Università di Firenze, Università di Perugia, INdAM consorziate nel CIAFM

DOTTORATO DI RICERCA

IN MATEMATICA, INFORMATICA, STATISTICA

CURRICULUM IN MATEMATICA

CICLO XXXIV

Sede amministrativa Università degli Studi di Firenze

Coordinatore Prof. Matteo Focardi

Ill-Posed Problems in Computer Vision

Settore Scientifico Disciplinare MAT/08

Dottorando:

Valentina Giorgetti

Tutore

Prof. Ivan Gerace

Coordinatore

Prof. Matteo Focardi

Anni 2018/2021

Contents

Introduction

The visual reconstruction problems have often an ill-posed nature. In this thesis we deal with analyzing and solving three kinds of visual reconstruction problems: Blind Source Separation, Demosaicing and Deblurring.

The demosaicing problem is related to the acquisition of RGB color images by means of CCD digital cameras. In the RGB model, each pixel of a digital color image is associated to a triple of numbers, which indicate the light intensity of the red, green and blue channel, respectively. However, most cameras use a single sensor, associated with a color filter that allows only the measure at each pixel of the reflectance of the scene at one of the three colors, according to a given scheme or pattern, called *Color Filter Array* (CFA). For this reason, at each pixel, the other two missing colors should be estimated. Different CFA's are proposed for the acquisition (see also [13, 86, 92]). The most common is the Bayer pattern (see also [15]). In this scheme, the numbers of pixels in which the green color is sampled are double with respect to those associated with the red and blue channels, because of the higher sensibility of the human eye to the green wavelengths. If we decompose the acquired image into three channels, we obtain three downsampled grayscale images, so that demosaicing could be interpreted as interpolating grayscale images from sparse data. In most cameras, demosaicing is a part of the processing required to obtain a visible images. The camera's built-in-firmware is substantially based on fast local interpolation algorithms.

The heuristic approaches, which do not try to solve an optimization problem defined in mathematical terms, are widely used in the literature. These methods, in general, are very fast. Our proposed technique is of heuristic kind. In general, the heuristic techniques consist of filtering operations, which are formulated by means of suitable observations on color images. The non-

adaptive algorithms, among which bilinear and bicubic interpolation, yield satisfactory results in smooth regions of an image, but they can fail in textured or edge areas. Edge-directed interpolation is an adaptive approach, where, by analyzing the area around each pixel, we choose the possible interpolation direction. In practice, the interpolation direction is chosen to avoid interpolating across the edges. In [89], for each pixel the horizontal and vertical gradients are compared with a constant threshold. If the gradient in one direction is greater than the threshold, then interpolation is not performed along this direction. Some other direct interpolation methods use larger neighborhoods by examining different color channels. In [110], to determine the edges of the green channels, the red and blue channels are employed. On the other hand, to determine the edges of the red and blue channels, some discrete derivation operators of the second order are used, while in [102], to determine the edges in the various channels, a suitable Jacobian operator is applied. In [90], local homogeneity is used as an indicator to choose horizontally or vertically interpolated intensities. Thanks to homogeneity-directed interpolation, the luminance and chrominance values have to be similar in a suitable neighborhood. In demosaicing it is often assumed that the differences or the ratios of the intensity values in different channels are locally constant (see also [2, 89, 108, 110, 137, 150, 161]). In [108] the probability of having an edge in a certain direction is determined and used to find the weights relative to the weighted average employed as an interpolation operator. In this algorithm, the color channels are updated iteratively according to the constant color ratio condition. In [115] a similar algorithm is proposed, where 7-size neighborhoods are employed to find the edges of the green channel, and 5×5-size neighborhoods are used to determine the edges of the red and blue channels. An analogous algorithm is defined in [165], where the interpolation can be done also in the diagonal direction, while in [158] the weighted directional interpolation is used by means of a fuzzy membership assignment. In [3] a second order operator is employed as a correction term.

To have more accurate results, several techniques, which use iterative methods, are proposed. However, they have a higher computational cost with respect to the heuristic techniques. One of well-known techniques is the algorithm of *Alternate Projections* (AP) (see [78]), which uses the strong correlation between the high frequences of the three colored components, by projecting alternately the estimated image in a constraint of observation and in a constraint which imposes similarity between the red and green edges and between the blue and green edges, until a fixed

point is found. Another widely used technique is regularization (see also [70, 124]). The algorithm in [107] is based on interpolation in a residual domain. The residuals are the differences between the observed and estimated pixel values which minimize a Laplacian energy.

The algorithm here presented consists of three steps. The first two ones are initialization steps, while the third one is an iterative steps. In the first one, the missing valued in the green component are determined, in particular a weighted average-type technique is used. The weights are determined in an edge-directed approach, in which we consider also the possible edges in the red and blue components. In the second step, we determine the missing values in the red and blue components. In this case we use two alternative techniques, according to the position of the involved pixel in the Bayer pattern. In the first technique, the missing value is determined by imposing that the second derivative of the intensity value of the red/blue channel is equal to the second derivative of the intensity values of the green channel. This is done according to the proposed approaches in the AP algorithm and the regularization algorithm given in [70]. In particular, in [70] a constraint is imposed, to get the derivatives of all channels similar as soon as possible. At the third step, all values of the three channels are recursively updated, by means of a constant-hue-based technique. In particular, we assume the constant color difference. The technique we propose at this step is similar to that used by W. T. Freeman in [65]. Indeed, even here a median filter is employed, in order to correct small spurious imperfections. We repeat iteratively the third step. However, to avoid increasing excessively the computational cost, we experimentally estimate that only four iterations are necessary to obtain an accurate demosaicing. We call our technique as *Local Edge Preserving* (LEP) algorithm. The results related to this technique have been published in [31].

In this thesis, we also propose an algorithm for image demosaicing that does not work within the framework of the regularization approaches and is suited, in a natural way, to deal with noisy data. More precisely, we propose an algorithm for joint demosaicing and denoising. Regularization requires the adoption of constraints for the solution. The constraints we consider are intra-channel and inter-channel local correlation. With respect to the intra-channel correlation, we assume the intensity of each channel to be locally regular, i.e. piecewise smooth, so that also noise can be removed. We describe this constraint through stabilizers that are functions discouraging intensity discontinuities of first, second and third order in a selective way, so that those

associated to truly edges in the scene are left to emerge. This allows to describe scenes even very complex. Indeed, first order local smoothness characterizes images consisting of constant patches, second order local smoothness describes patches whose pixels have values varying linearly, while third order local smoothness is used to represent images made up of quadratic-valued patches. As per the inter-channel correlation, we enforce it in correspondence with the intensity discontinuities, by means of constraints that promote their amplitude in the three channels to be equal almost everywhere.

Note that all these constraints are by no means biased in favor of one of the three channels, nor the geometry of the sampling pattern is in any way exploited. Thus, the method we propose is completely independent of the CFA considered, although, in the experimental result section, we present its application to images mosaiced through the Bayer CFA.

All the above constraints, including the data fidelity term, are merged in a non-convex energy function, whose minimizer is taken as our desired solution. The optimization is performed through an iterative deterministic algorithm entailing the minimization in a sequence of a family of approximating functions that, starting with a first componentwise convex function, gradually converges to the original energy, as suggested in [24].

Our regularization approach can produce image solutions that exhibit reliable discontinuities of both the intensity and the gradients, despite the necessary smoothness constraints. Therefore, we propose an edge-preserving regularization approach, which means that the significant discontinuities in the reconstructed image are geometrically consistent. In the very first works proposing edge-preserving regularization, the image discontinuities were often represented by means of extra, explicit variables, the so-called "line processes" (see [66]). In that way, it was relatively easy to formulate in terms of constraints the various properties required by significant discontinuities. Nevertheless, the use of explicit line variables entails large computational costs. Thus, so-called "duality theorems" were derived (see, e.g., [33, 34]) to demonstrate the edge-preserving properties of suitable stabilizers, without introducing extra variables. In particular, we developed duality theorems to determine the properties required for a stabilizer to implicitly manage lines with the desired regularity features. In this work, we choose a suitable family of approximations with the peculiarity that each function satisfies the conditions required for an implicit treatment of geometrically significant edges, as expressed in the duality theorems proposed

in [33]. This allows a better adherence of the approximations to the ideal energy function, with a consequent better coherence with the properties required for the desired solution.

In this thesis we also study a *Blind Source Separation* (*BSS*) problem. These topics have been widely investigated since the end of the last century, and have various applications.

In particular, we analyze the digital reconstruction of degraded documents. We observe that weathering, powder, humidity, seeping of ink, mold and light transmission can determine the degradation of the paper and the ink of written text. Some of the consequences in damaged documents are, for instance, stains, noise, transparency of writing on the verso side and on the close pages, unfocused or overlapping characters, and so on. Historically, the first techniques of restoration for degraded documents were manual, and they led to a material restoration. Recently, thanks to the diffusion of scanners and software for reconstruction of images, videos, texts, photographs and films, several new techniques were used in the recovery and restoration of deteriorated material, like for instance digital or virtual restoration. Digital imaging for documents is very important, because it allows to have digital achieves, to make always possible the accessibility and the readability. The *Digital Document Restoration* consists of a set of processes finalized to the visual and aesthetic improvement of a virtual reconstruction of a corrupted document, without risk of deterioration.

We deal with *show-through* and *bleed-through* effects. The show-through is a front-to-back interference, caused by the transparency of the paper and the scanning process, and by means of which the text in the recto side of the document can appear also in the verso side, and conversely. The bleed-through is an intrinsic front-to-back physical deterioration caused by ink seeping, and its effect is similar to that of show-through. The physical model for the show-through distortion, is very complex, because there are the spreading of light in the paper, the features of the paper, the reflectance of the verso and the transmittance parameters. In [148], Sharma gave a mathematical model was first analyzed and then further approximated so to become easier to handle. This model describes the observed recto and verso images as mixtures of the two uncorrupted texts.

Locally, we consider a classical linear and stationary recto-verso model (see also [49, 98, 97, 99, 156]) developed for this purpose, and are concerned with the problem of estimating both the ideal source images of the recto and the verso of the document and the mixture matrix producing the bleed-through or show-through effects. This problem is ill-posed in the sense of Hadamard

(see also [75]). In fact, as the estimated mixture matrix varies, the corresponding estimated sources are in general different, and thus infinitely many solutions exist. Many techniques to solve this ill-posed inverse problem have been proposed in the literature. Among them, the *Independent Component Analysis* (ICA) methods are based on the assumption that the sources are mutually independent (see also [52]). The best-known ICA technique is the so-called FastICA (see also [98, 97, 99, 105, 118]), which by means of a fixed point iteration finds an orthogonal rotation of the prewhitened data that maximizes a measure of non-Gaussianity of the rotated components. The FastICA algorithm is a parameter-free and extremely fast procedure, but ICA is not a viable approach in our setting, as for the problem we consider there is a clear correlation among the sources. On the other hand, several techniques for ill-posed inverse problems require that the estimated sources are only mutually uncorrelated. In this case, the estimated sources are determined via a linear transformation of the data, which is obtained by imposing either an orthogonality condition, as in *Principal Component Analysis* (PCA) (see also [49, 155, 156]), or an orthonormality condition, as in *Whitening* (W) and *Symmetric Whitening* (SW) techniques (see also [49, 155, 156]). These approaches all require only a single and very fast processing step. In [49, 156] it is observed that the results obtained by means of the SW method are substantially equivalent to those produced by an ICA technique in the symmetric mixing case.

Here we assume that the sum of all rows of the mixing matrix is equal to one, since we expect the color of the background of the source to be the same as that of the data. In our setting, we change the variables of the data so that high and low light intensities correspond to presence and absence of text in the document, respectively, and we impose a nonnegativity constraint on the estimated sources (see also [42, 50, 74, 134]). We define the *overlapping matrix* of both the observed data and the ideal sources, a quantity related to the cross-correlation between the signals. From the overlapping matrix we can deduce the *overlapping level*, which measures the similarity between the front and the back of the document.

In order to obtain an accurate estimate of the sources, it is necessary to determine a correct source overlapping level. To this aim, we propose the following iterative procedure. At each iteration, given the current source overlapping level, we estimate the mixture matrix that produces the sources with the lowest possible source overlapping level among those having light intensity in the desired range. This mixture matrix is computed by means of a suitable symmetric factor-

ization of the data overlapping matrix. We then use the estimated sources to update the source overlapping level, and iterate the procedure until a fixed point is reached. At the fixed point, the corresponding source overlapping level is the smallest one that allows to estimate the ideal recto and verso sides with the desired properties. We consider this level as an adequate estimate of the ideal source overlapping level. Thus, by means of this technique, we can estimate not only the ideal sources and the mixture matrix, but also the source overlapping level, a value that indicates the correlation between the ideal sources. Therefore, our method can be classified as a *Correlated Component Analysis* (CCA) technique (see also [14, 139, 152, 153]). We refer to this method as the *Minimum Amount of Text Overlapping in Document Separation* (MATODS) algorithm. Similarly to the FastICA technique, the MATODS algorithm is a parameter-free and extremely fast procedure. We use the MATODS algorithm to solve the non-stationary and locally linear model we propose, and in particular we present an extension of this technique that fits this model, which we call the *Not Invariant for Translation MATODS* (NIT-MATODS) algorithm. The related results have been published in [27].

In this thesis we modify the MATODS algorithm to deal with the derivatives of the images of the original sources. In this case, we assume that the overlapping level is equal to zero. By means of our experimental results, we show that the proposed technique improves the results obtained by MATODS in terms both of accuracy of the estimates and of computational costs. We refer to this method as the *Zero Edge Overlapping in Document Separation* (ZEODS) algorithm. The obtained results are published in [32].

In [148], Sharma gave a mathematical model was first analyzed and then further approximated so to become easier to handle. This model describes the observed recto and verso images as mixtures of the two uncorrupted texts. A nonlinear modified Sharma model is proposed in [119, 132, 133, 145]. Some nonlinear models which assume that the interference levels depend on the location are presented in [71, 104, 157]. So, the model turns to be non-stationary, that is not translation invariant. The algorithms in [71, 157] for the resolution of the related inverse problem are fast heuristics. In [27], a non-stationary model is proposed. However, in order to obtain more precise results, a computationally more expensive regularized problem has been sketched in [69] and [155]). Now we analyze in detail the iterative technique to solve such a model, in which the sources, the blur operators and the interference level are computed separately at every step, until

a fixed point is found. In this work, in particular, we deal with determining the interference level, by fixing the blur operators and the ideal sources. To this aim, we use a GNC-type technique (see, e.g., [18, 24, 25, 33, 34, 87, 127, 129, 130, 131, 140]). In forthcoming papers, the steps about finding the blur operators and the ideal sources will be treated. The results concerning such a technique have been published in [29].

The problem of restoring images consists of estimating the original image, starting from the observed image and the supposed blur. In our model, we suppose to know the blur mask. In general, this problem is ill–conditioned and/or ill–posed in the Hadamard sense (see also [84]). Thanks to known regularization techniques (see, e.g., [17, 55, 67]), it is possible to reduce this problem to a well–posed problem, whose solution is the minimum of the so-called *primal energy function*, which consists of the sum of two terms. The former indicates the faithfulness of the solution to the data, and the latter is in connection with the regularity properties of the solution (see also [55, 66]). In order to obtain more realistic restored images, the discontinuities in the intensity field is considered (see also [66]). Indeed, in images of real scenes, there are some discontinuities in correspondence with edges of several objects. To deal with such discontinuities, we consider some line variables (see also [66]). It is possible to minimize a priori the primal energy function in these variables, to determine a *dual energy function* (see, e.g., [34, 46, 67]), which treats implicitly discontinuities. Indeed, minimizing the dual energy function is more computationally efficient than minimizing directly the primal energy function. In general, the dual energy function has a quadratic term, related to the faithfulness with the data, and a not necessarily convex addend, the regularization term. In order to link these two kinds of energy functions, some suitable duality theorems are used (see, e.g., [10, 17, 18, 23, 24, 33, 34, 67]).

In order to improve the quality of the reconstructed images, it is possible to consider a dual energy function which implicitly treats Boolean line variables. The proposed duality theorems can be used even with such a function. However, the related dual energy function is not necessarily convex. So, to minimize it, we use a GNC-type technique, which considers as first convex approximation the proposed convex dual energy function (see also [18, 34, 127, 129, 130, 131, 140]).

It is possible to verify experimentally that the more expensive minimization is the first one, because the other ones just start with a good approximation of the solution. Hence, when we

minimize the first convex approximation, we will approximate every block of the blur operator by matrices whose product can be computed by a suitable fast discrete transform. As every block is a symmetric Toeplitz matrix, we deal with determining a class of matrices easy to handle from the computational point of view, which yield a good approximation of the Toeplitz matrices.

Toeplitz-type linear systems arise from numerical approximation of differential equations. Moreover, in restoration of blurred images, it is often dealt with Toeplitz matrices. (see, e.g., [33, 34, 59]). Thus, in this thesis we investigate a particular class, which is a sum of two families of simultaneously diagonalizable real matrices, whose elements we call β*-matrices*. Such a class includes both circulant and reverse circulant matrices. Symmetric circulant matrices have several applications to ordinary and partial differential equations (see, e.g., [60, 73, 81, 82]), images and signal restoration (see, e.g., [41, 88]), graph theory (see, e.g., [51, 58, 66, 68, 79, 80]). Reverse circulant matrices have different applications, for instance in exponential data fitting and signal processing (see, e.g., [7, 11, 57, 136, 138]). The obtained results have been published in [30].

The thesis is structured as follows. In Chapter 1 we deal with the demosaicing problem, proposing a fast technique which locally estimates the edges. In Chapter 2 we treat the same problem, by giving a regularization technique for solving it. In Chapter 3 we consider the BSS problem for ancient documents, proposing a technique which uses symmetric factorizations. In Chapter 4 we modify the technique illustrated in the previous chapter, by introducing discontinuities. In Chapter 5 we deal with the BSS problem, by giving a regularization technique, and in particular we study the estimates of the interference levels. In Chapter 6 we treat the problem of image deblurring, and in particular we analyze how symmetric Toeplitz operators can be approximated in the proposed GNC technique.

The structure of the thesis, in terms of the addressed problems and the used techniques can be summarized in the Table 1.

Problem	Local Technique	Regularization
Demosaicing	Chapter 1	Chapter 2
BSS	Chapter 3 and Chapter 4	Chapter 5
Deblurring		Chapter 6

Table 1: Structure of the thesis

Chapter 1

A fast algorithm for the demosaicing problem

This chapter is structured as follows. In Section 2 we give a mathematical formulation of the demosaicing problem. In Section 3 we describe the initialization of the proposed algorithm, which consists of the two first steps aforementioned. In Section 4 we give the third iterative step of our algorithm, highlighting the differences with the Freeman filter. In Section 5 our experimental results are presented. This section consists of two parts. In the first one, we determine the best detection function which can be used in order to evaluate the edges. In the second one, we compare our algorithm with some other techniques recently proposed in the literature and we show how the LEP method gives in mean more accurate reconstructions than the other considered algorithms.

1.1 The demosaicing problem

An RGB (red-green-blue) *color image* with height n and width m is a vector of the type

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}^{(\mathbf{r})} \\ \mathbf{x}^{(\mathbf{g})} \\ \mathbf{x}^{(\mathbf{b})} \end{pmatrix} \in \mathbb{R}^{3n \cdot m}$$

where $\mathbf{x}^{(\mathbf{r})}$, $\mathbf{x}^{(\mathbf{g})}$, $\mathbf{x}^{(\mathbf{b})} \in \mathbb{R}^{n\cdot m}$ are the red, green and blue channels according to the lexicographic order, respectively. We consider the problem of acquisition of data from a digital camera, and call it *mosaicing problem*. Given an ideal image $\mathbf{x} \in \mathbb{R}^{3n\cdot m}$, the acquired or *mosaiced image* is defined by

$$\mathbf{y} = \begin{pmatrix} \mathbf{y}^{(\mathbf{r})} \\ \mathbf{y}^{(\mathbf{g})} \\ \mathbf{y}^{(\mathbf{b})} \end{pmatrix} = M\mathbf{x}$$

where $\mathbf{y} \in \mathbb{R}^{3n\cdot m}$ and $M \in \mathbb{R}^{(3n\cdot m)\times(3n\cdot m)}$ is a linear operator defined by setting

$$M = \begin{pmatrix} M^{(r)} & O & O \\ O & M^{(g)} & O \\ O & O & M^{(b)} \end{pmatrix}$$

$O \in \mathbb{R}^{(n\cdot m)\times(n\cdot m)}$ is the null matrix, and $M^{(r)}, M^{(g)}, M^{(b)} \in \mathbb{R}^{(n\cdot m)\times(n\cdot m)}$ are diagonal matrices whose principal entries, if we use the Bayer pattern (see Figure 1.1), are given by

$$m^{(r)}_{(i,j),(i,j)} = \begin{cases} 1, & \text{if } i \equiv_2 0 \text{ and } j \equiv_2 0 \\ 0, & \text{otherwise} \end{cases}$$

$$m^{(g)}_{(i,j),(i,j)} = \begin{cases} 1, & \text{if } i \not\equiv_2 j \\ 0, & \text{otherwise} \end{cases}$$

$$m^{(b)}_{(i,j),(i,j)} = \begin{cases} 1, & \text{if } i \equiv_2 1 \text{ and } j \equiv_2 1 \\ 0, & \text{otherwise} \end{cases}$$

where the symbol $i \equiv_2 j$ indicates that $i - j$ is even.

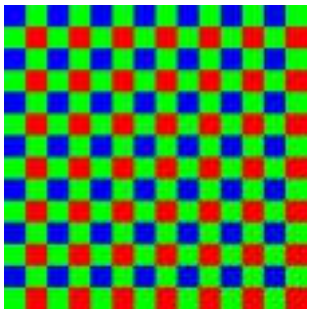


Figure 1.1: Bayer Pattern.

The corresponding demosaicing problem is the associated inverse problem, that is to determine the ideal color image $\mathbf{x}$, knowing the mosaiced image $\mathbf{y}$ and the linear operator M. An

inverse problem is said to be *well-posed* (in the sense of Hadamard) if and only if the solution exists, is unique and stable with respect to data variation. A not well-posed problem is said to be *ill-posed* (see also [83, 84]). Note that the demosaicing problem is ill-posed, since the matrix M in (1.1) is singular, as is readily seen, and so there are infinitely many solutions.

1.2 The initialization of the proposed algorithm

In the initialization phase we proceed as follows: first we initialize the green channel, since in this channel we have more data than in the other ones, and successively we update the other two.

1.2.1 The initialization of the green channel

We refer to a *clique* as a pair of adjacent pixels. Every missing value of the green channel is initialized by a weighted mean of the known green values in its neighborhood. The weights of the considered mean take into account possible discontinuities in a set of adjacent cliques. We consider cliques both in the blue and in the red channel, since it is well-known that there is a correlation between the discontinuities in the various channels related to edges, such as object borders and textures (see e.g. [70]).

Here we distinguish three cases: the first one is when we have the value of the green light intensity on a pixel; the second one is when we the blue value of the involved pixel is known, that is when i and j are both odd; the third one is when the red value on the considered pixel is known, namely when i and j are both even.

The first approximation $\mathbf{x}^{(g,0)}$ of the green ideal image $\mathbf{x}^{(g)}$ is given by

$$x_{(i,j)}^{(g,0)} = \begin{cases} y_{(i,j)}^{(g)} & \text{if } i \not\equiv_2 j \\ \dfrac{t_1 y_{(i-1,j)}^{(g)} + t_2 y_{(i+1,j)}^{(g)} + t_3 y_{(i,j-1)}^{(g)} + t_4 y_{(i,j+1)}^{(g)}}{t_1 + t_2 + t_3 + t_4} & \text{if } i \equiv_2 1 \text{ and } j \equiv_2 1 \\ \dfrac{t_5 y_{(i-1,j)}^{(g)} + t_6 y_{(i+1,j)}^{(g)} + t_7 y_{(i,j-1)}^{(g)} + t_8 y_{(i,j+1)}^{(g)}}{t_5 + t_6 + t_7 + t_8} & \text{if } i \equiv_2 0 \text{ and } j \equiv_2 0 \end{cases}$$

Note that, in the first case, we keep the value we already have. In the second case, we do a weighted mean of the intensity values taken on the adjacent pixels where the green value is

known. The weights t_1, t_2, t_3, t_4 of the mean are computed by using the green and the blue channels. We define

$$\mathbf{e} = \begin{pmatrix} e_1 = (i-1, j) \\ e_2 = (i+1, j) \\ e_3 = (i, j-1) \\ e_4 = (i, j+1) \end{pmatrix} \quad \mathbf{f} = \begin{pmatrix} f_1 = (i, j-1) \\ f_2 = (i, j-1) \\ f_3 = (i-1, j) \\ f_4 = (i-1, j) \end{pmatrix} \tag{1.1}$$

$$\mathbf{p} = \begin{pmatrix} p_1 = (i, j+1) \\ p_2 = (i, j+1) \\ p_3 = (i+1, j) \\ p_4 = (i+1, j) \end{pmatrix} \quad \mathbf{q} = \begin{pmatrix} q_1 = (i-2, j) \\ q_2 = (i+2, j) \\ q_3 = (i, j-2) \\ q_4 = (i, j+2) \end{pmatrix}$$

In particular, we get

$$t_k = \phi(\tau_k) \tag{1.2}$$

where $k = 1,2,3,4$, ϕ is a suitable positive decreasing *detection function* and τ_k is defined by

$$\tau_k = |y^{(g)}_{e_k} - y^{(g)}_{f_k}| + |y^{(g)}_{e_k} - y^{(g)}_{p_k}| + |y^{(b)}_{(i,j)} - y^{(b)}_{q_k}|$$

where the pixels e_k, f_k, p_k and q_k are as in (1.1). When the differences between the green values on the pixels e_k and f_k (see the yellow arc in Figure 1.2 for $k = 1$), e_k and p_k (see the cyan arc in Figure 1.2 for $k = 1$), and between the blue values on the pixels (i, j) and q_k (see the brown arc in Figure 1.2 for $k = 1$) are small enough, then we can assume that there are no discontinuities between the pixels e_k and (i, j) (see the red line in Figure 1.2 for $k = 1$). So, in the calculus of the green value on the pixel (i, j), we give a large weight t_1 to the green value in the position e_k. Thus, when the value τ_k is small, the probability of having a discontinuity between the pixels (i, j) and e_k in the green channel is large, and vice versa. The computation of τ_1 is illustrated in Figure 1.2. For $k = 1,2,3$ the computation of τ_k can be described by an appropriately rotated similar figure.

Even in the third case, we compute the weighted mean of the intensity values taken on the adjacent pixels where the green value is known. The weights t_{4+k}, $k = 1,2,3,4$ of the mean are computed by using the green and the red channels.

In particular, $t_{4+k} = \phi(\tau_{4+k})$, where

$$\tau_{4+k} = |y^{(g)}_{e_k} - y^{(g)}_{f_k}| + |y^{(g)}_{e_k} - y^{(g)}_{p_k}| + |y^{(r)}_{(i,j)} - y^{(r)}_{q_k}|$$

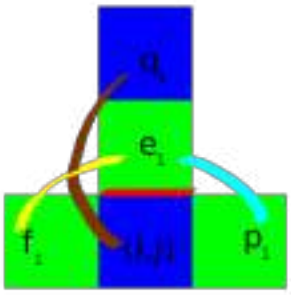


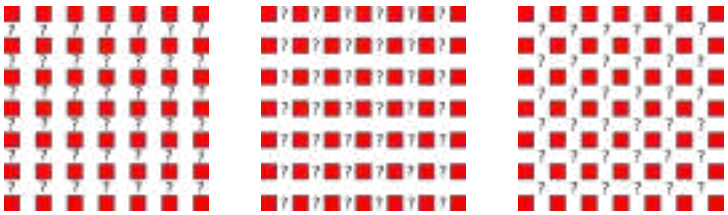
Figure 1.2: Computation of τ_k for $k = 1$.

where e_k, f_k, p_k and q_k are as in (1.1).

We argue analogously as in the computation of the weight t_k, $k = 1,2,3,4$, where the role of the blue channel is played by the red component.

1.2.2 The initialization of the red values

Here we distinguish four cases: the first one is when we already know the red value of a pixel; the second one is when we know the red values in the two adjacent pixels in the same column, that is i is odd and j is even (see Figure 1.3 (a)); the third one is when we know the red values in the two adjacent pixels in the same row, namely i is even and j is odd (see Figure 1.3 (b)); the fourth one is when we know the red values of the pixels adjacent in the corners of the involved pixel, that is i and j are both odd (see Figure 1.3 (c)). In the second and in the third case we equalize the second derivatives of the red and the green channels previously computed. In the last case we use the computed values of the red channel to determine the weights of a suitable mean. So, we define the initial estimate $\mathbf{x}^{(r,0)}$ of the red ideal image $\mathbf{x}^{(r)}$ by



(a) Second case. (b) Third case. (c) Fourth case.

Figure 1.3: Different cases in the initialization of the red channel.

$$x^{(r,0)}_{(i,j)} = \begin{cases} y^{(r)}_{(i,j)} & \text{if } i \equiv_2 0 \text{ and } j \equiv_2 0 \\ \dfrac{y^{(r)}_{(i-1,j)} + y^{(r)}_{(i+1,j)} - x^{(g,0)}_{(i-1,j)} + 2y^{(g)}_{(i,j)} - x^{(g,0)}_{(i+1,j)}}{2} & \text{if } i \equiv_2 1 \text{ and } j \equiv_2 0 \\ \dfrac{y^{(r)}_{(i,j-1)} + y^{(r)}_{(i,j+1)} - x^{(g,0)}_{(i,j-1)} + 2y^{(g)}_{(i,j)} - x^{(g,0)}_{(i,j+1)}}{2} & \text{if } i \equiv_2 0 \text{ and } j \equiv_2 1 \\ \dfrac{t_9 x^{(r,0)}_{(i-1,j)} + t_{10} x^{(r,0)}_{(i+1,j)} + t_{11} x^{(r,0)}_{(i,j-1)} + t_{12} x^{(r,0)}_{(i,j+1)}}{t_9 + t_{10} + t_{11} + t_{12}} & \text{if } i \equiv_2 1 \text{ and } j \equiv_2 1 \end{cases}$$

Note that, in the first case, we keep the value which we already have. In the second case, we pose that the finite difference of the second order in the vertical direction of the red channel coincides with that of the green channel, which we have already initialized, namely

$$y^{(r)}_{(i-1,j)} - 2x^{(r,0)}_{(i,j)} + y^{(r)}_{(i+1,j)} = x^{(g,0)}_{(i-1,j)} - 2y^{(g)}_{(i,j)} + x^{(g,0)}_{(i+1,j)} \tag{1.3}$$

Since we know $\mathbf{x}^{(g,0)}$, $\mathbf{y}^{(g)}$ and $\mathbf{y}^{(r)}$, we can deduce the value of $x^{(r,0)}_{(i,j)}$ from (1.3).

In the third case, we impose that the finite difference of the second order in the horizontal direction of the red channel coincides with that of the green channel, just already initialized, that is

$$y^{(r)}_{(i,j-1)} - 2x^{(r,0)}_{(i,j)} + y^{(r)}_{(i,j+1)} = x^{(g,0)}_{(i,j-1)} - 2y^{(g)}_{(i,j)} + x^{(g,0)}_{(i,j+1)} \tag{1.4}$$

By proceeding analogously as above, we obtain the value of $x^{(r,0)}_{(i,j)}$ from (1.4).

In the fourth case, we do a weighted mean of the intensity values taken on the adjacent pixels where the red value has just been computed. The weights t_{8+k}, $k = 1,2,3,4$, of the mean are calculated by using the just initialized red channel and the observed blue channel. In particular, t_{8+k} is given by $\phi(\tau_{8+k})$, $k = 1,2,3,4$, where ϕ is the detection function used in initializing the green channel, and

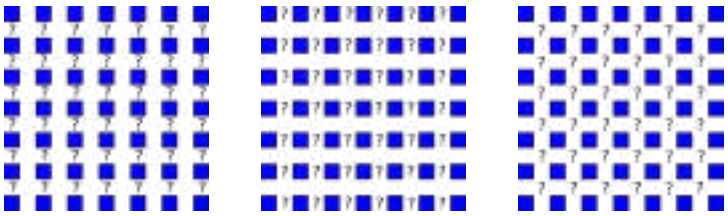
$$\tau_{8+k} = |x^{(r,0)}_{e_k} - x^{(r,0)}_{f_k}| + |x^{(r,0)}_{e_k} - x^{(r,0)}_{p_k}| + |y^{(b)}_{(i,j))} - y^{(b)}_{q_k}|$$

where e_k, f_k, p_k and q_k are as in (1.1). When the differences between the red values on the pixels e_k and f_k, e_k and p_k, and between the blue values on the pixels (i, j) and q_k, are sufficiently small,

then we can suppose that there are no edges between the pixels (i, j) and e_k. So, in the calculus of the red value on the pixel (i, j), we have a large weight t_{8+k}, $k = 1, 2, 3, 4$, in correspondence with the red value in the position e_k.

1.2.3 The initialization of the blue values

Also in this setting, we distinguish four cases: the first one is given when we know the blue value of a pixel; the second one is when we know the blue values in the two adjacent pixels in the same column, that is i is even and j is odd (see Figure 1.4 (a)); the third one is when we know the blue values in the two adjacent pixels in the same row, namely i is odd and j is even (see Figure 1.4 (b)); the fourth one is when we know the blue values of the pixels adjacent in the corners of the involved pixel, that is i and j are both even (see Figure 1.4 (c)). In the second and third cases we equalize the second derivatives of the blue and the green channels previously calculated. In the last case we use the computed values of the blue channel to determine the weights of a suitable mean.



(a) Second case. (b) Third case. (c) Fourth case.

Figure 1.4: Different cases in the initialization of the blue channel.

Thus, we define the estimate $\mathbf{x}^{(b,0)}$ of the blue ideal image $\mathbf{x}^{(b)}$ by

$$x^{(b,0)}_{(i,j)} = \begin{cases} y^{(b)}_{(i,j)} & \text{if } i \equiv_2 1 \text{ and } j \equiv_2 1 \\ \dfrac{y^{(b)}_{(i-1,j)} + y^{(b)}_{(i+1,j)} - x^{(g,0)}_{(i-1,j)} + 2y^{(g)}_{(i,j)} - x^{(g,0)}_{(i+1,j)}}{2} & \text{if } i \equiv_2 0 \text{ and } j \equiv_2 1 \\ \dfrac{y^{(b)}_{(i,j-1)} + y^{(b)}_{(i,j+1)} - x^{(g,0)}_{(i,j-1)} + 2y^{(g)}_{(i,j)} - x^{(g,0)}_{(i,j+1)}}{2} & \text{if } i \equiv_2 1 \text{ and } j \equiv_2 0 \\ \dfrac{t_{13}x^{(b,0)}_{(i-1,j)} + t_{14}x^{(b,0)}_{(i+1,j)} + t_{15}x^{(b,0)}_{(i,j-1)} + t_{16}x^{(b,0)}_{(i,j+1)}}{t_{13} + t_{14} + t_{15} + t_{16}} & \text{if } i \equiv_2 0 \text{ and } j \equiv_2 0 \end{cases}$$

Note that, in the first case, we keep the value which we already have.

In the second case, analogously as before, we impose

$$y^{(b)}_{(i-1,j)} - 2x^{(b,0)}_{(i,j)} + y^{(b)}_{(i+1,j)} = x^{(g,0)}_{(i-1,j)} - 2y^{(g)}_{(i,j)} + x^{(g,0)}_{(i+1,j)} \tag{1.5}$$

As we know $\mathbf{x}^{(g,0)}$, $\mathbf{y}^{(g)}$ and $\mathbf{y}^{(b)}$, we derive the value of $x^{(g,0)}_{(i,j)}$ from (1.5).

In the third case, similarly as above, we get

$$y^{(b)}_{(i,j-1)} - 2x^{(b,0)}_{(i,j)} + y^{(b)}_{(i,j+1)} = g^{(0)}_{(i,j-1)} - 2y^{(g)}_{(i,j)} + g^{(0)}_{(i,j+1)} \tag{1.6}$$

By arguing as in the previous section, we deduce the value of $b^{(0)}_{(i,j)}$ from (1.6).

In the fourth case, we do a weighted mean of the intensity values of the adjacent pixels where the blue value has just been computed. The weights t_{12+k}, $k = 1,2,3,4$, of the mean are calculated by using the observed red channel and the just initialized blue channel.

Analogously as before, we obtain $t_{12+k} = \phi(\tau_{12+k})$, where

$$\tau_{12+k} = |x^{(b,0)}_{e_k} - x^{(b,0)}_{f_k}| + |x^{(b,0)}_{e_k} - x^{(b,0)}_{p_k}| + |y^{(r)}_{(i,j)} - y^{(r)}_{q_k}|,$$

where e_k, f_k, p_k and q_k are as in (1.1).

1.3 The iterative phase of the proposed algorithm

A classical filter, often used to solve the demosaicing problem, is the *Freeman filter* (see also [65]). The phase described in this section is a suitable modification of this filter. The Freeman filter performes the initialization phase by means of the bilinear filter, which works as follows. When the value of a certain color of a pixel is not available, such a value is computed by the arithmetic mean of the values of that color, which are assumed in the neighborhood of this pixel, that is the *bilinear estimation* $\widetilde{\mathbf{x}} = (\widetilde{\mathbf{r}}, \widetilde{\mathbf{g}}, \widetilde{\mathbf{b}})$ is given as

$$\widetilde{g}_{(i,j)} = \begin{cases} y^{(g)}_{(i,j)} & \text{if } i \not\equiv_2 j \\ \dfrac{y^{(g)}_{(i-1,j)} + y^{(g)}_{(i+1,j)} + y^{(g)}_{(i,j-1)} + y^{(g)}_{(i,j+1)}}{4} & \text{otherwise} \end{cases}$$

$$\widetilde{r}_{(i,j)} = \begin{cases} y^{(r)}_{(i,j)} & \text{if } i \equiv_2 0 \text{ and } j \equiv_2 0 \\ \dfrac{y^{(r)}_{(i,j-1)} + y^{(r)}_{(i,j+1)}}{2} & \text{if } i \equiv_2 0 \text{ and } j \equiv_2 1 \\ \dfrac{y^{(r)}_{(i-1,j)} + y^{(r)}_{(i+1,j)}}{2} & \text{if } i \equiv_2 1 \text{ and } j \equiv_2 0 \\ \dfrac{y^{(r)}_{(i-1,j-1)} + y^{(r)}_{(i+1,j-1)} + y^{(r)}_{(i-1,j+1)} + y^{(r)}_{(i+1,j+1)}}{4} & \text{if } i \equiv_2 1 \text{ and } j \equiv_2 1 \end{cases}$$

$$\widetilde{b}_{(i,j)} = \begin{cases} y^{(b)}_{(i,j)} & \text{if } i \equiv_2 1 \text{ and } j \equiv_2 1 \\ \dfrac{y^{(b)}_{(i,j-1)} + y^{(b)}_{(i,j+1)}}{2} & \text{if } i \equiv_2 1 \text{ and } j \equiv_2 0 \\ \dfrac{y^{(b)}_{(i-1,j)} + y^{(b)}_{(i+1,j)}}{2} & \text{if } i \equiv_2 0 \text{ and } j \equiv_2 1 \\ \dfrac{y^{(b)}_{(i-1,j-1)} + y^{(b)}_{(i+1,j-1)} + y^{(b)}_{(i-1,j+1)} + y^{(b)}_{(i+1,j+1)}}{4} & \text{if } i \equiv_2 0 \text{ and } j \equiv_2 0 \end{cases}$$

Moreover, in [65] the following values are defined, by means of the median of the color differences of the channels red-green and blue-green:

$$\widetilde{rg}_{(i,j)} = \text{median}\{\widetilde{r}_{(k,l)} - \widetilde{g}_{(k,l)} : (k,l) \in B_\infty((i,j),t)\},$$

$$\widetilde{bg}_{(i,j)} = \text{median}\{\widetilde{b}_{(k,l)} - \widetilde{g}_{(k,l)} : (k,l) \in B_\infty((i,j),t)\},$$

where

$$B_\infty((i,j),t) := \{(k,l) \in \mathbb{N} \times \mathbb{N} : \|(i,j) - (k,l)\|_\infty \leq t\}, \tag{1.7}$$

with $\|(a,b)\|_\infty = \max\{|a|,|b|\}$. The median turns out to be very useul to correctly preserve the edges which are in the images. Indeed, the median filter is often used to restore images corrupted by salt-and-pepper noise, namely by the noise present only in a few pixels not adjacent each other.

In the Freeman filter it is assumed that the color differences are constant in a suitable subarea.

Thus, the *Freeman estimation* $\widehat{\mathbf{x}} = (\widehat{\mathbf{r}}^T, \widehat{\mathbf{g}}^T, \widehat{\mathbf{b}})^T$ is defined as follows:

$$\widehat{g}_{(i,j)} = \begin{cases} y^{(g)}_{(i,j)} & \text{if } i \not\equiv_2 j \\ \dfrac{\left(\widetilde{r}_{(i,j)} - \widetilde{rg}_{(i,j)}\right) + \left(\widetilde{b}_{(i,j)} - \widetilde{bg}_{(i,j)}\right)}{2} & \text{otherwise} \end{cases}$$

$$\widehat{r}_{(i,j)} = \begin{cases} y^{(r)}_{(i,j)} & \text{if } i \equiv_2 0 \text{ and } j \equiv_2 0 \\ y^{(g)}_{(i,j)} + \widetilde{rg}_{(i,j)} & \text{otherwise} \end{cases}$$

$$\widehat{b}_{(i,j)} = \begin{cases} y^{(b)}_{(i,j)} & \text{if } i \equiv_2 1 \text{ and } j \equiv_2 1 \\ y^{(g)}_{(i,j)} + \widetilde{bg}_{(i,j)} & \text{otherwise} \end{cases}$$

In this work we modify the Freeman filter as follows.

From the initial estimation $\mathbf{x}^{(0)} = ({\mathbf{x}^{(r,0)}}^T, {\mathbf{x}^{(g,0)}}^T, {\mathbf{x}^{(g,0)}}^T)^T$ we define the following variables:

$$rg^{(s)}_{(i,j)} = \text{median}\{x^{(r,s)}_{(k,l)} - x^{(g,s)}_{(k,l)} : (k,l) \in B_\infty((i,j),t)\}$$

$$bg^{(s)}_{(i,j)} = \text{median}\{x^{(b,s)}_{(k,l)} - x^{(g,s)}_{(k,l)} : (k,l) \in B_\infty((i,j),t)\}$$

$$rb^{(s)}_{(i,j)} = \text{median}\{x^{(r,s)}_{(k,l)} - x^{(b,s)}_{(k,l)} : (k,l) \in B_\infty((i,j),t)\}$$

where $s \in \mathbb{N} \cup \{0\}$. So, we define the estimates $\mathbf{x}^{(s)} = ({\mathbf{x}^{(r,s)}}^T, {\mathbf{x}^{(g,s)}}^T, {\mathbf{x}^{(b,s)}}^T)^T$ for $s = 1, 2, \ldots$ as

follows:

$$x_{(i,j)}^{(g,s)} = \begin{cases} y_{(i,j)}^{(g)} & \text{if } i \not\equiv_2 j \\ \dfrac{\left(x_{(i,j)}^{(r,s-1)} - rg_{(i,j)}^{(s-1)}\right) + \left(x_{(i,j)}^{(b,s-1)} - bg_{(i,j)}^{(s-1)}\right)}{2} & \text{otherwise} \end{cases}$$

$$x_{(i,j)}^{(r,s)} = \begin{cases} y_{(i,j)}^{(r)} & \text{if } i \equiv_2 0 \text{ and } j \equiv_2 0 \\ y_{(i,j)}^{(b)} + rb^{(s-1)} & \text{if } i \equiv_2 1 \text{ and } j \equiv_2 1 \\ y_{(i,j)}^{(g)} + rg^{(s-1)} & \text{otherwise} \end{cases}$$

$$x_{(i,j)}^{(b,s)} = \begin{cases} y_{(i,j)}^{(b)} & \text{if } i \equiv_2 1 \text{ and } j \equiv_2 1 \\ y_{(i,j)}^{(r)} - rb_{(i,j)}^{(s-1)} & \text{if } i \equiv_2 0 \text{ and } j \equiv_2 0 \\ y_{(i,j)}^{(g)} + bg_{(i,j)}^{(s-1)} & \text{otherwise} \end{cases}$$

We pose our final estimate as

$$\check{\mathbf{x}} = \lim_{s \to +\infty} \mathbf{x}^{(s)}$$

We saw experimentally that a good approximation is given by $\check{\mathbf{x}} = \mathbf{x}^{(4)}$. We call the technique associated to this estimate as *Local Edge Preserving* (LEP) algorithm.

1.4 Experimental results and discussion

In this section we present the experimental results obtained from the implementation of the proposed algorithm, which was tested for the Bayer CFA on the set of 24 Kodak sample images [109], of size 512×768, shown in Figure 1.5. This dataset represents the typical benchmark images used in the literature to compare the various demosaicing algorithms. These high quality images have been acquired as raw images, in order to minimize the compression. We have implemented our algorithm in the C language on an Ubuntu operating system by means of a computer having a processor of speed 3.40 GHz.

To define a specific LEP method, we assume that the radius t of the neighborhood of the

median filter in the equation (1.7) is equal to 1, and we experimentally choose the detection function $\phi : \mathbb{R}_0^+ \to \mathbb{R}^+$ in (1.2). In particular, the tested functions are

$$\phi_1(t) = \begin{cases} 2-t & \text{if } 0 \le t \le 1 \\ \frac{1}{t} & \text{if } t \ge 1 \end{cases}$$

$$\phi_2(t) = \begin{cases} \frac{3-2e}{e-1}t+2 & \text{if } 0 \le t \le 1 \\ \frac{1}{e^t-1} & \text{if } t \ge 1 \end{cases}$$

$$\phi_3(t) = \begin{cases} (\log 2-2)t+2 & \text{if } 0 \le t \le 1 \\ \frac{1}{\log(t+1)} & \text{if } t \ge 1 \end{cases}$$

$$\phi_4(t) = \begin{cases} 2-t & \text{if } 0 \le t \le 1 \\ \frac{1}{t^{13/10}} & \text{if } t \ge 1 \end{cases}$$

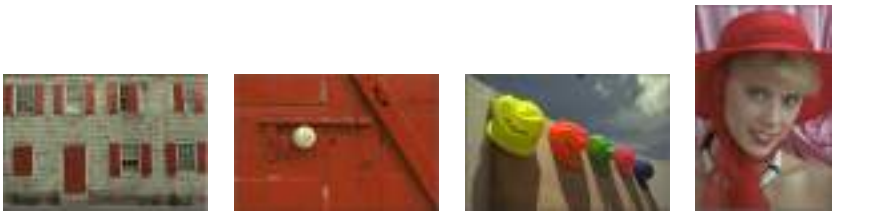
$$\phi_5(t) = \begin{cases} 2-t & \text{if } 0 \le t \le 1 \\ \frac{1}{t^{7/5}} & \text{if } t \ge 1 \end{cases}$$

$$\phi_6(t) = \begin{cases} 2-t & \text{if } 0 \le t \le 1 \\ \frac{1}{t^{3/2}} & \text{if } t \ge 1 \end{cases}$$

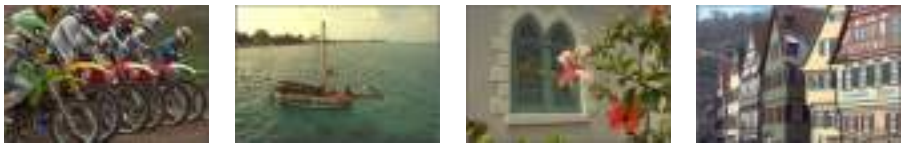
Observe that the detection functions ϕ_j, $j = 1,\ldots,6$ are decreasing and continuous. Moreover, we get

$$\phi_j(0) = 2 \text{ and } \lim_{t\to+\infty} \phi_j(t) = 0$$

In Table 1.1 there are the errors of the LEP algorithm in terms of *Mean Square Error* (MSE, see also [94]) in reconstructing the images of the Kodak set as the detection function varies. The values in bold are related to the best reconstruction of a specific image. In the last line there are the means of the MSE obtained in the reconstruction of the Kodak sample images, as the detection function varies. Note that the best result can be obtained by different detection functions, but, if one takes the means, the minimal error corresponds to the detection function ϕ_4. To evaluate whether the function ϕ_4 is actually the best detection function, we proceed as follows. For each sample image we give five points to the detection function which allows to obtain an estimate with the minimal error; four points to the detection function which obtain the second best minimal error; three points in correspondence with the third minimal error, and so on. In Table 1.2 there are the results obtained by the all detection functions on the single images,



(a) Image 01. (b) Image 02. (c) Image 03. (d) Image 04.



(e) Image 05. (f) Image 06. (g) Image 07. (h) Image 08.

(i) Image 09. (j) Image 10. (k) Image 11. (l) Image 12.

(m) Image 13. (n) Image 14. (o) Image 15. (p) Image 16.

(q) Image 17. (r) Image 18. (s) Image 19. (t) Image 20.

(u) Image 21. (v) Image 22. (w) Image 23.

Figure 1.5: Kodak image set.

and in the last line there is the global score. Observe that, even in this case, the highest score is obtained by the detection function ϕ_4.

From now on, we use the detection fuction ϕ_4 in the LEP algorithm. In Figure 1.6 (a) the reconstruction of Image 02 is shown. If we evaluate the results visually, it is very difficult to perceive the errors present during the reconstruction. Thus, in Figure 1.6 (b) we present the image of errors, which is given by

$$\check{\mathbf{x}} - \mathbf{x} + 128\,\mathbf{e}$$

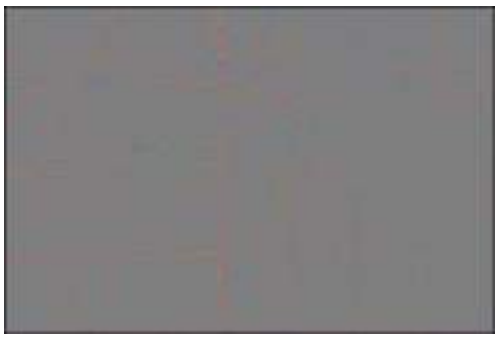
where $\check{\mathbf{x}}$ is the estimate obtained by the LEP algorithm, $\mathbf{x}$ is the ideal image and $\mathbf{e}$ is the column vector belonging to $\mathbb{R}^{n \cdot m}$, whose entries are equal to one. Again, it is difficult to note visually the errors of the algorithm. So, in Figure 1.6 (c) we show the image of the enlarged errors, that is

$$5\,(\check{\mathbf{x}} - \mathbf{x}) + 128\,\mathbf{e}$$

where it is possible to see in detail the errors of the algorithm.

(a) LEP result.

(b) LEP error image.



(c) Enlarged LEP error image.

Figure 1.6: LEP reconstruction of Figure 02.

Since most algorithms existing in the literature do not allow to see easily the errors related to

the reconstructions, because they seem to be perfect, then, to compare our algorithm with some of those proposed in the literature, we use the table of the errors in the reconstruction of the Kodak dataset. In Tables 3 and 4, we compare the LEP method with the original Freeman filter (see [65]) and with some other recently published algorithms (see also [40, 64, 78, 85, 90, 107, 112, 116, 128, 137, 144, 166]). Although the proposed algorithm gives the best reconstruction of two images, the total mean of the errors obtained with the LEP algorithm is the smallest of the selected methods.

In the literature there exist several other algorithms, for instance the one proposed in [70], which is one of the best performed algorithms, obtaining a MSE mean equal to 6.11. However, in order to reach this goal, the needed mean computation time is equal to 27 minutes and 4 seconds, while the mean computation time for the LEP algorithm is equal to 0.16 seconds. The aim of the LEP algorithm is to obtain good results in a very short period of time. This method can be used as an initialization algorithm for the technique proposed in [70], obtaining meaningful reductions of its computational cost.

In Figure 1.7 (a) the reconstruction of Image 08 by LEP is presented. Its MSE, between the original Image 08 is about 19.99, obtained in a computational time of 0.16 seconds. The relative enlarged error image is presented in In Figure 1.7 (b). In Figure 1.7 (c) the reconstruction of Image 08 by the algorithm proposend in [70] is illustrated. Its MSE between the original Image 08 is about 12.33, obtained in a computational time of 27 minutes and 54.74 seconds. The relative enlarged error image is given in In Figure 1.7 (d). From the enlarged error images it is possible to notice how the algorithm proposed in [70] specially refines the reconstruction of the buildings on the left part of the image, however it does not allow an immediate processing of the image.

(a) LEP result.

(b) Enlarged LEP error image.

(c) Result of the algorithm proposed in [70].

(d) Enlarged error image of the algorithm proposed in [70].

Figure 1.7: Reconstruction of Image 08.

Image	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6
01	10.02	16.86	13.68	9.77	9.75	**9.75**
02	6.96	8.66	7.63	6.90	**6.91**	6.92
03	4.22	6.01	4.95	4.14	4.13	**4.12**
04	**6.16**	9.15	6.66	6.17	6.18	6.21
05	13.41	22.46	15.05	**13.36**	13.39	13.44
06	9.29	13.12	11.76	9.10	9.09	**9.08**
07	5.05	8.10	5.77	**5.04**	5.05	5.07
08	20.63	26.39	27.39	19.99	19.87	**19.83**
09	**4.60**	7.28	5.68	4.63	4.66	4.69
10	4.82	6.94	5.82	**4.79**	4.80	4.81
11	7.79	11.17	8.99	7.70	**7.70**	7.71
12	3.88	5.81	5.41	**3.83**	3.84	3.85
13	**19.05**	29.38	20.47	19.11	19.19	19.27
14	15.96	22.12	17.59	15.79	**15.77**	15.77
15	8.77	11.77	10.29	**8.68**	8.69	8.71
16	4.26	5.36	5.32	4.11	4.08	**4.06**
17	**5.50**	7.83	6.00	5.52	5.54	5.57
18	**14.88**	21.30	15.36	15.00	15.04	15.09
19	8.68	11.36	11.24	8.48	8.47	**8.46**
20	6.71	8.24	8.75	6.43	6.39	**6.36**
21	8.14	12.32	9.41	**8.06**	8.07	8.08
22	12.12	16.09	12.93	**12.09**	12.11	12.13
23	**3.82**	6.17	4.02	3.85	3.87	3.89
mean	8.9002	12.7781	10.4428	**8.8074**	8.8080	8.8203

riptsize

Table 1.1: MSE of the LEP reconstructions of the Kodak set as the detection function varies.

Image	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6
01	2	0	1	3	4	5
02	2	0	1	4	5	3
03	2	0	1	3	4	5
04	5	0	1	4	3	2
05	3	0	1	5	4	2
06	2	0	1	3	4	5
07	4	0	1	5	3	2
08	2	1	0	4	3	5
09	5	0	1	4	3	2
10	2	0	1	5	4	3
11	2	0	1	4	5	3
12	2	0	1	5	4	3
13	5	0	1	4	3	2
14	2	0	1	3	5	4
15	2	0	1	5	4	3
16	2	0	1	3	4	5
17	5	0	1	4	3	2
18	5	0	1	4	3	2
19	2	0	1	3	4	5
20	2	1	0	3	4	5
21	2	0	1	5	4	3
22	3	0	1	5	4	2
23	5	0	1	4	3	2
total	68	2	21	**92**	87	75

riptsize

Table 1.2: Points of the LEP reconstructions of the Kodak set as the detection function varies.

Image	[128]	[112]	[144]	[85]	[137]	[166]	[116]
01	22.24	**9.57**	155.63	28.45	27.04	14.90	16.64
02	7.78	6.58	32.00	8.19	8.42	7.02	6.97
03	4.96	4.96	23.34	5.80	5.32	4.33	5.17
04	9.06	7.21	29.31	9.19	7.41	7.25	7.06
05	19.41	14.62	145.24	21.58	19.23	12.48	15.78
06	10.62	**8.04**	111.45	21.58	20.80	8.45	13.15
07	4.72	4.60	29.86	5.36	5.74	4.62	5.13
08	51.18	**16.83**	297.91	40.65	57.69	23.18	30.14
09	4.85	4.16	39.54	6.28	7.21	4.11	5.50
10	5.94	4.30	37.59	6.75	6.07	4.35	5.01
11	11.78	8.34	82.62	16.45	15.46	9.04	10.09
12	3.83	3.77	30.63	5.76	6.47	3.57	5.01
13	50.71	20.99	271.69	70.48	47.87	33.74	28.12
14	17.99	22.97	80.92	18.62	18.62	18.58	18.24
15	10.87	8.24	32.29	11.27	8.30	8.32	8.02
16	4.28	4.50	50.13	9.55	10.52	**3.47**	6.30
17	8.07	5.20	42.96	9.66	7.71	5.81	5.88
18	19.28	**12.19**	96.18	23.72	17.14	15.07	12.94
19	10.16	6.56	106.93	12.19	19.23	7.23	11.86
20	8.09	5.79	45.93	9.23	8.77	6.43	6.37
21	15.53	8.32	94.42	19.96	17.42	11.94	11.25
22	14.59	11.12	62.96	15.74	14.73	12.80	12.74
23	4.78	4.31	21.38	4.40	4.42	3.96	4.48
mean	13.9441	8.8330	83.5177	16.5586	15.7222	10.0269	10.9497

riptsize

Table 1.3: MSE of the reconstructions of the Kodak set by the algorithms in [85, 112, 116, 128, 137, 144, 166]

riptsize

Image	[126]	[90]	[40]	[85]	[64]	[107]	[78]	LEP
01	10.77	19.77	35.65	53.34	17.74	15.31	11.04	9.77
02	8.96	7.57	36.48	11.69	14.69	**6.14**	7.81	6.91
03	12.16	4.58	37.25	8.57	12.25	**3.52**	4.64	4.13
04	5.11	8.43	36.74	10.04	13.78	**4.93**	6.59	6.17
05	**10.52**	17.50	35.49	39.02	18.54	11.94	11.49	13.36
06	8.77	11.43	36.40	37.33	14.93	8.77	10.69	9.10
07	4.51	5.32	37.08	10.05	12.77	**3.61**	4.54	5.04
08	22.81	27.11	34.60	96.56	22.60	22.80	19.99	19.99
09	7.74	5.05	37.42	13.06	11.67	**4.05**	4.30	4.63
10	**3.86**	5.45	37.25	12.29	12.22	4.05	4.34	4.79
11	7.78	11.62	36.40	24.57	14.86	8.26	**7.59**	7.70
12	**2.91**	4.29	37.59	12.90	11.41	3.36	4.80	3.83
13	24.78	47.00	33.66	77.41	27.87	35.90	24.38	**19.11**
14	15.35	18.33	35.08	26.53	20.23	**11.33**	16.42	15.79
15	**7.64**	10.26	36.23	16.74	15.53	8.38	8.72	8.68
16	3.53	4.74	37.50	16.61	11.48	3.59	4.23	4.11
17	4.99	7.73	41.12	13.28	5.06	5.31	**4.90**	5.52
18	12.83	19.64	35.98	34.73	16.41	16.79	13.24	15.00
19	6.28	9.31	40.19	34.55	**6.21**	7.20	6.83	8.48
20	**5.51**	7.76	32.52	22.11	3.67	6.10	5.80	6.43
21	9.23	14.36	36.48	29.09	14.66	10.74	8.37	**8.06**
22	**9.40**	14.69	37.33	21.57	12.05	9.42	10.33	12.09
23	8.67	4.22	39.45	6.97	7.38	**3.06**	4.07	3.85
mean	9.13	12.4412	36.6901	26.9135	15.2602	9.32	8.9175	**8.8074**

Table 1.4: MSE of the reconstructions of the Kodak set by the algorithms in [40, 64, 78, 85, 90, 107, 126] and by the LEP method.

Chapter 2

An edge-preserving regularization model for the demosaicing of noisy color images

This chapter is organized as follows. In Section 2.1, the state-of-the-art in the field of color image demosaicing is surveyed, the problem is formulated, and the principles of the regularization approach are stated. Section 2.2 is devoted to problem formulation, the principle of the regularization approach are stated, and the specific edge-preserving regularization strategy we adopt is described. In Section 2.3, the solution algorithm is described in detail. Section 2.4 is devoted to the quantitative comparison between the results obtained with our method and those of some of the most performing algorithms proposed in the recent literature, using both the Kodak 24-image dataset [109] and the McMaster 18-image dataset [169] as benchmark sets, and with specific reference to the Bayer CFA. Finally, in Sections 2.6–2.8, some mathematical aspects are developed in detail.

2.1 Related work

A major problem of demosaicing is to avoid oversmoothing of the edges, so that the fundamental feature of any method is its ability to perform interpolation along and not across the edges. Some

methods perform directional interpolation after locating the image discontinuities through edge-detectors [85], or analyzing the variance of the color differences [47]. For example, the work in [12] proposes high-order interpolation and Sobel operators to compute the gradients, and in [64] a level set method is used to minimize an energy function that gives the direction of the edges. In [48] the interpolation direction is chosen by exploiting an edge-sensing parameter called integrated gradient, which simultaneously takes into account for both color intensity and color difference. As an alternative to color difference interpolation, the algorithm in [107] is based on interpolation in a residual domain. The residuals are differences between observed and tentatively estimated pixel values which minimize a Laplacian energy.

Some authors exploit known properties of the human visual system, to design linear, adaptive filters applied to the luminance component of the sampled color values (see [5, 113]), or to estimate every missing sample as a weighted sum of its neighboring pixels, assuming that the hue of an image does not change abruptly [116]. In this latter algorithm, in order to reduce interpolation across boundaries, the weights are calculated on the basis of an edge-sensing mechanism. In [93] the model above is enhanced by making the weights depend not only on the colors of the pixels but also on their location within the neighborhood.

In other methods, the best reconstruction of the missing data, first estimated by interpolating along horizontal and vertical directions, is chosen ([90, 120, 166]), or the two reconstructions are fused ([121, 168]). In particular, [90] proposes an algorithm based on the Laplacian filter, by selecting the interpolation directions having the least level of color artifacts. In [169], multiple local directional estimates of a missing color sample are computed and fused according to local gradients. Then, the image non-local redundancy is exploited to improve the local color reproduction. This allows the final reconstruction to be performed at the structural level as opposed to the pixel level. On this line, the method in [40] infers the missing colors by taking into account the local image geometry through the image self-similarity.

In [76, 78, 112] the strong correlation existing between the high frequencies of the three color components is directly exploited. In particular, [76] proposes reconstructing the high frequencies of the red and blue channels by replacing them with those of the green channel, which has the highest sampling rate. Based on a similar principle, the algorithm presented in [78] forces similarity between the high frequencies of the red and the green, and of the blue and the green.

This algorithm is called *Alternating Projections* (AP) algorithm, since it alternately projects the estimated image into an observation constraint set (faithfulness with the data) and a detail constraint set (similarity of the high–frequency components), until a fixed point is found. A faster version of the AP algorithm appeared in [114].

The sparse nature of color images has also been exploited for demosaicing. A suitable dictionary is designed and applied with the iterative K-SVD algorithm (specificare a parole che cosa vuol dire SVD) in [117], whereas in [126] the dictionary is constructed on the basis of a clear distinction between the inter-channel and intra-channel correlations of natural images, and the sparse representation of the image is found through compressed sensing. In [4] a locally adaptive approach is used for demosaicing dual-tree complex wavelet coefficients.

Regularization approaches to demosaicing have also been explored. The total-variation principle is used in [143], while [123] proposes first a general quadratic smoothness regularizer and then an adaptive filter, in order to improve the reconstruction near the edges of the first estimate.

The methods surveyed above have been mainly designed for noise-free data. An abundant literature has also been devoted to joint demosaicing and denoising, which is the problem considered in this work. Indeed, while the availability of noiseless data is an unrealistic assumption, performing denoising as a pre- or a post-processing has significant drawbacks. In the first case, denoising must be separately performed on the individual channels, and hence the full image resolution cannot be exploited. On the other hand, noisy images make more complicate the edge detection step that is often preliminary to demosaicing. Furthermore, demosaicing alters the characteristics of the noise, thus making more complex the subsequent denoising process. In [170] demosaicing and denoising is performed in two steps. First, the full resolution green component is recovered from the difference signals of the color channels. These are estimated by a MMSE technique that exploits both spectral and spatial correlations to simultaneously suppress sensor noise and interpolation error. Second, the CFA channel-dependent noise is removed from the reconstructed green channel with a wavelet-based approach. Finally, also the red and blue channels are estimated and denoised. The method in [135], specifically designed for signal-dependent (e.g. Poissonian) noise, is based on local polynomial approximation and the intersection of confidence intervals. These concepts, simultaneously utilizing the three color channels, are exploited to design and adaptively select the length of directional filters, then used to denoise and interpolate the

samples via convolution. In [53] the luminance and chrominance channels of a noisy mosaiced image are first reconstructed, by exploiting a frequency analysis of the sampling pattern induced by the Bayer CFA. Wiener filters are then designed to denoise the chrominances, whereas the luminance is linearly filtered as a grayscale image. An extended variant of this approach is also proposed, in which the demoisaiced image is mosaiced again and then demosaiced using the method in [168].

One of the advantages of regularization is that it provides a natural framework to couple demosaicing with other typical problems in image reconstruction and restoration. For instance, in [63] a regularization method combines demosaicing and super-resolution, and in [167] demosaicing is augmented with color de-crosstalk. Regularization is thus an ideal setting to approach joint demosaicing and denoising. The work [91] presents an algorithm that uses a modified total least squared estimation technique, to estimate an ideal demosaicing filter able to deal with the noise affecting the base vectors. In [122], the authors evaluate the statistical characteristics of the noise resulting from the demosaicing process performed through the space-varying filters designed in their previous work [123]. Then, they design an ad hoc post-processing denoising strategy.

2.2 Formulation of the demosaicing problem and its regularization

2.2.1 The data generation model

A color image of size $n \times m$ can be represented as a vector $\mathbf{x} \in \mathbb{R}^{3nm}$, $\mathbf{x} = \left((\mathbf{x}^{(r)})^T \, (\mathbf{x}^{(g)})^T \, (\mathbf{x}^{(b)})^T \right)^T$, where $\mathbf{x}^{(r)}, \mathbf{x}^{(g)}, \mathbf{x}^{(b)} \in \mathbb{R}^{nm}$ are the red, green and blue channels expressed in the lexicographic notation, respectively. The mosaicing problem is formulated as

$$\boldsymbol{y} = M \, (\mathbf{x} + \boldsymbol{n}), \tag{2.1}$$

where $\mathbf{x} \in [0,255]^{3nm}$, $\boldsymbol{y} \in [0,255]^{3nm}$, and $\boldsymbol{n} \in [0,255]^{3nm}$ denote the ideal color image, the mosaiced image, and the additive noise, respectively. We assume the noise to be independent, Gaussian, with null mean and variance σ^2. The matrix M, $M \in \{0,1\}^{3nm \times 3nm}$, is a linear operator

associated to the acquisition pattern, consisting of the following block diagonal matrix:

$$M = \begin{pmatrix} M^{(r)} & O & O \\ O & M^{(g)} & O \\ O & O & M^{(b)} \end{pmatrix}, \tag{2.2}$$

where $O \in \mathbb{R}^{nm \times nm}$ is the null matrix, and $M^{(r)}$, $M^{(g)}$, $M^{(b)}$ are diagonal matrices in $\mathbb{R}^{nm \times nm}$. For the Bayer pattern, the diagonal elements of these matrices are given by

$$m^{(r)}_{(i,j),(i,j)} = \begin{cases} 1, & i \equiv_2 j \equiv_2 0, \\ 0, & \text{otherwise}, \end{cases}$$

$$m^{(g)}_{(i,j),(i,j)} = \begin{cases} 1, & i \not\equiv_2 j, \\ 0, & \text{otherwise}, \end{cases} \tag{2.3}$$

$$m^{(b)}_{(i,j),(i,j)} = \begin{cases} 1, & i \equiv_2 j \equiv_2 1, \\ 0, & \text{otherwise}, \end{cases}$$

where (i, j) is the generic pixel index. The demosaicing problem is the inverse problem associ-

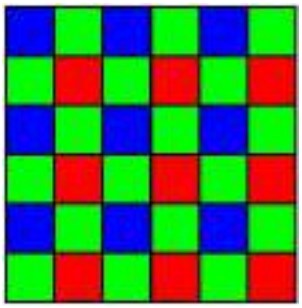


Figure 2.1: The Bayer pattern

ated to the direct problem formulated in (2.1), and consists of finding an estimate $\widetilde{\mathbf{x}}$ of the ideal image, given the mosaiced image $\boldsymbol{y}$ and the operator M. Since M is singular, the demosaicing problem is ill-posed in the Hadamard sense (see [83, 84]), because in general it does not admit a unique solution. Given $\boldsymbol{y}$, there are infinitely many feasible solutions, since at each pixel the values of the two unmeasured channels do not contribute to the data. Therefore, regularization techniques are necessary to reduce the number of solutions.

2.2.2 The regularization model

We define our regularized solution $\widetilde{\mathbf{x}}$ as an argument of the minimum of the following energy function:

$$E(\boldsymbol{x}) = \|M(\boldsymbol{x}-\boldsymbol{y})\|_2^2 + \sum_{k=1}^{3}\sum_{c\in C_k} \varphi\left(N_c^k\boldsymbol{x}, N_{p_k(c)}^k\boldsymbol{x}\right) + \sum_{k=1}^{3}\sum_{c\in C_k} \varphi\left(V_c^k\boldsymbol{x}, V_{p_k(c)}^k\boldsymbol{x}\right), \tag{2.4}$$

where $\|\cdot\|_2$ denotes the Euclidean norm, and the first term of the right hand of (2.4) expresses a data fidelity constraint, which is identically null in the noiseless case.

The second term in the right hand of (2.4) regulates the intra-channel smoothness of the involved image. The third term imposes a correlation between the different channels, i.e. an inter-channel smoothness. Intra-channel and inter-channel smoothness are measured through the operators N_c^k and V_c^k, respectively, and φ is a stabilizer that weights the degree of smoothness required and relaxes it when a discontinuity is expected.

Let us start by analyzing the form of the operator N_c^k, given by

$$N_c^k\boldsymbol{x} = \left\| \left(D_c^k\boldsymbol{x}^{(r)}, D_c^k\boldsymbol{x}^{(g)}, D_c^k\boldsymbol{x}^{(b)} \right) \right\|_2, \tag{2.5}$$

where D_c^k is a finite difference operator of order k applied to a suitable set c of adjacent pixels, called *clique* of order k. Therefore, from (2.5) it appears that N_c^k is the norm of the vector of the finite differences of the intensities of the red, green and blue channel computed on the clique c of order k. All cliques of order k are collected in the set C_k. Each of such cliques is uniquely associated with a discontinuity, of order k as well, which, in turn, is labeled by a hidden line element.

To reconstruct the finest details in the images, we consider finite differences, and then discontinuities, of first, second and third order, that is $k = 1,2,3$. The geometry of the associated cliques is described in Section 2.5.

The edges of the first order separate homogeneous patches in the image, the edges of the second order mark the slope of linearly varying areas, and the edges of the third order are associated with the intensity discontinuities in regions where intensity varies quadratically.

As the inter-channel correlation has the aim to maintain the clue of the objects in the image, the finite difference operators should have the same behavior in all three channels. So we define the operator V_c^k as follows:

$$V_c^k \boldsymbol{x} = \left\| \left(D_c^k \boldsymbol{x}^{(r)} - D_c^k \boldsymbol{x}^{(g)}, D_c^k \boldsymbol{x}^{(r)} - D_c^k \boldsymbol{x}^{(b)}, D_c^k \boldsymbol{x}^{(g)} - D_c^k \boldsymbol{x}^{(b)} \right) \right\|_2, \tag{2.6}$$

which is the norm of the vector of the inter-channel differences of the intra-channel k-order derivatives. Again, a hidden line variable is implicitly associated with the clique c for each order $k = 1,2,3$. These further sets of hidden line variables mark the discontinuities between areas having homogeneous clues.

In (2.4), N_c^k and V_c^k are weighted by suitable stabilizers. These stabilizers should regulate the degree of smoothness required in the two cases, and relax it when discontinuities are expected, and also in dependence of their amplitude. In (2.4) we adopted the same parametric stabilizer φ for both the operators N_c^k and V_c^k, and let its parameters possibly vary in the two terms (see also [33]).

To have a more accurate reconstruction, it is important that edges are not thick, or, equivalently, that the object contours are not blurred. To this aim it is advisable to inhibit the creation of discontinuities at two adjacent cliques. Specifically, to prevent double edges of order k, simultaneous discontinuities at the cliques c and at the previous one $p_k(c)$ should be inhibited (see Section 2.5 for the definition of adjacent cliques).

When $p_k(c)$ is not defined, that is for the mixed cliques and for the cliques on the border of the image, we automatically assume that the adjacent discontinuity is null.

Having in mind the above described properties to be featured by the stabilizer, we define a bivariate function $\varphi : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, having the following form:

$$\varphi(t_1,t_2) = \begin{cases} g_1(t_1), & \text{if } |t_2| \le s, \\ \left(1 - \dfrac{2(|t_2|-s)^2}{(\zeta-s)^2}\right) g_1(t_1) + \dfrac{2(|t_2|-s)^2}{(\zeta-s)^2}\, g_2(t_1), & \text{if } s < |t_2| \le \dfrac{\zeta+s}{2}, \\ \dfrac{2(|t_2|-\zeta)^2}{(\zeta-s)^2}\, f_1(t_1) + \left(1 - \dfrac{2(|t_2|-\zeta)^2}{(\zeta-s)^2}\right) f_2(t_1), & \text{if } \dfrac{\zeta+s}{2} < |t_2| < \zeta, \\ g_2(t_1), & \text{if } |t_2| \ge \zeta, \end{cases} \tag{2.7}$$

where $\zeta - s$ is a positive and sufficiently small quantity, and for $i = 1,2$ it is

$$g_i(t_1) = \begin{cases} \lambda^2 t_1^2, & \text{if } |t_1| < q_i, \\ \alpha_i - \dfrac{\tau}{2}(|t_1| - r_i)^2, & \text{if } q_i \le |t_1| \le r_i, \\ \alpha_i, & \text{if } |t_1| > r_i, \end{cases} \tag{2.8}$$

$$\alpha_i = \begin{cases} \alpha, & \text{if } i = 1, \\ \alpha + \varepsilon, & \text{if } i = 2, \end{cases} \tag{2.9}$$

$$q_i = \frac{\sqrt{\alpha_i}}{\lambda^2}\left(\frac{2}{\tau} + \frac{1}{\lambda^2}\right)^{-1/2}, \tag{2.10}$$

τ is a large enough real constant, and

$$r_i = \frac{\alpha_i}{\lambda^2 q_i}, \quad i = 1,2. \tag{2.11}$$

In general, the analytical form of the stabilizer determines the amplitude of the discontinuities in the reconstructed image, by promoting on-off discontinuities of large amplitude above a given threshold, or more slowly varying discontinuities of graded amplitudes. In the primal-dual formalism, the stabilizers of the first type are said to implicitly address "hard", Boolean line elements, while the second type addresses hidden "soft" line elements, ideally valued in $[0,1]$.

We recall that the functions g_i, $i = 1,2$, defined in (2.8), are approximations of class C^1 of the classical truncated parabola defined in [24] (see also [25, 33, 34]) that, when used as a stabilizer, implicitly addresses a Boolean line process. In the bivariate case, the function φ defined in (2.7) possesses the same characteristic when τ tends to $+\infty$ and ζ is very close to s. The actual form we propose is an approximation of such a function, with the property of being of class C^1, which is essential for the convergence of the minimization algorithm.

We observe that the function φ defined in (2.7), as a function of two variables, is not convex, and hence neither is the energy function $E(\mathbf{x})$, defined in (2.4) as a function of $3nm$ variables. Thus, to minimize E, we determine a finite family of approximating functions $\{E^{(p)}\}_p$, where $E^{(0)}$ is componentwise convex, and $E^{(2)}$ is the original energy function E. The initial point

to minimize the componentwise convex approximation is found by means of the *Local Edge Preserving* (LEP) algorithm presented in Chapter 1. The LEP is a very fast algorithm, consisting of two phases. In the first phase, the missing components are determined by a weighted mean, which guarantees to preserve the edges. In the second phase, the differences between the colors of the channels are imposed to be constant within homogeneous areas. Our algorithm is called *Graduated Componentwise Non-Convexity* (GCNC), and can be summarized as follows:

`initialize` $\boldsymbol{x}$ `by LEP; set` $p = 0$`;`

`repeat`

- `find the minimum of the function` $E^{(p)}$
  `starting from the initial point` $\boldsymbol{x}$`;`
- `set` $\boldsymbol{x}$ `to the reached minimizer;`
- `update the parameter` p`; until` $p = 2$

Note that our algorithm can be seen as a variant of the GNC algorithm (see, e.g., [19, 34, 127, 129, 130, 131, 140]).

To construct the first componentwise convex approximation $E^{(0)}$, it is sufficient to find a componentwise convex approximation for the stabilizers in (2.7), since the data term in (2.4) is globally convex. Such componentwise convex approximations can be constructed on the basis of a componentwise convex approximation of the bivariate function φ, as shown in Section 2.8.

To do this, we proceed as follows. First of all, we approximate the functions $g_i(t_1)$ with the following convex approximations given by

$$\overline{g_i}(t_1) = \begin{cases} \lambda^2 t_1^2, & \text{if } |t_1| \le q_i, \\ \lambda^2 (2q_i|t_1| - q_i^2), & \text{if } |t_1| \ge q_i, \qquad i = 1,2 \end{cases} \tag{2.12}$$

(see also [129]). Moreover, we find an approximation, convex with respect to the variable t_2, of the function φ defined in (2.7), in tthe following way:

$$\overline{\varphi}(t_1,t_2) = \frac{\bar{t}^2 - t_2^2}{\bar{t}^2}\,\overline{g_1}(t_1) + \frac{t_2^2}{\bar{t}^2}\,\overline{g_2}(t_1) = \overline{g_1}(t_1) + \frac{t_2^2}{\bar{t}^2}\,(\overline{g_2}(t_1) - \overline{g_1}(t_1)), \tag{2.13}$$

where $\bar{t}$ is the maximum value that a finite difference operator can assume ($\bar{t} = 2^k \cdot \sqrt{2} \cdot 255$, $k = 1,2,3$), for light intensity of the images in the range [0,255]. It is not difficult to check that $g_2 - g_1$ is convex, since $0 < q_1 < q_2$.

We recall that $t_1 = N_c^k \mathbf{x}$, $t_2 = N_{p_k(c)}^k \mathbf{x}$ in the second term of the right hand of (2.4), and $t_1 = V_c^k \mathbf{x}$, $t_2 = V_{p_k(c)}^k \mathbf{x}$ in the third term of the right hand of (2.4). Let us fix $k \in \{1,2,3\}$ and $c \in C_k$, and choose $\Xi_c^k \in \{N_c^k, V_c^k\}$. So, t_1 is a function of $\mathbf{x}$, and $t_1(\mathbf{x}) = \Xi_c^k(\mathbf{x})$. In particular, t_1 depends only on the variables involved in the clique c. Analogously, $t_2(\mathbf{x}) = \Xi_{p_k(c)}^k(\mathbf{x})$ depends only on the variables involved in the clique $p_k(c)$. Note that the function $\overline{\varphi}$ defined in (2.13) is componentwise convex, but the function $\overline{\Phi}(\mathbf{x}) = \overline{\varphi}(t_1(\mathbf{x}), t_2(\mathbf{x}))$ is not componentwise convex with respect to the components of the vector $\mathbf{x} \in \mathbb{R}^{3nm}$. This is due to the fact that $c \cap p_k(c) \neq \emptyset$. However, if we choose $t_2(\mathbf{x}) = \Xi_{\pi_k(c)}^k(\mathbf{x})$ instead of $\Xi_{p_k(c)}^k(\mathbf{x})$, then the function t_2 is componentwise convex with respect to $\mathbf{x}$, as shown in Section 2.8, since $c \cap \pi_k(c) = \emptyset$. Thus, in order to define the family of the approximations for the algorithm GCNC, we proceed as follows.

If $p \in [0,1]$, put

$$\begin{aligned} E^{(p)}(\boldsymbol{x}) &= \|M(\boldsymbol{x}-\boldsymbol{y})\|_2^2 + \sum_{k=1}^{3} \sum_{c \in C_k} \overline{\varphi}\left(N_c^k \boldsymbol{x}, p N_{p_k(c)}^k \boldsymbol{x} + (1-p) N_{\pi_k(c)}^k \boldsymbol{x}\right) + \\ &+ \sum_{k=1}^{3} \sum_{c \in C_k} \overline{\varphi}\left(V_c^k \boldsymbol{x}, p V_{p_k(c)}^k \boldsymbol{x} + (1-p) V_{\pi_k(c)}^k \boldsymbol{x}\right). \end{aligned} \tag{2.14}$$

When $p \in [1,2]$, set

$$E^{(p)}(\boldsymbol{x}) = \|M(\boldsymbol{x}-\boldsymbol{y})\|_2^2 + \sum_{k=1}^{3} \sum_{c \in C_k} \varphi^{(p)}\left(N_c^k \boldsymbol{x}, N_{p_k(c)}^k \boldsymbol{x}\right) + \sum_{k=1}^{3} \sum_{c \in C_k} \varphi^{(p)}\left(V_c^k \boldsymbol{x}, V_{p_k(c)}^k \boldsymbol{x}\right), \tag{2.15}$$

where

$$\varphi^{(p)}(t_1, t_2) = (2-p)\,\overline{\varphi}(t_1, t_2) + (p-1)\,\varphi(t_1, t_2). \tag{2.16}$$

In Section we will prove that, for each $p \in [1,2]$, $\varphi^{(p)}$ satisfies the duality conditions that guarantee the edge-preserving properties and the inhibition of double edges (see [33]). Note that, for $p \in [0,1]$, the stabilizer $\varphi(t_1, t_2)$ is equal to $\overline{\varphi}(t_1, t_2)$, and hence fulfils the same properties. Furthermore, in [27] it is proved that the associated line process is non-Boolean. In fact, the hidden line elements tend to become Boolean as far as p tends to 2. However, we experimentally observed that, in real images, graded discontinuities can be useful to prevent the aliasing effect. Thus, in the experiments, we stop the minimization algorithm at a suitable value of p different from 2, as explained in Section 2.4.

2.3 The NL-SOR algorithm

To minimize each approssimation $E^{(p)}$, we use a *Non-Linear Successive Over Relaxation* (NL-SOR) algorithm, which is widely used in the literature (see also [19, 24, 38]). The NL-SOR algorithm is defined as follows:

$l = 1$;

while $\|\nabla E^{(p)}(\mathbf{x})\| > \varepsilon$

for $i = 1, 2, \ldots, nm$

for $e = r, g, b$

$$\left(x_i^{(e)}\right)^{(l+1)} = \left(x_i^{(e)}\right)^{(l)} - \frac{\omega}{T} \frac{\partial E^{(p)}\left(\mathbf{x}^{(l)}\right)}{\partial x_i^{(e)}};$$

end for

end for

$l = l + 1$;

end while

where $\varepsilon > 0$ is a fixed threshold, $\omega > 0$ is the accelerator parameter,

$$T > \max_{i=1,2,\ldots,nm, e=r,g,b} \max_{\mathbf{x}} \left\{ \frac{\partial_+^2 E^{(p)}(\mathbf{x})}{(\partial x_i^{(e)})^2}, \frac{\partial_-^2 E^{(p)}(\mathbf{x})}{(\partial x_i^{(e)})^2} \right\},$$

and the symbols ∂_+^2 and ∂_-^2 denote the right and left second partial derivatives, respectively. In [38, Theorem 2] the convergence of the algorithm is proved when $E^{(p)}$ is strictly convex and of class C^2. However, such a theorem cannot be applied to our setting, since our first approximation is componentwise convex and C^1, but neither strictly convex nor C^2. Thus, in SectionNLSOR we propose an extension of [38, Theorem 2], in order to prove that in our case, when $p = 0$, the algorithm stops in correspondence with a stationary point.

2.4 Experimental results

The algorithm proposed in this work has been tested for the Bayer CFA on the set of 24 Kodak sample images (see [109]) of size 512×768, shown in Figure 2.2. This dataset represents the typical benchmark images used in the literature to compare the different demosaicing algorithms. These high quality images have been acquired as raw images, in such a way to minimize the effects of compression. We will refer to these images as ImageK, $1, 2, \ldots, 24$, listed from top to

bottom and from left to right in Figure 2.2. More recently, another dataset of images has started

Figure 2.2: Set of Kodak images (from: http://r0k.us/graphics/kodak)

to be used as a benchmark, namely, the McMaster, or IMAX, dataset first proposed in [169]. Indeed, it has been noticed that the statistics of the Kodak images are very different from those of other natural images, since they have very high spectral correlation, are smooth in the chromatic gradient and have low saturation. Conversely, the McMaster images are more saturated and have many sharp structures with abrupt color transitions. To generate this dataset, 8 high resolution color images of size 2310×1814 were originally captured by Kodak films, digitized, and then cropped in 18 sub-images, each of which of size 500×500. We will refer to these images as ImageM, $1, 2, \ldots, 18$, listed from top to bottom and from left to right in Figure 2.3.

Many demosaicing methods assume that the color differences change smoothly. As this assumption holds for the Kodak images, they are ideal to test these algorithms. Conversely, in [169], it has been shown that this assumption may not hold for the images in the McMaster dataset. Thus, the McMaster images are, in some sense, more suitable to test our algorithm, since, as said, it assumes that the changes in the color differences are only locally smooth. This conjecture, in fact, seems to be reflected in the experimental results below.

The free parameters appearing in the energy function have been calibrated on the images of the Kodak dataset, and then used for the images of the McMaster dataset as well. We employed a trial-and-error strategy to look for the set of parameters that give the best average Color Peak

Figure 2.3: Set of McMaster images (from http://www4.comp.polyu.edu.hk/∼cslzhang/CDM_Dataset.htm)

Signal-to-Noise Ratio (CPSNR) on all 24 images. The CPSNR quality index is defined as

$$CPSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right), \tag{2.17}$$

where MSE is the mean square error between the original image and the demosaiced one that, for color images, is defined as the arithmetic average of the mean square errors on the three channels.

The algorithm has been applied to both noiseless and noisy images.

In the noiseless case, and for the Kodak dataset, we compared the proposed method with some of the most popular and/or best performing methods in the literature, namely the algorithms proposed in [4, 12, 40, 48, 64, 78, 90, 107, 117, 123, 126]. For the noiseless McMaster dataset, a comparison has been made with the algorithms in [40, 78, 90, 107, 169], respectively. In particular, to collect the results of the algorithm in [78] we used the original MATLAB code provided by the authors, and for the results of the algorithm in [107] we used the source code available in the authors's web page.

In the noisy case, we compared our method with the algorithms in [78] and [91], by using the original MATLAB code provided by the authors, and with the algorithms in [53, 122, 170], by using the source codes available in the authors's web pages.

We chose to decrease p with a step of 0.01; for each sample image and for each value of p, we computed the RMSE between the ideal image and the minimizer of the approximated energy function, indicated as $\eta_j(p),\ j = 1,\ldots,24$. Then, the value $\bar{p}$ to which to stop the algorithm has

been determined based on the Kodak dataset as

$$\bar{p} = \arg\min_{p} \left\{ \sum_{j=1}^{24} \eta_j(p) \right\}. \tag{2.18}$$

2.4.1 Noiseless images

The best free parameters of the energy, used for the noiseless mosaiced images, are reported in Table 2.1.

Table 2.1: Parameters used for the noiseless case

	derivative order		
	$k=1$	$k=2$	$k=3$
λ^N	0.01	0.04	0.04
κ^N	30	8.66	7.07
λ^V	0.25	0.078	0.078
κ^V	3.46	6.19	6.19

For the given free parameters, we found $\bar{p} = 1.40$. For each test image, the reconstruction obtained with this value of p has then been taken as the optimal reconstruction provided by our algorithm.

The results obtained on the Kodak dataset are reported in Table 2.2, last column. As usual with the Kodak images, to compute the errors we removed the three pixels wide external frame. It is apparent that our method exhibit the highest CPSNR in more images than the other methods do (for each image, the highest CPSNR is highlighted in boldface).

The results obtained on the McMaster dataset are reported in Table 2.3, with a percentage of highest CPNR of 61%. These results seem to confirm that our method, for its edge-preserving property, is particularly suitable for images exhibiting sharp boundaries and fine details, as the McMaster images do.

Table 2.2: CPSNRs for noiseless Kodak images

ImageK	[78]	[90]	[117]	[123]	[64]	[40]	[48]	[4]	[12]	[126]	[107]	proposed
1	37.70	35.17	39.37	38.22	35.64	32.61	39.96	37.31	39.86	37.81	36.28	**40.66**
2	39.57	39.34	40.71	38.18	36.46	32.51	**40.99**	38.90	**40.99**	38.61	40.25	40.86
3	41.45	41.52	43.19	42.04	37.25	32.42	43.26	41.76	42.86	37.28	42.66	**43.31**
4	40.03	38.87	41.29	40.04	36.74	32.48	40.56	40.40	41.25	41.05	41.20	**41.84**
5	37.46	35.70	38.70	38.04	35.45	32.63	38.31	37.44	38.41	37.91	37.36	**38.94**
6	38.50	37.55	40.05	39.70	36.39	32.52	**41.00**	39.59	40.31	39.34	38.70	40.20
7	41.77	40.87	42.83	42.10	37.07	32.44	42.64	41.85	42.94	41.59	42.55	**43.62**
8	35.08	33.80	36.42	36.08	34.59	32.74	**37.35**	34.58	37.05	35.49	34.55	37.22
9	41.72	41.10	43.28	42.15	37.46	32.40	43.42	41.77	**43.44**	42.40	42.06	43.29
10	42.02	40.77	42.70	42.15	37.26	32.42	42.83	41.80	**43.12**	42.27	42.06	42.70
11	39.14	37.48	40.22	39.78	36.41	32.52	40.66	39.09	**40.92**	39.22	38.96	40.51
12	42.51	41.81	43.53	42.94	37.56	32.38	44.13	43.01	44.01	43.49	42.86	**44.40**
13	34.30	31.41	35.29	34.94	33.68	32.86	36.03	34.97	35.94	34.19	32.61	**36.24**
14	35.60	35.50	37.95	36.34	35.07	32.68	37.10	35.79	36.99	36.27	37.59	**38.26**
15	39.35	38.02	40.21	39.15	36.22	32.54	39.84	39.39	40.03	39.30	38.90	**40.35**
16	41.76	41.37	43.62	43.27	37.53	32.39	**44.47**	43.62	43.74	42.65	42.58	43.75
17	41.11	39.25	42.01	41.83	41.09	31.99	41.77	41.17	**42.24**	41.15	40.88	41.54
18	37.45	35.20	37.47	37.13	35.98	32.57	**37.96**	37.12	37.89	37.05	35.88	37.43
19	39.46	38.44	41.27	40.15	40.20	32.09	**41.79**	39.78	41.46	40.15	39.56	41.10
20	40.66	39.23	41.00	40.39	32.49	33.01	41.71	40.46	**41.85**	40.72	40.28	41.59
21	38.66	36.56	39.74	39.27	36.47	32.51	39.99	38.57	**40.37**	38.48	37.82	40.20
22	37.55	36.46	**38.87**	38.25	37.32	32.41	38.48	37.33	38.69	38.40	38.39	38.48
23	41.88	41.88	42.41	40.40	39.45	32.17	43.20	42.00	43.04	38.75	43.28	**43.89**
24	34.78	33.42	**35.63**	35.37	34.32	32.78	35.39	34.52	35.21	35.37	34.33	34.78

2.4.2 Noisy images

In a second set of experiments, we considered noisy images corrupted by independent, Gaussian noise, with zero mean and different values of the standard deviation σ. This time, the best free parameters for all 24 Kodak images, then used also for the 18 McMaster images, have been empirically found to be dependent on the noise standard deviation according to Table 2.4.

As done for the noiseless images, for each value of the noise variance, the suitable value $\bar{p}$ for stopping the algorithm has been determined according to the criterion in (2.18). The following empirical law that relates $\bar{p}$ to σ has also been found:

$$\bar{p}(\sigma) = \frac{3}{40}\sigma + \frac{4}{5}. \tag{2.19}$$

The CPSNR values computed for the case $\sigma = 16$ on the Kodak dataset are shown in Table 2.5.

Although the performance of our method is still satisfactory, this time the method in [53] is slightly superior.

We then computed another quality index, sometimes used in the demosaicing problem, i.e., the S-CIELAB metric. This metric indicates the percentage of color distortion between two

Table 2.3: CPSNRs for McMaster noiseless images

ImageM	[78]	[90]	[40]	[169]	[107]	proposed
1	25.59	26.63	27.69	29.56	29.41	**30.02**
2	32.46	33.64	34.47	**35.67**	35.35	35.51
3	31.63	31.42	32.93	33.29	34.05	**34.17**
4	33.23	33.63	36.28	36.63	38.00	**38.48**
5	29.98	31.01	32.00	34.79	34.43	**35.52**
6	31.98	33.87	35.55	39.26	38.83	**39.81**
7	37.82	35.99	36.87	36.00	37.04	**39.81**
8	36.62	36.46	37.47	37.76	37.30	**38.93**
9	33.28	34.51	36.21	37.84	36.84	**38.18**
10	34.97	36.01	37.56	39.24	39.12	**39.57**
11	35.97	36.73	38.39	40.02	**40.21**	39.81
12	35.78	36.64	37.39	39.15	**39.84**	39.27
13	37.47	38.76	40.34	41.60	40.66	**41.63**
14	36.25	37.43	38.53	**39.45**	39.11	39.26
15	36.35	37.33	38.29	**39.54**	39.25	39.44
16	29.02	30.05	31.17	34.03	**35.42**	34.36
17	27.99	28.63	30.41	33.56	33.19	**35.20**
18	32.49	33.30	34.20	35.38	**36.41**	35.10

Table 2.4: Parameters used for the noisy case

	derivative order		
	$k=1$	$k=2$	$k=3$
λ^N	0.1σ	0.05σ	0.05σ
κ^N	$\sqrt{\frac{10}{\sigma}}$	$5\sqrt{\frac{10}{\sigma}}$	$5\sqrt{\frac{10}{\sigma}}$
λ^V	0.1σ	0.05σ	0.05σ
κ^V	$\sqrt{\frac{10}{\sigma}}$	$5\sqrt{\frac{10}{\sigma}}$	$5\sqrt{\frac{10}{\sigma}}$

Table 2.5: CPSNRs for noisy Kodak images, $\sigma = 16$.

ImageK	bilinear	[78]	[91]	[170]	[122]	[53]	[53] variant	proposed
1	23.38	24.24	22.35	27.63	27.71	**28.18**	28.14	28.11
2	25.86	24.50	23.55	28.75	30.86	31.01	28.98	**31.47**
3	25.98	24.47	23.84	31.51	31.81	32.58	32.67	**32.71**
4	25.84	24.38	23.48	30.10	30.82	31.34	30.69	**31.36**
5	23.68	24.42	22.85	27.70	28.02	28.51	**28.60**	28.27
6	24.09	24.40	23.09	28.84	28.87	**29.51**	29.40	29.01
7	25.78	24.43	23.47	30.67	31.24	**32.29**	31.96	32.03
8	21.89	24.18	22.18	27.19	27.37	**28.40**	28.32	27.66
9	25.57	24.36	23.66	31.42	31.51	32.61	**32.83**	32.53
10	25.58	24.38	22.85	31.11	31.38	32.58	**32.60**	32.21
11	24.78	24.45	23.28	29.36	29.62	**30.20**	30.17	30.02
12	25.69	24.44	23.72	31.13	31.44	32.27	32.19	**32.34**
13	22.03	24.12	21.99	26.51	26.43	**26.78**	26.63	26.67
14	24.76	24.25	22.96	28.35	28.44	27.99	28.65	**29.01**
15	25.64	24.79	23.94	30.14	30.85	31.21	30.76	**31.30**
16	25.31	24.36	23.44	30.52	30.50	31.33	**31.37**	30.85
17	25.70	24.68	23.79	30.90	31.02	31.97	**32.05**	31.81
18	24.29	24.39	23.01	28.01	28.56	**28.99**	28.35	28.83
19	24.30	24.35	22.93	29.59	29.78	**30.74**	30.56	30.21
20	26.00	25.44	24.80	29.95	30.45	30.82	30.77	**30.93**
21	24.43	24.32	23.21	29.06	29.40	**30.07**	29.76	30.00
22	25.13	24.28	23.16	29.22	29.55	29.87	29.69	**29.91**
23	26.07	24.47	23.96	31.02	32.48	33.10	31.65	**33.31**
24	26.98	25.26	25.50	27.98	28.20	**28.81**	28.64	28.48

images, and accounts for the spatial-color sensitivity of the human eye (see [101, 172]). Since it returns a pixel-by-pixel matrix of errors, we assumed as the representative error index for the entire image the mean of the S-cielab matrix coefficients. The results obtained for the case $\sigma = 16$ on the Kodak dataset, along with the results of the most performing among the methods used for comparison, are shown in Table 2.6.

The results in this case are very good. The situation is even better when the noisy mosaiced MacMaster images are processed. The CPSNR results obtained for the same amount of noise ($\sigma = 16$), along with the results of the most performing methods used for comparison on the Kodak dataset, are shown in Table 2.7.

It is apparent that, this time, our method overcomes the other, with much higher values of CPSNR, which are only slightly lower than those we obtained in the noiseless case, for the same dataset. This excellent performance can be ascribed once again to our very fine modeling of natural images, in terms of local variations inside and between the color channels.

2.5 Geometry of the cliques and expression of the associated finite differences

The stabilizers used in this work are functions of the finite differences D_c^k of order k applied to sets c consisting of adjacent pixels. We call *clique of order k* the set of pixels on which the finite difference of order k is well-defined. We take $k = 1,2,3$ in order to reconstruct the finest details in images. Figures 2.4, 2.5 and 2.6 show the geometry of the sets c for the three orders of finite differences, respectively. As we can see, the cliques can be classified as *vertical* (Figures 2.4 (a), 2.5 (a), 2.6 (a)), *horizontal* (Figures 2.4 (b), 2.5 (c), 2.6 (d)), and *mixed* (Figures 2.5 (b), 2.6 (b) and (c)). The vertical cliques consist of the following pixels:

$$c = \{(i,j),(i+1,j),\ldots,(i+k,j)\}, \quad i = 1,\ldots,n-k,\ j = 1,\ldots,m,\ k = 1,2,3, \tag{2.20}$$

while the horizontal cliques have the form

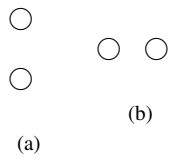
$$c = \{(i,j),(i,j+1),\ldots,(i,j+k)\}, \quad i = 1,\ldots,n,\ j = 1,\ldots,m-k,\ k = 1,2,3. \tag{2.21}$$

Table 2.6: S-Cielab errors for noisy Kodak images, $\sigma = 16$.

ImageK	[170]	[122]	[53]	[53] variant	proposed
1	4.11	4.22	3.82	3.82	**3.41**
2	3.34	2.81	2.74	3.19	**2.64**
3	3.28	3.24	2.97	2.89	**2.52**
4	3.44	3.32	3.16	3.23	**2.89**
5	4.20	4.07	4.02	3.92	**3.81**
6	3.86	3.83	3.47	3.48	**3.23**
7	3.54	3.43	3.14	3.21	**2.82**
8	4.48	4.57	4.08	4.02	**3.96**
9	3.10	3.19	2.85	2.72	**2.38**
10	3.16	3.21	2.81	2.73	**2.54**
11	3.26	3.31	3.02	2.98	**2.79**
12	3.10	3.00	2.75	2.74	**2.30**
13	4.65	4.78	4.42	4.47	**4.30**
14	3.87	3.97	3.93	3.67	**3.39**
15	3.09	2.82	2.82	2.79	**2.56**
16	3.23	3.36	2.97	2.89	**2.68**
17	2.67	2.81	2.46	2.32	**2.24**
18	4.28	3.90	3.80	4.14	**3.62**
19	3.69	3.66	3.32	3.37	**3.02**
20	3.85	3.36	3.26	3.30	**3.09**
21	3.67	3.63	3.26	3.30	**2.90**
22	4.14	3.92	3.77	3.92	**3.59**
23	3.22	2.94	2.76	2.99	**2.58**
24	4.28	4.10	3.72	3.81	**3.58**

Table 2.7: CPSNRs for noisy McMaster images, $\sigma = 16$.

ImageM	[170]	[122]	[53]	[53] variant	proposed
1	24.01	24.59	22.62	24.12	**29.36**
2	27.86	28.99	28.74	28.07	**34.85**
3	26.94	27.34	27.51	27.63	**33.53**
4	28.49	29.23	28.76	29.74	**37.44**
5	27.18	27.96	26.79	27.46	**34.55**
6	28.16	29.25	27.77	28.38	**39.12**
7	29.12	29.03	29.61	29.66	**36.04**
8	30.01	30.27	30.65	30.88	**38.47**
9	28.24	29.54	29.00	28.43	**37.65**
10	28.37	30.38	29.90	28.55	**39.09**
11	29.01	30.97	30.59	29.15	**39.79**
12	28.41	30.91	31.39	29.10	**38.98**
13	29.96	32.44	33.31	30.82	**40.94**
14	28.86	31.56	32.04	29.18	**38.52**
15	29.51	31.62	31.86	29.94	**39.08**
16	25.24	26.77	25.71	25.17	**34.24**
17	25.94	26.86	24.26	25.42	**34.69**
18	27.43	28.52	28.46	27.71	**35.00**



(a) (b)

Figure 2.4: Geometry of the sets c for $k = 1$.

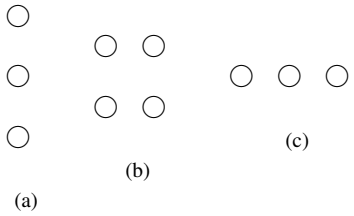


Figure 2.5: Geometry of the sets c for $k = 2$.

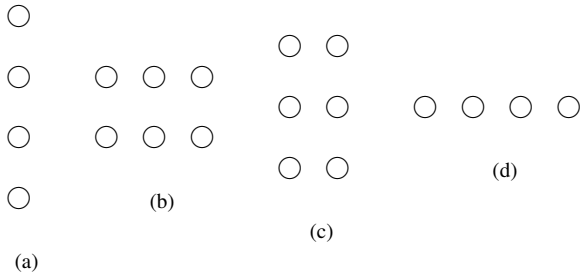


Figure 2.6: Geometry of the sets c for $k = 3$.

Let us now describe how finite differences are computed at a generic clique c for a generic color channel $\boldsymbol{x}^{(e)}$, $e \in \{r, g, b\}$. When $k = 1$, as is seen in Figure 2.4, we have two different kinds of finite difference operators, associated with a horizontal and a vertical finite difference, given by

$$D_c^1 \boldsymbol{x}^{(e)} = \begin{cases} \boldsymbol{x}_{(i,j)}^{(e)} - \boldsymbol{x}_{(i+1,j)}^{(e)} & \text{in case (a) of Figure 2.4;} \\ \boldsymbol{x}_{(i,j)}^{(e)} - \boldsymbol{x}_{(i,j+1)}^{(e)} & \text{in case (b) of Figure 2.4,} \end{cases} \tag{2.22}$$

respectively. When $k = 2$, we have three different kinds of finite difference operators, expressed by

$$D_c^2 \boldsymbol{x}^{(e)} = \begin{cases} \boldsymbol{x}_{(i,j)}^{(e)} - 2\boldsymbol{x}_{(i+1,j)}^{(e)} + \boldsymbol{x}_{(i+2,j)}^{(e)} & \text{in case (a) of Figure 2.5;} \\ \boldsymbol{x}_{(i,j)}^{(e)} - 2\boldsymbol{x}_{(i,j+1)}^{(e)} + \boldsymbol{x}_{(i,j+2)}^{(e)} & \text{in case (b) of Figure 2.5;} \\ \boldsymbol{x}_{(i,j)}^{(e)} - \boldsymbol{x}_{(i+1,j)}^{(e)} - \boldsymbol{x}_{(i,j+1)}^{(e)} + \boldsymbol{x}_{(i+1,j+1)}^{(e)} & \text{in case (c) of Figure 2.5.} \end{cases} \tag{2.23}$$

When $k = 3$, we get four different kinds of finite difference operators, given by

$$D_c^3 \boldsymbol{x}^{(e)} = \begin{cases} \boldsymbol{x}_{(i,j)}^{(e)} - 3\boldsymbol{x}_{(i+1,j)}^{(e)} + 3\boldsymbol{x}_{(i+2,j)}^{(e)} - \boldsymbol{x}_{(i+3,j)}^{(e)} & \text{in case (a) of Figure 2.6;} \\ \boldsymbol{x}_{(i,j)}^{(e)} - 3\boldsymbol{x}_{(i,j+1)}^{(e)} + 3\boldsymbol{x}_{(i,j+2)}^{(e)} - \boldsymbol{x}_{(i,j+3)}^{(e)} & \text{in case (b) of Figure 2.6;} \\ \boldsymbol{x}_{(i,j)}^{(e)} - 2\boldsymbol{x}_{(i+1,j)}^{(e)} + \boldsymbol{x}_{(i+2,j)}^{(e)} - \boldsymbol{x}_{(i,j+1)}^{(e)} + 2\boldsymbol{x}_{(i+1,j+1)}^{(e)} - \boldsymbol{x}_{(i+2,j+1)}^{(e)} & \text{in case (c) of Figure 2.6;} \\ \boldsymbol{x}_{(i,j)}^{(e)} - 2\boldsymbol{x}_{(i,j+1)}^{(e)} + \boldsymbol{x}_{(i,j+2)}^{(e)} - \boldsymbol{x}_{(i+1,j)}^{(e)} + 2\boldsymbol{x}_{(i+1,j+1)}^{(e)} - \boldsymbol{x}_{(i+1,j+2)}^{(e)} & \text{in case (d) of Figure 2.6.} \end{cases} \tag{2.24}$$

Let us introduce the concept of *adjacent clique of order k*, which is used to define the non-parallelism constraint. Given a vertical clique $c = \{(i,j),(i+1,j),\ldots,(i+k,j)\}$, $i = k+1,\ldots,n-k$, $j = 1,\ldots,m$, $k = 1,2,3$, we define its preceding clique $p_k(c)$ as follows:

$$p_k(c) = \{(i-k,j),(i-k+1,j),\ldots,(i,j)\}.$$

When $i = k+2,\ldots,n-k$, $j = 1,\ldots,m$, $k = 1,2,3$, a good approximation of $p_k(c)$ used to construct our approximating functions is given by

$$\pi_k(c) = \{(i-k-1,j),(i-k,j),\ldots,(i-1,j)\}.$$

We define $\pi_k(c)$ in such a way that $c \cap \pi_k(c) = \emptyset$. This will be useful to find the family of approximations of the energy function in Subsection 2.2.2.

If c is a horizontal clique, $c = \{(i,j),(i,j+1),\ldots,(i,j+k)\}$, $i = 1,\ldots,n$, $j = k+1,\ldots,m-k$, $k = 1,2,3$, then its preceding clique $p_k(c)$ is defined by

$$p_k(c) = \{(i,j-k),(i,j-k+1),\ldots,(i,j)\}.$$

When $i = 1,\ldots,n$, $j = k+2,\ldots,m-k$, $k = 1,2,3$, a good approximation of $p_k(c)$ is

$$\pi_k(c) = \{(i,j-k-1),(i,j-k),\ldots,(i,j-1)\}.$$

For mixed cliques and cliques on the board of the image, $p_k(c)$ and $\pi_k(c)$ are considered not to be defined.

2.6 Duality conditions on the stabilizer

In order that a stabilizer φ is edge-preserving and that the non-parallelism constraint on the implicit line process is satisfied, we require that the hypotheses of the following theorem are satisfied (see [33]):

Theorem 2.6.1. For every $p \in [1,2]$, let

$$\varphi^{(p)}(t_1,t_2) = (2-p)\,\overline{\varphi}(t_1,t_2) + (p-1)\,\varphi(t_1,t_2), \quad t_1 \in \mathbb{R}, t_2 \in [-\bar{t},\bar{t}], \tag{2.25}$$

where $\bar{t} = 2^k \cdot \sqrt{2} \cdot 255$, $k = 1,2,3$, for light intensity of the images in the range $[0,255]$, is the maximum value which the variable t_2 can assume, $\overline{\varphi}$ and φ are as in (2.13) and (2.7), respectively. Then $\varphi^{(p)}$ satisfies the following conditions:

H1) for every $t_2 \in [-\bar{t},\bar{t}]$, the function $\varphi_{t_2} : \mathbb{R} \to \mathbb{R} \cup \{-\infty\}$ defined by $\varphi_{t_2}(t_1) = \varphi(t_1,t_2)$ is upper semicontinuous and even on $\mathbb{R}$, and $\varphi_{t_2}(0) \in \mathbb{R}$;

H2) for each $t_2 \in [-\bar{t},\bar{t}]$, the function $\psi_{t_2} : \mathbb{R} \to \mathbb{R} \cup \{-\infty\}$ defined by

$$\psi_{t_2}(t_1) = \begin{cases} \varphi(\sqrt{t_1},t_2), & \text{if } t_1 \geq 0, \\ -\infty, & \text{if } t_1 < 0, \end{cases}$$

is concave on $\mathbb{R}_0^+$;

H3) φ_{t_2} is non-decreasing on $\mathbb{R}_0^+$ for every $t_2 \in [-\bar{t},\bar{t}]$;

H4) $\lim\limits_{t_1 \to +\infty} \dfrac{\psi_{t_2}(t_1)}{t_1} = 0$ for each $t_2 \in [-\bar{t},\bar{t}]$,

H5) there exists at least a real number t_1 such that the function $\varphi_{t_1}(t_2) = \varphi(t_1,t_2)$ is not constant on $[-\bar{t},\bar{t}]$, and φ_{t_1} is even on $[-\bar{t},\bar{t}]$ and non-decreasing on $[0,\bar{t}]$ for every $t_1 \in \mathbb{R}_0^+$.

Proof. We begin with proving that the function $\overline{\varphi}$ defined in (2.13) satisfies conditions H1), ..., H4).

It is readily seen that $\overline{\varphi}$ fulfils H1).

Now we prove H2). For $i = 1,2$ and $t_1 \in \mathbb{R}_0^+$, set

$$\overline{f_i}(t_1) = \overline{g_i}(\sqrt{t_1}) = \begin{cases} \lambda^2 t_1, & \text{if } 0 \leq t_1 \leq q_i^2, \\ \lambda^2 (2q_i\sqrt{t_1} - q_i^2), & \text{if } t_1 \geq q_i^2. \end{cases} \tag{2.26}$$

We have

$$\overline{\varphi}_{t_2}(\sqrt{t_1}) = \overline{\psi}_{t_2}(t_1) = \frac{\bar{t}^2 - t_2^2}{\bar{t}^2}\,\overline{f_1}(t_1) + \frac{t_2^2}{\bar{t}^2}\,\overline{f_2}(t_1),$$

and hence

$$\overline{f_i}'(t_1) = \begin{cases} \lambda^2, & \text{if } 0 \leq t_1 \leq q_i^2, \\ \lambda^2 q_i {t_1}^{-1/2}, & \text{if } t_1 \geq q_i^2; \end{cases}$$

$$\overline{f_i}''(t_1) = \begin{cases} 0, & \text{if } 0 \leq t_1 < q_i^2, \\ -\dfrac{1}{2}\lambda^2 q_i {t_1}^{-3/2}, & \text{if } t_1 > q_i^2. \end{cases}$$

Let $\gamma_{t_2} = \dfrac{\bar{t}^2 - t_2^2}{\bar{t}^2}$. Then, $1 - \gamma_{t_2} = \dfrac{t_2^2}{\bar{t}^2}$. It is not difficult to check that $0 \leq \gamma_{t_2} \leq 1$, since $|t_2| \leq \bar{t}$. Thus, for every $t_1 \in \mathbb{R}_0^+$ and $t_1 \neq q_1^2$, $t_1 \neq q_2^2$, $t_2 \in [-\bar{t},\bar{t}]$, we have

$$\overline{\psi}_{t_2}{}'(t_1) = \gamma_{t_2}\,\overline{f_1}'(t_1) + (1-\gamma_{t_2})\,\overline{f_2}'(t_1) \geq 0. \tag{2.27}$$

$$\overline{\psi}_{t_2}{}''(t_1) = \gamma_{t_2}\,\overline{f_1}''(t_1) + (1-\gamma_{t_2})\,\overline{f_2}''(t_1) \le 0.$$

Observe that the inequality in (2.27) will be useful to prove H3). Since $\overline{\psi}_{t_2}$ is of class C^1 on its domain (indeed, it is a composition of C^1 functions), then it is concave on $\mathbb{R}_0^+$ for all $t_2 \in [-\bar{t},\bar{t}]$. So, $\overline{\varphi}$ satisfies condition H2).

Now we prove H3). From (2.27) it follows that $\overline{\psi}_{t_2}$ is non-decreasing on $\mathbb{R}_0^+$, and hence so is $\overline{\varphi}_{t_2}$. Thus we get

$$\overline{\varphi}_{t_2}{}'(t_1) = \gamma_{t_2}\,\overline{g_1}'(t_1) + (1-\gamma_{t_2})\,\overline{g_2}'(t_1) \ge 0$$

for each $t_1 \in \mathbb{R}_0^+$. Thus, H3) holds.

Now we show that $\overline{\varphi}$ fulfils H4). Indeed, we have

$$\lim_{t_1\to+\infty} \frac{\overline{f_i}(t_1)}{t_1} = \lim_{t_1\to+\infty} \frac{\lambda^2\,(2\,q_i\sqrt{t_1} - q_i^2)}{t_1} = 0 \quad (i=1,2),$$

and hence

$$\lim_{t_1\to+\infty} \frac{\overline{\psi}_{t_2}(t_1)}{t_1} = 0 \quad \text{for every } t_2 \in [-\bar{t},\bar{t}].$$

Finally, we prove that $\overline{\varphi}$ satisfies H5).

Take $t_1 = q_2$. For each $t_2 \in [-\bar{t},\bar{t}]$, it is

$$\overline{\varphi}(q_2,t_2) = \frac{\bar{t}^2 - t_2^2}{\bar{t}^2}\,\lambda^2\,(2\,q_1\,q_2 - q_1^2) + \frac{t_2^2}{\bar{t}^2}\,\lambda^2\,q_2^2,$$

and hence $\overline{\varphi}(q_2,0) = \lambda^2\,(2\,q_1\,q_2 - q_1^2)$, $\overline{\varphi}(q_2,\bar{t}) = \lambda^2\,q_2^2$. We claim that $\overline{\varphi}(q_2,0) \neq \overline{\varphi}(q_2,\bar{t})$. If not, then we would have $2\,q_1\,q_2 - q_1^2 = q_2^2$, and hence $0 = q_2^2 - 2\,q_1\,q_2 + q_1^2 = (q_1-q_2)^2$, that is $q_1 = q_2$, which is absurd, since we know that $0 < q_1 < q_2$. Therefore, the function $t_2 \mapsto \overline{\varphi}(q_2,t_2)$ is not constant, and hence the first property of H5) is satisfied.

Moreover, it is easy to see that $\overline{\varphi}_{t_1}$ is even on $\mathbb{R}$ for each $t_1 \in \mathbb{R}$.

From (2.12) it is not difficult to deduce that $\overline{g_2}(t_1) - \overline{g_1}(t_1) \ge 0$ for all $t_1 \ge 0$. We get

$$\frac{d\overline{\varphi}_{t_1}}{dt_2}(t_2) = -\frac{2t_2}{\bar{t}^2}\,\overline{g_1}(t_1) + \frac{2t_2}{\bar{t}^2}\,\overline{g_2}(t_1) = \frac{2t_2}{\bar{t}^2}\,(\overline{g_2}(t_1) - \overline{g_1}(t_1)) \ge 0 \tag{2.28}$$

for any $t_1 \in \mathbb{R}_0^+$. Hence, the function $\overline{\varphi}_{t_1}$ is non-decreasing on $\mathbb{R}_0^+$ for every $t_1 \in \mathbb{R}_0^+$. Thus, H5) is proved.

Now we prove that for $i = 1,2$ the function φ defined in (2.7) satisfies conditions Hj), $j = 1,\ldots,4$.

It is not difficult to see that, by construction, H1) holds.

We now prove H2). We begin with the case when $|t_2| \leq s$ or $t_2 \geq \zeta$ in (2.7). For $i = 1,2$, set

$$f_i(t_1) = g_i(\sqrt{t_1}) = \begin{cases} \lambda^2 t_1, & \text{if } 0 \leq t_1 \leq q_i^2, \\ \alpha_i - \dfrac{\tau}{2}(\sqrt{t_1} - r_i)^2, & \text{if } q_i^2 \leq t_1 \leq r_i^2, \\ \alpha_i, & \text{if } t_1 \geq r_i^2. \end{cases} \tag{2.29}$$

We have

$$\varphi(\sqrt{t_1}, t_2) = \psi_{t_2}(t_1) = \begin{cases} f_1(t_1), & \text{if } |t_2| \leq s, \\ \Big(1 - \dfrac{2(|t_2|-s)^2}{(\zeta - s)^2}\Big) f_1(t_1) + \dfrac{2(|t_2|-s)^2}{(\zeta-s)^2} f_2(t_1), & \text{if } s < |t_2| \leq \dfrac{\zeta + s}{2}, \\ \dfrac{2(|t_2|-\zeta)^2}{(\zeta-s)^2} f_1(t_1) + \Big(1 - \dfrac{2(|t_2|-\zeta)^2}{(\zeta-s)^2}\Big) f_2(t_1), & \text{if } \dfrac{\zeta+s}{2} < |t_2| < \zeta, \\ f_2(t_1), & \text{if } |t_2| \geq \zeta. \end{cases} \tag{2.30}$$

We claim that f_i is non-decreasing and concave on $\mathbb{R}_0^+$.We get

$$f_i'(t_1) = \begin{cases} \lambda^2, & \text{if } 0 \leq t_1 \leq q_i^2, \\ \alpha_i - \dfrac{\tau}{2}\Big(1 - \dfrac{r_i}{\sqrt{t_1}}\Big), & \text{if } q_i^2 \leq t_1 \leq r_i^2, \\ 0, & \text{if } t_1 \geq r_i^2. \end{cases} \tag{2.31}$$

Note that f_i is C^1, since it is a composition of functions of class C^1. Moreover, we have

$$f_i''(t_1) = \begin{cases} 0, & \text{if } 0 \leq t_1 < q_i^2, \\ -\dfrac{\tau r_i}{4\sqrt{t_1^3}}, & \text{if } q_i^2 < t_1 < r_i^2, \\ 0, & \text{if } t_1 > r_i^2. \end{cases} \tag{2.32}$$

From this we deduce that φ fulfils H2), at least when $|t_2| \leq s$ or $|t_2| \geq \zeta$.

Now we examine the case

$$s < |t_2| \leq \frac{\zeta + s}{2}. \tag{2.33}$$

We have

$$\psi'_{t_2}(t_1) = \Big(1 - \frac{2(|t_2| - s)^2}{(\zeta - s)^2}\Big) f'_1(t_1) + \frac{2(|t_2| - s)^2}{(\zeta - s)^2} f'_2(t_1), \tag{2.34}$$

$$\psi''_{t_2}(t_1) = \Big(1 - \frac{2(|t_2| - s)^2}{(\zeta - s)^2}\Big) f''_1(t_1) + \frac{2(|t_2| - s)^2}{(\zeta - s)^2} f''_2(t_1). \tag{2.35}$$

Observe that $\frac{2(|t_2| - s)^2}{(\zeta - s)^2} \geq 0$. Now we claim that $\frac{2(|t_2| - s)^2}{(\zeta - s)^2} \leq 1$. Indeed, since $s < |t_2| \leq \frac{\zeta + s}{2}$, then $0 < |t_2| - s \leq \frac{\zeta - s}{2} < \frac{\zeta - s}{\sqrt{2}}$, and hence $(|t_2| - s)^2 \leq \frac{(\zeta - s)^2}{2}$, getting the claim. Therefore, $1 - \frac{2(|t_2| - s)^2}{(\zeta - s)^2} \geq 0$. From this, since $f'_i(t_1) \geq 0$ for every $t_1 \geq 0$ and $f''_i(t_1) \leq 0$ for each $t_1 \mathbb{R}_0^+$, $t_1 \neq q_1$, $t_1 \neq q_2$, in the case (2.33) we obtain

$$\psi'_{t_2}(t_1) \geq 0 \text{ for every } t_1 \in \mathbb{R}_0^+, \tag{2.36}$$

$$\psi''_{t_2}(t_1) \leq 0 \text{ for any } t_1 \in \mathbb{R}_0^+, t_1 \neq q_1, t_1 \neq q_2. \tag{2.37}$$

Note that the inequality in (2.36) will be useful to prove H3).

From (2.37), taking into account the continuity of ψ_{t_2}, we deduce that φ satisfies H2) also in the case (2.33).

Now we consider the case

$$\frac{\zeta + s}{2} < |t_2| < \zeta. \tag{2.38}$$

We get

$$\psi'_{t_2}(t_1) = \frac{2(|t_2| - \zeta)^2}{(\zeta - s)^2} f'_1(t_1) + \Big(1 - \frac{2(|t_2| - \zeta)^2}{(\zeta - s)^2}\Big) f'_2(t_1), \tag{2.39}$$

$$\psi''_{t_2}(t_1) = \frac{2(|t_2| - \zeta)^2}{(\zeta - s)^2} f''_1(t_1) + \Big(1 - \frac{2(|t_2| - \zeta)^2}{(\zeta - s)^2}\Big) f''_2(t_1). \tag{2.40}$$

Note that $\frac{2(|t_2|-\zeta)^2}{(\zeta-s)^2} \geq 0$. Now we claim that $\frac{2(|t_2|-\zeta)^2}{(\zeta-s)^2} \leq 1$. Indeed, as $\frac{\zeta+s}{2} < |t_2| < \zeta$, then $0 < \zeta - |t_2| \leq \frac{\zeta-s}{2} < \frac{\zeta-s}{\sqrt{2}}$, and so $(|t_2|-\zeta)^2 \leq \frac{(\zeta-s)^2}{2}$, getting the claim. Thus, $1 - \frac{2(|t_2|-\zeta)^2}{(\zeta-s)^2} \geq 0$. From this, in the case (2.38), analogously as in the case (2.33), we obtain

$$\psi'_{t_2}(t_1) \geq 0 \text{ for every } t_1 \in \mathbb{R}_0^+, \tag{2.41}$$

$$\psi''_{t_2}(t_1) \leq 0 \text{ for any } t_1 \in \mathbb{R}_0^+, t_1 \neq q_1, t_1 \neq q_2. \tag{2.42}$$

From (2.42) and thanks to the continuity of ψ_{t_2}, it follows that φ satisfies H2) also in the case (2.38).

Now we prove that φ satisfies H3).

First, when $|t_2| \leq s$ or $|t_2| \geq \zeta$, observe that it is readily seen that g_i is non-decreasing on $\mathbb{R}_0^+$ for $i = 1, 2$. Hence, ψ_{t_2} is non-decreasing on $\mathbb{R}_0^+$, and thus H3) holds.

Moreover, when t_2 satisfies the case (2.33) or the case (2.38), from (2.36) and (2.41) it follows that ψ_{t_2} is non-decreasing on $\mathbb{R}_0^+$, and hence φ_{t_2} is too. Thus, φ fulfils H3).

Now we prove H4). Let That φ satisfies H4) is a consequence of the fact that $\lim_{t_1 \to +\infty} \frac{f_i(t_1)}{t_1} = 0$.

Now we prove H5). Let

$$a(t_1) = 2\,\frac{g_2(t_1) - g_1(t_1)}{(\zeta-s)^2}, \quad t_1 \in \mathbb{R}.$$

First of all, observe that $\varphi(r_2, s) = g_1(r_2) = \alpha$, $\varphi(r_2, \zeta) = g_2(r_2) = \alpha + \varepsilon \neq \varphi(r_2, s)$. Thus, the function $t_2 \mapsto \varphi(r_2, t_2)$ is not constant. Moreover, it is easy to see that the function $\varphi_{t_1}(t_2)$ is even on $[-\bar{t}, \bar{t}]$ for every $t_1 \in \mathbb{R}_0^+$, since it depends on $|t_2|$. Furthermore, it is not difficult to check that

$$g_2(t_1) \geq g_1(t_1) \quad \text{for every } t_1 \in \mathbb{R}_0^+. \tag{2.43}$$

Let $t_2 \in [0, \bar{t}]$. We get

$$\varphi_{t_1}{}'(t_2) = -\frac{2(t_2-s)^2}{(\zeta-s)^2}\,g_1(t_1) + \frac{2(t_2-s)^2}{(\zeta-s)^2}\,g_2(t_1) \tag{2.44}$$

in the case (2.33), and

$$\varphi_{t_1}{}'(t_2) = -\frac{2(t_2-\zeta)^2}{(\zeta-s)^2}\,g_1(t_1) - \frac{2(t_2-\zeta)^2}{(\zeta-s)^2}\,g_2(t_1) \tag{2.45}$$

in the case (2.38). From (2.43), (2.44) and (2.45) it follows that $\varphi'_{t_1}(t_2) \geq 0$ for each $t_2 \in [0, \bar{t}]$. By arbitrariness of $t_1 \in \mathbb{R}_0^+$, we deduce that φ satisfies H5).

Now, we observe that the functions $\varphi^{(p)}$, $p \in [0,2]$, satisfy conditions Hj), $j = 1,\dots,4$, since they are non-negative linear combinations of functions satisfying Hj), $j = 1,\dots,4$. Since $\overline{\varphi}_{t_1}$ and φ_{t_1} are non-decreasing for each $t_1 \in \mathbb{R}_0^+$, $\overline{\varphi}_{t_2}$ and φ_{t_2} are non-decreasing for every $t_2 \in [-\bar{t},\bar{t}]$, and the functions $t_2 \mapsto \overline{\varphi}(q_2,t_2)$, $t_2 \mapsto \varphi(r_2,t_2)$ are not constant, it follows that for every $p \in [0,2]$ there exists at least a $t_1 \in \mathbb{R}_0^+$ such that the function $t_2 \mapsto \varphi^{(p)}(t_1,t_2)$ is not constant. The other properties of H5) hold, because the $\varphi^{(p)}$'s are non-negative linear combinations of functions satisfying H5). □

2.7 Convergence of the NL-SOR algorithm

To minimize each approximation $E^{(p)}$, $p \in [0,2]$, we use the NL-SOR algorithm.

We will prove the existence of suitable limit points, which are stationary points of $E^{(0)}$ (in general, they are not minimum points of $E^{(0)}$). In [38, Theorem 2] the convergence of the algorithm is proved when $E^{(0)}$ is strictly convex and of class C^2. Such assumptions are too strong for the componentwise convex approximation of the regularization term in our setting, because we deal with functions of class C^1. So, we give an extension of the theorem under these weaker hypotheses.

First, we state the following technical lemma.

Lemma 2.7.1. *Let $\phi : [x_0,\bar{x}] \to \mathbb{R}$ be convex, of class C^1, having both left and right second derivative on $[x_0,\bar{x}]$. Suppose that ϕ is second differentiable on $[x_0,\bar{x}] \setminus P$, where $P = \{x_j : j = 1,\dots,N\}$, with $x_0 < x_1 < \dots < x_N < \bar{x}$.*

Then, for every $x \in [x_0,\bar{x}]$ there exist $\xi \in]x_0,x[$ and $\mu \geq 0$, such that

$$\mu \in I_\xi = [\min\{\phi''_-(\xi),\phi''_+(\xi)\},\max\{\phi''_-(\xi),\phi''_+(\xi)\}] \tag{2.46}$$

and

$$\phi(x) = \phi(x_0) + (x-x_0)\phi'(x_0) + \frac{(x-x_0)^2}{2}\,\mu. \tag{2.47}$$

Proof. If $g = f'$, then g is continuous and non-decreasing in $[x_0,\bar{x}]$. Define $h : [x_0,\bar{x}] \to \mathbb{R}$ by

$$h(y) = g(x_0) + \frac{g(x)-g(x_0)}{x-x_0}\,(y-x_0). \tag{2.48}$$

Note that h is the equation of the line passing through the points $(x_0, g(x_0))$ and $(x, g(x))$. Let $\mu = \dfrac{g(x) - g(x_0)}{x - x_0}$ be its angular coefficient. We first treat the case

$$g'(x_0) < \mu < g'(x), \tag{2.49}$$

and begin with considering the interval $[x_0, x_1]$. If

$$\min\{g'(x_0), g'_-(x_1)\} \leq \mu \leq \max\{g'(x_0), g'_-(x_1)\},$$

then by the Darboux theorem there is $\xi \in]x_0, x_1[$ with $\mu = g'(\xi)$, and in particular $\mu \in I_\xi$. Moreover, observe that, if

$$\min\{g'_-(x_1), g'_+(x_1)\} \leq \mu \leq \max\{g'_-(x_1), g'_+(x_1)\},$$

then of course $\mu \in I_{x_1}$. From this it follows that, if

$$\min\{g'(x_0), g'_-(x_1), g'_+(x_1)\} \leq \mu \leq \max\{g'(x_0), g'_-(x_1), g'_+(x_1)\},$$

then there exists $\xi \in]x_0, x_1]$ such that $\mu \in I_\xi$.

By considering the interval $]x_0, x_2]$, proceeding analogously as above, it is possible to check that, if

$$\min\{g'(x_0), g'_-(x_1), g'_+(x_1), g'_-(x_2), g'_+(x_2)\} \leq \mu \leq \max\{g'(x_0), g'_-(x_1), g'_+(x_1), g'_-(x_2), g'_+(x_2)\},$$

then there is $\xi \in]x_0, x_2]$ with $\mu \in I_\xi$. Similarly, taking the interval $]x_0, x[$, we can prove that, if

$$\begin{aligned} & \min\{g'(x_0), g'_-(x_1), g'_+(x_1), \ldots, g'_-(x_N), g'_+(x_N), g'(x)\} \leq \mu \leq \\ \leq \quad & \max\{g'(x_0), g'_-(x_1), g'_+(x_1), \ldots, g'_-(x_N), g'_+(x_N), g'(x)\}, \end{aligned} \tag{2.50}$$

then there exists $\xi \in]x_0, x[$ such that $\mu \in I_\xi$. Therefore, this property holds in the case (2.49), since (2.49) implies (2.50). Analogously, it is possible to show that, even when

$$g'(x) < \mu < g'(x_0),$$

there exists $\xi \in]x_0, x_1]$ with $\mu \in I_\xi$.

Now we consider the case

$$g'(x) > \mu \text{ and } g'(x_0) > \mu. \tag{2.51}$$

Now we claim that there exists at least a point $c \in]x_0, x[$ with

$$g(c) = h(c). \tag{2.52}$$

Indeed, it is not difficult to check that there exist two positive real numbers ε_1 and ε_2 such that

$$g(x_0+\varepsilon_1) > h(x_0+\varepsilon_1) \text{ and } g(x-\varepsilon_2) < h(x-\varepsilon_2).$$

Set $h_0(y) = g(y) - h(y)$, $y \in [x_0, \bar{x}]$. We get that h_0 is continuous, $h_0(x_0+\varepsilon_1) > 0$ and $h_0(x-\varepsilon_2) < 0$. Thus there exists $c \in]x_0+\varepsilon_1, x-\varepsilon_2[$ with $h_0(c) = 0$, getting the claim. Let c_1 be the smallest point satisfying (2.52). Since $(g-h)(c_1) = 0$ and $(g-h)(x) > 0$ for every $x < c_1$, we obtain $(g-h)'_-(c_1) < 0$, and hence $g'(c_1) < \mu$. Now, arguing analogously as in the case (2.49) and taking the interval $]x_0, c_1[$ instead of $]x_0, x[$, we get the existence of an element $\xi \in]x_0, c_1[$ such that $\mu \in I_\xi$. This result can be proved even when

$$g'(x) < \mu \text{ and } g'(x_0) < \mu,$$

by arguing analogously as in the case (2.51). Then we get

$$g(x) = g(x_0) + (x - x_0)\,\mu. \tag{2.53}$$

By integrating between x_0 and x both members of (2.53), we obtain

$$f(x) = f(x_0) + (x-x_0)f'(x_0) + \frac{(x-x_0)^2}{2}\mu.$$

This ends the proof. □

Observe that the NL-SOR algorithm, when $p = 0$, can be formulated as follows:

given the initial vector $\mathbf{x}^{(0,mn,b)}$

for $l = 1, 2, \ldots$

for $i = 1, 2, \ldots, nm$

for $e = r, g, b$

set the vector $x^{(l,i,e)} \in \mathbb{R}^{3mn}$ as in Eq. (2.54)

end for

end for

end for

At the iterate $l \in \mathbb{N}$, fixed $i \in 1,2,\ldots,3m$ and $e \in \{r,g,b\}$, the vector $\mathbf{x}^{(l,i,e)}$ is defined by

$$(\mathbf{x}^{(l,i,e)})_j^{(q)} = \begin{cases} (x^{\text{prec}(l,i,e)})_j^{(q)} & \text{if } i \neq j \text{ or } q \neq e, \\ (x^{\text{prec}(l,i,e)})_j^{(q)} - \dfrac{\omega}{T} \dfrac{\partial E^{(0)}\left(x^{\text{prec}(l,i,e)}\right)}{\partial x_i^{(e)}} & \text{if } i = j \text{ and } q = e, \end{cases} \tag{2.54}$$

where

$$x^{\text{prec}(l,i,e)} = \begin{cases} x^{(l,i,e-1)} & \text{if } e \neq r; \\ x^{(l,i-1,b)} & \text{if } i \neq 1, e = r; \\ x^{(l-1,nm,b)} & \text{if } i = 1, e = r. \end{cases}$$

The formulated algorithm allows to denote the image vectors actually defined at each iterate l, at every pixel i and at each color e. We observe that the algorithm here proposed is a particular case of that given in Section 2.7, and has been suitably modified in order to give a rigorous definition of the image vector $\mathbf{x}$ at every iterate l, at every pixel i and at each color e.

Moreover, fixed the step (l,i,e), let $\mathbf{x}^{(l,i,e)} \setminus (x^{(l,i,e)})_i^{(e)} \in \mathbb{R}^{3nm-1}$ be the vector whose elements are those of $\mathbf{x}^{(l,i,e)}$ except $(x^{(l,i,e))})_i^{(e)}$. The value of this pixel $(x^{(l,i,e))})_i^{(e)}$ is an unknown variable, which we call z. For each fixed value of $\mathbf{x}^{(l,i,e)} \setminus (x^{(l,i,e))})_i^{(e)}$, let us define the following energy function $\overline{E}^{(l,i,e)} : \mathbb{R} \to \mathbb{R}$ by

$$\overline{E}^{(l,i,e)}(z) = E^{(0)}(\mathbf{x}^{(l,i,e)} \setminus (x^{(l,i,e))})_i^{(e)}, z). \tag{2.55}$$

Theorem 2.7.2. Let $E^{(0)} : \mathbb{R}^{3nm} \to \mathbb{R}$ be a function of class C^1 and coercive, that is

$$\lim_{\|x\| \to +\infty} E^{(0)}(x) = +\infty; \tag{2.56}$$

fix $\mathbf{x}^{(0,nm,b)} \in \mathbb{R}^{3nm}$, and let $\{\mathbf{x}^{(l,i,e)}\}$, $l \in \mathbb{N}, i = 1,2,\ldots,3m, e \in \{r,g,b\}$, be the sequence defined iteratively in (2.54).

where $0 < \omega < 2$ and

$$T > \max_{i=1,2,\ldots,nm, e=r,g,b} \max_{\mathbf{x}} \left\{ \frac{\partial_+^2 E^{(0)}(\mathbf{x})}{(\partial x_i^{(e)})^2}, \frac{\partial_-^2 E^{(0)}(\mathbf{x})}{(\partial x_i^{(e)})^2} \right\}. \tag{2.57}$$

Let $\overline{E}^{(l,i,e)} : \mathbb{R} \to \mathbb{R}$ be the function defined as in (2.55). Assume that $\overline{E}^{(l,i,e)}$ is convex, admits both left and right derivative on $\mathbb{R}$ and is not second differentiable (at most) at a finite number of points.

Then, $\lim_{(l,i,e)} \nabla E^{(0)}(x^{(l,i,e)}) = 0$.

Proof. We begin with proving that, during the updating of $(x^{(l,i,e))})_i^{(e)}$, the function $E^{(0)}$ is non-increasing. If

$$\overline{E}^{(l,i,e)'}((x^{\text{prec}(l,i,e))})_i^{(e)}) = 0, \tag{2.58}$$

then, since in (2.54) it is $(x^{\text{prec}(l,i,e))})_i^{(e)} = \dfrac{\partial E^{(0)}\left(x^{\text{prec}(l,i,e)}\right)}{\partial x_i^{(e)}}$, we get

$$(\mathbf{x}^{(l,i,e)})_i^{(e)} = (x^{\text{prec}(l,i,e)})_i^{(e)} \tag{2.59}$$

and hence $\mathbf{x}^{(l,i,e)} = \mathbf{x}^{\text{prec}(l,i,e)}$, and the value of the energy function does not change.

Now we treat the case when

$$(x^{\text{prec}(l,i,e))})_i^{(e)} \neq 0. \tag{2.60}$$

Note that, by (2.56), we get

$$\lim_{z\to+\infty} \overline{E}^{(l,i,e)}(z) = \lim_{z\to-\infty} \overline{E}^{(l,i,e)}(z) = +\infty, \tag{2.61}$$

that is the function $\overline{E}^{(l,i,e)}$ is coercive on $\mathbb{R}$. Since $\overline{E}^{(l,i,e)}$ is also continuous, then, by [16, Theorem 2.32], $\overline{E}^{(l,i,e)}$ assumes the minimum value, say $(t_*)^{(l,i,e)}$.

We get that, for any $t > (t_*)^{(l,i,e)}$, the level set $L_t = \{z \in \mathbb{R} : \overline{E}^{(l,i,e)}(z) = t\}$ has exactly two points, and $\overline{E}^{(l,i,e)}(z) < t$ whenever z is in the interior of the interval whose endpoints are the elements of L_t Now we claim that, for every $t > (t_*)^{(l,i,e)}$, the level set $L_t = \{z \in \mathbb{R} : \overline{E}^{(l,i,e)}(z) = t^{(l,i,e)}\}$ has exactly two points.

Since $\overline{E}^{(l,i,e)}$ is convex and differentiable, we get that $\overline{E}^{(l,i,e)}(z) = (t_*)^{(l,i,e)}$ if and only if $\overline{E}^{(l,i,e)'}(z) = 0$. From the continuity of $\overline{E}^{(l,i,e)}$ and (2.56) it follows that $\overline{E}^{(l,i,e)}$ assumes all values $t \in [(t_*)^{(l,i,e)}, +\infty[$. Since $\overline{E}^{(l,i,e)'}$ is non-decreasing, then $\overline{E}^{(l,i,e)'}$ is positive (resp. negative), and hence $\overline{E}^{(l,i,e)}$ is strictly increasing (resp. decreasing) at all points which are greater (resp. smaller) than the minimum points of $\overline{E}^{(l,i,e)}$. Thus, $\overline{E}^{(l,i,e)}$ assumes each value $t > (t_*)^{(l,i,e)}$ exactly two times, getting the claim.

Now, set $t^{(l,i,e)} = \overline{E}^{(l,i,e)}((x^{\mathrm{prec}(l,i,e))})_i^{(e)})$, and $L_{t^{(l,i,e)}} = \{(x^{\mathrm{prec}(l,i,e))})_i^{(e)}, \overline{z}^{(l,i,e)}\}$, where $\overline{E}^{(l,i,e)}(\overline{z}^{(l,i,e)}) = t^{(l,i,e)}$.

Without loss of generality, let us consider the case $\overline{z}^{(l,i,e)} < (x^{\mathrm{prec}(l,i,e))})_i^{(e)}$. Note that, in this case, $\overline{E}^{(l,i,e)'}(\overline{z}^{(l,i,e)}) < 0$, while $\overline{E}^{(l,i,e)'}((x^{\mathrm{prec}(l,i,e))})_i^{(e)}) > 0$.

By hypothesis, taking into account that $\overline{E}^{(l,i,e)}$ is of class C^1, from Lemma 2.7.1 applied to the interval $[\overline{z}^{(l,i,e)}, (x^{\mathrm{prec}(l,i,e))})_i^{(e)}]$ we find $\xi \in]\overline{z}^{(l,i,e)}, (x^{\mathrm{prec}(l,i,e))})_i^{(e)}[$ and $\mu \geq 0$, such that

$$\min\{\overline{E}^{(l,i,e)''}_{-}(\xi), \overline{E}^{(l,i,e)''}_{+}(\xi)\} \leq \mu \leq \max\{\overline{E}^{(l,i,e)''}_{-}(\xi), \overline{E}^{(l,i,e)''}_{+}(\xi)\} \tag{2.62}$$

and

$$\begin{aligned} \overline{E}^{(l,i,e)}(\overline{z}^{(l,i,e)}) &= \overline{E}^{(l,i,e)}((x^{\mathrm{prec}(l,i,e))})_i^{(e)}) + \overline{E}^{(l,i,e)'}((x^{\mathrm{prec}(l,i,e))})_i^{(e)})(\overline{z}^{(l,i,e)} - (x^{\mathrm{prec}(l,i,e))})_i^{(e)}) \\ &+ \frac{1}{2}\mu\,(\overline{z}^{(l,i,e)} - (x^{\mathrm{prec}(l,i,e))})_i^{(e)})^2. \end{aligned}$$

Since $\overline{E}^{(l,i,e)}((x^{\mathrm{prec}(l,i,e))})_i^{(e)}) = \overline{E}^{(l,i,e)}(\overline{z}^{(l,i,e)})$, then we have

$$\overline{E}^{(l,i,e)'}((x^{\mathrm{prec}(l,i,e))})_i^{(e)})\,(\overline{z}^{(l,i,e)} - (x^{\mathrm{prec}(l,i,e))})_i^{(e)}) + \frac{1}{2}\mu\,(\overline{z}^{(l,i,e)} - (x^{\mathrm{prec}(l,i,e))})_i^{(e)})^2 = 0. \tag{2.63}$$

Note that from (2.62) and (2.57) we get

$$\mu \leq T. \tag{2.64}$$

Now we claim that $\mu > 0$. Indeed, if $\mu = 0$, then from (2.63) we get

$$\overline{E}^{(l,i,e)'}((x^{\mathrm{prec}(l,i,e))})_i^{(e)})\,(\overline{z}^{(l,i,e)} - (x^{\mathrm{prec}(l,i,e))})_i^{(e)}) = 0,$$

and hence $\overline{z}^{(l,i,e)} = (x^{\mathrm{prec}(l,i,e))})_i^{(e)}$, because $\overline{E}^{(l,i,e)'}((x^{\mathrm{prec}(l,i,e))})_i^{(e)}) > 0$. This is absurd, because $\overline{z}^{(l,i,e)} < (x^{\mathrm{prec}(l,i,e))}{}_i^{(e)}$. Therefore, we get the claim. From (2.63) we obtain

$$(x^{\mathrm{prec}(l,i,e))})_i^{(e)} - \overline{z}^{(l,i,e)} = \frac{2}{\mu}\overline{E}^{(l,i,e)'}((x^{\mathrm{prec}(l,i,e))})_i^{(e)}). \tag{2.65}$$

We recall that, by (2.54), it is

$$(x^{(l,i,e))})_i^{(e)} = (x^{\mathrm{prec}(l,i,e))})_i^{(e)} - \frac{\omega}{T}\overline{E}^{(l,i,e)'}((x^{\mathrm{prec}(l,i,e))})_i^{(e)}). \tag{2.66}$$

Since $0 < \omega < 2$, from (2.64), (2.65) and (2.66) we have

$$\begin{aligned} 0 &< (x^{(l,i,e))})_i^{(e)} - x^{\mathrm{prec}(l,i,e)})_i^{(e)} = (x^{(l,i,e))})_i^{(e)} = \frac{\omega}{T}\overline{E}^{(l,i,e)'}((x^{\mathrm{prec}(l,i,e))})_i^{(e)}) < \\ &< \frac{2}{T}\overline{E}^{(l,i,e)'}(x^{\mathrm{prec}(l,i,e))}{}_i^{(e)}) \leq \frac{2}{\mu}\overline{E}^{(l,i,e)'}(x^{\mathrm{prec}(l,i,e))})_i^{(e)} = (x^{\mathrm{prec}(l,i,e))})_i^{(e)} - \overline{z}^{(l,i,e)}. \end{aligned} \tag{2.67}$$

From (2.67) it follows that

$$\overline{z}^{(l,i,e)} < \left(x^{(l,i,e))}\right)_i^{(e)} < \left(x^{\mathrm{prec}(l,i,e))}\right)_i^{(e)} \quad \text{when } \overline{z}^{(l,i,e)} < \left(x^{\mathrm{prec}(l,i,e))}\right)_i^{(e)}. \tag{2.68}$$

Analogously, it is possible to prove that

$$\overline{z}^{(l,i,e)} > \left(x^{(l,i,e))}\right)_i^{(e)} > \left(x^{\mathrm{prec}(l,i,e))}\right)_i^{(e)} \quad \text{when } \overline{z}^{(l,i,e)} > \left(x^{\mathrm{prec}(l,i,e))}\right)_i^{(e)}. \tag{2.69}$$

Thus, in the case (2.60), we get $\overline{E}^{(l,i,e)}(\left(x^{(l,i,e))}\right)_i^{(e)}) < \overline{E}^{(l,i,e)}(\left(x^{\mathrm{prec}(l,i,e))}\right)_i^{(e)})$. Therefore, in both cases (2.58) and (2.60), the sequence $\{\overline{E}^{(l,i,e)}(\left(x^{(l,i,e))}\right)_i^{(e)})\}$, $l \in \mathbb{N}$, $i = 1,2,\ldots,nm$, $e \in \{r,g,b\}$, is non-increasing. Since $E^{(0)}$ is bounded from below, then the sequence $\{\overline{E}^{(l,i,e)}(\left(x^{(l,i,e))}\right)_i^{(e)})\}$, $l \in \mathbb{N}$, $i = 1,2,\ldots,nm$, $e \in \{r,g,b\}$, is non-increasing, and hence it is convergent.

Now we claim that the sequence $\{\left(x^{(l,i,e))}\right)_i^{(e)} - \left(x^{\mathrm{prec}(l,i,e))}\right)_i^{(e)}\}$, $l \in \mathbb{N}$, $i = 1,2,\ldots,nm$, $e \in \{r,g,b\}$, converges to 0.

Fix $l \in \mathbb{N}$, $i = 1,2,\ldots,nm$, $e \in \{r,g,b\}$. If $\overline{E}^{(l,i,e)'}(\left(x^{\mathrm{prec}(l,i,e))}\right)_i^{(e)}) = 0$, then, as seen in (2.59), we get $\left(x^{(l,i,e))}\right)_i^{(e)} = \left(x^{\mathrm{prec}(l,i,e))}\right)_i^{(e)}$. Now we consider the case when $\overline{E}^{(l,i,e)'}(\left(x^{\mathrm{prec}(l,i,e))}\right)_i^{(e)}) \neq 0$. By an argument similar to that used in the proof of [38, Theorem 2], from Lemma 2.7.1 applied to the interval whose endpoints are $\left(x^{(l,i,e))}\right)_i^{(e)}$ and $\left(x^{\mathrm{prec}(l,i,e))}\right)_i^{(e)}$, we find a non-negative real number μ with

$$\begin{aligned} \overline{E}^{(l,i,e)}(\left(x^{\mathrm{prec}(l,i,e))}\right)_i^{(e)}) - \overline{E}^{(l,i,e)}(\left(x^{(l,i,e))}\right)_i^{(e)}) &= \overline{E}^{(l,i,e)'}(\left(x^{\mathrm{prec}(l,i,e))}\right)_i^{(e)})(\left(x^{\mathrm{prec}(l,i,e))}\right)_i^{(e)} - \left(x^{(l,i,e))}\right)_i^{(e)}) \\ &- \frac{\mu}{2}\left(\left(x^{(l,i,e))}\right)_i^{(e)} - \left(x^{\mathrm{prec}(l,i,e))}\right)_i^{(e)}\right)^2. \end{aligned} \tag{2.70}$$

From (2.66) we get

$$\left(x^{(l,i,e))}\right)_i^{(e)} - \left(x^{\mathrm{prec}(l,i,e))}\right)_i^{(e)} = \left(\left(x^{(l,i,e))}\right)_i^{(e)} - \left(x^{\mathrm{prec}(l,i,e))}\right)_i^{(e)}\right)^2 \frac{T}{\omega\, \overline{E}^{(l,i,e)'}(\left(x^{\mathrm{prec}(l,i,e))}\right)_i^{(e)})}. \tag{2.71}$$

From (2.70) and (2.71) we obtain

$$\begin{aligned} 0 &< \overline{E}^{(l,i,e)}(\left(x^{\mathrm{prec}(l,i,e))}\right)_i^{(e)}) - \overline{E}^{(l,i,e)}(\left(x^{(l,i,e))}\right)_i^{(e)}) = \frac{T}{\omega}\left(\left(x^{(l,i,e))}\right)_i^{(e)} - \left(x^{\mathrm{prec}(l,i,e))}\right)_i^{(e)}\right)^2 \\ &- \frac{\mu}{2}\left(\left(x^{(l,i,e))}\right)_i^{(e)} - \left(x^{\mathrm{prec}(l,i,e))}\right)_i^{(e)}\right)^2 = \\ &= \left(\frac{T}{\omega} - \frac{\mu}{2}\right)\left(\left(x^{(l,i,e))}\right)_i^{(e)} - \left(x^{\mathrm{prec}(l,i,e))}\right)_i^{(e)}\right)^2. \end{aligned} \tag{2.72}$$

As $0 < \omega < 2$ and $0 \leq \mu < T$, we get

$$\frac{T}{\omega} - \frac{\mu}{2} \geq T\left(\frac{1}{\omega} - \frac{1}{2}\right) > 0. \tag{2.73}$$

From (2.72) and (2.73) we obtain

$$0 \leq ((x^{(l,i,e))})_i^{(e)} - (x^{\mathrm{prec}(l,i,e))})_i^{(e)})^2 \leq \frac{2\,\omega}{T\,(2-\omega)}\,(\overline{E}^{(l,i,e)}((x^{\mathrm{prec}(l,i,e))})_i^{(e)}) - \overline{E}^{(l,i,e)}((x^{(l,i,e))})_i^{(e)})) \quad (2.74)$$

Note that (2.74) holds also when $\overline{E}^{(l,i,e)'}((x^{\mathrm{prec}(l,i,e))})_i^{(e)}) = 0$. Thus, in both cases (2.58) and (2.60), from (2.74) and the convergence of the sequence $\{\overline{E}^{(l,i,e)}((x^{(l,i,e))})_i^{(e)})\}$, $l \in \mathbb{N}$, $i = 1,2,\ldots,nm$, $e \in \{r,g,b\}$, it follows that the sequence $\{(\overline{E}^{(l,i,e)}((x^{\mathrm{prec}(l,i,e))})_i^{(e)}) - \overline{E}^{(l,i,e)}((x^{(l,i,e))})_i^{(e)}))\}$, $l \in \mathbb{N}$, $i = 1,2,\ldots,nm$, $e \in \{r,g,b\}$, converges to 0. From this it follows that

$$\lim_{(l,i,e)} (\,(x^{(l,i,e))})_i^{(e)} - (x^{\mathrm{prec}(l,i,e))})_i^{(e)}) = 0, \quad (2.75)$$

getting the claim.

By (2.54), we get

$$(\mathbf{x}^{(l,i,e)})_j^{(q)} - (x^{\mathrm{prec}(l,i,e)})_j^{(q)} = -\frac{\omega}{T}\,\frac{\partial E^{(0)}\left(x^{\mathrm{prec}(l,i,e)}\right)}{\partial x_i^{(e)}}. \quad (2.76)$$

By arbitrariness of $i \in \{1,2,\ldots,nm\}$ and $e \in \{r,g,b\}$, from (2.75) and (2.76) we deduce that $\lim_{(l,i,e)} \nabla E^{(0)}(x^{(l,i,e)}) = 0$, that is the assertion. □

2.8 Componentwise convexity of the first approximation

Now we prove that the first approximation is componentwise convex.

Theorem 2.8.1. When $p = 0$, the function $E^{(p)}$ in (2.14) is componentwise convex.

Proof. We recall that $\overline{\varphi}$ is componentwise convex on $\mathbb{R}^2$ with respect to t_1 and t_2. Fix $k \in \{1,2,3\}$ and $c \in C_k$, and choose $\Xi_c^k \in \{N_c^k, V_c^k\}$. Now we claim that the function $\mathbf{x} \mapsto \overline{\varphi}(\Xi_c^k \mathbf{x}, \Xi_{\pi_k(c)}\mathbf{x})$ is componentwise convex with respect to the components of $\mathbf{x} \in \mathbb{R}^{3nm}$. Indeed, fix $x_{i,j}^{(e)}$ with $i \in \{1,2,\ldots,n\}$, $j \in \{1,2,\ldots,m\}$ and $e \in \{r,g,b\}$.

Now we prove the convexity of $\overline{\varphi}$ with respect to the variable $x_{i,j}^{(e)}$, in the following three cases:

I) $(i,j) \notin c \cup \pi_k(c)$;

II) $(i,j) \in c$;

III) $(i,j) \in \pi_k(c)$.

We observe that it is impossible that $(i,j) \in c \cap \pi_k(c)$, thanks to our definition of $\pi_k(c)$.

Fix arbitrarily $u, w \in \mathbb{R}$ and $t \in [0,1]$.

In case I), note that $\Xi_c^k \mathbf{x}$ and $\Xi_{\pi_k(c)}^k \mathbf{x}$ are independent of the value of the variable $x_{i,j}^{(e)}$. So, we get the function $\overline{\varphi}$ evaluated in Ξ_c^k, where all pixels are fixed except $x_{i,j}^{(e)}$.

Fixed an image $\mathbf{x}$, let $\mathbf{x} \setminus x_{i,j}^{(e)} \in \mathbb{R}^{3nm-1}$ be the vector whose elements are those of $\mathbf{x}$ with the exception of $x_{i,j}^{(e)}$. Observe that the value of this pixel $x_{i,j}^{(e)}$ is an unknown variable. We have:

$$\begin{aligned} & \overline{\varphi}(\Xi_c^k(\mathbf{x} \setminus x_{i,j}^{(e)}, x_{i,j}^{(e)} = tu + (1-t)w), \Xi_{\pi_k(c)}^k(\mathbf{x} \setminus x_{i,j}^{(e)}, x_{i,j}^{(e)} = tu + (1-t)w)) = \\ = \quad & t\,\overline{\varphi}(\Xi_c^k(\mathbf{x} \setminus x_{i,j}^{(e)}, x_{i,j}^{(e)} = u), \Xi_{\pi_k(c)}^k(\mathbf{x} \setminus x_{i,j}^{(e)}, x_{i,j}^{(e)} = u)) \\ + \quad & (1-t)\overline{\varphi}(\Xi_c^k(\mathbf{x} \setminus x_{i,j}^{(e)}, x_{i,j}^{(e)} = w), \Xi_{\pi_k(c)}^k(\mathbf{x} \setminus x_{i,j}^{(e)}, x_{i,j}^{(e)} = w)), \end{aligned}$$

since

$$\Xi_c^k(\mathbf{x} \setminus x_{i,j}^{(e)}, x_{i,j}^{(e)} = a) = \Xi_c^k(\mathbf{x} \setminus x_{i,j}^{(e)}, x_{i,j}^{(e)} = b) \quad \text{and} \tag{2.77}$$

$$\Xi_{\pi_k(c)}^k(\mathbf{x} \setminus x_{i,j}^{(e)}, x_{i,j}^{(e)} = a) = \Xi_{\pi_k(c)}^k(\mathbf{x} \setminus x_{i,j}^{(e)}, x_{i,j}^{(e)} = b) \tag{2.78}$$

for each $a, b \in \mathbb{R}$.

Now we deal with the case II).

It is not difficult to see that, since the finite difference operators D_c^k are linear and the norm $\|\cdot\|_2$ is a convex function, the operators Ξ_c^k and $\Xi_{\pi_k(c)}$ are convex on their domain. We get

$$\begin{aligned} & \overline{\varphi}(\Xi_c^k(\mathbf{x} \setminus x_{i,j}^{(e)}, x_{i,j}^{(e)} = tu + (1-t)w), \Xi_{\pi_k(c)}^k(\mathbf{x} \setminus x_{i,j}^{(e)}, x_{i,j}^{(e)} = tu + (1-t)w)) \leq \\ \leq \quad & \overline{\varphi}(t\,\Xi_c^k(\mathbf{x} \setminus x_{i,j}^{(e)}, x_{i,j}^{(e)} = u) + (1-t)\Xi_c^k(\mathbf{x} \setminus x_{i,j}^{(e)}, x_{i,j}^{(e)} = w), \Xi_{\pi_k(c)}^k(\mathbf{x} \setminus x_{i,j}^{(e)}, x_{i,j}^{(e)} = tu + (1-t)w)) \leq \\ \leq \quad & t\,\overline{\varphi}(\Xi_c^k(\mathbf{x} \setminus x_{i,j}^{(e)}, x_{i,j}^{(e)} = u), \Xi_{\pi_k(c)}^k(\mathbf{x} \setminus x_{i,j}^{(e)}, x_{i,j}^{(e)} = u)) \\ + \quad & (1-t)\overline{\varphi}(\Xi_c^k(\mathbf{x} \setminus x_{i,j}^{(e)}, x_{i,j}^{(e)} = w), \Xi_{\pi_k(c)}^k(\mathbf{x} \setminus x_{i,j}^{(e)}, x_{i,j}^{(e)} = w)). \end{aligned} \tag{2.79}$$

The first inequality in (2.79) holds, since Ξ_c^k is (globally) convex. Note that $\overline{\varphi}$ is increasing in the first component. Furthermore, the third inequality in (2.79) follows from (2.78), since the function $(t_1, t_2) \mapsto \overline{\varphi}(t_1, t_2)$ is componentwise convex with respect to the variables t_1 and t_2, and since $(i,j) \notin \pi_k(c)$.

The case III) is analogous to the case II). Thus, the assertion follows.

□

Chapter 3

A blind source separation technique for document restoration

In Section 3.1 we present the features of the physical problem and in Section 3.2 we describe some of the approximated mathematical models proposed in the literature. In particular, in Subsection 3.2.1 we present the non-stationary and locally linear model we consider. In Section 3.3 we analyze the nature of this new model and formulate some constraints to reduce its ill-posedness. In Section 3.4 we develop the MATODS algorithm to deal with the local linear problem, and in Section 3.6 we discuss the NIT-MATODS algorithm for working with the new model. In Section 3.7 we compare experimentally the MATODS algorithm with other fast and unsupervised methods existing in the literature, and show how the NIT-MATODS algorithm performs in restoring real ancient documents.

3.1 The physical problem

The bleed-through is a complex physical phenomenon that involves several parameters, such as, for instance, the properties of the paper, the distribution of the paper fibers, and the quality and thickness of the ink. From a physical point of view, the bleed-through phenomenon is a diffusion process of ink through paper [95, 171]. Many similar phenomena, like the seepage of water or oil through soil, are usually described by diffusion models [141, 160]. In general, to consider

a model at micro scale is computationally unfeasible, because of the sheer number of variables that are involved, thus one typically considers an equivalent model at macroscopic scale, which describes with a good approximation the average behavior of the micro-scale phenomenon in question.

Various authors have proposed different mathematical models to approximate the physical phenomena of bleed-through and show-through when both the images of the recto and verso side of the document are available. The variational model proposed in [62] works with an estimated background, that is, the gray level of unprinted/unwritten paper. An anisotropic diffusion model is given in [61], and an invertible nonlinear model that considers the halftoning process of the printers, is considered in [6]. Other methods use just a single side observation and reduce the problem to a segmentation one (see also [164]). In [148], a physical analysis of the show-through effect produced by a scanner in a digital image of a document is performed, and a model which takes into account the reflection, transmission, and scattering parameters of the paper is developed. Although extremely simplified with respect to the physical phenomenon, this model is still quite complex, so much so that some approximations are needed to make it tractable. A generalized version of this model for bleed-through removal is discussed in [119, 145]. When the character of the side under examination is sufficiently dark, it does not change, independently of the degradation coming from the opposite side.

3.2 Approximated mathematical models

We represent a gray level image as a vector belonging to $\mathbb{R}^{n^2}$, whose elements are the light intensity (which varies between 0 and 255) of the pixels, taken in lexicographic order. We consider a document as a pair of images that represent its sides, the front (*recto*) and the back (*verso*). In particular, we denote by $\widehat{x}_r \in [0,255]^{n^2}$ the front image of the observed document, and by $\widehat{x}_v \in [0,255]^{n^2}$ the associated back image. Here, we assume that the recto data $\widehat{x}_r$ and the verso data $\widehat{x}_v$ are already spatially registered, that is, the pixel positions of the recto and of the verso of the document correspond, if we do a horizontal flip of the verso. However, the problem of registration of documents is an open and challenging issue (see for instance [21, 56, 77, 146, 173]). We denote the observed document by $\widehat{x} = \begin{bmatrix} \widehat{x}_r & \widehat{x}_v \end{bmatrix} \in [0,255]^{n^2 \times 2}$, and the source ideal document

by $\widehat{s} = \begin{bmatrix} \widehat{s_r} & \widehat{s_v} \end{bmatrix} \in [0,255]^{n^2 \times 2}$. The blind separation problem in document restoration amounts to estimating the ideal document from the observed document, without knowing the parameters underlying the back-to-front and front-to-back interference related to the model.

The nonlinear model proposed in [119, 132, 133, 145] is

$$\begin{aligned} \widehat{x_r}(i) &= \widehat{s_r}(i)\, \mathrm{e}^{q_v(1-\widehat{s_v}(i))}, \\ \widehat{x_v}(i) &= \widehat{s_v}(i)\, \mathrm{e}^{q_r(1-\widehat{s_r}(i))}, \quad i = 1,\ldots,n^2, \end{aligned} \tag{3.1}$$

where $q_v, q_b \in \mathbb{R}^+$ are the *interference levels* that affect the intensity values from the recto to the verso and vice versa, respectively. Some nonlinear models that assume that the interference levels depend on the location are proposed in [71, 104, 157]. This assumption makes the model non-stationary, that is, not translation invariant. In particular the nonlinear model proposed in [157] is

$$\begin{aligned} \widehat{x_r}(i) &= \widehat{s_r}(i) \left(\frac{\widehat{s_v}(i))}{255} \right)^{q_v(i)}, \\ \widehat{x_v}(i) &= \widehat{s_v}(i) \left(\frac{\widehat{s_r}(i))}{255} \right)^{q_r(i)}, \quad i = 1,\ldots,n^2. \end{aligned} \tag{3.2}$$

The hypothesis of non-stationarity is significant, as in real ancient documents the level of interference varies highly from pixel to pixel. Figure 3.1 shows a detail of an ancient document, where the verso has been horizontally mirrored for the reader's convenience. From this figure it is evident that an ink infiltration law independent of the position cannot be determined in general. The algorithms proposed in [71, 157] for the resolution of the related inverse problem are fast

Figure 3.1: Detail of the document in Figure 3.19 with a horizontally flipped verso.

heuristics. In order to obtain more precise results, a computationally more expensive regularized problem should be investigated (see also [69, 155]).

3.2.1 A non-stationary locally linear model

In this work, in order to obtain an accurate algorithm with low computational cost, we propose a new non-stationary and locally linear model. Namely, we partition the domain of the document, a set of pixels of size $n \times n$, into $(n/\nu)^2$ disjoint subdomains of size $\nu \times \nu$. On each subdomain we approximate the problem by means of the the classical linear model (see also [49, 98, 97, 99, 154, 156])

$$\widehat{x}^T = A\widehat{s}^T, \tag{3.3}$$

where the symbol $\cdot^T$ denotes the transpose operator of a matrix, $\widehat{x} \in [0,255]^{\nu^2}$ is the observed document in the involved subdomain, $\widehat{s} \in [0,255]^{\nu^2}$ is the ideal document, and $A \in \mathbb{R}^{2\times 2}$ is called *mixture matrix*. We assume that the entries of the matrix A vary smoothly with respect to the corresponding entries in the mixture matrices in the adjacent subdomains. The size of the subdomains should be chosen taking into account both the accuracy of the model (for small subdomain dimensions) and the computational cost for its resolution (for large dimensions).

In the next sections we focus on the linear problem related to the equation (3.3), while in Section 3.6 we use the results obtained for the resolution of the linear problem to solve the non-stationary model proposed here.

3.3 Analysis of the linear problem

In this section we discuss the problem of estimating both the ideal sources and the mixture matrix from the observed data using the linear equation (3.3), which is a BSS problem (see also [49, 155]). If we have an invertible estimate $\widetilde{A}$ of A, then an estimate of s is

$$\widetilde{s}^T = \widetilde{A}^{-1}\widehat{x}^T. \tag{3.4}$$

Since there are infinitely many choices of $\widetilde{A}$, our problem admits infinitely many solutions, and is ill-posed in the sense of Hadamard. Even if we assume that $\widetilde{A}$ and $\widetilde{s}$ are nonnegative matrices, the problem is NP-hard (see [159]) and ill-posed (see [75]). To overcome this, it is necessary to impose some constraints on the solutions.

Since the color of the paper is the same for each part of the document, we assume that the value of the source background, that is the graylevel of unprinted/unwritten paper, is the same as

the background of the data. This value corresponds to the light intensity of the paper on which the document is written. In order to satisfy this requirement, we assume that A is a *one row-sum matrix*, that is,

$$a_{11} + a_{12} = a_{21} + a_{22} = 1. \tag{3.5}$$

In [26] we prove that if $(\widetilde{A}, \widetilde{s})$ is a solution to the linear model in equation (3.3), with $\widetilde{A}$ nonsingular and such that $\widetilde{a}_{22} \neq \widetilde{a}_{12}$ and $\widetilde{a}_{11} \neq \widetilde{a}_{21}$, then there exist $t_1, t_2 \neq 0$ and a one row-sum matrix $\overline{A}$, such that $(\overline{A}, \overline{s})$ is a solution of (3.3), with $\overline{s} = \begin{bmatrix} t_1\widetilde{s}_r & t_2\widetilde{s}_v \end{bmatrix}$. In other words, for a given estimation of the solution, it is sufficient to multiply the estimated sources by some given nonzero parameters in order to obtain a solution with a one row-sum estimated mixture matrix. It is easy to see that if $(\widetilde{A}, \widetilde{s})$ is a solution to the linear model in equation (3.3), and $\widetilde{A}$ is singular, then there exists $t \in \mathbb{R}$ such that $x_v = tx_r$. In Subsection 3.5.11 we show how to find a more realistic solution with a one row-sum mixing matrix in this case. When $(\widetilde{A}, \widetilde{s})$ is a solution with $\widetilde{a}_{11} = \widetilde{a}_{21}$ or $\widetilde{a}_{12} = \widetilde{a}_{22}$. If $\widetilde{a}_{11} = \widetilde{a}_{21}$ or $\widetilde{a}_{12} = \widetilde{a}_{22}$, then one of the estimated sources usually corresponds to the common background of the recto and the verso, which is a pattern that is equally present on the two sides of the document. An example of this case is shown in Figure 3.2. Therefore, requiring that the mixture matrix is one row-sum is not a restriction.

(a) Observed document. (b) Solution with $\widetilde{a}_{11} = \widetilde{a}_{21}$. (c) Solution with a one row-sum mixture matrix.

Figure 3.2: One row-sum mixture matrix reconstruction.

A remarkable feature of our approach is that a high light intensity indicates the presence of meaningful information (for example, a letter or a figure), whereas a low light intensity corresponds to the absence thereof. Since the background is usually lighter in color while text or figures are darker, we apply the change of variables

$$x = mE - \widehat{x}, \qquad s = mE - \widehat{s}, \tag{3.6}$$

where $E \in \mathbb{R}^{n^2 \times 2}$ is the matrix such that

$$e_{i,j} = 1 \text{ for each } i = 1,2,\ldots,n^2 \text{ and } j = 1,2, \tag{3.7}$$

and m is the maximum between the light intensity on the two sides of the document. Note that, since we deal with paper documents, we assume that this maximum is achieved on the background. In view of (3.6), the values of the light intensity corresponding to the background are equal to 0, while the pixels containing information have positive light intensity values no greater than m. Motivated by the physical interpretation of these values, we impose that the estimated sources have to satisfy this property, as the values zero and m correspond to the background color and the black color, respectively. Since A is a one row-sum matrix, we get

$$E^T = AE^T, \tag{3.8}$$

and hence from (3.3) and (3.8) we obtain

$$x^T = mE^T - A\hat{s}^T = mAE^T - A\hat{s}^T = A(mE^T - \hat{s}^T) = As^T. \tag{3.9}$$

Here we define the following 2×2 *data overlapping matrix* of the observed data.

$$C = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} = x^T x = \begin{bmatrix} x_r^T \cdot x_r & x_r^T \cdot x_v \\ x_v^T \cdot x_r & x_v^T \cdot x_v \end{bmatrix}. \tag{3.10}$$

This matrix, when x_r and x_v have zero mean, corresponds to the data covariance matrix. The matrix C tells how much the text on the front overlaps with that on the back. Indeed in our case, since x is nonnegative, the data overlapping matrix is always nonnegative, and is diagonal if and only if there is no overlapping text from the recto to the verso of the document. In particular we refer to the entries $d = c_{12} = c_{21}$ as the *data overlapping level*.

The *source overlapping matrix* can be defined similarly as

$$P = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} = s^T s = \begin{bmatrix} s_r^T \cdot s_r & s_r^T \cdot s_v \\ s_v^T \cdot s_r & s_v^T \cdot s_v \end{bmatrix}.$$

It is easy to see that the matrices C and P are symmetric and positive semidefinite. We refer to the value

$$k = p_{12} = p_{21} = s_r^T \cdot s_v \tag{3.11}$$

as the *source overlapping level*. Since we assume that the text of the recto of the document partially overlaps with that on the verso, the estimation of the level k plays an important role in

the design of the technique we propose. In fact, we claim that a correct estimation of k leads to more accurate estimates of the original sources.

3.4 A new technique for solving the linear problem

We consider the two cases of singular and nonsingular data overlapping matrices separately. Now we treat the latter, while the former will be dealt with in Subsection 3.5.11.

We would like to estimate not only the ideal sources s_r and s_v and the mixture matrix A, but also the source overlapping level k. Since in our algorithm we impose a non-negativity constraint on the estimated sources $\widetilde{s}_r$ and $\widetilde{s}_v$, the corresponding value of k represents the level of overlapping of the recto of the source document with its verso or, equivalently, the portion of text of the estimated front source that is disjoint from that of the estimated back source. The value of k is different from zero, in general, thus the method we propose can be classified as a *Correlated Component Analysis* (CCA) technique (see also [14, 139, 152, 153]).

We define a *symmetric factorization* of a symmetric and positive definite matrix $H \in \mathbb{R}^{n\times n}$ as an identity of the type $H = ZZ^T$, where $Z \in \mathbb{R}^{n\times n}$. Observe that, given an orthogonal matrix $Q \in \mathbb{R}^{n\times n}$ and a symmetric factorization of the type $H = ZZ^T$, then $ZQ(ZQ)^T$ is also a symmetric factorization of H. Moreover, if we consider any two symmetric factorizations $H = Z_1Z_1^T$ and $H = Z_2Z_2^T$, then there is an orthogonal matrix $Q \in \mathbb{R}^{n\times n}$ such that $Z_1 = Z_2Q$.

In the 2×2 case, the set of the orthogonal matrices is the union of all rotations and reflections in $\mathbb{R}^2$, which are expressed as

$$Q^1(\theta) = \begin{bmatrix} \sin\theta & -\cos\theta \\ \cos\theta & \sin\theta \end{bmatrix} \quad \text{and} \quad Q^{-1}(\theta) = \begin{bmatrix} \sin\theta & \cos\theta \\ \cos\theta & -\sin\theta \end{bmatrix}, \tag{3.12}$$

respectively, as θ varies in $]0, 2\pi]$. As $C = C^{1/2}(C^{1/2})^T = C^{1/2}C^{1/2}$ is a symmetric factorization of C, then all possible factorizations of C are given by

$$Z^{(\iota)}(\theta) = C^{1/2}Q^{(\iota)}(\theta) = \begin{bmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{bmatrix} Q^{(\iota)}(\theta) = \begin{bmatrix} z_{11}^{(\iota)}(\theta) & z_{12}^{(\iota)}(\theta) \\ z_{21}^{(\iota)}(\theta) & z_{22}^{(\iota)}(\theta) \end{bmatrix}, \tag{3.13}$$

where $\theta \in]0, 2\pi]$ and $\iota \in \{-1, 1\}$. In particular, we have that

$$z_{11}^{(1)}(\theta) = z_{11}^{(-1)}(\theta), \quad z_{12}^{(1)}(\theta) = -z_{12}^{(-1)}(\theta), \quad z_{21}^{(1)}(\theta) = z_{21}^{(-1)}(\theta), \quad z_{22}^{(1)}(\theta) = -z_{22}^{(-1)}(\theta). \tag{3.14}$$

In order to obtain a joint estimation of the mixture matrix, the source matrices, and the source overlapping level, we use an iterative algorithm. At the lth step we assume that

$$C = x^T x = A s^T s A^T = A \widetilde{P} A^T, \tag{3.15}$$

where $\widetilde{P}$ is a symmetric and positive definite estimate of the source overlapping matrix P. In $\widetilde{P}$ we set

$$\widetilde{p}_{12} = \widetilde{p}_{21} = k^{(l)}, \tag{3.16}$$

where $k^{(l)}$ is the estimate of the source overlapping level obtained at the $(l-1)$th step (we assume that $k^{(0)} = 0$). Note that for the moment we do not assign a value to $\widetilde{p}_{11}$ and $\widetilde{p}_{22}$, as they will be determined later by imposing that the estimated mixture matrix is one row-sum. Let

$$\widetilde{P} = YY^T \tag{3.17}$$

be a symmetric factorization, where Y is a nonsingular matrix that by (4.12) satisfies

$$y_{11} y_{21} + y_{12} y_{22} = k^{(l)}. \tag{3.18}$$

By virtue of (4.11) and (4.13), it holds that

$$C \quad = \quad AYY^T A^T = AY(AY)^T,$$

that is, AY realizes a factorization of C. For any given choice of $\theta \in]0, 2\pi]$ and $\iota \in \{-1, 1\}$, we define an estimate $\widetilde{A}^{(\iota)}(\theta)$ of the mixture matrix A as a matrix such that $\widetilde{A}^{(\iota)}(\theta) = Z^{(\iota)}(\theta) Y^{-1}$, where $Z^{(\iota)}(\theta)$ is as in (4.9). We get that

$$\begin{aligned} a_{11}^{(\iota)}(\theta) &= \frac{z_{11}^{(\iota)}(\theta) y_{22} - z_{12}^{(\iota)}(\theta) y_{21}}{y_{11} y_{22} - y_{21} y_{12}}, \qquad a_{12}^{(\iota)}(\theta) = \frac{z_{12}^{(\iota)}(\theta) y_{11} - z_{11}^{(\iota)}(\theta) y_{12}}{y_{11} y_{22} - y_{21} y_{12}}, \\ a_{21}^{(\iota)}(\theta) &= \frac{z_{21}^{(\iota)}(\theta) y_{22} - z_{22}^{(\iota)}(\theta) y_{21}}{y_{11} y_{22} - y_{21} y_{12}}, \qquad a_{22}^{(\iota)}(\theta) = \frac{z_{22}^{(\iota)}(\theta) y_{11} - z_{21}^{(\iota)}(\theta) y_{12}}{y_{11} y_{22} - y_{21} y_{12}}, \end{aligned} \tag{3.19}$$

and by imposing that $\widetilde{A}^{(\iota)}(\theta)$ satisfies the one row-sum condition in equation (3.5), we have that

$$\begin{aligned} z_{11}^{(\iota)}(\theta) y_{22} - z_{12}^{(\iota)}(\theta) y_{21} + z_{12}^{(\iota)}(\theta) y_{11} - z_{11}^{(\iota)}(\theta) y_{12} &= y_{11} y_{22} - y_{21} y_{12}, \\ z_{21}^{(\iota)}(\theta) y_{22} - z_{22}^{(\iota)}(\theta) y_{21} + z_{22}^{(\iota)}(\theta) y_{11} - z_{21}^{(\iota)}(\theta) y_{12} &= y_{11} y_{22} - y_{21} y_{12}. \end{aligned} \tag{3.20}$$

Thus, the matrix Y has to satisfy the conditions in equations (4.14) and (4.16). The nonlinear system given by the equations (4.14) and (4.16) has infinitely many solutions. For the sake of

convenience, we choose the solution

$$y_{11} = \frac{\det C - k^{(l)}(z_{11}^{(\iota)}(\theta) - z_{21}^{(\iota)}(\theta))^2}{(z_{22}^{(\iota)}(\theta) - z_{12}^{(\iota)}(\theta))\det Z^{(\iota)}(\theta)}, \quad y_{12} = k^{(l)}\frac{z_{11}^{(\iota)}(\theta) - z_{21}^{(\iota)}(\theta)}{\det Z^{(\iota)}(\theta)}, \tag{3.21}$$

$$y_{21} = 0, \qquad y_{22} = \frac{\det Z^{(\iota)}(\theta)}{z_{11}^{(\iota)}(\theta) - z_{21}^{(\iota)}(\theta)}.$$

This choice has several desirable consequences. First, from equations (4.10) and (4.15) we get that $\widetilde{A}^{(1)}(\theta) = \widetilde{A}^{(-1)}(\theta)$ for all $\theta \in]0, 2\pi]$. Moreover, from equations (4.8) and (4.9) we deduce that $Z(\theta) = -Z(\theta + \pi)$, for $\theta \in]0, \pi]$, thus from equations (4.15) and (4.17) we can conclude that

$$\widetilde{A}(\theta) = \widetilde{A}(\theta + \pi), \tag{3.22}$$

for all $\theta \in]0, \pi]$.

Therefore, in the reminder we consider only the case $\iota = 1$, pose $\widetilde{A}(\theta) = \widetilde{A}^{(1)}(\theta)$ and $Z(\theta) = Z^{(1)}(\theta)$ for all $\theta \in]0, \pi]$, and in general consider only the values of θ belonging to the interval $]0, \pi]$.

Recall that Y must be non-singular, as Y realizes a symmetric factorization of the non-singular matrix P. It is not difficult to see that, if

$$k^{(l)} < k_{sup} = \frac{\det C}{(\rho_{11} - \rho_{21})^2 + (\rho_{12} - \rho_{22})^2}, \tag{3.23}$$

where $\rho_{i,j}$, for $i, j = 1, 2$, are the entries of the matrix $C^{1/2}$ in the equation (4.9), then Y is non-singular for all $\theta \in]0, \pi]$. We refer to k_{sup} in the equation (3.23) as the *source overlapping level upper bound*.

Moreover, the equations in (4.17) are well defined if $z_{11}(\theta) \neq z_{21}(\theta)$ and $z_{12}(\theta) \neq z_{22}(\theta)$. It easy to see that $z_{11}(\theta) = z_{21}(\theta)$ or $z_{12}(\theta) = z_{22}(\theta)$ when θ assumes the values $\varphi + t\frac{\pi}{2}$, with $t \in \mathbb{Z}$ and

$$\varphi = \begin{cases} \arctan\left(\dfrac{\rho_{22} - \rho_{12}}{\rho_{11} - \rho_{21}}\right), & \text{if } \rho_{11} \neq \rho_{21}, \\ \dfrac{\pi}{2}, & \text{if } \rho_{11} = \rho_{21} \end{cases}. \tag{3.24}$$

In Subsection 3.5.2, in formulating the minimization algorithm, we show how to avoid these values.

For any $\theta \in]\varphi, \varphi + \frac{\pi}{2}[\cup]\varphi + \frac{\pi}{2}, \varphi + \pi[$, from equation (3.9) we deduce that an estimate of the ideal sources s is given by

$$\widetilde{s}(\theta)^T = \begin{bmatrix} \widetilde{s}_r(\theta) & \widetilde{s}_v(\theta) \end{bmatrix}^T = \widetilde{A}^{-1}(\theta)x^T, \tag{3.25}$$

which combined with the fact that $\widetilde{A}^{-1}(\theta) = \widetilde{A}^{1}(\theta) = Z^{(1)}(\theta)Y^{-1}$ and (4.16), gives

$$\begin{aligned}\widetilde{s}_r(\theta) &= \left(z_{22}(\theta)\frac{\det C - k^{(l)}(z_{11}(\theta)-z_{21}(\theta))^2}{(z_{22}(\theta)-z_{12}(\theta))\det C} - z_{21}(\theta)\frac{k^{(l)}(z_{11}(\theta)-z_{21}(\theta))}{\det C}\right)x_r + \\ &\quad \left(-z_{12}(\theta)\frac{\det C - k^{(l)}(z_{11}(\theta)-z_{21}(\theta))^2}{(z_{22}(\theta)-z_{12}(\theta))\det C} + z_{11}(\theta)\frac{k^{(l)}(z_{11}(\theta)-z_{21}(\theta))}{\det C}\right)x_v, \quad (3.26)\end{aligned}$$

$$\widetilde{s}_v(\theta) = -\frac{z_{21}(\theta)}{z_{11}(\theta)-z_{21}(\theta)}x_r + \frac{z_{11}(\theta)}{z_{11}(\theta)-z_{21}(\theta)}x_v.$$

As we assumed that our estimated sources have intensity values between 0 and m, we take the orthogonal projection of the estimate $s_i^{(l)}(\theta)$ on the space $[0,m]^{\nu^2\times 2}$ with respect to the Frobenius norm. Namely, we apply to the estimate of the sources the function that maps a vector $s \in \mathbb{R}^{\nu^2}$ to the ν^2-dimensional vector $\tau(s)$, whose elements are

$$(\tau(s))_i = \begin{cases} 0, & \text{if } s_i \le 0, \\ s_i, & \text{if } 0 < s_i \le m, \\ m, & \text{if } s_i > m, \end{cases} \qquad i = 1,\ldots,\nu^2. \quad (3.27)$$

By this transformation, the projections of the estimated source images $\tau(\widetilde{s}_{r,i}^{(l)}(\theta))$ and $\tau(\widetilde{s}_{v,i}^{(l)}(\theta))$ are guaranteed to be nonnegative (see also [42, 50, 74, 134]). From now on, we consider the projections above as the new source estimates. Thus, the estimated source overlapping level is a nonnegative value, and it is zero if and only if there is no overlapping text from the recto to the verso of the estimated source document. Hence, among the possible values of θ in $]\varphi,\varphi+\frac{\pi}{2}[\cup]\varphi+\frac{\pi}{2},\varphi+\pi[$, we find a value $\widetilde{\theta}$ that minimizes the *objective function*

$$g(k^{(l)},\theta,C) = \tau(\widetilde{s}_r(\theta))^T \cdot \tau(\widetilde{s}_v(\theta)). \quad (3.28)$$

Note that, from equations (4.18) and (4.20), it follows that the function g is periodic in the variable θ with period π. Then we set

$$k^{(l+1)} = g(k^{(l)},\widetilde{\theta},C), \quad (3.29)$$

and we repeat this process until we find an index l such that $k^{(l+1)} = k^{(l)}$. It easy to see that if

$$\tau(\widetilde{s}_r(\widetilde{\theta})) = \widetilde{s}_r(\widetilde{\theta}) \qquad \text{and} \qquad \tau(\widetilde{s}_v(\widetilde{\theta})) = \widetilde{s}_v(\widetilde{\theta}), \quad (3.30)$$

then the condition (3.29) holds. In this case the estimated solution $\widetilde{s}(\widetilde{\theta})$ belongs to the space $[0,m]^{\nu^2\times 2}$, as required. We note that in all the experiments we performed, when a fixed point was reached the condition (3.30) was always satisfied.

The steps of the algorithm described in this section can be summarized as follows.

function MATODS$(\widehat{x})$

Determine the maximum value m of $\widehat{x}$;

$x = mE - \widehat{x}$;

$C = x^T x$;

$k^{(-1)} = -2\varepsilon$;

$k^{(0)} = 0$;

$l = 0$;

while $(|k^{(l)} - k^{(l-1)}| \geq \varepsilon)$ **do**

$\widetilde{\theta} = \text{argmin}(\textbf{function } g(k^{(l)}, \cdot, C))$;

$k^{(l+1)} = g(k^{(l)}, \widetilde{\theta}, C)$;

$l = l + 1$;

end while

$Z(\widetilde{\theta}) = C^{1/2} Q_1(\widetilde{\theta})$;

Compute $\widetilde{s_r}(\widetilde{\theta})$ and $\widetilde{s_v}(\widetilde{\theta})$ as in (4.21);

return $mE - \tau(\widetilde{s}(\widetilde{\theta}))$

Here ε is a fixed positive real number that represents a suitable tolerance threshold, while the function $g(\cdot,\cdot,\cdot)$ is computed as follows.

function $g(k, \theta, C)$

$Z(\theta) = C^{1/2} Q^1(\theta)$;

Compute $\widetilde{s_r}(\theta)$ and $\widetilde{s_v}(\theta)$ as in (4.21);

return $(\tau(\widetilde{s_r}(\theta)))^T \cdot \tau(\widetilde{s_v}(\theta))$

In the next subsection we describe the procedure we use to minimize the objective function g with respect to the variable θ. We refer to this method as the MATODS algorithm, which is a parameter-free, and thus unsupervised, technique.

3.5 The objective function minimization algorithms

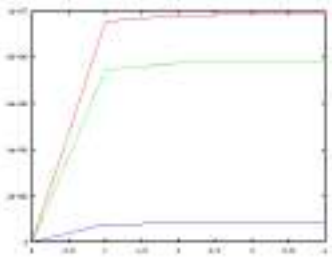
In this section we study the problem of finding the minimum of the objective function $g(k,\cdot,\iota,C)$ (see (3.28)), for $\iota \in \{1,-1\}$ and for a positive definite matrix $C \in \mathbb{R}^{2\times 2}$. We minimize the functions $g(k,\cdot,1,C)$ and $g(k,\cdot,-1,C)$, and pose $\iota^{(l)} = 1$ if $\min_{\theta\in[0,2\pi]} g(k,\cdot,1,C) \leq \min_{\theta\in[0,2\pi]} g(k,\cdot,-1,C)$, and $\iota^{(l)} = -1$ otherwise. We start by analyzing a stochastic technique that assures the convergence to the minimum in probability.

3.5.1 Local quasi-convexity of the objective function

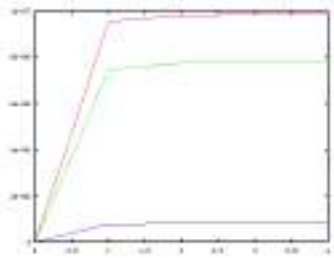
Here we analyze experimentally the trend of the objective function $g(k,\cdot,\iota,C)$ to be minimized, for fixed $k \geq 0$, $\iota \in \{1,-1\}$ and $C \in \mathbb{R}^{2\times 2}$ definite positive matrix. First, we observe that $g(k,\cdot,\iota,C)$ is a periodic function with period π. Indeed, from (4.8) we have $Q_\iota(\theta+\pi) = -Q_\iota(\theta)$, $\iota \in \{1,-1\}$. Then, from (4.9) we get $Z_{R,\iota}(\theta+\pi) = -Z_{R,\iota}(\theta)$, $\iota \in \{1,-1\}$. Finally, from (4.21), we obtain that the equation related to the estimated sources $\widetilde{s}^{(l)}_{R,\iota}(\theta+\pi) = \widetilde{s}^{(l)}_{R,\iota}(\theta)$ holds for $\iota \in \{1,-1\}$ and for every positive definite matrix $C_R \in \mathbb{R}^{2\times 2}$. In Figures 3.4–3.9 we present some examples of graphs of the function $g(k,\cdot,\iota,C)$. In order to obtain such graphs, we take the following mixing matrices

$$A_R = \begin{pmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{pmatrix}, A_G = \begin{pmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{pmatrix}, A_B = \begin{pmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{pmatrix},$$

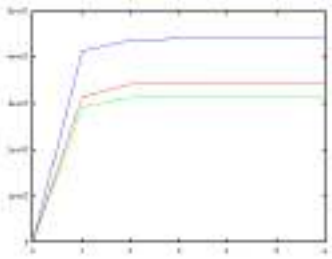
and consider as original sources the images in Figures 3.15–3.16. Then, by (3.3) we construct the observed data and the related overlapping matrix C_R, C_G, and C_B. Recalling that the value of k is estimated independently on each of the three channels, we saw experimentally that, during the execution of the MATODS algorithm, the value of k is always increasing, as is shown in Figure 3.3.



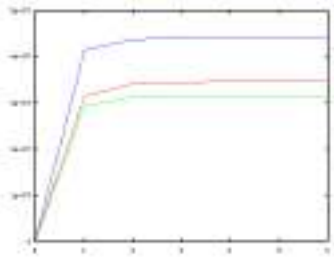
(a) Document in Figure 3.15 (a), $\iota = 1$.



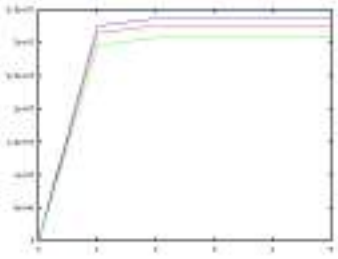
(b) Document in Figure 3.15 (b), $\iota = 1$.



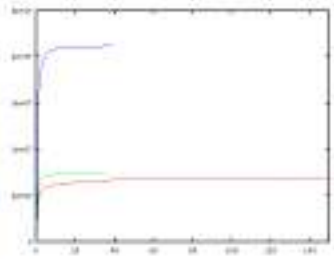
(c) Document in Figure 3.15 (b), $\iota = 1$.



(d) Document in Figure 3.15 (b), $\iota = -1$.



(e) Document in Figure 3.15 (c), $\iota = 1$.



(f) Document in Figure 3.15 (d), $\iota = 1$.

Figure 3.3: Trend of $k_R^{(l)}$, $k_G^{(l)}$ and $k_B^{(l)}$ during the execution of MATODS

In Figure 3.4, we deal with the document in Figure 3.15 (a), where $\iota = 1$ is fixed. We recall that, in order to assume that the system (4.17) is well-defined, we have to impose that (see (3.23))

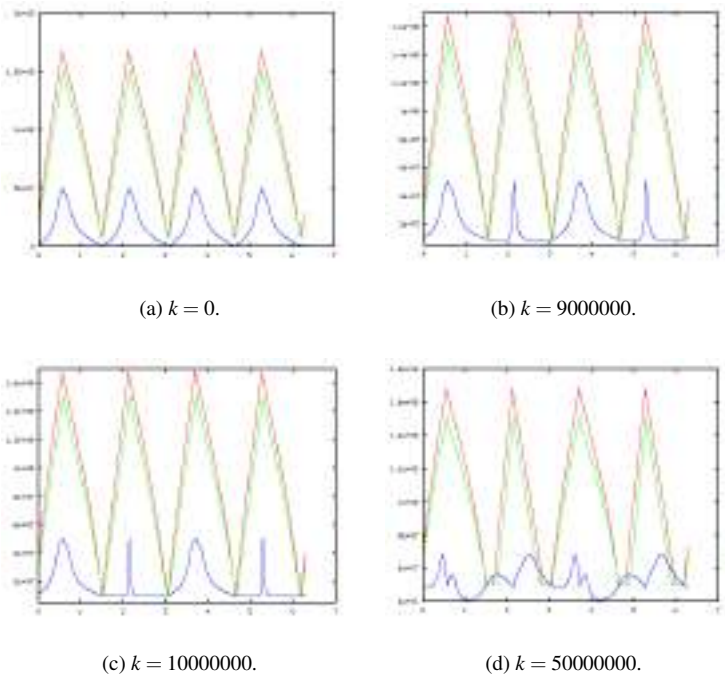
$$k \geq \frac{\det(C_R)}{\left(\bar{c}_{11}^R - \bar{c}_{21}^R\right)^2 + \left(\bar{c}_{12}^R - \bar{c}_{22}^R\right)^2} = k_{sup}^R,$$

$$k \geq \frac{\det(C_G)}{\left(\bar{c}_{11}^G - \bar{c}_{21}^G\right)^2 + \left(\bar{c}_{12}^G - \bar{c}_{22}^G\right)^2} = k_{sup}^G,$$

$$k \geq \frac{\det(C_B)}{\left(\bar{c}_{11}^B - \bar{c}_{21}^B\right)^2 + \left(\bar{c}_{12}^B - \bar{c}_{22}^B\right)^2} = k_{sup}^B.$$

In Figure 3.3 (a) we see that k, in the three RGB channels, converges monotonically to the source overlapping levels k_R, k_G and k_B, respectively. In this case, we have $k_R = 9855291 < k_{sup}^R = 132751132.62$, $k_G = 7753236 < k_{sup}^G = 105650226.17$, $k_B = 834224 < k_{sup}^B = 11122735.89$.

Indeed, the values source overlapping level upper bounds k_{sup}^R, k_{sup}^G and k_{sup}^B are much closer to data overlapping levels $d_R = 139503525.96$, $d_G = 108090739.20$, $d_B = 11444930.24$. Since the source overlapping levels are in general much smaller than the respective data overlapping levels, we can assume that, during the execution of the MATODS algorithm in the three channels, the value of k is always smaller than k_{sup}^R, k_{sup}^G or k_{sup}^B, respectively.



(a) $k = 0$. (b) $k = 9000000$.

(c) $k = 10000000$. (d) $k = 50000000$.

Figure 3.4: Graphs of the objective functions $g(k,\cdot,1,C_R)$, $g(k,\cdot,1,C_G)$ and $g(k,\cdot,1,C_B)$ in correspondence with the document in Figure 3.15 (a).

In Figure 3.4 the values of k are the following: $k = 0$, that is the MATODS source overlapping level initial value for all three channels; $k = 9000000$, which is near to the red ideal source overlapping level k_R; $k = 10000000$, which is close but smaller than the green source overlapping level upper bound k_{sup}^G; $k = 50000000$, which is greater than all source overlapping level upper bounds.

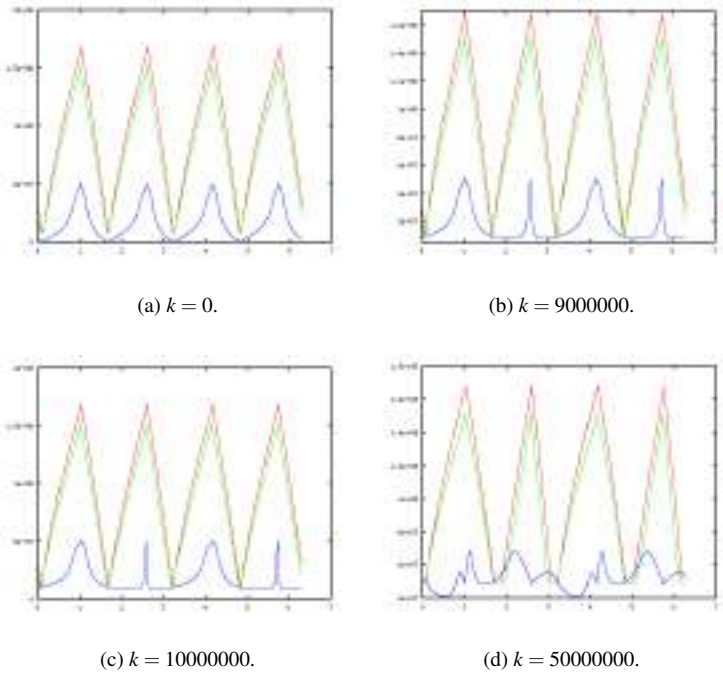
We observe that, in this case, the points of discontinuity of the objective functions $g(k,\cdot,1,C_R)$, $g(k,\cdot,1,C_G)$ and $g(k,\cdot,1,C_B)$, for all k smaller than their source overlapping level upper bounds, are $\varphi_R^{(1)} = 0.53873315$, $\varphi_R^{(2)} = 3.68032580$, $\varphi_R^{(5)} = 5.25112213$ and $\varphi_R^{(6)} = 2.10952948$, for the red channel, $\varphi_G^{(1)} = 0.57955014$, $\varphi_G^{(2)} = 3.72114279$, $\varphi_G^{(5)} = 5.29193912$ and $\varphi_G^{(6)} = 2.15034646$, for the green channel, $\varphi_B^{(1)} = 0.57021981$, $\varphi_B^{(2)} = 3.71181247$, $\varphi_B^{(5)} = 5.28260880$

and $\varphi_B^{(6)} = 2.14101614$, for the blue channel. In Figures 3.4 we note that, when k is smaller than the source overlapping level upper bounds then for all three channels the objective functions $g(k,\cdot,1,C_R)$, $g(k,\cdot,1,C_G)$ and $g(k,\cdot,1,C_B)$ turn to be quasi-convex in the intervals included between any two successive points of discontinuity. We recall that a function $f:[a,b]\rightarrow\mathbb{R}$ is *quasi-convex* iff

$$f((1-\alpha)\theta_1+\alpha\,\theta_2)\leq\max\{f(\theta_1),f(\theta_2)\},$$

for each $\alpha\in[0,1]$ and θ_1, $\theta_2\in[a,b]$ with $\theta_1\neq\theta_2$. A function $f:[a,b]\subset\mathbb{R}\rightarrow\mathbb{R}$ is said to be *weakly unimodal* iff there exists a value $\widehat{\theta}$, for which it is weakly monotonically increasing for $\theta\in[a,\widehat{\theta}]$ and weakly monotonically decreasing for $\theta\in[\widehat{\theta},b]$. A function $f:[a,b]\subset\mathbb{R}\rightarrow\mathbb{R}$ is *quasi-convex* in the convex and compact set $[a,b]\subset S$ iff it is weakly unimodal. When $f:S\subset\mathbb{R}^n\rightarrow\mathbb{R}$, similar definitions can be done. In this case quasi-convex functions are weakly unimodal functions, but not all the weakly unimodal functions are quasi-convex (see also [8, 103]).

Concerning Figure 3.5, we consider again the document in Figure 3.15 (a), choose $\iota=-1$ and take the same values of k. In this case, the values of the points of discontinuity of the objective functions $g(k,\cdot,-1,C_R)$, $g(k,\cdot,-1,C_G)$ and $g(k,\cdot,-1,C_B)$ are given by $\varphi_R^{(3)}=1.03206318$, $\varphi_R^{(4)}=4.17365583$, $\varphi_R^{(7)}=5.74445216$ and $\varphi_R^{(8)}=2.60285950$ for the red channel, $\varphi_G^{(3)}=0.99124619$, $\varphi_G^{(4)}=4.13283884$, $\varphi_G^{(7)}=5.70363517$ and $\varphi_G^{(8)}=2.56204252$ for the green channel, $\varphi_B^{(3)}=1.00057651$, $\varphi_B^{(4)}=4.14216917$, $\varphi_B^{(7)}=5.71296549$ and $\varphi_B^{(8)}=2.57137284$ for the blue channel. Such values are the unique ones which differ from those of the previous case. In Figure 3.5 we note that, when k is smaller than the upper bounds, the objective functions $g(k,\cdot,-1,C_R)$, $g(k,\cdot,-1,C_G)$ and $g(k,\cdot,-1,C_B)$ are quasi-convex on each interval which lies between any two successive points of discontinuity.



(a) $k = 0$. (b) $k = 9000000$.

(c) $k = 10000000$. (d) $k = 50000000$.

Figure 3.5: Graphs of the objective functions $g(k,\cdot,-1,C_R)$, $g(k,\cdot,-1,C_G)$ and $g(k,\cdot,-1,C_B)$ in correspondence with the document in Figure 3.15 (a).

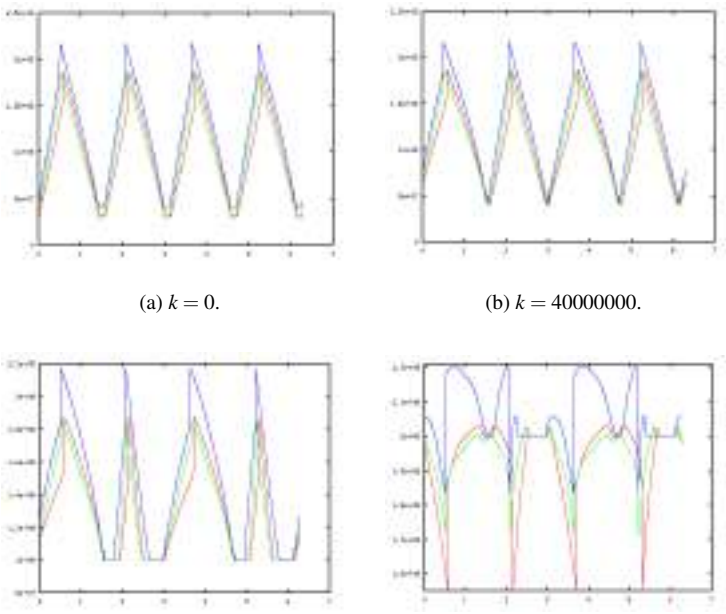
In Figure 3.6, we take the document in Figure 3.15 (b) and choose $\iota = 1$. Also in this case, it is

$$k_R = 34612679 < k_{sup}^R = 131593024.23 < d_R = 136166103.08,$$

$$k_G = 31495751 < k_{sup}^G = 130553408.73 < d_G = 141445576.28,$$

$$k_B = 44013514 < k_{sup}^B = 157271106.68 < d_B = 172518952.96.$$

The discontinuity of the objective functions $g(k,\cdot,1,C_R)$, $g(k,\cdot,1,C_G)$ and $g(k,\cdot,1,C_B)$ are given by $\varphi_R^{(1)} = 0.56948411$, $\varphi_R^{(2)} = 3.71107676$, $\varphi_R^{(5)} = 5.28187309$ and $\varphi_R^{(6)} = 2.14028043$ for the red channel, $\varphi_G^{(1)} = 0.50318323$, $\varphi_G^{(2)} = 3.64477588$, $\varphi_G^{(5)} = 5.215572207$ and $\varphi_G^{(6)} = 2.07397955$, for the green channel, and $\varphi_B^{(1)} = 0.48885097$, $\varphi_B^{(2)} = 3.63044362$, $\varphi_B^{(5)} = 5.20123995$ and $\varphi_B^{(6)} = 2.05964730$ for the blue channel. We choose $k = 0$, because it is the MATODS initial estimate, $k = 40000000$, since it is near to all ideal source overlapping levels, $k = 100000000$, as it is close, but inferior, to all source overlapping level upper bounds, and $k = 200000000$, because it is beyond these upper bounds. In Figure 3.6 we note that, when k is smaller than its source overlapping level upper bound, the objective functions $g(k,\cdot,1,C_R)$, $g(k,\cdot,1,C_G)$ and $g(k,\cdot,1,C_B)$ are quasi-convex on each interval which lies between any two successive points of discontinuity.

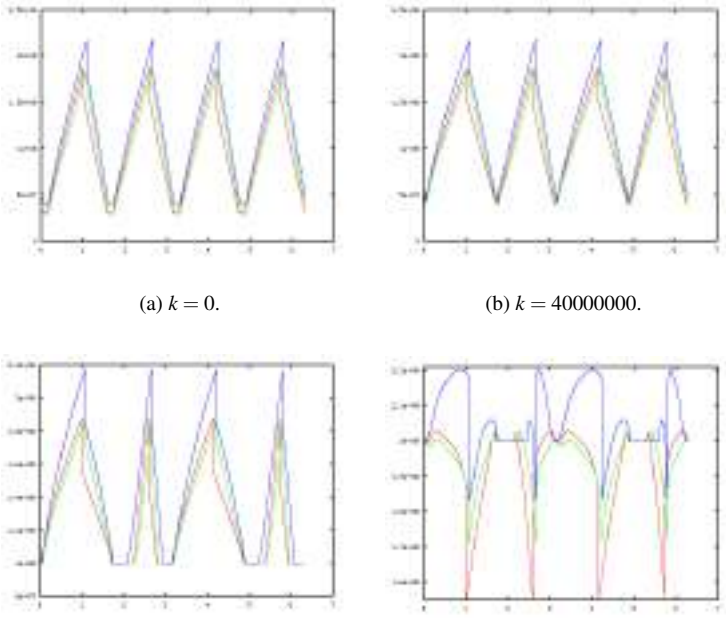


(a) $k = 0$. (b) $k = 40000000$.

(c) $k = 100000000$. (d) $k = 200000000$.

Figure 3.6: Graphs of the objective functions $g(k,\cdot,1,C_R)$, $g(k,\cdot,1,C_G)$ and $g(k,\cdot,1,C_B)$ in correspondence with the document in Figure 3.15 (b).

In Figure 3.7, we consider the document in Figure 3.15 (b) again, but we take $\iota = -1$ and use the same values of k. In this case, the values of the points of discontinuity of the objective functions $g(k,\cdot,-1,C_R)$, $g(k,\cdot,-1,C_G)$ and $g(k,\cdot,-1,C_B)$ are given by $\varphi_R^{(3)} = 1.00131222$, $\varphi_R^{(4)} = 4.14290487$, $\varphi_R^{(7)} = 5.71370120$ and $\varphi_R^{(8)} = 2.57210855$, for the red channel, $\varphi_G^{(3)} = 1.06761310$, $\varphi_G^{(4)} = 4.20920575$, $\varphi_G^{(7)} = 5.78000208$ and $\varphi_G^{(8)} = 2.63840943$ for the green channel, $\varphi_B^{(3)} = 1.08194536$, $\varphi_B^{(4)} = 4.22353801$, $\varphi_B^{(7)} = 5.79433434$ and $\varphi_B^{(8)} = 2.65274169$, for the blue channel. Note that, in Figure 3.7, when k is smaller than its upper bound, for all three channels the objective functions $g(k,\cdot,-1,C_R)$, $g(k,\cdot,-1,C_G)$ and $g(k,\cdot,-1,C_B)$ are quasi-convex on each interval which lies between any two successive points of discontinuity.

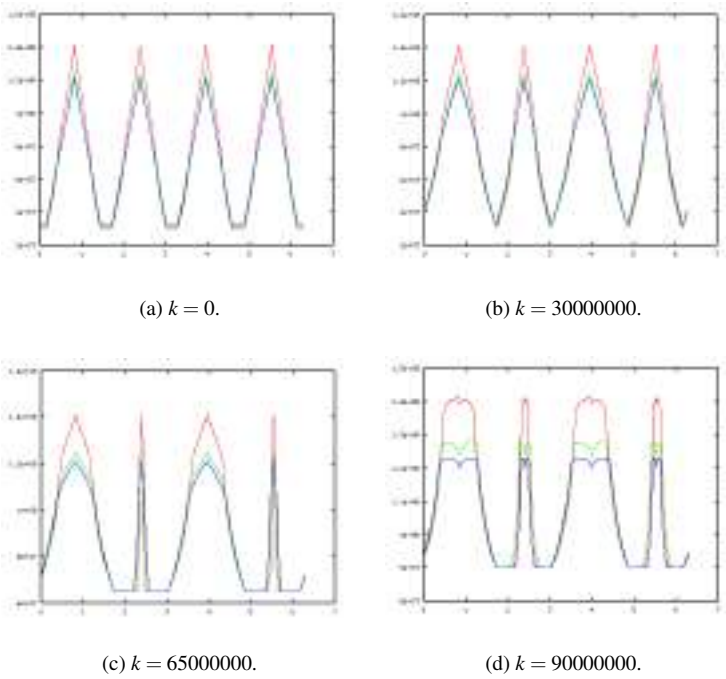


(a) $k = 0$. (b) $k = 40000000$.

(c) $k = 100000000$. (d) $k = 200000000$.

Figure 3.7: Graphs of the objective functions $g(k,\cdot,-1,C_R)$, $g(k,\cdot,-1,C_G)$ and $g(k,\cdot,-1,C_B)$ in correspondence with the document in Figure 3.15 (b).

From now on, since the graphs obtained with $\iota = 1$ and $\iota = -1$ are very similar, we consider only the case $\iota = 1$. Concerning the graphs in Figure 3.8, we take the document in Figure 3.15 (c). In this case we have the inequalities $k_R = 32685410 < d_R = 72365832.56 < k_{sup}^R = 73936335.04$, $k_G = 30815153 < d_G = 68222469.08 < k_{sup}^G = 69702847.74$, $k_B = 33805612 < d_B = 74981471.44 < k_{sup}^B = 76611523.60$. The discontinuity of the objective functions $g(k,\cdot,1,C_R)$, $g(k,\cdot,1,C_G)$ and $g(k,\cdot,1,C_B)$ are $\varphi_R^{(1)} = 0.81450982$, $\varphi_R^{(2)} = 3.95610247$, $\varphi_R^{(5)} = 5.52689880$ and $\varphi_R^{(6)} = 2.38530615$, for the red channel, $\varphi_G^{(1)} = 0.81453870$, $\varphi_G^{(2)} = 3.95613135$, $\varphi_G^{(5)} = 5.52692768$ and $\varphi_G^{(6)} = 2.38533503$, for the green channel, $\varphi_B^{(1)} = 0.81446681$, $\varphi_B^{(2)} = 3.95605946$, $\varphi_B^{(5)} = 5.52685579$ and $\varphi_B^{(6)} = 2.38526314$, for the blue channel. Here, we choose $k = 0$, that is the initial value, $k = 30000000$ which is close to all ideal solutions, $k = 65000000$ which is inferior but near to all upper bounds and $k = 90000000$ which is higher than all upper bounds. Finally in Figure 3.8, when k is smaller than its upper bound, the objective functions $g(k,\cdot,1,C_R)$, $g(k,\cdot,1,C_G)$ and $g(k,\cdot,1,C_B)$ are quasi-convex on each interval which lies between any two successive points of discontinuity.

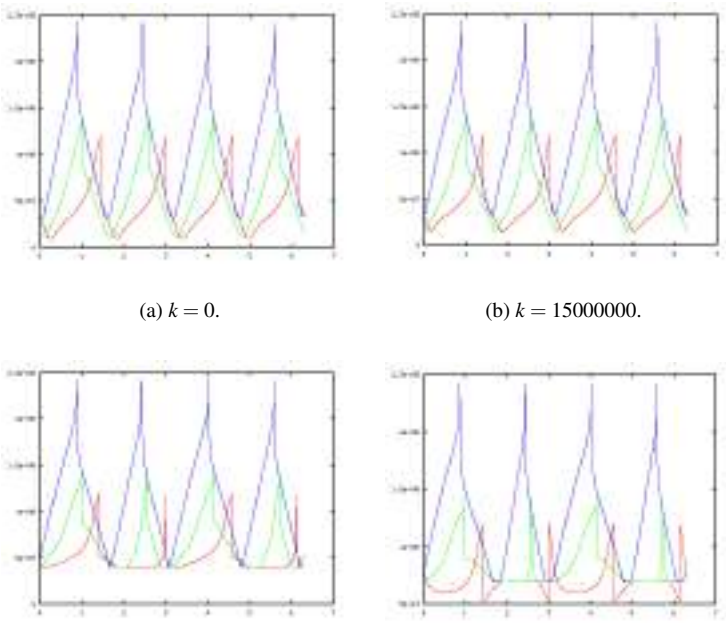


(a) $k = 0$. (b) $k = 30000000$.

(c) $k = 65000000$. (d) $k = 90000000$.

Figure 3.8: Graphs of the objective functions $g(k,\cdot,1,C_R)$, $g(k,\cdot,1,C_G)$ and $g(k,\cdot,1,C_B)$ in correspondence with the document in Figure 3.15 (c).

The Figure 3.9 is obtained by considering the document in Figure 3.16. Here we get $k_R = 15812614 < k_{sup}^R = 44683913.34 < d_R = 79303165.36$, $k_G = 14928144 < k_{sup}^G = 64082928.34 < d_G = 65712248.40$, $k_B = 78431743 < d_B = 144848191.56 < k_{sup}^B = 147729606.55$. Thus, we choose to show the graphs for $k = 0$, $k = 15000000$, $k = 40000000$ and $k = 70000000$. The points of discontinuity of the objective functions $g(k,\cdot,1,C_R)$, $g(k,\cdot,1,C_G)$ and $g(k,\cdot,1,C_B)$ are $\varphi_R^{(1)} = 1.41814710$, $\varphi_R^{(2)} = 4.55973976$, $\varphi_R^{(5)} = 6.13053609$ and $\varphi_R^{(6)} = 2.98894343$ for the red channel, $\varphi_G^{(1)} = 0.98719285$, $\varphi_G^{(2)} = 4.12878550$, $\varphi_G^{(5)} = 5.69958183$ and $\varphi_G^{(6)} = 2.55798917$ for the green channel, $\varphi_B^{(1)} = 0.85077026$, $\varphi_B^{(2)} = 3.99236291$, $\varphi_B^{(5)} = 5.56315924$ and $\varphi_B^{(6)} = 2.42156658$ for the blue channel. In Figure 3.9 we note again that, when k is smaller than its upper bound, for all three channels the objective functions are quasi-convex on every interval which lies between any two successive points of discontinuity.



(a) $k = 0$. (b) $k = 15000000$.

(c) $k = 40000000$. (d) $k = 70000000$.

Figure 3.9: Graphs of the objective functions $g(k,\cdot,1,C_R)$, $g(k,\cdot,1,C_G)$ and $g(k,\cdot,1,C_B)$ in correspondence with the document in Figure 3.16.

Thus, for all graphs in Figures 3.4–3.9, when k is smaller than its upper bound (which is always true during the execution of the MATODS algorithm) the objective functions $g(k,\cdot,\imath,C_R)$, $g(k,\cdot,\imath,C_G)$ and $g(k,\cdot,\imath,C_B)$ are quasi-convex on each interval which lies between any two successive points of discontinuity. Moreover the values of the local minima, on each interval where an objective function is quasi-convex, are almost identical. Thus, to find the minimum of an objective function, it is sufficient to minimize it in an interval which lies between any two successive points of discontinuity, where the involved function is quasi-convex. In our experiments similar results were obtained also by choosing any mixing matrix different from those chosen in (4.24).

In the sequel we give some different algorithms, which can be used to find the minimum in an interval in which the involved function is quasi-convex. Successively, we compare the obtained results, to establish the algorithm to use. In order to compare the convergence speed of such algorithms, we recall that the sequence $\{\theta^{(h)}\}_h$ converges to $\widehat{\theta}$ with *strong order* p and *asymptotic costant* $\gamma > 0$ if and only if

$$\lim_{h\to+\infty} \frac{|\theta^{(h+1)} - \widehat{\theta}|}{|\theta^{(h)} - \widehat{\theta}|^p} = \gamma.$$

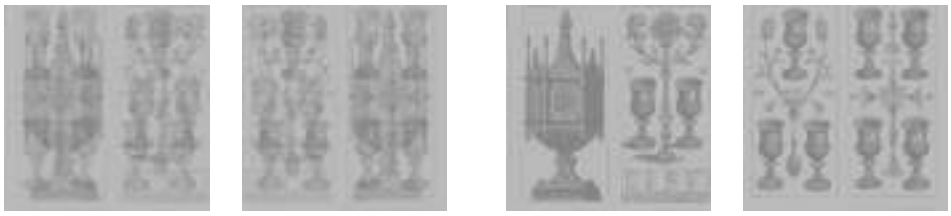
When $p = 1$, the asymptotic costant γ is also called *convergence factor*. We say that the sequence $\{\theta^{(h)}\}_h$ converges to $\widehat{\theta}$ with *weak order* p if and only if

$$\liminf_{h\to+\infty}(-\ln|\theta^{(h)} - \widehat{\theta}|^p)^{1/h} = p.$$

Note that strong convergence implies weak convergence, but in general the converse does not hold.

3.5.2 The objective function minimization algorithm

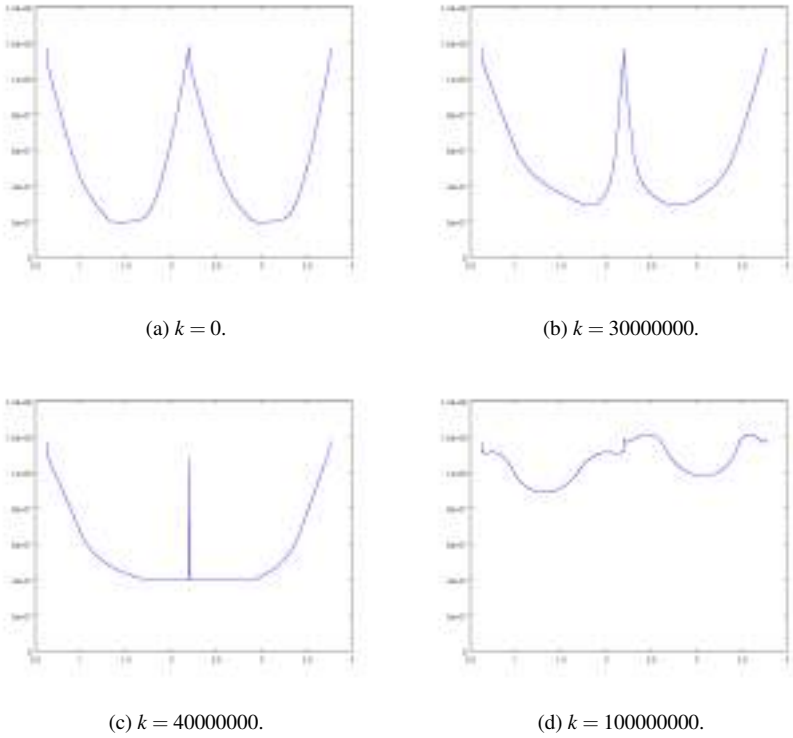
Consider the data document shown in Figure 3.10 (a), for which the source overlapping level upper bound is $k_{sup} = 40374184.63$ (see equation (3.23)) and the objective function g has discontinuities with respect to the variable θ at the points $\varphi + t\frac{\pi}{2}$, with $t \in \mathbb{Z}$ and $\varphi = 0.62377$. Figure 3.11 shows the graph of the function $g(k,\theta,C)$ as θ varies. In the plots, we use the the overlapping matrix of the document in Figure 3.10 (a), and test four values of k. The output of the MATODS algorithm is presented in Figure 3.10 (b). The source overlapping level estimated by MATODS is 29670911.87, which is smaller than k_{sup}. It easy to show that if $k = 0$, then the objective function g is periodic of period $\frac{\pi}{2}$ in the variable θ, but this property is not verified for $k > 0$, as shown in Figures 3.11 (b)-(d).



(a) Observed document. (b) Document estimated by MATODS.

Figure 3.10: MATODS restoration.

Notice that, when k is smaller than the source overlapping level upper bound k_{sup}, the objective function g is quasi-convex on each interval which lies between any two successive points of discontinuity. Moreover, on each interval where the objective function is quasi-convex, the local minima are almost identical. It easy to see that this behavior is typical among objective functions obtained from the documents we considered in our experiments. we have also seen that during the execution of the MATODS algorithm, the estimated source overlapping levels $k^{(l)}$



(a) $k = 0$. (b) $k = 30000000$.

(c) $k = 40000000$. (d) $k = 100000000$.

Figure 3.11: Graphs of the objective function $g(k,\cdot,C)$, where C is the overlapping matrix given by the document in Figure 3.10 (a).

have increasing values, and are always smaller than their source overlapping level upper bound k_{sup}. Thus, to find the minimum of the objective function, it is sufficient to find a minimum on an interval that lies between any two successive points of discontinuity, where the objective function is quasi-convex.

In order to minimize g on an interval where it is quasi-convex, we consider different algorithms. Some of them are developed specifically for strictly quasi-convex functions and do not rely on derivatives (see also [37, 100, 106]), whereas others are based on the gradient descent and the *Armijo Line Search* (ALS) (see also [9, 33]).

3.5.3 The simulated annealing

The simulated annealing techniques have the aim to define a sequence, which converges to the global minimum of a function, not necessarily convex (see also [66]). However, since it is dealt with an asymptotic behavior, in general it is not possible to assure the convergence to the mini-

mum after a finite number of steps.

To apply the annealing technique, for each temperature T_h, where $h \in \mathbb{N}$ and $\lim_{h\to+\infty} T_h = 0$, we use the Metropolis Sampler, in order to update the variable θ (see also [125, 163]).

$\theta_j^{(h)}$, $j = 0, \ldots, L_h$ of estimates of θ is constructed.

Given $\theta_j^{(h)}$, at the step $j+1$ the proposed $\theta_{j+1}^{(h)}$ is given by $\theta_j^{(h)} + \nu$, where ν is a random variable, having uniform distribution in the interval $(-\delta, \delta)$, with given $\delta \in \mathbb{R}^+$. Hence at the step $j+1$, as a new estimate of θ we choose either $\theta_j^{(h)}$ or $\theta_{j+1}^{(h)}$.

Let $\Delta g = g(k, \theta_j^{(h)}, C) - g(k, \theta_{j+1}^{(h)}, C)$. We accept $\theta_{j+1}^{(h)}$ when $\Delta g > 0$ or with probability $e^{\frac{\Delta g}{T_k}}$ when $\Delta g \leq 0$. By iterating, for every $h \in \mathbb{N}$ it is possible to construct a Markov chain $\theta_j^{(h)}$, $j = 0, 1, 2, \ldots$, convergent in L^2 and in probability to an equilibrium state having probability

$$\pi^{(h)}(\theta) = \frac{e^{-\frac{g(k,\theta,C)}{T_h}}}{\int_0^{2\pi} e^{-\frac{g(k,\theta,C)}{T_h}} d\theta},$$

fixed k, ι and C, where the involved integral is intended in the discrete sense (see also [163, Theorem 8.2.2 (a)]). As h tends to $+\infty$, if

$$T_h \geq \frac{\Delta}{\ln h}, \tag{3.31}$$

where Δ denotes the *maximal local increase* of $g(k, \cdot, C)$ (see also [163]), then the stationary probability distribution of the Markov chain converges in probability to the set of the global minima of $g(k, \cdot, C)$ (see also [163, Theorem 8.2.3]).

In the practical cases, it is impossible to obtain asymptotic results, and furthermore the assumption (3.31) it is not advisable in terms of computational times, and thus one has to establish: an initial value of the temperature T_0; the number of steps of the Metropolis technique, that is the length L_h of the involved Markov chain; a suitable function which expresses the decay of the temperature; a stop criterion.

The initial temperature T_0 must be sufficiently high, in order to accept the variations of configurations with high probability. In correspondence with the temperature T_0, let $\chi(T_0) = A(T_0)/P(T_0)$, where $A(T_0)$ and $P(T_0)$ are the numbers of the accepted and proposed transitions, respectively, at the temperature T_0. Successively, we impose $\chi(T_0) \simeq 1$. Let n_1 (resp. n_2) the number of the decreasing (resp. increasing) transitions in corrispondence with the temperature T_0. Observe that $n_1 + n_2 = L_0$, where L_0 is the length of the Markov chain associated with the

temperature T_0. Let us denote by $\langle \Delta g \rangle^+$ the mean value of Δg associated with the transition which increases the energy. We assume the following approximation:

$$\chi(T_0) = \frac{n_1 + n_2 e^{-\frac{\langle \Delta g \rangle^+}{T_0}}}{n_1 + n_2},$$

obtaining

$$T_0 = \frac{\langle \Delta g \rangle^+}{\ln\left(\frac{n_2}{n_2 \chi(T_0) - n_1(1-\chi(T_0))}\right)}. \tag{3.32}$$

In order to estimate T_0 by means of (3.32), we can compute experimentally n_1, n_2 and $\langle \Delta g \rangle^+$, where $\chi(T_0)$ is a suitable positive constant close to 1.

As mentioned before, to obtain convergence of the global minimum of the function $g(k,\cdot,C)$, it is necessary to have a logarithmic decay of the temperature. Anyway, to get good results, it is possible to suppose to have a linear decay, namely $T_{h+1} = \gamma T_h$, where γ is a suitable real constant, which in general is taken between 0.95 and 0.99 (see also [1, 163]). At the last step, we establish that the stop criterion is as follows: when the values of the estimated θ remain constant after a complete Markov chain, then we stop.

The simulated annealing algorithm can be expressed as follows:

function SA(k,C)

$h = 0$;

$\theta_1^{(0)} = 0$;

$\theta_1^{(-1)} = \theta_1^{(0)} + 2\varepsilon$;

while $(|\theta_1^{(h)} - \theta_1^{(h-1)}| > \varepsilon)$ **do**

for j=1 to $L_h - 1$ **do**

$\theta_{j+1}^{(h)} = \theta_j^{(h)} + random(-\delta, \delta)$;

$\Delta g = g(k, \theta_j^{(h)}, C) - g(k, \theta_{j+1}^{(h)}, C)$;

if $((\Delta g \leq 0)$ and $(random(0,1) > e^{\frac{\Delta g}{T_k}}))$ **then**

$\theta_{j+1}^{(h)} = \theta_j^{(h)}$;

end if

end for

$\theta_1^{(h+1)} = \theta_{L_h}^{(h)}$;

$T_{h+1} = \gamma T_h$;

$h = h + 1$;

end while

return $\theta_1^{(h)}$

where ε is a suitable tolerance threshold. We refer to this algorithm as *Simulated Annealing* (SA).

3.5.4 The three point search

In this section we describe an algorithm to minimize the objective function $g(k,\cdot,C)$ in one of the intervals in which it is supposed to be quasi-convex. Given a generic step of length p_h, we consider the vector

$$\psi^{(h)} = \left[\theta^{(h)} - p_h \qquad \theta^{(h)} \qquad \theta^{(h)} + p_h\right].$$

We now denote the corresponding values of the objective function by

$$\xi^{(h)} = \left[g(k,\psi_1^{(h)},C) \qquad g(k,\psi_2^{(h)},C) \qquad g(k,\psi_3^{(h)},C)\right].$$

Supposed that in the interval $[a,b]$ the function $g(k,\cdot,C)$ is quasi-convex, we apply the following algorithm.

function TPS(k,C_R,a,b)

$h = 0$;

$\psi^{(0)} = [a \quad (a+b)/2 \quad b]$;

$\xi^{(0)} = [g(k,\psi_1^{(0)},C) \quad g(k,\psi_2^{(0)},C) \quad g(k,\psi_3^{(0)},C)]$;

$p_0 = (b-a)/2$;

if $(\xi_1^{(0)} < \xi_2^{(0)})$ **then**

while $((\xi_1^{(0)} < \xi_2^{(0)})$ and $(p_h > \varepsilon))$ **do**

$\psi_3^{(0)} = \psi_2^{(0)}$;

$\xi_3^{(0)} = \xi_2^{(0)}$;

$\psi_2^{(0)} = (\psi_1^{(0)} + \psi_3^{(0)})/2$;

$\xi_2^{(0)} = g(k,\psi_2^{(0)},C)$;

$p_0 = p_0/2$;

end while

else

while $((\xi_3^{(0)} < \xi_2^{(0)})$ and $(p_h > \varepsilon))$ **do**

$\psi_1^{(0)} = \psi_2^{(0)};$

$\xi_1^{(0)} = \xi_2^{(0)};$

$\psi_2^{(0)} = (\psi_1^{(0)} + \psi_3^{(0)})/2;$

$\xi_2^{(0)} = g(k, \psi_2^{(0)}, C);$

$p_0 = p_0/2;$

end while

end if

while $(p_h > \varepsilon)$ **do**

if $((\xi_2^{(h)} < \xi_1^{(h)})$ and $(\xi_2^{(h)} < \xi_3^{(h)}))$ **then**

if $(\xi_1^{(h)} < \xi_3^{(h)})$ **then**

$\psi_3^{(h+1)} = \psi_2^{(h)};$

$\xi_3^{(h+1)} = \xi_2^{(h)};$

else

$\psi_1^{(h+1)} = \psi_2^{(h)};$

$\xi_1^{(h+1)} = \xi_2^{(h)};$

end if

$\psi_2^{(h+1)} = (\psi_1^{(h+1)} + \psi_3^{(h+1)})/2;$

$\xi_2^{(h+1)} = g(k, \psi_2^{(h+1)}, C);$

$p_{h+1} = p_h/2;$

else

if $(\xi_1^{(h)} < \xi_3^{(h)})$ **then**

$\psi^{(h+1)} = [\psi_1^{(h)} - p_h \quad \psi_1^{(h)} \quad \psi_2^{(h)}];$

$\xi^{(h+1)} = [g(k, \psi_1^{(h+1)}, C) \quad \xi_1^{(h)} \quad \xi_2^{(h)}];$

else

$\psi^{(h+1)} = [\psi_2^{(h)} \quad \psi_3^{(h)} \quad \psi_3^{(h)} + p_h];$

$\xi^{(h+1)} = [\xi_2^{(h)} \quad \xi_3^{(h)} \quad g(k, \psi_3^{(h+1)}, C)];$

end if

$p_{h+1} = p_h$;

end if

$h = h + 1$;

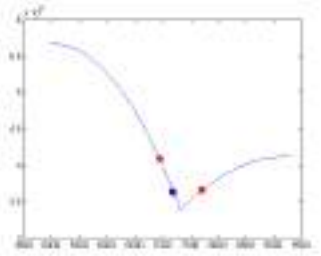
end while

return $\psi_2^{(h)}$

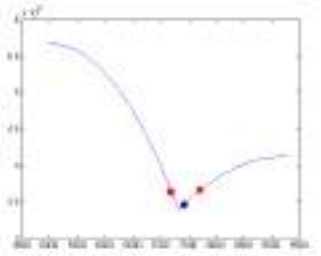
where ε is a positive real number which indicates a suitable tolerance. Such algorithm is formed by an if block and a while block. The if block is necessary to ensure that the conditions

$$\psi_1^{(0)}, \psi_2^{(0)}, \psi_3^{(0)} \in [a,b], \qquad \xi_2^{(0)} \leq \xi_1^{(0)}, \qquad \xi_2^{(0)} \leq \xi_3^{(0)} \tag{3.33}$$

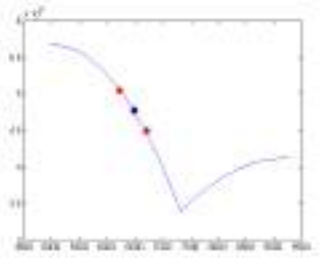
hold. The main while body has three cases. In the first one, the value of the function at the point $\psi_2^{(h)}$ is less than those evaluated at the other two nodes (see Figure 3.12 (a)). In this case, the node which assumes the greater value is removed and the intermediate point between the other two nodes is added, halving the size step $p^{(h)}$ (see Figure 3.12 (b)). In the second case, the value of the function at the node $\psi_1^{(h)}$ is greater than or equal to the one at the node $\psi_2^{(h)}$, which is greater than or equal to the one at the node $\psi_3^{(h)}$ (see Figure 3.12 (c)). In this case we eliminate the node $\psi_1^{(h)}$ and add a node to the right of $\psi_3^{(h)}$ with distance $p^{(h)}$ (see Figure 3.12 (d)). Analogously, in the third case, the value of the function at the node $\psi_3^{(h)}$ is greater than or equal to the one at the node $\psi_2^{(h)}$, which is greater than or equal to the one at the node $\psi_1^{(h)}$. So we delete the node $\psi_3^{(h)}$ and add a node to the left of $\psi_1^{(h)}$ with distance $p^{(h)}$. Since the function $g(k;\cdot,\iota,C)$ is quasi-convex, there are no other possible cases.



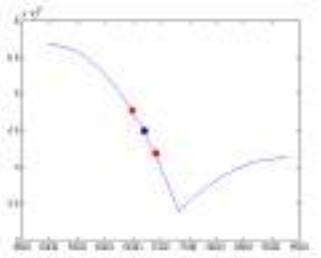
(a) First case.



(b) Successive configuration to the first case.



(c) Second case.



(d) Successive configuration to the second case.

Figure 3.12: Cases in the body of the while in the TPS algorithm.

Note that at each iteration it is necessary only one evaluation of the function $g(k,\cdot,C)$. Furthermore, observe that in the body of the while, at every step h at which the algorithm halves the length p_h, it is

$$\xi_2^{(h)} \leq \xi_1^{(h)}, \qquad \xi_2^{(h)} \leq \xi_3^{(h)}. \tag{3.34}$$

Let $\widehat{\theta}$ be a minimizer of the function $g(k,\cdot,C)$. From (3.34), since $g(k,\cdot,C)$ is quasi-convex, we have

$$\widehat{\theta} \in [\xi_1^{(h)}, \xi_2^{(h)}], \tag{3.35}$$

at every step h at which the algorithm halves the length of the step. Note that from (3.33) we deduce that the property (3.35) holds also for $h = 0$. Furthermore, observe that by the conditions (3.33), the algorithm halves the length of the step when $h = 0$.

Theorem 3.5.1. *Suppose that at the step $h-1$ the algorithm halves the length of the step p_{h-1}, then the TPS algorithm halves again the length of the step not later than the step $h+2$. That is*

$$p_{h+3} \leq \frac{1}{2} p_h \tag{3.36}$$

holds.

Proof. We suppose that at the step $h-1$ we halve the length of the step so that $p^{(h)} = p^{(h-1)}/2$, and we delete, let us say, the node $\psi_1^{(h-1)}$. Let $\widehat{\theta}$ be a minimizer of the functions $g(k,\cdot,C)$. Suppose first that $\psi_2^{(h-1)} = \psi_1^{(h)} \leq \widehat{\theta}$, then we have two cases. The first is when $\xi_2^{(h)} < \xi_1^{(h)}$. In this case, $\xi_2^{(h)}$ is the smallest value of the vector $\xi^{(h)}$, and hence the size of the step is halved at the step h. The second case is when $\xi_2^{(h)} \geq \xi_1^{(h)}$. In this case $\psi_1^{(h-1)} < \psi_1^{(h+1)} < \psi_2^{(h+1)}$, $\xi_1^{(h+1)} < \xi_2^{(h+1)}$ and $\xi_2^{(h+1)} = \xi_1^{(h)} < \xi_3^{(h+1)} = \xi_2^{(h)}$, hence the length step is halved at the step $h+1$. Now we assume that $\psi_2^{(h-1)} = \psi_1^{(h)} > \widehat{\theta}$. Then, $\xi_1^{(h)} < \xi_2^{(h)} < \xi_3^{(h)}$. So, $\psi_1^{(h+1)} = \psi_1^{(h)} - p^{(h)} = \psi_1^{(h-1)} + p^{(h)}$. If the length step is not halved at the step $h+1$, then $\xi_1^{(h+2)} = \xi_1^{(h-1)} > \xi_3^{(h+2)} = \xi_2^{(h-1)}$ and $\xi_2^{(h+2)} = \xi_1^{(h+1)} < \xi_1^{(h+2)} = \xi_2^{(h+1)}$. So, at the step $h+3$ the length step is halved. When we eliminate the node $\psi_2^{(h-1)}$ at the step $h-1$, we proceed similarly. □

The relation (3.36) can be also obtained by imposing the condition

$$\frac{p_{h+1}}{p_h} \leq \left(\frac{1}{2}\right)^{\frac{1}{3}} \simeq 0.7937,$$

so we obtain that the algorithm has a linear convergence with a factor of convergence of at least 0.7937. Note that, in the best cases, the length of the step can be halved at each step, and so a convergence factor of 0.5 is obtained. Note that, if at the step $h-1$ the algorithm halves the length of the step and if at the step $h+1$ the length of the step is not yet halved, it has to be halved at the next step. Moreover, at the $h+2$-th step, the value of the function $g(k,\cdot,C)$ to be evaluated is assumed exactly at the node deleted at the step $h-1$. Thus, the steps $h+1$ and $h+2$ can be unified using only one evaluation of the function $g(k,\cdot,C)$, by means of the following algorithm.

function TPS(k, C_R, a, b)

$h = 0$;

$\psi^{(0)} = [a \quad (a+b)/2 \quad b]$;

$\xi^{(0)} = [g(k,\psi_1^{(0)},C) \quad g(k,\psi_2^{(0)},C) \quad g(k,\psi_3^{(0)},C)]$;

$p_0 = (b-a)/2$;

if $(\xi_1^{(0)} < \xi_2^{(0)})$ **then**

while $((\xi_1^{(0)} < \xi_2^{(0)})$ and $(p_h > \varepsilon))$ **do**

$\psi_3^{(0)} = \psi_2^{(0)}$;

$\xi_3^{(0)} = \xi_2^{(0)}$;

$\psi_2^{(0)} = (\psi_1^{(0)} + \psi_3^{(0)})/2$;

$\xi_2^{(0)} = g(k, \psi_2^{(0)}, C)$;

$p_0 = p_0/2$;

end while

else

while $((\xi_3^{(0)} < \xi_2^{(0)})$ and $(p_h > \varepsilon))$ **do**

$\psi_1^{(0)} = \psi_2^{(0)}$;

$\xi_1^{(0)} = \xi_2^{(0)}$;

$\psi_2^{(0)} = (\psi_1^{(0)} + \psi_3^{(0)})/2$;

$\xi_2^{(0)} = g(k, \psi_2^{(0)}, C)$;

$p_0 = p_0/2$;

end while

end if

while $(p_h > \varepsilon)$ **do**

if $((\xi_2^{(h)} < \xi_1^{(h)})$ and $(\xi_2^{(h)} < \xi_3^{(h)}))$ **then**

if $(\xi_1^{(h)} < \xi_3^{(h)})$ **then**

$aux = \xi_3^{(h)}$;

$\psi_3^{(h+1)} = \psi_2^{(h)}$;

$\xi_3^{(h+1)} = \xi_2^{(h)}$;

$v = 0$;

else

$aux = \xi_1^{(h)}$;

$\psi_1^{(h+1)} = \psi_2^{(h)}$;

$\xi_1^{(h+1)} = \xi_2^{(h)}$;

$v = 0$;

end if

$\psi_2^{(h+1)} = (\psi_1^{(h+1)} + \psi_3^{(h+1)})/2$;

$\xi_2^{(h+1)} = g(k, \psi_2^{(h+1)}, C)$;

$p_{h+1} = p_h/2$;

else

if ($\xi_1^{(h)} < \xi_3^{(h)}$) **then**

if ($v \neq 1$) **then**

$\psi^{(h+1)} = [\psi_1^{(h)} - p_h \quad \psi_1^{(h)} \quad \psi_2^{(h)}]$;

$\xi^{(h+1)} = [g(k, \psi_1^{(h+1)}, C) \quad \xi_1^{(h)} \quad \xi_2^{(h)}]$;

$p_{h+1} = p_h$;

$v = 1$;

else

$p_{h+1} = p_h/2$;

$v = 0$;

if ($aux < \xi_2^{(h)}$) **then**

$\psi^{(h+1)} = [\psi_1^{(h)} - 2p_h \quad \psi_1^{(h)} - p_h \quad \psi_1^{(h)}]$;

$\xi^{(h+1)} = [aux \quad g(k, \psi_2^{(h+1)}, C) \quad \xi_1^{(h)}]$;

$aux = \xi_2^{(h)}$;

else

$\psi^{(h+1)} = [\psi_1^{(h)} \quad \psi_2^{(h)} - p_h \quad \psi_2^{(h)}]$;

$\xi^{(h+1)} = [\xi_1^{(h)} \quad g(k, \psi_2^{(h+1)}, C) \quad \xi_2^{(h)}]$;

end if

end if

else

if ($v \neq 1$) **then**

$\psi^{(h+1)} = [\psi_2^{(h)} \quad \psi_3^{(h)} \quad \psi_3^{(h)} + p_h]$;

$\xi^{(h+1)} = [\xi_2^{(h)} \quad \xi_3^{(h)} \quad g(k, \psi_3^{(h+1)}, C)]$;

$p_{h+1} = p_h$;

$v = 1$;

else

$p_{h+1} = p_h/2$;

$v = 0$;

if ($aux < \xi_2^{(h)}$) **then**

$\psi^{(h+1)} = [\psi_3^{(h)} \quad \psi_3^{(h)} + p_h \quad \psi_3^{(h)} + 2p_h]$;

$\xi^{(h+1)} = [\xi_3^{(h)} \quad g(k, \psi_2^{(h+1)}, C) \quad aux];$

$aux = \xi_2^{(h)};$

else

$\psi^{(h+1)} = [\psi_2^{(h)} \quad \psi_2^{(h)} + p_h \quad \psi_3^{(h)}];$

$\xi^{(h+1)} = [\xi_2^{(h)} \quad g(k, \psi_2^{(h+1)}, C) \quad \xi_3^{(h)}];$

end if

end if

end if

end if

$h = h + 1;$

end while

return $\psi_2^{(h)}$.

Thus, asymptotically we have the relation

$$p_{h+2} \leq \frac{1}{2} p_h.$$

This relation can be also obtained by imposing the condition

$$\frac{p_{h+1}}{p_h} \leq \left(\frac{1}{2}\right)^{\frac{1}{2}} \simeq 0.70711,$$

so the algorithm has a linear convergence with a factor of convergence smaller than or equal to 0.70711. We refer to this algorithm as *Three Point Search* (TPS).

3.5.5 The Golden Section Search (GSS)

In this section we present an algorithm in which the uncertainty interval is reduced by a constant factor by means of one valuation of the function $g(k, \cdot, C)$ (see also [106]). Here we consider the vector

$$\psi^{(h)} = \left[\psi_1^{(h)} \quad \psi_2^{(h)} \quad \psi_3^{(h)} \quad \psi_4^{(h)}\right]. \tag{3.37}$$

Let (a, b) be the initial uncertainty interval, containing the minimum of the function $g(k, \cdot, C)$, and let $\phi = (\sqrt{5}+1)/2$ be the *golden ratio* or *golden section*, then we apply the following algorithm:

function GSS(k, C_R, a, b)

$h = 0$;

$\psi_1^{(0)} = a$;

$\psi_4^{(0)} = b$;

$\psi_2^{(0)} = \psi_4^{(0)} - (\psi_4^{(0)} - \psi_1^{(0)})/\phi$;

$\psi_3^{(0)} = \psi_1^{(0)} + (\psi_4^{(0)} - \psi_1^{(0)})/\phi$;

while $((|\psi_4^{(h)} - \psi_1^{(h)}| > \varepsilon)$ **do**

if $(g(k, \psi_2^{(h)}, C) < g(k, \psi_3^{(h)}, C))$ **then**

$\psi^{(h+1)} = [\psi_1^{(h)} \quad \psi_4^{(h+1)} - (\psi_4^{(h+1)} - \psi_1^{(h+1)})/\phi \quad \psi_1^{(h+1)} + (\psi_4^{(h+1)} - \psi_1^{(h+1)})/\phi \quad \psi_3^{(h)}]$;

else

$\psi^{(h+1)} = [\psi_2^{(h)} \quad \psi_4^{(h+1)} - (\psi_4^{(h+1)} - \psi_1^{(h+1)})/\phi \quad \psi_1^{(h+1)} + (\psi_4^{(h+1)} - \psi_1^{(h+1)})/\phi \quad \psi_4^{(h)}]$;

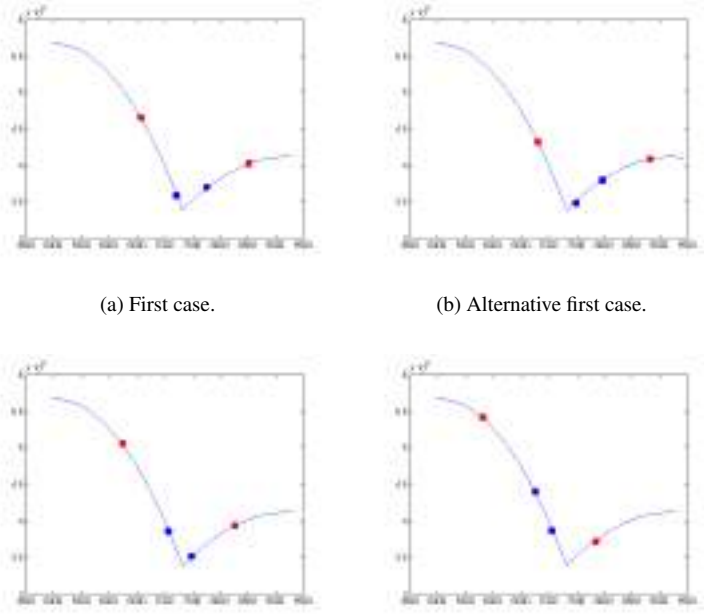
end if

$h = h + 1$;

end while

return $\psi_2^{(h)}$

where ε is a suitable tolerance threshold. In the body of the while we have two cases. In the first one, $g(k, \psi_2^{(h)}, C) < g(k, \psi_3^{(h)}, C)$ (see Figure 3.13 (a) and (b)). So the minimizer $\widehat{\theta}$ of the functions $g(k, \cdot, C)$ lies between $\psi_1^{(h)}$ and $\psi_3^{(h)}$, thus the new uncertainty interval is $[\psi_1^{(h)}, \psi_3^{(h)}]$. In the second one, $g(k, \psi_2^{(h)}, C) \geq g(k, \psi_3^{(h)}, C)$ (see Figure 3.13 (c) and (d)). Thus $\widehat{\theta}$ lies between $\psi_2^{(h)}$ and $\psi_4^{(h)}$, so the new uncertainty interval is $[\psi_1^{(h)}, \psi_3^{(h)}]$.



(a) First case. (b) Alternative first case.

(c) Second case. (d) Alternative second case.

Figure 3.13: Cases in the body of the while of the GSS algorithm.

In both cases it is

$$\psi_2^{(h)} = \psi_4^{(h)} - \frac{\psi_4^{(h)} - \psi_1^{(h)}}{\phi} \tag{3.38}$$

and

$$\psi_3^{(h)} = \psi_1^{(h)} + \frac{\psi_4^{(h)} - \psi_1^{(h)}}{\phi}. \tag{3.39}$$

Let $\ell^{(h+1)} = \psi_4^{(h+1)} - \psi_1^{(h+1)}$ be the length of the uncertainty interval at the step $h+1$. If $g(k, \psi_2^{(h)}, \iota, C) < g(k, \psi_3^{(h)}, \iota, C)$, then from the equation (3.39) we have

$$\ell^{(h+1)} = \psi_3^{(h)} - \psi_1^{(h)} = \psi_1^{(h)} + \frac{\psi_4^{(h)} - \psi_1^{(h)}}{\phi} - \psi_1^{(h)} = \frac{\psi_4^{(h)} - \psi_1^{(h)}}{\phi}, \tag{3.40}$$

while, if $g(k, \psi_2^{(h)}, C) \geq g(k, \psi_3^{(h)}, C)$, then from the equation (3.38) we get

$$\ell^{(h+1)} = \psi_4^{(h)} - \psi_2^{(h)} = \psi_4^{(h)} - \psi_4^{(h)} + \frac{\psi_4^{(h)} - \psi_1^{(h)}}{\phi} = \frac{\psi_4^{(h)} - \psi_1^{(h)}}{\phi}. \tag{3.41}$$

Thus, we obtain

$$\ell^{(h+1)} = \psi_4^{(h+1)} - \psi_1^{(h+1)} = \frac{\psi_4^{(h)} - \psi_1^{(h)}}{\phi} = \frac{\ell^{(h)}}{\phi}. \tag{3.42}$$

So, in any case, the uncertainty interval is reduced at each step by a constant factor. Now, in order to compute the factor of convergence of the method, first we have to show that at each step just one valuation of the function $g(k, \cdot, C)$ is necessary.

To prove this, we observe that the golden section has the following property:

$$\frac{1}{\phi^2} = \left(\frac{\sqrt{5}-1}{2}\right)^2 = \frac{3-\sqrt{5}}{2} = 1 - \frac{\sqrt{5}-1}{2} = 1 - \frac{1}{\phi}.$$

From this, if $g(k,\psi_2^{(h)},C) < g(k,\psi_3^{(h)},C)$ then from the equations (3.38), (3.39) and (3.40) we get

$$\begin{aligned}
\psi_3^{(h+1)} &= \psi_1^{(h+1)} + \frac{\psi_4^{(h+1)} - \psi_1^{(h+1)}}{\phi} = \psi_1^{(h)} + \frac{\psi_3^{(h)} - \psi_1^{(h)}}{\phi} \\
&= \psi_1^{(h)} + \frac{\frac{\psi_4^{(h)} - \psi_1^{(h)}}{\phi}}{\phi} = \left(1 - \frac{1}{\phi^2}\right)\psi_1^{(h)} + \frac{1}{\phi^2}\psi_4^{(h)} \\
&= \frac{1}{\phi}\psi_1^{(h)} + \left(1 - \frac{1}{\phi}\right)\psi_4^{(h)} = \psi_4^{(h)} - \frac{\psi_4^{(h)} - \psi_1^{(h)}}{\phi} \\
&= \psi_2^{(h)},
\end{aligned}$$

while, if $g(k,\psi_2^{(h)},C) \geq g(k,\psi_3^{(h)},C)$, from the equations (3.38), (3.39) and (3.41) we have

$$\begin{aligned}
\psi_2^{(h+1)} &= \psi_4^{(h+1)} - \frac{\psi_4^{(h+1)} - \psi_1^{(h+1)}}{\phi} = \psi_4^{(h)} - \frac{\psi_4^{(h)} - \psi_2^{(h)}}{\phi} \\
&= \psi_4^{(h)} - \frac{\frac{\psi_4^{(h)} - \psi_1^{(h)}}{\phi}}{\phi} = \left(1 - \frac{1}{\phi^2}\right)\psi_4^{(h)} + \frac{1}{\phi^2}\psi_1^{(h)} \\
&= \frac{1}{\phi}\psi_4^{(h)} + \left(1 - \frac{1}{\phi}\right)\psi_1^{(h)} = \psi_1^{(h)} + \frac{\psi_4^{(h)} - \psi_1^{(h)}}{\phi} \\
&= \psi_3^{(h)}.
\end{aligned}$$

Thus, if we define

$$\xi^{(h)} = \left[g(k,\psi_1^{(h)},C) \qquad g(k,\psi_2^{(h)},C) \qquad g(k,\psi_3^{(h)},C) \qquad g(k,\psi_4^{(h)},C)\right],$$

the algorithm can be written as follows:

function GSS(k, C_R, a, b)

$h = 0$;

$\psi_1^{(0)} = a$;

$\psi_4^{(0)} = b$;

$\psi_2^{(0)} = \psi_4^{(0)} - (\psi_4^{(0)} - \psi_1^{(0)})/\phi$;

$\psi_3^{(0)} = \psi_1^{(0)} + (\psi_4^{(0)} - \psi_1^{(0)})/\phi$;

$\xi^{(0)} = [g(k,\psi_1^{(0)},C) \quad g(k,\psi_2^{(0)},C) \quad g(k,\psi_3^{(0)},C) \quad g(k,\psi_4^{(0)},C)]$;

while $((|\psi_4^{(h)} - \psi_1^{(h)}| > \varepsilon)$ **do**

if $(\xi_2^{(h)} < \xi_3^{(h)})$ **then**

$$\psi^{(h+1)} = [\psi_1^{(h)} \quad \psi_4^{(h+1)} - (\psi_4^{(h+1)} - \psi_1^{(h+1)})/\phi \quad \psi_2^{(h)} \quad \psi_3^{(h)}];$$

$$\xi^{(h+1)} = [\xi_1^{(h)} \quad g(k, \psi_2^{(h+1)}, C) \quad \xi_2^{(h)} \quad \xi_3^{(h)}];$$

else

$$\psi^{(h+1)} = [\psi_2^{(h)} \quad \psi_3^{(h)} \quad \psi_1^{(h+1)} + (\psi_4^{(h+1)} - \psi_1^{(h+1)})/\phi \quad \psi_4^{(h)}];$$

$$\xi^{(h+1)} = [\xi_2^{(h)} \quad \xi_3^{(h)} \quad g(k, \psi_3^{(h+1)}, C) \quad \xi_4^{(h)}];$$

end if

$h = h + 1;$

end while

return $\psi_2^{(h)}$.

During each iterate, in every case, the function $g(k,\cdot,C)$ is evaluated only one time, so, from the equation (3.42), the algorithm has a linear convergence with factor of convergence given by

$$\frac{l^{(h+1)}}{l^{(h)}} = \frac{\psi_4^{(h+1)} - \psi_1^{(h+1)}}{\psi_4^{(h)} - \psi_1^{(h)}} = \frac{\frac{\psi_4^{(h)} - \psi_1^{(h)}}{\phi}}{\psi_4^{(h)} - \psi_1^{(h)}} = \frac{1}{\phi} = \frac{\sqrt{5}-1}{2} \simeq 0.61803.$$

3.5.6 Successive Parabolic Interpolation (SPI)

Given a finite sequence of approximations of the required minimum, the method introduced in this section constructs a parabola which interpolates the objective function $g(k,\cdot,C)$ in the last three terms of the considered sequence and add a new term to the sequence, corresponding to the argument of the minimum of the obtained parabola (see also [37, 100]). That is, given the sequence $\theta^{(0)},\ldots,\theta^{(h+2)}$, we call $p_2(\theta)$ the interpolation polynomial of the function $g(k,\cdot,C)$ at the point $\theta^{(h)}$, $\theta^{(h+1)}$, $\theta^{(h+2)}$, and choose $\theta^{(h+3)}$ by posing

$$p_2'(\theta^{(h+3)}) = 0. \tag{3.43}$$

We recall that the divided differences of the function $g(k,\cdot,C)$ are

$$g[\theta^{(h)}] = g(k, \theta^{(h)}, \iota, C), \quad h = 0, 1, \ldots,$$

and

$$g[\theta^{(h)}, \theta^{(h+1)}, \ldots, \theta^{(h+k-1)}, \theta^{(h+k)}] = \frac{g[\theta^{(h+1)}, \theta^{(h+2)}, ..., \theta^{(h+k)}] - g[\theta^{(h)}, \theta^{(h+1)} \ldots, \theta^{(h+k-1)}]}{\theta^{(h+k)} - \theta^{(h)}}$$

$$h = 0, 1, \ldots, k = 1, 2, \ldots.$$

It is possible to prove that, if $g(k,\cdot,C) \in C^2([a,b])$, where $[a,b]$ contains the argument of the minimum of $g(k,\cdot,C)$, then the sequence $\{\theta^{(h)}\}_{h\in\mathbb{N}}$ is well-defined (see also [37]). If

$$g[\theta^{(h)},\theta^{(h+1)},\theta^{(h+2)}] \neq 0,$$

then the unique solution of (3.43) is

$$\theta^{(h+3)} = \frac{1}{2}\left(\theta^{(h+1)} + \theta^{(h+2)} - \frac{g[\theta^{(h+1)},\theta^{(h+2)}]}{g[\theta^{(h)},\theta^{(h+1)},\theta^{(h+2)}]}\right). \tag{3.44}$$

Fix $n \in \mathbb{N}$, a function $f : [a,b] \to \mathbb{R}$ is said to be *of class* $LC^n([a,b])$ iff its n-th derivative exists and is Lipschitz, namely iff there exists a positive real number M_0 with

$$\sup_{x,y\in[a,b],|x-y|\leq\delta} |f(x) - f(y)| \leq M_0\,\delta$$

for each $\delta > 0$. The following result holds.

Theorem 3.5.2. (see also [37, Theorem 3.7.1]) *Let $g(k,\cdot,C) : [a,b] \to \mathbb{R}$ be of class $LC^3([a,b])$, and $\widehat{\theta} \in]a,b[$ be such that $g'(k,\widehat{\theta},C) = 0$ and $g''(k,\widehat{\theta},C) \neq 0$. If $\theta^{(0)}$, $\theta^{(1)}$, $\theta^{(2)}$ are distinct and sufficiently close to $\widehat{\theta}$, then a sequence $\{\theta^{(h)}\}_{h\in\mathbb{N}}$ is univoquely defined by (3.44), and $\{\theta^{(h)}\}_{h\in\mathbb{N}}$ either converges with strong order $p \simeq 1.325$, or converges with weak order $p = ((3+\sqrt{5})/2)^{1/3} \simeq 1.378$.*

Note that, if in the expression of $g(k,\cdot,1,C_R)$ we use the function $\bar{\tau}$ in (3.47) instead of τ in (4.22), then, using classical results of Analysis, it is not difficult to check that $g(k,\cdot,1,C_R)$ is of class $LC^3((\varphi_R^{(1)}+\eta,\varphi_R^{(6)}-\eta))$. So, Theorem 3.5.4 can be applied.

The relative algorithm is the following

function SPI(k,C_R)

$h = 0$;

$\theta^{(0)} = \varphi_R^{(1)} + \eta$;

$\theta^{(1)} = \varphi_R^{(6)} - \eta$;

$\theta^{(2)} = (\theta^{(0)} + \theta^{(1)})/2$;

$\xi^{(h)} = g(k,\theta^{(h)},C)$;

$\xi^{(h+1)} = g(k,\theta^{(h+1)},C)$;

$\xi^{(h+2)} = g(k,\theta^{(h+2)},C)$;

$g[\theta^{(h)},\theta^{(h+1)}] = \frac{\xi^{(h+1)}-\xi^{(h)}}{\theta^{(h+1)}-\theta^{(h)}}$;

while $(|\theta^{(h+2)}-\theta^{(h+1)}| > \varepsilon)$ **do**

$g[\theta^{(h+1)},\theta^{(h+2)}] = \frac{\xi^{(h+2)}-\xi^{(h+1)}}{\theta^{(h+2)}-\theta^{(h+1)}}$;

$g[\theta^{(h)},\theta^{(h+1)},\theta^{(h+2)}] = \frac{g[\theta^{(h+1)},\theta^{(h+2)}]-g[\theta^{(h)},\theta^{(h+1)}]}{\theta^{(h+2)}-\theta^{(h)}}$;

$\theta^{(h+3)} = \frac{1}{2}\Big(\theta^{(h+1)}+\theta^{(h+2)}-\frac{g[\theta^{(h+1)},\theta^{(h+2)}]}{g[\theta^{(h)},\theta^{(h+1)},\theta^{(h+2)}]}\Big)$;

$\xi^{(h+3)} = g(k,\theta^{(h+3)},C)$;

$h = h+1$;

end while

return $\theta^{(h+2)}$

where ε is the tolerance threshold. To accelerate the order of convergence of the sequence $(\theta^{(h)})$, we can pose

$$\theta^{(h+3)} = \Xi^{(h)} - \Big(\frac{g[\theta^{(h-1)},\theta^{(h)},\theta^{(h+1)},\theta^{(h+2)}]}{2\,g[\theta^{(h)},\theta^{(h+1)},\theta^{(h+2)}]}\Big)\Upsilon^{(h)}, \tag{3.45}$$

where

$$\Xi^{(h)} = \frac{1}{2}\left(\theta^{(h+1)}+\theta^{(h+2)}-\frac{g[\theta^{(h+1)},\theta^{(h+2)}]}{g[\theta^{(h)},\theta^{(h+1)},\theta^{(h+2)}]}\right)$$

and

$$\begin{aligned}\Upsilon^{(h)} &= (\theta^{(h)}-\Xi^{(h)})(\theta^{(h+1)}-\Xi^{(h)})+(\theta^{(h)}-\Xi^{(h)})(\theta^{(h+2)}-\Xi^{(h)})\\ &+ (\theta^{(h+1)}-\Xi^{(h)})(\theta^{(h+2)}-\Xi^{(h)}).\end{aligned}$$

Indeed, we have the following

Theorem 3.5.3. (see also [37, Theorem 3.8.1]) *Let $g(k,\cdot,C):[a,b]\to\mathbb{R}$ be of class $LC^3([a,b])$, $\widehat{\theta}\in]a,b[$ be such that $g'(k,\widehat{\theta},C)=0$ and $g''(k,\widehat{\theta},C)\neq 0$. If $\theta^{(0)}$, $\theta^{(1)}$, $\theta^{(2)}$ are distinct and sufficiently close to $\widehat{\theta}$, then the sequence $\{\theta^{(h)}\}_{h\in\mathbb{N}}$ is univoquely defined by (3.45), and $\{\theta^{(h)}\}_{h\in\mathbb{N}}$ converges with weak order $p\simeq 1.465$.*

The relative algorithm is the following

function SPI(k,C_R)

$h = 0$;

$\theta^{(0)} = \varphi_R^{(1)}+\eta$;

$\theta^{(1)} = \varphi_R^{(6)} - \eta;$

$\theta^{(2)} = (\theta^{(0)} + \theta^{(1)})/2;$

$\xi^{(h)} = g(k, \theta^{(h)}, C);$

$\xi^{(h+1)} = g(k, \theta^{(h+1)}, C);$

$\xi^{(h+2)} = g(k, \theta^{(h+2)}, C);$

$g[\theta^{(h)}, \theta^{(h+1)}] = \frac{\xi^{(h+1)} - \xi^{(h)}}{\theta^{(h+1)} - \theta^{(h)}};$

$g[\theta^{(h+1)}, \theta^{(h+2)}] = \frac{\xi^{(h+2)} - \xi^{(h+1)}}{\theta^{(h+2)} - \theta^{(h+1)}};$

$g[\theta^{(h)}, \theta^{(h+1)}, \theta^{(h+2)}] = \frac{g[\theta^{(h+1)}, \theta^{(h+2)}] - g[\theta^{(h)}, \theta^{(h+1)}]}{\theta^{(h+2)} - \theta^{(h)}};$

$\theta^{(h+3)} = \frac{1}{2}\left(\theta^{(h+1)} + \theta^{(h+2)} - \frac{g[\theta^{(h+1)}, \theta^{(h+2)}]}{g[\theta^{(h)}, \theta^{(h+1)}, \theta^{(h+2)}]}\right);$

$\xi^{(h+3)} = g(k, \theta^{(h+3)}, C);$

$h = h + 1;$

while $(|\theta^{(h+2)} - \theta^{(h+1)}| > \varepsilon)$ **do**

$g[\theta^{(h+1)}, \theta^{(h+2)}] = \frac{\xi^{(h+2)} - \xi^{(h+1)}}{\theta^{(h+2)} - \theta^{(h+1)}};$

$g[\theta^{(h)}, \theta^{(h+1)}, \theta^{(h+2)}] = \frac{g[\theta^{(h+1)}, \theta^{(h+2)}] - g[\theta^{(h)}, \theta^{(h+1)}]}{\theta^{(h+2)} - \theta^{(h)}};$

$\Xi^{(h)} = \frac{1}{2}\left(\theta^{(h+1)} + \theta^{(h+2)} - \frac{g[\theta^{(h+1)}, \theta^{(h+2)}]}{g[\theta^{(h)}, \theta^{(h+1)}, \theta^{(h+2)}]}\right);$

$\Upsilon^{(h)} = (\theta^{(h)} - \Xi^{(h)})(\theta^{(h+1)} - \Xi^{(h)}) + (\theta^{(h)} - \Xi^{(h)})(\theta^{(h+2)} - \Xi^{(h)}) + (\theta^{(h+1)} - \Xi^{(h)})(\theta^{(h+2)} - \Xi^{(h)});$

$g[\theta^{(h-1)}, \theta^{(h)}, \theta^{(h+1)}, \theta^{(h+2)}] = \frac{g[\theta^{(h)}, \theta^{(h+1)}, \theta^{(h+2)}] - g[\theta^{(h-1)}, \theta^{(h)}, \theta^{(h+1)}]}{\theta^{(h+2)} - \theta^{(h-1)}};$

$\theta^{(h+3)} = \Xi^{(h)} - \left(\frac{g[\theta^{(h-1)}, \theta^{(h)}, \theta^{(h+1)}, \theta^{(h+2)}]}{2g[\theta^{(h)}, \theta^{(h+1)}, \theta^{(h+2)}]}\right)\Upsilon^{(h)};$

$\xi^{(h+3)} = g(k, \theta^{(h+3)}, C);$

$h = h + 1;$

end while

return $\theta^{(h+2)}$.

where ε is the tolerance threshold.

3.5.7 Hybrid SPI and GSS

In our case the SPI algorithm could not converge to the desired solutions, since the derivative of the function $g(k, \cdot, C)$, on each interval lying between any two successive points of discontinuity, can vanish also in correspondence of the points which are not minimizers. Moreover, it is pos-

sible that the updates of the solution do not belong to the initial uncertainty interval, that is the interval in which the objective function is quasi-convex. We saw experimentally that, in general, the SPI algorithm does not converge to the minimum of the function $g(k,\cdot,C)$. To guarantee the convergence to that minimum a hybrid *Successive Parabolic Interpolation* and *Golden Section Search* technique is necessary (see also [37]).

This algorithm constructs a sequence $\{\theta^{(h)}\}_{h\in\mathbb{N}}$ such that

$$g(k,\theta^{(h)},C) \geq g(k,\theta^{(h+1)},C) \qquad h=0,1,\ldots, \tag{3.46}$$

while at the hth step we have an uncertainty interval $[a^{(h)},b^{(h)}]$, and we now discuss it in detail.

Let $\phi=(\sqrt{5}+1)/2$ be the *golden ratio* or *golden section*, let φ be as in the equation (4.19), let $\eta\in\mathbb{R}^+$ be small enough, and let $[a^{(0)}=\varphi+\eta, b^{(0)}=\varphi+\frac{\pi}{2}-\eta]$ be the initial uncertainty interval. The sequence is initialized as

$$\theta^{(0)}=\theta^{(1)}=\theta^{(2)}=a^{(2)}+\frac{b^{(2)}-a^{(2)}}{\phi},$$

which is equivalent to a golden section search step (see also [106]).

We will rely on the successive parabolic interpolation algorithm (see also [100]), which extends a finite sequence of approximations of the required minimum by adding the minimum of the parabola that interpolates the objective function on the last three terms of that sequence. The main step of the successive parabolic interpolation algorithm can be written as

$$\theta^{(h+3)}=\theta^{(h+2)}+\frac{p}{q},$$

where

$$\begin{aligned} p= \quad & (\theta^{(h+2)}-\theta^{(h)})^2(g(k,\theta^{(h+2)},C)-g(k,\theta^{(h+1)},C)) \\ & -(\theta^{(h+2)}-\theta^{(h+1)})^2(g(k,\theta^{(h+2)},C)-g(k,\theta^{(h)},C)) \end{aligned}$$

and

$$\begin{aligned} q= \quad & 2(\theta^{(h+2)}-\theta^{(h)})(g(k,\theta^{(h+2)},C)-g(k,\theta^{(h+1)},C)) \\ & -2(\theta^{(h+2)}-\theta^{(h+1)})(g(k,\theta^{(h+2)},C)-g(k,(\theta^{(h)},C)). \end{aligned}$$

If at any point any two of $\theta^{(h)}$, $\theta^{(h+1)}$, and $\theta^{(h+2)}$ coincide, or the parabola degenerates to a line (in which case, $q=0$), or the successive parabolic interpolation update is outside the

current uncertainty interval $[a^{(h)}, b^{(h)}]$, then the step is performed using the golden section search technique. The pseudocode of this algorithm is as follows.

function SPI-GSS(k,C)

$h = 0$;

$[a^{(0)}, b^{(0)}] = [\varphi + \eta, \varphi + \pi/2 - \eta]$;

$\theta^{(0)} = \theta^{(1)} = \theta^{(2)} = a^{(2)} + (b^{(2)} - a^{(2)})/\phi$;

while $(|\theta^{(h+2)} - \theta^{(h+1)}| > \varepsilon)$ **do**

$p = (\theta^{(h+2)} - \theta^{(h)})^2 (g(k, \theta^{(h+2)}, C) - g(k, \theta^{(h+1)}, C))$;

$p = p - (\theta^{(h+2)} - \theta^{(h+1)})^2 (g(k, \theta^{(h+2)}, C) - g(k, \theta^{(h)}, C))$;

$q = 2(\theta^{(h+2)} - \theta^{(h)})(g(k, \theta^{(h+2)}, C) - g(k, \theta^{(h+1)}, C))$;

$q = q - 2(\theta^{(h+2)} - \theta^{(h+1)})(g(k, \theta^{(h+2)}, C) - g(k, (\theta^{(h)}, C))$;

if $((q \neq 0)$ and $(\theta^{(h+2)} + p/q \in [a^{(h)}, b^{(h)}]))$ **then**

$\theta^{(h+3)} = \theta^{(h+2)} + p/q$;

else

if $(\theta^{(h+2)} < (a^{(h+2)} + b^{(h+2)})/2)$ **then**

$\theta^{(h+3)} = \theta^{(h+2)} + (b^{(h+2)} - \theta^{(h+2)})/r$;

else

$\theta^{(h+3)} = \theta^{(h+2)} + (a^{(h+2)} - \theta^{(h+2)})/r$;

end if

end if

Compute the new uncertainty interval $[a^{(h+1)}, b^{(h+1)}]$;

Order $\{\theta^{(i)}\}_{i=h,\ldots,h+3}$ in such a way that (3.46) holds;

$h = h + 1$;

end while

return $\theta^{(h+2)}$

Here $\varepsilon > 0$ is a suitable tolerance threshold.

During the last iterations, the algorithm usually stops choosing the golden section search steps, and performs only parabolic interpolation steps. Thus, the asymptotic convergence depends only on the SPI algorithm. We recall that the sequence $\{\theta^{(h)}\}_h$ converges to $\widehat{\theta}$ with *strong*

order p and *asymptotic constant* $\gamma > 0$ if

$$\lim_{h\to+\infty} \frac{|\theta^{(h+1)} - \widehat{\theta}|}{|\theta^{(h)} - \widehat{\theta}|^p} = \gamma,$$

and with *weak order p* if

$$\liminf_{h\to+\infty}(-\ln|\theta^{(h)} - \widehat{\theta}|^p)^{1/h} = p.$$

Note that strong convergence implies weak convergence, but in general the converse is not true.

Let $n \in \mathbb{N}$. A function $f : [a,b] \to \mathbb{R}$ is *of class* $LC^n([a,b])$ if its nth derivative exists and is Lipschitz. that is, if there exists a positive real number M_0 such that

$$\sup_{x,y\in[a,b],|x-y|\le\delta} |f^{(n)}(x) - f^{(n)}(y)| \le M_0\,\delta$$

for each $\delta > 0$. The following result holds.

Theorem 3.5.4 ([37, Theorem 3.7.1])**.** *Let $k \ge 0$, $\iota \in \{1,-1\}$, let $C \in \mathbb{R}^{2\times2}$ be a positive definite matrix, and let $g(k,\cdot,C)$ be a function of class $LC^3(N)$, where N is a neighborhood of its minimum $\widehat{\theta}$, such that $g''(k,\widehat{\theta},C) > 0$. Then the sequence $\{\theta^{(h)}\}_h$ obtained by the SPI algorithm either converges in the neighborhood N with either strong order $p \simeq 1.325$ or weak order $p = ((3+\sqrt{5})/2)^{1/3} \simeq 1.378$.*

Note that the function τ defined in (4.22) is not of class C^1, but can be approximated by the function

$$(\overline{\tau}(s))_i = \begin{cases} 0, & \text{if } s_i \le 0, \\ p_7(s_i), & \text{if } 0 < s_i \le 1, \\ s_i, & \text{if } 1 < s_i \le m-1, \\ q_7(s_i), & \text{if } m-1 < s_i \le m \\ m, & \text{if } s_i > m, \end{cases} \qquad i = 1,\dots,\nu^2, \tag{3.47}$$

where

$$\begin{aligned} p_7(x) &= -10x^7 + 36x^6 - 45x^5 + 20x^4, \\ q_7(x) &= m - p_7(m-x), \quad x \in \mathbb{R}, \end{aligned}$$

which is of class LC^3 on $\mathbb{R}^{n^2}$. If in (3.28) we replace the mapping τ in (4.22) with the function $\overline{\tau}$ in (3.47), then we obtain that $g(k,\cdot,C)$ is of class $LC^3((\varphi^{(1)}+\eta, \varphi^{(2)}+\pi-\eta))$. Therefore,

we are under the hypothesis of Theorem 3.5.4 and the minimization method has superlinear convergence.

3.5.8 The Newton method

To find the minimum of the function $g(k,\cdot,C)$, it is possible to apply the classical Newton method to its derivative, that is the following algorithm is performed:

function Newton(k,C)

$h=0$;

$\theta^{(1)}=(\varphi+\varphi+\pi/2)/2$;

$\theta^{(0)}=\theta^{(1)}+2\varepsilon$;

while $(|\theta^{(h+1)}-\theta^{(h)}|>\varepsilon)$ **do**

$h=h+1$;

$\theta^{(h+1)}=\theta^{(h)}-\frac{g'(k,\theta^{(h)},C)}{g''(k,\theta^{(h)},C)}$;

end while

return $\theta^{(h+1)}$

where ε is the tolerance threshold, and

$$\begin{aligned} g(k^{(l)},\theta,C) &= f^{(l)}(\theta,C)=(\tau(\widetilde{s}_r^{(l)}(\theta)))^T\cdot\tau(\widetilde{s}_v^{(l)}(\theta)) \\ &= \sum_{i=1}^{n^2}(\tau(\widetilde{s}_r^{(l)}(\theta)))_i(\tau(\widetilde{s}_v^{(l)}(\theta))_i, \end{aligned}$$

and τ is as in (4.22). Note that

$$\frac{\partial(\tau(\cdot)_i)}{\partial s_j}=\begin{cases} 0, & \text{if } s_j>0, \\ \delta_{i,j}, & \text{if } 0<s_j<m, \\ 0, & \text{if } s_j>m, \end{cases} \qquad i,j=1,\ldots,n^2, \tag{3.48}$$

where $\delta_{i,j}$ denotes the *Kronecker delta*. When the following quantities make sense, we get

$$\frac{d}{d\theta}f^{(l)}(\theta,C)=\sum_{i=1}^{n^2}\Big(\tau(\widetilde{s}_{v,}^{(l)}(\theta))_i\Big(\frac{d}{d\theta}(\tau(\widetilde{s}_{r,}^{(l)}(\theta))_i\Big)+(\tau(\widetilde{s}_r^{(l)}(\theta))_i\Big(\frac{d}{d\theta}(\tau(\widetilde{s}_v^{(l)}(\theta))_i\Big)\Big)$$

where for each $i = 1,2,\ldots,n^2$ it is

$$\begin{aligned}\frac{d}{d\theta}(\tau(\widetilde{s}_r^{(l)}(\theta))_i &= \sum_{j=1}^{n^2}\frac{\partial(\tau(\cdot)_i)}{\partial s_j}((\widetilde{s}_r^{(l)}(\theta))_j)\cdot((\widetilde{s}_r^{(l)}(\theta))_j)' = \\ &= \frac{\partial(\tau(\cdot)_i)}{\partial s_i}((\widetilde{s}_r^{(l)}(\theta))_i)\cdot((\widetilde{s}_r^{(l)}(\theta))_i)' = \\ &= \begin{cases}((\widetilde{s}_r^{(l)}(\theta))_i)', & \text{if } 0<(\widetilde{s}_r^{(l)}(\theta))_i<m,\\ 0, & \text{otherwise}\end{cases};\end{aligned}$$

$$\begin{aligned}\frac{d}{d\theta}(\tau(\widetilde{s}_v^{(l)}(\theta))_i &= \sum_{j=1}^{n^2}\frac{\partial(\tau(\cdot)_i)}{\partial s_j}((\widetilde{s}_v^{(l)}(\theta))_j)\cdot((\widetilde{s}_v^{(l)}(\theta))_j)' = \\ &= \frac{\partial(\tau(\cdot)_i)}{\partial s_i}((\widetilde{s}_{v,t}^{(l)}(\theta))_i)\cdot((\widetilde{s}_v^{(l)}(\theta))_i)' \\ &= \begin{cases}((\widetilde{s}_v^{(l)}(\theta))_i)', & \text{if } 0<(\widetilde{s}_v^{(l)}(\theta))_i<m,\\ 0, & \text{otherwise}\end{cases}\end{aligned}$$

Moreover, we have:

$$\begin{aligned}\frac{d^2}{d\theta^2}f^{(l)}(\theta,C) &= \sum_{i=1}^{n^2}\Big(\tau(\widetilde{s}_v^{(l)}(\theta))_i\Big(\frac{d^2}{d\theta^2}(\tau(\widetilde{s}_r^{(l)}(\theta))_i\Big)+(\tau(\widetilde{s}_r^{(l)}(\theta))_i\Big(\frac{d^2}{d\theta^2}(\tau(\widetilde{s}_v^{(l)}(\theta))_i\Big)\Big)+\\ &+ 2\Big(\frac{d}{d\theta}(\tau(\widetilde{s}_r^{(l)}(\theta))_i\Big)\Big(\frac{d}{d\theta}(\tau(\widetilde{s}_v^{(l)}(\theta))_i\Big),\end{aligned}$$

where

$$\frac{d^2}{d\theta^2}(\tau(\widetilde{s}_r^{(l)}(\theta))_i = \begin{cases}((\widetilde{s}_r^{(l)}(\theta))_i)'', & \text{if } 0<(\widetilde{s}_r^{(l)}(\theta))_i<m,\\ 0, & \text{otherwise}\end{cases};$$

$$\frac{d^2}{d\theta^2}(\tau(\widetilde{s}_v^{(l)}(\theta))_i = \begin{cases}((\widetilde{s}_v^{(l)}(\theta))_i)'', & \text{if } 0<(\widetilde{s}_v^{(l)}(\theta))_i<m,\\ 0, & \text{otherwise}\end{cases}.$$

For every $i = 1,2,\ldots,n^2$, we get:

$$(\widetilde{s}_r^{(l)}(\theta))_i = y_{11}^{(l)}(\theta)(x_r)_i + y_{12}^{(l)}(\theta)(x_v)_i,$$

$$(\widetilde{s}_v^{(l)}(\theta))_i = y_{21}^{(l)}(\theta)(x_r)_i + y_{22}^{(l)}(\theta)(x_v)_i,$$

$$((\widetilde{s}_r^{(l)}(\theta))_i)' = (y_{11}^{(l)})'(\theta)(x_r)_i + (y_{12}^{(l)})'(\theta)(x_v)_i,$$

$$((\widetilde{s}_v^{(l)}(\theta))_i)' = (y_{21}^{(l)})'(\theta)(x_r)_i + (y_{22}^{(l)})'(\theta)(x_v)_i,$$

$$((\widetilde{s}_r^{(l)}(\theta))_i)'' = (y_{11}^{(l)})''(\theta)(x_r)_i + (y_{12}^{(l)})''(\theta)(x_v)_i,$$

$$((\widetilde{s}_v^{(l)}(\theta))_i)'' = (y_{21}^{(l)})''(\theta)(x_r)_i + (y_{22}^{(l)})''(\theta)(x_v)_i,$$

where

$$\begin{aligned}
y_{11}^{(l)}(\theta) &= \frac{z_{22}(\theta)(\det(C) - k_R^{(l)}(z_{11}(\theta) - z_{21}(\theta))^2)}{(z_{22}(\theta) - z_{12}(\theta))\det(C)} - z_{21}(\theta)\frac{k^{(l)}(z_{11}(\theta) - z_{21}(\theta))}{\det(C)}, \\
y_{12}^{(l)}(\theta) &= -\frac{z_{12}(\theta)(\det(C) - k^{(l)}(z_{11}(\theta) - z_{21}(\theta))^2)}{(z_{22}(\theta) - z_{12}(\theta))\det(C)} + z_{11}(\theta)\frac{k^{(l)}(z_{11}(\theta) - z_{21}(\theta))}{\det(C)}, \\
y_{21}^{(l)}(\theta) &= -\frac{z_{21}(\theta)}{z_{11}(\theta) - z_{21}(\theta)}, \\
y_{22}^{(l)}(\theta) &= \frac{z_{11}^{(}\theta)}{z_{11}(\theta) - z_{21}(\theta)}.
\end{aligned}$$

We have

$$\begin{aligned}
(y_{11}^{(l)})'(\theta) &= -\frac{z_{22}(\theta)((z'_{22}(\theta) - (z_{12})'(\theta)))(\det(C) - k^{(l)}(z_{11}(\theta) - z_{21}(\theta))^2)}{(z_{22}(\theta) - z_{12}(\theta))^2\det(C)} + \\
&+ \frac{(z_{22})'(\theta)(\det(C) - k^{(l)}(z_{11}(\theta) - z_{21}(\theta))^2)}{(z_{22}(\theta) - z_{12}(\theta))\det(C)} - \\
&- \frac{k^{(l)}(z_{21})'(\theta)(z_{11}(\theta) - z_{21}(\theta))}{\det(C)} - \\
&- -2\frac{k^{(l)}z_{22}(\theta)(z_{11}(\theta)) - z_{21}(\theta))((z_{11})'(\theta) - (z_{21})'(\theta)}{(z_{22}(\theta) - z_{12}(\theta))\det(C)} - \\
&- \frac{k^{(l)}z_{21}(\theta)((z_{11})'(\theta) - (z_{21})'(\theta))}{\det(C)}
\end{aligned}$$

Therefore,

$$
\begin{aligned}
(y_{11}^{(l)})''(\theta) &= -\frac{z_{22}(\theta)((z_{22})''(\theta)-(z_{12})''(\theta))(\det(C)-k^{(l)}(z_{11}(\theta)-z_{21}(\theta))^2)}{(z_{22}(\theta)-z_{12}(\theta))^2\det(C)} + \\
&+ \frac{(z_{22})''(\theta)(\det(C)-k^{(l)}(z_{11}(\theta)-z_{21}(\theta))^2)}{(z_{22}(\theta)-z_{12}(\theta))\det(C)} + \\
&+ 2\frac{z_{22}(\theta)((z_{22})'(\theta)-(z_{12})'(\theta))^2(\det(C)-k^{(l)}(z_{11}(\theta)-z_{21}(\theta))^2)}{(z_{22}(\theta)-z_{12}(\theta))^3\det(C)} - \\
&- 2\frac{(z_{22})'(\theta)((z_{22})'(\theta)-(z_{12})'(\theta))(\det(C)-k^{(l)}(z_{11}(\theta)-z_{21}(\theta))^2)}{(z_{22}(\theta)-z_{12}(\theta))^2\det(C)} + \\
&+ 4\frac{k^{(l)}z_{22}(\theta)(z_{11}(\theta)-z_{21}(\theta))((z_{11})'(\theta)-(z_{21})'(\theta))((z_{22})'(\theta)-(z_{12})'(\theta))}{(z_{22}(\theta)-z_{12}(\theta))^2\det(C)} - \\
&- 4\frac{k^{(l)}z'_{22}(\theta)((z_{11})'(\theta)-(z_{21})'(\theta))(z_{11}(\theta)-z_{21}(\theta))}{(z_{22}(\theta)-z_{12}(\theta))\det(C)} - \\
&- \frac{k^{(l)}(z_{11}(\theta)-z_{21}(\theta))(z_{21})''(\theta)}{\det(C)} - \\
&- 2\frac{k^{(l)}z_{22}(\theta)(z_{11}(\theta)-z_{21}(\theta))((z_{11})''(\theta)-(z_{21})''(\theta))}{(z_{22}(\theta)-z_{12}(\theta))\det(C)} - \\
&- \frac{k^{(l)}(z''_{11}(\theta)-z''_{21}(\theta))z_{21}(\theta)}{\det(C)} - \\
&- 2\frac{k^{(l)}z'_{21}(\theta)(z'_{11}(\theta)-z'_{21}(\theta))}{(\det(C)} - \\
&- 2\frac{k^{(l)}z_{22}(\theta)(z'_{11}(\theta)-z'_{21}(\theta))}{(z_{22}(\theta)-z_{12}(\theta))\det(C)},
\end{aligned}
$$

and hence

$$
\begin{aligned}
(y_{12}^{(l)})''(\theta) &= \frac{z_{12}(\theta)((z_{22})''(\theta)-(z_{12})''(\theta))(\det(C)-k^{(l)}(z_{11}^{(}\theta)-z_{21}(\theta))^2)}{(z_{22}(\theta)-((z_{12}(\theta)))^2\det(C)} - \\
&- \frac{(z_{12})''(\theta)(\det(C)-k^{(l)}(z_{11}(\theta)-(z_{21}(\theta)))^2)}{(z_{22}(\theta)-z_{12}(\theta))\det(C)} - \\
&- 2\frac{z_{12}(\theta)((z_{22})'(\theta)-((z_{12})'(\theta)))^2(\det(C)-k^{(l)}(z_{11}(\theta)-z_{21}(\theta))^2)}{(z_{22}(\theta)-z_{12}(\theta))^3\det(C)} + \\
&+ 2\frac{(z_{12})'(\theta)((z_{22})'(\theta)-(z_{12})'(\theta))(\det(C)-k^{(l)}(z_{11}(\theta)-z_{21}(\theta))^2)}{(z_{22}(\theta)-z_{12}(\theta))^2\det(C)} - \\
&- 4\frac{k^{(l)}z_{12}(\theta)(z_{11}(\theta)-z_{21}(\theta))((z_{11})'(\theta)-(z_{21})'(\theta))((z_{22})'(\theta)-(z_{12})'(\theta))}{(z_{22}(\theta)-z_{12}(\theta))^2\det(C)} + \\
&+ 4\frac{k^{(l)}(z_{12})'(\theta)((z_{11})'(\theta)-(z_{21})'(\theta))(z_{11}(\theta)-z_{21}(\theta))}{(z_{22}(\theta)-z_{12}(\theta))\det(C)} + \\
&+ \frac{k^{(l)}(z_{11}(\theta)-z_{21}(\theta))(z_{11})''(\theta)}{\det(C)} + \\
&+ 2\frac{k^{(l)}z_{12}(\theta)(z_{11}(\theta)-z_{21}(\theta))((z_{11})''(\theta)-(z_{21})''(\theta))}{(z_{22}(\theta)-z_{12}(\theta))\det(C)} + \\
&+ \frac{k^{(l)}((z_{11})''(\theta)-(z_{21})''(\theta))z_{11}(\theta)}{\det(C)} + \\
&+ 2\frac{k^{(l)}(z_{11})'(\theta)((z_{11})'(\theta)-(z_{21})'(\theta))}{(\det(C)} + \\
&+ 2\frac{k^{(l)}z_{12}(\theta)((z_{11})'(\theta)-(z_{21}'(\theta))}{(z_{22}(\theta)-z_{12}(\theta))\det(C)}.
\end{aligned}
$$

Moreover, we get

$$
\begin{aligned}
(y_{21}^{(l)})'(\theta) &= \frac{z_{21}(\theta)((z_{11})'(\theta)-(z_{21})'(\theta))}{(z_{11}(\theta)-z_{21}(\theta))^2} - \frac{(z_{21})'(\theta)}{z_{11}(\theta)-z_{21}(\theta)}, \\
(y_{21}^{(l)})''(\theta) &= \frac{z_{21}(\theta)((z_{11})''(\theta)-(z_{21})''(\theta))}{(z_{11})(\theta)-z_{21}(\theta))^2} - \frac{(z_{21})''(\theta)}{z_{11}(\theta)-z_{21}(\theta)} + \\
&+ \frac{2(z_{21})'(\theta)((z_{11})'(\theta)-(z_{21})'(\theta))}{(z_{11}(\theta)-z_{21}(\theta))^2} - \frac{2z_{21}(\theta)((z_{11})'(\theta)-(z_{21}'(\theta))^2}{(z_{11}(\theta)-z_{21}(\theta))^3},
\end{aligned}
$$

$$
\begin{aligned}
(y_{22}^{(l)})'(\theta) &= \frac{z_{11}(\theta)((z_{11})'(\theta)-(z_{21})'(\theta))}{(z_{11}(\theta)-z_{21}(\theta))^2} + \frac{(z_{11})'(\theta)}{z_{11}(\theta)-z_{21}(\theta)}, \\
(y_{22}^{(l)})''(\theta) &= -\frac{z_{11}(\theta)((z_{11})''(\theta)-(z_{21})''(\theta))}{(z_{11}(\theta)-z_{21}(\theta))^2} + \frac{(z_{11})'(\theta)}{z_{11}(\theta)-z_{21}(\theta)} - \\
&- \frac{2(z_{11})'(\theta)((z_{11})'(\theta)-(z_{21})'(\theta))}{(z_{11}(\theta)-z_{21}(\theta))^2} + \frac{2z_{11}(\theta)((z_{11})'(\theta)-(z_{21})'(\theta))^2}{(z_{11}(\theta)-z_{21}(\theta))^3}.
\end{aligned}
$$

Let $\widehat{\theta}$ be a minimizer of the function $g(k,\cdot,C)$. If $g''(k,\widehat{\theta},C)\neq 0$, then the Newton method is locally convergent with order 2. Anyway, we experimentally check that in our case the Newton method does not converge to a minimizer of the function $g(k,\cdot,C)$.

3.5.9 The Armijo Line Search (ALS)

Another method based on the derivative of the function $g(k,\cdot,C)$ is the ALS. The relative algorithm is the following

function ALS(k,C)

$h=0;$

$\theta^{(1)}=(\varphi+\varphi+\pi/2)/2;$

$\theta^{(0)}=\theta^{(1)}+2\varepsilon;$

while $(|\theta^{(h+1)}-\theta^{(h)}|>\varepsilon)$ **do**

$h=h+1;$

$\theta^{(h+1)}=\theta^{(h)};$

$\xi^{(h+1)}=g(k,\theta^{(h)});$

$der=g'(k,\theta^{(h)},C);$

$i=0;$

$\overline{\theta}^{(i)}=\theta^{(h)}-der;$

while $(\overline{\theta}^{(i)}\notin[\varphi+\eta,\varphi+\pi/2-\eta])$ **do**

$i=i+1;$

$\overline{\theta}^{(i)}=\theta^{(h)}-der/2^{i};$

end while

$\overline{\xi}^{(i)}=g(k,\overline{\theta}^{(i)},C);$

while $(\overline{\xi}^{(i)}>\xi^{(h+1)}-|der|/2^{i+1})$ **do**

if $(\overline{\xi}^{(i)}<\xi^{(h+1)})$ **then**

$\theta^{(h+1)}=\overline{\theta}^{(i)};$

$\xi^{(h+1)}=\overline{\xi}^{(i)};$

end if

$i=i+1;$

$\overline{\theta}^{(i)}=\theta^{(h)}-der/2^{i};$

$\overline{\xi}^{(i)}=g(k,\overline{\theta}^{(i)},C);$

end while

if $(\overline{\xi}^{(i)}<\xi^{(h+1)})$ **then**

$\theta^{(h+1)} = \overline{\theta}^{(i)}$;

$\xi^{(h+1)} = \overline{\xi}^{(i)}$;

end if

end while

return $\theta^{(h+1)}$

where ε is a suitable threshold tolerance.

The following result holds.

Theorem 3.5.5. (see also [33, Theorem 11], [149, Theorem 5.4.1.8]) *Let* $g(k,\cdot,C) : [a,b] \to \mathbb{R}_0^+$ *and* $\theta^{(0)} \in [a,b]$ *be such that the set* $K = \{\theta \in [a,b] : g(k,\theta,C) \leq g(k,\theta^{(0)},C)\}$ *is compact and* $g(k,\cdot,C) \in C^1(A)$*, where* $K \subset A$ *and* $A \subset [a,b]$ *is open. Then every sequence* $\{\theta^{(h)}\}_{h\in\mathbb{N}}$ *defined by the ALS method has at least a limit point* $\theta \in K$*, and every limit point is a stationary point for h.*

3.5.10 Comparison of the results

We initially compared the results of methods which do not use derivatives, like the SA, TPS, GSS, SPI-GSS algorithms. We tested them in restoring the documents of the Figures (3.15)-(3.16), which were mixed with the mixture matrix (4.24). In Tables 3.1 and 3.2 there are the calculation times and the mean square errors, indicated with MSE, with respect to the ideal documents of the four previously presented algorithms. From these tables we deduce that the algorithm SPI-GSS is the most efficent in terms of computational costs, among the considered ones. Moreover, we tested the SPI-GSS algorithm. The related results are presented in Table 3.3. We observe that the errors in terms of MSE are similar to those found in Table 3.2 where ι was not fixed, while the computational costs are substantially halved. Successively we compare the SPI-GSS technique with the Armijo algorithm. The results are shown in Table 3.3, in which we deduce that the SPI-GSS algorithm is more efficient, and thus we choose it for minimizing functions with MATODS techniques.

Ideal	SA			TPS		
Document	Time	MSE recto	MSE verso	Time	MSE recto	MSE verso
Figure 3.15 (a)	$32.78\,s$	$1.15 \cdot 10^{-6}$	$2.24 \cdot 10^{-8}$	$0.50\,s$	$1.09 \cdot 10^{-10}$	$2.94 \cdot 10^{-11}$
Figure 3.15 (b)	$47.08\,s$	$3.01 \cdot 10^{-7}$	$8.36 \cdot 10^{-8}$	$0.67\,s$	$4.16 \cdot 10^{-9}$	$6.54 \cdot 10^{-10}$
Figure 3.15 (c)	$41.55\,s$	$7.40 \cdot 10^{-8}$	$1.02 \cdot 10^{-7}$	$0.71\,s$	$6.74 \cdot 10^{-8}$	$9.44 \cdot 10^{-8}$
Figure 3.15 (d)	$515.80\,s$	0.63	5.35	$7.24\,s$	0.63	5.35

Table 3.1: Results obtained by algorithms SA and TPS.

Ideal	GSS			SPI-GSS		
Docment	Time	MSE recto	MSE verso	Time	MSE recto	MSE verso
Figure 3.15 (a)	$0.42\,s$	$2.75 \cdot 10^{-10}$	$3.00 \cdot 10^{-12}$	$0.40\,s$	$1.47 \cdot 10^{-10}$	$1.71 \cdot 10^{-11}$
Figure 3.15 (b)	$0.64\,s$	$5.30 \cdot 10^{-9}$	$7.51 \cdot 10^{-10}$	$0.61\,s$	$5.62 \cdot 10^{-9}$	$9.02 \cdot 10^{-10}$
Figure 3.15 (c)	$0.59\,s$	$6.96 \cdot 10^{-8}$	$9.74 \cdot 10^{-8}$	$0.56\,s$	$6.92 \cdot 10^{-8}$	$9.68 \cdot 10^{-8}$
Figure 3.15 (d)	$7.64\,s$	0.63	5.35	$5.37\,s$	0.63	5.35

Table 3.2: Results obtained by algorithms GSS and SPI-GSS.

3.5.11 The empty page case

Now we consider the case $\det C = 0$. Since x_r and x_v are nonnegative vectors, from (4.6) and the Cauchy-Schwartz inequality it follows that there exists $\zeta > 0$ with $x_r = \zeta\, x_v$. An example is shown in Figure 3.14 (a).

(a) Observed document. (b) Symmetric reconstruction of the document.

Figure 3.14: Document whose recto is a multiple of the verso.

In this case, it is natural to assume that either $\widetilde{s}_r$, the estimate of s_r, or $\widetilde{s}_v$, the estimate of s_v, are zero, that is, that either the recto or the verso of the ideal source document is an empty page. When $\zeta \geq 1$, we assume that $\widetilde{s}_v = 0\,e$, and we get that $x_r = \widetilde{a}_{11}\,\widetilde{s}_r$, $x_v = \widetilde{a}_{21}\,\widetilde{s}_r$ and $\zeta = \frac{\widetilde{a}_{11}}{\widetilde{a}_{21}}$, where

Ideal	SPI-GSS			ALS		
Document	Time	MSE recto	MSE verso	Time	MSE recto	MSE verso
Figure 3.15 (a)	$0.20\,s$	$1.34 \cdot 10^{-10}$	$4.26 \cdot 10^{-11}$	$0.47\,s$	$3.78 \cdot 10^{-10}$	$1.28 \cdot 10^{-15}$
Figure 3.15 (b)	$0.30\,s$	$3.69 \cdot 10^{-9}$	$6.27 \cdot 10^{-10}$	$0.95\,s$	$6.05 \cdot 10^{-9}$	$9.12 \cdot 10^{-10}$
Figure 3.15 (c)	$0.27\,s$	$6.88 \cdot 10^{-8}$	$9.63 \cdot 10^{-8}$	$1.27\,s$	$7.02 \cdot 10^{-8}$	$9.81 \cdot 10^{-8}$
Figure 3.15 (d)	$3.28\,s$	0.63	5.35	$6.71\,s$	0.63	5.35

Table 3.3: Results obtained by algorithms SPI-GSS, NL-SOR and ALS by fixing $\iota = 1$.

$\widetilde{a}_{11}$ and $\widetilde{a}_{21}$ are estimates of a_{11} and a_{21}, respectively. Therefore we obtain

$$\widetilde{s}_r = \frac{1}{\widetilde{a}_{11}} x_r, \qquad \widetilde{s}_v = 0 \qquad \widetilde{A} = \begin{bmatrix} \widetilde{a}_{11} & 1-\widetilde{a}_{11} \\ \dfrac{1}{\zeta}\widetilde{a}_{11} & 1-\dfrac{1}{\zeta}\widetilde{a}_{11} \end{bmatrix},$$

where $\widetilde{a}_{11}$ is arbitrarily chosen in $]0,1]$ and $\widetilde{A}$ is an estimate of the mixing matrix A. If we impose that the matrix $\widetilde{A}$ is symmetric, then we have that $\widetilde{a}_{11} = \frac{\zeta}{\zeta+1}$. In Figure 3.14 (b) we present a symmetric reconstruction of the document shown in Figure 3.14 (a).

If $0 < \zeta < 1$, then we set $\widetilde{s}_r = 0\,e$ and get that $x_v = \widetilde{a}_{12}\widetilde{s}_v$, $x_r = \widetilde{a}_{22}\widetilde{s}_v$, and $\zeta = \frac{\widetilde{a}_{12}}{\widetilde{a}_{22}}$, where $\widetilde{a}_{12}$ and $\widetilde{a}_{22}$ are estimates of a_{12} and a_{22}, respectively. Therefore we obtain

$$\widetilde{s}_v = 0, \qquad \widetilde{s}_v = \frac{1}{\widetilde{a}_{22}} x_v, \qquad \widetilde{A} = \begin{bmatrix} 1-\zeta\,\widetilde{a}_{22} & \zeta\,\widetilde{a}_{22} \\ 1-\widetilde{a}_{22} & \widetilde{a}_{22} \end{bmatrix},$$

where $\widetilde{a}_{22}$ is arbitrarily chosen in $]0,1]$, and by requiring that the the estimated mixing matrix $\widetilde{A}$ is symmetric, we obtain that $\widetilde{a}_{22} = \frac{1}{\zeta+1}$.

Note that, since we consider a locally linear model, it may happen that in a given subdomain at least one of the original documents is empty.

3.5.12 Color image case

An $n \times n$ color image is usually encoded in the *RGB* space, where *R*, *G*, and *B* indicate the red, green, and blue color, respectively. We consider every color component of a document as a pair of images, the recto and the verso, and we denote the red, green, and blue data components as

$$\widehat{x}_R = \begin{bmatrix} \widehat{x}_{rR} & \widehat{x}_{vR} \end{bmatrix}, \quad \widehat{x}_G = \begin{bmatrix} \widehat{x}_{rG} & \widehat{x}_{vG} \end{bmatrix}, \quad \widehat{x}_B = \begin{bmatrix} \widehat{x}_{rB} & \widehat{x}_{vB} \end{bmatrix}$$

respectively, where $\widehat{x}_{rR},\widehat{x}_{rG},\widehat{x}_{rB},\widehat{x}_{vR},\widehat{x}_{vG},\widehat{x}_{vB} \in [0,255]^{v^2}$. We write the observed color document as

$$\widehat{x} = \begin{bmatrix} \widehat{x}_{rR} & \widehat{x}_{vR} & \widehat{x}_{rG} & \widehat{x}_{vG} & \widehat{x}_{rB} & \widehat{x}_{vB} \end{bmatrix},$$

which belongs to $[0,255]^{v^2\times 6}$. The source ideal document is given by the matrix

$$\widehat{s} = \begin{bmatrix} \widehat{s}_{rR} & \widehat{s}_{vR} & \widehat{s}_{rG} & \widehat{s}_{vG} & \widehat{s}_{rB} & \widehat{s}_{vB} \end{bmatrix},$$

where $\widehat{s} \in [0,255]^{v^2\times 6}$, and we set

$$\widehat{s}_R = \begin{bmatrix} \widehat{s}_{rR} & \widehat{s}_{vR} \end{bmatrix}, \quad \widehat{s}_G = \begin{bmatrix} \widehat{s}_{rG} & \widehat{s}_{vG} \end{bmatrix}, \quad \widehat{s}_B = \begin{bmatrix} \widehat{s}_{rB} & \widehat{s}_{vB} \end{bmatrix}.$$

The linear model for a color image is $\widehat{x}^T = A\widehat{s}^T$. In this case the mixture matrix $A \in \mathbb{R}^{6\times 6}$ is the block matrix

$$A = \begin{bmatrix} A_R & O & O \\ O & A_G & O \\ O & O & A_B \end{bmatrix}, \text{ with } \quad A_R = \begin{bmatrix} a_{11}^R & a_{12}^R \\ a_{21}^R & a_{22}^R \end{bmatrix}, A_G = \begin{bmatrix} a_{11}^G & a_{12}^G \\ a_{21}^G & a_{22}^G \end{bmatrix}, A_B = \begin{bmatrix} a_{11}^B & a_{12}^B \\ a_{21}^B & a_{22}^B \end{bmatrix},$$

where $O \in \mathbb{R}^{2\times 2}$ is the zero matrix. Thus, we get

$$\widehat{x}_R^T = A_R\widehat{s}_R^T, \qquad \widehat{x}_G^T = A_G\widehat{s}_R^T, \qquad \widehat{x}_B^T = A_B\widehat{s}_R^T.$$

According to our model, every observed channel is formed by a linear combination of components related to the same channel of the front and the back of the ideal source document, and we can solve the problem independently on each channel with the technique proposed for gray level images.

3.6 A new technique for solving the non-stationary problem

In this section we discuss how to use the MATODS algorithm to solve the non-stationary model proposed in Subsection 3.2.1. Given a document defined on a domain of dimension $n\times n$, in each non-overlapping subdomain of dimension $v\times v$ we model the problem by means of a linear operator, and assume that these linear operators vary smoothly between adjacent subdomains. For this purpose we solve the problem on an overlapping subimage of size $\overline{n}\times\overline{n}$, with $\overline{n} > v$, using the MATODS algorithm, and then we average the results obtained in each subdomain.

In other words, if n and $\overline{n}$ are multiples of ν, then we consider the subimages $\overline{x}^{(p,q)}$, for $p,q = 1,\ldots,\frac{n-\overline{n}}{\nu}$, which have fixed size $\overline{n}\times\overline{n}$, and solve the linear problem on each subimage. The domain of these subimages is obtained by shifting by ν pixels a window of $\nu\times\nu$ pixels either horizontally or vertically. Finally, we set the light intensity value of every pixel of the estimated source $\widetilde{s}$ to the arithmetic mean of the light intensity value of the estimated subsources to which the pixel belongs.

By this procedure, all pixels lying in a subdomain of dimension $\nu\times\nu$ belong to the same subimage, and on each subdomain the result is obtained by averaging the $\left(\frac{\overline{n}}{\nu}\right)^2$ linear operators. Note that the resulting average is a linear operator, being a linear combination of linear operators. In adjacent subdomains, the reconstruction is the average of linear operators which are almost coincident. Thus, the resulting operators on adjacent domains turn out to be similar, as required.

We now give the pseudocode of the approach discussed thus far.

function NIT-MATODS(x)

Initialize $\widetilde{s}$ as a null matrix;

for $p = 1$ to $n-\overline{n}$ with step ν **do**

for $q = 1$ to $n-\overline{n}$ with step ν **do**

for $i = 1$ to $\overline{n}$ **do**

for $j = 1$ to $\overline{n}$ **do**

$\overline{x}^{(p,q)}_{ri,j} = x_{ri+p,j+q}$;

$\overline{x}^{(p,q)}_{vi,j} = x_{vi+p,j+q}$;

end for

end for

$\overline{s}^{(p,q)}$=MATODS$(\overline{x}^{(p,q)})$

$dim_y = \min\{\overline{n}/\nu, \lceil (i+p)/\nu \rceil, \lceil (n+1-i-p)/\nu \rceil\}$;

$dim_x = \min\{\overline{n}/\nu, \lceil (j+q)/\nu \rceil, \lceil (n+1-j-q)/\nu \rceil\}$;

for $i = 1$ to $\overline{n}$ **do**

for $j = 1$ to $\overline{n}$ **do**

$\widetilde{s}_{ri+p,j+q} = \widetilde{s}_{ri+p,j+q} + \overline{s}^{(p,q)}_{ri,j}/(dim_x \cdot dim_y)$;

$\widetilde{s}_{vi+p,j+q} = \widetilde{s}_{vi+p,j+q} + \overline{s}^{(p,q)}_{vi,j}/(dim_x \cdot dim_y)$;

end for

end for

end for

end for

return $\widetilde{s}$

We refer to this method as the *Not Invariant for Translation MATODS* (NIT-MATODS) algorithm. A correct selection of the parameters ν and $\overline{n}$ can improve the quality of the reconstruction. Clearly, a smaller subdomains size ν corresponds to a more precise reconstruction at the price of an increase of the overall computational cost of the algorithm. The choice of the subimage size $\overline{n}$ is more complex, since as $\overline{n}$ increases, the total number of subimages decreases, while the degree of smoothness between subdomains increases.

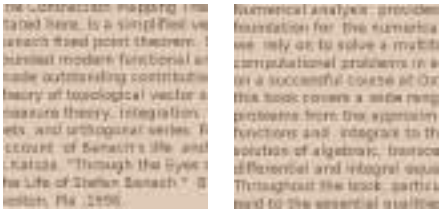
3.7 Experimental results

We implemented both the MATODS and NIT-MATODS algorithms in the C language, and the experiments were run on a linux machine equipped with a $2.80GHz$ processor. First we compared the MATODS algorithm with existing methods for the linear problem, and then we assessed how the NIT-MATODS algorithm performs real ancient documents, comparing it with fast algorithms that were developed by considering other approximated mathematical models.

For restoring color image documents of dimension $\nu = 256$, in most cases the MATODS algorithms requires less than one second, thus we compare it with fast and unsupervised methods, like the FastICA (see also [98, 97, 99, 105, 118]) and the *Symmetric Whitening* (SW) (see also [49, 155, 156]) algorithms. We proceeded as follows. First, we generated a synthetic document from a by applying a given mixing matrix to an uncorrupted source document, using the linear model in equation (3.3). Then we compared the estimated sources with the given source document by means of the *Mean Squared Error* (MSE). In order to compare the FastICA and the SW techniques with our algorithm, at the end of the execution of the FastICA and SW algorithms we transformed the estimated mixture matrices in equivalent one row-sum matrices, as described in Section 3, and then we applied an orthogonal projection operator, so that all the results are in the space $[0,255]^{\nu^2 \times 6}$. As source documents we considered the 256×256 images in Figure 3.15.

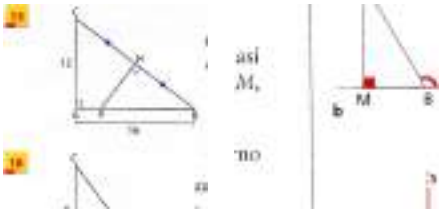
(a) Document 1.

(b) Document 2.



(c) Document 3.

(d) Document 4.



(e) Document 5.

(f) Document 6.

Figure 3.15: Ideal sources.

First, we mixed our documents using the mixture matrices

$$A_R = A_G = A_B = \begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix}. \tag{3.49}$$

In Table 3.4 we report the MSE of the MATODS, FastICA, and SW algorithms with respect to the original documents. Here,

$$\mathrm{MSE}(s_\rho, \widetilde{s}_\rho) = \frac{\|s_\rho - \widetilde{s}_\rho\|_F^2}{3\nu^2},$$

where $\|\cdot\|_F$ denotes the Frobenius norm, $\rho \in \{r, v\}$ the recto or the verso of the document, respectively, s_ρ is the ideal recto or verso, and $\widetilde{s}_\rho$ is an estimate produced by one of the three algorithms we consider. Figure 3.16 (a) shows the mixtures obtained by applying the mask (4.25) to Document 4, and Figure 3.16 (b)-(d) present the result of the MATODS, FastICA, and SW algorithms, respectively. Note that, in Table 3.4, the MATODS algorithm always obtains an error smaller than that of other methods, and in most cases its MSE is negligibly small.

Ideal	MATODS		FastICA		SW	
Document	MSE recto	MSE verso	MSE recto	MSE verso	MSE recto	MSE verso
1	$1.54 \cdot 10^{-10}$	$4.15 \cdot 10^{-11}$	53.27	1.45	0.95	27.79
2	$3.56 \cdot 10^{-9}$	$6.21 \cdot 10^{-10}$	17.71	12.14	25.74	15.75
3	$6.93 \cdot 10^{-8}$	$9.69 \cdot 10^{-8}$	171.23	30.92	3.69	6.25
4	1.09	6.99	20.62	25.67	8.57	58.14
5	$1.25 \cdot 10^{-5}$	$5.04 \cdot 10^{-12}$	4.46	1.78	5.55	3.44
6	$1.13 \cdot 10^{-9}$	$4.54 \cdot 10^{-11}$	130.81	24.84	159.96	67.11

Table 3.4: MSE of the MATODS, FastICA, and SW algorithms using the mixture matrices in (4.25).

(a) Observed document.

(b) Document estimated by MATODS.

(c) Document estimated by FastICA.

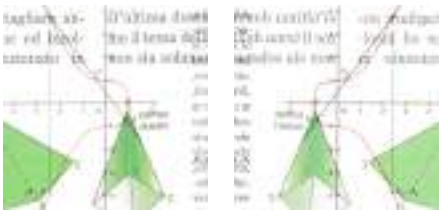
(d) Document estimated by SW.

Figure 3.16: Results for Document 4 mixed using the matrices in (4.25).

Now we consider the mixture matrices

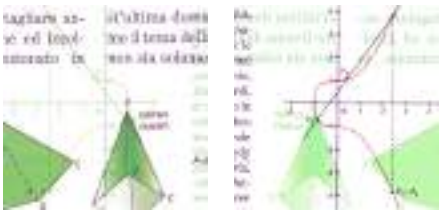
$$A_R = A_G = A_B = \begin{pmatrix} 0.55 & 0.45 \\ 0.45 & 0.55 \end{pmatrix}. \tag{3.50}$$

The mixtures obtained with these matrices will have the recto very similar to the verso, and the problem becomes more difficult to solve, since the matrices in (3.50) are more ill-conditioned than those in (4.25). The MSEs for the three algorithms are given in Table 3.5, Figure 3.17 (a) reports the mixtures obtained by applying the mask (4.25) to Document 6, while Figures 3.17 (b)-(d) present the reconstructions obtained by the MATODS, FastICA, and SW algorithms, respectively.



(a) Observed document.

(b) Document estimated by MATODS.



(c) Document estimated by FastICA.

(d) Document estimated by SW.

Figure 3.17: Results for Document 6 mixed using the matrices in (3.50).

Again we note that in several cases the reconstructions obtained with MATODS substantially correspond to the ideal document. In this case the data documents have higher overlapping levels, thus the approach of FastICA and SW, which force the estimated source overlapping levels to be zero, give results too far from the desired ones. On the other hand, the MATODS algorithm is not affected by the high data overlapping levels, which it estimates correctly. Recall that the MATODS stopping criterion is based on the estimated source overlapping level, whose correct computation requires, in this case, even more accurate source estimates.

Now we consider the case of non-symmetric and partially non-homogeneous mixture matrices, that is, we assume that A_R, A_G, and A_B are non-symmetric matrices and do not always

Ideal	MATODS		FastICA		SW	
Document	MSE recto	MSE verso	MSE recto	MSE verso	MSE recto	MSE verso
1	$5.76 \cdot 10^{-11}$	$5.88 \cdot 10^{-11}$	97.82	27.61	19.33	56.60
2	$4.49 \cdot 10^{-9}$	$8.52 \cdot 10^{-10}$	192.34	105.86	79.44	49.67
3	$6.95 \cdot 10^{-8}$	$9.71 \cdot 10^{-8}$	381.62	331.93	2.88	7.52
4	0.44	4.61	198.80	243.44	28.71	241.31
5	$1.24 \cdot 10^{-5}$	$3.67 \cdot 10^{-12}$	49.58	26.03	19.01	10.96
6	$4.44 \cdot 10^{-10}$	$2.50 \cdot 10^{-11}$	951.51	807.94	628.37	155.55

Table 3.5: MSE of the MATODS, FastICA, and SW algorithms using the mixture matrices in (3.50).

coincide. Let us take

$$A_R = A_B = \begin{pmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{pmatrix}, \quad A_G = \begin{pmatrix} 0.6 & 0.4 \\ 0.3 & 0.7 \end{pmatrix}. \tag{3.51}$$

The MSE of the algorithms we consider is presented in Table 3.6. In this case, the results obtained using the SW algorithm are not optimal, because the algorithm imposes a symmetry constraint on the estimated mixture matrices. On the other hand, the FastICA algorithm gives better results than in the previous case, because the condition numbers of the matrices in (3.51) are smaller.

Ideal	MATODS		FastICA		SW	
Document	MSE recto	MSE verso	MSE recto	MSE verso	MSE recto	MSE verso
1	$1.28 \cdot 10^{-10}$	$4.56 \cdot 10^{-11}$	43.71	2.27	43.79	54.42
2	$3.51 \cdot 10^{-9}$	$6.07 \cdot 10^{-10}$	19.90	17.64	84.33	50.01
3	$6.95 \cdot 10^{-8}$	$9.71 \cdot 10^{-8}$	175.96	68.05	24.94	16.33
4	0.78	5.93	18.94	33.65	18.37	60.96
5	$1.24 \cdot 10^{-5}$	$3.90 \cdot 10^{-12}$	3.33	2.91	28.21	17.33
6	$1.43 \cdot 10^{-10}$	$1.76 \cdot 10^{-11}$	258.04	96.14	509.88	129.25

Table 3.6: MSE of the MATODS, FastICA, and SW algorithms using the mixture matrices in (3.51).

Now we illustrate the non-symmetric and non-homogeneous case by using the matrices

$$A_R = \begin{pmatrix} 0.6 & 0.4 \\ 0.3 & 0.7 \end{pmatrix}, A_G = \begin{pmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{pmatrix}, A_B = \begin{pmatrix} 0.55 & 0.45 \\ 0.4 & 0.6 \end{pmatrix}. \quad (3.52)$$

The MSE for the three algorithms is reported in Table 3.7.

Ideal	MATODS		FastICA		SW	
Document	MSE recto	MSE verso	MSE recto	MSE verso	MSE recto	MSE verso
1	$1.22 \cdot 10^{-10}$	$4.27 \cdot 10^{-11}$	28.98	0.29	35.03	39.82
2	$3.54 \cdot 10^{-9}$	$6.16 \cdot 10^{-10}$	38.76	18.77	53.57	37.16
3	$6.94 \cdot 10^{-8}$	$9.71 \cdot 10^{-8}$	212.02	89.25	12.08	25.74
4	0.63	5.35	33.75	60.95	25.14	169.47
5	$1.24 \cdot 10^{-5}$	$5.10 \cdot 10^{-12}$	7.05	1.88	10.78	9.01
6	$6.76 \cdot 10^{-10}$	$4.87 \cdot 10^{-11}$	470.29	181.33	202.20	94.64

Table 3.7: MSE of the MATODS, FastICA, and SW algorithms using the mixture matrices in (3.52).

Note that, in general, the results obtained by the SW and FastICA algorithms are very similar, which confirms what observed in [156]. Moreover, the estimation of the source overlapping level is very useful for a correct reconstruction of the original sources. In all the cases we examine, the MATODS algorithm obtains the best results.

The MATODS algorithm has been developed to separate linearly mixed components, and does not remove deterioration phenomena, like noise in the data. In noisy document the brightest value does not necessarily coincide with that of the background, and in order to handle noisy data the MATODS algorithm, instead of computing the maximum value of the light intensity, uses the statistical mode of the noisy document, to which it subtracts $4\sigma^2$, in order to exclude noise tails. In Figure 3.18 (a) we present Document 1 mixed with the mixture matrices in (4.25) and corrupted with additive white independent Gaussian noise with variance $\sigma^2 = 4$ and mean zero. In Figure 3.18 (b) we show the result of MATODS, while in Figures 3.18 (c) and (d) we present those of FastICA and SW. Note that the MATODS algorithm separates the sources better than FastICA and SW, but does not reduce the noise disturbance.

Now we illustrate how the NIT-MATODS algorithm restores real ancient documents. In this case, we take some practical measures: we compare the maximum light intensity of the recto

(a) Observed document.

(b) Document estimated by MATODS (recto MSE 9.60, verso MSE 8.06).

(c) Document estimated by FastICA (recto MSE 60.29, verso MSE 8.88).

(d) Document estimated by SW (recto MSE 9.73, verso MSE 37.85).

Figure 3.18: Results for Document 1 mixed using the mixture matrices in (4.25) and corrupted with noise with variance $\sigma^2 = 4$.

with that of the verso, and if the two values do not coincide, we add a constant to the light intensities of the darker image, in order to reduce the difference between the background color of the two sides. This is justified by the fact that the color of the paper has to be the same on both sides of the paper. Since images of real documents present noise degradation phenomena, we compute the statistical mode of the document instead of the maximum. We assume that the determinant of the overlapping matrix of the observed data, C, which corresponds to each channel of every involved subimage, is zero when $\det(C)/\|C\|_\infty \leq \overline{\varepsilon}$, where $\overline{\varepsilon}$ is an accuracy threshold and $\|\cdot\|_\infty$ is the infinity norm. In the following experiments we deal with documents of size $n = 512$, and we set the subimage dimension $\overline{n}$ to 128, while the dimension of the subdomains is fixed to $\nu = 16$ pixels. These values were chosen empirically in order to obtain a reasonable trade-off between the quality of the result and the required computational time. Experimentally we noticed that, as ν increases, both the execution time and the quality of the result decrease, whereas as $\overline{n}$ increases, while remaining below $n/4$, both the execution time and the quality of the result increase. Although the choice of the values of ν and $\overline{n}$ is tricky for the data in exam, we experimentally found some values that give good results in the general case.

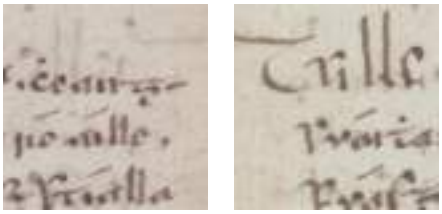
A more exhaustive analysis of the choice of these parameters would be of interest. In the

examples presented here, the average execution time of the the NIT-MATODS algorithm is 59.72 seconds.

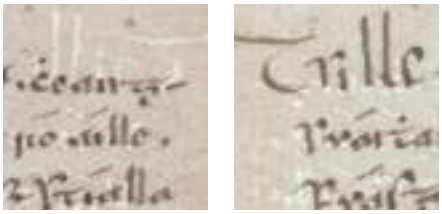
The reconstructions obtained by the NIT-MATODS algorithm are compared with those obtained considering the stationary linear model, the nonlinear stationary model in (3.1), and the non-stationary nonlinear model in (3.2). In particular, for the reconstructions based on the stationary linear model we use the MATODS algorithm, to treat the stationary nonlinear model we use the algorithm proposed in [119], and for the non-stationary nonlinear model we consider the algorithm in [157]. From the initials of the authors, we refer to the algorithm in [119] as the MSGT algorithm, and to that in [157] as the TSS algorithm. These methods were chosen because their computational cost is similar to that of the NIT-MATODS algorithm.

The data documents in Figures 3.19–3.21 (a) are taken from the database created as part of the *Irish Script on Screen* (ISOS) project of the School of Celtic Studies of the Dublin Institute for Advanced Studies, in conjunction with the SIGMEDIA group of the Department of Electrical and Electronic Engineering at Trinity College Dublin (see [142]). This database contains ancient documents affected by bleed-through. The results of MATODS are presented in Figures 3.19–3.21 (b), those of MSGT in Figures 3.19–3.21 (c), and those of TSS in Figures 3.19–3.21 (d). In Figures 3.19–3.21 (e) we present the reconstruction of the NIT-MATODS algorithm with $\nu = 4$ and $\overline{n} = 32$, in order to show how NIT-MATODS works using non-optimal parameters. Finally, the results obtained by NIT-MATODS using the optimal parameters $\nu = 16$ and $\overline{n} = 128$ are presented in Figures 3.19–3.21 (f).

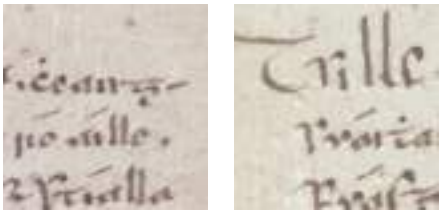
We note that NIT-MATODS improves upon the results of the MSGT and TSS algorithms. This is due to the fact that MSGT and TSS, in order to lower their computational cost, reduce the quality of the reconstructions. In order to obtain more accurate reconstructions from a non-stationary and nonlinear model, more expensive regularization technique may be adopted (see also [69, 155]).



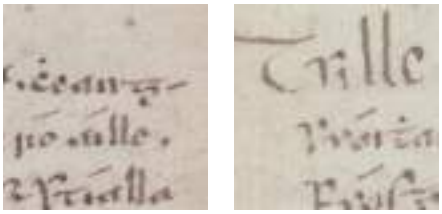
(a) Observed document.



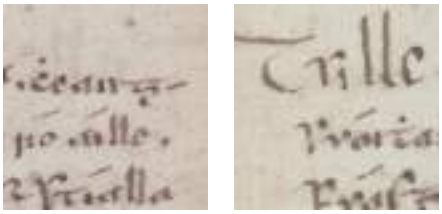
(b) Document estimated by MATODS.



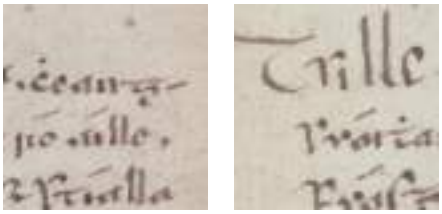
(c) Document estimated by MSGT.



(d) Document estimated by TSS.

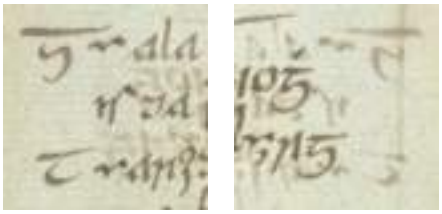


(e) Document estimated by NIT-MATODS ($\nu = 4, \bar{n} = 32$).

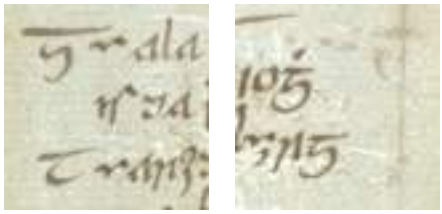


(f) Document estimated by NIT-MATODS ($\nu = 16, \bar{n} = 128$).

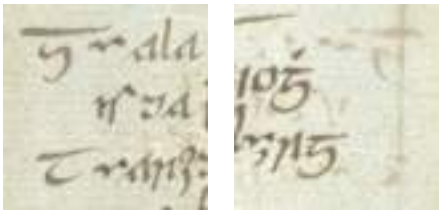
Figure 3.19: First ISOS document.



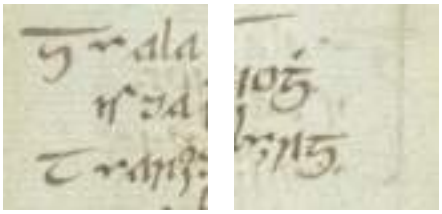
(a) Observed document.



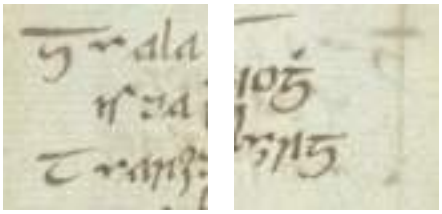
(b) Document estimated by MATODS.



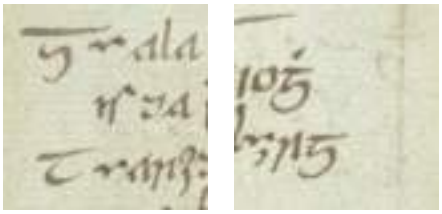
(c) Document estimated by MSGT.



(d) Document estimated by TSS.

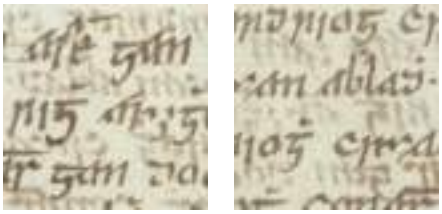


(e) Document estimated by NIT-MATODS ($\nu = 4, \bar{n} = 32$).

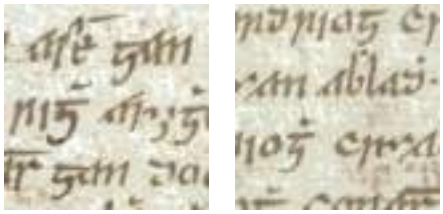


(f) Document estimated by NIT-MATODS ($\nu = 16, \bar{n} = 128$).

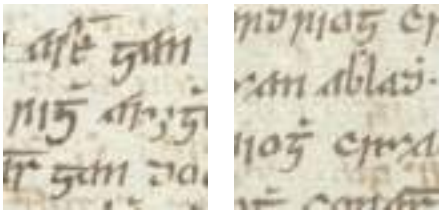
Figure 3.20: Second ISOS document.



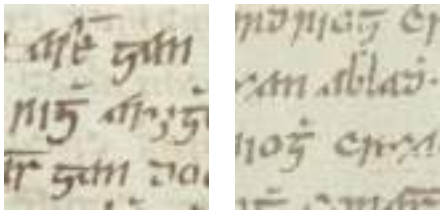
(a) Observed document.



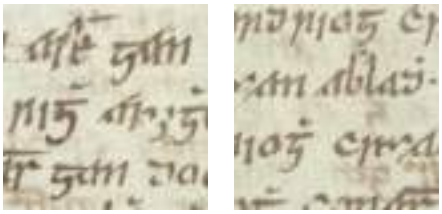
(b) Document estimated by MATODS.



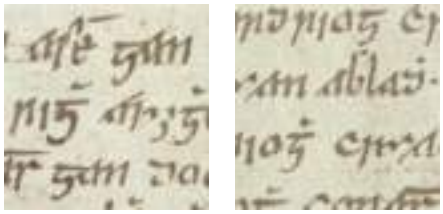
(c) Document estimated by MSGT.



(d) Document estimated by TSS.



(e) Document estimated by NIT-MATODS ($\nu = 4, \bar{n} = 32$).



(f) Document estimated by NIT-MATODS ($\nu = 16, \bar{n} = 128$).

Figure 3.21: Third ISOS document.

Chapter 4

Document restoration based on edge estimation

In Section 4.1 we introduce the concept of discrete derivative in an image. In Section 4.2 we develop the ZEODS algorithm to deal with the linear problem. In Section 4.3 we compare experimentally the ZEODS algorithm with other fast and unsupervised methods existing in the literature.

4.1 The discrete derivative in an image

We call *clique* the set of pixels on which the finite difference of first order is well-defined. The vertical cliques are of the type

$$c = \{(i, j), (i+1, j)\}, \tag{4.1}$$

while the horizontal cliques have the form

$$c = \{(i, j), (i, j+1)\}. \tag{4.2}$$

We denote by C the set of all cliques. Note that $|C| = 2nm - m - n$, where C denotes the cardinality of C.

Given a vertical clique $c = \{(i, j), (i+1, j)\}$, the finite difference operator on it is $\Delta_c \widehat{\mathbf{x}} = \widehat{x}_{i,j} - \widehat{x}_{i+1,j}$. Moreover, given a horizontal clique $c = \{(i, j), (i, j+1)\}$, the associated finite difference

operator is let $\Delta_c \widehat{\mathbf{x}} = \widehat{x}_{i,j} - \widehat{x}_{i,j+1}$. We consider the linear operator $D \in \mathbb{R}^{|C| \times nm}$. Note that, in this matrix, every row index corresponds to a clique, while every column index corresponds to a pixel. To every row it is possible to associate a vertical or horizontal clique. Then, if we consider a vertical clique $c = \{(i,j),(i+1,j)\}$, we get

$$D_{c,(l,k)} = \begin{cases} 1, & \text{if } (l,k) = (i,j), \\ -1, & \text{if } (l,k) = (i+1,j), \\ 0, & \text{otherwise;} \end{cases}$$

and, if $c = \{(i,j),(i,j+1)\}$ is a horizontal clique, we have

$$D_{c,(l,k)} = \begin{cases} 1, & \text{if } (l,k) = (i,j), \\ -1, & \text{if } (l,k) = (i,j+1), \\ 0, & \text{otherwise.} \end{cases}$$

Let $x \in \mathbb{R}^{|C| \times 2}$ be the *data derivative document matrix* defined by

$$x = D\widehat{x}. \tag{4.3}$$

Analogously, the *source derivative matrix* $s \in \mathbb{R}^{|C| \times 2}$ is defined by

$$s = D\widehat{s}. \tag{4.4}$$

Notice that the involved images contain letters. If we assume that the colors of the letters and of the background are uniform, then the finite differences are null, while they are different from zero in correspondence with the edges of the letters.

From (3.3), (4.3) and (4.4) we deduce

$$x^T = \widehat{x}^T D^T = A\widehat{s}^T D^T = As^T. \tag{4.5}$$

Note that the linear model obtained by considering the data document derivative matrix and the source derivative matrix is equal to that obtained by treating the data document and the source document in (3.3).

Analogously as in [27], here we define the following 2×2 *data derivative overlapping matrix* of the observed data:

$$C = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} = x^T x = \begin{bmatrix} x_r^T \cdot x_r & x_r^T \cdot x_v \\ x_v^T \cdot x_r & x_v^T \cdot x_v \end{bmatrix}. \tag{4.6}$$

The matrix C indicates how much the edges of the letters in the front overlap with those of the back. Indeed, in our case, the data derivative overlapping matrix is always nonnegative, and is diagonal if and only if there is no overlapping of the edges of text from the recto to the verso of the document. In particular we refer to the entries $d = c_{12} = c_{21}$ as the *data derivative overlapping level*.

The *source derivative overlapping matrix* can be defined similarly as

$$P = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} = s^T s = \begin{bmatrix} s_r^T \cdot s_r & s_r^T \cdot s_v \\ s_v^T \cdot s_r & s_v^T \cdot s_v \end{bmatrix}.$$

It is not difficult to see that the matrices C and P are symmetric and positive semidefinite. We refer to the value

$$k = p_{12} = p_{21} = s_r^T \cdot s_v \tag{4.7}$$

as the *source derivative overlapping level*. We assume that $k = 0$, that is the edges of the recto of the document do not overlap with those of the verso.

4.2 A technique for solving the linear problem

As in [27], we define a *symmetric factorization* of a symmetric and positive-definite matrix $H \in \mathbb{R}^{n \times n}$ as an expression of the type $H = ZZ^T$, where $Z \in \mathbb{R}^{n \times n}$. Note that, given an orthogonal matrix $Q \in \mathbb{R}^{n \times n}$ and a symmetric factorization of the type $H = ZZ^T$, then $ZQ(ZQ)^T$ is a symmetric factorization of H too. Furthermore, if we pick any two symmetric factorizations $H = Z_1 Z_1^T$ and $H = Z_2 Z_2^T$, then there exists an orthogonal matrix $Q \in \mathbb{R}^{n \times n}$ with $Z_1 = Z_2 Q$ (see, e.g., [26]).

In the 2×2 case, the set of the orthogonal matrices is the union of all rotations and reflections in $\mathbb{R}^2$, which are expressed as

$$Q^1(\theta) = \begin{bmatrix} \sin\theta & -\cos\theta \\ \cos\theta & \sin\theta \end{bmatrix} \quad \text{and} \quad Q^{-1}(\theta) = \begin{bmatrix} \sin\theta & \cos\theta \\ \cos\theta & -\sin\theta \end{bmatrix}, \tag{4.8}$$

respectively, as θ varies in $]0,2\pi]$. Since $C = C^{1/2}(C^{1/2})^T = C^{1/2}C^{1/2}$ is a symmetric factorization of C, then all factorizations of C are given by

$$Z^{(\iota)}(\theta) = C^{1/2}Q^{(\iota)}(\theta) = \begin{bmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{bmatrix} Q^{(\iota)}(\theta) = \begin{bmatrix} z_{11}^{(\iota)}(\theta) & z_{12}^{(\iota)}(\theta) \\ z_{21}^{(\iota)}(\theta) & z_{22}^{(\iota)}(\theta) \end{bmatrix}, \tag{4.9}$$

where $\theta \in]0,2\pi]$ and $\iota \in \{-1,1\}$. In particular, we get

$$z_{11}^{(1)}(\theta) = z_{11}^{(-1)}(\theta), \quad z_{12}^{(1)}(\theta) = -z_{12}^{(-1)}(\theta), \quad z_{21}^{(1)}(\theta) = z_{21}^{(-1)}(\theta), \quad z_{22}^{(1)}(\theta) = -z_{22}^{(-1)}(\theta). \tag{4.10}$$

We assume that

$$C = x^T x = A s^T s A^T = A\widetilde{P}A^T, \tag{4.11}$$

where $\widetilde{P}$ is a symmetric and positive-definite estimate of the source derivative overlapping matrix P. In $\widetilde{P}$ we put

$$\widetilde{p}_{12} = \widetilde{p}_{21} = 0. \tag{4.12}$$

Observe that we do not assign a value to $\widetilde{p}_{11}$ and $\widetilde{p}_{22}$, as they will be determined later by imposing that the estimated mixture matrix is one row-sum. Let

$$\widetilde{P} = YY^T \tag{4.13}$$

be a symmetric factorization, where Y is a nonsingular matrix that satisfies

$$y_{11}y_{21} + y_{12}y_{22} = 0, \tag{4.14}$$

thanks to (4.12). From (4.11) and (4.13) we get

$$C \quad = \quad AYY^TA^T = AY(AY)^T,$$

that is, AY is a factorization of C. For every given choice of $\theta \in]0,2\pi]$ and $\iota \in \{-1,1\}$, we define an estimate $\widetilde{A}^{(\iota)}(\theta)$ of the mixture matrix A as a matrix such that $\widetilde{A}^{(\iota)}(\theta) = Z^{(\iota)}(\theta)Y^{-1}$, where $Z^{(\iota)}(\theta)$ is as in (4.9). We have

$$\begin{aligned} a_{11}^{(\iota)}(\theta) &= \frac{z_{11}^{(\iota)}(\theta)y_{22} - z_{12}^{(\iota)}(\theta)y_{21}}{y_{11}y_{22} - y_{21}y_{12}}, \qquad a_{12}^{(\iota)}(\theta) = \frac{z_{12}^{(\iota)}(\theta)y_{11} - z_{11}^{(\iota)}(\theta)y_{12}}{y_{11}y_{22} - y_{21}y_{12}}, \\ a_{21}^{(\iota)}(\theta) &= \frac{z_{21}^{(\iota)}(\theta)y_{22} - z_{22}^{(\iota)}(\theta)y_{21}}{y_{11}y_{22} - y_{21}y_{12}}, \qquad a_{22}^{(\iota)}(\theta) = \frac{z_{22}^{(\iota)}(\theta)y_{11} - z_{21}^{(\iota)}(\theta)y_{12}}{y_{11}y_{22} - y_{21}y_{12}}, \end{aligned} \tag{4.15}$$

and by imposing that $\widetilde{A}^{(\iota)}(\theta)$ satisfies the one row-sum condition in (3.5), we get

$$\begin{aligned} z_{11}^{(\iota)}(\theta)y_{22} - z_{12}^{(\iota)}(\theta)y_{21} + z_{12}^{(\iota)}(\theta)y_{11} - z_{11}^{(\iota)}(\theta)y_{12} &= y_{11}y_{22} - y_{21}y_{12}, \\ z_{21}^{(\iota)}(\theta)y_{22} - z_{22}^{(\iota)}(\theta)y_{21} + z_{22}^{(\iota)}(\theta)y_{11} - z_{21}^{(\iota)}(\theta)y_{12} &= y_{11}y_{22} - y_{21}y_{12}. \end{aligned} \tag{4.16}$$

Thus, the matrix Y fulfils the conditions in equations (4.14) and (4.16). The nonlinear system given by the equations (4.14) and (4.16) admits infinitely many solutions. For the sake of convenience, we choose the solution

$$\begin{aligned} y_{11} = \frac{\det C}{(z_{22}^{(\iota)}(\theta) - z_{12}^{(\iota)}(\theta))\det Z^{(\iota)}(\theta)}, \quad y_{12} &= 0, \\ y_{21} = 0, \qquad\qquad\qquad\qquad\qquad\quad y_{22} &= \frac{\det Z^{(\iota)}(\theta)}{z_{11}^{(\iota)}(\theta) - z_{21}^{(\iota)}(\theta)}. \end{aligned} \tag{4.17}$$

This choice has several consequences. First, from (4.10) and (4.15) we obtain that $\widetilde{A}^{(1)}(\theta) = \widetilde{A}^{(-1)}(\theta)$ for all $\theta \in]0, 2\pi]$. Moreover, from equations (4.8) and (4.9) we get that $Z(\theta) = -Z(\theta + \pi)$, for $\theta \in]0, \pi]$, and hence from (4.15) and (4.17) we deduce that

$$\widetilde{A}(\theta) = \widetilde{A}(\theta + \pi), \tag{4.18}$$

for each $\theta \in]0, \pi]$.

So, in the following we consider only the case $\iota = 1$, we put $\widetilde{A}(\theta) = \widetilde{A}^{(1)}(\theta)$ and $Z(\theta) = Z^{(1)}(\theta)$ for each $\theta \in]0, \pi]$, and in general we consider only the values of θ belonging to $]0, \pi]$.

Recall that Y must be non-singular, since Y realizes a symmetric factorization of the non-singular matrix P.

Moreover, the equations in (4.17) are well defined if $z_{11}(\theta) \neq z_{21}(\theta)$ and $z_{12}(\theta) \neq z_{22}(\theta)$. In [26] we prove that $z_{11}(\theta) = z_{21}(\theta)$ or $z_{12}(\theta) = z_{22}(\theta)$ when θ assumes the values $\varphi + t\frac{\pi}{2}$, with $t \in \mathbb{Z}$ and

$$\varphi = \begin{cases} \arctan\left(\dfrac{\rho_{22} - \rho_{12}}{\rho_{11} - \rho_{21}}\right), & \text{if } \rho_{11} \neq \rho_{21}, \\ \dfrac{\pi}{2}, & \text{if } \rho_{11} = \rho_{21}, \end{cases} \tag{4.19}$$

where $\rho_{i,j}$, $i, j = 1, 2$, are the entries of the matrix $C^{1/2}$.

For any $\theta \in]\varphi, \varphi + \frac{\pi}{2}[\cup]\varphi + \frac{\pi}{2}, \varphi + \pi[$, we get that an estimate of the ideal sources s is given by

$$\widetilde{s}(\theta)^T = \begin{bmatrix} \widetilde{s}_r(\theta) & \widetilde{s}_v(\theta) \end{bmatrix}^T = \widetilde{A}^{-1}(\theta)x^T, \tag{4.20}$$

which, together with the fact that $\widetilde{A}^{-1}(\theta) = \widetilde{A}^{1}(\theta) = Z^{(1)}(\theta)Y^{-1}$ and (4.16), yields

$$\begin{aligned} \widetilde{s}_r(\theta) &= -\frac{z_{22}(\theta)}{z_{12}(\theta)-z_{22}(\theta)}x_r + \frac{z_{12}(\theta)}{z_{12}(\theta)-z_{22}(\theta)}x_v; \\ \widetilde{s}_v(\theta) &= -\frac{z_{21}(\theta)}{z_{11}(\theta)-z_{21}(\theta)}x_r + \frac{z_{11}(\theta)}{z_{11}(\theta)-z_{21}(\theta)}x_v. \end{aligned} \tag{4.21}$$

As we supposed that the derivatives of our estimated sources take values between 0 and $2m$, where m is the maximum value of the observed image, we take the orthogonal projection of the estimate $s_t(\theta)$ on the space $[0,2m]^{nm\times 2}$ with respect to the Frobenius norm. Namely, we apply to the estimate of the sources the function that maps a vector $s \in \mathbb{R}^{nm}$ to the nm-dimensional vector $\tau(s)$, whose elements are given by

$$(\tau(s))_i = \begin{cases} 0, & \text{if } s_i \leq 0, \\ s_i, & \text{if } 0 < s_i \leq 2m, \\ 2m, & \text{if } s_i > 2m, \end{cases} \qquad i = 1,\ldots,nm. \tag{4.22}$$

By this transformation, the projections of the estimated source derivative images $\tau(\widetilde{s}_{r,t}(\theta))$ and $\tau(\widetilde{s}_{v,t}(\theta))$ turn to be nonnegative (see also [42, 50, 74, 134]). From now on, we consider the projections above as the new estimates of the derivatives of the sources. Thus, among the possible values of θ in $]\varphi, \varphi+\frac{\pi}{2}[\cup]\varphi+\frac{\pi}{2}, \varphi+\pi[$, we find a value $\widetilde{\theta}$ that minimizes the *objective function*

$$g(\theta, C) = \tau(\widetilde{s}_r(\theta))^T \cdot \tau(\widetilde{s}_v(\theta)). \tag{4.23}$$

Observe that from (4.18) and (4.20) it follows that the function g is periodic in the variable θ with period π. The function g is minimized by means of the algorithm given in [27].

The steps of the algorithm described in this section are illustrated as follows.

function ZEODS$(\widehat{x})$

determine the maximum value m of $\widehat{x}$;

$x = D\widehat{x}$;

$C = x^T x$;

$\widetilde{\theta} = argmin($**function** $g(\cdot, C))$;

$Z(\widetilde{\theta}) = C^{1/2} Q_1(\widetilde{\theta})$;

compute $\widetilde{s}_r(\widetilde{\theta})$ and $\widetilde{s}_v(\widetilde{\theta})$ as in (4.21);

return $D^{-1}\tau(\widetilde{s}(\widetilde{\theta}))$

The function $g(\cdot,\cdot)$ is computed as follows:

function $g(\theta, C)$

$Z(\theta) = C^{1/2} Q^1(\theta)$;

compute $\widetilde{s_r}(\theta)$ and $\widetilde{s_v}(\theta)$ as in (4.21);

return $(\tau(\widetilde{s_r}(\theta)))^T \cdot \tau(\widetilde{s_v}(\theta))$

We refer to this method as the ZEODS algorithm, which is a parameter-free technique, and thus unsupervised.

4.3 Experimental results

We have used ideal images, from which the observed documents have been synthetically constructed from suitable mixture matrices. The ideal images used for the tests are represented in Figures 4.1 and 4.2.

In our tests, we have used both symmetric and asymmetric mixture matrices. In the following subsections, the obtained results are explained and compared with other techniques both computationally and from the visual point of view. We examined RGB color images. The channels R, G and B were treated separately.

4.3.1 Case 1: First symmetric matrix

The first case we investigate is a symmetric mixture matrix. For each channel R, G and B, the related matrices are

$$A_R = \begin{pmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{pmatrix}, A_G = \begin{pmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{pmatrix}, A_B = \begin{pmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{pmatrix}. \tag{4.24}$$

Now we see the behavior of the presented algorithms. We consider the ideal images in Figure 4.3, and using the above indicated mixture matrices, we synthetically obtain the degraded images in Figure 4.4.

By applying the algorithms we get, as estimates, the results in Figures 4.5-4.10.

In Table 4.1 we present the mean square errors with respect to the original documents obtained by means of the aforementioned algorithms for estimating the recto and the verso of Figure

(a) original recto

(b) original verso

(c) original recto

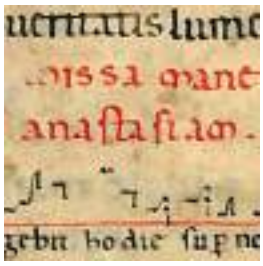
(d) original verso

(e) original recto

(f) original verso

Figure 4.1: Ideal images

(a) original recto

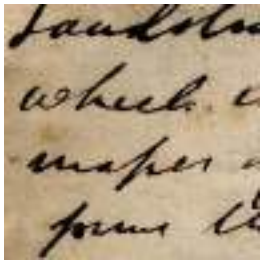


(b) original verso

(c) original recto

(d) original verso

(e) original recto



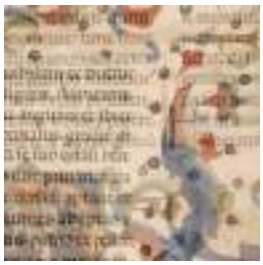
(f) original verso

Figure 4.2: Ideal images

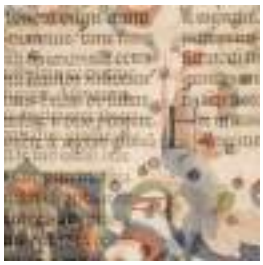
(a) original recto

(b) original verso

Figure 4.3: Ideal images



(a) degraded recto



(b) degraded verso

Figure 4.4: Degraded images

(a) recto estimated by ZEODS

(b) verso estimated by ZEODS

Figure 4.5: Estimates by ZEODS

(a) recto estimated by MATODS

(b) verso estimated by MATODS

Figure 4.6: Estimates by MATODS

(a) recto estimated by FastIca

(b) verso estimated by FastIca

Figure 4.7: Estimates by FastIca

(a) recto estimated by Symmetric Whitening

(b) verso estimated by Symmetric Whitening

Figure 4.8: Estimates by Symmetric Whitening

(a) recto estimated by Whitening

(b) verso estimated by Whitening

Figure 4.9: Estimates by Whitening

(a) recto estimated by PCA

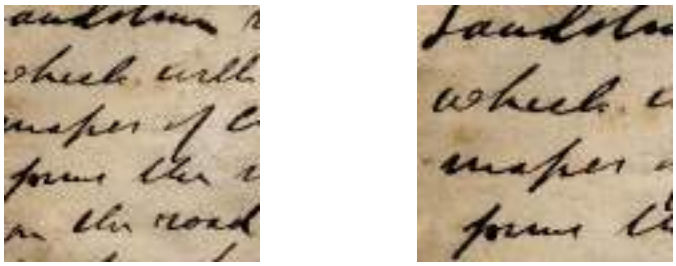
(b) verso estimated by PCA

Figure 4.10: Estimates by PCA

4.3. Now we consider the following ideal images in Figure 4.11. Using the above indicated mix-

Used Technique	MSE Recto	MSE Verso	MSE of A
ZEODS	5.0766	0.6228	$1.020 \cdot 10^{-4}$
MATODS	12.5173	49.0506	0.0011
FASTICA	58.2382	212.8663	0.0546
Symmetric Whitening	428.0422	373.6753	0.00183
Whitening	$7.7086 \cdot 10^3$	$6.2362 \cdot 10^3$	0.3561
PCA	$1.4943 \cdot 10^4$	$5.2861 \cdot 10^3$	0.3770

Table 4.1: Errors of the algorithms by using the mixture matrix in (4.24).



(a) original recto (b) original verso

Figure 4.11: Ideal images

ture matrices, we synthetically obtain the degraded images in Figure 4.12.

By applying the algorithms we obtain, as estimates, the results in Figures 4.13-4.18.

In Table 4.2 we give the mean square errors with respect to the original documents obtained by means of the above algorithms for the estimates of the recto and the verso of Figure 4.11.

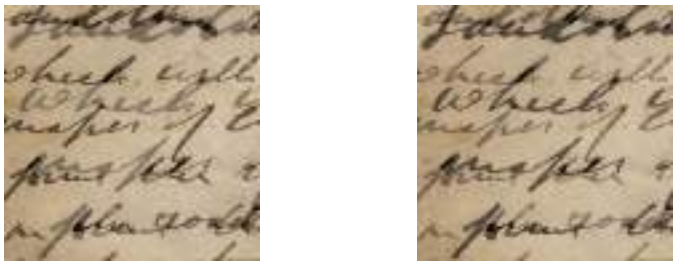
We consider the ideal images in Figure 4.19.

Using the above indicated mixture matrices, we synthetically obtain the degraded images in Figure 4.20.

By applying the algorithms we obtain, as estimates, the results in Figures 4.21-4.26.

In Table 4.3 we present the mean square errors with respect to the original documents obtained by means of the above algorithms for the estimate of the recto and the verso of Figure 4.19. We consider the ideal images in Figure 4.27.

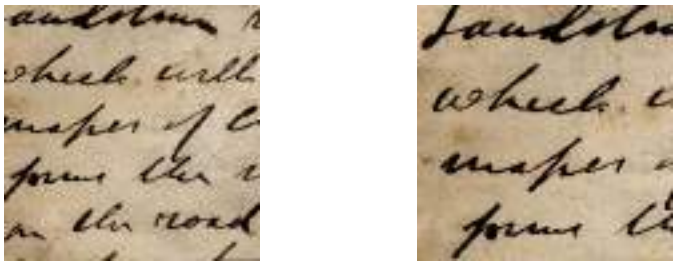
Using the above indicated mixture matrices, we synthetically obtain the degraded images in



(a) degraded recto

(b) degraded verso

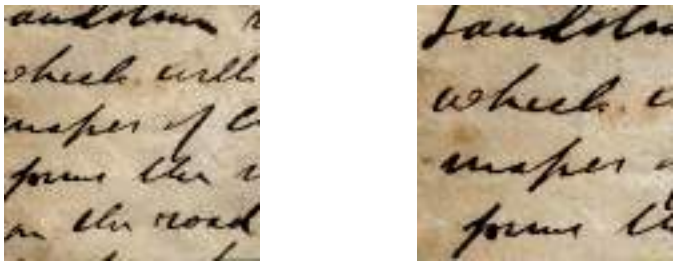
Figure 4.12: Degraded images



(a) recto estimated by ZEODS

(b) verso estimated by ZEODS

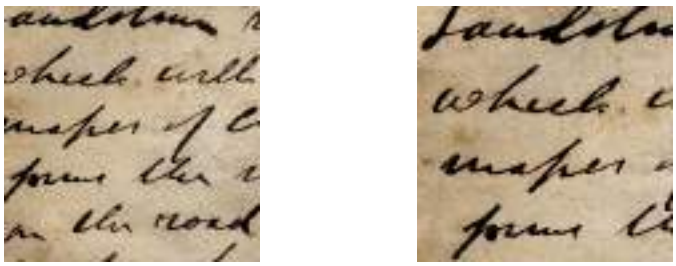
Figure 4.13: Estimates by ZEODS



(a) recto estimated by MATODS

(b) verso estimated by MATODS

Figure 4.14: Estimates by MATODS

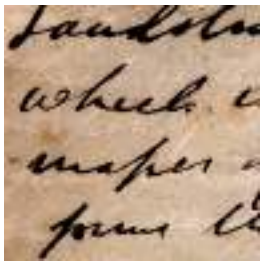


(a) recto estimated by FastIca

(b) verso estimated by FastIca

Figure 4.15: Estimates by FastIca

(a) recto estimated by Symmetric Whitening



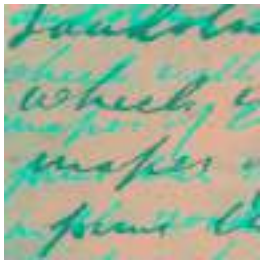
(b) verso estimated by Symmetric Whitening

Figure 4.16: Estimates by Symmetric Whitening

(a) recto estimated by Whitening

(b) verso estimated by Whitening

Figure 4.17: Estimates by Whitening



(a) recto estimated by PCA

(b) verso estimated by PCA

Figure 4.18: Estimates by PCA

(a) original recto

(b) original verso

Figure 4.19: Ideal images

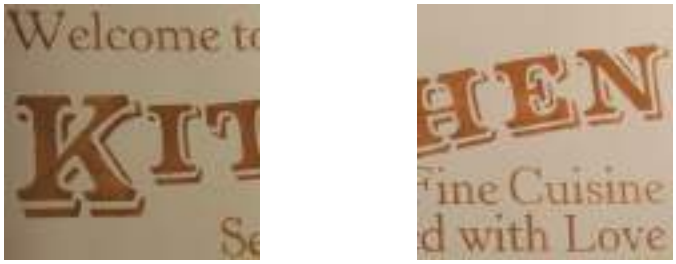
Used Technique	MSE Recto	MSE Verso	MSE of A
ZEODS	1.7592	0.4784	$5.4688 \cdot 10^{-5}$
MATODS	25.6900	52.0605	$1.2550 \cdot 10^{-4}$
FASTICA	3.3840	3.4516	0.0095
Symmetric Whitening	74.7709	80.8914	0.0110
Whitening	$8.4391 \cdot 10^{3}$	$5.98950 \cdot 10^{3}$	0.4561
PCA	$1.4068 \cdot 10^{4}$	$3.9386 \cdot 10^{3}$	0.4225

Table 4.2: Errors of the algorithms by using the mixture matrix in (4.24).

(a) degraded recto

(b) degraded verso

Figure 4.20: Degraded images



(a) recto estimated by ZEODS

(b) verso estimated by ZEODS

Figure 4.21: Estimates by ZEODS

(a) recto estimated by MATODS

(b) verso estimated by MATODS

Figure 4.22: Estimates by MATODS

(a) recto estimated by FastIca

(b) verso estimated by FastIca

Figure 4.23: Estimates by FastIca

(a) recto estimated by Symmetric Whitening

(b) verso estimated by Symmetric Whitening

Figure 4.24: Estimates by Symmetric Whitening

(a) recto estimated by Whitening

(b) verso estimated by Whitening

Figure 4.25: Estimates by Whitening

(a) recto estimated by PCA

(b) verso estimated by PCA

Figure 4.26: Estimates by PCA

Used Technique	MSE Recto	MSE Verso	MSE of A
ZEODS	0.8752	0.6474	$4.4766 \cdot 10^{-5}$
MATODS	172.146	180.0660	$2.8565 \cdot 10^{-4}$
FASTICA	12.1634	43.6463	0.0395
Symmetric Whitening	261.5776	259.4301	0.00168
Whitening	$3.5723 \cdot 10^{3}$	$1.5907 \cdot 10^{3}$	0.4596
PCA	$5.9609 \cdot 10^{3}$	$1.4281 \cdot 10^{3}$	0.4242

Table 4.3: Errors of the algorithms by using the mixture matrix in (4.24).

(a) original recto

(b) original verso

Figure 4.27: Ideal images

Figure 4.28.

(a) degraded recto

(b) degraded verso

Figure 4.28: Degraded images

By applying the algorithms we obtain, as estimates, the results in Figures 4.29-4.34.

(a) recto estimated by ZEODS

(b) verso estimated by ZEODS

Figure 4.29: Estimates by ZEODS

In Table 4.4 we indicate the mean square errors with respect to the original documents obtained by means of the above algorithms for the estimate of the recto and the verso of Figure 4.27.

We consider the ideal images in Figure 4.35.

Using the above indicated mixture matrices, we synthetically obtain the degraded images in Figure 4.36.

By applying the algorithms we obtain, as estimates, the results in Figures 4.37-4.42.

In Table 4.5 we present the mean square errors with respect to the original documents obtained by means of the above algorithms for the estimate of the recto and the verso of Figure 4.35.

As we can observe from the results of the previous subsection, the proposed and implemented ZEODS method obtains better results than algorithms FastIca, PCA, Whitening and Symmetric

(a) recto estimated by MATODS

(b) verso estimated by MATODS

Figure 4.30: Estimates by MATODS

(a) recto estimated by FastIca

(b) verso estimated by FastIca

Figure 4.31: Estimates by FastIca

(a) recto estimated by Symmetric Whitening

(b) verso estimated by Symmetric Whitening

Figure 4.32: Estimates by Symmetric Whitening

(a) recto estimated by Whitening

(b) verso estimated by Whitening

Figure 4.33: Estimates by Whitening

(a) recto estimated by PCA

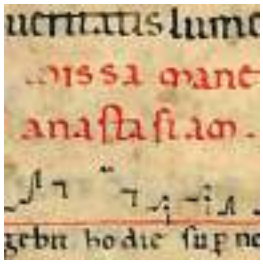
(b) verso estimated by PCA

Figure 4.34: Estimates by PCA

Used Technique	MSE Recto	MSE Verso	MSE of A
ZEODS	0.7829	0.5673	$1.095 \cdot 10^{-4}$
MATODS	0.9015	10.3131	0.0014
FASTICA	1.0849	0.6707	0.0136
Symmetric Whitening	8.5123	12.2799	0.0085
Whitening	$1.9433 \cdot 10^3$	$1.2006 \cdot 10^3$	0.4548
PCA	$2.9914 \cdot 10^3$	$716.5649 \cdot 10^3$	0.4234

Table 4.4: Errors of the algorithms by using the mixture matrix in (4.24).

(a) original recto



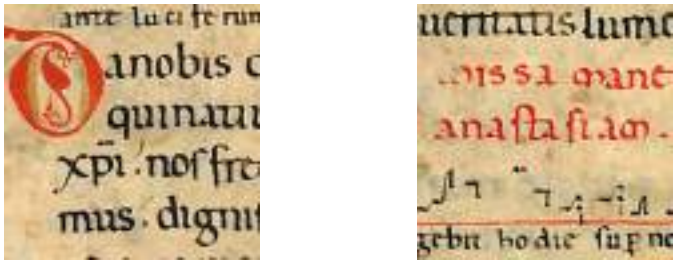
(b) original verso

Figure 4.35: Ideal images

(a) degraded recto

(b) degraded verso

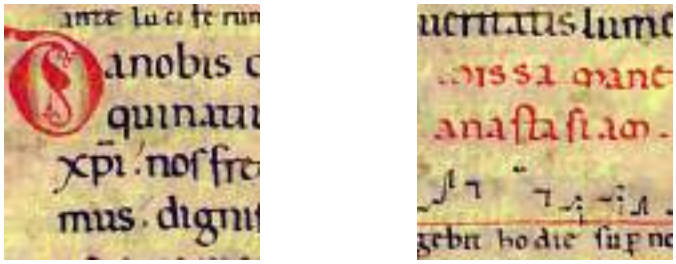
Figure 4.36: Degraded images



(a) recto estimated by ZEODS

(b) verso estimated by ZEODS

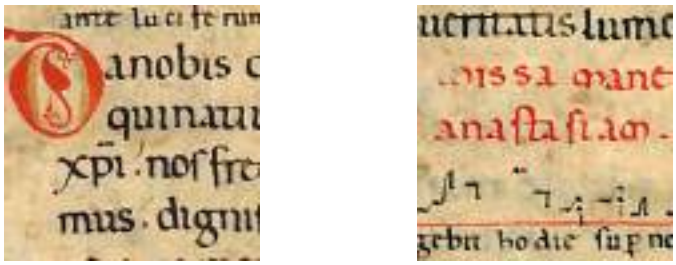
Figure 4.37: Estimates by ZEODS



(a) recto estimated by MATODS

(b) verso estimated by MATODS

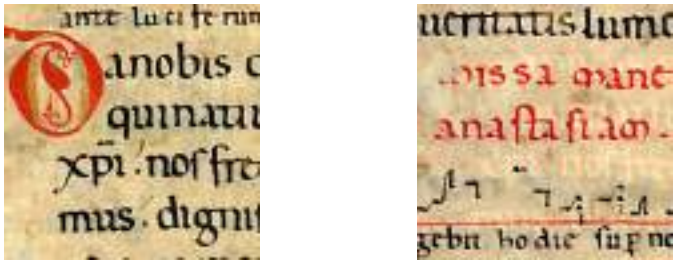
Figure 4.38: Estimates by MATODS



(a) recto estimated by FastIca

(b) verso estimated by FastIca

Figure 4.39: Estimates by FastIca



(a) recto estimated by Symmetric Whitening

(b) verso estimated by Symmetric Whitening

Figure 4.40: Estimates by Symmetric Whitening

(a) recto estimated by Whitening

(b) verso estimated by Whitening

Figure 4.41: Estimates by Whitening

(a) recto estimated by PCA

(b) verso estimated by PCA

Figure 4.42: Estimates by PCA

Used Technique	MSE Recto	MSE Verso	MSE of A
ZEODS	4.7486	1.6165	$1.4055 \cdot 10^{-4}$
MATODS	136.7090	120.7570	0.0015
FASTICA	58.2382	212.8663	0.0546
Symmetric Whitening	428.0422	373.6753	0.0183
Whitening	$7.7086 \cdot 10^3$	$6.2362 \cdot 10^3$	0.3561
PCA	$1.4943 \cdot 10^4$	$5.2861 \cdot 10^3$	0.3770

Table 4.5: Errors of the algorithms by using the mixture matrix in (4.24).

Whitening. However the MATODS algorithm obtains results close to those of the ZEODS algorithm only in the image in Figure 4.27. To see this, we compare the execution time of the two algorithms in the image in Figure 4.27. The results are presented in Table 4.14.

Used Technique	Time
ZEODS	0.3320*s*
MATODS	754.1420*s*

Table 4.6: Execution time of the algorithms MATODS and ZEODS by using the mixture matrix in (4.24) on the image in Figure 4.27

To see a further demonstration of what we said before, we now make a further test on another image, obtaining similar results by means of both algorithms obtaining similar results by means of both algorithms ZEODS and MATODS.

We consider the ideal images in Figure 4.43.

(a) original recto (b) original verso

Figure 4.43: Ideal images

Using the above indicated mixture matrices, we synthetically obtain the degraded images in Figure 4.44.

In Table 4.7 we present the mean square errors with respect to the original documents obtained by means of the above algorithms for the estimate of the recto and the verso of Figure 4.43.

The algorithms MATODS and ZEODS obtain very similar results. By applying the algorithms we obtain, as estimates, the results in Figures 4.45-4.46. Now we analyze the execution time of the algorithms. As in the previous case, we see that the ZEODS method gives results in a much shorter time than the MATODS method, as shown in Table 4.14.

(a) degraded recto

(b) degraded verso

Figure 4.44: Degraded images

Used Technique	MSE Recto	MSE Verso	MSE of A
ZEODS	0.0008	0.4494	$1.6842 \cdot 10^{-6}$
MATODS	0.0081	0.0019	$1.29 \cdot 10^{-4}$
FASTICA	42.7700	70.7900	0.0066
Symmetric Whitening	341.69	342.1863	0.0048
Whitening	245.8900	262.93	0.0086
PCA	$9249 \cdot 10^4$	$10330 \cdot 10^3$	0.038

Table 4.7: Errors of the algorithms by using the mixture matrix in (4.24).

(a) recto estimated by ZEODS

(b) verso estimated by ZEODS

Figure 4.45: Estimates by ZEODS

Used Technique	Time
ZEODS	0.3410s
MATODS	750.6980s

Table 4.8: Execution time of the algorithms MATODS and ZEODS by using the mixture matrix in (4.24) on the image in Figure 4.43

(a) recto estimated by MATODS

(b) verso estimated by MATODS

Figure 4.46: Estimates by MATODS

These results given in terms of time are consistent with the previously obtained results.

4.3.2 Case 2: Second symmetric matrix

The second case we investigate is another symmetric mixture matrix. For every channel R, G and B, the corresponding matrices are

$$A_R = \begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix}, A_G = \begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix}, A_B = \begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix}. \tag{4.25}$$

Now we see the behavior of the presented algorithms, in connection both with errors and with the visual point of view.

We consider the ideal images in Figure 4.47.

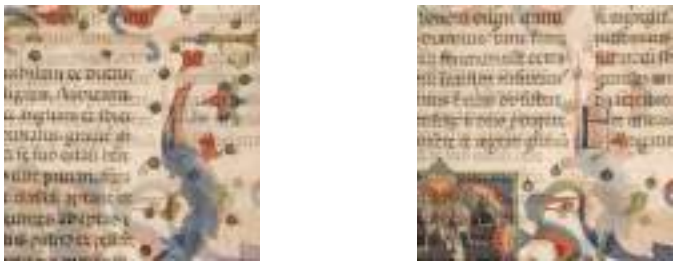
(a) original recto

(b) original verso

Figure 4.47: Ideal images

Using the above indicated mixture matrices, we synthetically obtain the degraded images in Figure 4.48.

By applying the algorithms we obtain, as estimates, the results in Figures 4.49-4.54. In



(a) degraded recto

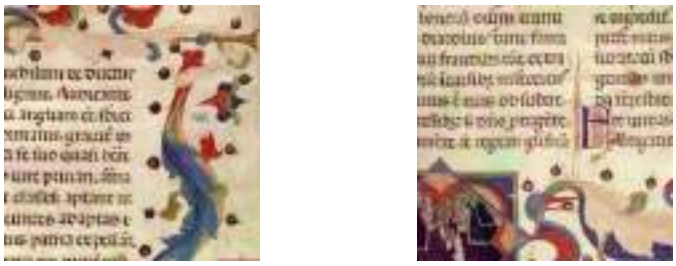
(b) degraded verso

Figure 4.48: Degraded images

(a) recto estimated by ZEODS

(b) verso estimated by ZEODS

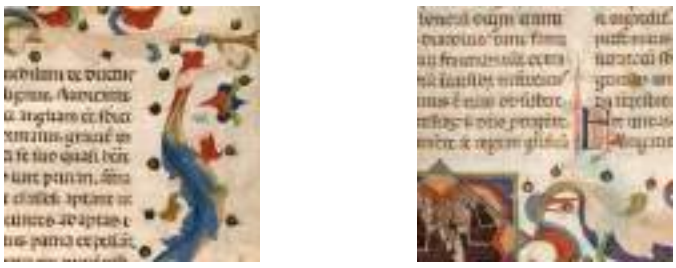
Figure 4.49: Estimates by ZEODS



(a) recto estimated by MATODS

(b) verso estimated by MATODS

Figure 4.50: Estimates by MATODS



(a) recto estimated by FastIca

(b) verso estimated by FastIca

Figure 4.51: Estimates by FastIca

(a) recto estimated by Symmetric Whitening

(b) verso estimated by Symmetric Whitening

Figure 4.52: Estimates by Symmetric Whitening

(a) recto estimated by Whitening

(b) verso estimated by Whitening

Figure 4.53: Estimates by Whitening

(a) recto estimated by PCA

(b) verso estimated by PCA

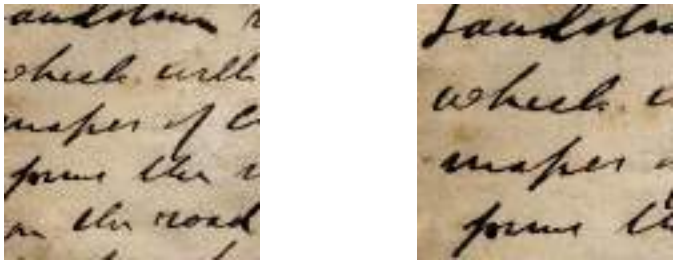
Figure 4.54: Estimates by PCA

Table 4.9 we present the mean square errors with respect to the original documents obtained by means of the above algorithms for the estimate of the recto and the verso of Figure 4.47.

Used Technique	MSE Recto	MSE Verso	MSE of A
ZEODS	0.8547	4.9596	$5.8836 \cdot 10^{-5}$
MATODS	17.6269	50.6982	0.0004
FASTICA	37.5413	86.2744	0.0783
Symmetric Whitening	519.4615	288.9082	0.0352
Whitening	$2.4090 \cdot 10^3$	400.2690	0.0352
PCA	$7.7310 \cdot 10^3$	$3.7087 \cdot 10^3$	0.3674

Table 4.9: Errors of the algorithms by using the mixture matrix in (4.25).

We consider the ideal images in Figure 4.55. Using the above mixture matrices, we synthetically obtain the degraded images in Figure 4.56.



(a) original recto (b) original verso

Figure 4.55: Ideal images

By applying the algorithms we obtain, as estimates, the results in Figures 4.57-4.62.

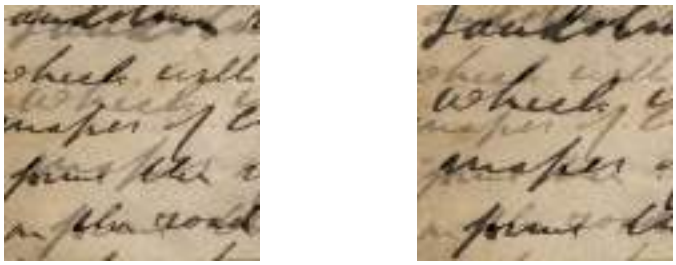
In Table 4.10 we present the mean square errors with respect to the original documents obtained by means of the above algorithms for the estimate of the recto and the verso of Figure 4.55.

We consider the ideal images in Figure 4.63.

Using the above indicated mixture matrices, we synthetically obtain the degraded images in Figure 4.64.

By applying the algorithms we obtain, as estimates, the results in Figures 4.65-4.70.

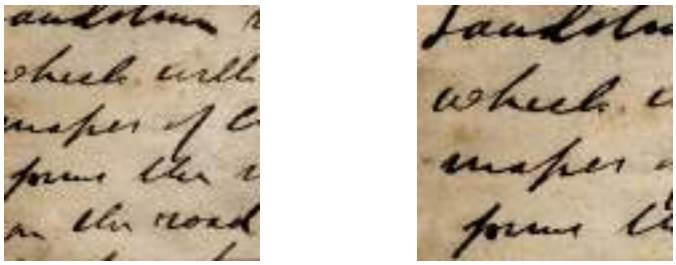
In Table 4.11 we present the mean square errors with respect to the original documents ob-



(a) degraded recto

(b) degraded verso

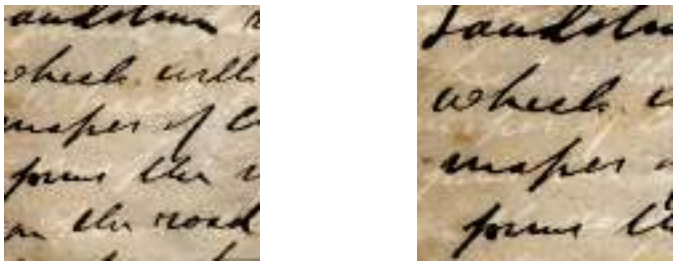
Figure 4.56: Degraded images



(a) recto estimated by ZEODS

(b) verso estimated by ZEODS

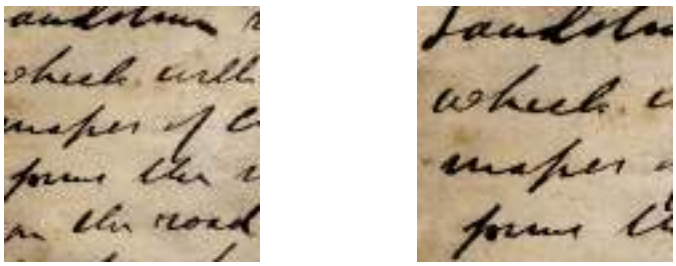
Figure 4.57: Estimates by ZEODS



(a) recto estimated by MATODS

(b) verso estimated by MATODS

Figure 4.58: Estimates by MATODS

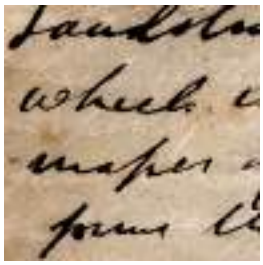


(a) recto estimated by FastIca

(b) verso estimated by FastIca

Figure 4.59: Estimates by FastIca

(a) recto estimated by Symmetric Whitening



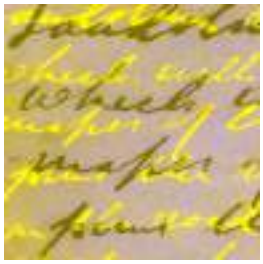
(b) verso estimated by Symmetric Whitening

Figure 4.60: Estimates by Symmetric Whitening

(a) recto estimated by Whitening

(b) verso estimated by Whitening

Figure 4.61: Estimates by Whitening



(a) recto estimated by PCA

(b) verso estimated by PCA

Figure 4.62: Estimates by PCA

(a) original recto

(b) original verso

Figure 4.63: Ideal images

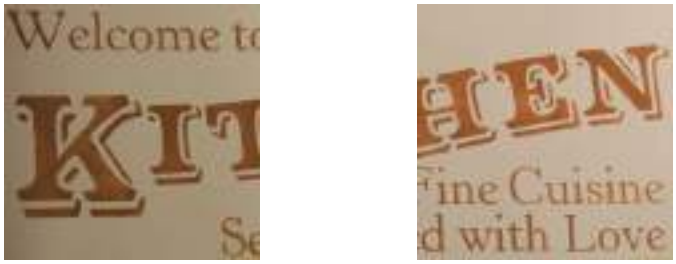
Used Technique	MSE Recto	MSE Verso	MSE of A
ZEODS	0.0648	2.9075	$1.3883 \cdot 10^{-5}$
MATODS	125.8860	213.165	0.0035
FASTICA	14.2089	1.7185	0.0215
Symmetric Whitening	71.8710	75.6985	0.224
Whitening	$1.1589 \cdot 10^4$	$6.9410 \cdot 10^3$	0.4281
PCA	$1.5428 \cdot 10^4$	$5.3671 \cdot 10^3$	0.4305

Table 4.10: Errors of the algorithms by using the mixture matrix in (4.25).

(a) degraded recto

(b) degraded verso

Figure 4.64: Degraded images



(a) recto estimated by ZEODS

(b) verso estimated by ZEODS

Figure 4.65: Estimates by ZEODS

(a) recto estimated by MATODS

(b) verso estimated by MATODS

Figure 4.66: Estimates by MATODS

(a) recto estimated by FastIca

(b) verso estimated by FastIca

Figure 4.67: Estimates by FastIca

(a) recto estimated by Symmetric Whitening

(b) verso estimated by Symmetric Whitening

Figure 4.68: Estimates by Symmetric Whitening

(a) recto estimated by Whitening

(b) verso estimated by Whitening

Figure 4.69: Estimates by Whitening

(a) recto estimated by PCA

(b) verso estimated by PCA

Figure 4.70: Estimates by PCA

tained by means of the above algorithms for the estimate of the recto and the verso of Figure 4.63. We consider the ideal images in Figure 4.71.

Used Technique	MSE Recto	MSE Verso	MSE of A
ZEODS	0.7132	1.8467	$2.1718 \cdot 10^{-5}$
MATODS	12.4361	48.8206	0.021
FASTICA	12.2312	42.1443	0.0407
Symmetric Whitening	190.6356	174.7290	0.0326
Whitening	$3.9342 \cdot 10^3$	$1.5761 \cdot 10^3$	0.4392
PCA	$5.7594 \cdot 10^3$	$1.5845 \cdot 10^3$	0.4368

Table 4.11: Errors of the algorithms by using the mixture matrix in (4.25).

Using the above indicated mixture matrices, we synthetically obtain the degraded images in Figure 4.72.

By applying the algorithms we obtain, as estimates, the results in Figures 4.73-4.78.

In Table 4.12 we present the mean square errors with respect to the original documents obtained by means of the above algorithms for the estimate of the recto and the verso of Figure 4.71.

We consider the ideal images in Figure 4.79.

Using the above indicated mixture matrices, we synthetically obtain the degraded images in Figure 4.80.

By applying the algorithms we obtain, as estimates, the results in Figures 4.81-4.86.

In Table 4.13 we present the mean square errors with respect to the original documents ob-

(a) original recto

(b) original verso

Figure 4.71: Ideal images

(a) degraded recto

(b) degraded verso

Figure 4.72: Degraded images

(a) recto estimated by ZEODS

(b) verso estimated by ZEODS

Figure 4.73: Estimates by ZEODS

(a) recto estimated by MATODS

(b) verso estimated by MATODS

Figure 4.74: Estimates by MATODS

(a) recto estimated by FastIca

(b) verso estimated by FastIca

Figure 4.75: Estimates by FastIca

(a) recto estimated by Symmetric Whitening

(b) verso estimated by Symmetric Whitening

Figure 4.76: Estimates by Symmetric Whitening

(a) recto estimated by Whitening

(b) verso estimated by Whitening

Figure 4.77: Estimates by Whitening

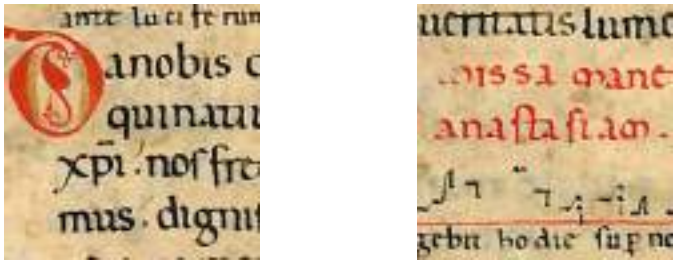
(a) recto estimated by PCA

(b) verso estimated by PCA

Figure 4.78: Estimates by PCA

Used Technique	MSE Recto	MSE Verso	MSE of A
ZEODS	0.2304	1.9043	$5.5429 \cdot 10^{-5}$
MATODS	1.4521	3.5621	0.0010
FASTICA	0.8686	0.4879	0.0120
Symmetric Whitening	5.1557	10.1508	0.0159
Whitening	$2.8938 \cdot 10^{3}$	$1.5686 \cdot 10^{3}$	0.5148
PCA	$3.5387 \cdot 10^{3}$	$1.0885 \cdot 10^{3}$	0.4658

Table 4.12: Errors of the algorithms by using the mixture matrix in (4.25).



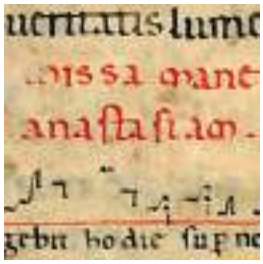
(a) original recto (b) original verso

Figure 4.79: Ideal images

(a) degraded recto (b) degraded verso

Figure 4.80: Degraded images

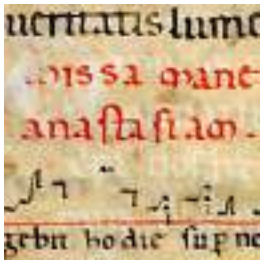
(a) recto estimated by ZEODS



(b) verso estimated by ZEODS

Figure 4.81: Estimates by ZEODS

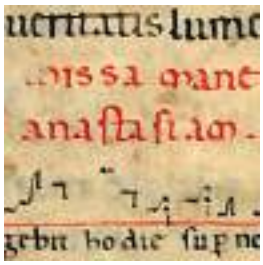
(a) recto estimated by MATODS



(b) verso estimated by MATODS

Figure 4.82: Estimates by MATODS

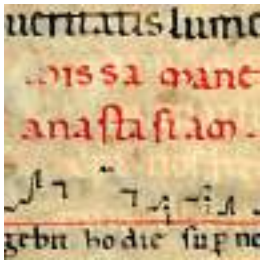
(a) recto estimated by FastIca



(b) verso estimated by FastIca

Figure 4.83: Estimates by FastIca

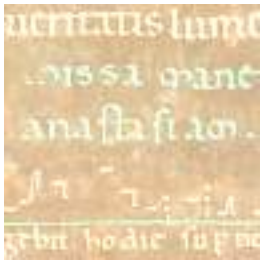
(a) recto estimated by Symmetric Whitening



(b) verso estimated by Symmetric Whitening

Figure 4.84: Estimates by Symmetric Whitening

(a) recto estimated by Whitening



(b) verso estimated by Whitening

Figure 4.85: Estimates by Whitening

(a) recto estimated by PCA

(b) verso estimated by PCA

Figure 4.86: Estimates by PCA

tained by means of the above algorithms for the estimate of the recto and the verso of Figure 4.79 and the corresponding distance between the ideal and the estimated sources.

As we can note in the results of the previous subsection, the ZEODS methods, in terms of errors, always obtains better results than the FastIca, PCA, Whitening and Symmetric Whitening algorithms. However the MATODS algorithm obtains results close to those of the proposed algorithm only in the image in Figure 4.71. But the execution time of the ZEODS algorithm is much shorter than those of the MATODS algorithm. To see this, we compare the execution time of the two algorithms in the image in Figure 4.71.

To see a further demonstration of what we said before, we now make a further test on another image, obtaining similar results by means of both algorithms ZEODS e MATODS.

We consider the ideal images in Figure 4.87.

Using the above indicated mixture matrices, we synthetically obtain the degraded images in Figure 4.88.

In Table 4.15 we present the mean square errors with respect to the original documents ob-

Used Technique	MSE Recto	MSE Verso	MSE of A
ZEODS	1.6564	4.5617	$9.4655 \cdot 10^{-5}$
MATODS	110.2154	85.9412	0.0015
FASTICA	19.2557	7.4678	0.0266
Symmetric Whitening	31.9505	84.1863	0.0220
Whitening	$1.8337 \cdot 10^4$	$8.4063 \cdot 10^3$	0.5216
PCA	$2.2485 \cdot 10^4$	$5.9284 \cdot 10^3$	0.4693

Table 4.13: Errors of the algorithms by using the mixture matrix in (4.25).

Used Technique	Time
ZEODS	0.3150*s*
MATODS	687.3250*s*

Table 4.14: Execution time of the algorithms MATODS and ZEODS by using the mixture matrix in (4.25) on the image in Figure 4.71

(a) original recto (b) original verso

Figure 4.87: Ideal images

(a) degraded recto (b) degraded verso

Figure 4.88: Degraded images

tained by means of the above algorithms for the estimate of the recto and the verso of Figure 4.87. The algorithms MATODS and ZEODS obtain very similar results. We obtain, as estimates,

Used Technique	MSE Recto	MSE Verso	MSE of A
ZEODS	5.2751	4.1563	$4.1236 \cdot 10^{-5}$
MATODS	0.1501	0.1910	$1.4301 \cdot 10^{-5}$
FASTICA	42.7700	70.7900	0.0066
Symmetric Whitening	341.69	342.1863	0.0048
Whitening	245.8900	262.93	0.0086
PCA	9249	10330	0.038

Table 4.15: Errors of the algorithms by using the mixture matrix in (4.25).

the results in Figures 4.89-4.90. However, if we analyze the excution time of the algorithm, we

(a) recto estimated by ZEODS

(b) verso estimated by ZEODS

Figure 4.89: Estimates by ZEODS

see that the ZEODS method gives results in a much shorter time than the MATODS method, as shown in Table 4.16.

Used Technique	Time
ZEODS	0.3330*s*
MATODS	489.0880*s*

Table 4.16: Execution time of the algorithms MATODS and ZEODS by using the mixture matrix in (4.25).

(a) recto estimated by MATODS

(b) verso estimated by MATODS

Figure 4.90: Estimates by MATODS

4.3.3 Case 3: First asymmetric matrix

The third case we deal with is an asymmetric mixture matrix. For every channel R, G and B, the related matrices are

$$A_R = \begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix}, A_G = \begin{pmatrix} 0.7 & 0.3 \\ 0.2 & 0.8 \end{pmatrix}, A_B = \begin{pmatrix} 0.6 & 0.4 \\ 0.3 & 0.7 \end{pmatrix}. \tag{4.26}$$

Now we see the behavior of the presented algorithms, concerning both errors and the visual point of view.

We consider the ideal images in Figure 4.91.

(a) original recto

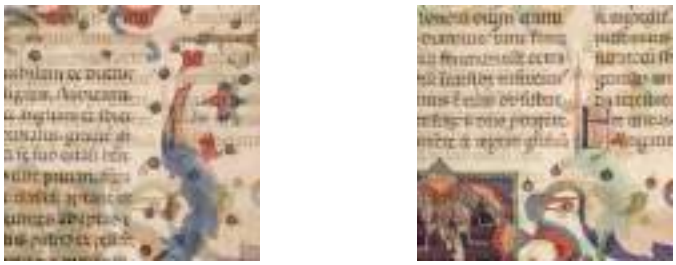
(b) original verso

Figure 4.91: Ideal images

Using the above indicated mixture matrices, we synthetically obtain the degraded images in Figure 4.92.

By applying the algorithms we obtain, as estimates, the results in Figures 4.93-4.98.

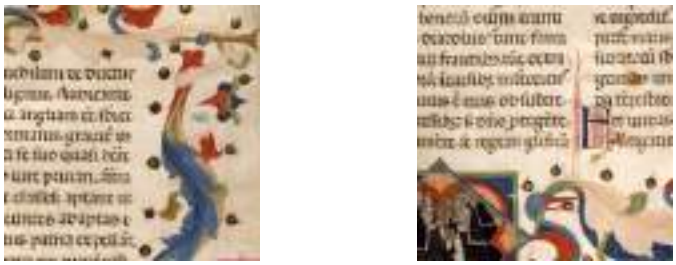
In Table 4.17 we present the mean square errors with respect to the original documents obtained by means of the above algorithms for the estimate of the recto and the verso of Figure 4.91. We consider the ideal images in Figure 4.99. Using the above indicated mixture matrices,



(a) degraded recto

(b) degraded verso

Figure 4.92: Degraded images



(a) recto estimated by ZEODS

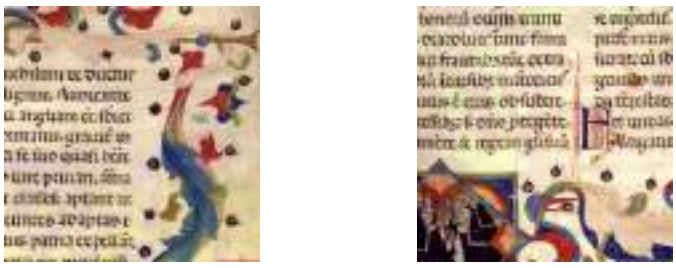
(b) verso estimated by ZEODS

Figure 4.93: Estimates by ZEODS

(a) recto estimated by MATODS

(b) verso estimated by MATODS

Figure 4.94: Estimates by MATODS



(a) recto estimated by FastIca

(b) verso estimated by FastIca

Figure 4.95: Estimates by FastIca

(a) recto estimated by Symmetric Whitening

(b) verso estimated by Symmetric Whitening

Figure 4.96: Estimates by Symmetric Whitening

(a) recto estimated by Whitening

(b) verso estimated by Whitening

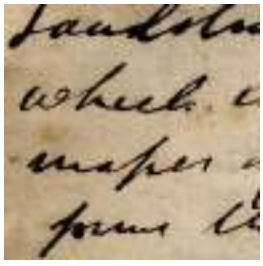
Figure 4.97: Estimates by Whitening

(a) recto estimated by PCA

(b) verso estimated by PCA

Figure 4.98: Estimates by PCA

(a) original recto



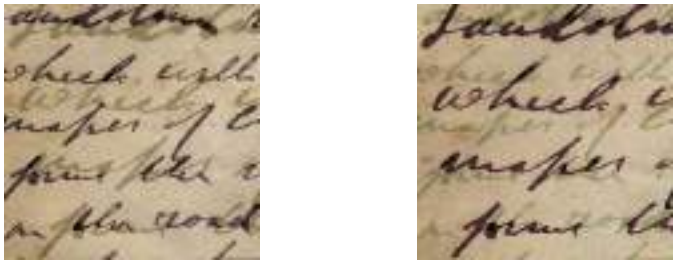
(b) original verso

Figure 4.99: Ideal images

Used Technique	MSE Recto	MSE Verso	MSE of A
ZEODS	0.9539	3.8356	$6.1210 \cdot 10^{-5}$
MATODS	45.2314	49.0506	0.0011
FASTICA	29.2027	148.9813	0.0701
Symmetric Whitening	451.6652	419.6792	0.0373
Whitening	$2.8741 \cdot 10^3$	352.5680	0.1792
PCA	$8.0327 \cdot 10^3$	$3.5478 \cdot 10^3$	0.3596

Table 4.17: Errors of the algorithms by using the mixture matrix in (4.26).

we synthetically obtain the degraded images in Figure 4.100.



(a) degraded recto (b) degraded verso

Figure 4.100: Degraded images

By applying the algorithms we obtain, as estimates, the results in Figures 4.101-4.106.

In Table 4.18 we present the mean square errors with respect to the original documents obtained by means of the above algorithms for the estimate of the recto and the verso of Figure 4.99.

We consider the ideal images in Figure 4.107.

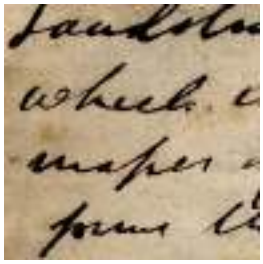
Using the above indicated mixture matrices, we synthetically obtain the degraded images in Figure 4.108.

By applying the algorithms we obtain, as estimates, the results in Figures 4.109-4.114.

In Table 4.19 we present the mean square errors with respect to the original documents obtained by means of the above algorithms for the estimate of the recto and the verso of Figure 4.107. We consider the ideal images in Figure 4.115.

Using the above indicated mixture matrices, we synthetically obtain the degraded images in

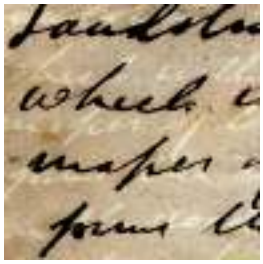
(a) recto estimated by ZEODS



(b) verso estimated by ZEODS

Figure 4.101: Estimates by ZEODS

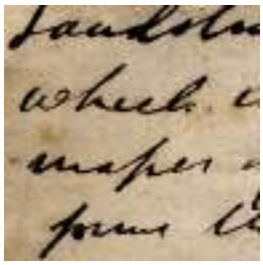
(a) recto estimated by MATODS



(b) verso estimated by MATODS

Figure 4.102: Estimates by MATODS

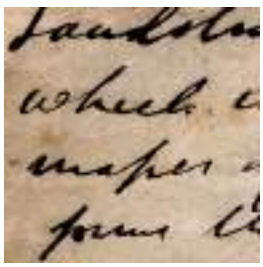
(a) recto estimated by FastIca



(b) verso estimated by FastIca

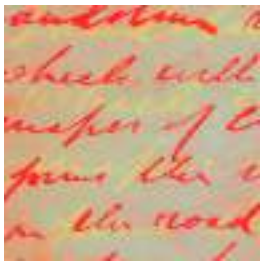
Figure 4.103: Estimates by FastIca

(a) recto estimated by Symmetric Whitening

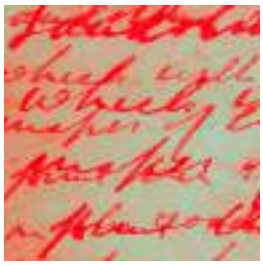


(b) verso estimated by Symmetric Whitening

Figure 4.104: Estimates by Symmetric Whitening

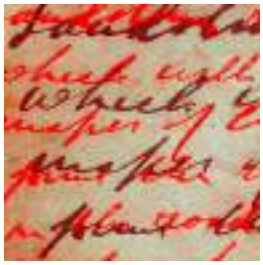


(a) recto estimated by Whitening



(b) verso estimated by Whitening

Figure 4.105: Estimates by Whitening



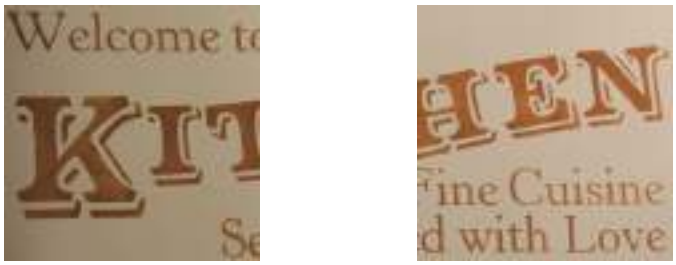
(a) recto estimated by PCA

(b) verso estimated by PCA

Figure 4.106: Estimates by PCA

Used Technique	MSE Recto	MSE Verso	MSE of A
ZEODS	0.2423	3.5365	$2.044 \cdot 10^{-5}$
MATODS	35.0330	51.3125	0.0002
FASTICA	4.4079	4.1418	0.0126
Symmetric Whitening	45.8355	117.4545	0.0305
Whitening	$6.7961 \cdot 10^3$	$3.7444 \cdot 10^3$	0.3297
PCA	$1.1179 \cdot 10^4$	$4.1416 \cdot 10^3$	0.3893

Table 4.18: Errors of the algorithms by using the mixture matrix in (4.26).

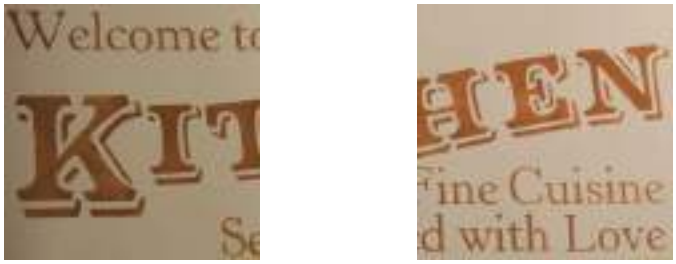


(a) original recto (b) original verso

Figure 4.107: Ideal images

(a) degraded recto (b) degraded verso

Figure 4.108: Degraded images

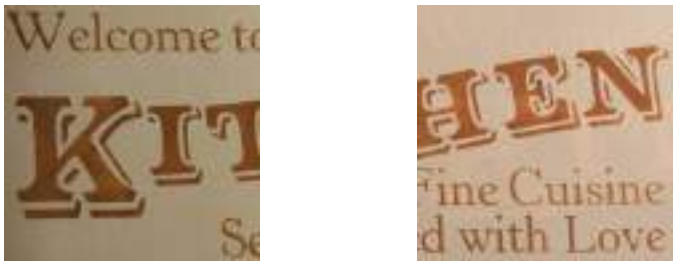


(a) recto estimated by ZEODS (b) verso estimated by ZEODS

Figure 4.109: Estimates by ZEODS

(a) recto estimated by MATODS (b) verso estimated by MATODS

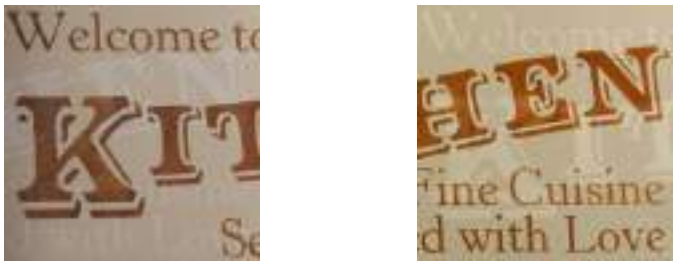
Figure 4.110: Estimates by MATODS



(a) recto estimated by FastIca

(b) verso estimated by FastIca

Figure 4.111: Estimates by FastIca



(a) recto estimated by Symmetric Whitening

(b) verso estimated by Symmetric Whitening

Figure 4.112: Estimates by Symmetric Whitening

(a) recto estimated by Whitening

(b) verso estimated by Whitening

Figure 4.113: Estimates by Whitening

(a) recto estimated by PCA

(b) verso estimated by PCA

Figure 4.114: Estimates by PCA

Used Technique	MSE Recto	MSE Verso	MSE of A
ZEODS	0.4521	1.3566	$1.9913 \cdot 10^{-5}$
MATODS	67.4521	75.6765	0.0007
FASTICA	14.8255	47.8983	0.0429
Symmetric Whitening	221.8945	190.5466	0.0377
Whitening	$1.6127 \cdot 10^3$	421.6936	0.0377
PCA	$3.7456 \cdot 10^3$	$1.3281 \cdot 10^3$	0.3954

Table 4.19: Errors of the algorithms by using the mixture matrix in (4.26).

(a) original recto (b) original verso

Figure 4.115: Ideal images

Figure 4.116.

(a) degraded recto (b) degraded verso

Figure 4.116: Degraded images

By applying the algorithms we obtain, as estimates, the results in Figures 4.117-4.122.

(a) recto estimated by ZEODS (b) verso estimated by ZEODS

Figure 4.117: Estimates by ZEODS

In Table 4.20 we present the mean square errors with respect to the original documents obtained by means of the above algorithms for the estimate of the recto and the verso of Figure 4.115.

We consider the ideal images in Figure 4.123.

Using the above indicated mixture matrices, we synthetically obtain the degraded images in Figure 4.124.

By applying the algorithms we obtain, as estimates, the results in Figures 4.125-4.130.

In Table 4.21 we present the mean square errors with respect to the original documents obtained by means of the above algorithms for the estimate of the recto and the verso of Figure 4.123.

As we observe in the previous results, the ZEODS methods, in terms of errors, always obtains better results than the FastIca, PCA, Whitening and Symmetric Whitening algorithms. However,

(a) recto estimated by MATODS

(b) verso estimated by MATODS

Figure 4.118: Estimates by MATODS

(a) recto estimated by FastIca

(b) verso estimated by FastIca

Figure 4.119: Estimates by FastIca

(a) recto estimated by Symmetric Whitening

(b) verso estimated by Symmetric Whitening

Figure 4.120: Estimates by Symmetric Whitening

(a) recto estimated by Whitening

(b) verso estimated by Whitening

Figure 4.121: Estimates by Whitening

(a) recto estimated by PCA

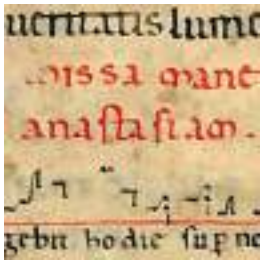
(b) verso estimated by PCA

Figure 4.122: Estimates by PCA

Used Technique	MSE Recto	MSE Verso	MSE of A
ZEODS	0.1486	0.1950	$6.2159 \cdot 10^{-5}$
MATODS	1.9025	2.3132	$2.1564 \cdot 10^{-5}$
FASTICA	0.9037	0.5265	0.0117
Symmetric Whitening	3.8798	12.8583	0.0270
Whitening	$1.7404 \cdot 10^{3}$	833.1407	0.3356
PCA	$2.5707 \cdot 10^{3}$	795.5274	0.3916

Table 4.20: Errors of the algorithms by using the mixture matrix in (4.26).

(a) original recto



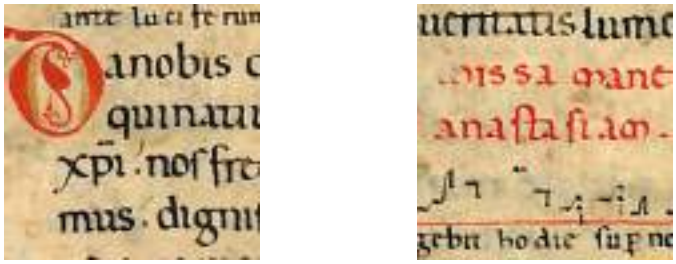
(b) original verso

Figure 4.123: Ideal images

(a) degraded recto

(b) degraded verso

Figure 4.124: Degraded images



(a) recto estimated by ZEODS

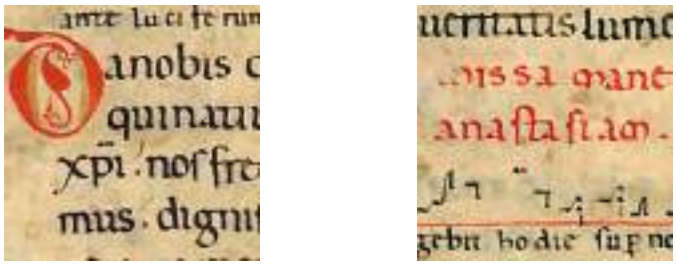
(b) verso estimated by ZEODS

Figure 4.125: Estimates by ZEODS

(a) recto estimated by MATODS

(b) verso estimated by MATODS

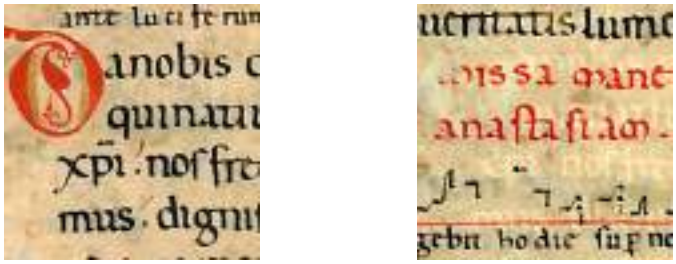
Figure 4.126: Estimates by MATODS



(a) recto estimated by FastIca

(b) verso estimated by FastIca

Figure 4.127: Estimates by FastIca

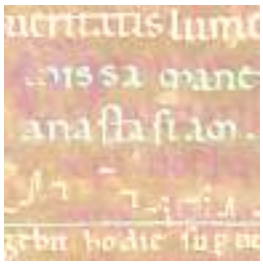


(a) recto estimated by Symmetric Whitening

(b) verso estimated by Symmetric Whitening

Figure 4.128: Estimates by Symmetric Whitening

(a) recto estimated by Whitening



(b) verso estimated by Whitening

Figure 4.129: Estimates by Whitening

(a) recto estimated by PCA

(b) verso estimated by PCA

Figure 4.130: Estimates by PCA

Used Technique	MSE Recto	MSE Verso	MSE of A
ZEODS	1.8024	5.2181	$1.012 \cdot 10^{-4}$
MATODS	20.7090	19.3665	0.0001
FASTICA	15.7847	3.3160	0.0223
Symmetric Whitening	7.2817	109.0196	0.0339
Whitening	$1.7703 \cdot 10^{4}$	$8.5767 \cdot 10^{3}$	0.0339
PCA	$2.17489 \cdot 10^{4}$	$5.9721 \cdot 10^{3}$	0.4655

Table 4.21: Errors of the algorithms by using the mixture matrix in (4.26).

the MATODS algorithm gives results close to those of the proposed algorithm only in the image in Figure 4.115. To see this, we compare the execution time of the two algorithms in the image in Figure 4.115.

Used Technique	Time
ZEODS	0.3510*s*
MATODS	956.3210*s*

Table 4.22: Execution time of the algorithms MATODS and ZEODS by using the mixture matrix in (4.26) on the image in Figure 4.115

To see a further demonstration of what we said before, we now make a further test on another image, obtaining similar results by means of both algorithms ZEODS e MATODS.

We consider the ideal images in Figure 4.131.

(a) original recto (b) original verso

Figure 4.131: Ideal images

Using the above indicated mixture matrices, we synthetically obtain the degraded images in Figure 4.132.

(a) degraded recto (b) degraded verso

Figure 4.132: Degraded images

In Table 4.23 we present the mean square errors with respect to the original documents obtained by means of the above algorithms for the estimate of the recto and the verso of Figure 4.131.

Used Technique	MSE Recto	MSE Verso	MSE of A
ZEODS	11.1003	10.4289	$3.7659 \cdot 10^{-5}$
MATODS	4.0124	3.1247	$2.2459 \cdot 10^{-5}$

Table 4.23: Errors of the algorithms by using the mixture matrix in (4.26).

By applying the algorithms we obtain, as estimates, the results in Figures 4.133-4.134.

(a) recto estimated by ZEODS

(b) verso estimated by ZEODS

Figure 4.133: Estimates by ZEODS

(a) recto estimated by MATODS

(b) verso estimated by MATODS

Figure 4.134: Estimates by MATODS

As we can note in the results of the previous subsection, the ZEODS method, in terms of errors, always obtains better results than the other algorithms, and is even faster than the MATODS method, as shown in Table 4.22.

These results given in terms of time are consistent with the previously obtained results.

Used Technique	Time
ZEODS	0.3440s
MATODS	910.1002s

Table 4.24: Execution time of the algorithms MATODS and ZEODS by using the mixture matrix in (4.26) on the image in Figure 4.131

4.3.4 Case 4: Second asymmetric matrix

In the fourth and last case we consider another asymmetric mixture matrix. For every channel R, G and B, the corresponding matrices are

$$A_R = \begin{pmatrix} 0.7 & 0.3 \\ 0.2 & 0.8 \end{pmatrix}, A_G = \begin{pmatrix} 0.45 & 0.55 \\ 0.4 & 0.6 \end{pmatrix}, A_B = \begin{pmatrix} 0.7 & 0.3 \\ 0.51 & 0.49 \end{pmatrix}. \tag{4.27}$$

Now we see the behavior of the presented algorithms, regarding both errors and the visual point of view. We consider the ideal images in Figure 4.135.

(a) original recto (b) original verso

Figure 4.135: Ideal images

Using the above indicated mixture matrices, we synthetically obtain the degraded images in Figure 4.136.

By applying the algorithms we obtain, as estimates, the results in Figures 4.137-4.142.

In Table 4.25 we present the mean square errors with respect to the original documents obtained by means of the above algorithms for the estimate of the recto and the verso of Figure 4.135. We consider the ideal images in Figure 4.143. Using the above indicated mixture matrices, we synthetically obtain the degraded images in Figure 4.144.

By applying the algorithms we obtain, as estimates, the results in Figures 4.145-4.150.

(a) degraded recto

(b) degraded verso

Figure 4.136: Degraded images

(a) recto estimated by ZEODS

(b) verso estimated by ZEODS

Figure 4.137: Estimates by ZEODS

(a) recto estimated by MATODS

(b) verso estimated by MATODS

Figure 4.138: Estimates by MATODS

(a) recto estimated by FastIca

(b) verso estimated by FastIca

Figure 4.139: Estimates by FastIca

(a) recto estimated by Symmetric Whitening

(b) verso estimated by Symmetric Whitening

Figure 4.140: Estimates by Symmetric Whitening

(a) recto estimated by Whitening

(b) verso estimated by Whitening

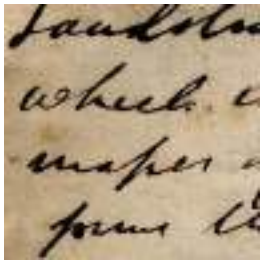
Figure 4.141: Estimates by Whitening

(a) recto estimated by PCA

(b) verso estimated by PCA

Figure 4.142: Estimates by PCA

(a) original recto



(b) original verso

Figure 4.143: Ideal images

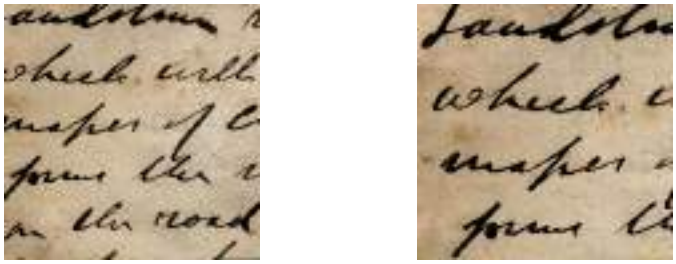
Used Technique	MSE Recto	MSE Verso	MSE of A
ZEODS	3.9507	4.9612	$7.6397 \cdot 10^{-5}$
MATODS	50.1485	41.1745	0.0098
FASTICA	615.3561	346.1334	0.0719
Symmetric Whitening	707.1949	631.6572	0.0520
Whitening	$2.3355 \cdot 10^3$	938.1797	0.2227
PCA	$6.5589 \cdot 10^3$	$4.1706 \cdot 10^3$	0.3401

Table 4.25: Errors of the algorithms by using the mixture matrix in (4.27).

(a) degraded recto

(b) degraded verso

Figure 4.144: Degraded images

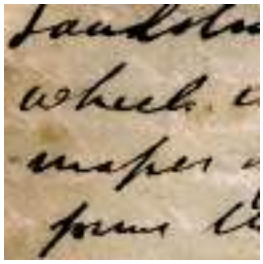


(a) recto estimated by ZEODS

(b) verso estimated by ZEODS

Figure 4.145: Estimates by ZEODS

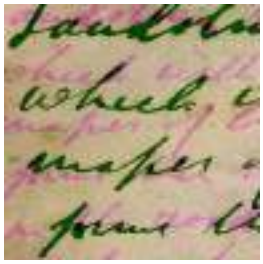
(a) recto estimated by MATODS



(b) verso estimated by MATODS

Figure 4.146: Estimates by MATODS

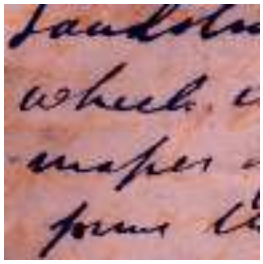
(a) recto estimated by FastIca



(b) verso estimated by FastIca

Figure 4.147: Estimates by FastIca

(a) recto estimated by Symmetric Whitening



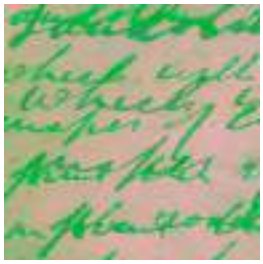
(b) verso estimated by Symmetric Whitening

Figure 4.148: Estimates by Symmetric Whitening

(a) recto estimated by Whitening

(b) verso estimated by Whitening

Figure 4.149: Estimates by Whitening



(a) recto estimated by PCA

(b) verso estimated by PCA

Figure 4.150: Estimates by PCA

In Table 4.26 we present the mean square errors with respect to the original documents obtained by means of the above algorithms for the estimate of the recto and the verso of Figure 4.143.

Used Technique	MSE Recto	MSE Verso	MSE of A
ZEODS	1.2642	2.6337	$2.2806 \cdot 10^{-5}$
MATODS	62.2418	85.4395	0.0026
FASTICA	353.226	182.7357	0.0303
Symmetric Whitening	409.8490	495.5137	0.1435
Whitening	$7.7216 \cdot 10^{3}$	$3.5975 \cdot 10^{3}$	0.4449
PCA	$1.2810 \cdot 10^{4}$	$2.5195 \cdot 10^{3}$	0.4473

Table 4.26: Errors of the algorithms by using the mixture matrix in (4.27).

We consider the ideal images in Figure 4.151.

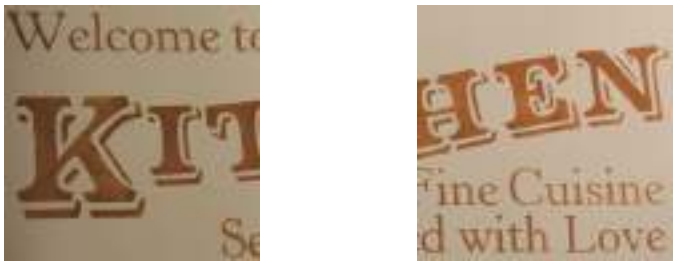
Using the above indicated mixture matrices, we synthetically obtain the images in Figure 4.152. By applying the algorithms we obtain, as estimates, the results in Figures 4.153-4.158.

In Table 4.27 we present the mean square errors with respect to the original documents obtained by means of the above algorithms for the estimate of the recto and the verso of Figure 4.151. We consider the ideal images in Figure 4.159.

Using the above indicated mixture matrices, we synthetically obtain the degraded images in Figure 4.160.

By applying the algorithms we obtain, as estimates, the results in Figures 4.161-4.166.

In Table 4.28 we present the mean square errors with respect to the original documents ob-

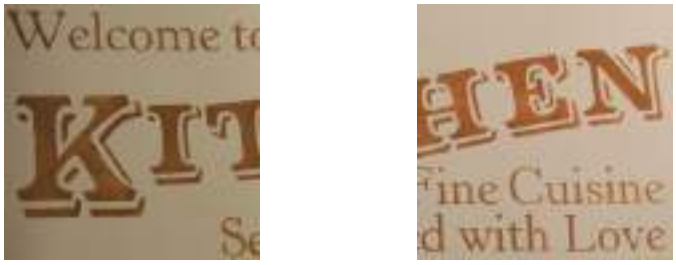


(a) original recto (b) original verso

Figure 4.151: Ideal images

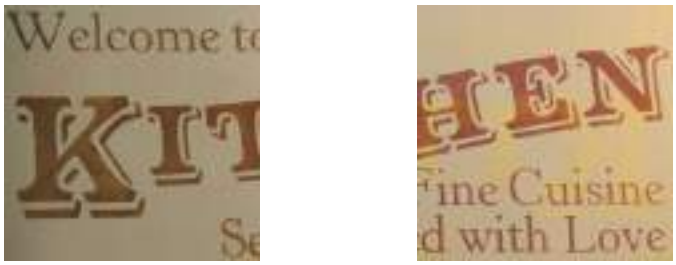
(a) degraded recto (b) degraded verso

Figure 4.152: Degraded images



(a) recto estimated by ZEODS (b) verso estimated by ZEODS

Figure 4.153: Estimates by ZEODS



(a) recto estimated by MATODS (b) verso estimated by MATODS

Figure 4.154: Estimates by MATODS

(a) recto estimated by FastIca

(b) verso estimated by FastIca

Figure 4.155: Estimates by FastIca

(a) recto estimated by Symmetric Whitening

(b) verso estimated by Symmetric Whitening

Figure 4.156: Estimates by Symmetric Whitening

(a) recto estimated by Whitening

(b) verso estimated by Whitening

Figure 4.157: Estimates by Whitening

(a) recto estimated by PCA

(b) verso estimated by PCA

Figure 4.158: Estimates by PCA

Used Technique	MSE Recto	MSE Verso	MSE of A
ZEODS	0.6289	1.3893	$9.6560 \cdot 10^{-6}$
MATODS	12.0247	30.8065	$8.8984 \cdot 10^{-4}$
FASTICA	166.6276	91.2465	0.0386
Symmetric Whitening	352.5150	410.2975	0.0579
Whitening	$1.6118 \cdot 10^{3}$	830.0139	0.3584
PCA	$3.0682 \cdot 10^{3}$	$1.8473 \cdot 10^{3}$	0.3767

Table 4.27: Errors of the algorithms by using the mixture matrix in (4.27).

(a) original recto

(b) original verso

Figure 4.159: Ideal images

(a) degraded recto

(b) degraded verso

Figure 4.160: Degraded images

(a) recto estimated by ZEODS

(b) verso estimated by ZEODS

Figure 4.161: Estimates by ZEODS

(a) recto estimated by MATODS

(b) verso estimated by MATODS

Figure 4.162: Estimates by MATODS

(a) recto estimated by FastIca

(b) verso estimated by FastIca

Figure 4.163: Estimates by FastIca

(a) recto estimated by Symmetric Whitening

(b) verso estimated by Symmetric Whitening

Figure 4.164: Estimates by Symmetric Whitening

(a) recto estimated by Whitening

(b) verso estimated by Whitening

Figure 4.165: Estimates by Whitening

(a) recto estimated by PCA

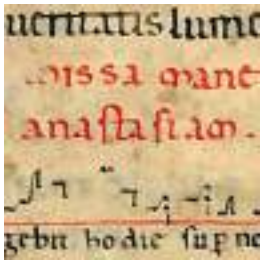
(b) verso estimated by PCA

Figure 4.166: Estimates by PCA

tained by means of the above algorithms for the estimate of the recto and the verso of Figure 4.159.

We consider the following images in Figure 4.167.

(a) original recto



(b) original verso

Figure 4.167: Ideal images

Using the above indicated mixture matrices, we synthetically obtain the degraded images in Figure 4.168.

By applying the algorithms we obtain, as estimates, the results in Figures 4.169-4.174.

In Table 4.29 we present the mean square errors with respect to the original documents ob-

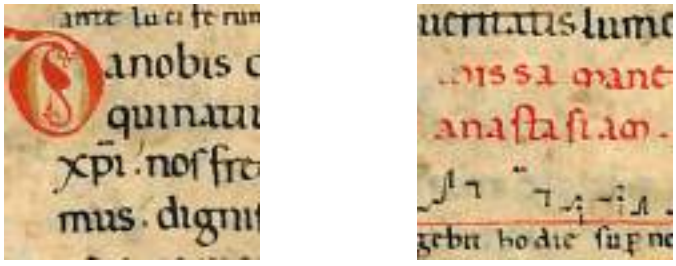
Used Technique	MSE Recto	MSE Verso	MSE of A
ZEODS	0.3876	1.7862	$6.7032 \cdot 10^{-5}$
MATODS	3.1985	5.1475	0.0002
FASTICA	34.7680	15.8122	0.0228
Symmetric Whitening	8.8713	17.2117	0.0458
Whitening	$2.2407 \cdot 10^3$	$1.2194 \cdot 10^3$	0.4580
PCA	$2.8462 \cdot 10^3$	941.9039	0.4180

Table 4.28: Errors of the algorithms by using the mixture matrix in (4.27).

(a) degraded recto

(b) degraded verso

Figure 4.168: Degraded images

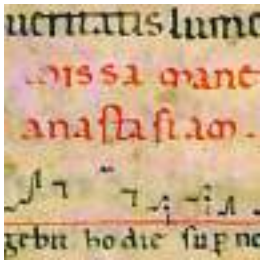


(a) recto estimated by ZEODS

(b) verso estimated by ZEODS

Figure 4.169: Estimates by ZEODS

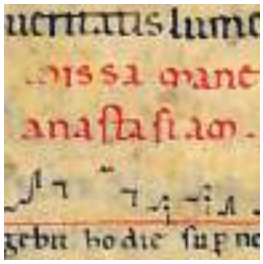
(a) recto estimated by MATODS



(b) verso estimated by MATODS

Figure 4.170: Estimates by MATODS

(a) recto estimated by FastIca



(b) verso estimated by FastIca

Figure 4.171: Estimates by FastIca

(a) recto estimated by Symmetric Whitening

(b) verso estimated by Symmetric Whitening

Figure 4.172: Estimates by Symmetric Whitening

(a) recto estimated by Whitening

(b) verso estimated by Whitening

Figure 4.173: Estimates by Whitening

(a) recto estimated by PCA

(b) verso estimated by PCA

Figure 4.174: Estimates by PCA

tained by means of the above algorithms for the estimate of the recto and the verso of Figure 4.167.

Used Technique	MSE Recto	MSE Verso	MSE of A
ZEODS	3.2977	3.6252	$1.090 \cdot 10^{-4}$
MATODS	35.0124	42.8569	$1.5041 \cdot 10^{-4}$
FASTICA	232.7229	147.4355	0.0304
Symmetric Whitening	235.6894	607.9245	0.1441
Whitening	$1.4669 \cdot 10^{4}$	$6.6340 \cdot 10^{3}$	0.5272
PCA	$1.9414 \cdot 10^{4}$	$3.9348 \cdot 10^{3}$	0.4795

Table 4.29: Errors of the algorithms by using the mixture matrix in (4.27).

As we observe in the results of the previous subsection, the ZEODS methods, in terms of errors, always obtains better results than the FastIca, PCA, Whitening and Symmetric Whitening algorithms. However the MATODS algorithm obtains results close to those of the proposed algorithm only in the image in Figure 4.159. But the execution time of the ZEODS algorithm is much shorter than those of the MATODS algorithm. To see this, we compare the execution time of the two algorithms in the image in Figure 4.159.

To see a further demonstration of what we said before, we now make a further test on another image, obtaining similar results by means of both algorithms ZEODS e MATODS. We consider the ideal images in Figure 4.175.

Using the above indicated mixture matrices, we synthetically obtain the degraded images in Figure 4.176.

Used Technique	Time
ZEODS	0.3390*s*
MATODS	845.1618*s*

Table 4.30: Execution time of the algorithms MATODS and ZEODS by using the mixture matrix in (4.27) on the image in Figure 4.159

(a) original recto (b) original verso

Figure 4.175: Ideal images

(a) degraded recto (b) degraded verso

Figure 4.176: Degraded images

In Table 4.31 we present the mean square errors with respect to the original documents obtained by means of the above algorithms for the estimate of the recto and the verso of Figure 4.175.

Used Technique	MSE Recto	MSE Verso	MSE of A
ZEODS	8.1003	7.4289	$3.7659 \cdot 10^{-5}$
MATODS	6.0247	5.1247	$2.2459 \cdot 10^{-5}$

Table 4.31: Errors of the algorithms by using the mixture matrix in (4.27).

The ZEODS algorithm obtains results very close to the MATODS algorithm. We get, as estimates, the results in Figures 4.177-4.178. We analyze the execution time of algorithms. As

(a) recto estimated by ZEODS

(b) verso estimated by ZEODS

Figure 4.177: Estimates by ZEODS

(a) recto estimated by MATODS

(b) verso estimated by MATODS

Figure 4.178: Estimates by MATODS

in the previous case, we get that the ZEODS method gives results in a much shorter time than the MATODS method, as we can see in Table 4.22.

These results given in terms of time are consistent with the previously obtained results.

Used Technique	Time
ZEODS	0.3510s
MATODS	812.1014s

Table 4.32: Execution time of the algorithms MATODS and ZEODS by using the mixture matrix in (4.26) on the image in Figure 4.175

Chapter 5

Interference level estimation in document restoration

This chapter is structured as follows. In Section 5.2 we deal with the regularization of the modified Sharma model. In Section 5.3 we describe the alternating iterative algorithm, used to find the minimum of the energy function. In Section 5.4 we analyze the technique to determine the interference levels, given the blur operator and the ideal sources. In Section 5.5 we propose different types of convex approximations. In Section 5.6 we present GNC-type alternative minimization techniques. In Section 5.7 we compare the proposed technique by means of the experimental results.

5.1 Regularization of the problem

In this work we consider a modified Sharma-type model related to the show-through phenomenon in paper documents, as follows (see, e.g., [69, 155]):

$$\begin{cases} f^s(i,j) = f(i,j)e^{q_r(i,j)\left(\frac{z_r(i,j)}{N}-1\right)} \\ r^s(i,j) = r(i,j)e^{q_f(i,j)\left(\frac{z_f(i,j)}{N}-1\right)} \end{cases}, \tag{5.1}$$

where N is the maximum value of the light intensity, which is assumed to correspond with the background of the analyzed document; $q_f(i,j)$ is the interference level which affects the light intensity of interferences from the recto to the verso; $q_r(i,j)$ is the interference level which affects

the light intensity of interferences from the verso to the recto; $f^s = [f^s(i,j)]_{i=1,\ldots,n,j=1,\ldots,m}, r^s = [r^s(i,j)]_{i=1,\ldots,n,j=1,\ldots,m} \in \mathbb{R}^{nm}$ are the vectors which represent the observed mixtures (expressed in the lexicographic form); $f = [f(i,j)]_{i=1,\ldots,n,j=1,\ldots,m}, r = [r(i,j)]_{i=1,\ldots,n,j=1,\ldots,m} \in \mathbb{R}^{nm}$ are the vectors which represent the ideal images of the recto and the verso of the document (expressed in the lexicographic form); $z_f = [z_f(i,j)]_{i=1,\ldots,n,j=1,\ldots,m} = Af, z_r = [z_r(i,j)]_{i=1,\ldots,n,j=1,\ldots,m} = Ar$ are the blurred images of the recto and the verso, where $A \in \mathbb{R}^{(nm)\times(nm)}$ is the blur operator, which in general has the form of a matrix with Toeplitz blocks.

The problem of the blind separations of components consists of finding an estimate of the recto/verso pair of the source document, which is denoted by $s = (f,r)$, of the interference level $q = (q_f, q_r)$ and of the blur operator A, given in input the observed images of the recto and the verso. This is an ill-posed problem in the Hadamard sense, because in general it can have no solutions, or the solution can be not unique and/or not stable with respect to small variations of the data.

To estimate the solution of the problem, some regularization techniques are used, which substantially consist of finding the minimum of a function, called *energy function*, by imposing some uniformity constraints on the solution.

The solution of the considered problem is

$$(f^*, r^*, q^*, A^*) = \arg \min_{(f,r,q,A)} E(f,r,q,A),$$

where

$$E(f,r,q,A) = T(f,r,q,A) + \widehat{S}(f) + \widehat{S}(r) + S(q_f) + S(q_r) + S_c(q_f, q_r) \tag{5.2}$$

is the *energy function*, and

$$\begin{aligned} T(f,r,q,A) &= T_f(q_r) + T_r(q_f) = \sum_{i=1}^{n}\sum_{j=1}^{m}\left(f^s(i,j) - f(i,j)\mathrm{e}^{-q_r(i,j)\left(1-\frac{z_r(i,j)}{N}\right)}\right)^2 + \\ &+ \sum_{i=1}^{n}\sum_{j=1}^{m}\left(r^s(i,j) - r(i,j)\mathrm{e}^{-q_f(i,j)\left(1-\frac{z_f(i,j)}{N}\right)}\right)^2 \end{aligned} \tag{5.3}$$

is the *consistency term*, which measures the faithfulness of the solution to the data, and $\widehat{S}(f) + \widehat{S}(r)$ is the *regularization term*, or *smoothness term*, which is chosen according to the properties which the estimated source has to satisfy, and measures the faithfulness of the estimated source

to a priori informations. Moreover, the last terms of (5.2) are given by

$$S(q_v) = \sum_{i=1}^{n}\sum_{j=1}^{m} \lambda_v^2 \Big(q_v(i,j) - q_v(i-1,j) \Big)^2 + \sum_{i=1}^{n}\sum_{j=1}^{m} \lambda_v^2 \Big(q_v(i,j) - q_v(i,j-1) \Big)^2,$$

where $v \in \{f,r\}$, λ_v is the regularization parameter related to the interference level of the recto (resp. verso), if $v = f$ (resp., $v = r$), and

$$S_c(q_f, q_r) = \sum_{i=1}^{n}\sum_{j=1}^{m} \lambda_c^2 (q_f(i,j) - q_r(i,j))^2$$

is the joint smoothness term. The parameter λ_c is the regularization parameter between the interference of the recto and the verso with respect to the same pixel.

5.2 Alternating techniques

To minimize the function in (5.2), we use a strategy of *alternating minimization*, which consists of the estimation of the minimum of the function with respect to each single variable, fixing the other ones. We proceed according the following scheme:

$k = 0$

initialize f_0, r_0, q_0, A_0

while a stationary point of E is not found

$k = k+1$

$f_k = \arg\min_f E(f, r_{k-1}, q_{k-1}, A_{k-1})$

$r_k = \arg\min_r E(f_{k-1}, r, q_{k-1}, A_{k-1})$

$q_k = \arg\min_q E(f_{k-1}, r_{k-1}, q, A_{k-1})$

$A_k = \arg\min_A E(f_{k-1}, r_{k-1}, q_{k-1}, A)$

To solve the problem of minimization of the dual energy, which in general is not convex, a technique introduced by Blake and Zisserman, called GNC, can be used (see, e.g., [18, 24, 25, 33, 34, 87, 127, 129, 130, 131, 140]). With such a technique, the energy function E, is approximated by means of a finite family $\{E^{(p)}\}$ of functions, in such a way that the first one is convex and the last one coincides with the given function. Moreover, we call $\mathbf{x}$ the variable with respect to which we will compute the minimum of E. The minimization of each of the approximating functions $E^{(p)}$ can be done by means of an algorithm called NL-SOR (see, e.g., [38]).

5.3 Determining the interference levels

In this work, we deal with finding only the interference levels, fixed the recto, the verso and the blur mask. The other steps of the alternating algorithm will be treated in forthcoming papers. The energy function with respect to the interference level is given by

$$\begin{aligned} & E(f_{k-1}, r_{k-1}, q, A_{k-1}) = \sum_{i=1}^{n}\sum_{j=1}^{m}\left(f^s(i,j) - f(i,j)\mathrm{e}^{-q_r(i,j)\left(1-\frac{z_r(i,j)}{N}\right)}\right)^2 + \\ + & \sum_{i=1}^{n}\sum_{j=1}^{m}\left(r^s(i,j) - r(i,j)\mathrm{e}^{-q_f(i,j)\left(1-\frac{z_f(i,j)}{N}\right)}\right)^2 + \\ + & S(q_f) + S(q_r) + S_c(q_f, q_r) + k, \end{aligned} \tag{5.4}$$

where k is a constant depending on f_{k-1} and r_{k-1}. Now, let

$$\psi(f) = r, \quad \psi(r) = f. \tag{5.5}$$

Given $v \in \{f, r\}$ and fixed a pixel (k,t), the partial derivative of the regularization terms with respect to $q_v(k,t)$ is

$$\begin{aligned} & \frac{\partial(S(q_v) + S(q_{\psi(v)}) + S_c(q_v, q_{\psi(v)}))}{\partial q_v(k,t)} = \\ = & 2\lambda_v^2\Big(q_v(k,t) - q_v(k-1,t) + q_v(k,t) - q_v(k+1,t) + q_v(k,t) - q_v(k,t-1) + \\ + & q_v(k,t) - q_v(k,t+1)\Big) + 2\lambda_c^2\Big(q_v(k,t) - q_{\psi(v)}(k,t)\Big) = \\ = & 2\lambda_v^2\Big(4q_v(k,t) - q_v(k-1,t) - q_v(k+1,t) - q_v(k,t-1) - q_v(k,t+1)\Big) + \\ + & 2\lambda_c^2\Big(q_v(k,t) - q_{\psi(v)}(k,t)\Big). \end{aligned}$$

Now we consider the Hessian matrix $\mathcal{H}$ related to the function $S(q_v) + S(q_{\psi(v)}) + S_c(q_v, q_{\psi(v)})$. Fix $v \in \{f, r\}$ and a pixel (k,t), on the associated row of $\mathcal{H}$, in correspondence with the principal diagonal we have

$$\frac{\partial^2(S(q_v) + S(q_{\psi(v)}) + S_c(q_v, q_{\psi(v)}))}{\partial q_v^2(k,t)} = 8\lambda_v^2 + 2\lambda_c^2,$$

and the non-null terms are given by

$$\frac{\partial^2(S(q_v) + S(q_{\psi(v)}) + S_c(q_v, q_{\psi(v)}))}{\partial q_v(k,t)\,\partial\,\eta} = -2\lambda_v^2,$$

where $\eta \in \{q_v(k-1,t),\, q_v(k+1,t),\, q_v(k,t-1),\, q_v(k,t+1)\}$, and

$$\frac{\partial^2(S(q_v) + S(q_{\psi(v)}) + S_c(q_v, q_{\psi(v)}))}{\partial q_v(k,t)\,\partial\, q_{\psi(v)}(k,t)} = -2\lambda_c^2.$$

Since λ_v and λ_c are different from zero, $\mathcal{H}$ is irreducible, and for $k=1$ and $t=1$ the variable η assumes only two values, then by virtue of the Gerschgorin theorems (see, e.g., [72]) the matrix $\mathcal{H}$ is positive-definite, and hence the sum of the smoothness terms is a convex function.

Now, fixed $v \in \{f,r\}$ and a pixel (k,t), let us consider the consistency term $T(q) = T_v(q_{\psi(v)}) + T_{\psi(v)}(q_v)$, related to the interference level of (5.4). We get:

$$\begin{aligned}
&\frac{\partial T(q)}{\partial q_v(k,t)} = \\
&= 2\left((\psi(v))^s(k,t) - \psi(v)(k,t)e^{-q_v(k,t)\left(1-\frac{z_v(k,t)}{N}\right)}\right)\psi(v)(k,t)e^{-q_v(k,t)\left(1-\frac{z_v(k,t)}{N}\right)}\left(1-\frac{z_v(k,t)}{N}\right) = \\
&= 2\left(1-\frac{z_v(k,t)}{N}\right)\psi(v)(k,t)\left(\psi(v)^s(k,t)e^{-q_v(k,t)\left(1-\frac{z_v(k,t)}{N}\right)} - \psi(v)(k,t)e^{-2q_v(k,t)\left(1-\frac{z_v(k,t)}{N}\right)}\right);
\end{aligned}$$

$$\begin{aligned}
\frac{\partial^2 T(q)}{\partial q_v(k,t)^2} &= 2\left(1-\frac{z_v(k,t)}{N}\right)^2\psi(v)(k,t) \\
&\left(-(\psi(v))^s(k,t)e^{-q_v(k,t)\left(1-\frac{z_v(k,t)}{N}\right)} + 2\psi(v)(k,t)e^{-2q_v(k,t)\left(1-\frac{z_v(k,t)}{N}\right)}\right).
\end{aligned} \tag{5.6}$$

Since the second mixed derivatives are equal to zero, then the Hessian matrix is a diagonal matrix, whose elements are given in (5.6). Such elements are positive if and only if

$$q_v(k,t) < -\frac{\ln\left(\frac{(\psi(v))^s(k,t)}{2\psi(v)(k,t)}\right)}{1-\frac{z_v(k,t)}{N}} \quad \text{for all } v \in \{f,r\},\ k \in \{1,\ldots,n\},\ t \in \{1,\ldots,m\}.$$

Thus, the energy function related to the interference level is given by the sum of the terms of data consistency, which are not necessarily convex, and the smoothness terms, which are convex. Hence, in general the uniqueness of the global minimum is not guaranteed.

5.4 Convex approximation of the data consistency term

To approximate the term of faithfulness to the data, it is possible to approximate T_r and T_f separately. Moreover, we can approximate each term of the sum in (5.3) separately too. Fixed $i \in \{1,\ldots,n\}$, $j \in \{1,\ldots,m\}$ and $v \in \{f,r\}$, let ψ be as in (5.5), and denoting by $\alpha = \dfrac{\psi(v)(i,j)}{\psi(v^s)(i,j)}$,

$\gamma = \dfrac{z_v(i,j)}{N} - 1$, $q = q_v(i,j)$, the term related to the faithfulness to the data can be expressed as

$$\begin{aligned} T(f,r,q,A) &= T_f(q_r) + T_r(q_f) = \sum_{i=1}^{n}\sum_{j=1}^{m}\left(f^s(i,j) - f(i,j)\mathrm{e}^{-q_r(i,j)\left(1-\frac{z_r(i,j)}{N}\right)}\right)^2 + \\ &+ \sum_{i=1}^{n}\sum_{j=1}^{m}\left(r^s(i,j) - r(i,j)\mathrm{e}^{-q_f(i,j)\left(1-\frac{z_f(i,j)}{N}\right)}\right)^2 = \\ &= \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{v\in\{f,r\}}\psi(v)^s(i,j)\,\varphi_{i,j,v}(q), \end{aligned} \tag{5.7}$$

where

$$\varphi_{i,j,v}(q) = \left(1-\alpha\, e^{q\gamma}\right)^2 = 1 - 2\alpha\, e^{q\gamma} + \alpha^2\, e^{2q\gamma}. \tag{5.8}$$

Therefore, to make the function $\varphi_{i,j,v}$ in (5.8) convex, we will proceed in several ways. In particular, in this work we approximate the quantity

$$g_{i,j,v}(q) = e^{q\gamma} \tag{5.9}$$

by a line of the type $\tilde{g}(q) = \tilde{A}\,q + \tilde{B}$. So, the approximation of $\varphi_{i,j,v}(q)$ is given by $\tilde{\varphi}(q) = (1 - \alpha\tilde{A}\,q - \tilde{B})^2$, which is a convex function, since $\tilde{\varphi}''(q) = 2\,\alpha^2\,\tilde{A}^2$.

5.4.1 Interpolating approximation

Now we approximate $g_{i,j,v}(q)$ with the line $p^{(1)}_{i,j,v}(q)$ interpolating at the points $(3, e^{3\gamma})$ and $(0,1)$. To compute the interpolating polynomial, we use the Lagrange method, obtaining

$$p^{(1)}_{i,j,v}(q) = L_0(q)\bar{y}_0 + L_1(q)\bar{y}_1,$$

where $L_0(q), L_1(q)$ are the Lagrange polynomial bases defined by

$$L_0(q) = \frac{q-\bar{x}_1}{\bar{x}_0 - \bar{x}_1} = \frac{q}{3}$$

and

$$L_1(q) = \frac{q-\bar{x}_0}{\bar{x}_1 - \bar{x}_0} = -\frac{1}{3}(q-3),$$

and so we get

$$p^{(1)}_{i,j,v}(q) = q\left(\frac{e^{3\gamma}}{3} - \frac{1}{3}\right) + 1.$$

Thus, the convex approximation of $\varphi_{i,j,v}(q)$ is given by

$$\varphi^{(1)}_{i,j,v}(q) = \left(1 - \alpha\, q\frac{e^{3\gamma}-1}{3} - \alpha\right)^2. \tag{5.10}$$

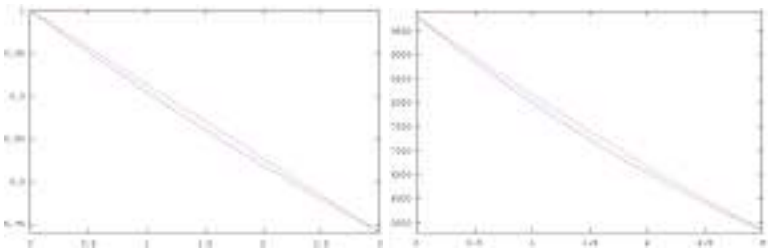


Figure 5.1: On the left side: Graph of $g_{i,j,v}(q)$ in blue and $p^{(1)}(q)$ in red. On the right side: Graph of $\varphi_{i,j,v}(q)$ in blue and $\varphi^{(1)}(q)$ in red ($\alpha = 100$, $\gamma = 0.1$).

5.4.2 The best line approximation

Now we approximate $g_{i,j,v}(q)$ by the line $p^{(2)}_{i,j,v}(q)$ of best approximation with respect to the 2-norm in $P_1([0,3]) = \{p : [0,3] \to \mathbb{R} |\ p \text{ is a polynomial of degree at most } 1\}$. We begin with using the Gram-Schmidt method to find an orthonormal basis $\{e_1, e_2\}$ for $P_1([0,3])$. We choose as basis the following polynomials:

$$x_1 = 1, \quad x_2 = q.$$

We normalize the first basis polynomial. We have

$$e_1 = \frac{x_1}{\|x_1\|} = \frac{1}{\sqrt{3}} = \frac{\sqrt{3}}{3}.$$

By applying the Gram-Schmidt algorithm, we get

$$z_2 = x_2 - < x_2, e_1 > e_1 = q_r - \frac{1}{3}\int_0^3 q\, dq = q - \frac{1}{3}\left[\frac{q^2}{2}\right]_0^3 = q - \frac{3}{2}.$$

By normalizing z_2 we obtain

$$e_2 = \frac{z_2}{\|z_2\|} = \frac{2}{3}\left(q - \frac{3}{2}\right),$$

where

$$\|z_2\| = \sqrt{\int_0^3 \left(q - \frac{3}{2}\right)^2 dq} = \frac{3}{2}.$$

So, we have constructed an orthonormal basis $\{e_1, e_2\}$ of $P_1([0,3])$. Thus, the best approximation polynomial of $g_{i,j,v}(q)$ is given by

$$p^{(3)}_{i,j,v}(q) = c_1 e_1 + c_2 e_2,$$

where

$$c_1 = < g_{i,j,v}, e_1 > = \int_0^3 g_{i,j,v}(q) e_1\, dq = \frac{\sqrt{3}}{3}\int_0^3 e^{q\gamma}\, dq = \frac{\sqrt{3}}{3}\left[\frac{e^{q\gamma}}{\gamma}\right]_0^3 = \frac{\sqrt{3}}{3}\frac{e^{3\gamma}}{\gamma} - \frac{\sqrt{3}}{3}\frac{1}{\gamma},$$

$$c_2 =< g, e_2 >= \int_0^3 g(q) e_2 \, dq = \frac{2}{3}\int_0^3 e^{q\gamma}\left(q-\frac{3}{2}\right) dq = \frac{2}{3}\int_0^3 e^{q\gamma} q \, dq - \int_0^3 e^{q\gamma} \, dq =$$

$$= \frac{2}{3}\left\{\left[\frac{e^{q\gamma}}{\gamma} q\right]_0^3 - \int_0^3 \frac{e^{q\gamma}}{\gamma} \, dq\right\} - \left[\frac{e^{q\gamma}}{\gamma}\right]_0^3 =$$

$$= \frac{2}{3}\left\{3\frac{e^{3\gamma}}{\gamma} - \left[\frac{e^{q\gamma}}{\gamma^2}\right]_0^3\right\} - \frac{e^{3\gamma}}{\gamma} + \frac{1}{\gamma} = 2\frac{e^{3\gamma}}{\gamma} - \frac{2}{3}\frac{e^{3\gamma}}{\gamma^2} + \frac{2}{3}\frac{1}{\gamma^2} - \frac{e^{3\gamma}}{\gamma} + \frac{1}{\gamma}.$$

Thus, we have

$$p^{(3)}_{i,j,v}(q) = c_1 e_1(q) + c_2 e_2(q) =$$

$$= \frac{\sqrt{3}}{3}\left(\frac{\sqrt{3}}{3}\frac{e^{3\gamma}}{\gamma} - \frac{\sqrt{3}}{3}\frac{1}{\gamma}\right) + \frac{2}{3}\left(q-\frac{3}{2}\right)\left(2\frac{e^{3\gamma}}{\gamma} - \frac{2}{3}\frac{e^{3\gamma}}{\gamma^2} + \frac{2}{3}\frac{1}{\gamma^2} - \frac{e^{3\gamma}}{\gamma} + \frac{1}{\gamma}\right) =$$

$$= \frac{1}{3}\frac{e^{3\gamma}}{\gamma} - \frac{1}{3}\frac{1}{\gamma} + \frac{4}{3}\frac{e^{3}\gamma}{\gamma} q - \frac{4}{9}\frac{e^{3\gamma}}{\gamma^2} q + \frac{4}{9}\frac{1}{\gamma^2} q - \frac{2}{3}\frac{e^{3\gamma}}{\gamma} q + \frac{2}{3}\frac{1}{\gamma} q - 2\frac{e^{3\gamma}}{\gamma} + \frac{2}{3}\frac{e^{3\gamma}}{\gamma^2} - \frac{2}{3}\frac{1}{\gamma^2} + \frac{e^{3\gamma}}{\gamma} - \frac{1}{\gamma} =$$

$$= \frac{2}{3}\frac{1}{\gamma}\left(2e^{3\gamma} - \frac{2}{3}\frac{e^{3\gamma}}{\gamma} + \frac{2}{3}\frac{1}{\gamma} - e^{3\gamma} + 1\right) - \frac{5}{3}\frac{e^{3\gamma}}{\gamma} - \frac{4}{3}\frac{1}{\gamma} + \frac{2}{3}\frac{e^{3\gamma}}{\gamma^2} - \frac{2}{3}\frac{1}{\gamma^2} + \frac{e^{3\gamma}}{\gamma},$$

and hence we obtain $p^{(2)}_{i,j,v}(q) = A^{(2)} q + B^{(2)}$, where

$$\begin{aligned} A^{(2)} &= \frac{2}{3\gamma}\left(2e^{3\gamma} - \frac{2(e^{3\gamma}-1)}{3\gamma} - e^{3\gamma} + 1\right), \\ B^{(2)} &= -\frac{5\,e^{3\gamma}}{3\gamma} - \frac{4}{3\gamma} + \frac{2e^{3\gamma}}{3\gamma^2} - \frac{2}{3\gamma^2} + \frac{e^{3\gamma}}{\gamma}. \end{aligned}$$

The convex approximation of $\varphi_{i,j,v}(q)$ is

$$\varphi^{(2)}_{i,j,v}(q) = \left(1 - \alpha A^{(2)} q - B^{(2)}\right)^2. \tag{5.11}$$

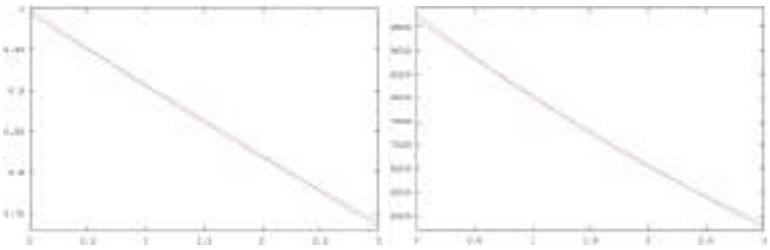


Figure 5.2: On the left side: Graph of $g_{i,j,v}(q)$ in blue and $p^{(2)}(q)$ in red. On the right side: Graph of $\varphi_{i,j,v}(q)$ in blue and $\varphi^{(2)}(q)$ in red ($\alpha = 100$, $\gamma = 0.1$).

5.4.3 Hybrid best approximation and interpolation

Now we approximate $g_{i,j,v}(q)$ by means of the line $p^{(5)}_{i,j,v}(q)$ of best approximation related to the 2-norm in $P_1([0,3]) = \{p : [0,3] \to \mathbb{R} |\ p$ is a polynomial of degree at most $1\}$, which interpolates $g_{i,j,v}(q)$ at a chosen point $\bar{q}$.

Now we make a change of coordinates, in such a way that the "new" origin coincides with $(\bar{q}, g_{i,j,v}(\bar{q}))$. Let us define

$$\tilde{g}(q) = g_{i,j,v}(q+\bar{q}) - g_{i,j,v}(\bar{q}). \tag{5.12}$$

Note that $\tilde{g}$ is a translation of $g_{i,j,v}$ in the Cartesian plane. Now we determine the polynomial $\tilde{p}^{(5)}$ of best approximation of $\tilde{g}$ with respect to the 2-norm in

$$P_1([-\bar{q}, 3-\bar{q}]) = \{p : [-\bar{q}, 3-\bar{q}] \to \mathbb{R} \mid p \text{ is a polynomial of degree at most } 1\}$$

which interpolates $\tilde{g}$ at 0.

We use the Gram-Schmidt method to find an orthonormal basis $\{e_1\}$ of the space

$$\{p \in P_1([-\bar{q}, 3-\bar{q}]) | \, p(0) = 0\}.$$

A basis for this space is given by $x_1 = q$. By normalizing, we get

$$||x_1|| \quad = \quad \sqrt{\int_{-\bar{q}}^{3-\bar{q}} q^2 \, dq} = \sqrt{3}\,(\sqrt{\bar{q}^2 - 3\bar{q} + 3}).$$

So, the normalized basis is given by

$$e_1 = \frac{x_1}{||x_1||} = \frac{q\sqrt{3}}{3(\sqrt{\bar{q}^2 - 3\bar{q} + 3})}.$$

The polynomial $\tilde{p}^{(5)}(q)$ of best approximation of $\tilde{g}_{i,j,v}(q)$ is

$$\tilde{p}^{(5)}(q) = c_1 e_1(q),$$

where

$$\begin{aligned} c_1 &= <\tilde{g}, e_1> = \int_{-\bar{q}}^{3-\bar{q}} \tilde{g}(q)\, e_1(q)\, dq = \\ &= -\frac{\sqrt{3}}{3(\sqrt{\bar{q}^2 - 3\bar{q} + 3})} e^{\bar{q}\gamma} \left(\frac{(\bar{q}-3)e^{(3-\bar{q})\gamma} - \bar{q}\, e^{-\bar{q}\gamma}}{\gamma} + \frac{e^{(3-\bar{q})\gamma} - e^{-\bar{q}\gamma}}{\gamma^2} + \frac{9 - 6\bar{q}}{2} \right). \end{aligned}$$

Therefore, we get

$$\begin{aligned} \tilde{p}^{(5)}_{i,j,v}(q) &= -\frac{q}{6(\bar{q}^2 - 3\bar{q} + 3)} e^{\bar{q}\gamma} \left(\frac{(2\bar{q}-6)e^{(3-\bar{q})\gamma} - 2\bar{q}\, e^{-\bar{q}\gamma}}{\gamma} + \right. \\ &+ \left. 2\frac{e^{(3-\bar{q})\gamma} - e^{-\bar{q}\gamma}}{\gamma^2} + 9 - 6\bar{q} \right). \end{aligned}$$

By taking the inverse translation, we obtain

$$p^{(5)}_{i,j,v}(q) = \tilde{p}^{(5)}(q-\bar{q}) + g_{i,j,v}(\bar{q}) = A^{(5)}(\bar{q})\,q + B^{(5)}(\bar{q}),$$

where

$$\begin{aligned} A^{(5)}(\bar{q}) &= -\frac{e^{\bar{q}\gamma}}{6(\bar{q}^2-3\bar{q}+3)}\left(\frac{(2\bar{q}-6)e^{(3-\bar{q})\gamma}-2\bar{q}\,e^{-\bar{q}\gamma}}{\gamma}+\right. \\ &+ \left. 2\frac{e^{(3-\bar{q})\gamma}-e^{-\bar{q}\gamma}}{\gamma^2}+9-6\bar{q}\right), \\ B^{(5)}(\bar{q}) &= \frac{\bar{q}\,e^{\bar{q}\gamma}}{6(\bar{q}^2-3\bar{q}+3)}\left(\frac{(2\bar{q}-6)e^{(3-\bar{q})\gamma}-2\bar{q}\,e^{-\bar{q}\gamma}}{\gamma}+\right. \\ &+ \left. 2\frac{e^{(3-\bar{q})\gamma}-e^{-\bar{q}\gamma}}{\gamma^2}+9-6\bar{q}\right)+e^{\bar{q}\gamma}. \end{aligned}$$

In particular, for $\bar{q}=0$ we get

$$\begin{aligned} A^{(3)} &= A^{(5)}(0) = \frac{1}{18}\left(\frac{6e^{3\gamma}}{\gamma}-2\frac{e^{3\gamma}-1}{\gamma^2}-9\right), \\ B^{(3)} &= B^{(5)}(0) = \frac{1}{18}e^{3\gamma}\left(\frac{6e^{-3\gamma}}{\gamma}+2\frac{e^{-3\gamma}-1}{\gamma^2}+9\right), \end{aligned}$$

while for $\bar{q}$=3 we have

$$\begin{aligned} A^{(4)} &= A^{(5)}(3) = \frac{1}{18}e^{3\gamma}\left(\frac{6e^{-3\gamma}}{\gamma}+2\frac{e^{-3\gamma}-1}{\gamma^2}+9\right), \\ B^{(4)} &= B^{(5)}(3) = \frac{1}{6}e^{3\gamma}\left(\frac{6e^{-3\gamma}}{\gamma}+2\frac{e^{-3\gamma}-1}{\gamma^2}+9\right)+e^{3\gamma}. \end{aligned}$$

From this it follows that two possible convex approximation of $\varphi_{i,j,v}(q)$ are

$$\varphi^{(\kappa)}_{i,j,v}(q) = \left(1-\alpha A^{(\kappa)}q-\alpha B^{(\kappa)}\right)^2, \quad \kappa = 3,4. \tag{5.13}$$

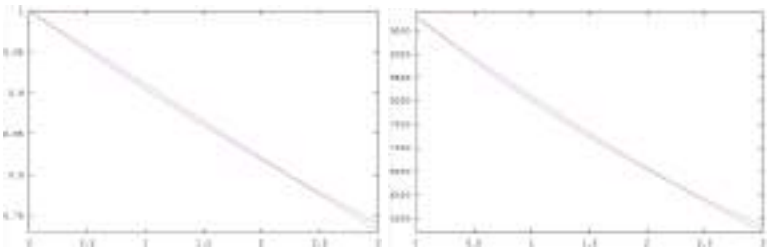


Figure 5.3: On the left side: Graph of $g_{i,j,v}(q)$ in blue and $p^{(3)}(q)$ in red. On the right side: Graph of $\varphi_{i,j,v}(q)$ in blue and $\varphi^{(3)}(q)$ in red ($\alpha = 100$, $\gamma = 0.1$).

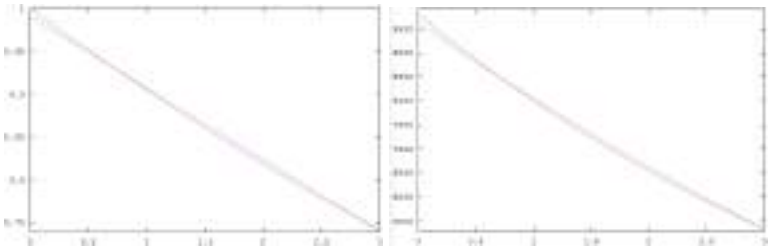


Figure 5.4: On the left side: Graph of $g_{i,j,v}(q)$ in blue and $p^{(4)}(q)$ in red. On the right side: Graph of $\varphi_{i,j,v}(q)$ in blue and $\varphi^{(4)}(q)$ in red ($\alpha = 100$, $\gamma = 0.1$).

5.5 The GNC approximation families

The first convex approximation of the consistency term of the energy function related to the interference level of the verso is expressed by

$$T^{(\kappa)}(f,r,q,A) = \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{v\in\{f,r\}} \psi(v)^s(i,j)\,\varphi_{i,j,v}^{(\kappa)}(q), \quad \kappa = 1,2,3,4. \tag{5.14}$$

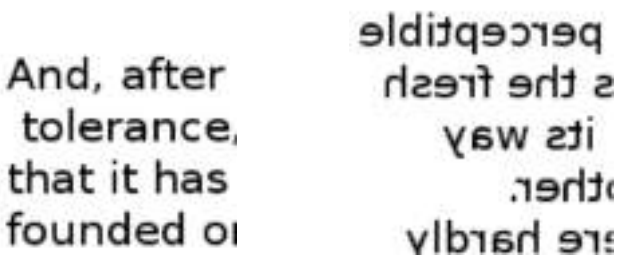
In this section, we define the following families of functions of convex approximations. Fixed $\kappa = 1,2,3,4$, let

$$T_p^{(\kappa)} = pT^{(\kappa)} + (1-p)T.$$

For $p = 1$, we get the first convex approximation associated with κ, while for $p = 0$ we have the original function T.

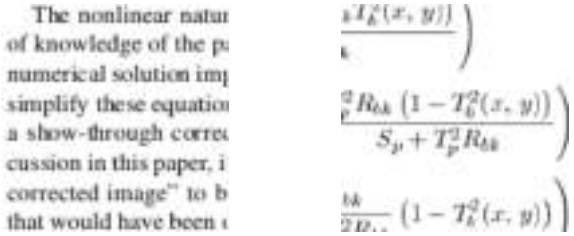
5.6 Experimental results

In this section we compare the experimental results, by using the four different GNC algorithms proposed in the previous sections. We have assumed two different pairs of original images, given in Figures 5.5 and 5.6. We have used a uniform blur mask of dimension 5×5, and we have considered the interference levels to be estimated given in the Figure 5.7. In this figure, if the interference value of a single pixel is 0, that pixel is presented in black while, if the interference value is 3 (that is very high), then that pixel is presented in white. The gray pixels represent interference values between 0 and 3. Note that we have assumed that the ideal interference levels of the recto and of the verso coincide. Considering the first pair of original sources given in Figure 5.5 and the interference levels given by 5.7, we obtain the data mixtures given in Figure 5.8. We have tested the four GNC algorithms, assuming the following regularization parameters:



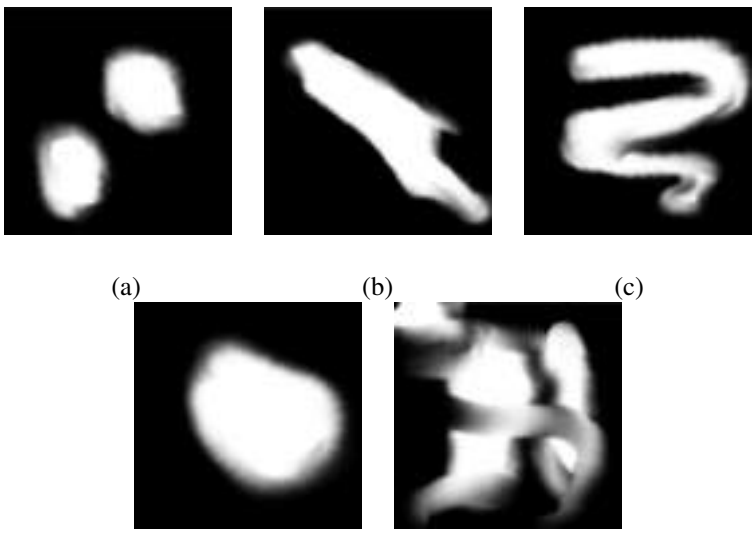
(a) (b)

Figure 5.5: First pair of ideal sources



(a) (b)

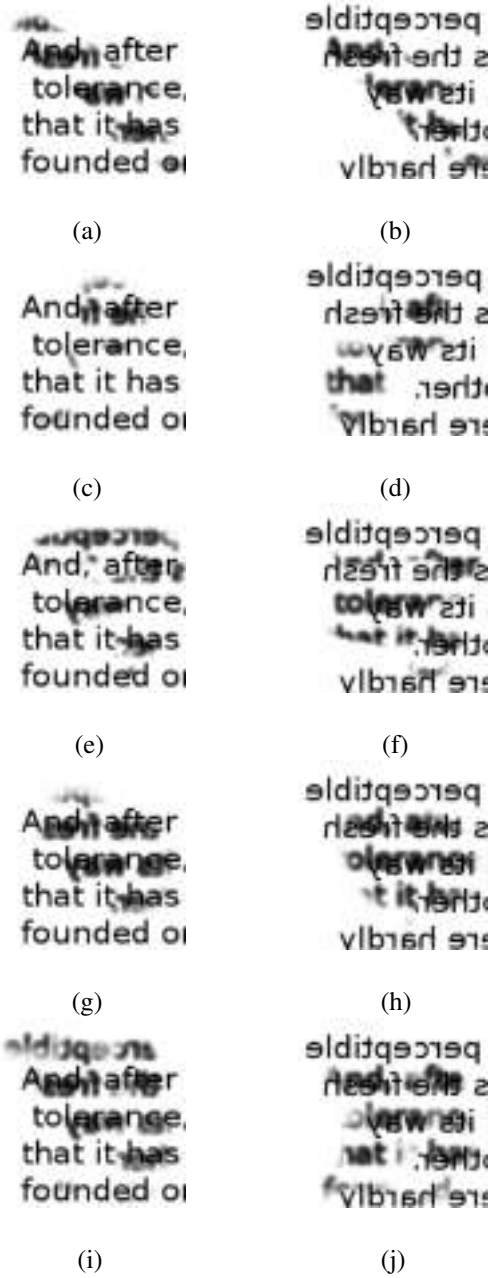
Figure 5.6: Second pair of ideal sources



(a) (b) (c)

(d) (e)

Figure 5.7: Interference levels: (a) $q_f^{(1)} = q_r^{(1)}$; (b) $q_f^{(2)} = q_r^{(2)}$; (c) $q_f^{(3)} = q_r^{(3)}$; (d) $q_f^{(4)} = q_r^{(4)}$; (e) $q_f^{(5)} = q_r^{(5)}$.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

(i) (j)

Figure 5.8: (a) Image in Figure 5.5 (a) with degraded by the interference level in Figure 5.7 (a); (b) image in Figure 5.5 (b) with degraded by the interference level in Figure 5.7 (a); (c) image in Figure 5.5 (a) with degraded by the interference level in Figure 5.7 (b); (d) image in Figure 5.5 (b) with degraded by the interference level in Figure 5.7 (b); (e) image in Figure 5.5 (a) with degraded by the interference level in Figure 5.7 (c); (f) image in Figure 5.5 (b) with degraded by the interference level in Figure 5.7 (c); (g) image in Figure 5.5 (a) with degraded by the interference level in Figure 5.7 (d); (h) image in Figure 5.5 (b) with degraded by the interference level in Figure 5.7 (d); (i) image in Figure 5.5 (a) with degraded by the interference level in Figure 5.7 (e); (j) image in Figure 5.5 (b) with degraded by the interference level in Figure 5.7 (e)

$$\lambda_f = \lambda_r = 50, \quad \lambda_c = 100.$$

We have compared the four proposed algorithms in terms of mean square error (MSE) between the estimate of the obtained interference level and the ideal interference level given in Figure 5.7. In Table 5.1 the errors in terms of MSE obtained by the proposed algorithms, by considering the pair of original sources presented in Figure 5.5 and the interference levels $q_f^{(i)} = q_r^{(i)}$, $i = 1, \ldots, 5$, given in Figure 5.7. The effectiveness of the algorithm has been tested by using some images,

	$q_f^{(1)}$	$q_r^{(1)}$	$q_f^{(2)}$	$q_r^{(2)}$	$q_f^{(3)}$	$q_r^{(3)}$	$q_f^{(4)}$	$q_r^{(4)}$	$q_f^{(5)}$	$q_r^{(5)}$
$\kappa = 1$	0.05712	0.05332	0.07191	0.08313	0.07980	0.07843	0.08213	0.07920	0.10213	0.09871
$\kappa = 2$	0.03575	0.03741	0.03375	0.04121	0.04401	0.03979	0.03142	0.02673	0.03725	0.04165
$\kappa = 3$	0.03176	0.04002	0.03723	0.03937	0.04183	0.04272	0.02734	0.02639	0.04128	0.03984
$\kappa = 4$	0.02144	0.02347	0.02317	0.02242	0.03143	0.03031	0.00979	0.01127	0.03521	0.03878

Table 5.1: MSE of the proposed algorithms, using the original sources in Figure 5.5.

created to highlight the capacity of the algorithm to eliminate the degradations due to the effect of show-through.

As we see in Table 5.1, the best algorithm for the first pair of images is that related to the family of approximations with $\kappa = 4$. In Figure 5.9 there are the interference levels estimated by the algorithm corresponding with $\kappa = 4$.

Considering the second pair of original sources given in Figure 5.6 and the interference levels given by 5.7, and considering a uniform mask of type 5×5, we obtain the data mixtures given in Figure 5.10. We have tested the four GNC algorithms, using the previous regularization parameters. In Table 5.2 the errors in terms of MSE obtained by the proposed algorithms, by considering the pair of original sources presented in Figure 5.6 and the interference levels $q_f^{(i)} = q_r^{(i)}$, $i = 1, \ldots, 5$, given in Figure 5.7.

	$q_f^{(1)}$	$q_r^{(1)}$	$q_f^{(2)}$	$q_r^{(2)}$	$q_f^{(3)}$	$q_r^{(3)}$	$q_f^{(4)}$	$q_r^{(4)}$	$q_f^{(5)}$	$q_r^{(5)}$
$\kappa = 1$	0.05229	0.04973	0.06127	0.06712	0.09415	0.08174	0.12176	0.13746	0.11288	0.15672
$\kappa = 2$	0.03374	0.03019	0.03626	0.03372	0.04791	0.03277	0.03711	0.04424	0.03845	0.04064
$\kappa = 3$	0.03147	0.03133	0.03317	0.03533	0.03179	0.03375	0.04176	0.05727	0.03973	0.03927
$\kappa = 4$	0.02379	0.02517	0.02433	0.02536	0.01017	0.01752	0.02578	0.03225	0.03320	0.03584

Table 5.2: MSE of the proposed algorithms, using the original sources in Figure 5.6.

As we see in Table 5.2, the best algorithm for the first pair of images is again that related to the family of approximations with $\kappa = 4$. In Figure 5.11 the interference levels estimated by the

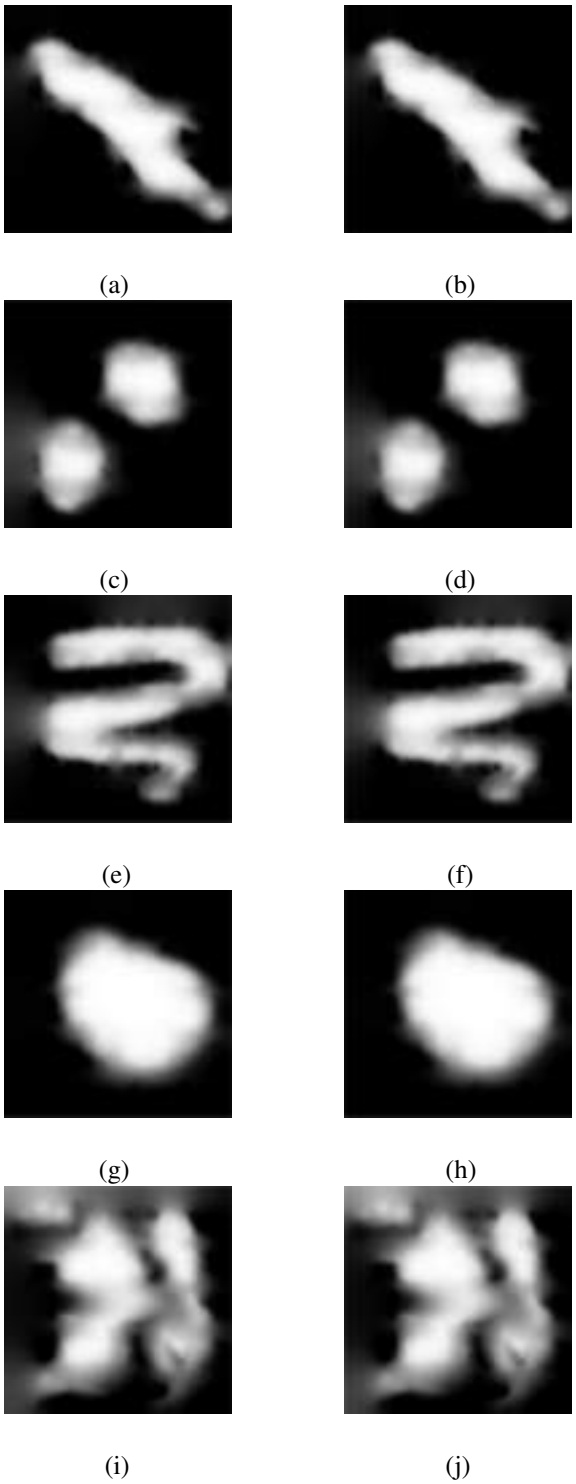


Figure 5.9: (a) estimation of $q_f^{(1)}$; (b) estimation of $q_r^{(1)}$; (c) estimation of $q_f^{(2)}$; (d) estimation of $q_r^{(2)}$; (e) estimation of $q_f^{(3)}$; (f) estimation of $q_r^{(3)}$; (g) estimation of $q_f^{(4)}$; (h) estimation of $q_r^{(4)}$; (i) estimation of $q_f^{(5)}$; (j) estimation of $q_r^{(5)}$.

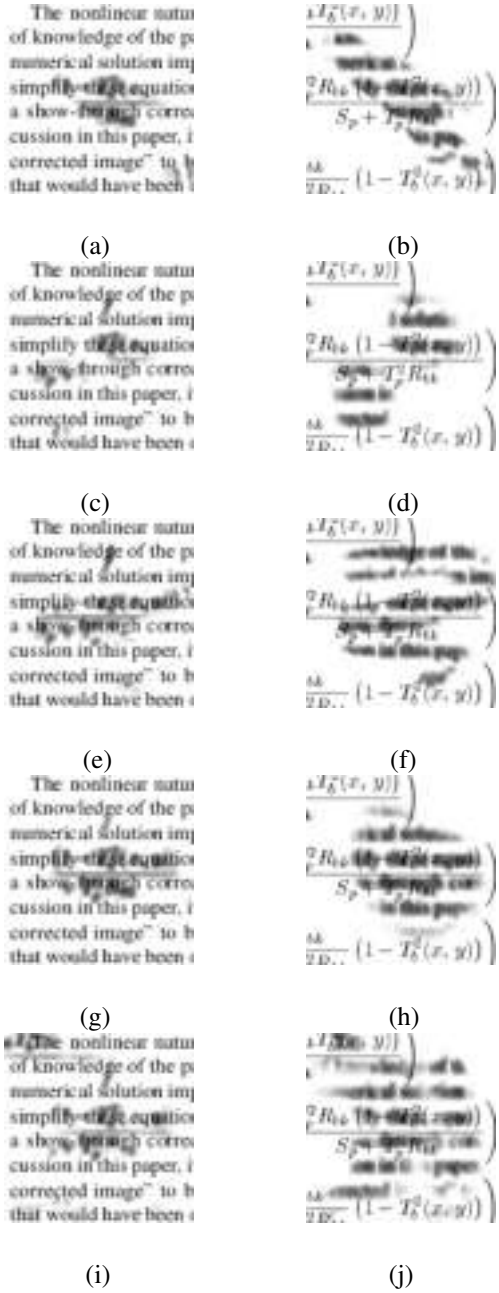


Figure 5.10: (a) Image in Figure 5.6 (a) with degraded by the interference level in Figure 5.7 (a); (b) image in Figure 5.6 (b) with degraded by the interference level in Figure 5.7 (a); (c) image in Figure 5.6 (a) with degraded by the interference level in Figure 5.7 (b); (d) image in Figure 5.6 (b) with degraded by the interference level in Figure 5.7 (b); (e) image in Figure 5.6 (a) with degraded by the interference level in Figure 5.7 (c); (f) image in Figure 5.6 (b) with degraded by the interference level in Figure 5.7 (c); (g) image in Figure 5.6 (a) with degraded by the interference level in Figure 5.7 (d); (h) image in Figure 5.6 (b) with degraded by the interference level in Figure 5.7 (d); (i) image in Figure 5.6 (a) with degraded by the interference level in Figure 5.7 (e); (j) image in Figure 5.6 (b) with degraded by the interference level in Figure 5.7 (e)

algorithm corresponding with $\kappa = 4$.

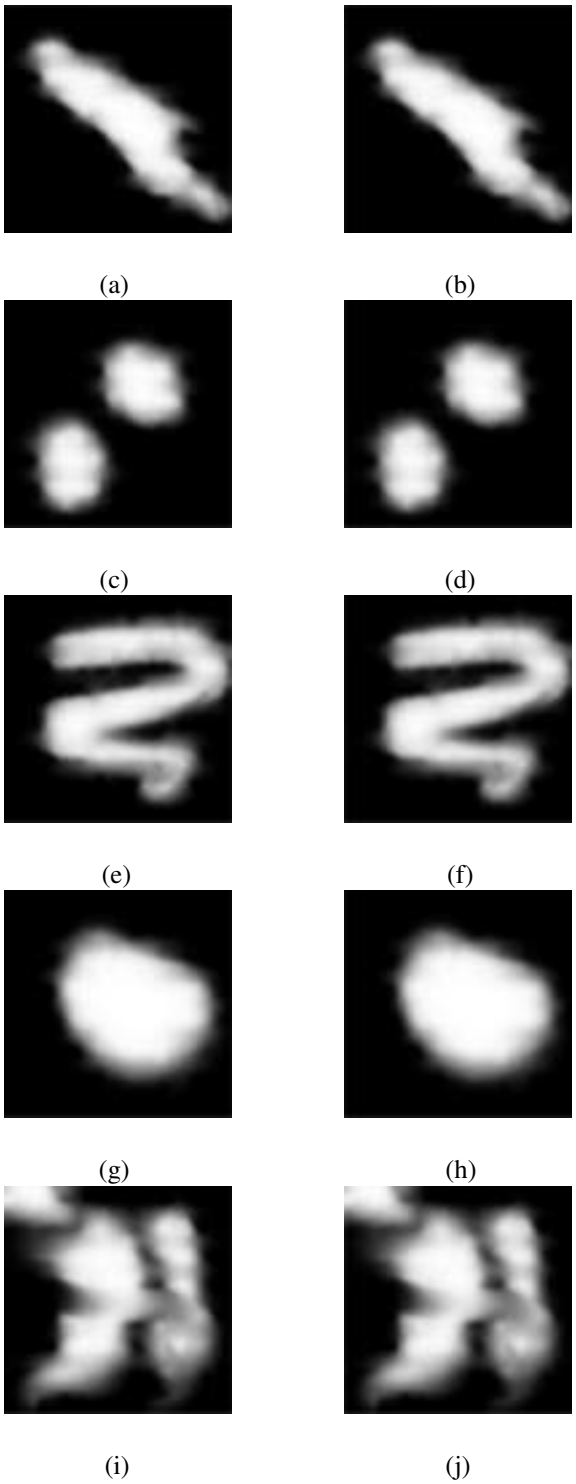


Figure 5.11: (a) estimation of $q_f^{(1)}$; (b) estimation of $q_r^{(1)}$; (c) estimation of $q_f^{(2)}$; (d) estimation of $q_r^{(2)}$; (e) estimation of $q_f^{(3)}$; (f) estimation of $q_r^{(3)}$; (g) estimation of $q_f^{(4)}$; (h) estimation of $q_r^{(4)}$; (i) estimation of $q_f^{(5)}$; (j) estimation of $q_r^{(5)}$.

Chapter 6

The problem of image restoration

In Section 6.1 we present the problem of image deblurring and the related regularization technique; in Section 6.2 we present a GNC-type technique for the minimization of the energy function; in Section 6.3 we investigate spectral properties of β-matrices; in Section 6.4 we deal with structural properties; in Section 6.5 we study the properties of the multiplications of our family of matrices; in Section 6.6 we determine some conditions in order that a β-matrix is invertible; in Section 6.7 we deal with the problem of approximating a real symmetric Toeplitz matrix by a β-matrix.

6.1 Regularization of the problem

The problem of image restoration consists of reconstructing the original image from an image blurred and/or corrupted by noise. In the sequel we will assume that all intensities of our involved pixels are put into one column, with the rule that $(i, j) < (i', j')$ if and only if $i < i'$ or $i = i'$ and $j < j'$. The direct problem is formulated as follows:

$$\boldsymbol{y} = A\boldsymbol{x} + \boldsymbol{n},$$

where the n^2-dimensional vectors $\boldsymbol{x}$, $\boldsymbol{y}$ are respectively the original and the observed image. In particular, the elements of these vectors indicate the light intensity of pixels in the corresponding image. The n^2-dimensional vector $\boldsymbol{n}$ expresses the additive noise on the image, which we assume to be independent and identically distributed (i.i.d.) Gaussian, with zero mean and known vari-

ance. The $n^2 \times n^2$ matrix A is a linear operator, which represents the translation invariant blur acting on the image. To obtain a blurred image, each pixel of original image turns to be equal to a weighted average of its neighbors. Given a positive matrix $M \in \mathbb{R}^{(2h+1)\times(2h+1)}$, called *blur mask,* the entries of matrix A are defined by

$$a_{(i,j),(i+w,j+v)} = \begin{cases} m_{h+1+w,h+1+v}, & \text{if } |w|,|v| \le h, \\ \\ 0, & \text{otherwise.} \end{cases}$$

Here, in lexicographic notation, the generic index $((i,j),(h,l))$ of matrix A is supposed to be equal to $((j-1)n+i,(l-1)n+h)$. The matrix A turns to be a block Toeplitz matrix with Toeplitz blocks. If we assume that the blur operator is uniform on each direction and is very wide (that is, $h \sim n$), then the matrix A is symmetric.

The image restoration problem consists of finding an estimation $\boldsymbol{x}$ of the unknown original image given the blurred image $\boldsymbol{y}$, the matrix A and the variance of the noise σ^2. This is an ill–posed inverse problem in the Hadamard sense.

A *clique* c of order k is the subset of points of a square grid on which the k–th order finite difference is defined. We denote by C_k the set of all cliques of order k. More precisely, we consider, for $k = 1$,

$$\begin{aligned} C_1 = \{c = \{(i,j),(h,l)\} : \quad & i = h, j = l+1 \text{ or} \\ & i = h+1, j = l\}; \end{aligned}$$

for $k = 2$,

$$\begin{aligned} C_2 = \ \{c = \{(i,j),(h,l),(r,q)\} : \\ i = h = r, j = l+1 = q+2, \text{ or} \\ i = h+1 = r+2, j = l = q\}; \end{aligned}$$

and for $k = 3$,

$$\begin{aligned} C_3 \ = \{c = \{(i,j),(h,l),(r,q),(w,z)\} : \\ i = h = r = w, j = l+1 = q+2 = z+3, \text{ or} \\ i = h+1 = r+2 = w+3, j = l = q = z\}. \end{aligned}$$

We denote by $D_c^k \boldsymbol{x}$ the k–th order finite difference operator of the vector $\boldsymbol{x}$ associated with the clique c, that is, if $c = \{(i,j),(h,l)\} \in C_1$, then

$$D_c^1 \boldsymbol{x} = x_{i,j} - x_{h,l};$$

if $c = \{(i,j),(h,l),(r,q)\} \in C_2$, then

$$D_c^2 \boldsymbol{x} = x_{i,j} - 2x_{h,l} + x_{r,q};$$

and if $c = \{(i,j),(h,l),(r,q),(w,z\} \in C_3$, then

$$D_c^3 \boldsymbol{x} = x_{i,j} - 3x_{h,l} + 3x_{r,q} - x_{w,q}.$$

In [34] it has been shown that the use of second order difference operators allows to obtain significantly better results than those obtained by first order difference operators. On the other hand, in [34] it is noted that third order difference operators give slightly better results than those obtained with second order difference operators to the detriment of an excessive increase in computational costs. Therefore we will only use second order difference operators, and hence we refer to C and D_c as C_2 and D_c^2. We associate with each clique c a non–negative weight b_c, called *line variable*, which has the role of dropping the regularity constraints, where discontinuities could appear. In particular, the zero value is associated with a discontinuity of the considered image in correspondence with the clique c. In our model, the original image is considered idealistically as a pair $(\boldsymbol{x},\boldsymbol{b})$, where $\boldsymbol{x}$, $\boldsymbol{b}$ are the vectors of the grey intensity of pixels and of the set of all line components b_c, $c \in C$, respectively.

A regularized solution of the investigated problem is the minimizer of the following function, called *primal energy function*, defined by

$$E(\boldsymbol{x},\boldsymbol{b}) = \|\boldsymbol{y} - A\boldsymbol{x}\|^2 + \sum_{c \in C} \left[\lambda^2 (D_c \boldsymbol{x})^2 b_c + \beta(b_c)\right], \tag{6.1}$$

where β is a suitable non-increasing function, called *balancing function*, and $\|\cdot\|$ is the Euclidean norm. The first term in the right hand indicates the faithfulness of the solution to the data and the last one is a regularization term, which is related to a smoothness condition on $\boldsymbol{x}$. The scalar parameter λ^2 is in connection with the confidence to the data and the degree of regularization of the solutions. In particular, when λ^2 is close to zero, we represent a strong faithfulness to the data, while when λ^2 is very large we have a confidence to the a priori information.

To find the minimum of the primal energy function (6.1), we first minimize with respect to $\boldsymbol{b}$. So, the dual energy function $E_d(\boldsymbol{x})$ (see, e.g., [23, 24, 34, 67]) is given by

$$E_d(\boldsymbol{x}) = \inf_{\mathbf{b} \in B^{|C|}} E(\boldsymbol{x},\boldsymbol{b}), \tag{6.2}$$

where $|C|$ is the cardinality of the set C. Observe that, by [35, Theorem 1], E_d is well-defined. Observe that

$$E_d(\boldsymbol{x}) = \|\boldsymbol{y} - A\boldsymbol{x}\|^2 + \sum_{c \in C} g(D_c \boldsymbol{x}), \tag{6.3}$$

where

$$g(t) = \inf_{b \in B} (\lambda^2 b t^2 + \beta(b)), \tag{6.4}$$

is the *potential function*, which associates a cost with each value of the finite difference operator and does not depend on the involved clique (see also [67]).

In general, to reduce computational costs, for reconstructing images, it is more advisable to use the dual energy rather the primal energy, because a lower number of variables have to be determined. Thus, some versions of the duality theorem were given in [33, 34] for energy functions which do not include the constraint of avoiding parallel lines. For other versions existing in the literature see, e.g., [10, 39, 46, 67].

6.2 GNC algorithm

In general, a function g satisfying duality theorems is not convex. So, neither is the dual energy function in 6.3. Thus, to minimize such a function, we use a GNC algorithm (see also [18, 34, 127, 129, 130, 131, 140]). The solution of the algorithms for minimizing a non–convex function depends on the choice of the initial point.

It is possible to verify experimentally that the more expensive minimization is the first one, because the other ones just start with a good approximation of the solution. Hence, in this thesis, when we minimize the first convex approximation, we propose to approximate every block of the operator A by means of matrices whose product can be computed by means a suitable fast discrete transform. Since every block of A is a symmetric Toeplitz matrix, we now deal with determining a class of matrices easy to handle from the computational point of view, that give a good approximation of the Toeplitz matrices.

6.3 Spectral characterization of β-matrices

We begin with presenting a new class of simultaneously diagonalizable matrices, so we define the following matrix. Let n be a fixed positive integer, and $Q_n = (q_{k,j}^{(n)})_{k,j}$, k, $j = 0, 1, \ldots, n-1$, where

$$q_{k,j}^{(n)} = \begin{cases} \alpha_j \cos\Big(\dfrac{2\pi k j}{n}\Big) & \text{if } 0 \le j \le \lfloor n/2 \rfloor, \\ \\ \alpha_j \sin\Big(\dfrac{2\pi k (n-j)}{n}\Big) & \text{if } \lfloor n/2 \rfloor \le j \le n-1, \end{cases} \tag{6.5}$$

$$\alpha_j = \begin{cases} \dfrac{1}{\sqrt{n}} = \overline{\alpha} & \text{if } j = 0, \text{ or } j = n/2 \text{ if } n \text{ is even}, \\ \\ \sqrt{\dfrac{2}{n}} = \widetilde{\alpha} & \text{otherwise}, \end{cases} \tag{6.6}$$

and put

$$Q_n = \Big(\mathbf{q}^{(0)} \Big| \mathbf{q}^{(1)} \Big| \cdots \Big| \mathbf{q}^{(\lfloor \frac{n}{2} \rfloor)} \Big| \mathbf{q}^{(\lfloor \frac{n+1}{2} \rfloor)} \Big| \cdots \Big| \mathbf{q}^{(n-2)} \Big| \mathbf{q}^{(n-1)} \Big), \tag{6.7}$$

where

$$\mathbf{q}^{(0)} = \frac{1}{\sqrt{n}} \Big(1\ 1\ \cdots\ 1\Big)^T = \frac{1}{\sqrt{n}}\, \mathbf{u}^{(0)}, \tag{6.8}$$

$$\begin{aligned} \mathbf{q}^{(j)} &= \sqrt{\frac{2}{n}} \left(1\ \cos\left(\frac{2\pi j}{n}\right) \cdots \cos\left(\frac{2\pi j(n-1)}{n}\right)\right)^T = \sqrt{\frac{2}{n}}\, \mathbf{u}^{(j)}, \\ \mathbf{q}^{(n-j)} &= \sqrt{\frac{2}{n}} \left(0\ \sin\left(\frac{2\pi j}{n}\right) \cdots \sin\left(\frac{2\pi j(n-1)}{n}\right)\right)^T = \sqrt{\frac{2}{n}}\, \mathbf{v}^{(j)}, \end{aligned} \tag{6.9}$$

$j = 1, 2, \ldots, \lfloor \frac{n-1}{2} \rfloor$. Moreover, when n is even, set

$$\mathbf{q}^{(n/2)} = \frac{1}{\sqrt{n}} \Big(1 - 1\ \ 1 - 1 \cdots - 1\Big)^T = \frac{1}{\sqrt{n}}\, \mathbf{u}^{(n/2)}. \tag{6.10}$$

In [111] it is proved that all columns of Q_n are orthonormal, and thus Q_n is an orthonormal matrix.

Now we define the following function. Given $\boldsymbol{\lambda} \in \mathbb{C}^n$, $\boldsymbol{\lambda} = (\lambda_0 \lambda_1 \cdots \lambda_{n-1})^T$, set

$$\operatorname{diag}(\boldsymbol{\lambda}) = \Lambda = \begin{pmatrix} \lambda_0 & 0 & 0 & \dots & 0 & 0 \\ 0 & \lambda_1 & 0 & \dots & 0 & 0 \\ 0 & 0 & \lambda_2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \lambda_{n-2} & 0 \\ 0 & 0 & 0 & \dots & 0 & \lambda_{n-1} \end{pmatrix},$$

where $\Lambda \in \mathbb{C}^{n \times n}$ is a diagonal matrix.

A vector $\boldsymbol{\lambda} \in \mathbb{R}^n$, $\boldsymbol{\lambda} = (\lambda_0 \lambda_1 \cdots \lambda_{n-1})^T$ is said to be *symmetric* (resp., *asymmetric*) iff $\lambda_j = \lambda_{n-j}$ (resp., $\lambda_j = -\lambda_{n-j}$) $\in \mathbb{R}$ for every $j = 0, 1, \dots, \lfloor n/2 \rfloor$.

Let Q_n be as in (6.7), and $\mathcal{G}_n$ be the space of the matrices *simultaneously diagonalizable* by Q_n, that is

$$\mathcal{G}_n = \operatorname{sd}(Q_n) = \{Q_n \Lambda Q_n^T : \Lambda = \operatorname{diag}(\boldsymbol{\lambda}), \boldsymbol{\lambda} \in \mathbb{R}^n\}.$$

A matrix belonging to $\mathcal{G}_n$, $n \in \mathbb{N}$, is called γ-*matrix*. Moreover, we define the following classes by

$$\mathcal{C}_n = \{Q_n \Lambda Q_n^T : \Lambda = \operatorname{diag}(\boldsymbol{\lambda}), \boldsymbol{\lambda} \in \mathbb{R}^n, \boldsymbol{\lambda} \text{ is symmetric}\}, \tag{6.11}$$

$$\mathcal{B}_n = \{Q_n \Lambda Q_n^T : \Lambda = \operatorname{diag}(\boldsymbol{\lambda}), \boldsymbol{\lambda} \in \mathbb{R}^n, \boldsymbol{\lambda} \text{ is asymmetric}\},$$

$$\begin{aligned} \mathcal{D}_n \quad = \quad & \{Q_n \Lambda Q_n^T : \Lambda = \operatorname{diag}(\boldsymbol{\lambda}), \boldsymbol{\lambda} \in \mathbb{R}^n, \boldsymbol{\lambda} \text{ is symmetric}, \\ & \lambda_0 = 0, \lambda_{n/2} = 0 \text{ if } n \text{ is even}\}, \end{aligned}$$

$$\begin{aligned} \mathcal{E}_n \quad = \quad & \{Q_n \Lambda Q_n^T : \Lambda = \operatorname{diag}(\boldsymbol{\lambda}),\ \boldsymbol{\lambda} \in \mathbb{R}^n, \lambda_j = 0,\ j = 1, \dots, n-1, \\ & j \neq n/2 \text{ when } n \text{ is even}\}. \end{aligned}$$

Proposition 6.3.1. *The class $\mathcal{G}_n$ is a matrix algebra of dimension n.*

Proof. We prove that $\mathcal{G}_n$ is an algebra. Let I_n be the identity $n \times n$-matrix. Since Q_n is orthogonal, then $Q_n I_n Q_n^T = I_n$. Hence, $I_n \in \mathcal{G}_n$.

If $C \in \mathcal{G}_n$, C is non-singular, $C = Q_n \Lambda Q_n^T$ and Λ is diagonal, then $C^{-1} = Q_n \Lambda^{-1} Q_n^T$, and hence $C^{-1} \in \mathcal{G}_n$, since Λ^{-1} is diagonal.

Moreover, if $C_r \in \mathcal{G}_n$, $\alpha_r \in \mathbb{R}$, $C_r = Q_n \Lambda_r Q_n^T$ and Λ_r is diagonal, $r = 1,2$, then $\alpha_1 C_1 + \alpha_2 C_2 = Q_n(\alpha_1 \Lambda_1 + \alpha_2 \Lambda_2) Q_n^T \in \mathcal{G}_n$, since $\alpha_1 \Lambda_1 + \alpha_2 \Lambda_2$ is diagonal. Furthermore, $C_1 C_2 = Q_n \Lambda_1 Q_n^T Q_n \Lambda_2 Q_n^T = Q_n \Lambda_1 \Lambda_2 Q_n^T$, since $\Lambda_1 \Lambda_2$ is diagonal. Therefore, $\mathcal{G}_n$ is an algebra.

Now we claim that $\dim(\mathcal{G}_n) = \dim(\boldsymbol{\lambda}) = n$. By contradiction, let $\boldsymbol{\lambda_1} \neq \boldsymbol{\lambda_2} \in \mathbb{R}^n$ be such that $Q_n \text{diag}(\boldsymbol{\lambda_1}) Q_n^T = Q_n \text{diag}(\boldsymbol{\lambda_2}) Q_n^T = C$. Then, the elements of $\boldsymbol{\lambda_2}$ are obtained by a suitable permutation of those of $\boldsymbol{\lambda_1}$. Since the order of the eigenvectors of C have been established, if a component $\lambda_j^{(1)}$ of $\boldsymbol{\lambda_1}$ is equal to a component $\lambda_k^{(2)}$ of $\boldsymbol{\lambda_2}$, then $\mathbf{q}^{(j)}$ and $\mathbf{q}^{(k)}$ belong to the same eigenspace, and hence $\lambda_j^{(1)} = \lambda_j^{(2)} = \lambda_k^{(1)} = \lambda_k^{(2)}$. This implies that $\boldsymbol{\lambda_1} = \boldsymbol{\lambda_2}$, a contradiction. This ends the proof.

□

Proposition 6.3.2. *The class $\mathcal{C}_n$ is a subalgebra of $\mathcal{G}_n$ of dimension $\lfloor \frac{n}{2} \rfloor + 1$.*

Proof. Obviously, $\mathcal{C}_n \subset \mathcal{G}_n$. Now we claim that $\mathcal{C}_n$ is an algebra. First, note that $I_n \in \mathcal{C}_n$, since $I_n = Q_n I_n Q_n^T$ and $I_n = \text{diag}(\mathbf{1})$, where $\mathbf{1} = (1\ 1 \cdots 1)^T$. So, $I_n \in \mathcal{C}_n$, since $\mathbf{1}$ is symmetric.

If $C \in \mathcal{C}_n$, C is non-singular, $C = Q_n \Lambda Q_n^T$, $\Lambda = \text{diag}(\boldsymbol{\lambda})$, $\boldsymbol{\lambda} = (\lambda_0\ \lambda_1 \cdots \lambda_{n-1})^T$ is symmetric, then $C^{-1} = Q_n \Lambda^{-1} Q_n^T$, and so $C^{-1} \in \mathcal{C}_n$, as $\Lambda^{-1} = \text{diag}(\boldsymbol{\lambda}')$, and $\boldsymbol{\lambda}' = (1/\lambda_0\ 1/\lambda_1 \cdots 1/\lambda_{n-1})^T$ is symmetric too.

If $C_r \in \mathcal{C}_n$, $\alpha_r \in \mathbb{R}$, $C_r = Q_n \Lambda_r Q_n^T$, $\Lambda_r = \text{diag}(\boldsymbol{\lambda}^{(r)})$, $\boldsymbol{\lambda}^{(r)} = (\lambda_0^{(r)}\ \lambda_1^{(r)} \cdots \lambda_{n-1}^{(r)})^T$ is symmetric, $r = 1,2$, then $\alpha_1 C_1 + \alpha_2 C_2 = Q_n(\alpha_1 \Lambda_1 + \alpha_2 \Lambda_2) Q_n^T \in \mathcal{C}_n$, since $\alpha_1 \Lambda_1 + \alpha_2 \Lambda_2 = \text{diag}(\boldsymbol{\lambda}^*)$, and $\boldsymbol{\lambda}^* = (\alpha_1 \lambda_0^{(1)} + \alpha_2 \lambda_0^{(2)} \cdots \alpha_1 \lambda_{n-1}^{(1)} + \alpha_2 \lambda_{n-1}^{(2)})^T$ is symmetric. Furthermore, $C_1 C_2 = Q_n \Lambda_1 \Lambda_2 Q_n^T$, since $\Lambda_1 \Lambda_2 = \text{diag}(\boldsymbol{\lambda}_*)$, and $\boldsymbol{\lambda}_* = (\lambda_0^{(1)} \lambda_0^{(2)} \cdots \lambda_{n-1}^{(1)} \lambda_{n-1}^{(2)})^T$ is symmetric. Therefore, $\mathcal{C}_n$ is an algebra.

Now we prove that $\dim(\mathcal{C}_n) = \lfloor \frac{n}{2} \rfloor + 1$. By the definition of $\mathcal{C}_n$, it is possible to choose at most $\lfloor \frac{n}{2} \rfloor + 1$ elements of $\boldsymbol{\lambda}$. The proof is analogous to that of the last part of Proposition 6.3.1. □

Proposition 6.3.3. *The class $\mathcal{B}_n$ is a linear subspace of $\mathcal{G}_n$, and has dimension $\lfloor \frac{n-1}{2} \rfloor$.*

Proof. First, let us prove that $\mathcal{B}_n$ is a linear subspace of $\mathcal{G}_n$. For $r = 1,2$, let $A_r \in \mathcal{B}_n$, $\alpha_r \in \mathbb{R}$, $\Lambda_r = \text{diag}(\boldsymbol{\lambda}^{(r)})$, with $\boldsymbol{\lambda}^{(r)}$ asymmetric, and $C_r = Q_n \Lambda_r Q_n^T$. Then $\alpha_1 C_1 + \alpha_2 C_2 = Q_n(\alpha_1 \Lambda_1 + \alpha_2 \Lambda_2) Q_n^T$ with $\alpha_1 \Lambda_1 + \alpha_2 \Lambda_2 = \text{diag}(\boldsymbol{\lambda}^*)$, and $\boldsymbol{\lambda}^* = (\alpha_1 \lambda_0^{(1)} + \alpha_2 \lambda_0^{(2)} \cdots \alpha_1 \lambda_{n-1}^{(1)} + \alpha_2 \lambda_{n-1}^{(2)})^T$ is asymmetric. Therefore, $\alpha_1 C_1 + \alpha_2 C_2 \in \mathcal{B}_n$. So, $\mathcal{B}_n$ is a linear subspace of $\mathcal{G}_n$.

Now we prove that $\dim(\mathcal{B}_n) = \lfloor \frac{n-1}{2} \rfloor$. By the definition of $\mathcal{B}_n$, it is possible to choose at most $\lfloor \frac{n-1}{2} \rfloor$ elements of $\boldsymbol{\lambda}$, because $\lambda_0 = 0$ and $\lambda_{n/2} = 0$ when n is even. The proof is analogous to that of the last part of Proposition 6.3.1. □

Similarly as in Propositions 6.3.1 and 6.3.2, it is possible to prove that $\mathcal{D}_n$ is a subalgebra of $\mathcal{G}_n$ of dimension $\lfloor \frac{n-1}{2} \rfloor$ and $\mathcal{E}_n$ is a subalgebra of $\mathcal{G}_n$ of dimension 1 when n is odd and 2 when n is even. Moreover, the following results hold.

Theorem 6.3.4. *One has*

$$\mathcal{G}_n = \mathcal{C}_n \oplus \mathcal{B}_n, \tag{6.12}$$

where $\oplus$ is the orthogonal sum, and $\langle \cdot, \cdot \rangle$ denotes the Frobenius product, defined by

$$\langle G_1, G_2 \rangle = tr(G_1^T G_2), \qquad G_1, G_2 \in \mathcal{G}_n,$$

where $tr(G)$ is the trace of the matrix G.

Proof. Observe that, to prove (6.12), it is enough to demonstrate the following properties:

6.3.4.1) $\mathcal{C}_n \cap \mathcal{B}_n = \{O_n\}$, where $O_n \in \mathbb{R}^{n \times n}$ is the matrix whose entries are equal to 0;

6.3.4.2) for any $G \in \mathcal{G}_n$, there exist $C \in \mathcal{C}_n$ and $B \in \mathcal{B}_n$ with $G = C + B$;

6.3.4.3) for any $C \in \mathcal{C}_n$ and $B \in \mathcal{B}_n$, it is $C + B \in \mathcal{G}_n$;

6.3.4.4) $\langle C, B \rangle = 0$ for each $C \in \mathcal{C}_n$ and $B \in \mathcal{B}_n$.

6.3.4.1) Let $G \in \mathcal{C}_n \cap \mathcal{B}_n$. Then, $G = Q_n \Lambda^{(G)} Q_n^T$, where $\Lambda^{(G)} = \text{diag}(\boldsymbol{\lambda}^{(G)})$ and $\boldsymbol{\lambda}^{(G)}$ is both symmetric and asymmetric. But this is possible if and only if $\boldsymbol{\lambda}^{(G)} = \mathbf{0}$, where $\mathbf{0}$ is the vector whose components are equal to 0. Thus, $\Lambda^{(G)} = O_n$ and hence $G = O_n$. This proves 6.3.4.1).

6.3.4.2) Let $G \in \mathcal{G}_n$, $\Lambda^{(G)} \in \mathbb{R}^{n \times n}$ be such that $G = Q_n \Lambda^{(G)} Q_n^T$, $\Lambda^{(G)} = \text{diag}(\boldsymbol{\lambda}^{(G)}) = \text{diag}(\lambda_0^{(G)}\ \lambda_1^{(G)} \cdots \lambda_{n-1}^{(G)})$. For $j \in \{0, 1, \ldots, n-1\}$, set

$$\lambda_j^{(C)} = \frac{\lambda_j^{(G)} + \lambda_{(n-j) \bmod n}^{(G)}}{2}, \qquad \lambda_j^{(B)} = \frac{\lambda_j^{(G)} - \lambda_{(n-j) \bmod n}^{(G)}}{2}.$$

For $r \in \{C, B\}$, set $\Lambda^{(r)} = \text{diag}(\lambda_0^{(r)}\ \lambda_1^{(r)} \cdots \lambda_{n-1}^{(r)})$, $C = Q_n \Lambda^{(C)} Q_n^T$ and $B = Q_n \Lambda^{(B)} Q_n^T$. Observe that $\boldsymbol{\lambda}^{(G)} = \boldsymbol{\lambda}^{(C)} + \boldsymbol{\lambda}^{(B)}$, where $\boldsymbol{\lambda}^{(C)}$ is symmetric and $\boldsymbol{\lambda}^{(B)}$ is asymmetric. Hence, $C \in \mathcal{C}_n$ and $B \in \mathcal{B}_n$. This proves 6.3.4.2).

6.3.4.3) Let $C \in \mathcal{C}_n$ and $B \in \mathcal{B}_n$. For $r \in \{C,B\}$, set $\Lambda^{(r)} = \mathrm{diag}(\lambda_0^{(r)}\ \lambda_1^{(r)} \cdots \lambda_{n-1}^{(r)})$, $C = Q_n\Lambda^{(C)}Q_n^T$ and $B = Q_n\Lambda^{(B)}Q_n^T$. Note that $\boldsymbol{\lambda}^{(C)}$ is symmetric and $\boldsymbol{\lambda}^{(B)}$ is asymmetric. We have $C+B = Q_n(\Lambda^{(C)}+\Lambda^{(B)})Q_n^T \in \mathcal{G}_n$.

6.3.4.4) Choose arbitrarily $C \in \mathcal{C}_n$ and $B \in \mathcal{B}_n$. For $r \in \{C,B\}$, put $\Lambda^{(r)} = \mathrm{diag}(\lambda_0^{(r)}\ \lambda_1^{(r)} \cdots \lambda_{n-1}^{(r)})$, $C = Q_n\Lambda^{(C)}Q_n^T$ and $B = Q_n\Lambda^{(B)}Q_n^T$. Observe that $\boldsymbol{\lambda}^{(C)}$ is symmetric and $\boldsymbol{\lambda}^{(B)}$ is asymmetric. In particular, $\lambda_0^{(B)} = 0$ and $\lambda_{n/2}^{(B)} = 0$ when n is even. Note that $C^TB = Q_n\Lambda^{(C)}\Lambda^{(B)}Q_n^T$, where $\Lambda^{(C)}\Lambda^{(B)}$ =diag $(\lambda_0^{(C)}\lambda_0^{(B)} \quad \lambda_1^{(C)}\lambda_1^{(B)} \ \cdots\ \lambda_{n-1}^{(C)}\lambda_{n-1}^{(B)})$. Thus, we obtain

$$\begin{aligned}\langle C,B\rangle &= \mathrm{tr}(C^TB) = \sum_{j=0}^{n-1}\lambda_j^{(C)}\lambda_j^{(B)} = \sum_{j=1}^{\lfloor (n-1)/2\rfloor}(\lambda_j^{(C)}\lambda_j^{(B)} + \lambda_{n-j}^{(C)}\lambda_{n-j}^{(B)}) = \\ &= \sum_{j=1}^{\lfloor (n-1)/2\rfloor}(\lambda_j^{(C)}\lambda_j^{(B)} - \lambda_j^{(C)}\lambda_j^{(B)}) = 0,\end{aligned}$$

that is 6.3.4.4). This ends the proof. □

Theorem 6.3.5. *It is*

$$\mathcal{C}_n = \mathcal{D}_n \oplus \mathcal{E}_n, \tag{6.13}$$

where $\oplus$ is the orthogonal sum with respect to the Frobenius product.

Proof. Analogously as in Theorem 6.3.4, to get (6.13) it is sufficient to prove the following properties:

6.3.5.1) $\mathcal{D}_n \cap \mathcal{E}_n = \{O_n\}$, where $O_n \in \mathbb{R}^{n\times n}$ is the matrix whose entries are equal to 0;

6.3.5.2) for any $C \in \mathcal{C}_n$, there exist $C_1 \in \mathcal{D}_n$ and $C_2 \in \mathcal{E}_n$ with $C = C_1 + C_2$;

6.3.5.3) for every $C_1 \in \mathcal{D}_n$ and $C_2 \in \mathcal{E}_n$, we get that $C_1 + C_2 \in \mathcal{C}_n$.

6.3.5.4) $\langle C_1, C_2\rangle = 0$ for each $C_1 \in \mathcal{D}_n$ and $C_2 \in \mathcal{E}_n$.

6.3.5.1) Let $C \in \mathcal{D}_n \cap \mathcal{E}_n$. Then, $C = Q_n\Lambda Q_n^T$, where $\Lambda = \mathrm{diag}(\boldsymbol{\lambda})$, $\boldsymbol{\lambda}$ is symmetric and such that $\lambda_0 = 0$ and $\lambda_{n/2} = 0$ when n is even, because $C \in \mathcal{D}_n$. Moreover, since $C \in \mathcal{E}_n$, we get that $\lambda_j = 0$ for $j = 1,\ldots,n-1$, $j \neq n/2$ when n is even, that is $\boldsymbol{\lambda} = \mathbf{0}$, Thus, $\Lambda = O_n$ and hence $C = O_n$. This proves 6.3.5.1).

6.3.5.2) Let $C \in \mathcal{C}_n$, $\Lambda \in \mathbb{R}^{n\times n}$ be such that $C = Q_n\Lambda Q_n^T$, Λ =diag$(\boldsymbol{\lambda})$ = diag$(\lambda_0\ \lambda_1 \cdots \lambda_{n-1})$.

For $j \in \{0, 1, \ldots, n-1\}$, set

$$\lambda_j^{(1)} = \begin{cases} \lambda_j & \text{if } j \neq 0 \text{ and } j \neq n/2, \\ \\ 0 & \text{otherwise,} \end{cases} \qquad \lambda_j^{(2)} = \begin{cases} \lambda_j & \text{if } j = 0 \text{ or } j = n/2, \\ \\ 0 & \text{otherwise.} \end{cases}$$

For $r = 1,2$, set $\Lambda_r = \text{diag}(\lambda_0^{(r)}\ \lambda_1^{(r)} \cdots \lambda_{n-1}^{(r)})$, and $C_r = Q_n \Lambda_r Q_n^T$. Note that $\boldsymbol{\lambda} = \boldsymbol{\lambda}^{(1)} + \boldsymbol{\lambda}^{(2)}$, where $\boldsymbol{\lambda}^{(1)}$ is symmetric. Hence, $C_1 \in \mathcal{D}_n$ and $C_2 \in \mathcal{E}_n$. This proves 6.3.5.2).

6.3.5.3) Let $C_1 \in \mathcal{D}_n$ and $C_2 \in \mathcal{E}_n$. For $r = 1,2$ there is $\Lambda_r = \text{diag}(\boldsymbol{\lambda}^{(r)}) = \text{diag}(\lambda_0^{(r)}\ \lambda_1^{(r)} \cdots \lambda_{n-1}^{(r)})$, such that $C_r = Q_n \Lambda_r Q_n^T$, $\boldsymbol{\lambda}^{(1)}$ is symmetric, $\lambda_0^{(1)} = 0$ and $\lambda_{n/2}^{(1)} = 0$ when n is even, $\lambda_j^{(2)} = 0$, $j = 1, \ldots, n-1$, $j \neq n/2$ when n is even. Thus, it is not difficult to check that $\boldsymbol{\lambda}^{(1)} + \boldsymbol{\lambda}^{(2)}$ is symmetric. So, $C_1 + C_2 = Q_n\, \text{diag}(\boldsymbol{\lambda}^{(1)} + \boldsymbol{\lambda}^{(2)}) Q_n^T \in \mathcal{C}_n$.

6.3.5.4) Pick $C_1 \in \mathcal{D}_n$ and $C_2 \in \mathcal{E}_n$. Then for $r = 1,2$ there exists $\Lambda_r = \text{diag}(\boldsymbol{\lambda}^{(r)}) = \text{diag}(\lambda_0^{(r)}\ \lambda_1^{(r)} \cdots \lambda_{n-1}^{(r)})$, such that $C_r = Q_n \Lambda_r Q_n^T$, $\boldsymbol{\lambda}^{(1)}$ is symmetric, $\lambda_0^{(1)} = 0$ and $\lambda_{n/2}^{(1)} = 0$ when n is even. Note that $C_1^T C_2 = Q_n \Lambda_1 \Lambda_2 Q_n^T$, where

$$\Lambda_1 \Lambda_2 = \text{diag}(\lambda_0^{(1)} \cdot \lambda_0^{(2)}\ \lambda_1^{(1)} \cdot \lambda_1^{(2)}\ \cdots\ \lambda_{n-1}^{(1)} \cdot \lambda_{n-1}^{(2)}) = \text{diag}(\mathbf{0}) = O_n.$$

Therefore, $C_1^T C_2 = O_n$, and thus we get $\langle C_1, C_2 \rangle = \text{tr}(C_1^T C_2) = 0$, that is 6.3.5.4). This ends the proof. □

Now we give a consequence of 6.3.4 and 6.3.5.

Corollary 6.3.5.1. *The following result holds:*

$$\mathcal{G}_n = \mathcal{B}_n \oplus \mathcal{D}_n \oplus \mathcal{E}_n.$$

We recall the definition of the classical Hartley matrix (see also [22] and the references therein). If n is odd, we have

$$H_n = \frac{1}{\sqrt{n}} \left(\mathbf{u}^{(0)}\ \mathbf{u}^{(1)} + \mathbf{v}^{(1)}\ \ldots\ \mathbf{u}^{\left(\frac{n-1}{2}\right)} + \mathbf{v}^{\left(\frac{n-1}{2}\right)}\ \ \mathbf{u}^{\left(\frac{n-1}{2}\right)} - \mathbf{v}^{\left(\frac{n-1}{2}\right)}\ \ldots\ \mathbf{u}^{(1)} - \mathbf{v}^{(1)} \right). \quad (6.14)$$

When n is even we get

$$H_n = \frac{1}{\sqrt{n}} \left(\mathbf{u}^{(0)}\ \mathbf{u}^{(1)} + \mathbf{v}^{(1)} \ldots \mathbf{u}^{\left(\frac{n}{2}-1\right)} + \mathbf{v}^{\left(\frac{n}{2}-1\right)}\ \mathbf{u}^{\left(\frac{n}{2}\right)}\ \mathbf{u}^{\left(\frac{n}{2}-1\right)} - \mathbf{v}^{\left(\frac{n}{2}-1\right)} \ldots \mathbf{u}^{(1)} - \mathbf{v}^{(1)} \right). \quad (6.15)$$

It is not difficult to see that

$$H_n = Q_n Y_n, \quad (6.16)$$

where

$$y_{k,j}^{(n)} = \begin{cases} 1 & \text{if } k = j = 0, \\ \dfrac{1}{\sqrt{2}} & \text{if } k = j \text{ and } 1 \le k \le \dfrac{n-1}{2}, \\ \dfrac{1}{\sqrt{2}} & \text{if } k + j = n \text{ and } 1 \le k \le n-1, \\ -\dfrac{1}{\sqrt{2}} & \text{if } k = j \text{ and } \dfrac{n+1}{2} \le k \le n-1, \\ 0 & \text{otherwise} \end{cases} \tag{6.17}$$

if n is odd, and

$$y_{k,j}^{(n)} = \begin{cases} 1 & \text{if } k = j = 0 \text{ or } k = j = \dfrac{n}{2}, \\ \dfrac{1}{\sqrt{2}} & \text{if } k = j \text{ and } 1 \le k \le \dfrac{n}{2} - 1, \\ \dfrac{1}{\sqrt{2}} & \text{if } k + j = n \text{ and } 1 \le k \le n-1, \\ -\dfrac{1}{\sqrt{2}} & \text{if } k = j \text{ and } \dfrac{n}{2} + 1 \le k \le n-1, \\ 0 & \text{otherwise} \end{cases} \tag{6.18}$$

if n is even. Now, set

$$\mathcal{H}_n = \mathrm{sd}(H_n) = \{H_n \Lambda H_n^T : \Lambda = \mathrm{diag}(\boldsymbol{\lambda}), \boldsymbol{\lambda} \in \mathbb{R}^n\}. \tag{6.19}$$

It is not difficult to see that

$$\begin{aligned} \mathcal{C}_n &= \{Q_n \Lambda Q_n^T : \Lambda = \mathrm{diag}(\boldsymbol{\lambda}), \boldsymbol{\lambda} \in \mathbb{R}^n, \boldsymbol{\lambda} \text{ is symmetric}\} = \\ &= \{H_n \Lambda H_n^T : \Lambda = \mathrm{diag}(\boldsymbol{\lambda}), \boldsymbol{\lambda} \in \mathbb{R}^n, \boldsymbol{\lambda} \text{ is symmetric}\}. \end{aligned} \tag{6.20}$$

From (6.19) and (6.20) it follows that

$$\mathcal{H}_n = \mathcal{C}_n \oplus \mathcal{F}_n, \tag{6.21}$$

where

$$\mathcal{F}_n = \{H_n \Lambda H_n^T : \Lambda = \mathrm{diag}(\boldsymbol{\lambda}), \boldsymbol{\lambda} \in \mathbb{R}^n, \boldsymbol{\lambda} \text{ is asymmetric}\}.$$

If i is the imaginary unit and $\omega_n = e^{\frac{2\pi \mathrm{i}}{n}}$, then the n-th roots of 1 are

$$\omega_n^j = e^{\frac{2\pi j \mathrm{i}}{n}} = \cos\Big(\frac{2\pi j}{n}\Big) + \mathrm{i}\sin\Big(\frac{2\pi j}{n}\Big), \qquad j = 0, 1, \ldots, n-1.$$

The *Fourier matrix* of dimension $n \times n$ is defined by $F_n = (f_{k,l}^{(n)})_{k,l}$, where

$$f_{k,l}^{(n)} = \frac{1}{\sqrt{n}}\, \omega_n^{kl}, \qquad k, l = 0, 1, \ldots, n-1.$$

Note that F_n is symmetric, and $F_n^{-1} = F_n^*$ (see, e.g., [54]).

Let $\mathcal{W}_n$ be the space of all real matrices *simultaneously diagonalizable* by F_n, that is

$$\mathcal{W}_n = \mathrm{sd}(F_n) = \{F_n \Lambda F_n^* \in \mathbb{R}^{n\times n} : \Lambda = \mathrm{diag}(\boldsymbol{\lambda}), \boldsymbol{\lambda} \in \mathbb{C}^n\}.$$

It is not difficult to see that $\mathcal{W}_n$ is a commutative matrix algebra. Moreover, we define the following class:

$$\mathcal{A}_n = \{F_n \Lambda F_n^* : \Lambda = \mathrm{diag}(\boldsymbol{\lambda}), \boldsymbol{\lambda} \in (\mathrm{i}\mathbb{R})^n, \boldsymbol{\lambda} \text{ is asymmetric}\}. \tag{6.22}$$

So we define the *β-matrices* as the matrices belonging to the following set:

$$\mathcal{V}_n = \mathcal{C}_n \oplus \mathcal{B}_n \oplus \mathcal{F}_n \oplus \mathcal{A}_n. \tag{6.23}$$

6.4 Structural characterizations of γ-matrices

In this section we show that $\mathcal{V}_n$ coincides with the direct sum of the sets of all real circulant matrices and of all reverse circulant matrices.

We consider the set of families

$$\begin{aligned}
\mathcal{L}_{n,k} &= \{A \in \mathbb{R}^{n\times n} : \text{ there is } \mathbf{a} = (a_0\, a_1 \ldots a_{n-1})^T \in \mathbb{R}^n \text{ with } a_{l,j} = a_{(j+kl) \bmod n}\},\\
\mathcal{K}_{n,k} &= \{A \in \mathbb{R}^{n\times n} : \text{ there is a symmetric } \mathbf{a} = (a_0\, a_1 \ldots a_{n-1})^T \in \mathbb{R}^n\\
&\qquad \text{with } a_{l,j} = a_{(j+kl) \bmod n}\},\\
\mathcal{J}_{n,k} &= \Big\{A \in \mathbb{R}^{n\times n} : \text{ there is a symmetric } \mathbf{a} = (a_0\, a_1 \ldots a_{n-1})^T \in \mathbb{R}^n \text{ with}\\
&\qquad \sum_{t=0}^{n-1} a_t = 0, \sum_{t=0}^{n-1} (-1)^t a_t = 0 \text{ when } n \text{ is even, and } a_{l,j} = a_{(j+kl) \bmod n}\Big\},
\end{aligned}$$

where $k \in \{1,2,\ldots,n-1\}$.

When $k = n-1$, $\mathcal{L}_{n,n-1}$ is the class of all *real circulant matrices*, that is the family of those matrices $C \in \mathbb{R}^{n\times n}$ such that every row, after the first, has the elements of the previous one shifted cyclically one place right (see, e.g., [54]).

Given a vector $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{c} = (c_0\, c_1 \cdots c_{n-1})^T$, let us define

$$\mathrm{circ}(\mathbf{c}) = C = \begin{pmatrix} c_0 & c_1 & c_2 & \ldots & c_{n-2} & c_{n-1} \\ c_{n-1} & c_0 & c_1 & \ldots & c_{n-3} & c_{n-2} \\ c_{n-2} & c_{n-1} & c_0 & \ddots & c_{n-4} & c_{n-3} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ c_2 & c_3 & c_4 & \ddots & c_0 & c_1 \\ c_1 & c_2 & c_3 & \ldots & c_{n-1} & c_0 \end{pmatrix},$$

where $C \in \mathcal{L}_{n,n-1}$.

Theorem 6.4.1. ([54, Theorems 3.2.2 and 3.2.3]) *The following result holds:*

$$\mathcal{W}_n = \mathcal{L}_{n,n-1}.$$

As a consequence of this theorem, we get that the n eigenvectors of every circulant matrix $C \in \mathbb{R}^{n\times n}$ are given by

$$\mathbf{w}^{(j)} = (1\ \omega_n^j\ \omega_n^{2j}\ \cdots\ \omega_n^{(n-1)j})^T,$$

and the eigenvalues of a matrix $C = \mathrm{circ}(\mathbf{c}) \in \mathcal{F}_n$ are expressed by

$$\lambda_j = \mathbf{c}^T \mathbf{w}^{(j)} = \sum_{k=0}^{n-1} c_k \omega_n^{jk}, \qquad j = 0,1,\ldots,n-1.$$

Now we present some results about symmetric circulant real matrices. Observe that, if $C = \mathrm{circ}(\mathbf{c})$, with $\mathbf{c} \in \mathbb{R}^n$, then C is symmetric if and only if $\mathbf{c}$ is symmetric. Thus, the class of all real symmetric circulant matrices coincides with $\mathcal{K}_{n,n-1}$ and has dimension $\lfloor \frac{n}{2} \rfloor + 1$ over $\mathbb{R}$.

Theorem 6.4.2. (see, e.g., [51, §4], [111, Lemma 3]) *Let $C \in \mathcal{K}_{n,n-1}$. Then, the set of all eigenvectors of C can be expressed as $\{\mathbf{q}^{(0)}, \mathbf{q}^{(1)}, \ldots, \mathbf{q}^{(n-1)}\}$, where $\mathbf{q}^{(j)}$, $j = 0, 1, \ldots, n-1$, is as in* (6.8), (6.9) *and* (6.10).

Note that from Theorem 6.4.2 it follows that the set of all real symmetric circulant matrices is contained in $\mathcal{G}_n$. The nest result holds.

Theorem 6.4.3. (see, e.g., [36, §1.2], [51, §4], [151, Theorem 1]) *Let $C = circ(\mathbf{c}) \in \mathcal{K}_{n,n-1}$. Then, the eigenvalues λ_j of C, $j = 0, 1, \ldots, \lfloor \frac{n}{2} \rfloor$, are given by*

$$\lambda_j = \mathbf{c}^T \mathbf{u}^{(j)}. \tag{6.24}$$

Moreover, for $j = 1, 2, \ldots, \lfloor \frac{n-1}{2} \rfloor$ it is

$$\lambda_j = \lambda_{n-j}.$$

From Theorem 6.4.3 it follows that, if C is a real symmetric circulant matrix and $\boldsymbol{\lambda}^{(C)}$ is the set of its eigenvalues, then $\boldsymbol{\lambda}^{(C)}$ is symmetric, thanks to (6.24). Hence,

$$\mathcal{K}_{n,n-1} \subset \mathcal{C}_n. \tag{6.25}$$

Now we prove that $\mathcal{C}_n$ is contained in the class of all real symmetric circulant matrices $\mathcal{K}_{n,n-1}$. First, we give the following

Theorem 6.4.4. *Every matrix $C \in \mathcal{C}_n$ is circulant, that is*

$$\mathcal{C}_n \subset \mathcal{L}_{n,n-1}. \tag{6.26}$$

Proof. Let $C \in \mathcal{C}_n$, $C = (c_{k,l})_{k.l}$ and $\Lambda^{(C)} = \mathrm{diag}(\lambda_0^{(C)}\ \lambda_1^{(C)} \cdots \lambda_{n-1}^{(C)})$ be such that $\lambda_j^{(C)} = \lambda_{(n-j) \bmod n}^{(C)}$ for every $j \in \{0, 1, \ldots, n-1\}$, and $C = Q_n \Lambda^{(C)} Q_n^T$. We have

$$c_{k,l} = \sum_{j=0}^{n-1} q_{k,j}^{(n)}\, \lambda_j^{(C)}\, q_{l,j}^{(n)}.$$

From this we get, if n is even,

$$c_{k,l} = \lambda_0^{(C)} q_{k,0}^{(n)} q_{l,0}^{(n)} + \lambda_{n/2}^{(C)} q_{k,n/2}^{(n)} q_{l,n/2}^{(n)} + \sum_{j=1}^{n/2-1} \lambda_j^{(C)} (q_{k,j}^{(n)} q_{l,j}^{(n)} + q_{k,n-j}^{(n)} q_{l,n-j}^{(n)}), \tag{6.27}$$

and, if n is odd,

$$c_{k,l} = \lambda_0^{(C)}\, q_{k,0}^{(n)} q_{l,0}^{(n)} + \sum_{j=1}^{(n-1)/2} \lambda_j^{(C)} (q_{k,j}^{(n)} q_{l,j}^{(n)} + q_{k,n-j}^{(n)} q_{l,n-j}^{(n)}). \tag{6.28}$$

When n is even, from (6.5) and (6.27) we deduce

$$\begin{aligned} c_{k,l} &= \frac{\lambda_0^{(C)}}{n} + (-1)^{k-l} \frac{\lambda_{n/2}^{(C)}}{n} + \\ &+ \frac{2}{n} \sum_{j=1}^{n/2-1} \lambda_j^{(C)} \cdot \left(\cos\left(\frac{2\pi k j}{n}\right) \cdot \cos\left(\frac{2\pi l j}{n}\right) + \sin\left(\frac{2\pi k j}{n}\right) \cdot \sin\left(\frac{2\pi l j}{n}\right) \right) = \\ &= \frac{\lambda_0^{(C)}}{n} + (-1)^{k-l} \frac{\lambda_{n/2}^{(C)}}{n} + \frac{2}{n} \sum_{j=1}^{n/2-1} \lambda_j^{(C)} \cdot \cos\left(\frac{2\pi (k-l) j}{n}\right). \end{aligned}$$

Let $\mathbf{c} = (c_0\, c_1 \cdots c_{n-1})^T$, where

$$c_t = \frac{\lambda_0^{(C)}}{n} + (-1)^t \frac{\lambda_{n/2}^{(C)}}{n} + \frac{2}{n} \sum_{j=1}^{n/2-1} \lambda_j^{(C)} \cdot \cos\left(\frac{2\pi t j}{n}\right), \quad t \in \{0,1,\ldots,n-1\}.$$

Then we get $C = \text{circ}(\mathbf{c})$, since for any k, $l \in \{0, 1, \ldots, n-1\}$ it is $c_{k,l} = c_{(k-l) \bmod n}$.

When n is odd, from (6.5) and (6.28) we obtain

$$\begin{aligned} c_{k,l} &= \frac{\lambda_0^{(C)}}{n} + \frac{2}{n} \sum_{j=1}^{(n-1)/2} \lambda_j^{(C)} \cdot \left(\cos\left(\frac{2\pi k j}{n}\right) \cdot \cos\left(\frac{2\pi l j}{n}\right) + \right. \\ &+ \left. \sin\left(\frac{2\pi k j}{n}\right) \cdot \sin\left(\frac{2\pi l j}{n}\right) \right) = \\ &= \frac{\lambda_0^{(C)}}{n} + \frac{2}{n} \sum_{j=1}^{(n-1)/2} \lambda_j^{(C)} \cdot \cos\left(\frac{2\pi (k-l) j}{n}\right). \end{aligned}$$

Let $\mathbf{c} = (c_0\, c_1 \cdots c_{n-1})^T$, where

$$c_t = \frac{\lambda_0^{(C)}}{n} + \frac{2}{n} \sum_{j=1}^{(n-1)/2} \lambda_j^{(C)} \cdot \cos\left(\frac{2\pi t j}{n}\right), \quad t \in \{0,1,\ldots,n-1\}.$$

Hence, $C = \text{circ}(\mathbf{c})$, because for each k, $l \in \{0, 1, \ldots, n-1\}$ it is $c_{k,l} = c_{(k-l) \bmod n}$. Therefore, $\mathcal{C}_n \subset \mathcal{L}_{n,n-1}$. □

A consequence of Theorem 6.4.4 is the following

Corollary 6.4.4.1. *The class $\mathcal{C}_n$ is the set of all real symmetric circulant matrices, that is*

$$\mathcal{C}_n = \mathcal{K}_{n,n-1}. \tag{6.29}$$

Proof. Since every matrix belonging to $\mathcal{C}_n$ is symmetric, we get that (6.29) is a consequence of (6.25) and (6.26). □

If $k = 1$, then $\mathcal{L}_{n,1}$ is the set of all *real reverse circulant* (or *real anti-circulant*) *matrices*, that is the class of all matrices $B \in \mathbb{R}^{n\times n}$ such that every row, after the first, has the elements of the previous one shifted cyclically one place left (see, e.g., [54]). Given a vector $\mathbf{b} = (b_0\, b_1 \cdots b_{n-1})^T \in \mathbb{R}^n$, set

$$\text{rcirc}(\mathbf{b}) = B = \begin{pmatrix} b_0 & b_1 & b_2 & \ldots & b_{n-2} & b_{n-1} \\ b_1 & b_2 & b_3 & \ldots & b_{n-1} & b_0 \\ b_2 & b_3 & b_4 & \ldots & b_0 & b_1 \\ \vdots & \vdots & \vdots & \ldots & \vdots & \vdots \\ b_{n-2} & b_{n-1} & b_0 & \ldots & b_{n-4} & b_{n-3} \\ b_{n-1} & b_0 & b_1 & \ldots & b_{n-3} & b_{n-2} \end{pmatrix},$$

with $B \in \mathcal{L}_{n,1}$.

Observe that every matrix $B \in \mathcal{B}_{n,1}$ is symmetric, and the set $\mathcal{L}_{n,1}$ is a linear space over $\mathbb{R}$, but not an algebra. Note that, if $B_1, B_2 \in \mathcal{L}_{n,1}$, then $B_1 B_2, B_2 B_1 \in \mathcal{L}_{n,n-1}$ (see [54, Theorem 5.1.2]).

Now we give the next results.

Theorem 6.4.5. *The following inclusion holds:*

$$\mathcal{B}_n \subset \mathcal{L}_{n,1}.$$

Proof. Let $B \in \mathcal{B}_n$, $B = (b_{k,l})_{k.l}$ and $\Lambda^{(B)} = \mathrm{diag}(\lambda_0^{(B)}\ \lambda_1^{(B)} \cdots \lambda_{n-1}^{(B)})$ be such that $\lambda_j^{(B)} = -\lambda_{(n-j) \bmod n}^{(B)}$ for every $j \in \{0, 1, \ldots, n-1\}$, and $B = Q_n \Lambda^{(B)} Q_n^T$. We have

$$b_{k,l} = \sum_{j=0}^{n-1} q_{k,j}^{(n)} \lambda_j^{(B)} q_{l,j}^{(n)}. \tag{6.30}$$

Observe that $\lambda_0^{(B)} = 0$ and $\lambda_{n/2}^{(B)} = 0$, if n is even. From this and (6.30) we get

$$b_{k,l} = \sum_{j=1}^{\lfloor (n-1)/2 \rfloor} \lambda_j^{(B)} \cdot \left(q_{k,j}^{(n)} q_{l,j}^{(n)} - q_{k,n-j}^{(n)} q_{l,n-j}^{(n)} \right), \tag{6.31}$$

both when n is even and when n is odd. From (6.5) and (6.31) we deduce

$$\begin{aligned} b_{k,l} &= \frac{2}{n} \sum_{j=1}^{\lfloor (n-1)/2 \rfloor} \lambda_j^{(B)} \left(\cos\left(\frac{2\pi k j}{n}\right) \cos\left(\frac{2\pi l j}{n}\right) - \sin\left(\frac{2\pi k j}{n}\right) \sin\left(\frac{2\pi l j}{n}\right) \right) = \\ &= \frac{2}{n} \sum_{j=1}^{\lfloor (n-1)/2 \rfloor} \lambda_j^{(B)} \cos\left(\frac{2\pi (k+l) j}{n}\right). \end{aligned}$$

Let $\mathbf{b} = (b_0\, b_1 \cdots b_{n-1})^T$, where

$$b_t = \frac{2}{n} \sum_{j=1}^{\lfloor (n-1)/2 \rfloor} \lambda_j^{(B)} \cdot \cos\left(\frac{2\pi t j}{n}\right), \quad t \in \{0, 1, \ldots, n-1\}. \tag{6.32}$$

Thus, $B = \mathrm{circ}(\mathbf{b})$, because for each $k, l \in \{0, 1, \ldots, n-1\}$ we have $b_{k,l} = b_{(k-l) \bmod n}$. For any k, $l \in \{0, 1, \ldots, n-1\}$ it is $b_{k,l} = b_{(k+l) \bmod n}$. Hence, $\mathcal{B}_n \subset \mathcal{L}_{n,1}$. □

Theorem 6.4.6. *One has*

$$\mathcal{B}_n \subset \mathcal{K}_{n,1}.$$

Proof. We recall that

$$\begin{aligned} \mathcal{K}_{n,1} = \Big\{ & B \in \mathbb{R}^{n \times n} : \text{ there is a symmetric } \mathbf{b} = (b_0\, b_1 \ldots b_{n-1})^T \in \mathbb{R}^n \\ & \text{with } b_{k,j} = b_{(j+k) \bmod n} \Big\}. \end{aligned}$$

By Theorem 6.4.5, we get $\mathcal{B}_n \subset \mathcal{L}_{n,1}$. Now we prove the symmetry of $\mathbf{b}$.

Let $B \in \mathcal{B}_n$ be such that there exists $\Lambda^{(B)} \in \mathbb{R}^{n\times n}$, $\Lambda^{(B)} = \mathrm{diag}(\lambda_0^{(B)}\ \lambda_1^{(B)} \cdots \lambda_{n-1}^{(B)})$, such that $C = Q_n \Lambda^{(B)} Q_n^T$ and $\lambda_j^{(B)} = -\lambda_{(n-j) \bmod n}^{(B)}$ for all $j \in \{0,\ 1, \ldots, n-1\}$. By Theorem 6.4.5, $b_{k,j} = b_{(j+k) \bmod n}$. Moreover, by arguing as in Theorem 6.4.5, we get (6.32), and hence

$$\begin{aligned} b_t &= \frac{2}{n} \sum_{j=1}^{\lfloor (n-1)/2 \rfloor} \lambda_j^{(B)} \cdot \cos\left(\frac{2\pi t j}{n}\right) = \frac{2}{n} \sum_{j=1}^{\lfloor (n-1)/2 \rfloor} \lambda_j^{(B)} \cdot \cos\left(2\pi j - \frac{2\pi t j}{n}\right) = \\ &= \frac{2}{n} \sum_{j=1}^{\lfloor (n-1)/2 \rfloor} \lambda_j^{(B)} \cdot \cos\left(\frac{2\pi (n-t) j}{n}\right) = b_{n-t} \end{aligned}$$

for any $t \in \{0,\ 1, \ldots, n-1\}$. Thus, $\mathbf{b}$ is symmetric. □

Theorem 6.4.7. *Let $B = \mathrm{rcirc}(\mathbf{b}) \in \mathcal{B}_n$. Then, the eigenvalues $\lambda_j^{(B)}$ of B, $j = 0, 1, \ldots, \lfloor \frac{n}{2} \rfloor$, can be expressed as*

$$\lambda_j^{(B)} = \mathbf{b}^T \mathbf{u}^{(j)}. \tag{6.33}$$

Moreover, for $j = 1,\ 2, \ldots \lfloor \frac{n-1}{2} \rfloor$, we get

$$\lambda_{n-j}^{(B)} = -\lambda_j^{(B)}.$$

Furthermore, it is $\lambda_0^{(B)} = 0$, and $\lambda_{n/2}^{(B)} = 0$ if n is even.

Proof. Since $\mathbf{u}^{(j)}$, $j = 0, 1, \ldots \lfloor n/2 \rfloor$, is an eigenvector of B, then

$$B\mathbf{u}^{(j)} = \lambda_j^{(B)} \mathbf{u}^{(j)},$$

that is every component of the vector $B\mathbf{u}^{(j)}$ is equal to the respective component of the vector $\lambda_j^{(B)} \mathbf{u}^{(j)}$. In particular, if we consider the first component, we obtain (6.33). The last part of the assertion is a consequence of the asymmetry of the vector $\boldsymbol{\lambda}^{(B)}$. □

For the general computation of the eigenvalues of reverse circulant matrices, see, e.g., [36, §1.3 and Theorem 1.4.1], [147, Lemma 4.1].

Now we give the following

Theorem 6.4.8. *The following result holds:*

$$\mathcal{B}_n = \mathcal{J}_{n,1}.$$

Proof. First of all, we recall that

$$\begin{aligned}\mathcal{J}_{n,1} \quad = \quad & \Big\{B \in \mathbb{R}^{n\times n} : \text{ there is a symmetric } \mathbf{b} = (b_0\, b_1 \ldots b_{n-1})^T \in \mathbb{R}^n \text{ with} \\ & \sum_{t=0}^{n-1} b_t = 0, \sum_{t=0}^{n-1} (-1)^t b_t \,=\, 0 \text{ when } n \text{ is even, and } b_{k,j} = b_{(j+k) \bmod n}\Big\}.\end{aligned}$$

We begin with proving that $\mathcal{B}_n \subset \mathcal{J}_{n,1}$.

Let $B \in \mathcal{B}_n$. In Theorem 6.4.8 we proved that $B \in \mathcal{K}_{n,1}$, that is $\mathbf{b}$ is symmetric and $b_{k,j} = b_{(j+k) \bmod n}$.

Now we prove that

$$\sum_{t=0}^{n-1} b_t = 0. \tag{6.34}$$

Since $B \in \mathcal{B}_n$, the vector

$$\mathbf{u}^{(0)} = \Big(1\, 1 \,\cdots\, 1\Big)^T$$

is an eigenvector for the eigenvalue $\lambda_0^{(B)} = 0$. Hence, the formula (6.34) is a consequence of (6.33).

Again by (6.33), we get

$$\sum_{t=0}^{n-1} (-1)^t b_t \,=\, 0,$$

since the vector

$$\mathbf{u}^{(n/2)} = \Big(1 -1 \;\; 1 -1 \cdots -1\Big)^T$$

is an eigenvector for the eigenvalue $\lambda_{n/2}^{(B)} = 0$ if n is even. Thus, $\mathcal{B}_n \subset \mathcal{J}_{n,1}$.

Now observe that $\mathcal{J}_{n,1}$ is a linear space of dimension $\lfloor (n-1)/2 \rfloor$. Thus, by Proposition 6.3.3, $\mathcal{B}_n$ and $\mathcal{J}_{n,1}$ have the same dimension. So, $\mathcal{B}_n = \mathcal{J}_{n,1}$. This ends the proof. □

Theorem 6.4.9. *The next result holds:*

$$\mathcal{D}_n = \mathcal{J}_{n,n-1}.$$

Proof. We recall that

$$\begin{aligned}\mathcal{J}_{n,n-1} \quad = \quad & \Big\{C \in \mathbb{R}^{n\times n} : \text{ there is a symmetric } \mathbf{c} = (c_0\, c_1 \ldots c_{n-1})^T \in \mathbb{R}^n \text{ with} \\ & \sum_{t=0}^{n-1} c_t = 0, \sum_{t=0}^{n-1} (-1)^t c_t \,=\, 0 \text{ when } n \text{ is even, and } c_{k,j} = c_{(j-k) \bmod n}\Big\}.\end{aligned}$$

We first prove that $\mathcal{D}_n \subset \mathcal{J}_{n,n-1}$. From Theorem 6.3.5 and (6.25) we deduce that $\mathcal{D}_n \subset \mathcal{C}_n = \mathcal{K}_{n,n-1}$. Therefore, if $C = \text{circ}(\mathbf{c}) \in \mathcal{D}_n$, then $\mathbf{c}$ is symmetric.

Now we prove that

$$\sum_{t=0}^{n-1} c_t = 0. \tag{6.35}$$

Since $C \in \mathcal{C}_n$, the vector

$$\mathbf{u}^{(0)} = \begin{pmatrix} 1 & 1 & \cdots & 1 \end{pmatrix}^T$$

is an eigenvector for the eigenvalue $\lambda_0 = 0$. Hence, the formula (6.35) is a consequence of (6.24).

Again by (6.24), we get

$$\sum_{t=0}^{n-1} (-1)^t c_t = 0,$$

since the vector

$$\mathbf{u}^{(n/2)} = \begin{pmatrix} 1 & -1 & 1 & -1 & \cdots & -1 \end{pmatrix}^T$$

is an eigenvector for the eigenvalue $\lambda_{n/2} = 0$ if n is even. Thus, $\mathcal{D}_n \subset \mathcal{J}_{n,n-1}$.

Now observe that $\mathcal{J}_{n,n-1}$ is a linear space of dimension $\lfloor (n-1)/2 \rfloor$. Thus, by Proposition 6.3.3, $\mathcal{D}_n$ and $\mathcal{J}_{n,n-1}$ have the same dimension. So, $\mathcal{D}_n = \mathcal{J}_{n,n-1}$. This completes the proof. □

Theorem 6.4.10. *The next result holds:*

$$\mathcal{E}_n = \mathcal{P}_n = \mathcal{L}_{n,n-1} \cap \mathcal{L}_{n,1},$$

where

$$\mathcal{P}_n = \begin{cases} \left\{ C \in \mathbb{R}^{n \times n:} \text{ there are } k_1, k_2 \text{ with } c_{i,j} = \begin{cases} k_1 & \text{if } i+j \text{ is even} \\ k_2 & \text{if } i+j \text{ is odd} \end{cases} \right\} & \text{if } n \text{ is even}, \\ \\ \{ C \in \mathbb{R}^{n \times n:} \text{ there is } k \text{ with } c_{i,j} = k \text{ for all } i,j = 0,1,\ldots,n-1 \} & \text{if } n \text{ is odd}. \end{cases}$$

Proof. We first claim that $\mathcal{E}_n = \mathcal{P}_n$.

We begin with the inclusion $\mathcal{E}_n \subset \mathcal{P}_n$. Let $C \in \mathcal{E}_n$, $C = (c_{k,l})_{k.l}$ and $\Lambda = \text{diag}(\lambda_0 \ \lambda_1 \cdots \lambda_{n-1})$ be such that $\lambda_0 = 0$ for every $j \in \{1,2,\ldots,n-1\}$ except $n/2$ when n is even, and $C = Q_n \Lambda Q_n^T$. We have

$$c_{k,l} = \sum_{j=0}^{n-1} q_{k,j}^{(n)} \lambda_j q_{l,j}^{(n)}.$$

From this we get

$$c_{k,l} = \begin{cases} \lambda_0 q_{k,0}^{(n)} q_{l,0}^{(n)} + \lambda_{n/2} q_{k,n/2}^{(n)} q_{l,n/2}^{(n)} & \text{if } n \text{ is even,} \\ \\ \lambda_0 q_{k,0}^{(n)} q_{l,0}^{(n)} & \text{if } n \text{ is odd.} \end{cases} \tag{6.36}$$

When n is even, from (6.5) and (6.36) we deduce

$$c_{k,l} = \frac{\lambda_0}{n} + (-1)^{k-l} \frac{\lambda_{n/2}}{n}.$$

Note that

$$c_{k,l} = \begin{cases} \frac{\lambda_0}{n} + \frac{\lambda_{n/2}}{n} & \text{if } k-l \text{ is even,} \\ \\ \frac{\lambda_0}{n} - \frac{\lambda_{n/2}}{n} & \text{if } k-l \text{ is odd,} \end{cases}$$

and thus $C \in \mathcal{P}_n$.

When n is odd, from (6.5) and (6.36) we obtain

$$c_{k,l} = \frac{\lambda_0}{n}$$

for $k, l \in \{0, 1, \ldots, n-1\}$. Hence, $C \in \mathcal{P}_n$.

Now we prove that $\mathcal{P}_n \subset \mathcal{E}_n$. First of all, note that $\mathcal{P}_n \subset \mathcal{C}_n$. Let $C \in \mathcal{P}_n$. If n is even, then $\text{rank}(C) \leq 2$, and hence C has at least $n-2$ eigenvalues equal to 0. By contradiction, suppose that at least one of the eigenvalues different from λ_0 and $\lambda_{n/2}$, say λ_j, is different from 0. Thus, $\lambda_0 = 0$ or $\lambda_{n/2} = 0$. If $\lambda_0 = 0$, then

$$\sum_{t=0}^{n-1} c_t = 0,$$

and hence $k_1 = -k_2$. Therefore, $\text{rank}(C) \leq 1$, and thus there is at most one non-zero eigenvalue. This implies that $\lambda_{n/2} = 0$, and hence

$$\sum_{t=0}^{n-1} (-1)^t c_t = 0. \tag{6.37}$$

From (6.37) it follows that $k_1 = k_2 = 0$. Thus we deduce that $C = O_n$, which obviously implies that $\lambda_j = 0$. This is absurd, and hence $\lambda_0 \neq 0$.

When $\lambda_{n/2} = 0$, then (6.37) holds, and hence $k_1 = k_2$. Thus, $\text{rank}(C) \leq 1$, which implies that $\lambda_0 = 0$, because we know that $\lambda_j \neq 0$. This yields a contradiction. Thus, $\mathcal{P}_n \subset \mathcal{E}_n$, at least when n is even.

Now we suppose that n is odd. If $C \in \mathcal{P}_n$, then $\text{rank}(C) \leq 1$. This implies that C has at most a non-zero eigenvalue. We claim that $\lambda_j = 0$ for all $j \in \{1,2,\ldots,n-1\}$. By contradiction, suppose that there exists $q \in \{1,2,\ldots,n-1\}$ such that $\lambda_q \neq 0$. Hence, $\lambda_0 = 0$, and thus

$$0 = \sum_{t=0}^{n-1} c_t = nk.$$

This implies that $C = O_n$. Hence, $\lambda_q = 0$, which is absurd. Therefore, $\mathcal{P}_n \subset \mathcal{E}_n$ even when n is odd.

Now we claim that $\mathcal{P}_n = \mathcal{L}_{n,n-1} \cap \mathcal{L}_{n,1}$.

Observe that, if $C \in \mathcal{P}_n$, then C is both circulant and reverse circulant, and hence $\mathcal{P}_n \subset \mathcal{L}_{n,n-1} \cap \mathcal{L}_{n,1}$. Now we claim that $\mathcal{L}_{n,n-1} \cap \mathcal{L}_{n,1} \subset \mathcal{P}_n$. Let $C \in \mathcal{L}_{n,n-1} \cap \mathcal{L}_{n,1}$. If $\mathbf{c} = (c_0\, c_1 \ldots c_{n-1})^T$ is the first row of C, and $C = \text{circ}(\mathbf{c}) = \text{rcirc}(\mathbf{c})$, then we get

$$c_{1,j} = c_{(j-1) \bmod n} = c_{(j+1) \bmod n}, \quad j \in \{0,1,\ldots,n-1\}.$$

If n is even, then $c_{2j} = c_0$ and $c_{2j+1} = c_1$ for $j \in \{0,1,\ldots,n/2-1\}$, while when n is odd we have $c_j = c_0$ for $j \in \{0,1,\ldots,n-1\}$, getting the claim. □

Theorem 6.4.11. *The following result holds:*

$$\mathcal{K}_{n,1} = \mathcal{B}_n \oplus \mathcal{E}_n.$$

Proof. By Theorem 6.4.9, we know that

$$\begin{aligned} \mathcal{B}_n = \mathcal{J}_{n,1} \quad = \quad & \Big\{ C \in \mathbb{R}^{n\times n} : \text{ there is a symmetric } \mathbf{c} = (c_0\, c_1 \ldots c_{n-1})^T \in \mathbb{R}^n \text{ with} \\ & \sum_{t=0}^{n-1} c_t = 0, \sum_{t=0}^{n-1} (-1)^t c_t = 0 \text{ if } n \text{ is even, and } c_{i,j} = c_{(j+i) \bmod n} \Big\}. \end{aligned}$$

Moreover, by Theorem 6.4.10 we have

$$\begin{aligned} \mathcal{E}_n \quad = \quad & \Big\{ C \in \mathbb{R}^{n\times n} : \text{ there is } \mathbf{c} = (c_0\, c_1 \ldots c_{n-1})^T \in \mathbb{R}^n \text{ such that: there is } k \in \mathbb{R} \text{ with} \\ & c_t = k,\ t = 0,1,\ldots,n-1 \text{ if } n \text{ is even, and } k_1, k_2 \in \mathbb{R} \text{ with } c_t = k_{((-1)^t+3)/2} \\ & \text{if } n \text{ is odd, } t = 0,1,\ldots,n-1,\ c_{i,j} = c_{(j+i) \bmod n} \Big\}. \end{aligned}$$

First, we show that

$$\mathcal{K}_{n,1} \supset \mathcal{B}_n \oplus \mathcal{E}_n. \tag{6.38}$$

Let $C = \text{rcirc}(\mathbf{c}) \in \mathcal{B}_n \oplus \mathcal{E}_n$. There are $C = \text{rcirc}(\mathbf{c}^{(r)})$, with $C^{(1)} \in \mathcal{B}_n$, $C^{(2)} \in \mathcal{E}_n$. $r = 1,2$, $C = C^{(1)} + C^{(2)}$, $\mathbf{c} = \mathbf{c}^{(1)} + \mathbf{c}^{(2)}$. Note that $\mathbf{c}$ is symmetric, because both $\mathbf{c}^{(1)}$ and $\mathbf{c}^{(2)}$ are. Thus, (6.38) is proved.

We now prove the converse inclusion. Let $C = \text{rcirc}(\mathbf{c}) \in \mathcal{K}_{n,1}$. Then, $\mathbf{c}$ is symmetric. Suppose that n is odd and

$$\sum_{t=0}^{n-1} c_t = \tau.$$

Let $\mathbf{c}^{(2)} = (\tau/n\ \tau/n \ldots \tau/n)^T$, $\mathbf{c}^{(1)} = \mathbf{c} - \mathbf{c}^{(2)}$, and $C = \text{rcirc}(\mathbf{c}^{(r)})$, $r = 1,2$. Then, $C = C^{(1)} + C^{(2)}$. Note that $C^{(2)} \in \mathcal{E}_n$. Moreover, $\mathbf{c}^{(1)}$ is symmetric, and

$$\sum_{t=0}^{n-1} c_t^{(1)} = \sum_{t=0}^{n-1} (c_t - \tau/n) = 0.$$

Therefore, $C^{(1)} \in \mathcal{B}_n$.

Now assume that n is even. We get

$$\sum_{t=0}^{n-1} c_t = \tau = \frac{n}{2}(k_1 + k_2) \tag{6.39}$$

and

$$\sum_{t=0}^{n-1} (-1)^t c_t = \gamma_* = \frac{n}{2}(k_1 - k_2). \tag{6.40}$$

Then,

$$k_1 = (\tau + \gamma_*)/n, \qquad k_2 = (\tau - \gamma_*)/n. \tag{6.41}$$

Let $\mathbf{c}^{(2)} = (k_1\, k_2 \ldots k_1\, k_2)^T$, $\mathbf{c}^{(1)} = \mathbf{c} - \mathbf{c}^{(2)}$, and $C = \text{rcirc}(\mathbf{c}^{(r)})$, $r = 1,2$. Then, $C = C^{(1)} + C^{(2)}$. Note that $C^{(2)} \in \mathcal{E}_n$. Moreover, $\mathbf{c}^{(1)}$ is symmetric, and from (6.39), (6.41) we obtain

$$\sum_{t=0}^{n-1} c_t^{(1)} = \sum_{t=0}^{n-1} c_t - \frac{n}{2}(k_1 + k_2) = \tau - \tau = 0.$$

Moreover, from (6.40), (6.41) we deduce

$$\sum_{t=0}^{n-1} (-1)^t c_t^{(1)} = \sum_{t=0}^{n-1} (-1)^t c_t - \frac{n}{2}(k_1 - k_2) = \gamma_* - \gamma_* = 0.$$

Thus, $C^{(1)} \in \mathcal{B}_n$.

Moreover observe that, by Theorem 6.3.5, $\mathcal{E}_n \subset \mathcal{C}_n$, and thanks to Theorem 6.3.4, $\mathcal{C}_n$ and $\mathcal{B}_n$ are orthogonal. This implies that $\mathcal{E}_n$ and $\mathcal{B}_n$ are orthogonal. This ends the proof. □

We note that

$$\mathcal{F}_n = \{A \in \mathcal{L}_{n,1} : \text{ there is an asymmetric } \mathbf{a} \in \mathbb{R}^n \text{ with } A = \text{rcirc}(\mathbf{a})\} \tag{6.42}$$

(see also [22]). Now we prove the following

Proposition 6.4.12. *It is*

$$\mathcal{A}_n = \{A \in \mathcal{L}_{n,n-1} : \textit{ there is an asymmetric } \mathbf{a} \in \mathbb{R}^n \textit{ with } A = \text{circ}(\mathbf{a})\}. \tag{6.43}$$

Proof. We begin with the inclusion $\supset$. Let $A \in \mathcal{A}_n$, $A = \text{circ}(\mathbf{a})$, with $\mathbf{a}$ asymmetric. Since $A \in \mathcal{L}_{n,n-1}$, its eigenvectors are given by

$$\mathbf{w}^{(j)} = (1 \ \omega_n^j \ \omega_n^{2j} \ \cdots \ \omega_n^{(n-1)j})^T,$$

and the eigenvalues of A are expressed by

$$\lambda_j = \mathbf{a}^T \mathbf{w}^{(j)}, \qquad j = 0, 1, \ldots, n-1.$$

Note that

$$\mathbf{w}^{(j)} = \mathbf{u}^{(j)} + \mathrm{i}\,\mathbf{v}^{(j)}, \tag{6.44}$$

if $j = 0, 1, \ldots, \lceil \frac{n-1}{2} \rceil$, and

$$\mathbf{w}^{(n-j)} = \mathbf{u}^{(j)} - \mathrm{i}\,\mathbf{v}^{(j)}, \tag{6.45}$$

if $j = 0, 1, \ldots \lfloor \frac{n-1}{2} \rfloor$. From (6.44) and (6.45) it follows that

$$\lambda_j = \mathbf{a}^T(\mathbf{u}^{(j)} + \mathrm{i}\,\mathbf{v}^{(j)}) = \mathrm{i}\,\mathbf{a}^T \mathbf{v}^{(j)} \in \mathrm{i}\,\mathbb{R}$$

for $j = 0, 1, \ldots, \lceil \frac{n-1}{2} \rceil$, and

$$\lambda_{n-j} = \mathbf{a}^T(\mathbf{u}^{(j)} - \mathrm{i}\,\mathbf{v}^{(j)}) = -\mathrm{i}\,\mathbf{a}^T \mathbf{v}^{(j)} = -\lambda_i \in \mathrm{i}\,\mathbb{R}$$

for $j = 0, 1, \ldots, \lfloor \frac{n-1}{2} \rfloor$.

Now we turn to the converse inclusion. Suppose that $A = F_n \Lambda F_n^*$, where $\Lambda = \text{diag}(\boldsymbol{\lambda})$, $\boldsymbol{\lambda} \in (\mathrm{i}\,\mathbb{R})^n$ and $\boldsymbol{\lambda}$ is asymmetric. The element $a_{k,l}$ is given by

$$\begin{aligned} a_{k,l} &= \frac{1}{n}\sum_{j=0}^{n-1} \omega_n^{kj}\,\lambda_j\,\overline{\omega_n^{lj}} = \frac{1}{n}\sum_{j=1}^{\lfloor (n-1)/2 \rfloor} \lambda_j(\omega_n^{kj}\,\overline{\omega_n^{lj}} - \omega_n^{k(n-j)}\,\overline{\omega_n^{l(n-j)}}) = \\ &= \frac{1}{n}\sum_{j=1}^{\lfloor (n-1)/2 \rfloor} \lambda_j(\omega_n^{(k-l)j} - \overline{\omega_n^{(k-l)(n-j)}}) = \\ &= \frac{2\mathrm{i}}{n}\sum_{j=1}^{\lfloor (n-1)/2 \rfloor} \lambda_j \sin\left(\frac{2\pi\, j(k-l)}{n}\right). \end{aligned}$$

For $l = 0, 1, \ldots, n-1$, we get

$$a_{0,l} = -\frac{2\mathrm{i}}{n} \sum_{j=1}^{\lfloor (n-1)/2 \rfloor} \lambda_j \sin\left(\frac{2\pi j l}{n}\right) \in \mathbb{R}.$$

Now we claim that the first row of A is asymmetric. Indeed, we have

$$\begin{aligned} a_{0,n-l} &= -\frac{2\mathrm{i}}{n} \sum_{j=1}^{\lfloor (n-1)/2 \rfloor} \lambda_j \sin\left(\frac{2\pi j (n-l)}{n}\right) = \\ &= \frac{2\mathrm{i}}{n} \sum_{j=1}^{\lfloor (n-1)/2 \rfloor} \lambda_j \sin\left(\frac{2\pi j l}{n}\right) = -a_{0,l}, \end{aligned}$$

getting the claim. □

From Proposition 6.4.12 it follows that

$$\mathcal{L}_{n,n-1} = \mathcal{C}_n \oplus \mathcal{A}_n. \tag{6.46}$$

Moreover, note that $\mathcal{B}_n \oplus \mathcal{F}_n \oplus \mathcal{E}_n = \mathcal{L}_{n,1}$. Since $\mathcal{E}_n \subset \mathcal{L}_{n,n-1}$, from (6.46) it follows that

$$\mathcal{V}_n = \mathcal{C}_n \oplus \mathcal{B}_n \oplus \mathcal{F}_n \oplus \mathcal{A}_n = \mathcal{L}_{n,1} \oplus \mathcal{L}_{n,n-1}.$$

6.5 Multiplication between β-matrices

It is not difficult to see that $\mathcal{V}_n$ is closed under the operations of sum between matrices. Now we recall that the eigenvalues $\lambda_j^{(C)}$ of $C = \mathrm{circ}(\mathbf{c}) \in \mathcal{C}_n$, $j = 0, 1, \ldots, \lfloor \frac{n}{2} \rfloor$, are given by

$$\lambda_j^{(C)} = \mathbf{c}^T \mathbf{u}^{(j)}. \tag{6.47}$$

Moreover, for $j = 1, 2, \ldots \lfloor \frac{n-1}{2} \rfloor$, we have

$$\lambda_{n-j}^{(C)} = \lambda_j^{(C)}.$$

Furthermore, the eigenvalues $\lambda_j^{(B)}$ of $B = \mathrm{rcirc}(\mathbf{b}) \in \mathcal{B}_n$, $j = 0, 1, \ldots, \lfloor \frac{n}{2} \rfloor$, can be expressed as

$$\lambda_j^{(B)} = \mathbf{b}^T \mathbf{u}^{(j)}, \tag{6.48}$$

and for $j = 1, 2, \ldots \lfloor \frac{n-1}{2} \rfloor$, we have

$$\lambda_{n-j}^{(B)} = -\lambda_j^{(B)}.$$

Now we give the following

Proposition 6.5.1. *Let* $\mathbf{a} = (a_0\, a_1 \cdots a_{n-1})^T$, $\mathbf{b} = (b_0\, b_1 \cdots b_{n-1})^T \in \mathbb{R}^n$ *be such that* $\mathbf{a}$ *is symmetric and* $\mathbf{b}$ *is asymmetric. Then,* $\mathbf{a}^T\,\mathbf{b} = 0$.

Proof. First of all, we observe that $b_0 = b_{n/2} = 0$. So, we have

$$\begin{aligned}\mathbf{a}^T\,\mathbf{b} &= \sum_{j=0}^{n-1} a_j b_j = \sum_{j=1}^{n-1} a_j b_j = \sum_{j=1}^{n/2-1} a_j b_j + a_{n/2} b_{n/2} + \sum_{j=n/2+1}^{n} a_j b_j = \\ &= \sum_{j=1}^{n/2-1} a_j b_j + \sum_{j=1}^{n/2-1} a_{n-j} b_{n-j} = \sum_{j=1}^{n/2-1} a_j b_j - \sum_{j=1}^{n/2-1} a_j b_j = 0.\end{aligned}$$

□

Proposition 6.5.2. *The eigenvalues* $\lambda_j^{(F)}$ *of* $F = rcirc(\mathbf{f}) \in \mathcal{F}_n$, $j = 0, 1, \ldots, \lfloor \frac{n}{2} \rfloor$, *are given by*

$$\lambda_j^{(F)} = \mathbf{f}^T \mathbf{v}^{(j)}, \tag{6.49}$$

and for $j = 1,\ 2, \ldots \lfloor \frac{n-1}{2} \rfloor$, *we get*

$$\lambda_{n-j}^{(F)} = -\lambda_j^{(F)}.$$

Proof. We consider the following set of eigenvectors, whose first component is 1.

$$\mathbf{u}^{(j)} + \mathbf{v}^{(j)}, \quad j = 0, 1, \ldots, \lceil \tfrac{n-1}{2} \rceil;$$

$$\mathbf{u}^{(j)} - \mathbf{v}^{(j)}, \quad j = 1, 2, \ldots \lfloor \tfrac{n-1}{2} \rfloor.$$

Hence, by Proposition 6.5.1, we obtain

$$\lambda_j^{(F)} = \mathbf{f}^T(\mathbf{u}^{(j)} + \mathbf{v}^{(j)}) = \mathbf{f}^T\,\mathbf{v}^{(j)}, \qquad j = 0, 1, \ldots, \lceil \tfrac{n-1}{2} \rceil;$$

$$\lambda_{n-j}^{(F)} = \mathbf{f}^T(\mathbf{u}^{(j)} - \mathbf{v}^{(j)}) = -\mathbf{f}^T\,\mathbf{v}^{(j)}, \quad j = 1, 2, \ldots \lfloor \tfrac{n-1}{2} \rfloor.$$

□

Proposition 6.5.3. *The eigenvalues* $\lambda_j^{(A)}$ *of* $A = circ(\mathbf{a}) \in \mathcal{A}_n$, $j = 0, 1, \ldots, \lfloor \frac{n}{2} \rfloor$, *are given by*

$$\lambda_j^{(A)} = \mathrm{i}\,\mathbf{a}^T \mathbf{v}^{(j)}, \tag{6.50}$$

and for $j = 1,\ 2, \ldots \lfloor \frac{n-1}{2} \rfloor$, *we get*

$$\lambda_{n-j}^{(A)} = -\lambda_j^{(A)}.$$

Proof. We consider the following set of eigenvectors, whose first component is 1.

$$\mathbf{u}^{(j)} + \mathrm{i}\,\mathbf{v}^{(j)}, \quad j = 0, 1, \ldots, \lceil \tfrac{n-1}{2} \rceil;$$

$$\mathbf{u}^{(j)} - \mathrm{i}\,\mathbf{v}^{(j)}, \quad j = 1, 2, \ldots \lfloor \tfrac{n-1}{2} \rfloor.$$

Hence, by Proposition 6.5.1, we obtain

$$\lambda_j^{(A)} = \mathbf{a}^T(\mathbf{u}^{(j)} + \mathrm{i}\,\mathbf{v}^{(j)}) = \mathrm{i}\,\mathbf{a}^T\,\mathbf{v}^{(j)}, \qquad j = 0, 1, \ldots, \lceil \tfrac{n-1}{2} \rceil;$$

$$\lambda_{n-j}^{(A)} = \mathbf{a}^T(\mathbf{u}^{(j)} - \mathrm{i}\,\mathbf{v}^{(j)}) = -\mathrm{i}\,\mathbf{a}^T\,\mathbf{v}^{(j)}, \quad j = 1, 2, \ldots \lfloor \tfrac{n-1}{2} \rfloor.$$

□

Now we give the following

Proposition 6.5.4. *Given two γ-matrices G_1, G_2, then the following results hold.*

6.5.4.1) *If G_1, $G_2 \in \mathcal{C}_n$, then $G_1 G_2 \in \mathcal{C}_n$;*

6.5.4.2) *If G_1, $G_2 \in \mathcal{B}_n$, then $G_1 G_2 \in \mathcal{C}_n$;*

6.5.4.3) *If $G_1 \in \mathcal{C}_n$ and $G_2 \in \mathcal{B}_n$, then $G_1 G_2 = G_2 G_1 \in \mathcal{B}_n$.*

Proof. 6.5.4.1) It follows immediately from the fact that $\mathcal{G}_n$ is an algebra.

6.5.4.2) If $G_1 = Q_n \Lambda^{(G_1)} Q_n^T$ and $G_2 = Q_n \Lambda^{(G_2)} Q_n^T$, then $G_1 G_2 = Q_n \Lambda^{(G_1)} \Lambda^{(G_2)} Q_n^T$, where $\Lambda^{(G_1)}\Lambda^{(G_2)} = \mathrm{diag}\,(\lambda_0^{(G_1)}\lambda_0^{(G_2)} \quad \lambda_1^{(G_1)}\lambda_1^{(G_2)} \cdots \lambda_{n-1}^{(G_1)}\lambda_{n-1}^{(G_1)})$. Since the eigenvalues of G_1 and G_2 are asymmetric, we get that the eigenvalues of $G_1 G_2$ are symmetric. Hence, $G_1 G_2 \in \mathcal{C}_n$.

6.5.4.3) We first note that, since $\mathcal{G}_n$ is an algebra, we get that $G_1 G_2 = G_2 G_1$. Since the eigenvalues of G_1 are symmetric and those of G_2 are asymmetric, arguing analogously as in the proof of 6.5.4.2) it is possible to check that the eigenvalues of $G_1 G_2$ are asymmetric. Therefore, $G_1 G_2 \in \mathcal{B}_n$. □

It is not difficult to see that, given $C \in \mathcal{C}_n$ and $V \in \mathcal{V}_n$, the eigenvalues of CV are equal to those of VC and are given by

$$\lambda_j^{(CV)} = \lambda_j^{(VC)} = \lambda_j^{(C)}\,\lambda_j^{(V)}, \quad j = 0, 1, \ldots, n-1.$$

Now we prove the following

Theorem 6.5.5. *Let $B \in \mathcal{B}_n$, $B = \mathrm{rcirc}(\mathbf{b})$, and $F \in \mathcal{F}_n$, $F = \mathrm{rcirc}(\mathbf{f})$. Then, $BF \in \mathcal{A}_n$ and the eigenvalues of BF are expressed by*

$$\lambda_j^{(BF)} = \mathrm{i}\,\lambda_j^{(B)}\,\lambda_j^{(F)}, \quad j = 0,1,\ldots,\lceil \tfrac{n-1}{2} \rceil;$$

$$\lambda_{n-j}^{(BF)} = -\lambda_j^{(BF)}, \qquad j = 1,2,\ldots \lfloor \tfrac{n-1}{2} \rfloor.$$

Proof. Let $A = BF$. Since $B, F \in \mathcal{L}_{n,1}$, then $A \in \mathcal{L}_{n,n-1}$ (see, e.g., [54]). So, to prove that $A \in \mathcal{A}_n$ it is enough to show that the first row of the matrix A is asymmetric, that is $a_{0,n-j} = -a_{0,j}$, $j = 0,1,\ldots \lceil \frac{n-1}{2} \rceil$. Indeed, if n is odd, we get

$$\begin{aligned} a_{0,n-j} &= b_0 f_{n-j} + \sum_{l=1}^{(n-1)/2} b_l \left(f_{(n-j+l) \pmod n} + f_{(2n-j-l) \pmod n} \right) = \\ &= -b_0 f_j - \sum_{l=1}^{(n-1)/2} b_l \left(f_{(j+l) \pmod n} + f_{(n+j-l) \pmod n} \right) = -a_{0,j}, \end{aligned}$$

and when n is even, we have

$$\begin{aligned} a_{0,n-j} &= b_0 f_{n-j} + b_{n/2} f_{(n/2-j) \pmod n} + \\ &+ \sum_{l=1}^{n/2-1} b_l \left(f_{(n-j+l) \pmod n} + f_{(2n-j-l) \pmod n} \right) = \\ &= -b_0 f_j - b_{n/2} f_{(n/2-j) \pmod n} - \\ &- \sum_{l=1}^{n/2-1} b_l \left(f_{(j+l) \pmod n} + f_{(n+j-l) \pmod n} \right) = -a_{0,j}. \end{aligned}$$

Thus, $A \in \mathcal{A}_n$.

We consider the following set of eigenvectors, whose first component is 1:

$$\mathbf{u}^{(j)} + \mathrm{i}\,\mathbf{v}^{(j)}, \quad j = 0,1,\ldots,\lceil \tfrac{n-1}{2} \rceil.$$

Hence, by Proposition 6.5.1, we obtain

$$\begin{aligned} \lambda_j^{(A)} &= \mathbf{b}^T F(\mathbf{u}^{(j)} + \mathrm{i}\,\mathbf{v}^{(j)}) = \left(\frac{1-\mathrm{i}}{2} \right) \mathbf{b}^T F(\mathbf{u}^{(j)} - \mathbf{v}^{(j)} + \mathrm{i}(\mathbf{u}^{(j)} + \mathbf{v}^{(j)})) = \\ &= \left(\frac{1-\mathrm{i}}{2} \right) \mathbf{b}^T (-\lambda_j^{(F)} (\mathbf{u}^{(j)} - \mathbf{v}^{(j)}) + \mathrm{i}\,\lambda_j^{(F)} (\mathbf{u}^{(j)} + \mathbf{v}^{(j)})) = \\ &= \left(\frac{1-\mathrm{i}}{2} \right) (-\lambda_j^{(F)} \lambda_j^{(B)} + \mathrm{i}\,\lambda_j^{(F)} \lambda_j^{(B)}) = \mathrm{i}\,\lambda_j^{(B)}\,\lambda_j^{(F)} \end{aligned}$$

for $j = 0,1,\ldots,\lceil \frac{n-1}{2} \rceil$;

$$\lambda_{n-j}^{(A)} = -\lambda_j^{(A)} = -\mathrm{i}\,\lambda_j^{(B)}\,\lambda_j^{(F)}$$

for $j = 1,2,\ldots \lfloor \frac{n-1}{2} \rfloor$, since the eigenvalues of $A \in \mathcal{A}_n$ are asymmetric. □

Now we demonstrate the following

Theorem 6.5.6. *Let $B \in \mathcal{B}_n$, $B =$rcirc$(\mathbf{b})$, and $F \in \mathcal{F}_n$, $F =$rcirc$(\mathbf{f})$. Then, $FB \in \mathcal{A}_n$ and the eigenvalues of FB are expressed by*

$$\lambda_j^{(FB)} = -\mathrm{i}\,\lambda_j^{(B)}\,\lambda_j^{(F)}, \quad j = 0,1,\ldots,\lceil \tfrac{n-1}{2} \rceil;$$

$$\lambda_{n-j}^{(FB)} = -\lambda_j^{(FB)}, \qquad j = 1,2,\ldots \lfloor \tfrac{n-1}{2} \rfloor.$$

Proof. Let $A = FB$. As $B, F \in \mathcal{L}_{n,1}$, then $A \in \mathcal{L}_{n,n-1}$ (see, e.g., [54]). So, to prove that $A \in \mathcal{A}_n$ it is sufficient to show that the first row of the matrix A is asymmetric, that is $a_{0,n-j} = -a_{0,j}$, $j = 0,1,\ldots \lceil \tfrac{n-1}{2} \rceil$. Indeed, we have

$$\begin{aligned} a_{0,n-j} &= \sum_{l=1}^{\lfloor (n-1)/2 \rfloor} f_l \,(b_{(n-j+l) \pmod n} - b_{(2n-j-l) \pmod n}) = \\ &= -\sum_{l=1}^{\lfloor (n-1)/2 \rfloor} f_l \,(b_{(j+l) \pmod n} - b_{(n+j-l) \pmod n}) = -a_{0,j} \end{aligned}$$

for $j = 1,2,\ldots \lfloor \tfrac{n-1}{2} \rfloor$. Therefore, $A \in \mathcal{A}_n$.

We consider the following set of eigenvectors, whose first component is 1:

$$\mathbf{u}^{(j)} + \mathrm{i}\,\mathbf{v}^{(j)}, \quad j = 0,1,\ldots,\lceil \tfrac{n-1}{2} \rceil.$$

Hence, by Proposition 6.5.1, we obtain

$$\begin{aligned} \lambda_j^{(A)} &= \mathbf{f}^T B(\mathbf{u}^{(j)} + \mathrm{i}\,\mathbf{v}^{(j)}) = \mathbf{f}^T(\lambda_j^{(B)}\,\mathbf{u}^{(j)} - \mathrm{i}\,\lambda_j^{(B)}\,\mathbf{v}^{(j)}) = \\ &= -\mathrm{i}\,\lambda_j^{(B)}\,(\mathbf{f}^T\mathbf{v}^{(j)}) = -\mathrm{i}\,\lambda_j^{(F)}\,\lambda_j^{(B)} \end{aligned}$$

for $j = 0,1,\ldots,\lceil \tfrac{n-1}{2} \rceil$;

$$\lambda_{n-j}^{(A)} = -\lambda_j^{(A)} = \mathrm{i}\,\lambda_j^{(F)}\,\lambda_j^{(B)}$$

for $j = 1,2,\ldots \lfloor \tfrac{n-1}{2} \rfloor$, because the eigenvalues of $A \in \mathcal{A}_n$ are asymmetric. □

Observe that, given $B \in \mathcal{B}_n$ and $F \in \mathcal{F}_n$, we get that $\lambda_j^{(FB)} = -\lambda_j^{(BF)}$. Therefore, $FB = -BF$.

Now we prove the following

Theorem 6.5.7. *Let $A \in \mathcal{A}_n$, $A =$circ$(\mathbf{a})$ and $B \in \mathcal{B}_n$, $B =$rcirc$(\mathbf{b})$. Then, $AB \in \mathcal{F}_n$ and the eigenvalues of AB are expressed by*

$$\lambda_j^{(AB)} = -\mathrm{i}\,\lambda_j^{(A)}\,\lambda_j^{(B)}, \quad j = 0,1,\ldots,\lceil \tfrac{n-1}{2} \rceil;$$

$$\lambda_{n-j}^{(AB)} = -\lambda_j^{(AB)}, \qquad j = 1,2,\ldots \lfloor \tfrac{n-1}{2} \rfloor.$$

Proof. Let $F = AB$. Since $A \in \mathcal{L}_{n,n-1}$ and $B \in \mathcal{L}_{n,1}$, then $F \in \mathcal{L}_{n,1}$ (see, e.g., [54]). So, to prove that $F \in \mathcal{F}_n$ it is enough to show that the first row of the matrix F is asymmetric, that is $f_{0,n-j} = -f_{0,j}$, $j = 0,1,\ldots\lceil\frac{n-1}{2}\rceil$. Indeed, we have

$$\begin{aligned} f_{0,n-j} &= \sum_{l=1}^{\lfloor (n-1)/2 \rfloor} a_l \left(b_{(n-j+l) \pmod n} - b_{(2n-j-l) \pmod n}\right) = \\ &= -\sum_{l=1}^{\lfloor (n-1)/2 \rfloor} a_l \left(b_{(j+l) \pmod n} - b_{(n+j-l) \pmod n}\right) = -f_{0,j} \end{aligned}$$

for $j = 1,2,\ldots\lfloor\frac{n-1}{2}\rfloor$. Therefore, $F \in \mathcal{F}_n$.

We consider the following set of eigenvectors, whose first component is 1:

$$\mathbf{u}^{(j)} + \mathbf{v}^{(j)}, \quad j = 0,1,\ldots,\lceil\tfrac{n-1}{2}\rceil.$$

Hence, by Proposition 6.5.1, we obtain

$$\begin{aligned} \lambda_j^{(F)} &= \mathbf{a}^T B(\mathbf{u}^{(j)} + \mathbf{v}^{(j)}) = \mathbf{a}^T(\lambda_j^{(B)}\mathbf{u}^{(j)} - \lambda_j^{(B)}\mathbf{v}^{(j)}) = \\ &= -\lambda_j^{(B)}(\mathbf{a}^T\mathbf{v}^{(j)}) = \mathrm{i}\,\lambda_j^{(A)}\lambda_j^{(B)} \end{aligned}$$

for $j = 0,1,\ldots,\lceil\frac{n-1}{2}\rceil$;

$$\lambda_{n-j}^{(F)} = -\lambda_j^{(F)} = -\mathrm{i}\,\lambda_j^{(A)}\lambda_j^{(B)}$$

for $j = 1,2,\ldots\lfloor\frac{n-1}{2}\rfloor$, because the eigenvalues of $F \in \mathcal{F}_n$ are asymmetric. □

Theorem 6.5.8. *Let $B \in \mathcal{B}_n$, $B =$rcirc$(\mathbf{b})$, and $A \in \mathcal{A}_n$, $A =$circ$(\mathbf{a})$. Then, $BA \in \mathcal{F}_n$ and the eigenvalues of BA are given by*

$$\lambda_j^{(BA)} = -\mathrm{i}\,\lambda_j^{(B)}\lambda_j^{(A)}, \quad j = 0,1,\ldots,\lceil\tfrac{n-1}{2}\rceil;$$

$$\lambda_{n-j}^{(BA)} = -\lambda_j^{(BA)}, \qquad j = 1,2,\ldots\lfloor\tfrac{n-1}{2}\rfloor.$$

Proof. Let $F = BA$. Since $A \in \mathcal{L}_{n,n-1}$ and $B \in \mathcal{L}_{n,1}$, then $F \in \mathcal{L}_{n,1}$ (see, e.g., [54]). So, to prove that $F \in \mathcal{F}_n$ it is enough to show that the first row of the matrix F is asymmetric, namely $f_{0,n-j} = -f_{0,j}$, $j = 0,1,\ldots\lceil\frac{n-1}{2}\rceil$. Indeed, if n is odd, we get

$$\begin{aligned} f_{0,n-j} &= b_0 a_{n-j} + \sum_{l=1}^{(n-1)/2} b_l \left(a_{(l-n+j) \pmod n} + a_{(l-j) \pmod n}\right) = \\ &= -b_0 a_j - \sum_{l=1}^{(n-1)/2} b_l \left(a_{(j-l) \pmod n} + a_{(-j-l) \pmod n}\right) = -f_{0,j}, \end{aligned}$$

and when n is even, we have

$$\begin{aligned} f_{0,n-j} &= b_0\, a_{n-j} + b_{n/2}\, a_{(n/2-j) \pmod n} + \\ &+ \sum_{l=1}^{n/2-1} b_l \left(a_{(l-n+j) \pmod n} + a_{(l-j) \pmod n}\right) = \\ &= -b_0\, a_j - b_{n/2}\, a_{(n/2-j) \pmod n} - \\ &- \sum_{l=1}^{n/2-1} b_l \left(a_{(j-l) \pmod n} + a_{(-j-l) \pmod n}\right) = -f_{0,j}. \end{aligned}$$

Thus, $F \in \mathcal{F}_n$.

We consider the following set of eigenvectors, whose first component is 1:

$$\mathbf{u}^{(j)} + \mathbf{v}^{(j)}, \quad j = 0, 1, \ldots, \lceil \tfrac{n-1}{2} \rceil.$$

By Proposition 6.5.1, we have

$$\begin{aligned} \lambda_j^{(F)} &= \mathbf{b}^T A(\mathbf{u}^{(j)} + \mathbf{v}^{(j)}) = \mathbf{b}^T A \left(\left(\frac{1-\mathrm{i}}{2} \right) (\mathbf{u}^{(j)} + \mathrm{i}\mathbf{v}^{(j)}) + \left(\frac{1+\mathrm{i}}{2} \right) (\mathbf{u}^{(j)} - \mathrm{i}\mathbf{v}^{(j)}) \right) = \\ &= \mathbf{b}^T \lambda_j^{(F)} \left(\frac{1-\mathrm{i}}{2} \right) (\mathbf{u}^{(j)} + \mathrm{i}\mathbf{v}^{(j)}) - \mathbf{b}^T \lambda_j^{(F)} \left(\frac{1+\mathrm{i}}{2} \right) (\mathbf{u}^{(j)} - \mathrm{i}\mathbf{v}^{(j)}) = \\ &= \left(\frac{1-\mathrm{i}}{2} \right) \lambda_j^{(B)} \lambda_j^{(A)} - \left(\frac{1+\mathrm{i}}{2} \right) \lambda_j^{(B)} \lambda_j^{(A)} = -\mathrm{i}\, \lambda_j^{(B)}\, \lambda_j^{(A)} \end{aligned}$$

for $j = 0, 1, \ldots, \lceil \frac{n-1}{2} \rceil$;

$$\lambda_{n-j}^{(F)} = -\lambda_j^{(F)} = \mathrm{i}\lambda_j^{(B)}\, \lambda_j^{(A)}$$

for $j = 1, 2, \ldots \lfloor \frac{n-1}{2} \rfloor$, since the eigenvalues of $F \in \mathcal{F}_n$ are asymmetric. □

Note that, given $A \in \mathcal{A}_n$ and $B \in \mathcal{B}_n$, we have that $\lambda_j^{(AB)} = -\lambda_j^{(BA)}$. Hence, $AB = -BA$.

Now we give the following

Theorem 6.5.9. *Let $A \in \mathcal{A}_n$, $A = \mathrm{circ}(\mathbf{a})$ and $F \in \mathcal{F}_n$, $F = \mathrm{rcirc}(\mathbf{f})$. Then, $AF \in \mathcal{B}_n$ and the eigenvalues of AF are expressed by*

$$\lambda_j^{(AF)} = -\mathrm{i}\, \lambda_j^{(A)}\, \lambda_j^{(F)}, \quad j = 0, 1, \ldots, \lceil \tfrac{n-1}{2} \rceil;$$

$$\lambda_{n-j}^{(AF)} = -\lambda_j^{(AF)}, \qquad j = 1, 2, \ldots \lfloor \tfrac{n-1}{2} \rfloor.$$

Proof. Let $B = AF$. Since $A \in \mathcal{L}_{n,n-1}$ and $F \in \mathcal{L}_{n,1}$, then $B \in \mathcal{L}_{n,1}$ (see, e.g., [54]). Thus, to prove that $B \in \mathcal{B}_n$ it is sufficient to demonstrate that the first row of the matrix B is asymmetric,

that is $b_{0,n-j} = -b_{0,j}$, $j = 0,1,\ldots\lceil\frac{n-1}{2}\rceil$. Indeed, it is

$$\begin{aligned} b_{0,n-j} &= \sum_{l=1}^{\lfloor(n-1)/2\rfloor} a_l \left(f_{(n-j+l) \pmod n} - f_{(2n-j-l) \pmod n}\right) = \\ &= -\sum_{l=1}^{\lfloor(n-1)/2\rfloor} a_l \left(f_{(j+l) \pmod n} - f_{(n+j-l) \pmod n}\right) = -b_{0,j} \end{aligned} \tag{6.51}$$

for $j = 1,2,\ldots\lfloor\frac{n-1}{2}\rfloor$. Therefore, $B \in \mathcal{B}_n$.

We consider the following set of eigenvectors, whose first component is 1:

$$\mathbf{u}^{(j)}, \quad j = 0,1,\ldots,\lceil\tfrac{n-1}{2}\rceil.$$

Hence, by Proposition 6.5.1, we obtain

$$\begin{aligned} \lambda_j^{(B)} &= \mathbf{a}^T F\,\mathbf{u}^{(j)} = \frac{1}{2}\mathbf{a}^T F(\mathbf{u}^{(j)} + \mathbf{v}^{(j)}) - \frac{1}{2}\mathbf{a}^T F\,(\mathbf{u}^{(j)} - \mathbf{v}^{(j)}) = \\ &= \frac{1}{2}\mathbf{a}^T \lambda_j^{(F)}\,(\mathbf{u}^{(j)} + \mathbf{v}^{(j)}) - \frac{1}{2}\mathbf{a}^T \lambda_j^{(F)}\,(\mathbf{u}^{(j)} - \mathbf{v}^{(j)}) = -\mathrm{i}\,\lambda_j^{(A)}\,\lambda_j^{(F)} \end{aligned}$$

for $j = 0,1,\ldots,\lceil\frac{n-1}{2}\rceil$;

$$\lambda_{n-j}^{(B)} = -\lambda_j^{(B)} = \mathrm{i}\,\lambda_j^{(A)}\,\lambda_j^{(F)}$$

for $j = 1,2,\ldots\lfloor\frac{n-1}{2}\rfloor$, because the eigenvalues of $B \in \mathcal{B}_n$ are asymmetric. □

Now we prove the following

Theorem 6.5.10. *Let $A \in \mathcal{A}_n$, $A = \mathrm{circ}(\mathbf{a})$ and $F \in \mathcal{F}_n$, $F = \mathrm{rcirc}(\mathbf{f})$. Then, $FA \in \mathcal{B}_n$ and the eigenvalues of FA are given by*

$$\lambda_j^{(FA)} = -\mathrm{i}\,\lambda_j^{(F)}\,\lambda_j^{(A)}, \quad j = 0,1,\ldots,\lceil\tfrac{n-1}{2}\rceil;$$

$$\lambda_{n-j}^{(FA)} = -\lambda_j^{(FA)}, \qquad j = 1,2,\ldots\lfloor\tfrac{n-1}{2}\rfloor.$$

Proof. Let $B = FA$. Since $A \in \mathcal{L}_{n,n-1}$ and $F \in \mathcal{L}_{n,1}$, then $B \in \mathcal{L}_{n,1}$ (see, e.g., [54]). Thus, to prove that $B \in \mathcal{B}_n$ it is sufficient to demonstrate that the first row of the matrix B is asymmetric, that is $b_{0,n-j} = -b_{0,j}$, $j = 0,1,\ldots\lceil\frac{n-1}{2}\rceil$. Indeed, we get

$$\begin{aligned} b_{0,n-j} &= \sum_{l=1}^{\lfloor(n-1)/2\rfloor} f_l \left(a_{(l-n+j) \pmod n} - a_{(l-j) \pmod n}\right) = \\ &= -\sum_{l=1}^{\lfloor(n-1)/2\rfloor} f_l \left(a_{(j-l) \pmod n} - a_{(-j-l) \pmod n}\right) = -b_{0,j} \end{aligned} \tag{6.52}$$

for $j = 1,2,\ldots\lfloor\frac{n-1}{2}\rfloor$. Hence, $B \in \mathcal{B}_n$.

We consider the following set of eigenvectors, whose first component is 1:

$$\mathbf{u}^{(j)}, \quad j = 0, 1, \ldots, \lceil \tfrac{n-1}{2} \rceil.$$

Hence, by Proposition 6.5.1, we obtain

$$\begin{aligned} \lambda_j^{(B)} &= \mathbf{f}^T A\, \mathbf{u}^{(j)} = \frac{1}{2}\mathbf{f}^T A(\mathbf{u}^{(j)} + \mathrm{i}\,\mathbf{v}^{(j)}) + \frac{1}{2}\mathbf{f}^T A\,(\mathbf{u}^{(j)} - \mathrm{i}\,\mathbf{v}^{(j)}) = \\ &= \frac{1}{2}\mathbf{f}^T \lambda_j^{(A)}\,(\mathbf{u}^{(j)} + \mathrm{i}\,\mathbf{v}^{(j)}) + \frac{1}{2}\mathbf{f}^T \lambda_j^{(A)}\,(\mathbf{u}^{(j)} - \mathrm{i}\,\mathbf{v}^{(j)}) = \mathrm{i}\,\lambda_j^{(F)}\,\lambda_j^{(A)} \end{aligned}$$

for $j = 0, 1, \ldots, \lceil \frac{n-1}{2} \rceil$;

$$\lambda_{n-j}^{(B)} = -\lambda_j^{(B)} = -\mathrm{i}\,\lambda_j^{(F)}\,\lambda_j^{(A)}$$

for $j = 1, 2, \ldots \lfloor \frac{n-1}{2} \rfloor$, since the eigenvalues of $B \in \mathcal{B}_n$ are asymmetric. □

Observe that, if $A \in \mathcal{A}_n$ and $F \in \mathcal{F}_n$, then $\lambda_j^{(AF)} = -\lambda_j^{(FA)}$. Hence, $AF = -FA$.

Moreover note that, if $B_1, B_2 \in \mathcal{B}_n$, $F_1, F_2 \in \mathcal{F}_n$, $A_1, A_2 \in \mathcal{A}_n$, then $B_1 B_2, F_1 F_2, A_1 A_2 \in \mathcal{C}_n$.

6.6 Invertible β-matrices

In this section we present some results about invertibility of β-matrices. We prove the following

Theorem 6.6.1. *Given $V_1 \in \mathcal{V}_n$, $V_1 = C_1 + B_1 + F_1 + A_1$, with $C_1 \in \mathcal{C}_n$, $B_1 \in \mathcal{B}_n$, $F_1 \in \mathcal{F}_n$, $A_1 \in \mathcal{A}_n$, set $\sigma_j^{(A_1)} = -\mathrm{i}\,\lambda_j^{(A_1)}$, $j = 0, 1, \ldots, \lceil \frac{n-1}{2} \rceil$. If the matrices*

$$\Theta_j = \begin{pmatrix} \lambda_j^{(C_1)} & \lambda_j^{(B_1)} & \lambda_j^{(F_1)} & -\sigma_j^{(A_1)} \\ \lambda_j^{(B_1)} & \lambda_j^{(C_1)} & \sigma_j^{(A_1)} & -\lambda_j^{(F_1)} \\ \lambda_j^{(F_1)} & -\sigma_j^{(A_1)} & \lambda_j^{(C_1)} & -\lambda_j^{(B_1)} \\ \sigma_j^{(A_1)} & -\lambda_j^{(F_1)} & \lambda_j^{(B_1)} & \lambda_j^{(C_1)} \end{pmatrix} \in \mathbb{R}^{4\times 4},$$

$j = 0, 1, \ldots, \lceil \frac{n-1}{2} \rceil$, are invertible, then there exists $V_2 \in \mathcal{V}_n$ such that $V_1 V_2 = I_n$.

Proof. First of all note that, if $V_2 \in \mathcal{V}_n$, then $V_2 = C_2 + B_2 + F_2 + A_2$, with $C_2 \in \mathcal{C}_n$, $B_2 \in \mathcal{B}_n$, $F_2 \in \mathcal{F}_n$, $A_2 \in \mathcal{A}_n$.

Observe that $V_1 V_2 = C_3 + B_3 + F_3 + A_3$, where

$$\begin{aligned} C_3 &= C_1C_2 + B_1B_2 + F_1F_2 + A_1A_2 \in \mathcal{C}_n, \\ B_3 &= C_1B_2 + B_1C_2 + F_1A_2 + A_1F_2 \in \mathcal{B}_n, \\ F_3 &= C_1F_2 + F_1C_2 + B_1A_2 + A_1B_2 \in \mathcal{F}_n, \\ A_3 &= C_1A_2 + A_1C_2 + B_1F_2 + F_1B_2 \in \mathcal{A}_n. \end{aligned}$$

By imposing $C_3 = I_n$, we get

$$\lambda_j^{(C_1)}\lambda_j^{(C_2)} + \lambda_j^{(B_1)}\lambda_j^{(B_2)} + \lambda_j^{(F_1)}\lambda_j^{(F_2)} + \lambda_j^{(A_1)}\lambda_j^{(A_2)} = 1$$

for $j = 0, 1, \ldots, \lceil \frac{n-1}{2} \rceil$.

Moreover, by imposing $B_3 = O_n$, by virtue of Theorems 6.5.9 and 6.5.10 it follows that

$$\lambda_j^{(B_1)}\lambda_j^{(C_2)} + \lambda_j^{(C_1)}\lambda_j^{(B_2)} - \mathrm{i}\,\lambda_j^{(A_1)}\lambda_j^{(F_2)} + \mathrm{i}\,\lambda_j^{(F_1)}\lambda_j^{(A_2)} = 0$$

for $j = 0, 1, \ldots, \lceil \frac{n-1}{2} \rceil$.

Furthermore, we impose $F_3 = O_n$. Then, from Theorems 6.5.7 and 6.5.8, it follows that

$$\lambda_j^{(F_1)}\lambda_j^{(C_2)} + \mathrm{i}\,\lambda_j^{(A_1)}\lambda_j^{(B_2)} + \lambda_j^{(C_1)}\lambda_j^{(F_2)} + \mathrm{i}\,\lambda_j^{(B_1)}\lambda_j^{(A_2)} = 0$$

for $j = 0, 1, \ldots, \lceil \frac{n-1}{2} \rceil$.

Finally, by imposing $A_3 = O_n$, from Theorems 6.5.5 and 6.5.6 we obtain

$$\lambda_j^{(A_1)}\lambda_j^{(C_2)} - \mathrm{i}\,\lambda_j^{(F_1)}\lambda_j^{(B_2)} + \mathrm{i}\,\lambda_j^{(B_1)}\lambda_j^{(F_2)} + \lambda_j^{(C_1)}\lambda_j^{(A_2)} = 0$$

for $j = 0, 1, \ldots, \lceil \frac{n-1}{2} \rceil$.

Now, put $\sigma_j^{(A_2)} = -\mathrm{i}\,\lambda_j^{(A_2)}$, $j = 0, 1, \ldots, \lceil \frac{n-1}{2} \rceil$, $\boldsymbol{\vartheta}_j{}^T = (\lambda_j^{(C_2)}\ \lambda_j^{(B_2)}\ \lambda_j^{(F_2)}\ \sigma_j^{(A_2)})$. Since Θ_j is invertible, then the system $\Theta_j\,\boldsymbol{\vartheta}_j = (1\,0\,0\,0)^T$ has a unique solution. This ends the proof. □

Thus, it is not difficult to show that in most cases it is possible to compute the inverse of a β-matrix by means of DFFT and Hartley-type transforms.

6.7 Toeplitz matrix preconditioning

For each $n \in \mathbb{N}$, let us consider the following class:

$$\mathcal{T}_n = \{ T_n \in \mathbb{R}^{n\times n} : T_n = (t_{k,j})_{k,j}, t_{k,j} = t_{|k-j|},\ k, j \in \{0, 1, \ldots, n-1\}\ \}. \tag{6.53}$$

Observe that the class defined in (6.53) coincides with the family of all real symmetric Toeplitz matrices.

Now we consider the following problem.

Given $T_n \in \mathcal{T}_n$, find

$$V_n(T_n) = \min_{V \in \mathcal{V}_n} \|V - T_n\|_F,$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

It is not difficult to see that, since T_n is symmetric, then we can assume that $V_n(T_n)$ is symmetric. Therefore, $V_n(T_n) = C_n(T_n) + B_n(T_n) + F_n(T_n)$, where $C_n(T_n) \in \mathcal{C}_n$, $B_n(T_n) \in \mathcal{B}_n$, and $F_n(T_n) \in \mathcal{F}_n$.

Theorem 6.7.1. *Let $\widehat{\mathcal{G}}_n = \mathcal{S}_n + \mathcal{H}_{n,1}$. Given $T_n \in \mathcal{T}_n$, one has*

$$G_n(T_n) = C_n(T_n) + B_n(T_n) = \min_{G \in \widehat{\mathcal{G}}_n} \|G - T_n\|_F = \min_{G \in \mathcal{G}_n} \|G - T_n\|_F, \tag{6.54}$$

*where $C_n(T_n) =$*circ$(\mathbf{c})$*, with*

$$c_j = \frac{(n-j)t_j + j t_{n-j}}{n}, \quad j \in \{1,2,\ldots,n-1\}; \tag{6.55}$$

$$c_0 = t_0, \tag{6.56}$$

*and $B_n(T_n) =$*rcirc$(\mathbf{b})$*, where: for n even and $j \in \{1,2,\ldots,n-1\} \setminus \{n/2\}$,*

$$\begin{aligned} b_j &= \frac{1}{2n}\left(\frac{4j-2n}{n}(t_j - t_{n-j}) + 4\sum_{k=1}^{(j-3)/2}\frac{2k+1}{n}(t_{2k+1} - t_{n-2k-1}) + \right. \\ &+ \left. 4\sum_{k=1}^{(n-j-3)/2}\frac{2k+1}{n}(t_{2k+1} - t_{n-2k-1})\right), \quad j \text{ odd;} \end{aligned} \tag{6.57}$$

$$\begin{aligned} b_j &= \frac{1}{2n}\left(\frac{4j-2n}{n}(t_j - t_{n-j}) + 4\sum_{k=1}^{j/2-1}\frac{2k}{n}(t_{2k} - t_{n-2k}) + \right. \\ &+ \left. 4\sum_{k=1}^{(n-j)/2-1}\frac{2k}{n}(t_{2k} - t_{n-2k})\right), \quad j \text{ even;} \end{aligned} \tag{6.58}$$

for n even,

$$b_0 = \frac{2}{n}\left(\sum_{k=1}^{n/2-1}\frac{2k}{n}(t_{2k} - t_{n-2k})\right), \tag{6.59}$$

$$b_{n/2} = \frac{4}{n}\left(\sum_{k=1}^{n/4-1}\frac{2k}{n}(t_{2k} - t_{n-2k})\right); \tag{6.60}$$

for n odd and $j \in \{1,2,\ldots,n-1\}$,

$$\begin{aligned} b_j &= \frac{1}{2n}\left(\frac{4j-2n}{n}(t_j - t_{n-j}) + 4\sum_{k=0}^{(j-3)/2}\frac{2k+1}{n}(t_{2k+1} - t_{n-2k-1}) + \right. \\ &+ \left. 4\sum_{k=1}^{(n-j)/2-1}\frac{2k}{n}(t_{2k} - t_{n-2k})\right), \quad j \text{ odd;} \end{aligned} \tag{6.61}$$

$$\begin{aligned} b_j &= \frac{1}{2n}\left(\frac{4j-2n}{n}(t_j - t_{n-j}) + 4\sum_{k=1}^{j/2-1}\frac{2k}{n}(t_{2k} - t_{n-2k}) + \right. \\ &+ \left. 4\sum_{k=0}^{(n-j-3)/2}\frac{2k+1}{n}(t_{2k+1} - t_{n-2k-1})\right), \quad j \text{ even;} \end{aligned} \tag{6.62}$$

for n odd,

$$b_0 = \frac{2}{n}\left(\sum_{k=0}^{(n-3)/2}\frac{2k+1}{n}(t_{2k+1} - t_{n-2k-1})\right). \tag{6.63}$$

Proof. Let us define

$$\phi(\mathbf{c},\mathbf{b}) = \|T_n - \mathrm{circ}(\mathbf{c}) - \mathrm{circ}(\mathbf{b})\|_F^2$$

for any two symmetric vectors $\mathbf{c}, \mathbf{b} \in \mathbb{R}^n$. If $j \in \{1,2,\ldots,n-1\}$, then we get

$$\frac{\partial\phi(\mathbf{c},\mathbf{b})}{\partial c_j} = -4(n-j)t_j - 4jt_{n-j} + 4\sum_{j=0}^{n-1} b_j + 4nc_j. \tag{6.64}$$

Furthermore, one has

$$\frac{\partial\phi(\mathbf{c},\mathbf{b})}{\partial c_0} = -2nt_0 + 2\sum_{j=0}^{n-1} b_j + 2nc_0. \tag{6.65}$$

If n is even and j is odd, $j \in \{1,\ldots,n-1\}$, then, since $c_{n-j} = c_j$, we have

$$\begin{aligned} \frac{\partial\phi(\mathbf{c},\mathbf{b})}{\partial b_j} &= -2\left(2t_j + 4\sum_{k=0}^{(j-3)/2} t_{2k+1} + 2t_{n-j} + 4\sum_{k=0}^{(n-j-3)/2} t_{2k+1} - \right. \\ &\quad \left. -4\sum_{k=0}^{n/4-1} c_{2k+1} - 2nb_j\right) = \\ &= -2\left(2(t_j - c_j) + 4\sum_{k=0}^{(j-3)/2}(t_{2k+1} - c_{2k+1}) + 2(t_{n-j} - c_{n-j}) + \right. \\ &\quad \left. +4\sum_{k=0}^{(n-j-3)/2}(t_{2k+1} - c_{2k+1}) - 2nb_j\right). \end{aligned} \tag{6.66}$$

If both n and j are even, $j \in \{1,2,\ldots,n-1\} \setminus \{n/2\}$, then, by arguing analogously as in the previous case, we deduce

$$\begin{aligned} \frac{\partial\phi(\mathbf{c},\mathbf{b})}{\partial b_j} &= -2\left(2(t_j - c_j) + 4\sum_{k=1}^{j/2-1}(t_{2k} - c_{2k}) + 2(t_{n-j} - c_{n-j}) + \right. \\ &\quad \left. +4(t_0 - c_0) + 4\sum_{k=1}^{(n-j)/2-1}(t_{2k} - c_{2k}) - 2nb_j\right). \end{aligned} \tag{6.67}$$

Moreover, if n is even, then one has

$$\frac{\partial \phi(\mathbf{c},\mathbf{b})}{\partial b_0} = -2\left(t_0 - c_0 + 2\sum_{k=1}^{n/2-1}(t_{2k}-c_{2k}) - n b_0\right), \tag{6.68}$$

getting (6.81). Furthermore, for n even, we have

$$\frac{\partial \phi(\mathbf{c},\mathbf{b})}{\partial b_{n/2}} = -2\left(2(t_{n/2}-c_{n/2}) + 4\sum_{k=1}^{n/4-1}(t_{2k}-c_{2k}) - n b_{n/2}\right). \tag{6.69}$$

Now, if both n and j are odd, $j \in \{0,1,\ldots,n-1\}$, then, taking into account (6.77), we obtain

$$\begin{aligned}\frac{\partial \phi(\mathbf{c},\mathbf{b})}{\partial b_j} &= -2\left(2(t_j - c_j) + 4\sum_{k=0}^{(j-3)/2}(t_{2k+1}-c_{2k+1}) + 2(t_{n-j}-c_{n-j}) + \right. \\ &\quad \left. + 4\sum_{k=1}^{(n-j)/2-1}(t_{2k}-c_{2k}) + 2(t_0-c_0) - 2n b_j\right).\end{aligned} \tag{6.70}$$

If n is odd and j is even, $j \in \{0,1,\ldots,n-1\}$, then we have

$$\begin{aligned}\frac{\partial \phi(\mathbf{c},\mathbf{b})}{\partial b_j} &= -2\left(2(t_j - c_j) + 4\sum_{k=1}^{j/2-1}(t_{2k}-c_{2k}) + 2(t_{n-j}-c_{n-j}) + \right. \\ &\quad \left. + 4\sum_{k=1}^{(n-j-3)/2}(t_{2k+1}-c_{2k+1}) + 2(t_0-c_0) - 2n b_j\right).\end{aligned} \tag{6.71}$$

Finally, for n odd, one has

$$\frac{\partial \phi(\mathbf{c},\mathbf{b})}{\partial b_0} = -2\left(t_0 - c_0 + 2\sum_{k=0}^{(n-3)/2}(t_{2k+1}-c_{2k+1}) - n b_0\right). \tag{6.72}$$

It is not difficult to see that the function ϕ is convex. Therefore, ϕ has exactly one point of minimum. From this it follows that ϕ admits exactly one stationary point. Now we claim that this point satisfies

$$\sum_{k=0}^{n/4-1} b_{2k+1} = 0 \tag{6.73}$$

and

$$b_0 + 2\sum_{k=1}^{n/4-1} b_{2k} + b_{n/2} = 0 \tag{6.74}$$

when n is even, and

$$b_0 + 2\sum_{j=1}^{(n-1)/2} b_j = 0 \tag{6.75}$$

if n is odd, that is $B_n(T_n) \in \mathcal{B}_n$. From (6.73)-(6.75) and (6.64)-(6.65) we get (6.77)-(6.78). Furthermore, from (6.77)-(6.78) and (6.66)-(6.72) we obtain (6.79)-(6.85). Finally, (6.73)-(6.74) follow from (6.79)-(6.82), while (6.75) is a consequence of (6.83)-(6.85). □

Theorem 6.7.2. *Given $T_n \in \mathcal{T}_n$, one has*

$$V_n(T_n) = C_n(T_n) + B_n(T_n) + F_n(T_n) = \min_{V \in \mathcal{V}_n} \|V - T_n\|_F, \tag{6.76}$$

*where $C_n(T_n)$ =*circ$(\mathbf{c})$*, with*

$$c_j = \frac{(n-j)t_j + j t_{n-j}}{n}, \quad j \in \{1,2,\ldots,n-1\}; \tag{6.77}$$

$$c_0 = t_0, \tag{6.78}$$

*and $B_n(T_n)$ =*rcirc$(\mathbf{b})$*, where: for n even and $j \in \{1,2,\ldots,n-1\} \setminus \{n/2\}$,*

$$\begin{aligned} b_j &= \frac{1}{2n}\left(\frac{4j-2n}{n}(t_j - t_{n-j}) + 4\sum_{k=1}^{(j-3)/2}\frac{2k+1}{n}(t_{2k+1} - t_{n-2k-1}) + \right. \\ &+ \left. 4\sum_{k=1}^{(n-j-3)/2}\frac{2k+1}{n}(t_{2k+1} - t_{n-2k-1})\right), \quad j \text{ odd;} \end{aligned} \tag{6.79}$$

$$\begin{aligned} b_j &= \frac{1}{2n}\left(\frac{4j-2n}{n}(t_j - t_{n-j}) + 4\sum_{k=1}^{j/2-1}\frac{2k}{n}(t_{2k} - t_{n-2k}) + \right. \\ &+ \left. 4\sum_{k=1}^{(n-j)/2-1}\frac{2k}{n}(t_{2k} - t_{n-2k})\right), \quad j \text{ even;} \end{aligned} \tag{6.80}$$

for n even,

$$b_0 = \frac{2}{n}\left(\sum_{k=1}^{n/2-1}\frac{2k}{n}(t_{2k} - t_{n-2k})\right), \tag{6.81}$$

$$b_{n/2} = \frac{4}{n}\left(\sum_{k=1}^{n/4-1}\frac{2k}{n}(t_{2k} - t_{n-2k})\right); \tag{6.82}$$

for n odd and $j \in \{1,2,\ldots,n-1\}$,

$$\begin{aligned} b_j &= \frac{1}{2n}\left(\frac{4j-2n}{n}(t_j - t_{n-j}) + 4\sum_{k=0}^{(j-3)/2}\frac{2k+1}{n}(t_{2k+1} - t_{n-2k-1}) + \right. \\ &+ \left. 4\sum_{k=1}^{(n-j)/2-1}\frac{2k}{n}(t_{2k} - t_{n-2k})\right), \quad j \text{ odd;} \end{aligned} \tag{6.83}$$

$$\begin{aligned} b_j &= \frac{1}{2n}\left(\frac{4j-2n}{n}(t_j - t_{n-j}) + 4\sum_{k=1}^{j/2-1}\frac{2k}{n}(t_{2k} - t_{n-2k}) + \right. \\ &+ \left. 4\sum_{k=0}^{(n-j-3)/2}\frac{2k+1}{n}(t_{2k+1} - t_{n-2k-1})\right), \quad j \text{ even;} \end{aligned} \tag{6.84}$$

for n odd,

$$b_0 = \frac{2}{n}\left(\sum_{k=0}^{(n-3)/2}\frac{2k+1}{n}(t_{2k+1} - t_{n-2k-1})\right); \tag{6.85}$$

$$f_j = \frac{t_j - t_{n-j}}{n}, \quad j \in \{1,2,\ldots,n-1\}; \tag{6.86}$$

$$f_0 = 0. \tag{6.87}$$

Proof. Set

$$\widetilde{\phi}(\mathbf{c},\mathbf{b},\mathbf{f}) = \|T_n - \mathrm{circ}(\mathbf{c}) - \mathrm{rcirc}(\mathbf{b}) - \mathrm{rcirc}(\mathbf{f})\|_F^2$$

for each symmetric vector $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^n$ and for every asymmetric vector $\mathbf{f} \in \mathbb{R}^n$. By proceeding analogously as in (6.64)-(6.72) and taking into account the asymmetry of $\mathbf{f}$, we get that the derivatives

$$\frac{\partial \widetilde{\phi}(\mathbf{c},\mathbf{b},\mathbf{f})}{\partial c_j}, \quad \frac{\partial \widetilde{\phi}(\mathbf{c},\mathbf{b},\mathbf{f})}{\partial b_j}$$

have the same expressions as the respective derivatives

$$\frac{\partial \phi(\mathbf{c},\mathbf{b})}{\partial c_j}, \quad \frac{\partial \phi(\mathbf{c},\mathbf{b})}{\partial b_j}$$

in (6.64)-(6.72), $j = 0,\ 1,\ldots,n-1$. Furthermore, for any $n \in \mathbb{N}$ and $j \in \{1,2,\ldots,n-1\}$ we get

$$\frac{\partial \widetilde{\phi}(\mathbf{c},\mathbf{b},\mathbf{f})}{\partial f_j} = 4(n f_j - t_n + t_{n-j}). \tag{6.88}$$

Proceeding similarly as we dealt with the function ϕ in Theorem 6.7.1, it is not difficult to prove the convexity of the function $\widetilde{\phi}$. From this it follows that $\widetilde{\phi}$ has exactly one point of minimum, and hence $\widetilde{\phi}$ admits exactly one stationary point. By arguing analogously as in Theorem 6.7.1, it is possible to show that the same conditions as in (6.73)-(6.75) are satisfied, and the assertion of the theorem follows. □

Now we show how the approximation found in β-matrices allows to obtain also preconditioned linear systems with eigenvalues clustered around 1. For every $n \in \mathbb{N}$, set

$$\begin{aligned} \widehat{\mathcal{T}}_n \quad = \quad & \{t \in \mathcal{T}_n : \text{ there is a function } f(z) = \sum_{j=-\infty}^{+\infty} t_j z^j, \\ & \text{with } z \in \mathbb{C}, |z| = 1, \text{ and such that } \sum_{j=-\infty}^{+\infty} |t_j| < +\infty\}. \end{aligned} \tag{6.89}$$

Observe that any function defined by a power series as in the first line of (6.89) is real-valued, and the set of such functions satisfying the condition

$$\sum_{j=-\infty}^{+\infty} |t_j| < +\infty$$

is called *Wiener class* (see, e.g., [22], [44, §3]).

Given a function f belonging to the Wiener class and a matrix $T_n \in \widehat{\mathfrak{T}_n}$, $T_n(f) = (t_{k,j})_{k,j}$: $t_{k,j} = t_{|k-j|}$, $k, j \in \{0, 1, \ldots, n-1\}$, and $f(z) = \sum_{j=-\infty}^{+\infty} t_j z^j$, then we say that $T_n(f)$ is *generated by* f.

We will often use the following property of absolutely convergent series (see, e.g., [22, 43]).

Lemma 6.7.3. *Let* $\sum_{j=1}^{\infty} t_j$ *be an absolutely convergent series. Then, we get*

$$\lim_{n\to+\infty} \left[\frac{1}{n} \left(\sum_{k=1}^{n} k\,|t_k| + \sum_{k=\lceil (n+1)/2 \rceil}^{n} (n-k)\,|t_k| \right) \right] = 0.$$

Proof. Let $S = \sum_{j=1}^{\infty} |t_j|$. Choose arbitrarily $\varepsilon > 0$. By hypothesis, there is a positive integer n_0 with

$$\sum_{k=n_0+1}^{\infty} |t_k| \leq \frac{\varepsilon}{4}. \tag{6.90}$$

Let $n_1 = \max\left\{ \frac{2\,n_0\,S}{\varepsilon}, 2\,n_0 \right\}$. Taking into account (6.90), for every $n > n_1$ it is

$$\begin{aligned} 0 &\leq \frac{1}{n} \left(\sum_{k=1}^{n} k\,|t_k| + \sum_{k=\lceil (n+1)/2 \rceil}^{n} (n-k)\,|t_k| \right) = \\ &= \frac{1}{n} \sum_{k=1}^{n_0} k\,|t_k| + \frac{1}{n} \sum_{k=n_0+1}^{n} k\,|t_k| + \frac{1}{n} \sum_{k=\lceil (n+1)/2 \rceil}^{n} (n-k)\,|t_k| \leq \\ &\leq \frac{1}{n_1}\, n_0 \sum_{k=1}^{n_0} |t_k| + 2 \sum_{k=n_0+1}^{n} |t_k| \leq \frac{\varepsilon}{2\,n_0\,S}\, n_0\,S + 2\,\frac{\varepsilon}{4} = \varepsilon. \end{aligned}$$

So, the assertion follows. □

Theorem 6.7.4. *For* $n \in \mathbb{N}$, *given* $T_n(f) \in \widehat{\mathfrak{T}_n}$, *let* $C_n(f) = C_n(T_n(f))$, $B_n(f) = B_n(T_n(f))$, $F_n(f) = F_n(T_n(f))$ *be as in Theorem* 6.7.2, *and set* $V_n(f) = C_n(f) + B_n(f) + F_n(f)$. *Then, the following statements hold.*

6.7.4.1) *For every* $\varepsilon > 0$ *there is a positive integer* n_0, *such that for each* $n \geq n_0$ *and for every eigenvalue* $\lambda_j^{(V_n(f))}$ *of* $V_n(f)$, *it is*

$$\lambda_j^{(V_n(f))} \in [f_{\min} - \varepsilon, f_{\max} + \varepsilon], \quad j \in \{0, 1, \ldots, n-1\}, \tag{6.91}$$

where $f_{\min}$ *and* $f_{\max}$ *denote the minimum and the maximum value of* f, *respectively.*

6.7.4.2) *For every* $\varepsilon > 0$ *there are* k, $n_1 \in \mathbb{N}$ *such that for each* $n \geq n_1$ *the number of eigenvalues* $\lambda_j^{((V_n(f))^{-1}\,T_n(f))}$ *of* $V_n^{-1}(f)\,T_n(f)$ *such that* $|\lambda_j^{((V_n(f))^{-1}\,T_n(f))} - 1| > \varepsilon$ *is less than* k, *namely the spectrum of* $(V_n(f))^{-1}\,T_n(f)$ *is clustered around* 1.

Proof. We begin with proving 6.7.4.1). Let $G_n(f) = C_n(f) + B_n(f)$. Choose arbitrarily $\varepsilon > 0$. We denote by $\lambda_j^{(C_n(f))}$ (resp., $\lambda_j^{(B_n(f))}$, $\lambda_j^{(F_n(f))}$, $\lambda_j^{(G_n(f))}$) the generic j-th eigenvalue of $C_n(f)$ (resp., $B_n(f)$, $F_n(f)$, $G_n(f)$) in the order given by Theorem 6.4.3 (resp., Theorem 6.4.7, Proposition 6.5.2). First, we claim that

$$\lambda_j^{(G_n(f))} \in [f_{\min} - \varepsilon/2, f_{\max} + \varepsilon/2], \quad j \in \{0,1,\ldots,n-1\}. \tag{6.92}$$

To prove (6.92) it is enough to show that this property holds (in correspondence with $\varepsilon/4$) for each $\lambda_j^{(C_n(f))}$, $j = 0, 1, \ldots, n-1$, and that

$$\lambda_j^{(B_n(f))} \in [-\varepsilon/4, \varepsilon/4] \text{ for every } n \geq n_0 \text{ and } j \in \{0,1,\ldots,n-1\}. \tag{6.93}$$

Indeed, since $C_n(f), B_n(f) \in \mathcal{G}_n$, we have

$$\lambda_j^{(G_n(f))} = \lambda_j^{(C_n(f))} + \lambda_j^{(B_n(f))} \text{ for all } j \in \{0,1,\ldots,n-1\},$$

getting the claim.

Now we consider the case n odd. For every $j \in \{0,1,\ldots,n-1\}$, since $c_j = c_{n-j}$ and thanks to (6.77), one has

$$\begin{aligned}
\left|\lambda_j^{(C_n(f))}\right| &= \left|\sum_{h=0}^{n-1} c_h \cos(2\pi h j)\right| = \left|c_0 + 2\sum_{h=1}^{(n-1)/2} c_h \cos(2\pi h j)\right| = \\
&= \left| t_0 + 2\sum_{h=1}^{(n-1)/2} t_h \cos(2\pi h j) - \right. \\
&- \left. \sum_{h=1}^{(n-1)/2} \frac{h}{n} t_h \cos(2\pi h j) + \sum_{h=1}^{(n-1)/2} \frac{h}{n} t_{n-h} \cos(2\pi h j)\right| \leq \\
&\leq \sum_{h=-(n-1)/2}^{(n-1)/2} |t_h| \left(e^{\mathrm{i}\frac{2\pi j}{n}}\right)^h + \sum_{h=1}^{(n-1)/2} \frac{h}{n}|t_h| + \sum_{h=1}^{(n-1)/2} \frac{h}{n}|t_{n-h}| \leq \\
&\leq \sum_{h=-\infty}^{+\infty} |t_h| \left(e^{\mathrm{i}\frac{2\pi j}{n}}\right)^h + \sum_{h=1}^{(n-1)/2} \frac{h}{n}|t_h| + \sum_{h=(n+1)/2}^{n-1} \frac{n-h}{n}|t_h|.
\end{aligned} \tag{6.94}$$

Choose arbitrarily $\varepsilon > 0$. Note that the first addend of the last term in (6.94) tends to $f\left(e^{\mathrm{i}\frac{2\pi j}{n}}\right)$ as n tends to $+\infty$, and hence, without loss of generality, we can suppose that it belongs to the interval $[f_{\min} - \varepsilon/12, f_{\max} + \varepsilon/12]$ for n sufficiently large. By Lemma 6.7.3, it is

$$\lim_{n\to+\infty} \left(\sum_{h=1}^{(n-1)/2} \frac{h}{n}|t_h| + \sum_{h=(n+1)/2}^{n-1} \frac{n-h}{n}|t_h| \right) = 0.$$

When n is even, we get

$$\begin{aligned}
\left|\lambda_j^{(C_n(f))}\right| &= \left|\sum_{h=0}^{n-1} c_h \cos(2\pi h j)\right| = \left|c_0 + 2\sum_{h=1}^{n/2-1} c_h \cos(2\pi h j) + (-1)^{n/2} c_{n/2}\right| = \\
&= \left| t_0 + 2\sum_{h=1}^{n/2-1} \cos(2\pi h j)\, t_h + (-1)^{n/2} t_{n/2} - \right. \\
&- \left. \sum_{h=1}^{n/2-1} \frac{h}{n} t_h \cos(2\pi h j) + \sum_{h=1}^{n/2-1} \frac{h}{n} t_{n-h} \cos(2\pi h j)\right| \leq \\
&\leq \sum_{h=-n/2+1}^{n/2} |t_h| \left(e^{\mathrm{i}\frac{2\pi j}{n}}\right)^h + \sum_{h=1}^{n/2-1} \frac{h}{n}|t_h| + \sum_{h=1}^{n/2-1} \frac{h}{n}|t_{n-h}| \\
&\leq \sum_{h=-\infty}^{+\infty} |t_h| \left(e^{\mathrm{i}\frac{2\pi j}{n}}\right)^h + \sum_{h=1}^{n/2-1} \frac{h}{n}|t_h| + \sum_{h=n/2+1}^{n-1} \frac{n-h}{n}|t_h|.
\end{aligned}$$

Thus, it is possible to repeat the same argument used in the previous case, getting 6.7.4.1).

Now we turn to 6.7.4.2). From Theorem 6.4.7 we obtain

$$\lambda_j^{(B_n(f))} = \begin{cases} \displaystyle\sum_{h=0}^{n-1} b_h \cos\left(\frac{2\pi j h}{n}\right) & \text{if } j \leq n/2, \\ \\ \displaystyle -\sum_{h=0}^{n-1} b_h \cos\left(\frac{2\pi (n-j) h}{n}\right) & \text{if } j > n/2. \end{cases}$$

So, without loss of generality, it is enough to prove 6.7.4.2) for $j \leq n/2$.

We first consider the case when n is even. We get

$$\begin{aligned}
\left|\lambda_j^{(B_n(f))}\right| &\leq \left|\sum_{h=0}^{n-1} b_h \cos\left(\frac{2\pi j h}{n}\right)\right| \leq \sum_{h=0}^{n-1} |b_h| = \\
&= |b_0| + \sum_{h=1}^{n/4-1} |b_{2h}| + \sum_{h=n/4+1}^{n/2-1} |b_{2h}| + |b_{n/2}| + \sum_{h=0}^{n/2-1} |b_{2h+1}| = \qquad (6.95) \\
&= I_1 + I_2 + I_3 + I_4 + I_5.
\end{aligned}$$

So, in order to obtain 6.7.4.2), it is enough to prove that each addend of the last line of (6.95) tends to 0 as n tends to $+\infty$. We get:

$$\begin{aligned}
I_1 &= |b_0| \leq \sum_{k=1}^{n/2-1} \frac{4k}{n^2}|t_{2k}| + \sum_{k=1}^{n/2-1} \frac{4k}{n^2}|t_{n-2k}| = \qquad (6.96) \\
&= \sum_{k=1}^{n/2-1} \frac{4k}{n^2}|t_{2k}| + \sum_{k=1}^{n/2-1} \frac{2n-4k}{n^2}|t_{2k}| \leq \frac{4n-8}{n^2} \sum_{h=1}^{\infty} |t_h| \leq \frac{4S}{n},
\end{aligned}$$

where $S = \displaystyle\sum_{h=1}^{\infty} |t_h|$. From (6.96) it follows that $I_1 = |b_0|$ tends to 0 as n tends to $+\infty$. Analogously it is possible to check that $I_4 = |b_{n/2}|$ tends to 0 as n tends to $+\infty$.

Now we estimate the term $I_2 + I_3$. We first observe that

$$\begin{aligned} & \frac{1}{n^2}\sum_{h=1}^{n/4-1}(4h-n)(|t_{2h}|+|t_{n-2h}|)+\frac{1}{n^2}\sum_{h=n/4+1}^{n/2-1}(4h-n)(|t_{2h}|+|t_{n-2h}|)\leq \\ \leq\ & \frac{1}{n^2}\sum_{h=1}^{n/2-1}(4h-n)(|t_{2h}|+|t_{n-2h}|)\leq \\ \leq\ & \frac{1}{n^2}\sum_{h=1}^{n/2-1}4h|t_{2h}|+\frac{1}{n^2}\sum_{h=1}^{n/2-1}(4h-n)|t_{n-2h}|= \\ =\ & \frac{1}{n^2}\sum_{h=1}^{n/2-1}4h|t_{2h}|+\frac{1}{n^2}\sum_{h=1}^{n/2-1}(2n-4h)|t_{2h}|. \end{aligned} \tag{6.97}$$

Arguing analogously as in (6.96), it is possible to see that the quantities at the first hand of (6.97) tend to 0 as n tends to $+\infty$.

Furthermore, we have

$$\frac{2}{n}\sum_{h=1}^{n/2-1}\left(\sum_{k=1}^{h-1}\frac{2k}{n}|t_{2k}|\right)=\sum_{k=1}^{n/2-2}\frac{4k}{n^2}\left(\frac{n}{2}-2-k\right)|t_{2k}|\leq\sum_{k=1}^{n/2-2}\frac{2k}{n}|t_{2k}|, \tag{6.98}$$

$$\frac{2}{n}\sum_{h=1}^{n/2-1}\left(\sum_{k=1}^{h-1}\frac{2k}{n}|t_{n-2k}|\right)=\sum_{k=1}^{n/2-2}\frac{4k}{n^2}\left(\frac{n}{2}-2-k\right)|t_{n-2k}|\leq\sum_{k=2}^{n/2-1}\frac{2k}{n}|t_{2k}|, \tag{6.99}$$

$$\frac{2}{n}\sum_{h=1}^{n/2-1}\left(\sum_{k=1}^{n/2-h-1}\frac{2k}{n}|t_{2k}|\right)\leq\sum_{k=1}^{n/2-2}\frac{2k}{n}|t_{2k}|, \tag{6.100}$$

and

$$\begin{aligned} & \frac{2}{n}\sum_{h=1}^{n/2-1}\left(\sum_{k=1}^{n/2-h-1}\frac{2k}{n}|t_{n-2k}|\right)\leq\sum_{k=1}^{n/2-2}\frac{4k}{n^2}\left(\frac{n-2k-2}{2}\right)|t_{n-2k}|= \\ =\ & \sum_{k=2}^{n/2-1}\frac{(n-2k)(2k-2)}{n^2}|t_{2k}|\leq\sum_{k=2}^{n/2-1}\frac{2k}{n}|t_{2k}|. \end{aligned} \tag{6.101}$$

Summing up (6.97)-(6.101), from (6.80) we obtain

$$\begin{aligned} I_2+I_3 =\ & \sum_{h=1}^{n/4-1}|b_{2h}|+\sum_{h=n/4+1}^{n/2-1}|b_{2h}|\leq\frac{1}{n^2}\sum_{h=1}^{n/2-1}4h|t_{2h}|+\frac{1}{n^2}\sum_{h=1}^{n/2-1}4h|t_{n-2h}|+ \\ +\ & \sum_{k=1}^{n/2-2}\frac{4k}{n}|t_{2k}|+\sum_{k=2}^{n/2-1}\frac{4k}{n}|t_{2k}|. \end{aligned} \tag{6.102}$$

Thus, taking into account Lemma 6.7.3, it is possible to check that the terms at the right hand of (6.102) tend to 0 as n tends to $+\infty$.

Now we estimate the term I_5. One has

$$\begin{aligned} & \frac{2}{n}\sum_{h=0}^{n/2-1}\left(\sum_{k=0}^{h-1}\frac{2k+1}{n}|t_{2k+1}|\right)=\sum_{k=0}^{n/2}\frac{2k+1}{n}|t_{2k+1}|= \\ =\ & \sum_{k=0}^{n/2}\frac{2k}{n}|t_{2k+1}|+\sum_{k=1}^{n/2}\frac{1}{n}|t_{2k+1}|=J_1+J_2. \end{aligned}$$

Thanks to Lemma 6.7.3, it is possible to check that J_1 tends to 0 as n tends to $+\infty$. Moreover, we have

$$0 \leq J_2 \leq \frac{S}{n}, \tag{6.103}$$

and hence I_4 tends to 0 as n tends to $+\infty$. Analogously as in the previous case, it is possible to prove that

$$\begin{aligned} I_5 &= \sum_{k=0}^{n/2-1} |b_{2k+1}| \leq \frac{1}{n^2} \sum_{k=0}^{n/2-1} 2(2k+1)|t_{2k+1}| + \\ &+ \frac{2}{n} \sum_{k=1}^{n/2-2} \left(\frac{n}{2} - 1 - k\right) \left(\frac{2k+1}{n}\right) |t_{n-2k-1}| \leq \\ &\leq \sum_{k=1}^{n/2-2} \frac{2(2k+1)}{n} |t_{2k+1}| + \sum_{k=2}^{n/2-1} \frac{2(2k+1)}{n} |t_{2k+1}|. \end{aligned} \tag{6.104}$$

By virtue of Lemma 6.7.3 and (6.103), we get that I_5 tends to 0 as n tends to $+\infty$. Therefore, all addends of the right hand of (6.95) tend to 0 as n tends to $+\infty$. Thus, (6.93) follows from (6.95), (6.96), (6.102) and (6.104).

When n is odd, it is possible to proceed analogously as in previous case. This proves (6.92).

Now we claim that the eigenvalues of $F_n(f)$ lie between $-\varepsilon/2$ and $\varepsilon/2$ for n large enough. We have:

$$|\lambda_j^{F_n(f)}| = \left| \sum_{k=0}^{n-1} f_j \, \sin\left(\frac{2\pi k j}{n}\right) \right| \leq \sum_{k=0}^{n-1} |f_j| \leq \frac{1}{n} \sum_{k=0}^{n-1} |t_j| + \frac{1}{n} \sum_{k=0}^{n-1} |t_{n-j}|. \tag{6.105}$$

Since f belongs to the Wiener class, we get the claim.

Moreover, we observe that

$$G_n(f) + F_n(f) = Q_n \left(\Lambda^{(G_n(f))} + Y_n \Lambda^{(F_n(f))} Y_n^T\right) Q_n^T,$$

where

$$\begin{aligned} \Lambda^{(G_n(f))} &= \boldsymbol{\lambda}^{(G_n(f))} = (\lambda_0^{(G_n(f))} \lambda_1^{(G_n(f))} \cdots \lambda_{n-1}^{(G_n(f))})^T, \\ \Lambda^{(F_n(f))} &= \boldsymbol{\lambda}^{(F_n(f))} = (\lambda_0^{(F_n(f))} \lambda_1^{(F_n(f))} \cdots \lambda_{n-1}^{(F_n(f))})^T. \end{aligned}$$

Thus, the matrix $G_n(f)+F_n(f)$ is similar to $\Lambda^{(G_n(f))}+Y_n\Lambda^{(F_n(f))}Y_n^T$. Note that

$$Y_n\Lambda^{(F_n(f))}Y_n^T=\begin{pmatrix} 0 & 0 & \dots & 0 & \lambda_0^{(F_n(f))} \\ 0 & \dots & 0 & \lambda_1^{(F_n(f))} & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & \lambda_2^{(F_n(f))} & 0 & \dots & 0 \\ \lambda_1^{(F_n(f))} & 0 & \dots & 0 & 0 \end{pmatrix}.$$

Therefore, 6.7.4.1) follows from the Gerschgorin theorem (see, e.g., [72]).

Now we turn to 6.7.4.2), that is we prove that the spectrum of $(V_n(f))^{-1}T_n(f)$ is clustered around 1. Since $(V_n(f))^{-1}(T_n(f)-V_n(f))=(V_n(f))^{-1}T_n(f)-I_n$, where I_n is the identity matrix, it is enough to check that the eigenvalues of $(V_n(f))^{-1}(T_n(f)-V_n(f))$ are clustered around 0.

Choose arbitrarily $\varepsilon>0$. Since f belongs to the Wiener class, there exists a positive integer $n_0=n_0(\varepsilon)$ such that

$$\sum_{j=n_0+1}^{\infty}|t_j|\le\varepsilon.$$

Proceeding similarly as in the proof of [22, Theorem 3 (ii)], we get

$$T_n(f)-V_n(f)=T_n(f)-C_n(f)-B_n(f)-F_n(f)=W_n^{(n_0)}+Z_n^{(n_0)}+E_n^{(n_0)},$$

where $W_n^{(n_0)}$, $Z_n^{(n_0)}$, $E_n^{(n_0)}$ are suitable matrices such that $W_n^{(n_0)}$ and $Z_n^{(n_0)}$ agree with the $(n-n_0)\times(n-n_0)$ leading principal submatrices of $T_n(f)-C_n(f)$ and $B_n(f)+F_n(f)$, respectively. We have:

$$\begin{aligned} \operatorname{rank}(E_n^{(n_0)}) &\le 2n_0; \\ \|W_n^{(n_0)}\|_1 &\le \frac{2}{n}\sum_{k=1}^{n-n_0-1}k|t_{n-k}-t_k|\le\frac{2}{n}\sum_{k=1}^{n_0}k|t_k|+4\sum_{k=n_0+1}^{\infty}|t_k|; \\ \|Z_n^{(n_0)}\|_1 &\le \sum_{h=0}^{n-1}(|b_h|+|f_h|), \end{aligned} \tag{6.106}$$

where the symbol $\|\cdot\|_1$ denotes the 1-norm of the involved matrix. Let $n_1>n_0$ be a positive integer with

$$\frac{1}{n_1}\sum_{k=1}^{n_0}k|t_k|\le\varepsilon \text{ and } \sum_{h=0}^{n-1}(|b_h|+|f_h|)\le\varepsilon. \tag{6.107}$$

Note that such an n_1 does exist, thanks to Lemma 6.7.3 and since all terms of (6.95) and (6.105) tend to 0 as n tends to $+\infty$. From (6.106) and (6.107) it follows that

$$\|W_n^{(n_0)}+W_n^{(n_0)}\|_1\le\|W_n^{(n_0)}\|_1+\|Z_n^{(n_0)}\|_1\le 8\varepsilon. \tag{6.108}$$

From (6.108) and the Cauchy interlace theorem (see, e.g., [162]) we deduce that the eigenvalues of $T_n(f) - V_n(f)$ are clustered around 0, with the exception of at most $k = 2n_0$ of them. By the Courant-Fisher minimax characterization of the matrix $(V_n(f))^{-1}(T_n(f) - V_n(f))$ (see, e.g., [162]), we obtain

$$\lambda_j^{(V_n(f))^{-1}(T_n(f)-V_n(f))} \leq \frac{\lambda_j^{(T_n(f)-V_n(f))}}{f_{\min}} \tag{6.109}$$

for n large enough. From (6.109) we deduce that the spectrum of $(V_n(f))^{-1}(T_n(f) - V_n(f))$ is clustered around 0, namely for every $\varepsilon > 0$ there are k, $n_1 \in \mathbb{N}$ with the property that for each $\varepsilon > 0$ the number of eigenvalues $\lambda_j^{(V_n(f))^{-1}T_n(f)}$ such that $\left|\lambda_j^{(V_n(f))^{-1}T_n(f)} - 1\right| > \varepsilon$ is at most equal to k. □

Note that a similar result can be obtained by approximating $G_n(f) = C_n(f) + B_n(f)$ (see [28]).

6.8 Experimental results

In order to test the goodness of the proposed approximations, we have proceeded as follows: fixed the dimension n and the range of values which the involved Toeplitz matrices can assume, we have created 10000 different instances of Toeplitz symmetric matrices T_n, whose values have been randomly and uniformly chosen in the interior of the prefixed range. Moreover, we have computed the approximation $C_n(T_n)$ given in [45], the approximation $H_n(T_n)$ presented in [22] and the approximations $G_n(T_n)$ and $V_n(T_n)$ given in (6.54) and (6.76), respectively. Furthermore, we have computed the mean error in terms of difference between the matrix T_n and the preconditioning matrix evaluated with respect to the Frobenius norm. In Table 6.1 the considered range is $[0,1]$. In this case, as expected, $V_n(T_n)$ turns to be the best approximation, while $G_n(T_n)$ is the second best approximation in mean. In Table 6.2, the considered interval is $[-1,1]$, and the obtained results are analogous to the previous ones. In Table 6.3, to generate the first row of the Toeplitz symmetric matrix, we have proceeded as follows. We have taken the value of the first entry equal to 1. To determinate the value of the i-th entry, we have multiplied the value of the $i-1$-th entry by a random constant chosen uniformly in $[0.9,1]$. Such a choice allows to better simulate the Toeplitz matrices present in the blur operators. The behavior of the errors is similar to that of

	$\|T_n - C_n(T_n)\|_F$	$\|T_n - H_n(T_n)\|_F$	$\|T_n - G_n(T_n)\|_F$	$\|T_n - V_n(T_n)\|_F$
$n = 20$	3.1389	3.1156	3.0770	3.0532
$n = 25$	4.1076	4.0885	3.9591	3.9392
$n = 30$	4.8062	4.7903	4.7369	4.7207
$n = 35$	5.7528	5.7390	5.5989	5.5847
$n = 40$	6.4536	6.4416	6.3811	6.3689
$n = 45$	7.4243	7.4135	7.2649	7.2538
$n = 50$	8.1211	8.1114	8.0471	8.0373
$n = 100$	16.46786	16.46293	16.38939	16.38444
$n = 1000$	166.48101	166.48051	166.39821	166.39771

Table 6.1: Mean error obtained by the various approximations with respect to 10000 instances of randomly generated Toeplitz matrices T_n with entries in $[0,1]$.

the prevous cases. Moreover, from Tables 6.1-6.3 it is possible to see that, for large numbers, the approximations $C_n(T_n)$ and $H_n(T_n)$ give similar results, while the approximations $G_n(T_n)$ and $V_n(T_n)$. Furthermore, as seen in Table 6.4, for large numbers the approximation $G_n(T_n)$ is always better than the approximation $H_n(T_n)$. Since the multiplication of $V_n(T_n)$ by a vector needs three fast discrete transforms, while the multiplication of $V_n(T_n)$ by a vector requires only one fast discrete transform. Thus we deduce that, for n very large, $G_n(T_n)$ is the better solution in terms both of approximation and in computational costs.

	$\|T_n - C_n(T_n)\|_F$	$\|T_n - H_n(T_n)\|_F$	$\|T_n - G_n(T_n)\|_F$	$\|T_n - V_n(T_n)\|_F$
$n = 5$	0.73470	0.65623	0.65593	0.56618
$n = 10$	1.44816	1.40540	1.40531	1.36116
$n = 15$	2.42566	2.39475	2.28953	2.25668
$n = 20$	6.2564	6.2098	6.1313	6.0838
$n = 25$	8.2016	8.1633	7.8982	7.8584
$n = 30$	9.6160	9.5842	9.4776	9.4453
$n = 35$	11.517	11.489	11.210	11.182
$n = 40$	12.915	12.891	12.771	12.747
$n = 45$	14.835	14.813	14.521	14.499
$n = 50$	16.292	16.272	16.141	16.121
$n = 100$	32.92819	32.91833	32.76966	32.75976
$n = 1000$	332.72496	332.72396	332.56154	332.56054

Table 6.2: Mean error obtained by the various approximations with respect to 10000 instances of randomly generated Toeplitz matrices T_n with entries in $[-1, 1]$.

	$\|T_n - C_n(T_n)\|_F$	$\|T_n - H_n(T_n)\|_F$	$\|T_n - G_n(T_n)\|_F$	$\|T_n - V_n(T_n)\|_F$
$n = 5$	0.18725	0.16362	0.18190	0.15743
$n = 10$	0.71534	0.68775	0.67302	0.64363
$n = 15$	1.43778	1.41100	1.33331	1.30439
$n = 20$	2.28601	2.26095	2.10745	2.08025
$n = 25$	3.17788	3.15482	2.92053	2.89542
$n = 30$	4.07270	4.05158	3.73644	3.71341
$n = 35$	4.95798	4.93865	4.54353	4.52243
$n = 40$	5.79877	5.78109	5.31037	5.29105
$n = 45$	6.59117	6.57494	6.03320	6.01547
$n = 50$	7.30809	7.29317	6.68763	6.67133
$n = 100$	11.56697	11.55943	10.60308	10.59485
$n = 1000$	13.68293	13.68225	13.43137	13.43068

Table 6.3: Mean error obtained by the various approximations with respect to 10000 instances of randomly generated Toeplitz matrices T_n with entries in $[0, 1]$ in decreasing way.

	$range = [-1,1]$	$range = [-1,1], decreasing$
$n = 5$	4994	0
$n = 10$	5019	9992
$n = 15$	8989	10000
$n = 20$	8727	10000
$n = 25$	9794	10000
$n = 30$	9765	10000
$n = 35$	9973	10000
$n = 40$	9943	10000
$n = 45$	9993	10000
$n = 50$	9990	10000
$n = 100$	10000	10000
$n = 1000$	10000	10000

Table 6.4: Number of times in which the first proposed approximation gives better results than that in [22] with respect to 10000 instances of randomly generated Toeplitz matrices T_n with entries in $[0,1]$ in decreasing way.

Ringraziamenti

Questi tre anni sono volati.

Tra la ricerca, la pandemia, gli impegni vari, mi sembra ieri che ho scoperto di aver superato il test di ammissione per il dottorato e poter cosí intraprendere questo entusiasmante percorso. Ed ora eccoci qui, sono giunta al termine di questo viaggio e non posso non essere un pochino malinconica, non solo perché aver intrapreso questo percorso mi ha permesso di ampliarmi culturalmente sotto tantissimi aspetti (non solo puramente matematici), ma anche perché lascio una famiglia: il professor Ivan Gerace, il mio tutor, ed il professor Antonio Boccuto, che ha da sin dall'inizio collaborato con noi. Essi non sono stati dei semplici docenti, ma delle persone, che ognuna con il suo modo di essere, hanno saputo trasmettermi l'amore per la professione che svolgono. La stima nei confronti di una persona non la si chiede, né la si puó pretendere, ma la si guadagna di giorno in giorno, facendo bene e con amore, il proprio lavoro. Grazie per ció che avete fatto per me, io vi stimo tantissimo ed é proprio vero che spesso non sono i riconoscimenti che ti rendono un grande, ma sono le impronte che lasci nelle persone a cui insegni, con cui passi del tempo, con cui lavori. Ancora grazie di cuore per aver collaborato con me in questi tre anni e per ció che avete fatto per me.

Un ringraziamento speciale va alla mia famiglia, alle persone che mi amano, Massimiliano, Roberta, Giada, Natascia, Luca, che mi sono state vicino in momenti, soprattutto tu Massi, che senza che te lo chiedessi, lo hai fatto anche in quei frangenti in cui neanche io sarei voluta essere in compagnia di me stessa. Un grazie unico, va alla mia mamma, il merito se ho intrapreso questo dottorato é tantissimo anche suo, perché, come dice spesso, ″mi devi sempre dare retta!!″, in effetti é cosí, ma forse questa é una caratterista di tutte le mamme, non solo della mia, ci danno la vita e ce la migliorano ogni giorno con i loro consigli.

Infine un grazie va a me. Nonostante alcuni momenti non facili, ce l'ho fatta. Valentina hai stretto i denti, forse alcune volte anche contro le tue aspettative comunque ce l'hai fatta, hai tenuto talk, hai pubblicato articoli su importanti riviste, e tante altre piccole grandi soddisfazioni che terrai sempre nel tuo cuore e che ti hanno fatta crescere.

Ora sei un'altra, da quando hai iniziato sino ad ora, hai acquisito ancora piú sicurezza in te e nelle tue potenzialitá. Un percorso che, nonostante il Covid, che sicuramente ti ha tolto tanto, in merito ad opportunitá ed esperienze che avresti potuto intraprendere, ti ha permesso, ad ogni modo, di stringere amicizie ed inoltre hai potuto conoscere dei professionisti che sono e saranno sempre fonte di ispirazione sotto diversi punti di vista. Questo viaggio ti ha arricchito.
Ed infine una domanda a te stessa Valentina, la rifaresti questa scelta, riprenderesti il dottorato?
Altre duemila volte sí.
Grazie Dio per avermi concesso questa opportunitá.

Bibliography

[1] E. Aarts, J. Korst, Simulated Annealing and Boltzmann Machines. John Wiley & Sons, New York, 1989.

[2] J. E. Adams, *Interactions between color plane interpolation and other image processing functions in electronic photography*. In: Proceedings of SPIE, 2416 (1995), pp. 144–151.

[3] J. E. Adams and J. F. Hamilton, *Design of practical color filter array interpolation algorithms for digital cameras*. In: Proceedings of SPIE, 3028 (1997), pp. 117–125.

[4] J. Aelterman, B. Goossens, J. De Vylder, A. Pizurica and W. Philips, *Computationally Efficient Locally Adaptive Demosaicing of Color Filter Array Images Using the Dual-Tree Complex Wavelet Packet Transform.* PLoS ONE 8 (5) (2013), 1–18.

[5] D. Alleysson, S. Susstrunk and J. Herault, *Linear demosaicing inspired by the human visual system.* IEEE Trans. Image Process. 14 (4) (2005), 439-449.

[6] M. S. C. Almeida and L. B. Almeida, *Nonlinear Separation of Show-Through Image Mixtures Using a Physical Model Trained with ICA.* Signal Processing 92 (2012), 872–884.

[7] M. Andrecut, *Applications of left circulant matrices in signal and image processing.* Modern Physics Letters B 22 (2008), 231–241.

[8] G. Anescu, *A Heuristic Fast Gradient Descent Method for Unimodal Optimization,* J. Adv. Math. Computer Sci. 26 (5) (2018), 1–20.

[9] L. Armijo, *Minimization of Functions having Lipschitz Continuous First Partial Derivatives.* Pacific J. Math. 16 (1) (1966), 1–3.

[10] G. Aubert and P. Kornprobst, Mathematical Problems in Image Processing - Partial Differential Equations and the Calculus of Variations. Second Edition. Springer, New York, 2006.

[11] R. Badeau and R. Boyer, *Fast multilinear singular value decomposition for structured tensors.* SIAM J. Matrix Anal. Appl. 30 (2008), 1008–1021.

[12] M. Baek and J. Jeong, *Demosaicing algorithm using high-order interpolation with sobel operators*. In: Proceedings of the World Congress on Engineering, WCE 2014, Lecture Notes in Engineering and Computer Science 1 (2014), pp. 521-524.

[13] C. Bai, J. Li, Z. Lin, J. Yu and Y.-W. Chen, *Penrose demosaicking.* IEEE Trans. Image Process. 24 (2015), 1672–1684.

[14] A. K. Barros, *The Independence Assumption: Dependent Component Analysis.* In: M. Girolami (ed.), Advances in Independent Component Analysis, Springer, Berlin-Heidelberg-New York (2000), pp. 63–71.

[15] B. E. Bayer, Color imaging array, U.S. Patent 3 971 065, July 1976.

[16] A. Beck, Introduction to nonlinear optimization - Theory, Algorithms, and Applications with MATLAB. SIAM Mathematical Optimization, Philadelphia, PA, USA, 2014.

[17] L. Bedini, I. Gerace, E. Salerno and A. Tonazzini, *Models and Algorithms for Edge-Preserving Image Reconstruction.* Adv. Imaging Electron Physics 97 (1996), 86–189.

[18] L. Bedini, I. Gerace and A. Tonazzini, *A Deterministic Algorithm for Reconstruction of Images with Interacting Discontinuities.* Computer Vision Graphics and Image Processing (CVGIP): Graphical Models Images Processing 56 (1994), 109–123.

[19] L. Bedini, I. Gerace and A. Tonazzini, *A GNC algorithm for constrained image reconstruction with continuous-valued line processes*. Pattern Recognition Letters 15 (9) (1994), 907–918.

[20] X. Benlin, L. Fangfang, M. Xingliang and J. Huazhong, *Study on Independent Component Analysis Application in Classification and Change Detection of Multispectral Images.* The

International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 37 (2008), 871–876 .

[21] G. Bianco, F. Bruno, A. Tonazzini, E. Salerno and E. Conso, *Recto-verso registration, enhancement and segmentation of ancient documents.* In: Virtual Systems and Multimedia, 2009 (VSMM '09), 15th International Conference on (2009), pp. 131–136.

[22] D. Bini and P. Favati, *On a matrix algebra related to the discrete Hartley transform.* SIAM J. Matrix Anal. Appl. 14 (2) (1993), 500–507.

[23] A. Blake, *Comparison of the Efficiency of Deterministic and Stochastic Algorithms for Visual Reconstruction.* IEEE Trans. Pattern Anal. Machine Intell. 11 (1989), 2–12.

[24] A. Blake and A. Zisserman, Visual Reconstruction. MIT Press, Cambridge, MA, 1987.

[25] A. Boccuto and I. Gerace, *Image reconstruction with a non-parallelism constraint.* In: Proceedings of the International Workshop on Computational Intelligence for Multimedia Understanding, Reggio Calabria, Italy, 27-28 October 2016, IEEE Conference Publications (2016), pp. 1–5.

[26] A. Boccuto, I. Gerace and V. Giorgetti, *Minimum Amount of Text Overlapping in Document Separation.* http://viXra.org/abs/1805.0284 (2018).

[27] A. Boccuto, I. Gerace and V. Giorgetti, *A Blind Source Separation Technique for Document Restoration.* SIAM J. Imaging Sci. 12 (2) (2019), 1135–1162.

[28] A. Boccuto, I. Gerace, V. Giorgetti and F. Greco, Gamma-matrices: a new class of simultaneously diagonalizable matrices. https://arxiv.org/abs/2107.05890 (2021).

[29] A. Boccuto, I. Gerace and V. Giorgetti, *Blind Source Separation in Document Restoration: an Interference Level Estimation.* http://viXra.org/abs/2201.0050 (2022).

[30] A. Boccuto, I. Gerace and V. Giorgetti, *Image deblurring: a class of matrices approximating Toeplitz matrices* http://viXra.org/abs/2201.0155 (2022).

[31] A. Boccuto, I. Gerace, V. Giorgetti and M. Rinaldi, *A Fast Algorithm for the Demosaicing Problem Concerning the Bayer Pattern.* The Open Signal Processing Journal 6 (2019), 1–14.

[32] A. Boccuto, I. Gerace, V. Giorgetti and G. Valenti, *A Blind Source Separation Technique for Document Restoration Based on Edge Estimation.* http://viXra.org/abs/2201.0141 (2022).

[33] A. Boccuto, I. Gerace and F. Martinelli, *Half-Quadratic Image Restoration with a Non-Parallelism Constraint.* J. Math. Imaging Vis. 59 (2) (2017), 270–295.

[34] A. Boccuto, I. Gerace and P. Pucci, *Convex Approximation Technique for Interacting Line Elements Deblurring: A New Approach.* J. Math. Imaging Vis. 44 (2) (2012), 168–184.

[35] C. Bouman and K. Sauer, *A Generalized Gaussian Image Model for Edge-Preserving MAP Estimation.* IEEE Trans. Image Process. 2 (3) (1993), 296–310.

[36] A. Bose and K. Saha, Random circulant matrices. CRC Press, Taylor & Francis Group, Boca Raton-London-New York, 2019.

[37] R. Brent, Algorithms for Minimization without Derivatives. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1973.

[38] M. E. Brewster and R. Kannan, *Nonlinear Successive Over-Relaxation.* Numer. Math. 44 (2) (1984), 3019–3015.

[39] H. Brézis, Functional Analysis, Sobolev Spaces and Partial Differential Equations. Springer, New York-Dordrecht-Heidelberg-London, 2011.

[40] A. Buades, B. Coll, J. M. Morel and C. Sbert, *Self-similarity driven color demosaicking*. IEEE Trans. Image Process. 18 (6) (2009), 1192–1202.

[41] E. Carrasquinha, C. Amado, A. M. Pires and L. Oliveira: *Image reconstruction based on circulant matrices.* Signal Processing: Image Communication, 63 (2018), 72–80.

[42] T.-H. Chan, W.-K. Ma, C.-Y. Chi and Y. Wang, *A Convex Analysis Framework for Blind Separation of Non-Negative Sources.* IEEE Trans. Signal Process. 56 (10) (2008), 5120–5134.

[43] R. Chan, *The spectrum of a family of circulant preconditioned Toeplitz systems.* SIAM J. Numer. Anal. 26 (1989), 503–506.

[44] R. H. Chan, X. -Q. Jin and M.-C. Yeung, *The Spectra of Super-Optimal Circulant Preconditioned Toeplitz Systems.* SIAM J. Numer. Anal. 28 (3) (1991), 871–879.

[45] R. H. Chan and G. Strang, Toeplitz equations by conjugate gradients with circulant preconditioner. SIAM J. Sci. Stat. Comput. 10 (1989), 104–119.

[46] P. Charbonnier, L. Blanc-Féraud, G. Aubert and M. Barlaud, *Deterministic Edge-Preserving Regularization in Computed Imaging.* IEEE Trans. Image Process. 6 (1997), 298–311.

[47] K.-H. Chung and Y.-H. Chan, *Color demosaicing using variance of color differences*. IEEE Trans. Image Process. 15 (10) (2006), 2944–2955.

[48] K.-H. Chung and Y.-H. Chan, *Low-complexity color demosaicing algorithm based on integrated gradients*. J. Electron. Imaging 19 (2) (2010), 1-15.

[49] A. Cichocki and S.-I. Amari, Adaptive Blind Signal and Image Processing. John Wiley & Sons, Chichester, 2002.

[50] A. Cichocki, R. Zdunek and S.-I. Amari, *New Algorithms for Non-Negative Matrix Factorization in Applications to Blind Source Separation.* In: Proceedings of the 2006 IEEE International Conference Acoustics, Speech and Signal Processing, Toulouse, France (2006), pp. 1–4.

[51] B. Codenotti, I. Gerace and S. Vigna, *Hardness results and spectral techniques for combinatorial problems on circulant graphs.* Linear Algebra Appl. 285 (1998), 123–142.

[52] P. Comon, *Independent Component Analysis, A New Concept?* Signal Processing 36 (1994), 287–314.

[53] L. Condat, *A simple, fast and efficient approach to demoisaicking: Joint demosaicking and denoising.* In: Proceedings of IEEE International Conference on Image Processing, 50 (2010), pp. 905–908.

[54] P. J. Davis, Circulant matrices. John Wiley & Sons, New York, 1979.

[55] G. Demoment, *Image Reconstruction and Restoration: Overview of Common Estimation Structures and Problems.* IEEE Trans. Acoust., Speech, and Signal Processing 37 (1989), 2024–2036.

[56] M. P. Deshmukh and U. Bhosle, *A survey of image registration.* International Journal of Image Processing (IJIP) 5 (3) (2011), 245–269.

[57] W. Ding, L. Qi and Y. Wei, *Fast Hankel tensor-vector product and applications to exponential data fitting.* Numerical Linear Algebra Appl. 22 (2015), 814–832.

[58] M. Discepoli, I. Gerace, R. Mariani and A. Remigi, *A Spectral Technique to Solve the Chromatic Number Problem in Circulant Graphs.* In: Computational Science and its Applications - International Conference on Computational Science and Its Applications 2004, Lecture Notes in Computer Sciences 3045 (2004), pp. 745–754.

[59] M. Donatelli and S. Serra–Capizzano, *Antireflective boundary conditions for deblurring problems.* J. Electr. Comput. Eng., Art. ID 241467, 18 (2010).

[60] D. J. Evans and S. O. Okolie, *Circulant matrix methods for the numerical solution of partial differential equations by FFT convolutions.* J. Computational Appl. Math. 8 (4) (1982), 238–241.

[61] R. Farrahi Moghaddam and M. Cheriet, *Low quality Document Image Modeling and Enhancement.* Int. J. Document Analysis and Recognition 11 (2009), 183–201.

[62] R. Farrahi Moghaddam and M. Cheriet, *A Variational Approach to Degraded Document Enhancement*. IEEE Trans. Pattern Analysis and Machine Intelligence 32 (8) (2010), 1347–1361.

[63] S. Farsiu, M. Elad and P. Milanfar, *Multiframe demosaicing and super-resolution of color images.* IEEE Trans. Image Process. 15 (1) (2006), 141–159.

[64] S. Ferradans, M. Bertalmío and V. Caselles, *Geometry-based demosaicking*. IEEE Trans. Image Process. 18 (3) (2009), 665–670.

[65] W. T. Freeman, Median filter for reconstructing missing color samples, U.S. Patent 4, 774, 565, 1988.

[66] S. Geman and D. Geman, *Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images.* IEEE Trans. Pattern Anal. Machine Intell. 6 (1984), 721–740.

[67] D. Geman and G. Reynolds, *Constrained restoration and the recovery of discontinuities*. IEEE Trans. Pattern Analysis and Machine Intelligence 14 (3) (1992), 367–383.

[68] I. Gerace and F. Greco, *The Travelling Salesman Problem in symmetric circulant matrices with two stripes.* Mathematical Structures in Computer Science, Special Issue 1: In memory of Sauro Tulipani, 18 (2008), 165–175.

[69] I. Gerace, F. Martinelli and A. Tonazzini, *Restoration of Recto-Verso Archival Documents Through a Regularized Nonlinear Model.* In: Proceedings of 20th European Signal Processing Conference EUSIPCO (2012), pp. 1588–1592.

[70] I. Gerace, F. Martinelli and A. Tonazzini, *Demosaicing of noisy color images through edge-preserving regularization*. In: Proceeding of 2014 International Workshop on Computational Intelligence for Multimedia Understanding (IWCIM) (2014), pp. 1–5.

[71] I. Gerace, C. Palomba and A. Tonazzini, *An inpainting technique based on regularization to remove bleed-through from ancient documents.* In: 2016 International Workshop on Computational Intelligence for Multimedia Understanding (IWCIM) (2016), pp. 1–5.

[72] S. Gerschgorin, *Über die Abgrenzung der Eigenwerte einer Matrix.* Izv. Akad. Nauk. USSR Otd. Fiz.-Mat. Nauk 6 (1931), 749–754.

[73] A. E. Gilmour, *Circulant matrix methods for the numerical solution of partial differential equations by FFT convolutions.* Appl. Math. Modelling 12 (1988), 44-50.

[74] N. Gillis, *Successive Nonnegative Projection Algorithm for Robust Nonnegative Blind Source Separation.* SIAM J. Imaging Sci. 7 (2) (2014), 1420-1450.

[75] N. Gillis, *Sparse and unique nonnegative matrix factorization through data preprocessing.* J. Machine Learning Research 13 (2012), 3349–3386.

[76] J. W. Glotzbach, R. W. Schafer and K. Illgner, *A method of color filter array interpolation with alias cancellation properties*. In: Proceedings of IEEE Int. Conf. Image Processing, Thessaloniki, Greece, October 7-10 2001 (2001), pp. 141–144.

[77] L. Gottesfeld Brown, *A Survey of Image Registration Techniques.* ACM Comput. Surveys 24 (4) (1992), 325–376.

[78] B. K. Gunturk, Y. Altunbasak and R. Mersereau, *Color plane interpolation using alternating projections*. IEEE Trans. Image Process. 11 (2002), 997–1013.

[79] S. C. Gutekunst and D. P. Williamson, *Characterizing the Integrality Gap of the Subtour LP for the Circulant Traveling Salesman Problem.* SIAM J. Discrete Math. 33 (2019), 2452–2478.

[80] S. C. Gutekunst, B. Jin an D. P. Williamson, *The Two-Stripe Symmetric Circulant TSP is in P.* https://people.orie.cornell.edu/dpw/2stripe.pdf (2021).

[81] I. Györi and L. Horváth, *Utilization of Circulant Matrix Theory in Periodic Autonomous Difference Equations.* Int. J. Difference Equations 9 (2) (2014), 163–185.

[82] I. Györi and L. Horváth, *Existence of periodic solutions in a linear higher order system of difference equations.* Computers and Math. with Appl. 66 (11) (2013), 2239–2250.

[83] J. Hadamard, *Sur les problèmes aux derivées partielles et leur signification physique*. Annales Scientifiques de l'École Normale Supérieure 21 (3) (1902), 535–556.

[84] J. Hadamard, Lectures on Cauchy's Problem in Linear Partial Differential Equations. New Haven, Yale Univ. Press, Yale, 1923.

[85] J. F. Hamilton and J. E. Adams, Adaptive color plane interpolation in single sensor color electronic camera, U. S. Patent 5, 629–734, 1997.

[86] J. F. Hamilton and J. T. Compton, Processing color and panchromatic pixels, U.S. Patent 2007 0024879, 2007.

[87] E. Hazan, K. Y. Levy and S. Shalev-Shwartz. *On Graduated Optimization for Stochastic Non-Convex Problems.* In: Proceedings of the 33 rd International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W& CP 48 (2016), pp. 1–9.

[88] J. F. Henriques, Circulant structures in computer vision. Ph. D. thesis., Department of Electrical and Computer Engineering, Faculty of Science and Technology, Coimbra, 2015.

[89] R. H. Hibbard, Apparatus and method for adaptively interpolating a full color image utilizing luminance gradients, U.S. Patent 5 382 976, 1995.

[90] K. Hirakawa and T. W. Parks, *Adaptive homogeneity-directed demosaicing algorithm*. IEEE Trans. Image Process. 14 (3) (2005), 360–369.

[91] K. Hirakawa and T.W. Parks, *Joint Demosaicing and Denoising.* IEEE Trans. Image Process. 15 (8) (2006), 2146–2157.

[92] K. Hirakawa and P. Wolfe, *Spatio-spectral color filter array design for enhanced image fidelity.* In: 2007 IEEE International Conference on Image Processing (ICIP), (2007), pp. 81–84.

[93] A. Hore and D. Ziou, *An edge-sensing generic demosaicing algorithm with application to image resampling.* IEEE Trans. Image Process. 20 (11) (2011), 3136–3150.

[94] O. Hosam, *Side-informed image watermarking scheme based on dither modulation in the frequency domain*. The Open Signal Process. J. 5 (2013), pp. 1–6.

[95] S.-W. Huang, D.-L. Way and Z.-C. Shih, *Physical-based Model of Ink Diffusion in Chinese Paintings.* J.WGCG 10 (3) (2003), 520–527.

[96] A. Hyvärinen, *New Approximations of Differential Entropy for Independent Component Analysis and Projection Pursuit.* In: Advances in Neural Information Processing Systems, 10 (1998), pp. 273–279.

[97] A. Hyvärinen, *The Fixed-Point Algorithm and Maximum Likelihood Estimation for Independent Component Analysis.* Neural Process. Letters 10 (1) (1999), 1–5.

[98] A. Hyvärinen, *Fast and Robust Fixed-Point Algorithms for Independent Component Analysis,* IEEE Trans. Neural Networks, 626–634, 10 (3) (1999).

[99] A. Hyvärinen and E. Oja, *A Fast Fixed-Point Algorithm for Independent Component Analysis.* Neural Computation 9 (7) (1997), 1483–1492.

[100] P. Jarratt, *An iterative method for locating turning points.* The Computer Journal 10 (1) (1967), 82–84.

[101] G. M. Johnson and M. Fairchild, *A top down description of S-CIELAB and CIEDE2000*. Color Research and Application 28 (2003), 425–435.

[102] R. Kakarala and Z. Baharav, *Adaptive demosaicking with the principal vector method*. IEEE Trans. Consumer Electron. 48 (2002), 932–937.

[103] H. Kanemitsu, M. Miyakoshi and M. Shimbo, *Properties of Unimodal and multimodal Functions Defined by the Use of Local Minimal Value Set.* Electronics and Communications in Japan (Part III: Fundamental Electronic Science) 81 (1) (1998), 42–51.

[104] M. R. Khan, H. Imtiaz and M. K. Hasan, *Show-Through Correction in Scanned Images using Joint Histogram.* Signal, Image and Video Processing 4 (3) (2010), 337–351.

[105] A. Khaparde, M. Madhavilatha, M. B. L. Manasa and S. Pradeep Kumar, *FastICA Algorithm for the Separation of Mixed Images.* WSEAS Trans. Signal Process. 4 (5) (2008), 271–278.

[106] J. Kiefer, *Sequential Minimax Search for a Maximum*. Proc. Amer. Math. Soc. 4 (1953), 502–506.

[107] D. Kiku, Y. Monno, M. Tanaka and M. Okutomi, *Beyond color difference: Residual interpolation for color image demosaicking.* IEEE Trans. Image Process. 25 (2016), 1288–1300.

[108] R. Kimmel, *Demosaicing: image reconstruction from CCD samples.* IEEE Trans. Image Process. 8 (1999), 1221–1228.

[109] Kodak Lossless True Color Image Suiter, `http://r0k.us/graphics/kodak/`

[110] C. A. Laroche and M. A. Prescott, Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients. U. S. Patent 5, 373, 32, 1994.

[111] Y.J. Lei, W.R. Xu, Y. Lu, Y. R. Niu and X. M. Gu, *On the symmetric doubly stochastic inverse eigenvalue problem.* Linear Algebra Appl. 445 (2014), 181–205.

[112] X. Li, *Demosaicing by successive approximation.* IEEE Trans. Image Process. 14 (3) (2005), 370–379.

[113] N.-X. Lian, L. Chang, Y.-P. Tan and V. Zagorodnov, *Adaptive filtering for color filter array demosaicking.* IEEE Trans. Image Process. 16 (10) (2007), 2515–2525.

[114] Y. M. Lu, M. Karzand and M. Vetterli, *Demosaicking by alternating projections: Theory and fast one-step implementation.* IEEE Trans. Image Process. 19 (8) (2010), 2085–2098.

[115] W. Lu and Y.-P. Tan, "Color filter array demosaicing: New method and performance measures", *IEEE Trans. Image Process.*, vol. 12, 1194–1210, 2003.

[116] R. Lukac and K. N. Plataniotis, *Universal demosaicking for imaging pipelines with an RGB color filter array*. Pattern Recognition 38 (11) (2005), 2208–2212.

[117] J. Mairal, M. Elad and G. Sapiro, *Sparse representation for color image restoration.* IEEE Trans. Image Process. 17 (2008), 53–69.

[118] R. K. Malik and K. Solanki, *FastICA Based Blind Source Separation for CT Imaging Under Noise Conditions.* Int. J. Advances in Engineering and Technology, 5 (1) (2012), 47–55.

[119] F. Martinelli, E. Salerno, I. Gerace and A. Tonazzini, *Nonlinear model and constrained ML for removing back–to–front interferences from recto–verso documents.* Pattern Recognition 45 (2012), 596–605.

[120] D. Menon, S. Andriani and G. Calvagno, *Demosaicing with directional filtering and a posteriori decision.* IEEE Trans. Image Process. 16 (1) (2007), 132–141.

[121] D. Menon and G. Calvagno, *Demosaicing based on wavelet analysis of the luminance component.* In: Proceedings of the 2007 14th IEEE Int. Conf. Image Processing (ICIP), (2007), pp. 105-110.

[122] D. Menon and G. Calvagno, *Joint demosaicking and denoising with space-varying filters*. In: Proceedings of the 2009 16th IEEE International Conference on Image Processing (ICIP), (2009), pp. 477–480.

[123] D. Menon and G. Calvagno, *Regularization Approaches to Demosaicking.* IEEE Trans. Image Process. 18 (2009), 2209-2220.

[124] D. Menon and G. Calvagno, *Regularization approaches to demosaicking.* IEEE Trans. Image Process. 18 (2009), 2209–2220.

[125] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth and E. Teller, *Equations of State Calculations by Fast Computing Machines.* J. Chem. Phys. 21 (1953), 1087–1092.

[126] A. Moghadam, M. Aghagolzadeh, M. Kumar and R. Radha, *Compressive framework for demosaicing of natural images.*IEEE Trans. Image Process. 22 (2013), 2356–2371.

[127] H. Mobahi and J. W. Fisher, *A Theoretical Analysis of Optimization by Gaussian Continuation.* In: W.-K. Wong and D. Lowd (Eds.), Twenty–Ninth Conference on Artificial Intelligence of the Association for the Advancement of Artificial Intelligence (AAAI), Proceedings. Austin, Texas, USA, January 25-30, 2015 (2015), pp. 1205–1211.

[128] D. D. Muresan and T. W. Parks, *Demosaicing using optimal recovery.* IEEE Trans. Image Process. 14 (2) (2005), 267–278.

[129] M. Nikolova, *Markovian reconstruction using a GNC approach.* IEEE Trans. Image Process. 8 (9) (1999), 1204–1220.

[130] M. Nikolova, M. K. Ng and C.–P. Tam, *On ℓ_1 Data Fitting and Concave Regularization for Image Recovery*. SIAM J. Sci. Comput. 35 (1) (2013), A397–A430.

[131] M. Nikolova, M. K. Ng, S. Zhang and W.-K. Ching, *Efficient Reconstruction of Piecewise Constant Images Using Nonsmooth Nonconvex Minimization.* SIAM J. Imaging Sci. 1 (1) (2008), 2–25.

[132] B. Ophir and D. Malah, *Improved Cross-Talk Cancellation in Scanned Images by Adaptive Decorrelation.* In: Proceedings of 23rd IEEE Convention of Electrical and Electronics Engineers in Israel (2004), pages 4.

[133] B. Ophir and D. Malah, *Show-Through Cancellation in Scanned Images using Blind Source Separation Techniques.* In: Proceedings of IEEE International Conference on Image Processing, 3 (2007), pp. 233–236.

[134] W. S. B. Ouedraogo, A. Souloumiac, M. Jaidane and C. Jutten, *Non-Negative Blind Source Separation Algorithm Based on Minimum Aperture Simplicial Cone.* IEEE Trans. Signal Process. 62 (2) (2014), 376–389.

[135] D. Paliy, A. Foi, R. Bilcu and V. Katkovnik, *Denoising and interpolation of noisy Bayer data with adaptive cross-color filters.* In: Proceedings of SPIE-IS&T EI, VCIP, 2008, 13 pp.

[136] J. M. Papy, L. De Lauauer and S. Van Huffel, *Exponential data fitting using multilinear algebra: The single-channel and the multi-channel case.* Numerical Linear Algebra Appl. 12 (2005), 809–826.

[137] S.-C. Pei and I.-K. Tam, *Effective color interpolation in CCD color filter arrays using signal correlation*. IEEE Trans. Circuits Syst. Video Technol. 13 (2003), 503–513.

[138] L. Qi, *Hankel tensors: Associated Hankel matrices and Vandermonde decomposition.* Commun. Math. Sci. 13 (2015), 113–125.

[139] S. Ricciardi, A. Bonaldi, P. Natoli, G. Polenta, C. Baccigalupi, E. Salerno, K. Kayabol, L. Bedini and G. De Zotti, *Correlated Component Analysis for Diffuse Component Separation with Error Estimation on Simulated Plank Polarization Data.* Mon. Not. R. Astron. Soc. 406 (2010), 1644–1658.

[140] M. C. Robini and I. E. Magnin, *Optimization by Stochastic Continuation.* SIAM J. Imaging Sci. 3 (4) (2010), 1096–1121.

[141] K. Roth, *Scaling of Water Flow Through Porous Media and Soils.* European J. Soil Science 59 (2008), 125–130.

[142] R. Rowley-Brooke, F. Pitié and A. Kokaram, *A Ground Truth Bleed-Through Document Image Database.* In: P. Zaphiris, G. Buchanan, E. Rasmussen and F. Loizides (eds), Theory and Practice of Digital Libraries. (TPDL) 2012. Lecture Notes in Computer Science 7489 (2012), pp. 185–196.

[143] T. Saito and T. Komatsu, *Demosaicing approach based on extended color total-variation regularization*. In: Proceeding IEEE Int. Conf. Image Processing, 1 (2008), pp. 885–888.

[144] T. Sakamoto, C. Nakanishi and T. Hase, *Software pixel interpolation for digital style camera suitable for a 32-bit MCU.* IEEE Trans. Consum. Electron. 44 (4) (1998), 1342–1352.

[145] E. Salerno, F. Martinelli and A. Tonazzini, *Nonlinear Model Identification and See-Through Cancelation from Recto/Verso Data.* Int. J. Document Analysis and Recognition (IJDAR) 16 (2) (2013), 177–187.

[146] P. Savino, L. Bedini and A. Tonazzini, *Joint non-rigid registration and restoration of recto-verso ancient manuscripts.* In: 2016 International Workshop on Computational Intelligence for Multimedia Understanding (IWCIM) (2016), pp. 1–5.

[147] S. Serra-Capizzano and D. Sesana, *A note on the eigenvalues of g-circulants (and of g-Toeplitz, g-Hankel matrices).* Calcolo 51 (2014), 639–659.

[148] G. Sharma, *Show-Through Cancellation in Scans of Duplex Printed Documents.* IEEE Trans. Image Process. 10 (5) (2001), 736–754.

[149] J. V. Stone, *Independent Component Analysis: an introduction*. Trends in Cognitive Sciences 6 (2) (2002), 59–64.

[150] B. Tao, I. Tastl, T. Cooper, M. Blasgen, and E. Edwards, *Demosaicing using human visual properties and wavelet interpolation filtering.* In: Proceeding of Color Imaging Conference: Color Science, Systems, Applications, (1999), pp. 252–256.

[151] G. J. Tee, *Eigenvectors of block circulant and alternating circulant matrices.* New Zealand J. Math. 36 (2007), 195–211.

[152] A. Tonazzini and L. Bedini, *Restoration of Recto-Verso Colour Documents using Correlated Component Analysis.* EURASIP J. Adv. Signal Process. 58 (2013), 1–10.

[153] A. Tonazzini, L. Bedini, E. E. Kuruoglu and E. Salerno, *Blind Separation of Auto-Correlated Images from Noisy Mixtures using MRF Models.* In: Proceedings of the 4th International Symposium on Independent Component Analysis and Blind Source Separation, Nara, Japan (2003), pp. 675–680.

[154] A. Tonazzini, I. Gerace and F. Martinelli, *Multichannel Blind Separation and Deconvolution of Images for Document Analysis.* IEEE Trans. Image Process. 19 (4) (2010), 912–925.

[155] A. Tonazzini, I. Gerace and F. Martinelli, *Document Image Restoration and Analysis as Separation of Mixtures of Patterns: From Linear to Nonlinear Models.* In: B. K. Gunturk and X. Li (Eds.), Image Restoration - Fundamentals and Advances, CRC Press, Taylor & Francis, Boca Raton (2013), pp. 285–310.

[156] A. Tonazzini, E. Salerno and L. Bedini, *Fast Correction of Bleed-Through Distortion in Greyscale Documents by a Blind Source Separation Technique.* Int. J. Document Analysis 10 (1) (2007), 17–25.

[157] A. Tonazzini, P. Savino and E. Salerno, *A non-stationary density model to separate overlapped texts in degraded documents.* SIViP 9 (Suppl. 1) (2015), S155–S164.

[158] P.-S. Tsai, T. Acharya and A. K. Ray, *Adaptive fuzzy color interpolation.* J. Electron. Imaging 11 (2002), 293–305.

[159] S. Vavasis, *On the Complexity of Nonnegative Matrix Factorization.* SIAM J. Optimization 20 (3) (2009), 1364–1377.

[160] H. H. Vaziri, Y. Xiao, R. Islam and A. Nouri, *Numerical Modeling of Seepage-Induced Sand Production in Oil and Gas Reservoirs.* Journal of Petroleum Sci. Engineering 36 (2002), 71–86.

[161] J. A. Weldy, *Optimized design for a single-sensor color electronic camera system.* In: Proceeding of SPIE, 1071 (1988), pp. 300–307.

[162] J. Wilkinson, The algebraic eigenvalue problem. Clarendon Press, Oxford, 1965.

[163] G. Winkler, Image Analysis, Random Fields and Dynamic Monte Carlo Methods-A Mathematical Introduction. Springer-Verlag, Berlin-Heidelberg, 1995.

[164] C. Wolf, *Document Ink Bleed-Through Removal with Two Hidden Markov Random Fields and a Single Observation Field.* IEEE Trans Pattern Anal. Mach. Intell. 32 (3) (2010), 431–447.

[165] X. Wu, W. K. Choi and P. Bao, *Color restoration from digital camera data by pattern matching.* In: Proceeding of SPIE, 3018 (1997), pp. 12–17.

[166] X. Wu and N. Zhang, *Primary-consistent soft decision color demosaicing for digital cameras*. IEEE Trans. Image Process. 13 (9) (2004), 1263–1274.

[167] X. Wu and X. Zhang, *Joint Color Decrosstalk and Demosaicking for CFA Cameras.* IEEE Trans. Image Process. 19 (12) (2010), 3181–3189.

[168] L. Zhang and X. Wu, *Color demosaicking via directional linear minimum mean square-error estimation.* IEEE Trans. Image Process. 14 (12) (2005), 2167–2177.

[169] L. Zhang, X. Wu, A. Buades and X. Li, *Color Demosaicking by Local Directional Interpolation and Non-local Adaptive Thresholding.* J. of Electronic Imaging 20 (2), 023016 (2011).

[170] L. Zhang, X. Wu and D. Zhang, *Color reproduction from noisy CFA data of single sensor digital cameras.* IEEE Trans. Image Process. 16 (9) (2007), 2184–2197.

[171] Q. Zhang, Y. Sato, J.-Y Takahashi, K. Muraoka and N. Chiba, *Simple Cellular Automaton-based Simulation of Ink Behaviour and Its Application to Suibokuga-like 3D Rendering of Trees.* J. Visual. Comput. Animat. 10 (1999), 27–37.

[172] X. Zhang and B. Wandell, *A spatial extension of CIELAB for digital color–image reproduction*. J. of the Society for Information Display 5 (1) (1997), 61–63.

[173] B. Zitová and J. Flusser, *Image registration metods: A survey.* Image Vision Comput. 21 (2003), 977–1000.