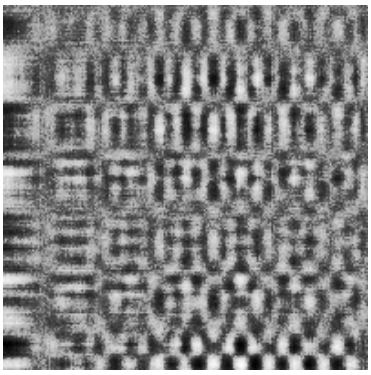
(a) First drawing



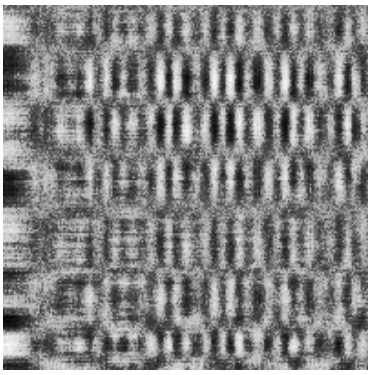
(b) Second drawing



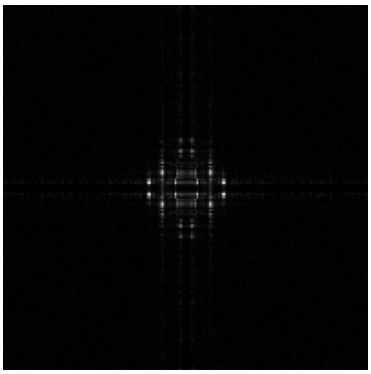
(c) Logtransf. of first drawing



(d) Logtransf. of second drawing



(e) 2-D DFT of first drawing



(f) 2-D DFT of second drawing

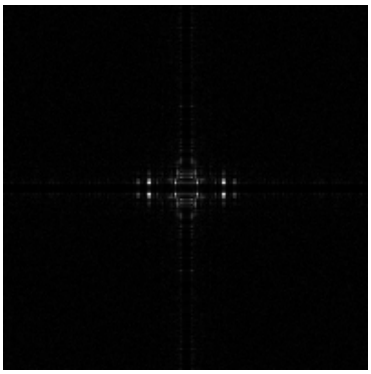
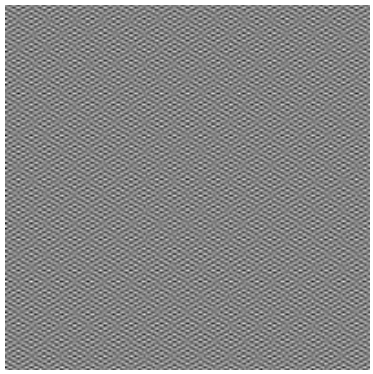
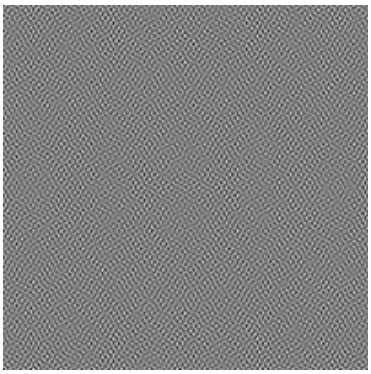


Figure 33: M and N are equal to Figure 7g

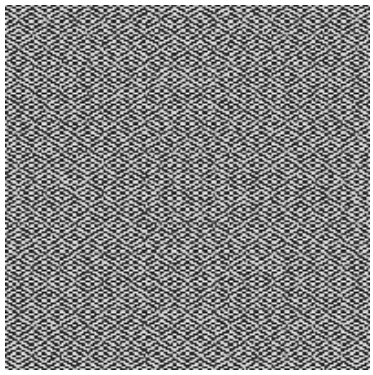
(a) First drawing



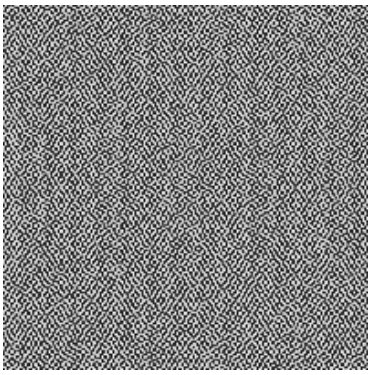
(b) Second drawing



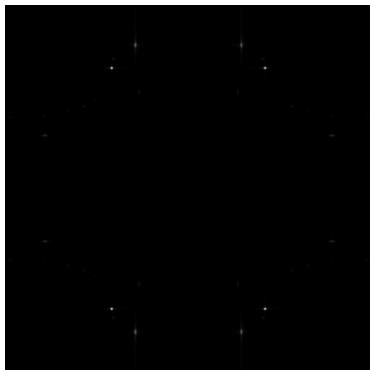
(c) Logtransf. of first drawing



(d) Logtransf. of second drawing



(e) 2-D DFT of first drawing



(f) 2-D DFT of second drawing

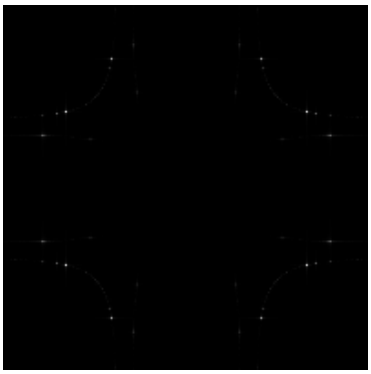
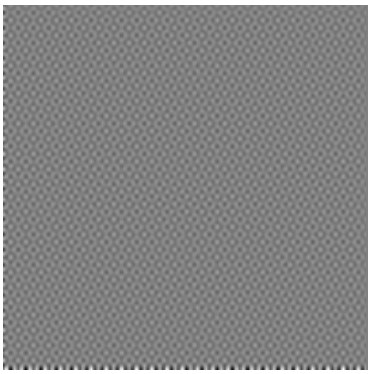
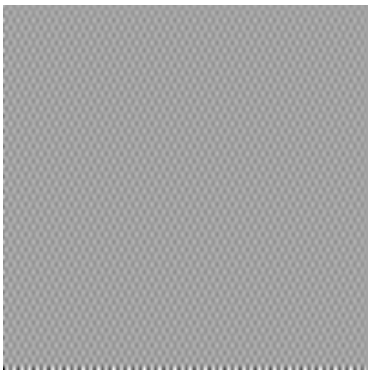


Figure 34: M and N are equal to Figure 7i

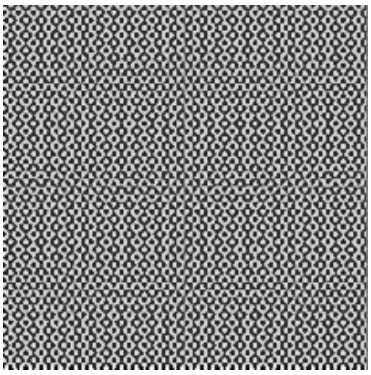
(a) First drawing



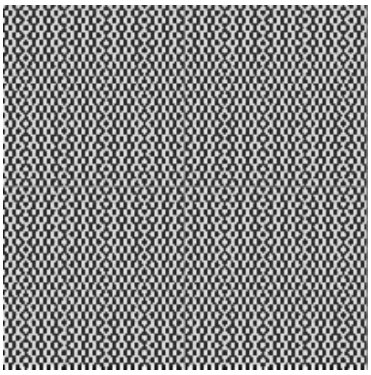
(b) Second drawing



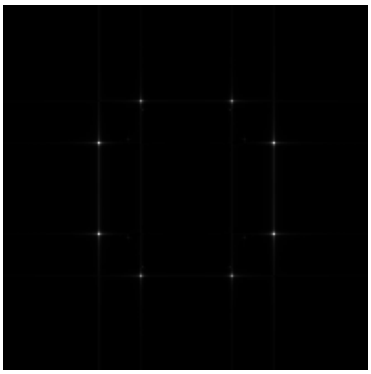
(c) Logtransf. of first drawing



(d) Logtransf. of second drawing



(e) 2-D DFT of first drawing



(f) 2-D DFT of second drawing

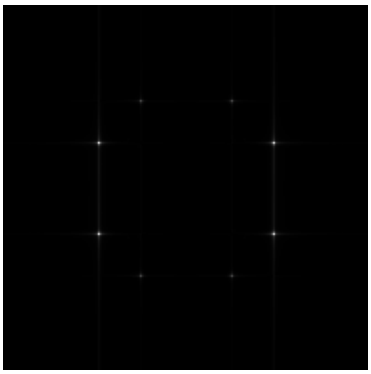
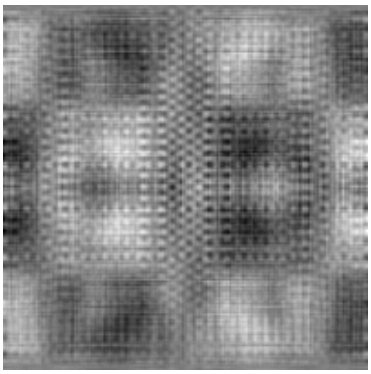
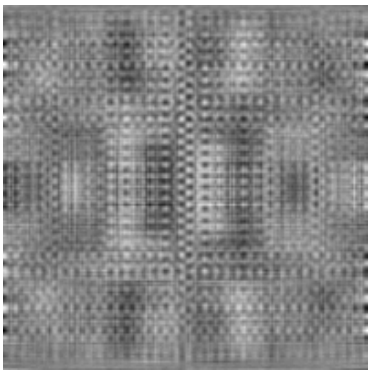


Figure 35: M and N are equal to Figure 8c

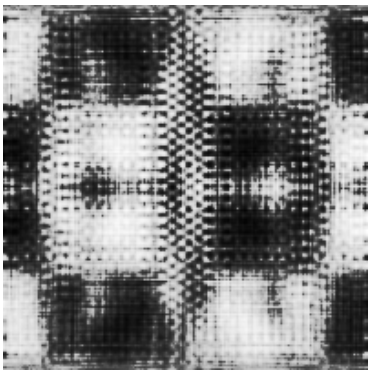
(a) First drawing



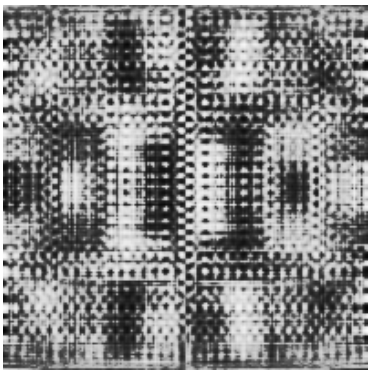
(b) Second drawing



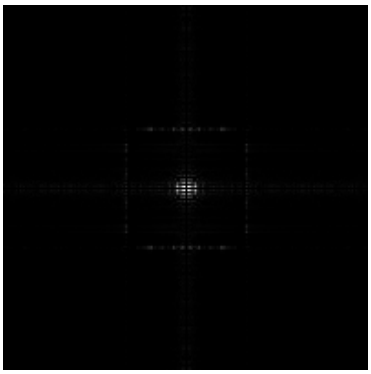
(c) Logtransf. of first drawing



(d) Logtransf. of second drawing



(e) 2-D DFT of first drawing



(f) 2-D DFT of second drawing

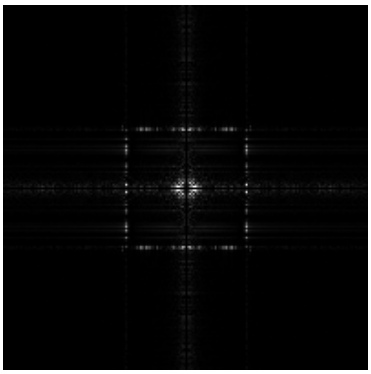
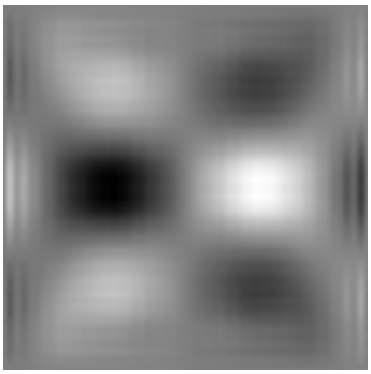
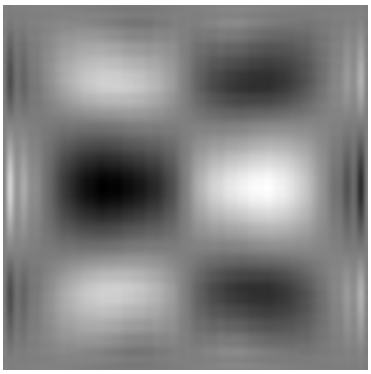


Figure 36: M and N are equal to Figure 8k

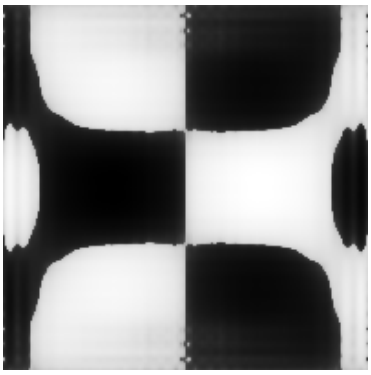
(a) First drawing



(b) Second drawing

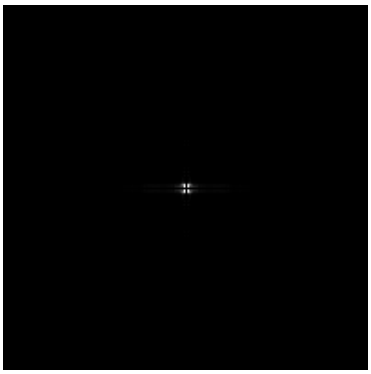


(c) Logtransf. of first drawing



(d) Logtransf. of second drawing

(e) 2-D DFT of first drawing



(f) 2-D DFT of second drawing

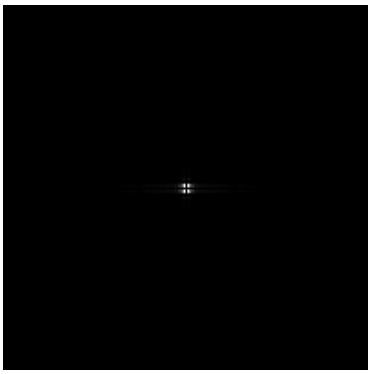
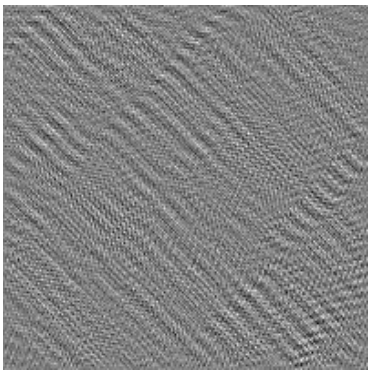
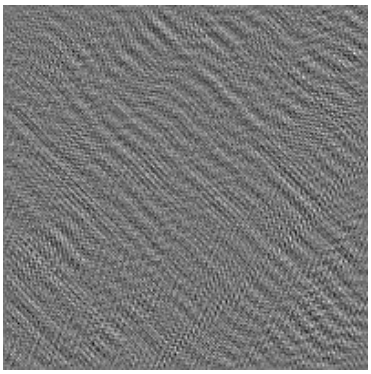


Figure 37: M and N are equal to Figure 13c

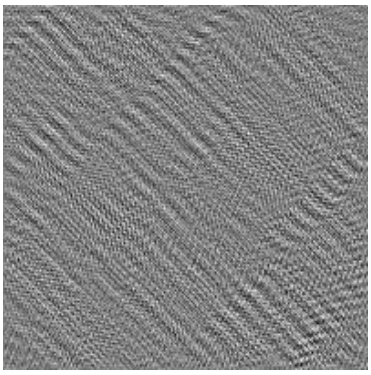
(a) First drawing



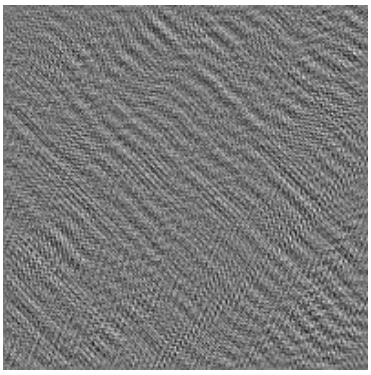
(b) Second drawing



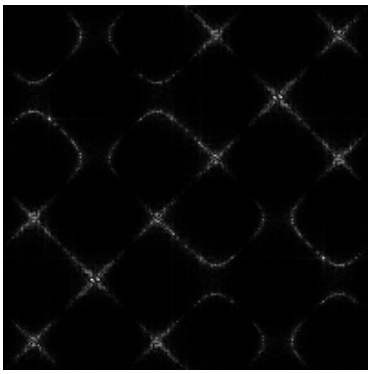
(c) Logtransf. of first drawing



(d) Logtransf. of second drawing



(e) 2-D DFT of first drawing



(f) 2-D DFT of second drawing

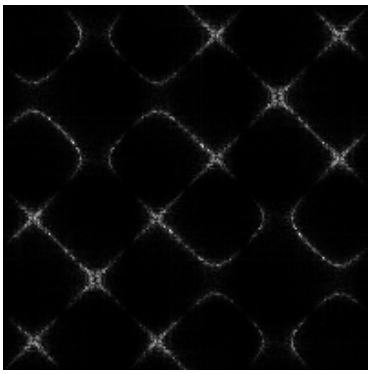
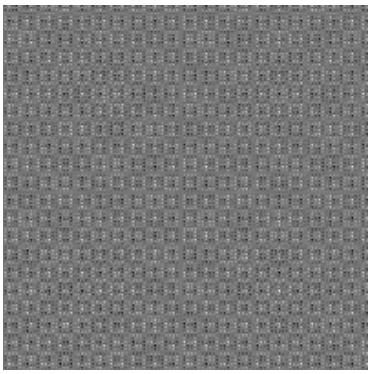
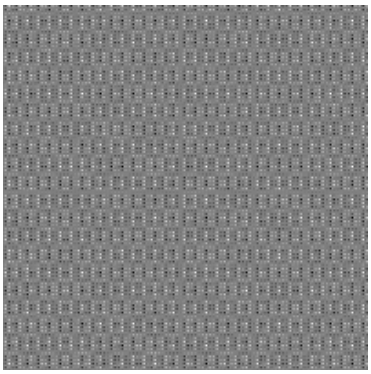


Figure 38: M and N are equal to Figure 13k

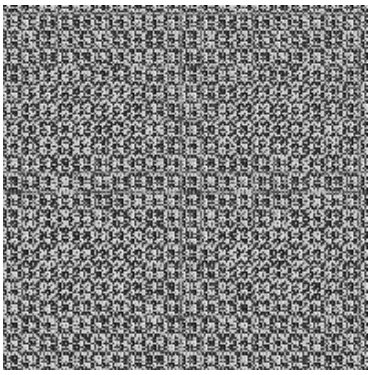
(a) First drawing



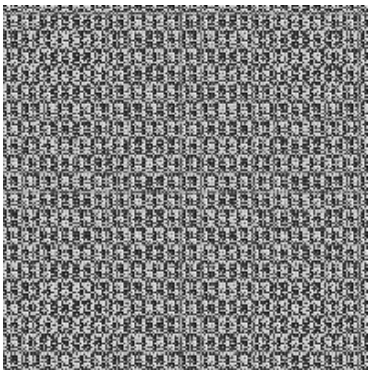
(b) Second drawing



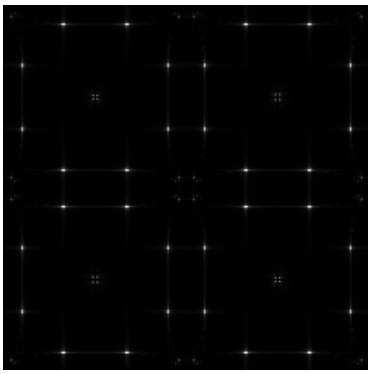
(c) Logtransf. of first drawing



(d) Logtransf. of second drawing



(e) 2-D DFT of first drawing



(f) 2-D DFT of second drawing

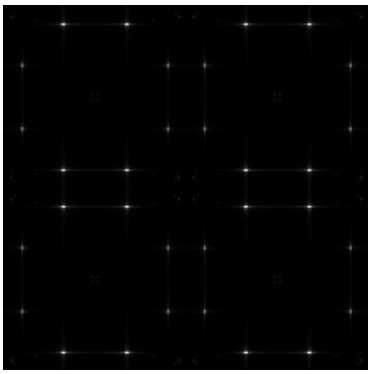
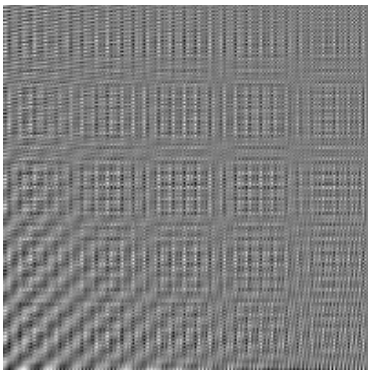
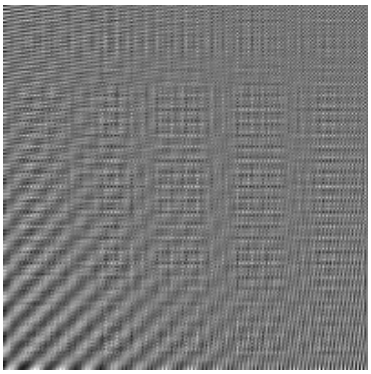


Figure 39: M and N are equal to Figure 13m

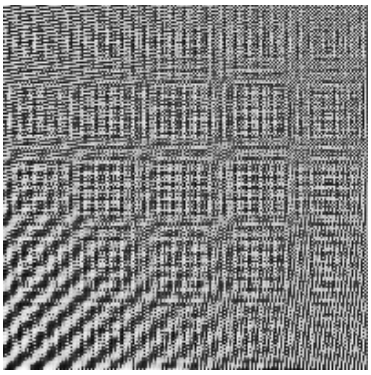
(a) First drawing



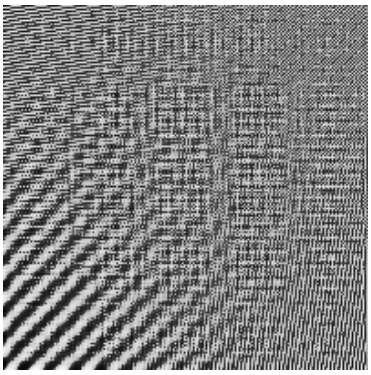
(b) Second drawing



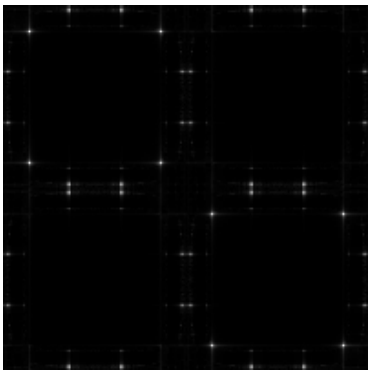
(c) Logtransf. of first drawing



(d) Logtransf. of second drawing



(e) 2-D DFT of first drawing



(f) 2-D DFT of second drawing

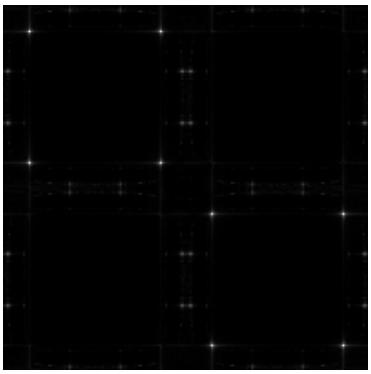
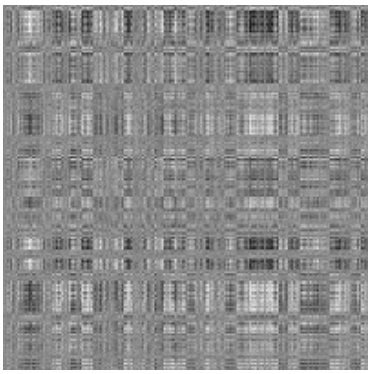
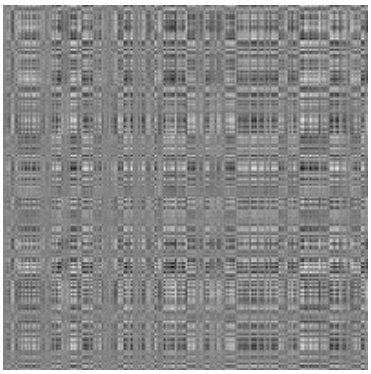


Figure 40: M and N are equal to Figure 15e

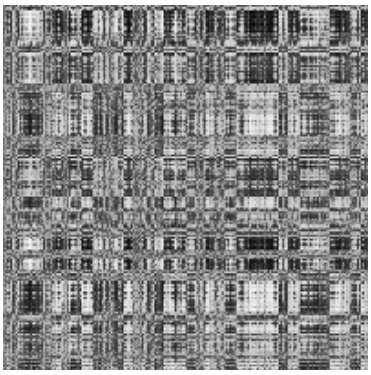
(a) First drawing



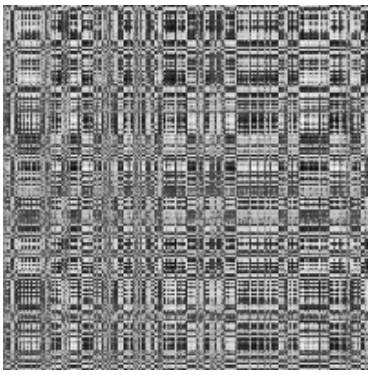
(b) Second drawing



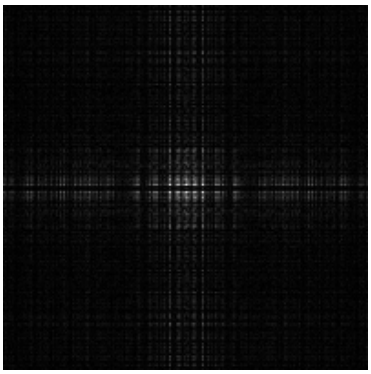
(c) Logtransf. of first drawing



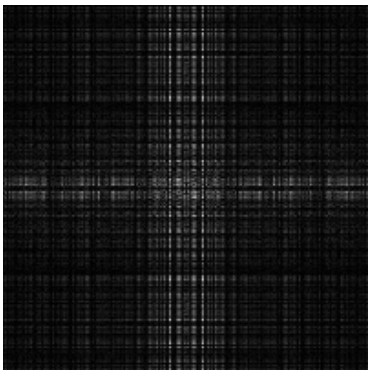
(d) Logtransf. of second drawing



(e) 2-D DFT of first drawing



(f) 2-D DFT of second drawing



4.2.4 Summary

This section has proposed an approach for describing textures using the theory of GMRFs and assuming that the precision matrix is a compound of two structured matrices. The main advantage of the GMRF appraoch is that the texture doesn't have to be exact, since the GMRF incorporates a noise term. Furthermore the basic setup involves only vertical and horizontal field interactions, which implies a very sparse structure of the precision matrix. However, because a precision matrix has to be SPD to represent a proper density, the variety of matrices which compose the precision matrix is restricted. Most of the structured matrices presented in section 4.1.3 do not comply with those restrictions, so a deviation of the basic setup is proposed.

It is assumed that the precision matrix used in the basic setup is multiplied with its transpose to create a new precision matrix. On one hand, this ensures its symmetry and, in case the former matrix has full rank, the positive definiteness of the newly created precision matrix. On the other hand, in most cases the sparseness gets lost and every site depends on all the other sites.

A diversity of textures can be generated using only $2n^2$ parameters for the precision matrix. The abundance and variety of the produced textures shows that characterizing textures with the help of displacement operators is a very promising approach. However, as in the previous section, the process of classifying unknown textures with this approach is not treated. In order to do so using the deviation of the basic setup, one would face the difficulty of extracting the 2 structured matrices from the computed precision matrix.

5 Transformation of Textures

Often textures are transformed by applying functions such as the Wavelet transform, the Fourier transform or a FIR filter, which in turn impacts the texture's displacement operator. It is assumed that for a given texture/structured matrix M, the displacement operator L is known, and the corresponding displacement $L(M)$ has low rank $r_{A,B}(M)$. This section explores how the displacement operator L, and its operator matrices A and B change when a function $f(\cdot)$ is applied to the texture/structured matrix M.

5.1 Linear Transformation

First the case where $f(\cdot)$ is a linear function is examined. The function $f(\cdot)$ can be represented by a matrix $F \in \mathbb{R}^{k\times l}$.

Throughout this section, let $M \in \mathbb{R}^{l\times m}$ be a texture/structured matrix with displacement rank $r_{A,B}(M)$ with respect to the Stein $\Delta_{A,B}$ or the Sylvester $\nabla_{A,B}$ displacement operator with operator matrices $A \in \mathbb{R}^{l\times l}$ and $B \in \mathbb{R}^{m\times m}$. Let r_F and r_M be the rank of F and M respectively, and $U_F S_F V_F^T = F$ the Singular Value Decomposition of F.

The first Lemma supposes that F is a square matrix and invertible:

Lemma 5.1 *Let $F \in \mathbb{R}^{l\times l}$ and $r_F = l$. Then $r_{R,B}(FM) = r_{A,B}(M)$, where $R = FAF^{-1}$*

Proof.
Stein:

$$FM - RFMB = FM - FAF^{-1}FMB = F\Delta_{A,B}(M)$$

Sylvester:

$$RFM - FMB = FAF^{-1}FM - FMB = F\nabla_{A,B}(M)$$

The rank of the displacement stays unaltered, as F has full rank. □

For the next Lemma, a generalization of the matrix inverse, which can be applied to singular and rectangular matrices, is needed [2].

Definition. Moore-Penrose Generalized Inverse. Let $M \in \mathbb{C}^{m\times n}$ be nonzero, $r = \text{rank}(M)$, and $USV^* = M$ the Singular Value Decomposition of M. The Moore-Penrose generalized inverse $M^\dagger \in \mathbb{C}^{n\times m}$ of M is given by

$$M^\dagger = V_{(:,1:r)} S^{-1}_{(1:r,1:r)} U^*_{(:,1:r)}. \tag{70}$$

M and $M^\dagger$ satisfy the following equations:

$$\begin{aligned} MM^\dagger M &= M, \\ M^\dagger MM^\dagger &= M^\dagger, \\ (MM^\dagger)^* &= MM^\dagger, \\ (M^\dagger M)^* &= M^\dagger M. \end{aligned}$$

For the remaining Lemmata, the assumption that F is square and invertible is no longer used.

Lemma 5.2 *$r_{R,B}(FM) \leq r_{A,B}(M) + (-|r_F - \frac{l}{2}| + \frac{l}{2})$, where R is computed as follows*

$$\begin{aligned} D &= V_F^T A V_F, \\ K &= S_F D S_F^\dagger, \\ R &= U_F K U_F^T. \end{aligned} \tag{71}$$

Proof.

Stein:

$$\begin{aligned} & FM - RFMB \\ = \; & FM - U_F K S_F V_F^T MB \\ = \; & FM - U_F (S_F D - (S_F D - K S_F)) V_F^T MB \\ = \; & FM - U_F S_F D V_F^T MB + U_F (S_F D - K S_F) V_F^T MB \\ = \; & F\Delta_{A,B}(M) + U_F (S_F D - K S_F) V_F^T MB \end{aligned}$$

Sylvester:

$$\begin{aligned} & RFM - FMB \\ = \; & U_F K S_F V_F^T M - FMB \\ = \; & U_F (S_F D - (S_F D - K S_F)) V_F^T M - FMB \\ = \; & U_F S_F D V_F^T M - U_F (S_F D - K S_F) V_F^T M - FMB \\ = \; & F\nabla_{A,B}(M) - U_F (S_F D - K S_F) V_F^T M \end{aligned}$$

where

$$S_F D = \begin{pmatrix} (S_F D)_{(1:r_F, 1:r_F)} & (S_F D)_{(1:r_F, l-r_F+1:l)} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$$

and

$$K S_F = \begin{pmatrix} (S_F D)_{(1:r_F, 1:r_F)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}.$$

It follows that

$$(S_F D - K S_F) = \begin{pmatrix} \mathbf{0} & (S_F D)_{(1:r_F, l-r_F+1:l)} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}.$$

As the submatrix $(S_F D)_{(1:r_F, l-r_F+1:l)}$ has r_F rows and $l - r_F$ columns, the matrix $U_F(S_F D - K S_F) V_F^T MB$ in the Stein case and the matrix $U_F(S_F D - K S_F) V_F^T M$ in the Sylvester case have a maximum rank of $\min(r_F, l - r_F)$.

This proves the upper limit of the displacement rank. □

So far only one operator matrix was affected by the transformation. By also changing the second operator matrix, the upper limit of the displacement rank can be further decreased. A permutation matrix P is employed to maximize the rank of $D_{P|(1:r_L,1:r_L)}$ in the equation $D_P = DP$, where D is from equation (71). If the submatrix $D_{P|(1:r_L,1:r_L)}$ already has full rank, P is equal to the identity matrix. In addition, the following variables are defined:

$$K_P = S_F D_P S_F^{\dagger}, \tag{72}$$

$$R = U_F K_P U_F^T. \tag{73}$$

For the next two Lemmata it is assumed that the texture/structured matrix M is invertible. The first Lemma explores the case with the Stein displacement operator

Lemma 5.3 *Let $M \in \mathbb{R}^{l\times l}$, $r_M = l$, and the displacement $\Delta_{A,B}(M)$ have low rank $r_{A,B}(M)$. Then $r_{R,Y}(FM) \leq r_{A,B}(M)$, where R is defined by equation* (73) *and Y is given by*

$$\begin{aligned} D_Z &= (K_P S_F)^{\dagger}(S_F D - K_P S_F), \\ D_L &= M^{-1} V_F D_Z V_F^T M, \\ Y &= B + D_L B, \end{aligned} \tag{74}$$

where D and K_P are given by (71) *and* (72)*, respectively.*

Proof. Let $[M]$ denote the column space of a matrix M, and $\subseteq$ and $\subsetneq$ symbolize "a subset of" and "a proper subset of", respectively.

First it is necessary to show that $K_P S_F D_Z = S_F D - K_P S_F$: Assuming that $[K_P S_F] \subsetneq [S_F D]$, a vector $\mathbf{v}$ exists with $\mathbf{v} \in [S_F D]$ and $\mathbf{v} \notin [K_P S_F]$. As $(K_P S_F)_{(:,1:r_F)} = (S_F DP)_{(:,1:r_F)}$, it follows that $\mathbf{v} \in \left[(S_F DP)_{(:,r_F+1:l)}\right]$ and $\mathbf{v} \notin \left[(S_F DP)_{(:,1:r_F)}\right]$. However, that creates a contradiction to the condition

that P maximizes the rank of $D_{P|(1:r_L,1:r_L)}$. Therefore $[K_P S_F] \supseteq [S_F D]$. It is easy to see that $[K_P S_F] \subseteq [S_F D]$. As a result, $[K_P S_F] = [S_F D]$ and $K_P S_F D_Z = S_F D - K_P S_F$, with $D_{Z|(i,j)} = 0$ for $i > r_F$.

$$
\begin{aligned}
& FM - RFMY \\
&= FM - RFMB - RFMD_L B \\
&= FM - U_F K_P S_F V_F^T MB - U_F K_P S_F D_Z V_F^T MB \\
&= FM - U_F (S_F D - (S_F D - K_P S_F)) V_F^T MB - U_F (S_F D - K_P S_F) V_F^T MB \\
&= FM - U_F S_F D V_F^T MB - U_F (K_P S_F - S_F D + S_F D - K_P S_F) V_F^T MB \\
&= F\Delta_{A,B}(M)
\end{aligned}
$$

Because F may have a rank less than l, $r_{R,Y}(FM) \leq r_{A,B}(M)$. □

The following Lemma is equivalent to the previous one for the Sylvester displacement operator.

Lemma 5.4 *Let $M \in \mathbb{R}^{l \times l}$, $r_M = l$, and the displacement $\nabla_{A,B}(M)$ have low rank $r_{A,B}(M)$. Then $r_{R,Y}(FM) \leq r_{A,B}(M)$, where R is defined by equation* (73) *and Y is given by*

$$
\begin{aligned}
D_Z &= S_F^{\dagger}(S_F D - K_P S_F), \\
D_L &= M^{-1} V_F D_Z V_F^T M, \\
Y &= B - D_L,
\end{aligned}
\tag{75}
$$

where D and K_P are given by (71) *and* (72)*, respectively.*

Proof. Because $[S_F] \supseteq [K_P]$, it follows that $S_F D_Z = S_F D - K_P S_F$ with

$D_{Z|(i,j)} = 0$ for $i > r_F$.

$$\begin{aligned}
& RFM - FMY \\
&= U_F K_P S_F V_F^T M - FMB + FMD_L \\
&= U_F(S_F D - (S_F D - K_P S_F))V_F^T M - FMB + U_F S_F D_Z V_F^T M \\
&= U_F S_F D V_F^T M - U_F(S_F D - K_P S_F - S_F D + K_P S_F)V_F^T M - FMB \\
&= F\nabla_{A,B}(M)
\end{aligned}$$

Because F may have a rank less than l, $r_{R,Y}(FM) \leq r_{A,B}(M)$. □

The two following Lemmata are extensions of the two previous ones, when the texture/structured matrix M is not invertible.

Let $U_M S_M V_M^T$ be the singular value decomposition of M, $m_M^F = \min(r_F, r_M)$ the minimum of r_F and r_M, $\Lambda = V_F^T U_M$, $\lambda = \Lambda_{(1:m_M^F, 1:m_M^F)}$ a leading principal submatrix of Λ, $r_\lambda = \text{rank}(\lambda)$ the rank of the submatrix λ, and $r_{loss} = m_M^F - r_\lambda$ the nullity of the submatrix λ. J_{col} and J_{row} denote two index sets, such that $\lambda_{(:,j)}$ $j \in J_{col}$ is a basis of the column space of λ and $\lambda_{(j,:)}$ $j \in J_{row}$ is a basis of the row space of λ. Clearly J_{col} and J_{row} have both r_λ elements.

Moreover the subsequent definitions are needed:

$$\hat{D}_{Z|(J_{row},:)} = D_{Z|(J_{row},:)}, \qquad \hat{D}_Z \in \mathbb{R}^{l \times l}, \tag{76}$$

with D_Z from either equation (74), which will be applied in Lemma 5.5, or equation (75) in Lemma 5.6, and the other rows of $\hat{D}_Z$ set to zero.

$$\begin{aligned}
D_t &= \hat{D}_Z \Lambda, \\
\tilde{D}_{(J_{col},:)} &= \lambda^{-1}_{(J_{row},J_{col})} D_{t|(J_{row},:)}, \qquad \tilde{D} \in \mathbb{R}^{l \times l},
\end{aligned}$$

with the other rows of $\tilde{D}$ set to zero.

$$\begin{aligned} \check{D} &= S_M^{\dagger} \tilde{D} S_M, \\ D_L &= V_M \check{D} V_M^T. \end{aligned} \tag{77}$$

Let $D_{e_1} = (V_F^T U_M \tilde{D} - \hat{D}_Z V_F^T U_M) U_M^T V_F$ denote the first error matrix and $D_{e_2} = D_Z - \hat{D}_Z$ the second. From the definitions of $\tilde{D}$ and $\hat{D}_Z$, it can be deduced that $D_{e_1|(i,j)} = 0$ for $i \in J_{row}$. In addition, equation (76) verifies that $D_{e_2|(i,j)} = 0$ for $i \in J_{row}$ or $i > r_F$.

As before, the Stein displacement operator is examined first.

Lemma 5.5 *Let the displacement $\Delta_{A,B}(M)$ have low rank $r_{A,B}(M)$. Then*

$$r_{R,Y}(FM) \leq \begin{cases} r_{A,B}(M) + r_{Loss} & \text{if } r_F \leq r_M \\ r_{A,B}(M) + r_{Loss} + r_F - r_M & \text{if } r_F > r_M, \end{cases}$$

where R is defined like in equation (73) *and Y is given by*

$$Y = B + D_L B,$$

where D_L is defined by equation (77)*, while D_Z from equation* (74) *is used in equation* (76)*.*

Proof.

$$\begin{aligned} & FM - RFMY \\ =& FM - U_F K_P S_F V_F^T MB - U_F K_P S_F V_F^T M D_L B \\ =& FM - U_F(S_F D - (S_F D - K_P S_F)) V_F^T MB - U_F K_P S_F V_F^T U_M \tilde{D} S_M V_M^T B \\ =& FM - U_F(S_F D - (S_F D - K_P S_F)) V_F^T MB - U_F K_P S_F (\hat{D}_Z + D_{e_1}) V_F^T MB \\ =& FM - FAMB + U_F(S_F D - K_P S_F - K_P S_F \hat{D}_Z - K_P S_F D_{e_1}) V_F^T MB \\ =& F\Delta_{A,B}(M) + U_F(S_F D - K_P S_F - K_P S_F D_Z - K_P S_F D_{e_1} + K_P S_F D_{e_2}) V_F^T MB \\ =& F\Delta_{A,B}(M) - U_F K_P S_F (D_{e_1} - D_{e_2}) V_F^T MB \end{aligned}$$

If $E = S_F(D_{e_1} - D_{e_2})$, then $E_{(i,j)} = 0$ for $i \in J_{row}$ or $i > r_F$. This yields

$$\text{rank}(E) \leq \begin{cases} r_{Loss} & \text{if } r_F \leq r_M \\ r_{Loss} + r_F - r_M & \text{if } r_F > r_M. \end{cases}$$

The same is valid for the term $U_F K_P S_F (D_{e_1} - D_{e_2}) V_F^T MB$, which proves the Lemma. □

The same upper border can be found for the Sylvester displacement operator:

Lemma 5.6 *Let the displacement $\nabla_{A,B}(M)$ have low rank $r_{A,B}(M)$. Then*

$$r_{R,Y}(FM) \leq \begin{cases} r_{A,B}(M) + r_{Loss} & \textit{if } r_F \leq r_M \\ r_{A,B}(M) + r_{Loss} + r_F - r_M & \textit{if } r_F > r_M, \end{cases}$$

where R is defined by equation (73) *and Y is given by*

$$Y = B - D_L,$$

where D_L is defined by equation (77)*, while D_Z from equation* (75) *is used in equation* (76)*.*

Proof.

$$\begin{aligned} & RFM - FMY \\ =& U_F K_P S_F V_F^T M - FMB + FMD_L \\ =& U_F(S_F D - (S_F D - K_P S_F))V_F^T M - FMB + U_F S_F V_F^T U_M \tilde{D} S_M V_M^T \\ =& U_F(S_F D - (S_F D - K_P S_F) + S_F \hat{D}_Z + S_F D_{e_1})V_F^T M - FMB \\ =& FAM - U_F(S_F D - K_P S_F - S_F D_Z - S_F D_{e_1} + S_F D_{e_2})V_F^T M - FMB \\ =& F\nabla_{A,B}(M) + U_F S_F (D_{e_1} - D_{e_2}) V_F^T M \end{aligned}$$

If $E = S_F(D_{e_1} - D_{e_2})$, then $E_{(i,j)} = 0$ for $i \in J_{row}$ or $i > r_F$. This yields

$$\operatorname{rank}(E) \leq \begin{cases} r_{Loss} & \text{if } r_F \leq r_M \\ r_{Loss} + r_F - r_M & \text{if } r_F > r_M. \end{cases}$$

The same is valid for the term $U_F S_F(D_{e_1} - D_{e_2}) V_F^T M$, which proves the Lemma. □

Lemmata 5.1-5.6 demonstrate ways to alter the operator matrices in case the texture/structured matrix M is linearly transformed. In the first Lemma the transformation matrix F is invertible and therefore the solution is straightforward. In Lemma 5.2, F doesn't have to be regular, but still only one operator matrix is changed and the solution is independent of M. The border of the new displacement rank depends on the rank of F and the number of rows of M. If r_F is close to $\frac{l}{2}$, then the displacement rank may rise substantially. On the other hand, if r_F is close to zero or l, then the displacement rank may change just slightly. Lemmata 5.3 and 5.4 affect the second operator matrix as well and the computation depends on M, which is assumed to be invertible. By also transforming the second operator matrix, a lower displacement rank is achieved. In addition to the two previous ones, Lemmata 5.5 and 5.6 handle the case where M is not invertible. Thereby the border of the new displacement rank is subject to r_F, r_M, and the rank of a leading principal submatrix of the product $V_F^T U_M$.

Examples of interesting linear transforms especially for textures are (cyclic) shift, Wavelet transform, Fourier transform or FIR filter. A model for a cosine transform will be introduced here.

Example. Cosine transformed texture. The discrete cosine transform (DCT) of a square matrix $M \in \mathbb{R}^{n \times n}$ is given by

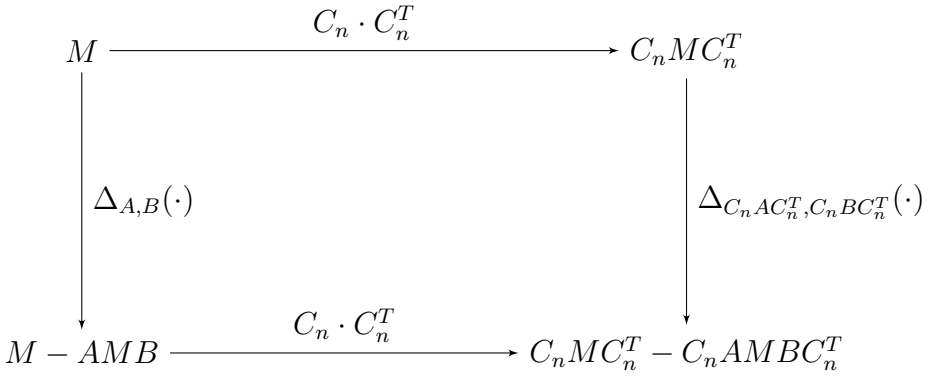
$$C_n M C_n^T,$$

where C_n denotes the orthogonal $n \times n$ DCT matrix:

$$C_{n|(i,j)} = \mu_i \cos \frac{\pi(2(j-1)+1)(i-1)}{2n}, \quad \text{for } i,j \in \{1,\cdots,n\},$$

with $\mu_1 = \frac{1}{\sqrt{n}}$, and $\mu_i = \sqrt{\frac{2}{n}}$ for $i \in \{2,\cdots,n\}$. Figure 41 demonstrates a commutative diagram. It is equivalent to either apply the DCT first and then use the adapted displacement operator, or first utilize the original displacement operator and then employ the DCT. Of course, the arrows could be reversed by appyling the inverse DCT and the inverse of the displacement operator, provided that the conditions from Theorem 2.1 are fulfilled. For the transformation of the displacement operator matrices, Lemma 5.1 is used.

Figure 41: Commutative diagram of the DCT and a displacement operator



Remark. Depending on which transformation matrix F is applied, the altered operator matrices will be in most cases not sparse anymore. Exceptions are for example cyclic shifts Z_1 or shifts like Z_0 (see equation (2)). Applying Lemma 5.1 or Lemma 5.2 will yield two sparse operator matrices once again.

5.2 Upsampling

Upsampling is a commonly utilized operation in the field of image analysis. It signifies the increase of the spatial resolution. An image upsampled by an integer L is given by

$$M_{U|(i,j)} = \begin{cases} M_{(\frac{i-1}{L}+1,\frac{j-1}{L}+1)} & \text{if } \mathrm{mod}(i-1,L)=0 \text{ and } \mathrm{mod}(j-1,L)=0 \\ 0 & \text{otherwise,} \end{cases}$$

where $i \in \{1,\cdots,Lm\}$, $j \in \{1,\cdots,Ln\}$, $M \in \mathbb{R}^{m\times n}$, and mod denotes the modulo operator. Upsampling can also be defined with the help of the Kronecker product:

$$M_U = M \otimes \mathcal{U}_{\mathcal{L}},$$

where

$$\mathcal{U}_{\mathcal{L}} = \underbrace{\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 \end{pmatrix}}_{L\times L}.$$

The following Lemma illustrates how the operator matrices change when the texture/structured matrix M is upsampled.

Lemma 5.7 *Let $M \in \mathbb{R}^{l\times m}$, $A \in \mathbb{R}^{l\times l}$ and $B \in \mathbb{R}^{m\times m}$ such that the displacement $\Delta_{A,B}(M)$ or $\nabla_{A,B}(M)$ has low rank $r_{A,B}(M)$. If $M_U = M \otimes \mathcal{U}_{\mathcal{L}}$ is the upsampled version of M, then $r_{R,Y}(M_U) = r_{A,B}(M)$, where $R = A \otimes I_L$ and $Y = B \otimes I_L$.*

Proof.

Stein:

$$\begin{aligned} M_U - RM_UY &= M \otimes \mathcal{U}_\mathcal{L} - (A \otimes I_L)(M \otimes \mathcal{U}_\mathcal{L})(B \otimes I_L) \\ &= M \otimes \mathcal{U}_\mathcal{L} - (AMB) \otimes \mathcal{U}_\mathcal{L} \\ &= \Delta_{A,B}(M) \otimes \mathcal{U}_\mathcal{L} \end{aligned}$$

Sylvester:

$$\begin{aligned} RM_U - M_UY &= (A \otimes I_L)(M \otimes \mathcal{U}_\mathcal{L}) - (M \otimes \mathcal{U}_\mathcal{L})(B \otimes I_L) \\ &= (AM) \otimes \mathcal{U}_\mathcal{L} - (MB) \otimes \mathcal{U}_\mathcal{L} \\ &= \nabla_{A,B}(M) \otimes \mathcal{U}_\mathcal{L} \end{aligned}$$

The new displacements $\Delta_{A,B}(M) \otimes \mathcal{U}_\mathcal{L}$ and $\nabla_{A,B}(M) \otimes \mathcal{U}_\mathcal{L}$ have the same rank as $\Delta_{A,B}(M)$ and $\nabla_{A,B}(M)$ respectively. □

Remark. In the previous Lemma, any $L \times L$ matrix D could have been used for $R = A \otimes D$ and $Y = B \otimes D$, as long as $D\mathcal{U}_\mathcal{L} = \mathcal{U}_\mathcal{L}$ and $\mathcal{U}_\mathcal{L}D = \mathcal{U}_\mathcal{L}$ hold true. For example, $D = \mathcal{U}_\mathcal{L}$ would be possible.

6 Conclusion

This thesis has considered the classification of textures, presenting two new techniques that are based on the displacement rank approach. The first assumes that two textures are equivalent if and only if they have low displacement rank with respect to the same displacement operator. The second assumes that the texture is generated by a GMRF and that the respective precision marix is a composition of structured matrices. While the former has a simpler framework, the advantage of the latter is that the structure doesn't have to be exact.

In both cases, numerous textures were generated by reversing the classification process. This variety of pictures shows the potency of these approaches and encourages further research to be conducted.

An effective and computationally inexpensive optimization algorithm that works as a classifier would need to be developed. In addition, more insights concerning the change of the displacement operator matrices once the texture is transformed are needed. For instance, certain existing transformations need to be altered such that the sparseness of the new operator matrices is ensured, and new maps have to be added.

With these considerations in mind, texture classification with displacement operators may proof to be an emerging technique in image analysis.

References

[1] T. Acharya and A. K. Ray. *Image processing: principles and applications*. John Wiley and Sons, 2005.

[2] D. S. Bernstein. *Matrix Mathematics*. Princeton University Press, 2005.

[3] J. E. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B*, 36(2):pp. 192–236, 1974.

[4] J. E. Besag and P. A. P. Moran. On the estimation and testing of spatial interaction in gaussian lattice processes. *Biometrika*, 62(3):pp. 555–562, 1975.

[5] P. Brodatz. *Texture: A Photographic Album for Artists and Designers*. New York: Dover, 1966.

[6] T. Chang and C.-C. Kuo. Texture analysis and classification with tree-structured wavelettransform. *IEEE Transactions on Image Processing*, 2(4):pp. 429–441, 1993.

[7] R. Chellappa. Image understanding and statistics: What each can do for the other? In *Proceedings of the section on statistical Graphics of the American Statistical Association*, 1994.

[8] R. Chellappa and S. Chatterjee. Classification of textures using gaussian markov random fields. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-33(4):pp. 959–963, 1985.

[9] C.-C. Chen and C.-C. Chen. Filtering methods for texture discrimination. *Pattern Recognition Letters*, 20(8):pp. 783–790, 1999.

[10] F. S. Cohen, Z. Fan, and M. A. Patel. Classification of rotated and scaled textured images using gaussian markov random field models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(2):pp. 192–202, 1991.

[11] J. N. Darroch, S. L. Lauritzen, and T. P. Speed. Markov fields and log-linear interaction models for contingency tables. *The Annals of Statistics*, 8(3):pp. 522–539, 1980.

[12] A. P. Dawid. Conditional independence in statistical theory. *Journal of the Royal Statistical Society. Series B*, 41(1):pp. 1–31, 1979.

[13] I. Gohberg, T. Kailath, and I. Koltracht. Efficient solution of linear systems of equations with recursive structure. *Linear Algebra and Its Applications*, 80:pp. 81–113, 1986.

[14] I. Gohberg, T. Kailath, I. Koltracht, and P. Lancaster. Linear complexity parallel algorithms for linear systems of equations with recursive structure. *Linear Algebra and Its Applications*, 88/89:pp. 271–315, 1987.

[15] R. M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):pp. 786–804, 1979.

[16] M. Hassner and J. Sklansky. The use of markov random fields as models of texture. *Computer Graphics and Image Processing*, 12(4):pp. 357–370, 1980.

[17] G. Heinig and K. Rost. *Algebraic Methods for Toeplitz-like Matrices and Operators*. Akademie-Verlag, Berlin, Birkhäuser, Boston, 1984.

[18] A. K. Jain and F. Farrokhnia. Unsupervised texture segmentation using gabor filters. *Pattern Recognition*, 24(12):pp. 1167–1186, 1991.

[19] T. Kailath, S. Y. Kung, and M. Morf. Displacement ranks of matrices and linear equations. *Journal of Mathematical Analysis and Applications*, 68:pp. 395–407, 1979.

[20] T. Kailath and A. H. Sayed. *Fast reliable algorithms for matrices with structure*. SIAM, 1999.

[21] P. Lévy. *Processes stochastiques et mouvement brownien*. Gauthier-Villars, 1948.

[22] P. Lévy. A special problem of brownian motion and a general theory of gaussian random functions. In *Proceedings 3rd Berkeley Symp. Mathematical Statistics and Probability*, 1956.

[23] The MathWorks Inc., Natick, Massachusetts. *MATLAB 7.5.0*, 2007.

[24] P. Moran. A gaussian markovian process on a square lattice. *Journal of Applied Probability*, 10(1):pp. 54–62, 1973.

[25] P. A. P. Moran. Necessary conditions for markovian processes on a lattice. *Journal of Applied Probability*, 10(3):pp. 605–612, 1973.

[26] J. M. F. Moura and N. Balram. Recursive structure of noncausal gauss-markov random fields. *IEEE Transactions on Information Theory*, 38(2):pp. 334–354, 1992.

[27] V. Y. Pan. On computations with dense strucutred matrices. *Mathematics of Computation*, 55(191):pp. 179–190, 1990.

[28] V. Y. Pan. *Structured matrices and polynomials*. Springer, 2001.

[29] V. Y. Pan and X. Wang. Inversion of displacement operators. *SIAM Journal on Matrix Analysis and Applications*, 24(3):pp. 660–677, 2002.

[30] T. Randen and J. H. Husøy. Filtering for texture classification: A comparative study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):pp. 291–310, 1999.

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

[32] R.L.Kashyap, R. Chellappa, and A.Khotanzad. Texture classification using features derived from random field models. *Pattern Recognition Letters*, 1(4):pp. 43–50, 1982.

[33] Y. A. Rosanov. On gaussian fields with given conditional distributions. *Theory of Probability and Applications*, 12(3):pp. 381–391, 1967.

[34] H. Rue and L. Held. *Gaussian Markov random fields: theory and applications*. CRC Press, 2005.

[35] T. P. Speed and H. T. Kiiveri. Gaussian markov distributions over finite graphs. *The Annals of Statistics*, 14(1):pp. 138–150, 1986.

[36] M. Tuceryan and A. K. Jain. Texture analysis. In C. H. Chen, L. F. Pau, and P. S. P. Wang, editors, *The Handbook of Pattern Recognition and Computer Vision (2nd Edition)*, chapter 2, pages 207–248. World Scientific Publishing Co., 1998.

[37] P. Whittle. On stationary processes in the plane. *Biometrika*, 41:pp. 434–449, 1954.

[38] E. Wong. Two-dimensional random fields and representation of images. *SIAM Journal on Applied Mathematics*, 16(4):pp. 756–770, 1968.

[39] J. W. Woods. Two-dimensional discrete markovian fields. *IEEE Trans. on Inform. Theory*, IT-18(2):pp. 232–240, 1972.

[40] J. Zhang and T. Tan. Brief review of invariant texture analysis methods. *Pattern Recognition*, 35(3):pp. 735–747, 2002.