University of Niš
Faculty of Natural Sciences and Mathematics
Department of Computer Science

Igor R. Stojanović

NON-ITERATIVE METHODS FOR DIGITAL IMAGE RESTORATION

Ph.D. Dissertation

Niš, November 2013

Univerzitet u Nišu
Prirodno-matematički fakultet
Departman za računarske nauke

Igor R. Stojanović

NEITERATIVNI METODI ZA DIGITALNU RESTAURACIJU SLIKA

Doktorska disertacija

Niš, Novembar 2013

Članovi Komisije

dr Predrag Stanimirović, mentor

redovni profesor, Prirodno-matematički fakultet, Univerzitet u Nišu

dr Gradimir Milovanović, član

redovni član SANU

dr Marko Miladinović, član

docent, Prirodno-matematički fakultet, Univerzitet u Nišu

Datum odbrane: ________________ .

Research in the field of image restoration belong to computer science and applied mathematics. Each image can be presented as the matrix. Thus creating the possibility that the processes of the image processing are presented with the appropriate mathematical models.

The goal of Ph.D. dissertation is to develop an efficient and reliable methods for digital image restoration using mathematical models to analyze the process of blurring. In this way, we will focus on methods to remove blur caused by uniform and nonuniform motion. They are especially important in applications related to the removal of blur from X-ray images, in ANPR (Automatic Number Plate Recognition) system, with images of barcodes, LCD TVs and monitors, and other areas. This topic is extensively treated in recent years, as evidenced by the large number of books, monographs, papers and computer implementation in the field.

I want to express my gratitude to my supervisor Prof. Dr. Predrag Stanimirović for his comprehensive support, desire and commitment during the preparation of the dissertation. I am very grateful for his support, the time spent with me, and motivation for research in the field of image restoration.

I want to express my gratitude to Academician Prof. Dr. Gradimir Milovanović and Prof. Dr. Marko Miladinović for their full and constructive cooperation.

Warmest and most sincere thanks to our relatives from Nis for the selfless support, motivation and inspiration definitely contributed to making this work.

Huge thanks to my parents and my brother for their generous support of any kind, moral, material, etc..

Finally I want to express my sincere gratitude to my wife Dragana and sons Nikola and Ognen, for generous support in all the joys and temptations which we have passed in the previous period.

To my three worlds:
Ognen, Nikola and Dragana

ПРИРОДНО - МАТЕМАТИЧКИ ФАКУЛТЕТ
НИШ

KEY WORDS DOCUMENTATION

Accession number, **ANO**:	
Identification number, **INO**:	
Document type, **DT**:	**monograph**
Type of record, **TR**:	**textual / graphic**
Contents code, **CC**:	**doctoral dissertation**
Author, **AU**:	**Igor R. Stojanović**
Mentor, **MN**:	**Predrag S. Stanimirović**
Title, **TI**:	**Non-iterative methods for digital image restoration**
Language of text, **LT**:	**English**
Language of abstract, **LA**:	**English**
Country of publication, **CP**:	**Serbia**
Locality of publication, **LP**:	**Serbia**
Publication year, **PY**:	**2013**
Publisher, **PB**:	**author's reprint**
Publication place, **PP**:	**Niš, Višegradska 33.**
Physical description, **PD**: (chapters/pages/ref./tables/pictures/graphs/appendixes)	**119 p. ; graphic representations**
Scientific field, **SF**:	**Computer science**
Scientific discipline, **SD**:	**Digital image processing**
Subject/Key words, **S/KW**:	**Image restoration, blur of image, Moore-Penrose inverse, matrix equation, least square solution, partitioning method**
UC	**005.922.52:004.932+7.02**
Holding data, **HD**:	**library**
Note, **N**:	
Abstract, **AB**:	The goal of this research is to develop an efficient and reliable methods for digital image restoration using mathematical models to analyze the process of blurring. In this way, we will focus on methods to remove blur caused by uniform and nonuniform motion. They are especially important in applications related to the removal of blur from X-ray images, in ANPR (Automatic Number Plate Recognition) system, with images of barcodes, LCD TVs and monitors, and other areas. This topic is extensively treated in recent years, as evidenced by the large number of books, monographs, papers and computer implementation in the field.
Accepted by the Scientific Board on, **ASB**:	**22.05.2013**
Defended on, **DE**:	
Defended Board, **DB**: President:	
Member:	
Member, Mentor:	

ПРИРОДНО - МАТЕМАТИЧКИ ФАКУЛТЕТ
НИШ

КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, **РБР**:	
Идентификациони број, **ИБР**:	
Тип документације, **ТД**:	монографска
Тип записа, **ТЗ**:	текстуални / графички
Врста рада, **ВР**:	докторска дисертација
Аутор, **АУ**:	Игор Р. Стојановић
Ментор, **МН**:	Предраг С. Станимировић
Наслов рада, **НР**:	Неитеративни методи за дигиталну рестаурацију слика
Језик публикације, **ЈП**:	енглески
Језик извода, **ЈИ**:	енглески
Земља публиковања, **ЗП**:	Србија
Уже географско подручје, **УГП**:	Србија
Година, **ГО**:	2013.
Издавач, **ИЗ**:	ауторски репринт
Место и адреса, **МА**:	Ниш, Вишеградска 33.
Физички опис рада, **ФО**: (поглавља/страна/ цитата/табела/слика/графика/прилога)	119 стр., граф. прикази
Научна област, **НО**:	Рачунарске науке
Научна дисциплина, **НД**:	Дигитална обрада слика
Предметна одредница/Кључне речи, **ПО**:	Рестаурација слика, замућење слике, Moore-Penrose инверз, матрична једначина, најмање квадратно решење, метод преграђивања
УДК	005.922.52:004.932+7.02
Чува се, **ЧУ**:	библиотека
Важна напомена, **ВН**:	
Извод, **ИЗ**:	Циљ овог истраживања је да се развију ефикасне и поуздане методе за рестаурацију слика помоћу математичког модела за анализу процеса замућења. На тај начин ћемо се фокусирати на методе за уклањање замућења услед униформног и неуниформног кретања. Ово је важно код апликацијама за уклањање замућивања са рендгенских слика, у системима за аутоматско препознавање регистарских таблица, код LCD телевизора и монитора и у другим областима. Ова тема се интензивно обрађује последњих година, о чему сведочи велики број књига, монографија, радова као и компјутерских имплементација из ове области.
Датум прихватања теме, **ДП**:	22.05.2013
Датум одбране, **ДО**:	
Чланови комисије, **КО**: Председник:	
Члан:	
Члан, ментор:	

Contents

Chapter 1

Introduction

1.1 The notion of Image Restoration

Recording images is a very frequent event in everyday human life. Due to imperfections in the imaging and capturing process, the recorded image inevitably represents a degraded version of the original scene. The question of removing these imperfections is crucial to many of image analysis and image processing tasks. There exists a wide range of different degradations that need to be taken into account, covering for instance noise, geometrical degradations, illumination and color imperfections, and blur. Image restoration methods are aimed for the reconstruction of the original image from a degraded model.

The field of image restoration has seen a tremendous growth in interest over the last two decades. There are many excellent overview articles, journal papers, and textbooks on the subject of image restoration and identification [2, 5, 6, 26, 32, 43, 44]. A number of various algorithms have been proposed and intensively studied for achieving a fast-recovered and high-resolution reconstructed images see, e.g. [49, 50]. The recovery of an original image from degraded observations is of crucial importance and finds application in several scientific areas including medical imaging and diagnosis, military surveillance, satellite and astronomical imaging, reconstruction of poor-quality family portraits, and remote sensing.

Blurring is a form of bandwidth reduction of an ideal image owing to the imperfect image formation process. It can be caused by a relative motion between the camera and the original scene, or by an optical system that is out of focus. When aerial photographs are produced for remote sensing purposes, blurs are conditioned by atmospheric turbulence, aberrations in the optical system, and relative motion between the camera and the background. Such blurring is not confined to optical images; for example, electron micrographs are corrupted by spherical aberrations of the electron lenses, and computed tomography scans suffer from X-ray scatter.

The field of *image restoration* (also called as *image deblurring* or *image deconvolution*) is concerned with the reconstruction or estimation of the uncorrupted image from a blurred one [6]. Fundamentally, it tries to perform an operation on the image that is the inverse of the imperfections in the image formation system. In the use of image restoration methods, the attributes of the degrading system are assumed to be known a priori.

In practical situations, sometimes may not be able to obtain this information directly from the image formation process. The objective of blur identification is to estimate the attributes of the real imaging system from the observed degraded image itself prior to the restoration process. The combination of image restoration and blur identification is often referred to as *blind image deconvolution* [49].

Blind deconvolution algorithm based on the total variational (TV) minimization method is extremely effective for recovering edges of images as well as some blurring functions, e.g., motion blur and out-of-focus blur [11]. In the paper [99] the authors present anisotropic regularization techniques to exploit the piecewise smoothness of the image and the point spread function (PSF) in order to mitigate the severe lack of information encountered in blind restoration of shift-invariantly and shift-variantly blurred images. These techniques are demonstrated on linear motion blur and out-of-focus blur. Edge preserving regularization methods, in the context of image restoration and denoising, are presented in [73].

Images are aimed to memorize useful information, but unfortunately the presence of the blur is unavoidable. Motion blur is the effect caused by relative motion between the camera and the scene during image exposure time. Restoration of motion-blurred images has been a fundamental problem in digital imaging for a long time. We assume that the blurring function acts as a convolution kernel or point-spread function $h(n_1, n_2)$ and the image restoration methods that are described here fall under the class of linear spatially invariant restoration filters. It is also assumed that the statistical properties defined by the mean and correlation functions of the image do not change spatially.

Under these conditions the restoration process can be carried out by means of a linear filter of which the point-spread function is spatially invariant, i.e., it is constant throughout the image. These modeling assumptions can be mathematically formulated as follows. If we denote by $f(n_1, n_2)$ the desired ideal spatially discrete image that does not contain any blur or noise, then the recorded image $g(n_1, n_2)$ is modeled by the convolution which is determined using the two-dimensional point spread function $h(n_1, n_2)$ [6]:

$$\begin{aligned} g(n_1,n_2) &= h(n_1,n_2) * f(n_1,n_2) \\ &= \sum_{k_1=0}^{N-1}\sum_{k_2=0}^{M-1} h(k_1,k_2) f(n_1-k_1, n_2-k_2). \end{aligned} \tag{1.1.1}$$

The symbol $*$ denotes the convolution operation.
The objective of the image restoration is to make an estimate $f(n_1, n_2)$ of the ideal image, under the assumption that only the degraded image $g(n_1, n_2)$ and the blurring function $h(n_1, n_2)$ are given.

The goal of the Ph.D. dissertation is to develop an efficient and reliable methods for digital image restoration or image deblurring using mathematical models to analyze the process of blurring. In this way, we will focus on methods to remove blur caused by uniform and nonuniform motion. They are particularly important in applications related to the removal of blur from X-ray images, in automated number plate recognition systems, with bar code images, LCD televisors and monitors and other areas. This topic is intensively research in recent years, as evidenced by a large number of books, monographs, papers and computer implementation from this field.

One of our main motivations for developing the methods for digital image restoration is applicability in everyday life. Blurring images can be appropriate for generating background effects and image shadows. In our days, creating motion blur in images is something that many image artists, mainly photographers use in order to capture feigned movement. Moreover, two dimensional filtering based on the separable motion blur is also useful for smoothing the effects of the staircase like effect, known as 'aliasing'. The separable anti-alias filtering procedure is efficient on smoothing edges of images and can also round out features to produce highlighting effects.

Also, the process of the de-identification is an interesting application of the motion blur. Advances in imaging devices and web technologies have made it easy to capture and share large amounts of video data over the internet. The effluence of privacy information becomes an important issue in both academia and industry. Examples include the Google Street View, EveryScape, public and private surveillance video, the collection and distribution of medical face databases [30]. De-identification is intended for the elimination of identification information from images and videos, prior to sharing of the data, while keeping as much information on the action and its context. Recognition and de-identification are opposites, the recognition making use of all possible features to identify an object while the de-identification trying to obfuscate the features to thwart recognition. De-identification should be resistant to recognition by humans and algorithms [1]. Three types of videos need de-identification to not compromise the privacy of individuals [1]:

- *Casual videos* that are captured for other purposes and get shared. For example the images used by Google StreetView, EveryScape, the cameras setup in public spaces that can be viewed over the internet, videos or photos on sharing sites, etc. There is no need to know the identity of individuals who appear in these videos. All individuals should be de-identified irrevocably and early, perhaps at the camera itself.

- *Public surveillance videos* come from cameras watching spaces such as airports, streets, stores, etc. These type of videos usually are displayed on public monitors and a recorded version may be accessible to many people. The types of actions performed by individuals in these videos is important, but not their identities. Consequently de-identification is necessary.

- *Private surveillance videos* come from cameras placed at the entrances of semi-private spaces like offices. These type of videos usually are with higher quality and are likely to have a more detailed view of the individuals. De-identification may not be essential, but could be suggested to take care of possible viewing by non-authorized people.

Also automatic license plate de-identification is an important application [21]. According the above it is important to develop the automated methods for de-identifying individuals or items without affecting the context of the action in the image or in the video. Motion blur can be used for de-identification in the images or in the videos. Because we know the PSF and if necessary we can use the image deblurring methods in the process of identification of the people or actions in the recorded video.

Another practical example where we can use the image deblurring methods are barcode character recognition. Barcodes can be found on numerous items, such as packaged food, books, newspapers and more. There are different ways of reading these bar codes. One way is to use dedicated barcode readers. The second option is to acquire an image of the barcode using the camera that is anyway built into the device and process the image in order to decode the barcode [98]. Reading barcodes by image processing is slower and less reliable than using dedicated barcode scanners, but in some cases they are better. For example, dedicated systems based on reflected laser light do not work for reading barcodes on a monitor screen.

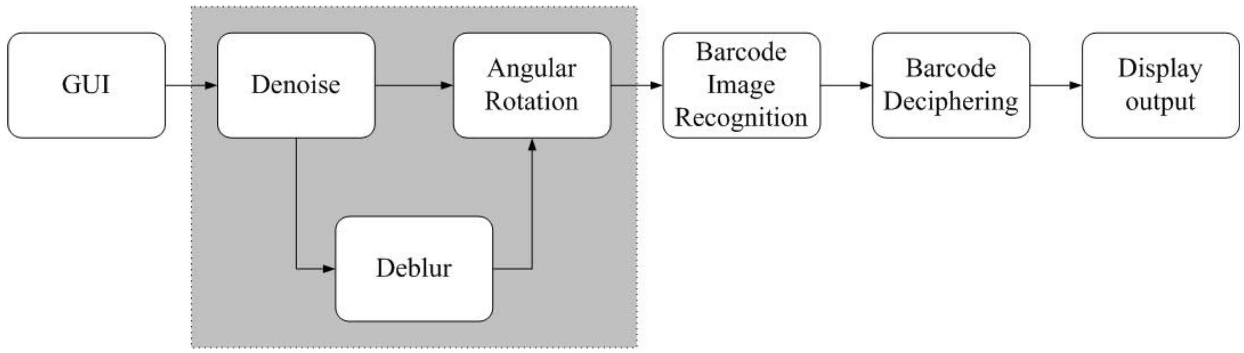


Figure 1.1.1: Barcode character recognition.

Barcode images acquired with cameras sometimes will not have the required quality to be recognized and decoded [98]. Reason for that could be motion blur. Our methods can be use for deblurring of the degraded images, and after that the deblurred image can be use in the image recognition process. The process of the recognition of the barcodes based on the image processing techniques is presented in the Figure 1.1.1. Some of the images may be too blurred and for successful deciphering must be implemented effective deblurred method.

The increasing consumption of liquid crystal displays (LCD) for computer monitors and home use has led to great interest in improvement of image quality especially when we have movements. If we compared with other types of displays as traditional cathode ray tube (CRT), plasma and projection displays, LCDs offer a lower cost, lower power consumption and higher resolution. Despite the great interest in solving the problem, LCDs still suffer from motion-blur the image. LCD motion blur is caused by two factors: the slow liquid crystal response time and the inherent sample-and-hold characteristic of LCD image formation.

With continuous improving of the physical properties of the liquid crystals and with using of method of overdrive is significantly mitigate response time [58]. By this way the problem of motion blur is reduced, but it's not eliminated. In [68, 69] are presented that when response time is $16ms$, 70% of the visible motion blur is part of the sample-and-hold property of the LCD display. Sample-and-hold motion blur will be present even with a zero response time. This blur is inherent to LCD image formation causes each pixel to emit approximately constant light through the frame period.

On Figure 1.1.2 is present the case when the object is moving horizontally with a constant velocity and a response time is zero [33]. Since the output at each pixel is held constant light through the frame period the displayed image does not match to the target trajectory predicted by Human Visual System (HVS). The dissimilarity between the eye tracking trajectory and the displayed data corresponds to the motion blur perceived by the human observer.

The methods for reduction of sample-and-hold LCD motion blur can be divided into several groups: back light flashing [24], frame rate doubling (black frame insertion [36], full frame insertion [51, 61]) and data pre-processing (motion-compensated inverse filters (MCIF) [47, 34]). The first method is used from Philips, it's consist of back light flashing at a faster rate than the frame period. With increasing the frame rate reduces the hold time and thus motion blur. MCIF engages estimating the motion and then apply a high pass filter.

Also we assume that the PSF(point spread function) is horizontal (if not, we can approximate the PSF with separable PSF and take its x component). The reason for this is because human eyes give much more consideration to horizontal motions from vertical motions [8].

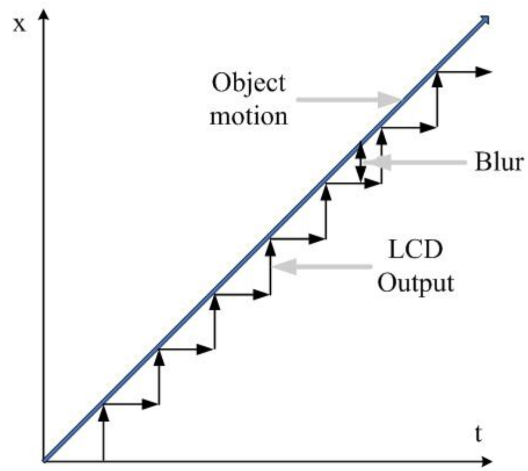


Figure 1.1.2: Position versus frame time.

In order to develop efficient algorithms and methods for reducing LCD motion blur is crucial to developing an accurate model of the occurrence of LCD blur. The model of LCD blurring is introduced in [9] and [68].

The process of degradation of the image is illustrated in Figure 1.1.3, where dynamic discrete content $I_d(x, y, t)$ is shown on the LCD display as $I_s(x, y, t)$. Firstly the image is degraded from the LCD display device, more accurate by sample-hold feature of the LCD device. After that the human visual system (HVS) will be degraded the displayed image and the perceived image is $I_o(x, y, t)$. The eye tracking and low pass filter (LPF) formed HVS.

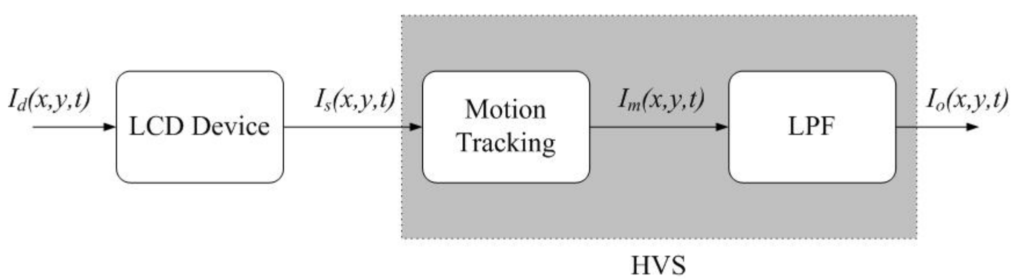


Figure 1.1.3: Process of the perception chain.

Our methods for image restoration can be used as pre-processing technique for reducing of the LCD motion blur. In this approach the signal is pre-processing before it is sent to the display. This means that the frame sampled at time t is $I_c(x, y, t)$ have to process with the method for image deblurring and we get the signal $I_d(x, y, t)$, that is input signal on Figure 1.1.3.

1.2 Organization of the Ph.D. dissertation

Generally, the Ph.D. dissertation is divided on three main parts: Chapter 2, Chapter 3 and Chapter 4. The first one is devoted to the modeling of the process of image formation and presentation of the standard methods of image restoration. The second one deals with the definition and description of the new non-iterative methods for image restoration. And finally, the third one presents experimental results and comparative analysis when we use the new non-iterative methods and standard methods for image restoration.

In the next chapter two commonly used filters for reconstructions of blurred image, namely Wiener filter and the constrained least-squares filter [6] are presented. After them, iterative nonlinear method for image restoration based on the Lucy-Richardson algorithm [26, 27] is shown. Also for image reconstruction we can use as well as the symmetric minimal rank (SMRS) solution of the inverse matrix problem [96]. The restoration methods based on moments (the Fourier and the Haar basis) [42] which are also used are described in this chapter. End of this chapter is devoted to the Truncated Singular Value Decomposition (TSVD) and Tikhonov (TIK) restoration methods [32].

The Chapter 3 consists of the five new non-iterative methods:

- The first method is a direct method for removing uniform linear motion blur from images. The method is based on a straightforward construction of the Moore-Penrose inverse of the blurring matrix for a given mathematical model. The computational load of the method is decreased significantly with respect to other competitive methods, while the resolution of the restored images remains at a very high level.

- The second method is based on an application of the partitioning method for determination of the Moore-Penrose inverse of a matrix augmented by a block-column matrix of arbitrary size. The adaptation of the partitioning method is applicable in the image restoration. The main contribution of the introduced method is a significant reduction in computational time required to calculate the Moore–Penrose inverse of a blurring matrix compared to other known methods for the pseudoinverse computation. The resolution of the restored image remains at a very high level.

- The next method generalizes image restoration algorithms which are based on the Moore–Penrose solution of certain matrix equations that define the linear motion blur. Our approach is based on the usage of least squares solutions of these matrix equations, wherein an arbitrary matrix of appropriate dimensions is included besides the Moore-Penrose inverse. In addition, the method is a useful tool for improving results obtained by other image restoration methods. Towards that direction, we investigate the case where the arbitrary matrix is replaced by the matrix obtained by the Haar basis reconstructed image. The method has been tested by reconstructing an image after the removal of blur caused by the uniform linear motion and filtering the noise that is corrupted with the image pixels. Quality of the restoration is observable by a human eye.

- The following method for reconstruction of blurred images damaged by a separable motion blur can be used after the application of currently developed image restoration algorithms. Our approach is based on the usage of least squares solutions of certain matrix equations which define the separable motion blur. The method uses appropriately selected matrices besides the Moore-Penrose inverse. The method is tested by reconstructing a set of images after the removal of blur caused by uniform and separable motion.

- Previously performed analyzes have confirmed that the method proposed in [83], can be used as a useful tool for improving restorations obtained by other image restoration methods. Continuing investigations in that direction, we investigate the case where arbitrary matrix is replaced by the matrix obtained by the Tikhonov regularization method or by the Truncated Singular Value decomposition method.

The Chapter 4 is devoted to experimental results and application of methods in some real cases. For comparison of the image restoration methods we use the following criteria: Improvement in Signal to Noise Ratio (ISNR) and Peak Signal to Noise Ratio (PSNR) [6, 26]. The methods are tested on standard images from `MATLAB` such as Lena, Balrbara, Cameraman etc., and real cases images: X-ray images and images from automated number plate recognition systems. The new non-iterative methods for digital image restoration improve the performance in terms of the quality of the restored images than the standard built-in image restoration methods in the software package `MATLAB`. Also, the new methods reduces the time required to complete the restoration process compared to conventional methods.

In the final chapter the conclusions with regard to the results obtained in the Ph.D. dissertation are presented. The following are comments on the results, their scientific and practical significance. At the end of this chapter a brief overview of the ideas for future work and possible further research in this and related areas is presented.

Chapter 2

Standard methods for image restoration

This chapter presents the modeling of the process of image formation and commonly used standard methods for image restoration. These methods are implemented as standard functions embedded in the software package `MATLAB`. Two commonly used filters for reconstructions of blurred image are from the collection of least-squares filters, namely Wiener filter and the constrained least-squares filter [6]. One iterative nonlinear method for image restoration is based on the Lucy-Richardson algorithm [26, 27].

Also for image reconstruction we can use the method based on the usage of symmetric minimal rank solution (SMRS) of the matrix equation $AX = B$ from [96]. Our motivation to use the symmetric minimal rank solution comes from the fact that in [96] is given a solution to the matrix A of the matrix equation $AX = B$, which is of the same form with the matrix equation of the image formation process. The problem of solution of the matrix equation $AX = B$ is one of the topics of especially active study in the computational mathematics, and has been broadly useful in various areas.

Short overview of three methods for the image restoration, SMRS solution of the matrix equation which models the image blurring process, the restoration methods based on moments, TSVD method and the TIK method are described in the next subsections. These methods will be used for comparison with the new restoration methods presented in the next chapter.

2.1 Modeling of the process of image formation

When we use the camera, we want the recorded image to be a good interpretation of the scene, but each image is more or less blurred. We assume that the blurring function acts as a convolution kernel or point-spread function $h(n_1, n_2)$ and the image restoration methods that are described here fall under the class of linear spatially invariant restoration filters. It is also assumed that the statistical properties (mean and correlation function) of the image do not change spatially. Under these conditions the restoration process can be carried out by means of a linear filter of which the point-spread function (PSF) is spatially invariant.

These modeling assumptions can be mathematically formulated as follows. If we denote by $f(n_1, n_2)$ the desired ideal spatially discrete image that does not contain any blur or noise, then

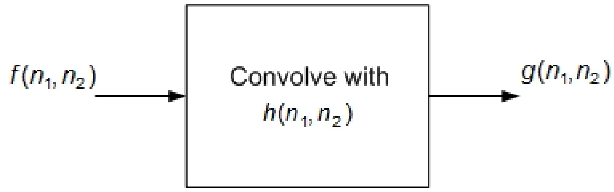


Figure 2.1.1: Model of the image formation in the spatial domain.

the recorded image is modeled as [6] (see also Figure 2.1.1):

$$\begin{aligned} g(n_1, n_2) &= h(n_1, n_2) * f(n_1, n_2) \\ &= \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{M-1} h(k_1, k_2) f(n_1 - k_1, n_2 - k_2). \end{aligned} \tag{2.1.1}$$

The objective of the image restoration is to make an estimate of the ideal image $f(n_1, n_2)$, under the assumption that only the degraded image $g(n_1, n_2)$ and the blurring function $h(n_1, n_2)$ are given.

An alternative way of describing (2.1.1) is through its spectral equivalence. By applying discrete Fourier transforms to (2.1.1), we obtain the following representation (see also Figure 2.1.2):

$$G(u, v) = H(u, v) F(u, v), \tag{2.1.2}$$

where (u, v) are the spatial frequency coordinates and capitals represent Fourier transforms. Either Figure 2.1.1 or Figure 2.1.2 can be used for developing restoration algorithms.

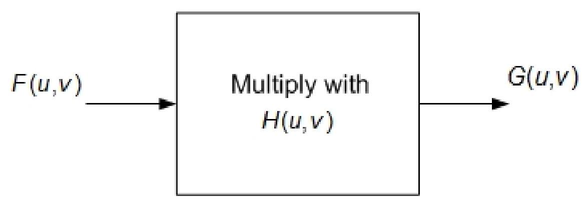


Figure 2.1.2: Model of the image formation in the Fourier domain.

Since we are examining the process of blurring as a process that is spatially independent, it means that the image is blurred in the same way at each spatial location. PSFs that do not follow these assumptions are rotating blurring as spinning wheel or local blur as the person is unfocused, while the background is focused. Modeling, identification and restoration of images degraded by spatially variable blurring is still unsolved problem [5, 6].

As a result of imperfections in the process of image formation, they are modeled as passive operations data, i.e. not absorb or generate energy. Consequently the impulse response of spatially continuous blur satisfies the relation:

$$\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} h(x, y) dx dy = 1. \tag{2.1.3}$$

The discrete representation of the relation (2.1.3) becomes:

$$\sum_{n_1=0}^{N-1} \sum_{n_2=0}^{M-1} h(n_1, n_2) = 1. \tag{2.1.4}$$

In the next section we present impulse responses encountered in real situations. We can distinguish different types of the motion blurring and all as a result of the relative movement of the device for recording and scene. It can be in the form of translation, rotation, sudden change in size of objects or a combination of them. For us it is important and we will consider the case of a global translation.

When the scene which you are shooting moves relative to the camera with constant speed v in horizontal direction during the interval of exposure $[0, t]$, degradation is one-dimensional. If L is indicated the length of the blur, then L and impulse response or PSF h for linear motion blur are linked by the relation:

$$L = v \times t, \tag{2.1.5}$$

$$h(n_1, n_2, L) = \begin{cases} \dfrac{1}{L} & \text{for} \quad n_1 = 0, \quad 0 \leq n_2 \leq L - 1, \\ 0 & \text{elsewhere.} \end{cases} \tag{2.1.6}$$

Knowing the physical process that causes blurring allows an explicit formulation of the impulse response. In this case, the elements of the array of impulse responses are presented with precise mathematical terms [32]. For example impulse response in case the blur is due of unfocused (out of focus) is given by:

$$h(n_1, n_2, R) = \begin{cases} \dfrac{1}{\pi R^2} & \text{for} \quad (n_1 - k)^2 + (n_2 - l)^2 \leq R^2, \\ 0 & \text{elsewhere,} \end{cases} \tag{2.1.7}$$

where (k, l) is the center of the impulse response, and R is the radius of the blurring.

Impulse response of the blurring caused by atmospheric turbulence can be described by two-dimensional Gaussian function [32]:

$$h(n_1, n_2) = \exp\left(-\frac{1}{2} \begin{bmatrix} i - k \\ j - l \end{bmatrix}^T \begin{bmatrix} s_1^2 & \rho^2 \\ \rho^2 & s_2^2 \end{bmatrix}^{-1} \begin{bmatrix} i - k \\ j - l \end{bmatrix} \right), \tag{2.1.8}$$

where the parameters s_1, s_2, and ρ determine the width and orientation of the impulse response, which is centered at element (k, l).

If the PSF of the linear restoration filter is designed and marked with $h_r(n_1, n_2)$, then the restored image is given by:

$$\begin{aligned} \widetilde{f}(n_1, n_2) &= h_r(n_1, n_2) * g(n_1, n_2) \\ &= \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{M-1} h_r(k_1, k_2) g(n_1 - k_1, n_2 - k_2). \end{aligned} \tag{2.1.9}$$

Relation (2.1.9) in the spectral domain transforms into:

$$\widetilde{F}(u, v) = H_r(u, v) G(u, v). \tag{2.1.10}$$

The goal is to design an appropriate restoration filter with $h_r(n_1, n_2)$ or $H_r(u, v)$ for their application in relations (2.1.9) or (2.1.10). Filters designed with the criterion of least squares, which are called Least Squares Filters, are commonly used for the restoration of images. From this collection the interesting cases for us are: Wiener filter and constrained least-squares filter.

2.2 The Wiener Filter

Wiener filter (WF shortly) is a linear and space invariant filter in which PSF h_r is selected to minimize the Mean Squared Error (MSE) between the ideal and the restored image. MSE is given by the following relation [6]:

$$MSE \approx \frac{1}{NM} \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{M-1} (f(n_1, n_2) - \widetilde{f}(n_1, n_2))^2, \tag{2.2.1}$$

where $\widetilde{f}(n_1, n_2)$ is given in relation (2.1.9).

The solution of this minimization problem is known as the Wiener filter and the easiest way is to define the frequency domain:

$$\begin{aligned} \widetilde{F}(u,v) &= \left[\frac{H^*(u,v) S_f(u,v)}{S_f(u,v)|H(u,v)|^2 + S_\eta(u,v)} \right] G(u,v) \\ &= \left[\frac{H^*(u,v)}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)} \right] G(u,v) \\ &= \left[\frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)} \right] G(u,v). \end{aligned} \tag{2.2.2}$$

Here is used the fact that multiplying a complex variable with its conjugate value is equal to the square of the magnitude of the complex variable. Filter that includes what is shown in brackets on the last line of the relation (2.2.2) is called the Wiener filter or filter with minimum MSE:

$$H_{wiener}(u,v) = \frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)}, \tag{2.2.3}$$

where:
- $H(u,v)$ is the blurring function in spectral domain,
- $|H(u,v)|^2 = H^*(u,v)H(u,v)$, $H^*(u,v)$ is the complex conjugate of $H(u,v)$,
- $S_\eta(u,v)$ is the power spectrum of the noise, and
- $S_f(u,v)$ is the power spectrum of the ideal image.

To obtain the restored image in the original spatial domain is necessary on the outcome in frequency domain $\widetilde{F}(u,v)$ obtained by relation (2.2.2) to make the inverse Fourier transform.

In the absence of noise we have $S_\eta(u,v) = 0$, so that Wiener-filter approximates the inverse filter:

$$H_{wiener}(u,v)|_{S_\eta(u,v)\to 0} = \begin{cases} \dfrac{1}{H(u,v)} & \text{for} \quad H(u,v) \neq 0, \\ 0 \quad \text{for} & H(u,v) = 0. \end{cases} \tag{2.2.4}$$

In many situations where an recorded image is noisy, Wiener filter is a compromise between the inverse filtering restoration and suppression of noise for those frequencies where $H(u,v) \to 0$. Key factors in this compromise are the power spectra of the ideal image and the noise. For frequencies where $S_\eta(u,v) \ll S_f(u,v)$, the Wiener filter approaching the inverse filter, while the frequencies where $S_\eta(u,v) \gg S_f(u,v)$ the Wiener filter acts as a filter that do not miss frequencies i.e. $H_{wiener}(u,v) \to 0$.

2.3 The constrained least-squares filter

Wiener filter presented in the previous section has some disadvantages. One of the disadvantages is that the power spectra of the original image and the noise must be known. The method presented in this section requires knowledge of the mean and variance of noise. Since these two parameters can usually be calculated from the given degraded image and this is an important advantage of this method [26].

Another difference is that the Wiener filter is based on the minimization of statistical criteria and as such it is optimal in an average sense. The method presented here has the noteworthy feature that gives optimal result for each image to which it is applied. Of course you need to take into account that these optimal criteria that are met in a theoretical point of view, not associated with the dynamics of visual perception. The choice that one algorithm or method is better than another, almost always determined by the obtained visual quality of the received images.

If restoration is good, blurred version of the restored image should be approximately equal to the recorded degraded image:

$$h(n_1,n_2) * \widetilde{f}(n_1,n_2) \approx g(n_1,n_2). \tag{2.3.5}$$

More reasonable expectation is that restored image to satisfy the relation:

$$\begin{aligned} \left\| g(n_1,n_2) - h\left(n_1, n_2 * \widetilde{f}(n_1,n_2)\right)\right\|^2 &= \frac{1}{NM}\sum_{k_1=0}^{N-1}\sum_{k_2=0}^{M-1}\left(g(k_1,k_2) - h(k_1,k_2) * \widetilde{f}(k_1,k_2)\right)^2 \\ &\approx \sigma_\eta^2 \end{aligned} \tag{2.3.6}$$

where σ_η^2 is the variance of the noise. There are many potential solutions that satisfy the relation (2.3.6). Another criterion is need to obtain the proper solution from the many solutions. The constrained least-squares filter (CLS filter) H_{cls} is obtained as one of many potential solutions that satisfy the relation:

$$\begin{aligned}\Omega(\widetilde{f}(n_1,n_2)) &= \|c(n_1,n_2) * \widetilde{f}(n_1,n_2)\|^2 \\ &= \frac{1}{NM}\sum_{k_1=0}^{N-1}\sum_{k_2=0}^{M-1}(c(k_1,k_2) * \widetilde{f}(k_1,k_2))^2,\end{aligned} \tag{2.3.7}$$

where $c(n_1,n_2)$ is the PSF of a two dimensional highpass filter.

Interpretation of $\Omega(\widetilde{f}(n_1,n_2))$ is that it gives a measure of the content of the restored image at high frequencies. The minimization of this measure and the limit given by (2.3.6) we get a solution that satisfies both conditions: it is from the set of potential solutions that satisfy the relation (2.3.6) and have less content at high frequency. A typical choice for $c(n_1,n_2)$ is a discrete approximation of the second derivative, known as two-dimensional Laplacian operator:

$$c(n_1,n_2) = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}. \tag{2.3.8}$$

Solution in the spectral domain [26] of the described optimization problem is:

$$\widetilde{F}(u,v) = \left[\frac{H^*(u,v)}{|H(u,v)|^2 + \alpha|C(u,v)|^2}\right] G(u,v). \tag{2.3.9}$$

Thus, the constrained least-squares filter (CLS filter) H_{cls} in discrete Fourier domain is given by the relation:

$$H_{cls}(u,v) = \frac{H^*(u,v)}{H^*(u,v)H(u,v) + \alpha C^*(u,v)C(u,v)}, \tag{2.3.10}$$

where α is a regularization parameter that should be chosen so the relation (2.3.6) is satisfied. Although there are analytical approaches for assessing the regularization parameter, usually parameter α is adjusted interactively until you get acceptable results [6, 43].

It should be noted that although the motivation for designing the two filters are quite different formulation of WF (2.2.3) and CLS filter (2.3.10) are quite similar. These filters work equally well, and they behave in a similar way when the noise variance σ_η^2 is approaching zero.

2.4 The Lucy-Richardson algorithm

The two above defined methods for image restoration are linear. They are also known as direct methods, in the sense that once you specify the restoration filter, the solution is obtained by applying once the filter. This ease of implementation associated with small needs of computer computational and well established theoretical base makes linear techniques essential tool in the image restoration.

Over the past two decades, nonlinear iterative techniques were accepted as tools for image restoration and sometimes gave better results than those obtained with linear methods.Basic limitations of nonlinear methods that their behavior is not always predictable and generally require significantly more computational resources. The first limitation is losing in importance given the fact that nonlinear methods have proved better in terms of linear methods in a wide range of applications [39]. The second limitation is becoming less important because in the last decade we are witness of dramatically increase in the computational power.

One iterative nonlinear method is the Lucy-Richardson algorithm, developed independently by Richardson in 1972 [77] and Lucy in 1974 [52]. Lucy-Richardson algorithm (LR shortly), derived from the formulation for maximum likelihood in which the image is modeled with Poisson statistics. By maximizing the likelihood function of the model gives the following relation which is satisfied when iteration converges:

$$\widetilde{f}_{k+1}(n_1, n_2) = \widetilde{f}_k(n_1, n_2)\left[h(-n_1, -n_2) * \frac{g(n_1, n_2)}{h(n_1, n_2) * \widetilde{f}_k(n_1, n_2)}\right] \tag{2.4.1}$$

The iterative nature of the algorithm is obvious. Nonlinear nature comes from dividing with $\widetilde{f}$ on the right side in relation (2.4.1). As with other nonlinear methods, the question of when to stop Lucy-Richardson algorithm is generally difficult to answer. The most common approach is to observe the output and to terminate the algorithm when the result is acceptable for the given application.

2.5 The symmetric minimal rank solution

The notation $\mathbb{R}^{s\times t}$ denotes the set of all $s \times t$ real matrices. Let $\mathbb{SRR}^{s\times s}$, $\mathbb{ASR}^{s\times s}$ and $\mathbb{OR}^{s\times s}$ be the sets of all $s \times s$ real symmetric, antisymmetric and orthogonal matrices, respectively. Let $\|A\|$, $A^\dagger$, A^T, $R(A)$ and $\mathrm{rang}(A)$ denote the Frobenius norm, the Moore-Penrose generalized inverse, the transpose, range and rank of $A \in \mathbb{R}^{s\times t}$, respectively. The matrix I_s is identity matrix of order s. The inverse matrix problem assumes that $X \in \mathbb{R}^{s\times t}$ and $B \in \mathbb{R}^{s\times t}$ and a positive integer r are given. Find $A \in \mathbb{R}^{s\times s}$ such that $AX = B$ and $\mathrm{rang}(A) = r$.

In this section we use the algorithm from [96] for finding the matrix $\tilde{A}$, for a given A^*, such that

$$\|A^* - \tilde{A}\| = \min_{A\in S_{\tilde{m}}} \|A^* - A\|, \tag{2.5.1}$$

where $S_1 = \{A \in \mathbb{SR}^{s\times s}|\ AX = B\}$ is the solution set, $\tilde{m} = \min\limits_{A\in S_1} \mathrm{rang}(A)$ and

$$S_{\tilde{m}} = \{A|\ \mathrm{rang}(A) = \tilde{m},\ A \in S_1\}$$

is the symmetric minimal rank solution from the solution set S_1. Construction of the matrix $\tilde{A}$ is based on the following statement from [96].

Proposition 2.5.1. [96, Theorem 4.1] *Given $X, B \in \mathbb{R}^{s\times m}$, assume that the singular value decomposition of X is given by*

$$X = U \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} V^T = U_1\Sigma V_1^T, \tag{2.5.2}$$

where

$$\begin{aligned} &U = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \in \mathbb{OR}^{s\times s},\ U_1 \in \mathbb{R}^{s\times k}, \\ &V = \begin{bmatrix} V_1 & V_2 \end{bmatrix} \in \mathbb{OR}^{m\times m},\ V_1 \in \mathbb{R}^{m\times k},\ k = r(X), \\ &\Sigma = diag(\sigma_1, \sigma_2, \ldots, \sigma_k),\ \sigma_1 \geq \ldots \geq \sigma_k > 0. \end{aligned}$$

Let

$$A = U \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22}, \end{bmatrix} U^T \tag{2.5.3}$$

where

$$A_{11} = U_1^T B V_1 \Sigma^{-1} \in \mathbb{SR}^{k\times k},\ A_{12} = \Sigma^{-1} V_1^T B^T U_2 \in \mathbb{SR}^{k\times(n-k)}$$

and

$$A_{21} = U_2 B V_1 \Sigma^{-1} \in \mathbb{SR}^{(n-k)\times k}$$

satisfy

$$A_{11} = A_{11}^T,\ A_{21} = A_{21}^T,\ A_{22} = A_{22}^T.$$

Suppose that the next equations, ensuring solvability of the matrix equation $AX = B$, are satisfied:

$$BV_2 = 0, \quad X^T B = B^T X, \tag{2.5.4}$$

the singular value decomposition of $G_1 = A_{21}\left(I - A_{11}^{\dagger} A_{11}\right)$ is given by

$$G_1 = P \begin{bmatrix} \Gamma & 0 \\ 0 & 0 \end{bmatrix} Q^T = P_1 \Gamma Q_1^T, \tag{2.5.5}$$

where

$$\begin{aligned} &P = \begin{bmatrix} P_1 & P_2 \end{bmatrix} \in \mathbb{OR}^{(s-k)\times(s-k)},\ P_1 \in \mathbb{R}^{(s-k)\times t}, \\ &Q = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \in \mathbb{OR}^{k\times k},\ Q_1 \in \mathbb{R}^{k\times t},\ t = r(G_1), \\ &\Gamma = diag(\alpha_1, \alpha_2, \ldots, \alpha_k),\ \alpha_1 \geq \ldots \geq \alpha_k > 0. \end{aligned}$$

Then (2.5.1) *has a unique solution $\tilde{A}$, which can be written as*

$$\tilde{A} = A_0 + U_2 P_1 P_1^T (A_{22}^* - A_{04}) P_1 P_1^T U_2^T, \tag{2.5.6}$$

where

$$A_0 = BX^{\dagger} + (BX^{\dagger})^T (I - XX^{\dagger}) + (I - XX^{\dagger}) BX^{\dagger} (XX^{\dagger} BX^{\dagger})^{\dagger} (BX^{\dagger})^T (I - XX^{\dagger}), \tag{2.5.7}$$

$X^{\dagger}$ is the Moore-Penrose inverse of X, equal to

$$X^{\dagger} = V_1 \Sigma^{-1} U_1^T$$

and A_{22}^* and A_{04} are given by

$$U^T A_0 U = \begin{bmatrix} A_{01} & A_{02} \\ A_{03} & A_{04} \end{bmatrix},\ U^T A_1^* U = \begin{bmatrix} A_{11}^* & A_{12}^* \\ A_{21}^* & A_{22}^* \end{bmatrix}, \tag{2.5.8}$$

where $A_{01} \in \mathbb{SR}^{k\times k}$ and $A_{11}^* \in \mathbb{SR}^{k\times k}$.

The matrix A_1^* comes from the notion that for any $A^* \in \mathbb{R}^{s\times s}$, we have

$$A^* = A_1^* + A_2^*, \tag{2.5.9}$$

where $A_1^* \in \mathbb{SR}^{s\times s}$, $A_2^* \in (\mathbb{SR}^{s\times s})^{\perp}$, and $(\mathbb{SR}^{s\times s})^{\perp}$ is orthogonal complement space of $\mathbb{SR}^{s\times s}$.

2.6 Moment based image reconstruction method

Moment functions have been used extensively in a number of applications in image analysis, such as pattern recognition, compression, image reconstruction. Furthermore, moment of images have been used in computer vision applications as well as in medical imagining (see [22, 63, 64, 66, 79, 91, 90, 89]).

Moments introduced by Teague [90] are orthogonal moments and based on the theory of orthogonal polynomials. These type of moments are less sensitive to noise and invariant to linear transformation [90]. The moments with discrete orthogonal bases assure very precise image reconstruction, acceptable noise tolerance and are applicable for implementation [63, 64].

Although moments are originally defined in continuous form, discrete formulae are regularly in use for practical reasons. Discrete orthogonal moments have better accuracy in image restoration process with respect to continuous moments [65]. A number of orthogonal discrete moments have been recently introduced. A detailed description of these methods can be found in [15]. In this subsection we show some of the results from [15] for the sake of completeness.

Moments are particularly popular due to their compact description, their capability to select differing levels of detail and their known performance attributes (see [22, 59, 66, 90, 89, 91]). It is a well-recognized property of moments that they can be used to reconstruct the original function, i.e., none of the original image information is lost in the projection of the image on to the moment basis functions, assuming an infinite number of moments are calculated. This is also consistent with work on other types of reconstruction, such as eigenanalysis where it has been found that increasing numbers of eigenvectors are required to capture image detail [79] and again exceed the number required for recognition.

Describing images with moments instead of other more commonly used image features means that global properties of the image are used rather than local properties. The most common reconstruction method of an image from some of its moments is based on the least squares approximation of the image using orthogonal polynomials [63, 70]. In contrast, an image can also be expressed as an element of a vector space, therefore it can be expressed in as a linear combination of the elements of any not necessarily orthogonal basis of this space.

The reconstruction of an image from its moments is not necessarily unique. Thus, all possible methods must impose extra constraints in order to its moments uniquely solve the reconstruction problem.

In this section the constraint that introduced is related to the number of coefficients and the spatial resolution of the image. The Haar basis is unique among the functions we have examined as it actually defines what is referred to as a 'wavelet'. Wavelet functions are a class of functions in which a 'mother' function is translated and scaled to produce the full set of values required for the full basis set. Limiting the resolution of an image means eliminating those regions of smaller size than a given one. The Haar coefficients are obtained from the projection of the image onto the discrete Haar functions $B_{k,l}(m)$ for k which is a power of 2, and are defined as

$$B_{k,l}(m) = \frac{1}{\sqrt{k}},$$

in the case $l = 1$, and for $l > 1$

$$B_{k,l}(m) = \begin{cases} +\sqrt{\frac{q}{k}}, & if \qquad p \le m < p + \frac{k}{2q} \\ -\sqrt{\frac{q}{k}}, & if \qquad p + \frac{k}{2q} \le m \le p + \frac{k}{q} \\ 0, & otherwise \end{cases}$$

with $q = 2^{[log_2(l-1)]}$ and $p = \frac{k(l-1-q)}{q} + 1$, where [.] stands for the function fix(x), which rounds the elements of x to the nearest integer towards zero.

We showed that it is possible to use basis functions in the reconstruction different from orthogonal polynomials, such as Haar basis. The Haar basis allows introducing constraints relative to the spatial resolution on the image to be reconstructed. Following that, the standard least-squares orthogonal reconstruction method can be seen as a particular case of our basis functions reconstruction method. The limitation of the image resolution corresponds to the elimination of the corresponding Haar function coefficients.

The approximation of an image F_{XY} in the least square sense, can be expressed in terms of the projection matrix P_{kl}:

$$P_{kl} = (B_{Xk})^T F_{XY} B_{Yl}$$

as

$$F_{XY}^T = B_{Xk}^T (B_{Xk} B_{Xk}^T)^{-1} P_{kl} (B_{Yl}^T B_{Yl})^{-1} B_{Yl}^T = (B_{Xk})_R^{-1} P_{kl} (B_{Yl})_L^{-1}, \tag{2.6.1}$$

where $()^T$ and $()^{-1}$ denote the transpose and the inverse of the given matrix. The operations $()_L^{-1}$ and $()_R^{-1}$ stand for the left and right inverses, both equal to the Moore-Penrose inverse and are unique. Among the multiple inverse solutions it chooses the one with minimum norm. As the higher order moments reached the reconstruction of the image, it will become more accurate.

We use the approach described in [15, page 350]. For more details about the Fourier basis, see also [19]. Definition of Haar coefficients can be found in [15, page 351].

2.7 Tikhonov and TSVD image restoration methods

In this chapter we present a short overview of the two methods, namely Truncated Singular Value Decomposition (TSVD shortly) and Tikhonov (shortly denoted by TIK). More details for the methods are presented in [32].

The SVD (Singular Value Decomposition) is a matrix computation tool for analyzing of the linear system of equation $Ax = b$. The x and b are long vectors obtained by stacking the columns from the images X and B. The matrix $X \in \mathbb{R}^{m\times n}$ is desired sharp image, the matrix $B \in \mathbb{R}^{m\times n}$ is recorded blurred image and the large blurring matrix $A \in \mathbb{R}^{N\times N}$, $N = m * n$. For this two methods is used the following SVD of the matrix A [32]:

$$A = U\Sigma V^T, \tag{2.7.1}$$

where U and V are orthogonal matrices which satisfy $U^T U = I_N$ and $V^T V = I_N$. The matrix Σ is diagonal matrix with entries $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_N \geq 0$.

These two methods belongs to the family of the spectral filtering methods because they give us control on the spectral content of the deblurred image with the filter factors ϕ_i [32]. With this approach the form of the approximation solution for the $Ax = b$ is

$$x_{filt} = \sum_{i=1}^{N} \phi_i \frac{u_i^T b}{\sigma_i} v_i, \tag{2.7.2}$$

where the columns u_i of U are called the left singular vectors and the columns v_i of V are the right singular vectors. From $U^T U = I_N$ follow that $u_i^T u_j = 0$ if $i \neq j$, and similarly for $v_i^T v_j = 0$

if $i \neq j$. With different selection of the filter factors are obtained different spectral filtering algorithm, for example TSVD and TIK methods.

For TSVD method, also called pseudo-inverse filter, the filter factors are define to be one for large singular value and sero for the rest [32]. The filter factor for TSVD method are given by

$$\phi_i \equiv \begin{cases} 1 & i = 1, \ldots, k, \\ 0 & i = k+1, \ldots, N, \end{cases} \tag{2.7.3}$$

where k is parameter called truncation and determines the number of the SVD components in the regularized solution and the truncation parameter satisfies $1 \leq k \leq N$.

The filter factors for the TIK method [32] are defined as

$$\phi_i \equiv \frac{\sigma_i^2}{\sigma_i^2 + \alpha^2}, \quad i = 1, \ldots, N. \tag{2.7.4}$$

The parameter α is regularization parameter and $\alpha > 0$. Tikhonov solution is related with the minimization problem

$$\min_x \{\|b - Ax\|_2^2 + \alpha^2 \|x\|_2^2\}, \tag{2.7.5}$$

and gives the solution in the form

$$x_{filt} = \sum_{i=1}^{N} \frac{\sigma_i^2}{\sigma_i^2 + \alpha^2} \frac{u_i^T b}{\sigma_i} v_i. \tag{2.7.6}$$

The filter factors of the Tikhonov solution [32] satisfy

$$\phi_i = \begin{cases} 1 - \left(\frac{\alpha}{\sigma_i}\right)^2 + O\left(\left(\frac{\alpha}{\sigma_i}\right)^4\right), & \sigma_i \gg \alpha, \\ \left(\frac{\sigma_i}{\alpha}\right)^2 + O\left(\left(\frac{\sigma_i}{\alpha}\right)^4\right), & \sigma_i \ll \alpha. \end{cases} \tag{2.7.7}$$

The relation (2.7.7) comes from the Taylor expansion

$$(1+\epsilon)^{-1} = 1 - \epsilon + \frac{1}{2}\epsilon^2 + O(\epsilon^3), \tag{2.7.8}$$

in the following relation:

$$\phi_i = \begin{cases} \frac{1}{1+\alpha^2/\sigma_i^2}, & \sigma_i \gg \alpha, \\ \frac{1}{1+\sigma_i^2/\alpha^2}, & \sigma_i \ll \alpha. \end{cases} \tag{2.7.9}$$

Chapter 3

Non-iterative methods for image restoration

3.1 Application of the pseudoinverse computation in reconstruction of blurred images

In the present section, we investigate the problem of removing the blur from images, caused by a uniform linear motion. Our assumptions are that the linear motion corresponds to an integral number of pixels, and it is aligned with the horizontal (or vertical) sampling. We are concentrated on the usage of the Moore-Penrose inverse solution of a given matrix equation which represents a mathematical model of the uniform linear motion blur.

The methods of image restoration, based on the usage of the Moore-Penrose inverse, have been exploited in many recent papers [12, 13, 14]. Several methods for computing the Moore-Penrose inverse have been introduced in [3]. One of the most commonly used methods, is the method of Singular Value Decomposition (SVD). This method is very accurate but also time-intensive since it requires a large amount of computational resources, especially in case of large matrices. An algorithm for fast computation of the Moore-Penrose inverse is also presented in the recent work of P. Courrieu [17]. Courrieu's algorithm is based on the reverse order law for matrix pseudoinverse (eq. 3.2 from [74]), and on the full-rank Cholesky factorization of possibly singular symmetric positive matrices. Another very fast and reliable method for estimation of the Moore-Penrose inverse of full rank rectangular matrices is given by V. Katsikis and D. Pappas [46]. The method uses a special type of tensor product of two vectors.

All methods for computing the Moore-Penrose inverse, mentioned above, are either iterative or use some kind of matrix factorization. The method we propose, explore the structure of the degradation matrix of the model and generates the Moore-Penrose inverse analytically, by means of a set of rules. The motivation behind, is the very proper structure of the matrix which participates as a degradation system in the image formation process. The introduced method is very fast, which is its main advantage. On the other hand, the main disadvantage of the proposed method is its limitation to uniform linear motion blur degradations. The presented numerical results claim the expected decrease in CPU time.

In the next subsections we present the ideas behind the processes of image restoration which are based on the usage of the Moore-Penrose inverse and a new method for reconstruction of blurred images, by generating the Moore-Penrose inverse of the model matrix analytically. In

Section 4.1, by reporting numerical results, we observe certain enhancement in the computational time and the Improvement in Signal to Noise Ration (*ISNR*), compared to other standard methods for image restoration.

3.1.1 Preliminaries

The matrix $A^\dagger$ which satisfies the following properties

$$AA^\dagger A = A, \quad A^\dagger AA^\dagger = A^\dagger, \quad (AA^\dagger)^T = AA^\dagger, \quad (A^\dagger A)^T = A^\dagger A,$$

is called the Moore-Penrose inverse of the matrix A.

This section refers to the formation of the mathematical model that reflects the process of removing the blur in images, which is caused by a uniform linear motion as well as the matrix pseudoinverse solution of the problem.

Suppose that the matrix $F \in \mathbb{R}^{r\times m}$ corresponds to the original image with picture elements $f_{i,j}$, $i = 1,\dots,r$, $j = 1,\dots,m$ and $G \in \mathbb{R}^{r\times m}$ with pixels $g_{i,j}$, $i = 1,\dots,r$, $j = 1,\dots,m$, is the matrix corresponding to the degraded image. Let l be an integer indicating the length of the linear motion blur in pixels and $n = m + l - 1$. In practice the degradation (index l) is rarely known exactly, so that it must be identified from the blurred image itself. To estimate the index l, two different cepstral methods can be used: one dimensional or two dimensional cepstral method [48]. To avoid the problem when the information from the exact image spills over the edges of the recorded image, we supplement the original image with boundary pixels that best reflect the original scene. Without any confusion we are using the same symbol F for the enlarged original image (matrix) with remark that F now becomes a matrix of dimensions $r \times n$. First, we suppose that the blurring is a horizontal phenomenon. Let us denote the degradation matrix by $H \in \mathbb{R}^{m\times n}$. For each row f_i of the matrix F and the respective row g_i of the matrix G we consider an equation of the form

$$g_i^T = Hf_i^T, \;\; g_i^T \in \mathbb{R}^m, \;\; f_i^T \in \mathbb{R}^n, \;\; H \in \mathbb{R}^{m\times n}. \tag{3.1.1}$$

The objective is to estimate the original image F, row by row, using the corresponding rows of the known blurred image G and a priori knowledge of the degradation phenomenon H.

Equation (3.1.1) can be written in the matrix form as

$$G = \left(HF^T\right)^T = FH^T, \;\; G \in \mathbb{R}^{r\times m}, \;\; H \in \mathbb{R}^{m\times n}, \;\; F \in \mathbb{R}^{r\times n}. \tag{3.1.2}$$

There is an infinite number of exact solutions for f that satisfy the equation (3.1.1). But, only the Moore-Penrose inverse solution solves uniquely the next minimization problem (see, for example [3]):

$$\min \|f\|_2, \text{ subject to } \min \|Hf - g\|_2. \tag{3.1.3}$$

The unique vector $\tilde{f}$ satisfying (3.1.3) represents a row of the restored image [7, 12, 13, 14], and it is defined by

$$\tilde{f} = H^\dagger g. \tag{3.1.4}$$

The matrix form of the equation (3.1.4) i.e., the restored image $\tilde{F}$ is given by

$$\tilde{F} = G(H^\dagger)^T. \tag{3.1.5}$$

The matrix $\tilde{F}$ defined in (3.1.5) is the minimum-norm least-squares solution of the matrix equation (3.1.2).

The matrix equation which characterizes the vertical motion blurring process is given by

$$G = HF, \quad G \in \mathbb{R}^{r\times m}, \ H \in \mathbb{R}^{r\times n}, \ F \in \mathbb{R}^{n\times m}, \ n = r + l - 1. \tag{3.1.6}$$

The corresponding restored image can be computed using the Moore-Penrose inverse by the following formula

$$\tilde{F} = H^{\dagger}G. \tag{3.1.7}$$

We assume that the blurring is a local phenomenon, spatially invariant as well that the imaging process captures all light and no additional noise is included. Taking into consideration the given requirements, the degradation matrix of the blurring process reduces to a matrix $H = \text{toeplitz}(h^1, h_1)$. The matrix H is non-symmetric Toepltiz matrix consisting of m rows and $n = m + l - 1$ columns, determined by its first column $h^1 = (h_{i,1})_{i=1}^{m}$ and its first row $h_1 = (h_{1,j})_{j=1}^{n}$ as follows:

$$h_{i,1} = \begin{cases} 1/l, & i = 1, \\ 0, & i = 2, \dots, m, \end{cases} \quad \text{and} \quad h_{1,j} = \begin{cases} 1/l, & j = 1, \dots, l, \\ 0, & j = l+1, \dots, n. \end{cases} \tag{3.1.8}$$

An arbitrary ith row of the blurred image can be expressed using the ith row of the original image extended with the boundary pixels as

$$\begin{bmatrix} g_{i,1} \\ g_{i,2} \\ g_{i,3} \\ \vdots \\ g_{i,m} \end{bmatrix} = \begin{bmatrix} \frac{1}{l} & \frac{1}{l} & \cdots & \frac{1}{l} & 0 & 0 & 0\dots & 0 \\ 0 & \frac{1}{l} & \frac{1}{l} & \cdots & \frac{1}{l} & 0 & 0\dots & 0 \\ 0 & 0 & \frac{1}{l} & \frac{1}{l} & \cdots & \frac{1}{l} & 0\dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{1}{l} & \frac{1}{l} & \cdots & \frac{1}{l} \end{bmatrix} \begin{bmatrix} f_{i,1} \\ f_{i,2} \\ f_{i,3} \\ \vdots \\ f_{i,n} \end{bmatrix}, \tag{3.1.9}$$

where $l-1$ elements of the vector f_i, are not actually the pixels from the original scene; rather they are *boundary pixels*. How many boundary pixels will be added above the vector f depends of the nature and direction of the movement. However, the rest of them, i.e., $l-1$ minus the number of pixels added above the vector f, would present the boundary pixels right of the horizontal line, and are added below the vector f [32].

The process of non–uniform blurring assumes that the blurring of the columns in the image is independent with respect to the blurring of the rows. In this case two matrices participate in the formation of the process and the relation between the original and the blurred image can be displayed with the following relation

$$G = H_c F H_r^T, \quad G \in \mathbb{R}^{m_1\times m_2}, \ H_c \in \mathbb{R}^{m_1\times r}, \ F \in \mathbb{R}^{r\times n}, \ H_r \in \mathbb{R}^{m_2\times n}, \tag{3.1.10}$$

where $n = m_2 + l_r - 1$, $r = m_1 + l_c - 1$, l_r is the length of the horizontal blurring in pixels and l_c is the length of the vertical blurring in pixels. In this case, the Moore-Penrose solution of the system (3.1.10) is given by

$$\tilde{F} = H_c^{\dagger} G (H_r^T)^{\dagger}. \tag{3.1.11}$$

3.1.2 New image restoration method

First, we define a method of image restoration in the case when the number of columns of the image, enlarged by boundary pixels, can be divided by an appropriate number of blurring pixels, i.e., when the equality $n = l \cdot p$ holds. We show that in this case the Moore-Penrose inverse $H^\dagger$ can be generated analytically, without any iterations. Later, we generalize the method to the case when the dimension n is arbitrary.

Let us suppose that

$$n = m + l - 1 = l \cdot p,$$

where the number of blurring pixels l is a positive integer. In the rest of the section, we construct the matrix $\widetilde{H} = [\widetilde{h}_{ij}], i = 1, \ldots, n,\ j = 1, \ldots, m$ and show that it is actually the Moore-Penrose inverse of the degradation matrix H.

All elements of the matrix $\widetilde{H}$, excluding zero elements, can be represented by the following two sequences:

$$x_k = -\frac{l}{n}(m - l(k-1) - 1) = -\frac{m - l(k-1) - 1}{p}, \qquad k = 1, 2, \ldots, p-1,$$
$$y_k = \frac{l}{n}(m - l(k-1)) = \frac{m - l(k-1)}{p}, \qquad k = 1, 2, \ldots, p.$$

Additionally, we put $z = y_p = \frac{1}{p}$.

The general layout of the matrix $\widetilde{H} \in \mathbb{C}^{n \times n}$ in the case $n = m + l - 1 = l \cdot p$ is given in Figure 3.1.1.

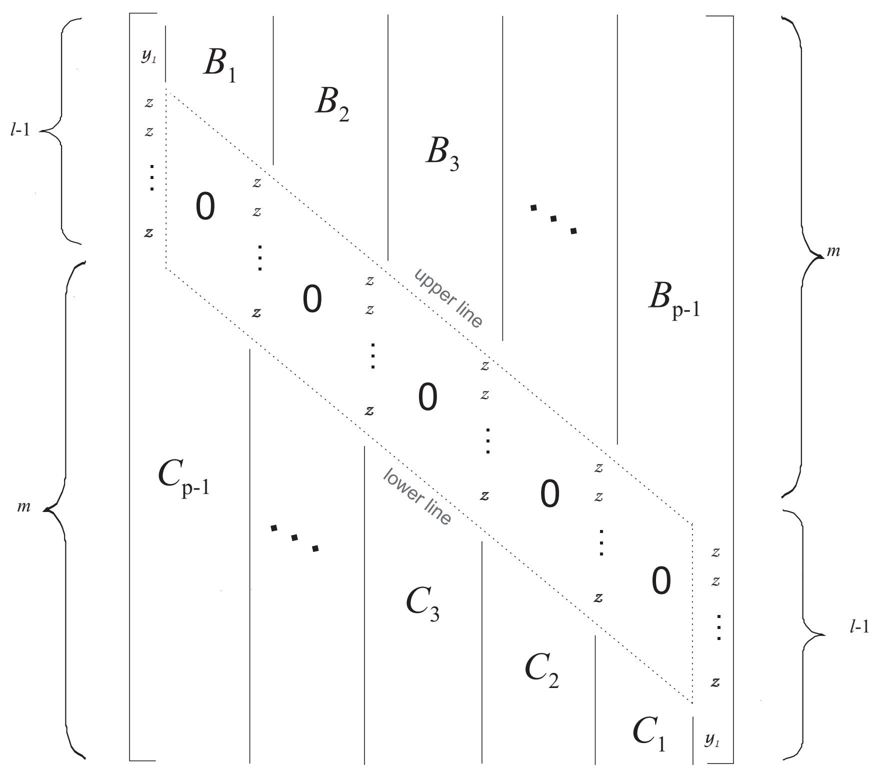


Figure 3.1.1: General layout of the matrix $H^\dagger$ in the case $n = m + l - 1 = l \cdot p$.

The parallelogram between the blocks B_k and C_k is called *zero layer*. The line denoted by *upper line* refers to the elements above the zero layer of the matrix $\widetilde{H}$ which actually constitute the diagonal of the square $m \times m$ matrix formed from the first m rows of the matrix $\widetilde{H}$. Similarly, *lower line* refers to the elements below the zero layer of the matrix $\widetilde{H}$ which constitute the diagonal of the square $m \times m$ matrix formed from the last m rows of the matrix $\widetilde{H}$. The upper line, the lower line, the first l elements of the first row of $\widetilde{H}$ and the last l elements of the last column of $\widetilde{H}$ will be denominated as *sides of the zero layer.*

Further, we preview the structure of the blocks B_k, $k = 1, \ldots, p-1$. Each block B_k can be represented via appropriate block P_k of the following form:

$$P_k = \left|\begin{array}{ccccccc}
0 & & & & & & \\
0 & 0 & & & & & \\
0 & 0 & 0 & & & & \\
\vdots & \ddots & \ddots & \ddots & & & \\
0 & 0 & \cdots & \cdots & 0 & & \\
x_k & 0 & 0 & \cdots & \cdots & 0 & \\
-x_k & x_k & 0 & 0 & \cdots & 0 & z \\
 & -x_k & x_k & 0 & 0 & \vdots & z \\
 & & \ddots & \ddots & \ddots & & \vdots \\
 & & & \ddots & \ddots & 0 & \vdots \\
 & & & & -x_k & x_k & z \\
 & & & & & -x_k & z + x_k \\
 & & & & & & y_{k+1}
\end{array}\right|$$

Zero parallelogram blocks are of dimensions $(l-2) \times (l-1)$. The block B_k is obtained by pasting up k times the block P_k, from the bottom upwards, and then from the resulting block we cut by horizontal line the most upper triangle which has a vertical side containing the upper $l-2$ elements of the block P_k. The missing element of the resulting block B_k is filled by the value of y_{k+1}. The three steps in the formation process of the blocks B_k are presented in Figure 3.1.2.

In order to write the analytical form of the matrix $\widetilde{H}$ we will use the following notation. Let q_i, q_j, r_i and r_j be integers such that for a given ith row and jth column hold $i = q_i l + r_i$ and $j = q_j l + r_j$. From the previous considerations it is clear that: $i \leq j$ if and only if $\widetilde{h}_{ij} \in B_k,\ k = 1, 2, \ldots, p-1$. Similarly, $i \geq l$ and $i > j$ if and only if $\widetilde{h}_{ij} \in C_k,\ k = 1, 2, \ldots, p-1$. Taking

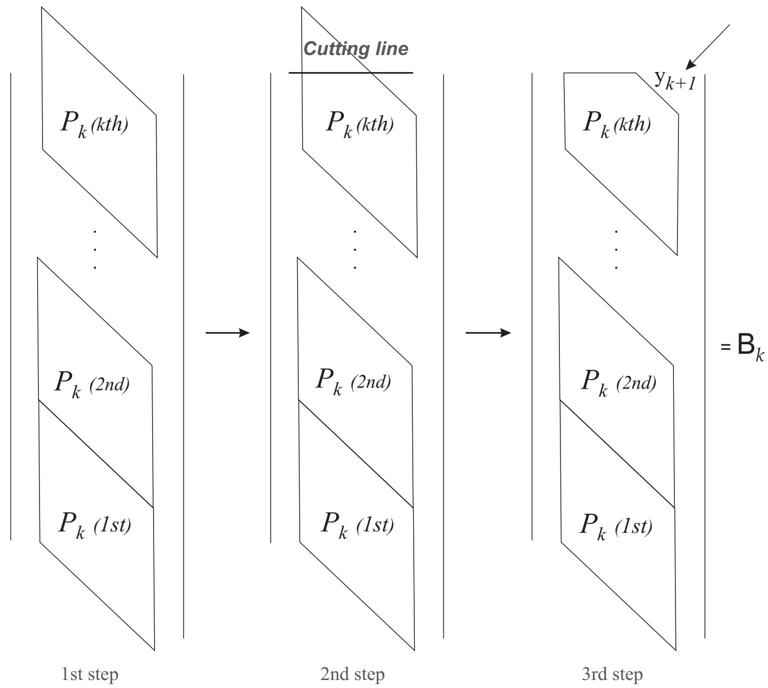


Figure 3.1.2: Formation of the block B_i from the block P_i.

this into account, we present the analytical form of the matrix $\widetilde{H}$ as follows

$$\widetilde{h}_{ij} = \begin{cases} y_{q_j+1}, & i \le j,\ r_j = 1,\ r_i = 1, \\ z + x_{q_j}, & i \le j,\ r_j = 1,\ r_i = 0, \\ (-1)^{d+1} x_{q_{j-1}+1}, & i \le j,\ r_j \ne 1,\ q_j \ge q_i,\ (r_{i-j} = 0 \text{ or } r_{i-j} = l-1) \\ & \\ z + x_{p-q_j-1}, & i \ge l,\ i > j,\ r_j = 1,\ r_i = 1, \\ y_{p-q_j}, & i \ge l,\ i > j,\ r_j = 1,\ r_i = 0, \\ (-1)^{d} x_{p-q_{j-1}-1}, & i \ge l,\ i > j,\ r_j \ne 1,\ q_j \le q_i,\ (r_{i-j} = 0 \text{ or } r_{i-j} = l-1) \\ & \\ z, & r_j = 1,\ r_i \ne 0,\ r_i \ne 1 \\ 0, & \text{otherwise}, \end{cases} \tag{3.1.12}$$

where d is 0 if $r_{i-j} = 0$ and $d = 1$ otherwise. The first case in (3.1.12) gives the elements that are not equal to zero or z of the blocks B_k, $k = 0, \dots, p-1$ plus y_1 from the first column. The second case produces the elements that are not equal to zero or z of the blocks C_k, $k = 0, \dots, p-1$ plus y_1 from the last column. In order to store the whole matrix we need only to store $2p-1$ elements which is lower than n in the case $l > 2$.

To illustrate our description we give the full form of the matrix $\widetilde{H}$ observing the case $p = 4$.

$$
\left[\begin{array}{ccccccccccccccccccc}
y_1 & x_1 & & & & & y_2 & x_2 & & & & & y_3 & x_3 & & & & & y_4 \\
z & -x_1 & x_1 & & 0 & & z & -x_2 & x_2 & & 0 & & z & -x_3 & x_3 & & 0 & & z \\
z & & -x_1 & x_1 & & & z & & -x_2 & x_2 & & & z & & -x_3 & x_3 & & & z \\
\vdots & & & \ddots & \ddots & & \vdots & & & \ddots & \ddots & & \vdots & & & \ddots & \ddots & & \vdots \\
z & & & & -x_1 & x_1 & z & & & & -x_2 & x_2 & z & & & & -x_3 & x_3 & z \\
y_4 & & & & & -x_1 & z+x_1 & & & & & -x_2 & z+x_2 & & & & & -x_3 & z+x_3 \\
z+x_3 & -x_3 & & 0 & & & y_2 & x_2 & & & 0 & & y_3 & x_3 & & & 0 & & y_4 \\
z & x_3 & -x_3 & & & & z & -x_2 & x_2 & & & & z & -x_3 & x_3 & & & & z \\
\vdots & & \ddots & \ddots & & & \vdots & & -x_2 & x_2 & & & z & & -x_3 & x_3 & & & z \\
z & & & x_3 & -x_3 & & z & & & \ddots & \ddots & & \vdots & & & \ddots & \ddots & & \vdots \\
z & & & & x_3 & -x_3 & z & & & & -x_2 & x_2 & z & & & & -x_3 & x_3 & z \\
y_4 & & 0 & & & x_3 & y_3 & & & & & -x_2 & z+x_2 & & & & & -x_3 & z+x_3 \\
z+x_3 & -x_3 & & & & & z+x_2 & -x_2 & & 0 & & & y_3 & x_3 & & & 0 & & y_4 \\
z & x_3 & -x_3 & & & & z & x_2 & -x_2 & & & & z & -x_3 & x_3 & & & & z \\
\vdots & & \ddots & \ddots & & & \vdots & & \ddots & \ddots & & & \vdots & & -x_3 & x_3 & & & z \\
z & & & x_3 & -x_3 & & z & & & x_2 & -x_2 & & z & & & \ddots & \ddots & & \vdots \\
z & & & & x_3 & -x_3 & z & & & & x_2 & -x_2 & z & & & & -x_3 & x_3 & z \\
y_4 & & 0 & & & x_3 & y_3 & & 0 & & & x_2 & y_2 & & & & & -x_3 & z+x_3 \\
z+x_3 & -x_3 & & & & & z+x_2 & -x_2 & & & & & z+x_1 & -x_1 & & 0 & & & y_4 \\
z & x_3 & -x_3 & & & & z & x_2 & -x_2 & & & & z & x_1 & -x_1 & & & & z \\
\vdots & & \ddots & \ddots & & & \vdots & & \ddots & \ddots & & & \vdots & & \ddots & \ddots & & & \vdots \\
z & & & x_3 & -x_3 & & z & & & x_2 & -x_2 & & z & & & x_1 & -x_1 & & z \\
z & & 0 & & x_3 & -x_3 & z & & 0 & & x_2 & -x_2 & z & & 0 & & x_1 & -x_1 & z \\
y_4 & & & & & x_3 & y_3 & & & & & x_2 & y_2 & & & & & x_1 & y_1
\end{array}\right].
$$

In order to verify that the matrix $\widetilde{H}$ is actually the Moore-Penrose inverse of the matrix H, i.e., that $\widetilde{H} = H^\dagger$ we will use the following two lemmas:

Lemma 3.1.1. *The equality* $H\widetilde{H} = I$ *holds for the matrix* $\widetilde{H}$ *given by* (3.1.12).

Dokaz. Each row of the matrix H contains l non zero constant elements equal to $1/l$, so that it is obvious that the elements of the matrix $H\widetilde{H}$ are

$$(H\widetilde{H})_{ij} = \frac{1}{l}\sum_{s=i}^{i+l-1} \widetilde{h}_{sj}, \quad i = 1, \ldots, m \ \text{ and } \ j = 1, \ldots, m.$$

Therefore, we need to explore the properties of jth column of the matrix $\widetilde{H}$, $j = 1, \ldots, m$, i.e., the sums of its l consecutive elements. For this reason, from the representation of the matrix $\widetilde{H}$ given by (3.1.12) we distinguish two different cases:

1 case : $r_j = 1$.

Each set of l consecutive elements contains only two elements different from z. If both of them are above the zero layer their sum is $y_{q_j+1} + (l-1)z + x_{q_j} = 0$. If both of them are below the zero layer their sum is $x_{p-q_j-1} + (l-1)z + y_{p-q_j} = 0$. Otherwise, if one of them is above the zero layer and the other one is below the zero layer, then their sum is $y_{q_j+1} + y_{p-q_j} = l$. Unfortunately, as we mentioned before, the last is the case only when $i = j$.

2 case : $r_j \neq 1$.

Each set of l consecutive elements contains only two non zero elements. If those two elements are above the zero layer or both of them are below the zero layer then it is obvious that their sum is 0. If one of them is above the zero layer and the other one is below the zero layer, thus $i = j$, their sum is $-x_{q_{j-1}+1} - x_{p-q_{j-1}-1}$, which by easy calculations can be shown that equals l.

So finally for the both cases we have

$$(H\widetilde{H})_{ij} = \frac{1}{l}\sum_{s=i}^{i+l-1} \widetilde{h}_{sj} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}, \quad i = 1, \ldots m, \ j = 1, \ldots m.$$

i.e. $H\widetilde{H} = I$. $\square$

Lemma 3.1.2. *The equality* $(\widetilde{H}H)^T = \widetilde{H}H$ *holds for the matrix* $\widetilde{H}$ *given by* (3.1.12).

Dokaz. For a given $i = 1, \ldots, n$ and $j = 1, \ldots, n$, we should show that $(\widetilde{H}H)_{ij} = (\widetilde{H}H)_{ji}$. The elements of the matrix $\widetilde{H}H$ can be presented as

$$(\widetilde{H}H)_{ij} = \frac{1}{l}\sum_{s=\max\{1,j-l+1\}}^{\min\{j,m\}} \widetilde{h}_{is}, \quad i = 1, \ldots, n \ \text{ and } \ j = 1, \ldots, n. \tag{3.1.13}$$

Let us denote by $s = \min\{j, m\} - \max\{1, j-l+1\}$.

Consequently, we should show that the sum of s consecutive elements in the ith row of the matrix $\widetilde{H}H$ where the last element is in the jth column; equals the sum s consecutive elements in the jth row of the matrix $\widetilde{H}H$, where the last element is in the ith column. So the case when $i = j$ is clear, actually, for a given i these elements actually present the sum of the s consecutive elements that belong to the zero layer as well as his sides. We continue with the opposite case when $i \neq j$.

Let us explore the properties of each row i of the matrix $\widetilde{H}$, $i = 1, \ldots, n$. First we recall that if $i \leq j$ that means that $\widetilde{h}_{ij}$ is either y_1 or belongs in a block B_k, $k = 1, \ldots, p-1$. And, if $i > l$ and $i > j$ that means that $\widetilde{h}_{ij}$ is either y_1 or belongs in a block C_k.

1 case : $i < j$, $r_i = 1$ and $r_j \neq 1$ $(i = 1, \ldots, n-1$ and $j = 1, \ldots, n)$.

From (3.1.13) and definition of the matrix $\widetilde{H}$ we have

$$(\widetilde{H}H)_{ij} = \frac{1}{l}(y_{q_j+1} + x_{q_j+1}) = \frac{z}{l}.$$

That means that the sum of each s consecutive elements equals z, if the elements are in the $(kl+1)st$ row, $k = 0, \ldots, p-1$, and the last element is not in the $(kl+1)st$ column, $k = 0, \ldots, p-1$. Also since $i < j$ the last element is above the diagonal of the square matrix constituted of the first m rows of $\widetilde{H}$.

We need to compare these values with the values of $(\widetilde{H}H)_{ji}$. For this situation we analyze the opposite case i.e. we interchange the conditions for i and j. Suppose,

$$i > j,\ r_i \neq 1 \text{ and } \ r_j = 1 \ \ (i = 1,\dots,n \text{ and } j = 1,\dots,n-1).$$

From here we continue with two different possibilities, denoted by $a1$ and $a2$.

a1: if $r_i = 0$ then

$$(\widetilde{H}H)_{ij} = \frac{1}{l}(-x_{q_j} + z + x_{q_j}) = \frac{z}{l}.$$

a2: if $r_i \neq 0$ then

$$(\widetilde{H}H)_{ij} = \frac{1}{l}\begin{cases} z, & j = 1 \\ z + x_{p-q_j} - x_{p-q_j} = z, & j > 1. \end{cases}$$

Thus the first case is completed i.e. if $i < j,\ r_i = 1$ and $r_j \neq 1$ $(i = 1,\dots,n-1$ and $j = 1,\dots,n)$ then $(\widetilde{H}H)_{ij} = (\widetilde{H}H)_{ji}$.

2 case : $i < j,\ r_i = 1$ and $r_j = 1$ $(i = 1,\dots,m-1$ and $j = 1,\dots,m)$ then

$$(\widetilde{H}H)_{ij} = y_{q_j+1} + x_{q_j} = z(1-l).$$

Note: Since $m = l(p-1)+1$ and $n = m+l-1$, if $j > m$ it follows that $r_j \neq 1$.

If the conditions $i > j,\ r_i = 1$ and $r_j = 1$ $(i = 1,\dots,m$ and $j = 1,\dots,m-1)$ are satisfied we obtain

$$(\widetilde{H}H)_{ij} = \begin{cases} z + x_{p-1} = z(1-l), & j = 1 \\ z + x_{p-q_j-1} - x_{p-q_j} = z(1-l), & j \neq 1, \end{cases}$$

which completes the proof in the case 2.

3 case : $i < j,\ r_i \neq 1$ and $r_j = 1$ $(i = 1,\dots,m-1$ and $j = 1,\dots,m)$ then

$$(\widetilde{H}H)_{ij} = \frac{1}{l}(-x_{q_j} + z + x_{q_j}) = \frac{z}{l}.$$

Under the assumptions $i > j,\ r_i = 1$ and $r_j \neq 1$ $(i = 1,\dots,m$ and $j = 1,\dots,m-1)$ one can verify

$$(\widetilde{H}H)_{ij} = \frac{1}{l}(z + x_{p-q_j-1} - x_{p-q_j-1}) = \frac{z}{l},$$

so that the verification of the statement in case 3 is completed.

4 case : $i < j,\ r_i \neq 1$ and $r_j \neq 1$ $(i = 1,\dots,n-1$ and $j = 1,\dots,n)$. Similarly as in the previous cases, after considering several possibilities, one can derive the following

$$(\widetilde{H}H)_{ij} = \frac{1}{l}\begin{cases} z(l-1), & r_i = r_j,\ i \leq m \\ z, & \text{otherwise.} \end{cases}$$

Under the opposite assumptions $i > j,\ r_i \neq 1$ and $r_j \neq 1$ $(i = 1,\dots,n$ and $j = 1,\dots,n-1)$ we get

$$(\widetilde{H}H)_{ij} = \frac{1}{l}\begin{cases} z(l-1), & r_i = r_j,\ j \leq m \\ z, & \text{otherwise,} \end{cases}$$

and the proof is completed. $\square$

Algorithm 3.1.1 Image deblurring method (MP method).

Require: The blurred image G of dimensions $r \times m$ defined in the blurring process (3.1.9).

Step 1. If $m + l - 1 \mod l \neq 0$ then add $l * quotient(m + l - 1, l) + l - m$ boundary pixels, else add $l - 1$ boundary pixels.

Step 2. Compute the matrix $H^\dagger$ according to the formula (3.1.12).

Step 3. Apply formula (3.1.5).

Step 4. Return $\widetilde{F}$.

Theorem 3.1.1. *The matrix* $\widetilde{H}$ *given by* (3.1.12) *is the Moore-Penrose inverse of the matrix* H*.*

Proof. Since the matrix H is full row rank matrix its Moore-Penrose inverse is its right inverse. From this fact and from the previous two lemmas follows the proof of the theorem. □

3.2 Application of partitioning method on specific Toeplitz matrices

We consider the problem of removing non–uniform blur, which corresponds to an integral number of pixels, in images. The real-life linearly blurred image (denoted by the image array G), can be modeled as a linear convolution of the original image (denoted by the image array F) with a PSF, also known as the blurring kernel (represented by the matrix H).

The Moore-Penrose inverse is a useful tool for solving linear systems and matrix equations [3, 71]. These useful properties of the Moore-Penrose inverse cause the appearance of the Moore–Penrose inverse in image restoration process [7, 12, 13, 14]. The approach based on the usage of the matrix pseudo-inverse in the image restoration is one of the most common techniques [7].

Appearance of the blur caused by the linear motion is modeled by the matrix equations $FH^T = G$ and $HF = G$ with respect to the unknown matrix F, where H and G are given matrices of appropriate dimensions. The Moore-Penrose inverse $H^\dagger$ of the matrix H, causing the blur of the original image F into the degraded image G, has been used to solve these equations [12, 13].

In other words, the main problem we are faced with is to choose an efficient algorithm for computing the Moore-Penrose inverse $H^\dagger$. The algorithm used in [12, 13] for computing $H^\dagger$ is based on the fast computational method for finding the Moore-Penrose inverse of full rank matrix, introduced in [42, 46]. Approximations obtained in [12] are reliable and very accurate. A lot of direct methods have been proposed to compute the Moore-Penrose generalized inverse of a matrix (see for instance [3, 81]). According to [81], they can be classified as: methods based on matrix decomposition; methods applicable on bordered matrices and others methods (including Greville's recursive method, methods based on the formula $A^\dagger = (A^*A)^{(1,3)}A^*$ and Pyle's gradient projection methods). The method based on Singular-Value Decomposition possesses very high computation load (approximately $\mathcal{O}(n^3)$ operations). P. Courrieu in [17] proposed an algorithm for fast computation of the Moore-Penrose inverse which is based on the reverse order law property and the full-rank Cholesky factorization of possibly singular symmetric positive matrices. A fast method for computing the Moore-Penrose inverse of full rank $m \times n$ matrices and of square matrices with at least one zero row or column is introduced in [42, 46]. This method exploits a special type of tensor product of two vectors, that is usually used in infinite dimensional Hilbert spaces. Greville in [29] proposed a recurrent rule for determining the Moore-Penrose inverse. Udwadia and Kalaba gave an alternative and a simple constructive proof of the Greville's formula [92]. Due to its ability to undertake sequential computing, the Greville's partitioning method has been extensively applied in statistical inference, filtering theory, linear estimation theory, system identification, optimization as well as in analytical dynamics [28, 38, 40, 41, 75]. Recursive computation of the Moore-Penrose inverse of a matrix to which a block is added, is presented by Bhimsankaram [4]. However, Bhimsankaram proposes a proof which simply verified that the output of his algorithm satisfies the four Penrose equations. Udwadia and Kalaba in [93] provided a constructive proof for the recursive determination of the Moore-Penrose inverse of a matrix to which a block of columns is added. These results are also extended to other types of generalized inverses in [93].

Our intention is the application of the recursive block partitioning method from [93] as well as the partitioning method from [29] in the process of removing non–uniform blur in the image restoration. More precisely, both the block partitioning method and the Greville's single-column partitioning method are appropriately modified and applied in computing the Moore-Penrose inverse solution of the matrix equations $H_C F H_R^T = G$ with respect to unknown matrix F. The definitions of the given matrices are given by (3.2.1). The motivation for using these methods lies in the specific structure of convolution matrices H_C and H_R. The appropriate structure of matrices H_C and H_R reduces the computational complexity of the partitioning method in calculating pseudoinverses $H_C^\dagger$ and $H_R^\dagger$.

In the next subsection we restate some basic definitions, motivations as well as both the recursive block partitioning method and the usual partitioning method for computing the Moore-Penrose inverse. We give an outline of the process of forming the mathematical model that reflects the removal of the non–uniform blur in images. Also we describe a new method for restoring the blurred images which is based on appropriate adaptation of the block partitioning method from [93] and the partitioning method from [29]. A few illustrative examples and comparisons are presented in Section 4.2. Additionally, an example based on the blind deconvolution is given in Section 4.2.2.

3.2.1 Preliminaries and motivation

We firstly describe the mathematical model that reflects the process of removing the non–uniform linear motion blur in images. Let $\mathbb{R}$ be the set of real numbers, $\mathbb{R}^{m\times n}$ be the set of $m\times n$ real matrices and $\mathbb{R}_r^{m\times n}$ be the set of $m\times n$ real matrices of rank r. Suppose that the matrix $F \in \mathbb{R}^{r\times m}$ corresponds to the original image with picture elements $f_{i,j}$, $i = 1,\ldots,r$, $j = 1,\ldots,m$ and $G \in \mathbb{R}^{r\times m}$ with pixels $g_{i,j}$, $i = 1,\ldots,r$, $j = 1,\ldots,m$ is the matrix corresponding to the degraded image.

The process of the non–uniform blurring assumes that the blurring of columns in the image is independent with respect to the blurring of its rows. In this case, the relation between the original and blurred image can be expressed by the matrix equation

$$\begin{aligned} G = H_C F H_R^T,\ \ G \in \mathbb{R}^{r\times m},\ \ H_C \in \mathbb{R}^{r\times n}, \\ F \in \mathbb{R}^{n\times t},\ \ H_R \in \mathbb{R}^{m\times t}, \end{aligned} \tag{3.2.1}$$

where $n = r + l_c - 1$, $t = m + l_r - 1$, l_c is length of the vertical blurring and l_r is length of the horizontal blurring (in pixels).

To avoid the problem when the information from the original image spills over the edges of the recorded image, we supplement the original image with boundary pixels that best reflect the original scene. Without any confusion we are using the same symbol F for the enlarged original image (matrix) with remark that F now becomes of dimensions $n \times t$.

In order to restore the blurred image G included in the model (3.2.1) we use the Moore–Penrose inverse approach, which leads to the solution

$$\widetilde{F} = H_C^\dagger G (H_R^\dagger)^T. \tag{3.2.2}$$

We define an adaptation of the well-known partitioning method to compute the Moore–Penrose inverses $H_C^\dagger$ and $H_R^\dagger$. The notation A_i, $i \in \{1, \dots, n\}$ denotes the first i columns of a matrix $A \in \mathbb{R}^{m \times n}$. Particularly, a_i (resp. a^i) means the ith column (resp. the ith row) of A. By ${}_iA_k$, $i \in \{1, \dots, n-1\}$, $k \in \{1, \dots, n-i\}$ we denote the submatrix of A which consists of the columns $a_{i+1}, \dots, a_{i+k}$. The $m \times m$ identity matrix is denoted by I_m and O_m is zero matrix of order $m \times m$. The notation $\mathbf{0}$ stands for the zero column vector of an appropriate dimension.

For the sake of completeness, we restate the block recursive algorithm for computing Moore-Penrose inverse of matrix $B = [A|C]$, which denotes a matrix A augmented by an appropriate matrix C.

Lemma 3.2.1. [93] *Let $B = [A|\ C]$ be an $m \times (r + p)$ complex matrix whose last p columns are denoted by C. Let*

$$\begin{aligned} R &= I - AA^\dagger, \quad Q = (RC)^T RC, \\ F &= I - Q^\dagger Q, \quad Z = A^\dagger C F. \end{aligned} \tag{3.2.3}$$

Then

$$B^\dagger = \begin{bmatrix} A^\dagger (I - CV) \\ V \end{bmatrix} \tag{3.2.4}$$

where

$$V = Q^\dagger C^T R + (I + Z^T Z)^{-1} Z^T A^\dagger (I - CQ^\dagger C^T R). \tag{3.2.5}$$

We also restate the Greville's single-column finite recursive algorithm from [29].

Lemma 3.2.2. [29] *Let A be an $m \times n$ complex matrix and a be an $m \times 1$ constant vector. By $[A|a]$ we denote the matrix A augmented by an appropriate vector a. Then*

$$[A|a]^\dagger = \begin{bmatrix} A^\dagger - db^* \\ b^* \end{bmatrix}, \tag{3.2.6}$$

where

$$\begin{aligned} d &= A^\dagger a, \\ b &= \begin{cases} \frac{1}{c^* c} c, & c \neq \mathbf{0} \\ \frac{1}{1 + d^* d} (A^\dagger)^* d, & c = \mathbf{0}, \end{cases} \end{aligned} \tag{3.2.7}$$

and

$$c = (I - AA^\dagger) a. \tag{3.2.8}$$

3.2.2 Adaptation of the partitioning method

It is known that an arbitrary linear blurring process can be represented by (3.2.1) where the matrices H_C and H_R are characteristic Toeplitz matrices of the following form, see for example [32]:

$$H = [\ H_m \mid {}_{m+1}H_n\ ] \in \mathbb{R}^{m\times n},\ n = m + l - 1, \tag{3.2.9}$$

where

$$H_m = \begin{bmatrix} h_1 & h_2 & h_3 & \dots & h_l & 0 & 0 & 0 \\ 0 & h_1 & h_2 & h_3 & \dots & h_l & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & \dots & \ddots & 0 \\ 0 & 0 & 0 & h_1 & h_2 & h_3 & \dots & h_l \\ 0 & 0 & 0 & 0 & h_1 & h_2 & h_3 & \dots \\ 0 & 0 & 0 & 0 & 0 & h_1 & h_2 & h_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & \ddots & \ddots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & h_1 \end{bmatrix}, \tag{3.2.10}$$

$${}_{m+1}H_n = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ h_l & 0 & 0 & 0 \\ \dots & h_l & 0 & 0 \\ \ddots & \dots & \ddots & 0 \\ h_2 & h_3 & \dots & h_l \end{bmatrix} \tag{3.2.11}$$

and the sum of the elements $h_1, \dots, h_l$ is equal to 1. The parameter l represent the length of the horizontal/vertical blur (in pixels).

It is important to emphasize that our method can be successfully applied to restore a blurred image in which the blur is caused by an arbitrary linear motion.

We pay special attention to Gaussian blur. Blurring that is caused by atmospheric turbulence, out-of-focus and motion of the camera can be modeled by Gaussian blurring function [37]. In Gaussian blur model the vector $h = [h_1, h_2, \dots, h_l]$ is equal to

$$h = [\gamma(-p), \dots, \gamma(0), \dots, \gamma(k)]$$

where $\gamma(i) = e^{-i^2/(2s^2)}$, $p = \lfloor l/2 \rfloor$, $k = \lceil l/2 \rceil$. The parameter s represents the width of the blurring function. The vector h is normalized by dividing each element of h by the sum of its elements. This vector represents the so called one-dimensional Gaussian function. The non-uniform Gaussian blurring model $G = H_C F H_R^T$ corresponds to the model where the blurring matrix is obtained by convolving the original matrix F by the PSF function which is equal with two-dimensional Gaussian matrix $PSF = [p_{i,j}]$ with entries $p_{i,j} = e^{-i^2/(2s^2) - j^2/(2s^2)}$.

In order to see how boundary conditions can be incorporated in the model, for the sake of the simplicity, let us retain on the horizontal blurring model ($H_C = I, H_R = H$). An arbitrary ith row g^i of the blurred image can be expressed using the ith row f^i of the original image extended by incorporating the boundary conditions as

$$\left(g^i\right)^T = H\ \left(f^i\right)^T \Longleftrightarrow \begin{bmatrix} g_{i,1} \\ \vdots \\ g_{i,m} \end{bmatrix} = H \begin{bmatrix} f_{i,1} \\ \vdots \\ f_{i,n} \end{bmatrix}, \qquad i = 1, 2, \dots, r, \tag{3.2.12}$$

where $l-1$ elements of the vector f^i, are not actually the pixels from the original scene; rather they are *boundary pixels*. How many boundary pixels are placed at the top of the vector f^i depends of the nature and direction of the movement (causer of the blur). However, the rest of them, i.e. $l-1$ minus the number of pixels placed at the top of the vector f^i, would present the boundary pixels right of the horizontal line, and should be placed to the bottom of the vector f^i [32].

Courrieu in the paper [17] compared the introduced method *geninv* with four usual algorithms (the Greville's partitioning method, the SVD method, full rank QR and an iterative method of optimized order [3]). It is claimed that the best results are achieved by the *geninv* method, while the worst results are generated by the partitioning method. In the present section we propose an adaptation of the partitioning method to the Toeplitz matrices of the form (3.2.9)–(3.2.11).

Our motivation for using the block partitioning method [93] and the Partitioning method from [29] in order to find $H^\dagger$ is explained as follows. The quadratic block H_m of the matrix H is clearly nonsingular upper triangular Toeplitz matrix, so that its inverse can be computed very easy. Later, recursive rules (3.2.3)–(3.2.5) and (3.2.6)–(3.2.8) can be significantly simplified, according to specific structure of the block C and the vector a, respectively.

The following particular case of the Lemma 3.2.1 defines simplifications of recursive steps (3.2.3)–(3.2.5) in calculating the Moore-Penrose inverse $H^\dagger$.

Lemma 3.2.3. *Assume that the matrix $H \in \mathbb{R}^{m\times n}$, $n = m + l - 1$ causes the blurring process in* (3.2.12)*. The Moore-Penrose inverse of its first $p+k$ columns, partitioned in the block form $H_{p+k} = [H_p|\ {}_pH_k]$, $p \in \{1,\dots,n-1\}$, $k \in \{1,\dots,n-p\}$, is defined by*

$$H^\dagger_{p+k} = \begin{bmatrix} H_p^\dagger\left(I - {}_pH_k \cdot B^T\right) \\ B^T \end{bmatrix} = \begin{bmatrix} H_p^\dagger - DB^T \\ B^T \end{bmatrix}, \tag{3.2.13}$$

where

$$D = H_p^\dagger \cdot\ {}_pH_k \text{ and } B = (H_p^\dagger)^T D (I + D^T D)^{-1}. \tag{3.2.14}$$

Proof. Follows from Lemma 3.2.1, taking into account that the degradation matrix is of full row rank and the fact that equalities in (3.2.3) reduce to

$$R = Q = O_m,\ F = I_m,\ V = B^T,\ Z = D = H_p^\dagger \cdot\ {}_pH_k,$$

observing this particular case. □

Also, the Greville's Partitioning method (3.2.6)–(3.2.8) reduces to the following computational procedure.

Lemma 3.2.4. *The Moore-Penrose inverse of the matrix H_i is equal to*

$$H_i^\dagger = \begin{bmatrix} H_{i-1}^\dagger - d_i b_i^T \\ b_i^T \end{bmatrix}, \tag{3.2.15}$$

where

$$d_i = H_{i-1}^{\dagger} \cdot h_i \text{ and } b_i = \left(1 + d_i^T d_i\right)^{-1} (H_{i-1}^{\dagger})^T d_i. \tag{3.2.16}$$

Since we know the inverse H_m^{-1}, which is completely determined by vector x from (3.2.17), any pragmatical implementation of the new method uses only partitions of the form $H_{p+k} = [H_p|\ {}_pH_k]$, $p \geq m$, $k \in \{1, \dots, n-m\}$.

According to Lemma 3.2.3 we propose the Algorithm 3.2.1 for computing the Moore-Penrose inverse of the specific Toeplitz matrix H.

Algorithm 3.2.1 Computing the Moore-Penrose inverse of the matrix H.

Input: The matrix H of dimensions $m \times (m+l-1)$ given by (3.2.9).

Step 1. Separate the block H_m of the matrix H.

Step 2. Generate $H_m^{\dagger} = H_m^{-1}$ using the vector x from (3.2.17).

Step 3. Take $p = m$ and choose k such that $1 \leq k \leq l-1$ as well as $\frac{l-1}{k} \in \mathbb{N}$.

Step 4. Compute $H_{p+k}^{\dagger}=[H_p|\ {}_pH_k]^{\dagger}$ according to Lemma 3.2.3.

Step 5. Set $p = p + k$.

Step 6. If $p \neq n$ then go to Step 4; otherwise, go to the next step.

Step 7. Return $H_n^{\dagger}$.

It is not difficult to verify that the choice $k = 1$ in all recursive steps of Algorithm 3.2.1 produces the particular case of Greville's recursive method corresponding to Lemma 3.2.4. Also, in the case $p = m, k = l-1$ Algorithm 3.2.1 reduces to the Algorithm 3.2.2.

Algorithm 3.2.2 Computing the Moore-Penrose inverse of the matrix H in the case $k = l-1$.

Input: The matrix H of dimensions $m \times (m+l-1)$ defined in the blurring process (3.1.9).

Step 1. Separate matrix H into two blocks H_m and ${}_mH_{l-1}$, that is $H = [H_m|\ {}_mH_{l-1}]$.

Step 2. Generate $H_m^{\dagger} = H_m^{-1}$ using the vector x from (3.2.17).

Step 3. Compute $H^{\dagger}=[H_m|\ {}_mH_{l-1}]^{\dagger}$ according to Lemma 3.2.3.

Choosing the most efficient case with respect to the computational time, we derive an efficient method for computing the Moore-Penrose inverse of the degradation matrix H, and respectively an efficient method for image restoring processes based on the equation (3.2.2).

In order to invert the matrix H_m defined in (3.2.10) look at the matrix equation $H_m H_m^{-1} = I$. Since the matrix H_m is upper triangular Toeplitz matrix, it is well-known that its inverse is also upper triangular Toeplitz matrix. Therefore, the whole matrix H_m^{-1} is determined by its last column. We denote the last column of H_m^{-1} by x. To generate the vector x we consider the following equation

$$H_m \cdot x = e_m, \tag{3.2.17}$$

where e_m denotes the last column of the identity matrix I_m. Looking at the methods incorporated in the programming package `MATLAB` we decide to use the `linsolve()` function using the option `opts.UT = true` that imposes computations adopted to upper triangular matrices. After computing the vector x, it is easy to determines the whole matrix H_m^{-1}.

Complexity of Partitioning method

In order to determine the best choice of the positive integer k in Algorithm 3.2.1, we compare computational complexities of Algorithm 3.2.1 and Algorithm 3.2.2. Let us denote by $I(n)$ the complexity of the algorithm for inverting a given $n \times n$ matrix (as in [16]). Also by $A(n)$ we denote the complexity of the addition/subtraction of two $n \times n$ matrices and by $M(m,n,k)$ the complexity of multiplying $m \times n$ matrix by $n \times k$ matrix. The simpler notation $M(n)$ (taken from [16]) is used instead of $M(n,n,n)$.

In the remaining of this subsection we consider the computational complexity of the two opposite choices in Algorithm 3.2.1. The choice $p = m, k = 1$ is called *Partitioning method* (PM method shortly). The opposite choice $p = m, k = l-1$, used in Algorithm 3.2.2, is called *Block Partitioning Method* (shortly BPM method).

It is well-known that the complexity of matrix inversion is equal to complexity of matrix multiplication. More precisely, the ordinary inverse of any real nonsingular $n \times n$ matrix can be computed in time $I(n) = \mathcal{O}(M(n))$ [16]. The notation $\mathcal{O}(f(n))$ is described, also, in [16].

The complexity of Algorithm 3.2.2 is of the order

$$\begin{aligned} E_{BPM} = {} & I(m)+3\,M(m,m,l-1)+2\,M(m,l-1,l-1) \\ & + I(l-1) + A(l-1) + A(m). \end{aligned} \tag{3.2.18}$$

Scanning Algorithm 3.2.1 in a similar way, it is not difficult to verify that its ith recursive step requires complexity of the order

$$\begin{aligned} C_i = M(m+i-1,m,1) + M(1,m+i-1,1) \\ + M(m,m+i-1,1), \end{aligned}$$

for each $i = 1,\ldots,l-1$. Therefore, the complexity of the complete algorithm is

$$E_{PM} \;=\; I(m)+\sum_{i=1}^{l-1} C_i. \tag{3.2.19}$$

Since $l \ll m$, we conclude that the computational complexity for $(I + D^T D)^{-1}$, equal to $I(l-1)$, is substantially smaller than the complexity of required matrix multiplications. Also, upon the adopted implementation for computing $H_m^\dagger = H_m^{-1}$, based on (3.2.17), we have

$$I(m) \approx \mathcal{O}\left(\frac{(m-1)m}{2}\right) = \mathcal{O}(m^2).$$

Therefore, the upper bound estimation of complexities E_{BPM} and E_{PM} does not include the computational effort of the included matrix inversions. The upper bounds for the complexity of Algorithm 3.2.2 and Algorithm 3.2.1 are given, respectively, by

$$\begin{aligned} E_{BPM} &\le \mathcal{O}(M(m,m,l-1)), \\ E_{PM} &\le l \cdot \mathcal{O}(M(m,m+l,1)). \end{aligned}$$

If A is $m \times n$ and B is $n \times k$ matrix, then the computational complexity of the product $A \cdot B$ in `MATLAB` is $M(m,n,k) = \mathcal{O}(nmk)$, since `MATLAB` does not use Strassen's method (or any other rapid method) for matrix multiplication. Therefore, according to (3.2.18) and (3.2.19)

$$E_{BPM} \le \mathcal{O}(m^2 l - m^2),\ E_{PM} \le \mathcal{O}(m^2 l + ml^2).$$

Consequently, the upper bound for the computational complexity of Algorithm 3.2.2 is less than the computational complexity of Algorithm 3.2.1. According to these theoretical investigations as well as on the basis of performed numerical experiments, we conclude that Algorithm 3.2.2 is better choice. The CPU times depend upon two parameters: computational complexity and implementation details incorporated into the programming language `MATLAB`.

On the other hand, according to known result from [67], the number of required operations for Greville's method is equal to

$$\phi(Greville) = 2m^2n - \frac{nr^2}{2},$$

where m, n are dimensions of the input matrix and r is its rank. In our case, the number of arithmetic operations required by the original Greville's method for computing $H^\dagger$, $H \in \mathbb{R}_m^{m\times(m+l-1)}$, is equal to

$$\begin{aligned} E_{Greville} &= 2m^2(m+l-1) - \frac{(m+l-1)m^2}{2} \\ &\approx \mathcal{O}(m^3). \end{aligned}$$

Analysis of methods for computing $H_m^\dagger$

In order to confirm the efficiency of Algorithm 3.2.2, we compared *Block Partitioning Method* with three recently announced methods for computing the Moore-Penrose inverse in [12, 14, 17, 46]. Therefore, the following algorithms for computing the Moore-Penrose inverse are compared:
1. *Block Partitioning Method* (shortly *BPM*), presented by Algorithm 3.2.2,
2. *Ginv* method, defined by the `MATLAB` function `ginv.m` from [46],
3. *Qrginv* method, defined by the `MATLAB` function `qrginv.m` from [14, 45],
4. *Courrieu* method from [17].

A comparison of several direct algorithms for computing the Moore–Penrose inverse of full column rank matrices is presented in [82]. Also, computational cost of these methods for computing the Moore–Penorse inverse of full column rank matrix $A \in \mathbb{R}^{m\times n}$ is given in Table 1 from [82]. In our case, we have the situation $A = H^T \in \mathbb{R}^{m+l-1\times m}$. According to computational complexities presented in [82], the complexity of *Courrieu* method is equal to

$$E_{Chol} = 3(m+l-1)m^2 + m^3/3$$

and the complexity of *Qrginv* method is

$$E_{Qrpivot} = 5(m+l-1)m^2 - 4m^3/3.$$

Ginv method for computing $A^\dagger$ is based on the formula $A^\dagger = (AA^T)^{-1}A$ and `MATLAB` implementation is based on the least squares solution of the matrix equation $(A^TA)X = A^T$. In the particular case $A = H$, computational complexity of *Ginv* method is

$$E_{Ginv} = \mathcal{O}(M(m, m+l-1, m)) + I(m) + M(m, m, m+l-1).$$

Taking into account $l < m$ we derive the following computational complexities:

$$\begin{aligned} E_{Chol} \approx \mathcal{O}(m^3),\ E_{Qrpivot} \approx \mathcal{O}(m^3), \\ E_{Ginv} \approx \mathcal{O}(m^3),\ E_{Greville} \approx \mathcal{O}(m^3). \end{aligned}$$

Since

$$E_{BPM} \approx \mathcal{O}(m^2 l),$$

we conclude that the *Block Partitioning Method* has the smallest computational complexity.

3.3 Removal of blur in images based on least squares solutions

The main goal of this section is the development of an algorithm that allows us to remove a linear motion blur from images. The algorithm is based on the least squares solution of a matrix equation which represents the mathematical model of the linear motion blur. The least squares solution includes the Moore-Penrose inverse of the blurring matrix as well as an arbitrary matrix Y. Satisfactory results are obtained when the matrix Y is suitably defined. Appropriate choice of the matrix Y leads to significant improvements with respect to the classical image restoration algorithms as well as the image restoration approach based on the Moore-Penrose solution of certain matrix equations, which is investigated in [12, 13].

Following, the proposed reconstruction method the image was expressed as an element of a vector space that can be seen as a linear combination of the elements of any not necessarily orthogonal basis of this space. In this section we showed that it is possible to use basis functions in the reconstruction different from orthogonal polynomials, such as Haar basis. Overall, this section presents a reconstruction method that sheds new light on the inverse problems and finds application on the image processing and analysis.

Two main contributions of this method can be displayed. First, we define robust and resolute image reconstruction method which can be used as improvement of an arbitrary image restoration method. The second contribution is a freshly vision of the image analysis through a basis function approach. The numerical and experimental results related to this method are described in Section 4.3.

3.3.1 Description of the method

We start from the mathematical model in which the linear motion is a local phenomenon, the imaging process captures all light and no additional noise is included. Under these assumptions, an arbitrary ith row $g_i = [g_{i,1}, \dots, g_{i,m}]$ of the blurred image $G \in \mathbb{R}^{p\times m}$ can be related with the ith row $f_i = [f_{i,1}, \dots, f_{i,m}]$ of the original image F_0, extended by taking into considerations the boundary conditions, as in (3.3.1)

$$g_i^T = \left[\begin{array}{ccc|ccccccccc|ccc}
h_u & \cdots & h_1 & h_0 & h_{-1} & \cdots & h_{-u} & & & & & & & & \\
 & \ddots & \vdots & h_1 & h_0 & \ddots & \vdots & \ddots & & & & & & & \\
 & & h_u & \vdots & h_1 & \ddots & h_{-1} & \cdots & h_{-u} & & & & & & \\
 & & & h_u & \cdots & \ddots & h_0 & \ddots & \vdots & \ddots & & & & & \\
 & & & & h_u & & h_1 & \ddots & h_{-1} & & h_{-u} & & & & \\
 & & & & & \ddots & \vdots & \ddots & h_0 & \ddots & & h_{-u} & & & \\
 & & & & & & h_u & & h_1 & \ddots & & \vdots & h_{-u} & & \\
 & & & & & & & \ddots & \vdots & \ddots & h_0 & h_{-1} & \vdots & \ddots & \\
 & & & & & & & & h_u & \cdots & h_1 & h_0 & h_{-1} & \cdots & h_{-u}
\end{array}\right]
\left[\begin{array}{c}
w_{i,1} \\ \vdots \\ w_{i,u} \\ \hline f_{i,1} \\ \vdots \\ f_{i,m} \\ \hline v_{i,1} \\ \vdots \\ v_{i,u}
\end{array}\right]. \tag{3.3.1}$$

The quantities h_i are real numbers and $n - 1 = 2u$ elements of the right hand vector, denoted by $w_{i,1}, \dots w_{i,u}$ and $v_{i,1}, \dots v_{i,u}$, are not actually the pixels from the original scene; rather they are *boundary pixels*. The boundary pixels left of the horizontal line, which are added above the initial vector $f_i = \{f_{i,1}, \ldots, f_{i,m}\}^T \in \mathbb{R}^m$, depends of the nature and direction of the movement. The u boundary pixels right of the horizontal line are added below the vector f_i [32]. The equation (3.3.1) can be written in the form

$$g_i^T = \left[\ H_{-1} \mid H_0 \mid H_1 \ \right] \left[\begin{array}{c} f_{i,-1} \\ \hline f_i^T \\ \hline f_{i,1} \end{array} \right], \tag{3.3.2}$$

where

$$f_{i,-1} = \begin{bmatrix} w_{i,1} \\ \vdots \\ w_{i,u} \end{bmatrix}, \ f_{i,1} = \begin{bmatrix} v_{i,1} \\ \vdots \\ v_{i,t} \end{bmatrix}, \ H_{-1} \in \mathbb{R}^{m\times u}, \ H_1 \in \mathbb{R}^{m\times u}, \ H_0 \in \mathbb{R}^{m\times m}.$$

The mathematical model (3.3.2) is reused mainly from [23, 32]. The matrix H is $m \times s$ real matrix, the index n indicates the length of linear motion blur in pixels and $s = m + n - 1 = m + 2u$, $m \gg n$.

The objective is to estimate the original image $F_0 \in \mathbb{R}^{p\times m}$ row per row (contained in the vector f_i^T), by exploiting given row of a blurred image (contained in the vector g_i^T) and a priori knowledge of the degradation phenomenon

$$H = \left[\ H_{-1} \mid H_0 \mid H_1 \ \right].$$

Denote by F_{-1} (resp. by F_1) the matrix whose columns are $f_{i,-1}$ (resp. $f_{i,1}$). By F we denote the block matrix

$$F = \left[\ F_{-1} \mid F_0^T \mid F_1 \ \right] \in \mathbb{R}^{p\times s},$$

where $F_{-1} \in \mathbb{R}^{p\times u}$, $F_1 \in \mathbb{R}^{p\times u}$, $F_0 \in \mathbb{R}^{p\times m}$. Then the equation (3.3.2) can be written in matrix form as

$$\begin{aligned} G &= \left(\left[\ H_{-1} \mid H_0 \mid H_1 \ \right] \begin{bmatrix} F_{-1} \\ F_0^T \\ F_1 \end{bmatrix} \right)^T = \left[\ F_{-1}^T \mid F_0 \mid F_1^T \ \right] \begin{bmatrix} H_{-1}^T \\ H_0^T \\ H_1^T \end{bmatrix} \\ &= FH^T, \quad G \in \mathbb{R}^{p\times m}, \ H \in \mathbb{R}^{m\times s}, \ F \in \mathbb{R}^{p\times s}, \ s = m + n - 1. \end{aligned} \tag{3.3.3}$$

Different boundary conditions (BCs) are known. Main of them are restated from [80] as follows.

- The *zero* (Dirichlet) BCs assume that pixels outside the domain of consideration are black. This implies $F_{-1} = F_1 = O$, where O denotes the zero $p \times u$ matrix. Then the matrix system in (3.3.3) can be shortly written by

$$G = F_0 H_0^T.$$

- The *periodic* BCs assume that the true scene is comprised of periodic copies of F_0. The matrix equation (3.3.3) obtains the form

$$G = F_0 \left(\left[\ O \mid H_{-1} \ \right] + H_0 + \left[\ H_1 \mid O \ \right] \right)^T,$$

where O is the $m \times (m-u)$ zero block. Equivalently, in (3.3.3) we can use

$$f_{i,-1} = \begin{bmatrix} f_{i,m-u} \\ \cdots \\ f_{i,m} \end{bmatrix}, \quad f_{i,1} = \begin{bmatrix} f_{i,1} \\ \cdots \\ f_{i,u} \end{bmatrix}.$$

- The *reflective* BCs means that the data outside the domain of consideration are taken as a reflection of the data inside. In this case, the matrix equation (3.3.3) becomes

$$G = F_0 \left(\left[\ O \,\middle|\, H_{-1}\ \right] J + H_0 + \left[\ H_1 \,\middle|\, O\ \right] J \right)^T,$$

where O is the $m \times (m-u)$ zero block and

$$J = \begin{bmatrix} & & 1 \\ & \cdots & \\ 1 & & \end{bmatrix}$$

is the $m \times m$ reversal matrix. In this case, in (3.3.3) we have

$$f_{i,-1} = \begin{bmatrix} f_{i,m} \\ \cdots \\ f_{i,m-u} \end{bmatrix}, \quad f_{i,1} = \begin{bmatrix} f_{i,u} \\ \cdots \\ f_{i,1} \end{bmatrix}.$$

Since there is an infinite number of exact solutions for F which satisfy the equation $G = FH^T$, additional criterions that ensure a sharp restored matrix is required. This section provides a new method for restoration of a blurred image using the set of least squares solutions of the matrix equation (3.3.3). The least squares solutions in this section are generated using the generalized inverses.

Generalized inverses are used in the case of a singular square matrix, or in the case of a rectangular $m \times n$ matrix. In fact, many kinds of generalized inverses are widely used in the literature. The Moore-Penrose inverse of a matrix $A \in \mathbb{C}^{m \times n}$ is the unique matrix $A^\dagger$ satisfying the following four matrix equations:

$$(1) \quad AXA = A \qquad (2) \quad XAX = X \qquad (3) \quad (AX)^* = AX \qquad (4) \quad (XA)^* = XA. \tag{3.3.4}$$

A matrix X is called an $\{i,j,k\}$–inverse of A (with $i,j,k \in \{1,2,3,4\}$) if X satisfies the ith, jth and kth Penrose equations. Then this matrix is not unique. Many problems in applied linear algebra have been solved using $\{i,j,k\}$–inverses. A particular $\{1\}$–inverse (or g–inverse) of A is noted by $A^{(1)}$ and finds applications in solving linear systems of equations or matrix equations.

We start from the following well-known result [3, 94].

Proposition 3.3.1. *Let $A \in \mathbb{C}^{n \times p}, B \in \mathbb{C}^{s \times m}, D \in \mathbb{C}^{n \times m}$. The matrix equation*

$$AXB = D \tag{3.3.5}$$

is consistent if and only if

$$AA^{(1)}DB^{(1)}B = D \tag{3.3.6}$$

for some g–inverses $A^{(1)}$ and $B^{(1)}$, in which case the general solution is

$$X = A^{(1)}DB^{(1)} + Y - A^{(1)}AYBB^{(1)} \tag{3.3.7}$$

for arbitrary $Y \in \mathbb{C}^{p \times s}$.

The matrix equation (3.3.3) is a particular case ($A \equiv I, B \equiv H^T, D \equiv G, X \equiv F$) of (3.3.5), where I is an appropriate identity matrix. We use $A^\dagger$ and $B^\dagger$ instead of $A^{(1)}$ and $B^{(1)}$, respectively. Then the condition for the existence (3.3.6) yields

$$G(H^T)^\dagger H^T = G, \tag{3.3.8}$$

which is evidently true (since H^T is left invertible). The general solution of the form (3.3.7) becomes

$$\begin{aligned} F &= G(H^T)^\dagger + Y\left(I - H^T(H^T)^\dagger\right) \\ &= G(H^T)^\dagger + Y\left(I - (H^\dagger H)^T\right) \\ &= G(H^T)^\dagger + Y\left(I - H^\dagger H\right). \end{aligned} \tag{3.3.9}$$

The matrix $Y \in \mathbb{R}^{p\times s}$, which can be randomly chosen, can be determined in different ways.

1. The first approach.

In the case $Y = O$, where O is the zero $p\times s$ matrix, (3.3.9) produces the next approximation F of the original image F:

$$F = G(H^T)^\dagger. \tag{3.3.10}$$

This solution is investigated in [12, 13]. Since the Moore-Penrose inverse possesses well-known minimal properties (see, for example [53, 54, 55, 71]), the approximation F is the least squares solution of (3.3.3), i.e.

$$\|FH^T - G\|_2 \leq \|XH^T - G|\ X \in \mathbb{R}^{p\times s}\|_2. \tag{3.3.11}$$

2. The second approach.

So far developed algorithms based on the usage of the Moore-Penrose inverse assume the condition $Y = O$. This assumption exploits the Moore-Penrose solution of the matrix equation, i.e. the least squares solution of minimal norm. But, the minimal norm attribute, imposed to the restored image, may be in most of cases only the redundant property. Therefore, it is realistic to expect that the approach based on the opposite assumption $Y \neq O$ will show better performances in some cases. According to (3.3.9), approximation of the original image is given by

$$E(Y) = \widetilde{F} = G(H^T)^\dagger + Y\left(I - H^\dagger H\right) = F + Y\left(I - H^\dagger H\right), \quad Y \in \mathbb{R}^{p\times s}. \tag{3.3.12}$$

In the case $X = \widetilde{F}$ the inequality in (3.3.11) becomes equality, which means that $\widetilde{F}$ is also the least squares solution. The minimal norm property

$$\|F\|_2 \leq \|\widetilde{F}\|_2 = \|F + Y\left(I - H^\dagger H\right)\|_2, \tag{3.3.13}$$

associated with the Moore-Penrose solution, is needless in the image restoration process. It is necessary to determine Y in such a way that the approximation $\widetilde{F}$ produces better values for the $ISNR$ and $PSNR$ with respect to the solution F generated by (3.4.8), which is used in [12, 13].

One possible way to choose optimal values for Y is described in the sequel. It is necessary to minimize the matrix norm

$$\|\widetilde{F} - F\|_2,$$

i.e. to solve the optimization problem

$$\min\ Q(Y) = \|G(H^T)^\dagger + Y\left(I - H^\dagger H\right) - F\|_2 \tag{3.3.14}$$

with respect to unknown matrix Y and an arbitrary matrix norm.

Remark 3.3.1. *The inequalities of the form* (3.3.11) *and* (3.3.13) *was originally stated for arbitrary matrix equation* $AX = B$ *by Penrose* [71] *for the Frobenius norm* $\|\cdot\|_F$ *of matrices. The Frobenius norm is defined by*

$$\|A\|_F = \sqrt{\sum_i \sum_j |a_{ij}|^2}.$$

Penrose's inequalities (3.3.11) *and* (3.3.13) *has been extended in* [53, 54, 55] *to the supremum norm and the* L_p *norm as well as to the set of* $\{1, 3\}$ *and* $\{1, 4\}$ *inverses. Throughout this section, the matrix norm used is the* $\|A\|_2$ *norm, which is the most commonly used together with the Frobenius norm in numerical linear algebra. The* $\|A\|_2$ *norm is defined as*

$$\|A\|_2 = sup_{\ x \neq 0} \frac{\|Ax\|_2}{\|x\|_2},$$

where

$$\|x\|_2 = \sqrt{\sum_i x_i^2}$$

is the well–known vector norm. Both norms are measuring distance, they are equivalent and are related with a well known inequality:

$$\|A\|_2 \leq \|A\|_F \ \leq \sqrt{r}\,\|A\|_2$$

where r *is the rank of* A. *(For more, see* [25]*, p. 52- 58)*

3. Ideal solution of the problem (3.3.14) can be generated from the matrix equation

$$Y_{\text{opt}}(I - H^\dagger H) - (F - G(H^T)^\dagger) = O, \tag{3.3.15}$$

which implies

$$Y_{\text{opt}}(I - H^\dagger H) - (F - FH^T(H^T)^\dagger) = Y_{\text{opt}}(I - H^\dagger H) - F(I - H^\dagger H) = 0$$

and later

$$Y_{\text{opt}} = F.$$

But, the choice $Y = F$ is not allowed (confirms the claim that there is no hope that we can recover the original image exactly! [32]) Therefore, the only possible approach is to use values for Y as close as possible to the original image F which will give us better results than in the case $Y = O$. We have observed that the operator $E(Y)$, defined in (3.5.2), behaves as the improvement of the picture Y. Further, our experience is that better values for Y (closer to F) produce better improvements. On the contrary, when the matrix Y is selected to be "far" from the original image, the improvement of Y is still worse with respect to the Moore-Penrose reconstruction (corresponding to the choice $Y = O$).

The method proposed in this section is an improvement of image restoration methods. It can not be considered as independent, but in symbiosis with another image restoration methods. Therefore, it has been tested against well known restoration methods. If the image restoration method is denoted by Y, then the improvement is denoted by $E(Y)$. We later compare values $ISNR(Y)$ against corresponding values $ISNR(E(Y))$ as well as values $PSNR(Y)$ against values $PSNR(E(Y))$. The possible choice for Y is any image restoration method. Among others, Y may coincide with any typical image restoration method, and even with some of iterative methods that present results of the restoration process with and without noise.

In the next statement we prove the basic property of the transformation E: the improvement $E(Y)$ of Y is only single-stage.

Proposition 3.3.2. *The operator E is idempotent on the set $\mathcal{R}(E)$, where $\mathcal{R}(E)$ denotes the range of E:*

$$E^2(Y) = E(E(Y)) = E(Y).$$

Dokaz. Proof follows from the facts that $G(H^T)^\dagger \left(I - H^\dagger H\right)$ is the zero $p \times s$ matrix and $I - H^\dagger H$ is idempotent. □

In order to estimate the computational complexity of the operator E we use standard notations from [16]. The complexity of the algorithm for computing the pseudoinverse of a given $m \times n$ matrix is denoted by $\mathrm{P}inv(m,n)$, and we identify it with the complexity of the standard *Matlab* function `pinv`. Later, by $\mathcal{A}(m,n)$ we denote the complexity of the addition/subtraction of two $m \times n$ matrices. The notation $\mathrm{T}(m,n)$ denotes the complexity of the transposition of $m \times n$ matrix, and by $\mathcal{M}(m,n,k) = mnk$ the complexity of multiplying $m \times n$ matrix with $n \times k$ matrix.

The complexity for computation of the value $F = G(H^T)^\dagger$ is equal to $\mathrm{Pinv}(s,m)+\mathcal{M}(p,m,s)$. The computational complexity of $E(Y) = \widetilde{F}$ is equal to

$$\mathrm{Pinv}(s,m) + \mathcal{M}(p,m,s) + \mathrm{T}(s,m) + \mathcal{M}(s,m,s) + \mathcal{M}(p,s,s) + \mathcal{A}(p,s),$$

where Y is appropriately selected matrix.

We propose a Haar reconstruction of the original image, as it is presented in the Section 2.6, as a candidate for the matrix Y.

3.4 Image deblurring process based on separable restoration methods

The method is based on the least squares solution of a matrix equation which models the linear motion blur and includes the Moore-Penrose inverse of the blurring matrix as well as an appropriately chosen matrix. Significant improvements are attained with respect to the classical approach based on the Moore-Penrose solution of certain matrix equations, which is investigated in [12]. Experimental evaluation and numerical results of the proposed method are described in Section 4.4.

3.4.1 Motivation and description of the method

The objective is to estimate the original image F row per row by exploiting a given row of a blurred image G and a priori knowledge of the degradation phenomenon H. Denote the ith row of the matrix F by $f_i = [f_{i,1}, \ldots, f_{i,s}]$ and ith row of G by $g_i = [g_{i,1}, \ldots, g_{i,m}]$. It is assumed that the linear motion is a local phenomenon, the imaging process captures all light and no additional noise is included. Under these assumptions, an arbitrary ith row of the blurred image can be related with the ith row of the original image [12]:

$$\begin{bmatrix} g_{i,1} \\ g_{i,2} \\ \vdots \\ g_{i,m} \end{bmatrix} = \begin{bmatrix} h_n & h_{n-1} & \cdots & h_1 & 0 & 0 & 0\dots & 0 \\ 0 & h_n & h_{n-1} & \cdots & h_1 & 0 & 0\dots & 0 \\ 0 & 0 & h_n & h_{n-1} & \cdots & h_1 & 0\dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & h_n & h_{n-1} & \cdots & h_1 \end{bmatrix} \begin{bmatrix} f_{i,1} \\ f_{i,2} \\ \vdots \\ f_{i,s} \end{bmatrix}, \tag{3.4.1}$$

where h_i are real numbers. The equation (3.4.1) can be rewritten as

$$g_i^T = H f_i^T, \quad g_i^T \in \mathbb{R}^m, \; f^T \in \mathbb{R}^s, \; H \in \mathbb{R}^{m \times s}. \tag{3.4.2}$$

The positive integer n indicates the length of linear motion blur in pixels and $s = m + n - 1$.

Equation (3.4.2) can be written in matrix form as

$$G = \left(HF^T\right)^T = FH^T, \quad G \in \mathbb{R}^{r \times m}, \; H \in \mathbb{R}^{m \times s}, \; F \in \mathbb{R}^{r \times s}. \tag{3.4.3}$$

The process of the separable blurring assumes that the blurring of the columns in the image is independent of the blurring of the rows. The separable blurring is modeled by two matrices, so that the relation between the original and blurred image can be expressed by the following relation [32]:

$$G = H_c F H_r^T, \quad G \in \mathbb{R}^{m_1 \times m_2}, \; H_c \in \mathbb{R}^{m_1 \times r}, \; F \in \mathbb{R}^{r \times s}, \; H_r \in \mathbb{R}^{m_2 \times s}, \tag{3.4.4}$$

where $s = m_2 + n_1 - 1$, $r = m_1 + n_2 - 1$, n_1 is the length of the horizontal blurring in pixels and n_2 is length of the vertical blurring in pixels.

The following matrix equations in X are used to define various generalized inverses of any $m \times n$ real matrix A:

$$(1) \quad AXA = A, \quad (2) \quad XAX = X, \quad (3) \quad (AX)^T = AX, \quad (4) \quad (XA)^T = XA,$$

where the superscript T denotes the transpose matrix. The set of matrices obeying the equations represented in $\mathcal{S}$ is denoted by $A\{\mathcal{S}\}$, for arbitrary sequence $\mathcal{S}$ of the elements from $\{1, 2, 3, 4\}$. Any matrix $X \in A\{\mathcal{S}\}$ is known as an $\mathcal{S}$-inverse of A and it is denoted by $A^{(\mathcal{S})}$ [3, 94]. the Moore-Penrose inverse $X = A^\dagger$ of A satisfies the set of the equations (1), (2), (3) and (4). The matrix equation (1) characterizes those generalized inverses X that are of use in analyzing the solutions of some matrix equations.

This section provides a new method for restoration of a blurred image using the set of the least squares solutions of the matrix equations presented in (3.4.4) and particularly in (3.4.3). Using well-known result about the general matrix equation $AXB = D$ from [3, 94] we immediately obtain general solution of the matrix equation (3.4.4) in the form

$$H_c^{(1)} G (H_r^T)^{(1)} + Y - H_c^{(1)} H_c Y H_r^T (H_r^T)^{(1)}. \tag{3.4.5}$$

The consistency condition

$$H_c H_c^{(1)} G (H_r^T)^{(1)} (H_r^T) = G$$

is evidently satisfied, since H_c is right invertible and H_r^T is left invertible. Without loss of generality, the pseudoinverses $H_c^\dagger$ and $(H_r^T)^\dagger$ can be used instead of $H_c^{(1)}$ and $(H_r^T)^{(1)}$, respectively.

Our main idea can be described as follows. A blurred image or its restoration we equalize with Y. Then we recommend the next transformation of the picture Y, which is based on the general solution of the form (3.4.5):

$$\begin{aligned} E_1(Y) = \breve{F} &= H_c^\dagger G(H_r^T)^\dagger + Y - H_c^\dagger H_c Y H_r^T (H_r^T)^\dagger \\ &= H_c^\dagger G(H_r^T)^\dagger + Y - H_c^\dagger H_c Y H_r^\dagger H_r. \end{aligned} \tag{3.4.6}$$

In the particular case $H_c = I$ the operator $E_1(Y)$ reduces to the operator $E(Y)$, which is investigated in Stanimirović et al, [83]. The present section is an extension of the ideas presented in [83], that is, to investigate possible values of the matrix Y in (3.4.6). The authors in [83] observed that the transformation (a least squares solution of (3.4.3))

$$E(Y) = \widetilde{F} = G(H^T)^\dagger + Y\left(I - H^\dagger H\right) = F + Y\left(I - H^\dagger H\right), \quad Y \in \mathbb{R}^{r\times s} \tag{3.4.7}$$

improves the restoration Y of the degraded image. The matrix $Y \in \mathbb{R}^{r\times s}$, which can be randomly chosen, can be determined in different ways. There are no specific conditions for that, any random matrix Y can be transformed into $E(Y)$, so an investigation for possible candidates will be presented in order to find the appropriate ones.

The authors in [83] observed that the best choice is $Y_{\rm opt} = F$ (which is an unacceptable choice). Also, our experience from [83] is that better values for Y (closer to F) produce better improvements. In the case $Y = O$, where O is the zero matrix of appropriate dimensions, $E(Y)$ produces the next approximation F of the original image F:

$$E(O) = F = G(H^T)^\dagger. \tag{3.4.8}$$

Numerical experiments based on the transformation (3.4.8) are investigated in [12, 13].

A number of image restoration algorithms is based on the usage of the Moore-Penrose inverse. The approach which assumes the condition $Y = O$ in (3.4.7) exploits the Moore-Penrose solution of the matrix equation, i.e. the least squares solution of minimal norm (see, for example [53, 71]). But, the minimal norm attribute

$$\|F\|_2 \leq \|\widetilde{F}\|_2 = \|F + Y\left(I - H^\dagger H\right)\|_2,$$

associated with the Moore-Penrose solution, may be in most of cases only the redundant property. Therefore, it is realistic to expect that the approach based on the opposite assumption $Y \neq O$ will show better performances in some cases. Only when the matrix Y is selected to be "far" from the original image, the improvement of Y is still worse with respect to the Moore-Penrose reconstruction (corresponding to the case $Y = O$). Some of the examples that confirm this expectation are studied in [83]. In addition, it is reasonable to expect that the minimal norm property

$$\|H_c^\dagger G(H_r^T)^\dagger\|_2 \leq \|H_c^\dagger G(H_r^T)^\dagger + Y - H_c^\dagger H_c Y H_r^\dagger H_r\|_2$$

is needless in the image restoration process. Our goal in [83] was to determine Y in such a way that the approximation $\widetilde{F}$ produces better values for ISNR and PSNR with respect to the solution F which is used in [12, 13]. Extending this idea, in the present section we investigate values for Y that provide better values for ISNR and PSNR values corresponding to $\breve{F}$ with respect to values that correspond to $H_c^\dagger G(H_r^T)^\dagger$.

Two main aims in this section may be featured as follows:

– Extend investigations of the operator $E(Y)$ from [83], corresponding to uniform linear motion blur, to the operator $E_1(Y)$, defined in (3.4.6) and corresponding to the separable motion blur.

– Continue the research of the operators $E(Y)$ and $E_1(Y)$ by choosing another appropriate values for the matrix Y.

The authors in [83] have shown that the operator $E(Y)$ is idempotent. In Proposition 3.4.1 we prove that the generalization $E_1(Y)$ of $E(Y)$ satisfies the same property.

Proposition 3.4.1. *Consider the blurring model* (3.4.4), *i.e.* F *is the original and* G *is the blurred image. Let* $\Phi(E_1)$ *be the set of fixed points of the operator* E_1 *and* $\mathcal{R}(E_1)$ *be the range of* E_1*. We have that*

$$\Phi(E_1) \supseteq \mathcal{R}(E_1) \cup \{F\}. \tag{3.4.9}$$

Proof. Firstly we show that the operator $E_1(Y)$ is idempotent, i.e.

$$E_1^2(Y) = E_1(Y).$$

The proof of this statement follows from

$$\begin{aligned} E_1^2(Y) = H_c^\dagger G(H_r^T)^\dagger + & \left[H_c^\dagger G(H_r^T)^\dagger + Y - H_c^\dagger H_c Y H_r^\dagger H_r\right] \\ & - H_c^\dagger H_c \left[H_c^\dagger G(H_r^T)^\dagger + Y - H_c^\dagger H_c Y H_r^\dagger H_r\right] H_r^\dagger H_r \end{aligned}$$

and several transformations based on elementary properties of the Moore-Penrose inverse.

In the rest of the proof we prove that $E_1(F) = F$. Using (3.4.4) we get

$$E_1(F) = H_c^\dagger (H_c F H_r^T)(H_r^T)^\dagger + F - H_c^\dagger H_c F H_r^\dagger H_r.$$

Using basic properties of the Moore–Penrose inverse we get $H_r^T(H_r^T)^\dagger = H_r^\dagger H_r$ and prove $E_1(F) = F$. $\square$

For appropriate choice of the matrix Y we propose, for example, reconstructions of blurred image by two commonly used filters from the collection of least-squares filters, namely the Wiener filter (WF) and the Constrained least-squares filter (CLS) [6]. We also propose the Symmetric Minimal Rank Solution (SMRS) of the inverse matrix problem $AX = B$ from [96] as a possible choice of the matrix Y. Another approach for the selection of the matrix Y is to use a moment base methods, such as the Haar basis and the Fourier basis. These methods are selected because of their capability to transform the intensity information of an image into spectral and spatial associated information, correspondingly. Short overview of some least squares filters for the image restoration as well as the SMRS solution of the matrix equation which models the image blurring process and also the moments approach are presented in Chapter 2. Some specific things related to SMRS is presented in the next subsection.

3.4.2 The symmetric minimal rank solution

We observe that the matrix equation (3.4.3) with respect to the unknown matrix F is one manifestation of the inverse matrix problem. The symmetric minimal rank solution (SMRS solution) of the inverse matrix problem, applied to the particular case (3.4.3), is used as a candidate for the matrix Y because it has a very interesting property with respect to the operator E.

Our motivation to use the symmetric minimal rank solution comes from the following two facts: firstly, the inverse matrix problem $AX = B$ is of the same general form as the mathematical model (3.4.3); secondly, the symmetric minimal rank solution of the matrix equation $AX = B$ with respect to the matrix A is given in [96]. Additionally, it was shown that symmetric minimal rank solution shows interesting properties with respect to the operator E. The SMRS solution is not applicable to the operator E_1.

We also propose the usage of the symmetric minimal rank solution (SMRS solution) of the matrix equation (3.4.3) in the role of the matrix Y. Why the SMRS solution, irrelevant for image restoration, is a good choice for Y? According to (3.4.9), so far we discovered the following subset of the fixed points set of the operator E: unallowable choice $Y = F$ and the choice $Y = E(Y_1)$, Y_1 arbitrary. In the next statement we prove that the property $E^2(SMRS) = E(SMRS)$ is also caused by the more strong property $E(SMRS) = SMRS$. In this way, $Y = SMRS$ is another fixed point of the operator E.

A simple comparison of the matrix equations $AX = B$ and (3.4.3) gives

$$A \equiv F, \;\; X \equiv H^T, \;\; B \equiv G. \tag{3.4.10}$$

The criterion for restoration of a blurred image that we are using is the optimal approximation to the set of the minimal rank solution $S_{\tilde{m}}$, given by (2.5.1), can be written alternatively:

$$\|F - \widetilde{F}\| = \min_{F_m \in S_{\tilde{m}}} \|F - F_m\|, \tag{3.4.11}$$

where F is given and $\widetilde{F}$ is restored image (denoted by SMRS). The solution for restored image, according to (2.5.6) from Proposition 2.5.1, is the following:

$$\widetilde{F} = SMRS = A_0 + U_2 P_1 P_1^T (A_{22}^* - A_{04}) P_1 P_1^T U_2^T. \tag{3.4.12}$$

Since in our case the matrix X is right invertible

$$XX^\dagger = H^T \left(H^T\right)^\dagger = I$$

we have

$$A_0 = BX^T = G\left(H^T\right)^\dagger = F, \tag{3.4.13}$$

where U_2, P_1, A_{22}^*, and A_{04} are presented above.

Theorem 3.4.1. *In the case $Y = SMRS$ we have $E(Y) = Y$.*

Proof. We just need to prove that in the case $Y = SMRS$ we have

$$G(H^T)^\dagger - YH^\dagger H = O.$$

By replacing Y from (3.4.12) and A_0 from (3.4.13), we get

$$\begin{aligned} G(H^T)^\dagger - YH^\dagger H &= G(H^T)^\dagger - \left(G\left(H^T\right)^\dagger + U_2 P_1 P_1^T (A_{22}^* - A_{04}) P_1 P_1^T U_2^T\right) H^\dagger H \\ &= G(H^T)^\dagger - G\left(H^T\right)^\dagger H^\dagger H - U_2 P_1 P_1^T (A_{22}^* - A_{04}) P_1 P_1^T U_2^T H^\dagger H. \end{aligned}$$

Since the equality $\left(H^T\right)^\dagger H^\dagger H = \left(H^T\right)^\dagger$ holds, we conclude that

$$G(H^T)^\dagger - YH^\dagger H = -U_2P_1P_1^T(A_{22}^* - A_{04})P_1P_1^TU_2^TH^\dagger H.$$

In order to prove that $G(H^T)^\dagger - YH^\dagger H = 0$ we will show that $U_2^TH^\dagger H = 0$.
Indeed,

$$U_2^TH^\dagger H = U_2^T \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma^{-1} & 0 \\ 0 & 0 \end{bmatrix} V^TV \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_1^T \\ U_2^T \end{bmatrix}$$

where the columns of U_1 form a basis for the range of H and the columns of U_2 form a basis for the kernel of H^T.
So, we have that

$$U_2^TH^\dagger H = U_2^T \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} U_1^T \\ 0 \end{bmatrix} = U_2^TU_1U_1^T.$$

Using the fact that $U_1U_1^T$ is a projection matrix on the range of H along the kernel on H^T (The SVD factorization is a special case of the URV factorization, see eg [57] pages 404 and 555) we get that

$$U_2^TU_1U_1^T = 0 \tag{3.4.14}$$

Therefore,

$$G(H^T)^\dagger - YH^\dagger H = 0$$

which is equivalent to $E(Y) = Y$. □

Regularization of the symmetric minimal rank solution

Since image restoration is an ill-posed inverse problem, some kind of regularization is of always very critical. This fact was the motivation to enforce a regularization for SMRS solution of the inverse matrix problem. The regularization is defined applying two weighting coefficients, denoted by α and β. For this purpose we consider (2.5.7) in the form

$$A_0 = BX^\dagger + A_{01},$$

where

$$A_{01} = (BX^\dagger)^T(I - XX^\dagger) + (I - XX^\dagger)BX^\dagger(XX^\dagger BX^\dagger)^\dagger(BX^\dagger)^T(I - XX^\dagger). \tag{3.4.15}$$

Then the SMRS regularization is defined by

$$\begin{aligned} A_0' &= BX^\dagger + \alpha A_{01}, \\ \tilde{A}' &= A_0' + \beta\left(U_2P_1P_1^T(A_{22}^* - A_{04})P_1P_1^TU_2^T\right). \end{aligned} \tag{3.4.16}$$

For the sake of simplicity, we denote the SMRS regularization, defined in (3.4.15)-(3.4.16), by $SMRS_{\alpha,\beta}$.
Two important cases in the regularization $SMRS_{\alpha,\beta}$ can be distinguished:
– $SMRS_{0,0} = E(Y = 0) = \widetilde{F} = G(H^T)^\dagger$;
– $SMRS_{1,1} = SMRS$.
Numerical results show that some choices of the parameters α, β provide better restorations with respect to usual SMRS, which is confirmed by $ISNR(SMRS_{\alpha,\beta}) > ISNR(SMRS)$.

3.5 Improving the Tikhonov and TSVD image deblurring methods

The main goal of this section is further investigation of the algorithm, introduced in [83], that allows us to remove a linear motion blur from images. In this section we showed that it is possible to use Tikhonov (TIK) and Truncated Singular Value Decomposition (TSVD) image restoration methods as two possible candidates for the matrix Y. In this way we show that it is possible to improve the restoration of images in terms of restoration when using only Tikhonov and TSVD methods for the restoration of the images.

In Chapter 2 we restate two image restoration methods which we use in Section 3.5: Truncated Singular Value Decomposition (TSVD) and Tikhonov method (TIK), presented in [32].

3.5.1 Motivation and presentation of the method

Mathematical model, in which the linear motion is a local phenomenon and no additional noise is included, relates an arbitrary ith row $g_i = [g_{i,1}, \dots, g_{i,m}]$ of the blurred image $G \in \mathbb{R}^{p\times m}$ with corresponding ith row $f_i = [f_{i,1}, \dots, f_{i,m}]$ of the original image F_0, by the following matrix equation [23, 32]:

$$g_i^T = \left[\begin{array}{c|c|c} H_{-1} & H_0 & H_1 \end{array}\right] \left[\begin{array}{c} f_{i,-1} \\ \hline f_i^T \\ \hline f_{i,1} \end{array}\right], \tag{3.5.1}$$

where

$$f_{i,-1} = \begin{bmatrix} w_{i,1} \\ \vdots \\ w_{i,u} \end{bmatrix}, \quad f_{i,1} = \begin{bmatrix} v_{i,1} \\ \vdots \\ v_{i,u} \end{bmatrix}.$$

Elements $w_{i,1}, \dots w_{i,u}$ and $v_{i,1}, \dots v_{i,u}$ of the right hand vector are *boundary pixels*. The boundary pixels left of the horizontal line, which are added above the initial vector $f_i^T \in \mathbb{R}^m$, depends of the nature and direction of the movement. The u boundary pixels right of the horizontal line are added below the vector f_i [32]. Further, the matrices

$$H_{-1} \in \mathbb{R}^{m\times u}, \; H_1 \in \mathbb{R}^{m\times u}, \; H_0 \in \mathbb{R}^{m\times m}$$

are determined implicitly in the block matrix

$$H = \left[\begin{array}{c|c|c} H_{-1} & H_0 & H_1 \end{array}\right],$$

which is defined by

$$H = \left[\begin{array}{ccc|ccccccccc|ccc}
h_u & \cdots & h_1 & h_0 & h_{-1} & \cdots & h_{-u} & & & & & & & & \\
 & \ddots & \vdots & h_1 & h_0 & \ddots & \vdots & \ddots & & & & & & & \\
 & & h_u & \vdots & h_1 & \ddots & h_{-1} & \cdots & h_{-u} & & & & & & \\
 & & & h_u & \cdots & \ddots & h_0 & \ddots & \vdots & \ddots & & & & & \\
 & & & & h_u & & h_! & \ddots & h_{-1} & & h_{-u} & & & & \\
 & & & & & \ddots & \vdots & \ddots & h_0 & \ddots & & h_{-u} & & & \\
 & & & & & & h_u & & h_1 & \ddots & & \vdots & h_{-u} & & \\
 & & & & & & & \ddots & \vdots & \ddots & h_0 & h_{-1} & \vdots & \ddots & \\
 & & & & & & & & h_u & \cdots & h_1 & h_0 & h_{-1} & \cdots & h_{-u}
\end{array}\right].$$

The quantities h_i are real numbers, $n-1=2u$. The real matrix H is with dimension $m \times s$. The index n indicates the length of linear motion blur in pixels and $s = m+n-1 = m+2u$, $m \gg n$.

If we know the degradation phenomenon H, the objective is to estimate the original image $F_0 \in \mathbb{R}^{p\times m}$ row per row (vector f_i^T) from the row of a blurred image (vector g_i^T). Then it is possible to consider the block matrix

$$F = \left[\ F_{-1} \ \middle|\ F_0^T \ \middle|\ F_1\ \right] \in \mathbb{R}^{p\times s},$$

whose blocks are of the order $F_{-1} \in \mathbb{R}^{p\times u}$, $F_1 \in \mathbb{R}^{p\times u}$, $F_0 \in \mathbb{R}^{p\times m}$ and F_{-1} (resp. by F_1) represent the matrix whose columns are $f_{i,-1}$ (resp. $f_{i,1}$). Different boundary conditions: zero, periodic and reflective are presented in Section 3.5.1.

In papers [83, 85] we derive and investigate a new method for restoration of a blurred image using the set of least squares solutions of the matrix equation (3.3.3). The least squares solutions in this section are generated using the Moore–Penrose inverse inverse. The Moore-Penrose inverse of a matrix $A \in \mathbb{C}^{m\times n}$ is the unique matrix, denoted by $A^\dagger$, satisfying the following four matrix equations:

$$(1)\quad AXA = A \qquad (2)\quad XAX = X \qquad (3)\quad (AX)^* = AX \qquad (4)\quad (XA)^* = XA.$$

A matrix X is called an $\{i,j,k\}$–inverse of A (with $i,j,k \in \{1,2,3,4\}$) if X satisfies the ith, jth and kth Penrose equations. Particularly, any $\{1\}$–inverse (or g–inverse) of A is noted by $A^{(1)}$ and finds applications in solving linear systems of equations or matrix equations.

The general solution of the the matrix equation (3.3.3) is given by

$$E(Y) = \widetilde{F} = G(H^T)^\dagger + Y\left(I - H^\dagger H\right), \tag{3.5.2}$$

where the matrix $Y \in \mathbb{R}^{p\times s}$ can be randomly chosen. Our original intention was to use values for Y as close as possible to the original image F which will give us better results than in the case $Y = O$, which produces the matrix

$$\hat{F} = G(H^T)^\dagger.$$

The choice $Y = O$ is investigated in [12, 13]. Furthermore, we have observed in [83, 85] that the operator $E(Y)$ frequently behaves better improvement of blurred image with respect to restoration contained in Y. Therefore, the method proposed in [83, 85] is an improvement of image restoration methods.

There is no practical reason to consider the operator $E(Y)$ as independent, but in symbiosis with the image restoration method Y. Therefore, it has been tested against well known restoration methods. If the image restoration method is denoted by Y, then the improvement is denoted by $E(Y)$. We later compare values $ISNR(Y)$ against corresponding values $ISNR(E(Y))$ as well as values $PSNR(Y)$ against values $PSNR(E(Y))$. Comparison of $ISNR$ and $PSNR$ values corresponding to $E(Y)$ with other image restoration methods is not of interest; it is only meaningful to compare mutually values corresponding to restorations Y and $E(Y)$. The possible choice for Y is any image restoration method.

Regularization methods for computing stabilized solutions to the ill-posed problems occur often enough in science and engineering to make it meaningful to present a general framework for their numerical treatment [31]. Matrices with ill-determined numerical rank, on the other hand, are obtained from underlying ill posed problems where the concept of rank has no instinctive interpretation. Examples of such problems are: digital image restoration, solution of integral equations in solid state physics, inverse Radon and Laplace transformation [31]. Continuing investigation from [83, 85], we propose two direct regularization methods: Truncated Singular Value Decomposition and Tikhonov reconstruction of the original image as two possible choices for the matrix Y.

Through experimental results presented in Section 4.5 we confirm that if we select for the matrix Y the results obtained from the TIK and TSVD methods we will improve the restoration of images, comparing with the using only TIK and TSVD methods for the restoration of the images.

Chapter 4

Experimental and numerical results

In image restoration the improvement in quality of the original image $F(n_1, n_2)$ over the recorded blurred one $G(n_1, n_2)$ is measured by the signal-to-noise ratio (*SNR*) improvement. The *SNR* of the recorded (blurred) image is defined as follows in decibels:

$$SNR_G = 10\log_{10}\left(\frac{\sum_{n1,n2} F^2(n_1,n_2)}{\sum_{n1,n2}(G(n_1,n_2)-F(n_1,n_2))^2}\right).$$

The *SNR* of the restored image $\widetilde{F}(n_1, n_2)$ is similarly defined by

$$SNR_{\widetilde{F}} = 10\log_{10}\left(\frac{\sum_{n1,n2} F^2(n_1,n_2)}{\sum_{n1,n2}(\widetilde{F}(n_1,n_2)-F(n_1,n_2))^2}\right).$$

Then, the improvement in *SNR* is given by

$$ISNR = SNR_{\widetilde{F}} - SNR_G = 10\log_{10}\left(\frac{\sum_{n1,n2}(G(n_1,n_2)-F(n_1,n_2))^2}{\sum_{n1,n2}(\widetilde{F}(n_1,n_2)-F(n_1,n_2))^2}\right). \tag{4.0.1}$$

The improvement in *SNR* is basically a measure that expresses the reduction of disagreement with the ideal image when comparing the distorted and restored image. Note that all of the above signal-to-noise measures can only be computed in case when the ideal image is available, i.e., in an experimental setup or in a design phase of the restoration algorithm. The unit of the $ISNR$ is given in dB.

The peak signal-to-noise ratio ($PSNR$) is defined as the ratio between a signal's maximum power and the power of the signal's noise. In image processing and analysis it has been extensively used as a criterion to measure the quality of reconstructed images that have usually been compressed. Each picture element (pixel) has a color value that can change when an image is compressed and then uncompressed. In this section we use the following definition for $PSNR$:

$$PSNR = 20\log_{10}\left(\frac{\max\{F(n_1,n_2)\}}{\sqrt{(1/(n_1 * n_2))\sum_{i=0}^{n_1}\sum_{j=0}^{n_2}\left(F(i,j)-\widetilde{F}(i,j)\right)^2}}\right), \tag{4.0.2}$$

where $\max\{F(n_1, n_2)\}$ is the largest possible value of the original image and the denominator is defined as the root mean square difference between the original and the reconstructed images. The unit of the $PSNR$ is also given in dB.

4.1 Application of the pseudoinverse computation in reconstruction of blurred images

In this section we present numerical results which are obtained by testing the method proposed in Algorithm 3.1.1 (MP method) on X-ray images. In order to confirm the efficiency, we compared our method with three recently announced methods for computing the Moore-Penrose inverse of the matrix H. Summarizing, the following four methods are compared:

1. The MP method,
2. $Pappas1$ method, defined by the MATLAB function `ginv.m` from [46],
3. $Pappas2$ method, defined by the MATLAB function `qrginv.m` from [14],
4. *Courrieu* method from [17].

The experiments are done using *Matlab* programming language [38] on an Intel(R) CPU T2130 @ 1.86 GHz 1.87 GHz 32-bit system with 2 GB of RAM memory. Tests are made for several images of dimensions $r \times m$. The index l that takes values between 10 and 100 is the varying parameter for a given image.

Also we compared the efficiency of four different strategies of image restoration: the approach based on the Moore-Penrose inverse, the Wiener filter (WF), the constrained least-squares (*CLS*) filter, and the Lucy-Richardson (*LR*) algorithm. For the implementation of the Wiener filter, the constrained least-squares filter, and Lucy-Richardson algorithm we used built–in functions from the *Matlab* package [27].

Figure 4.1.1 presents one practical example for restoring blurred X-ray image. The image is taken from the results obtained from Google Image search with the keywords "X-ray image". The picture in Figure 4.1.1 called *Original image* shows the original X-ray image. The image is divided into $r = 843$ rows and $m = 1050$ columns. To prevent loosing information from the boundaries of the image, we assumed zero boundary conditions, which implies that values of the pixels of the original image F outside the domain of consideration are zero (black). This choice is natural for X-ray images since the background of these images is black. The picture named as *Degraded image* presents the degraded X-ray image with a uniform horizontal motion of length $l = 80$. The pictures named as *Moore-Penrose Inverse*, *Wiener Restored Image*, *Constrained LS Restored Image* and *Lucy-Richardson Restored Image* denote the images obtained after the application of the corresponding restoration algorithms. From Figure 4.1.1, it is clear that the MP method produces the best result.

The difference in quality of the restored images among the three methods (WF, CLS and *LR*) is insignificant, and can hardly be seen by human eye. For this reason, the *ISNR* is applied in order to compare the quality of the restored images. Figure 4.1.2 shows the corresponding *ISNR* value of the restored images as a function of l for the MP method and the other mentioned classical methods. This figure illustrates that the MP method for image restoration has the best quality of the restored image with respect to the other methods.

As we have already mentioned, the main advantage of the proposed method for computing the Moore-Penrose inverse, is the time required to obtain the restored image compared to other methods for computing the Moore-Penrose inverse. Figure 4.1.3 shows the corresponding computational time, denoted by *t(sec)*, as a function of $l < 100$ pixels for the MP method and the other three considered methods for computing the Moore–Penrose inverse. Obviously, the minimal computational time is reached by the MP method.

The next two examples refer to the case when, in the inception, the image is degraded by including an image noise and later it is followed by a uniform linear blur. The corresponding

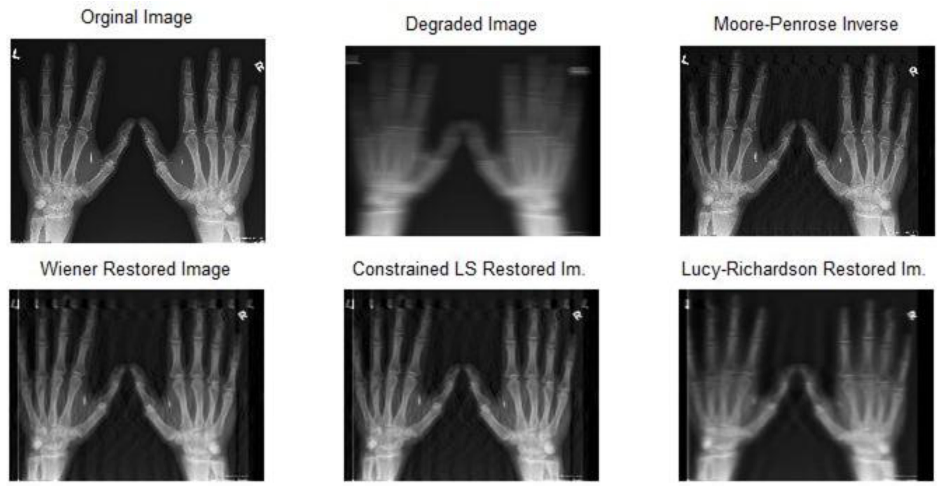


Figure 4.1.1: Removal of blur, caused by a uniform horizontal motion.

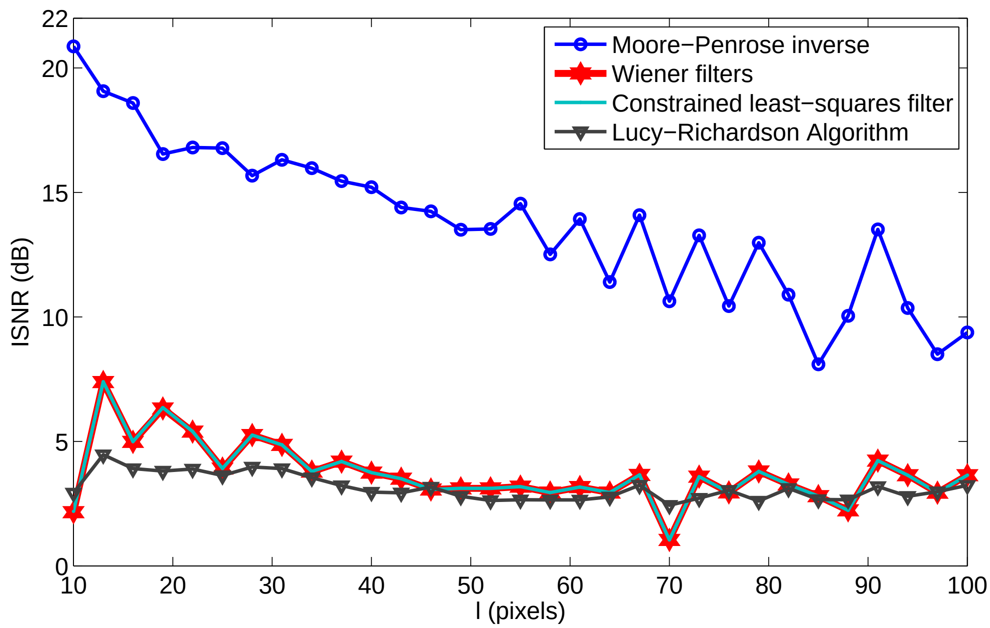


Figure 4.1.2: Improvement in signal-to-noise-ratio vs. length of the blurring process.

mathematical model generalizes the horizontal blurring process presented with (3.1.2) and it becomes

$$G_N = (F + N)H^T = F_N H^T, \tag{4.1.1}$$

where N is an additive noise and G_N is the blurred noisy image. To obtain approximation of the original image, we apply two steps:
1. Calculate the restored matrix F_N by using (3.1.5), to produce $\tilde{F}_N = G_N(H^\dagger)^T$;
2. Obtain the restored image $\tilde{F}$ by applying filtering process on the image $\tilde{F}_N$ obtained in Step 1.

Similarly, we can formulate a process in the case of having two ways degraded image (with

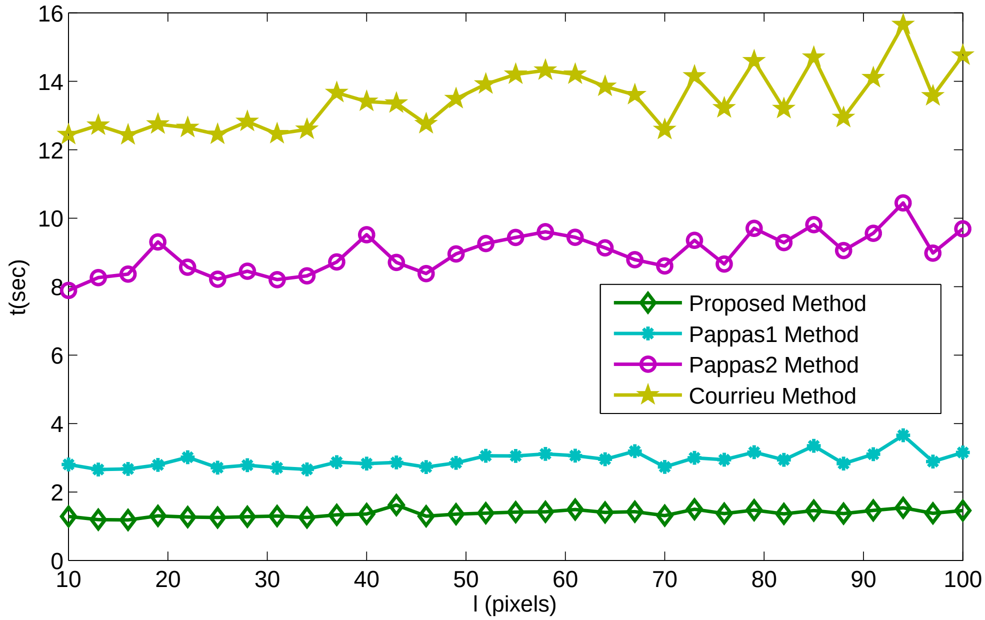


Figure 4.1.3: Computational time vs. length of the blurring process.

noise and vertical blur). The noisy image, the blurred noisy image and the restored images obtained by using different methods are presented in Figure 4.1.4.

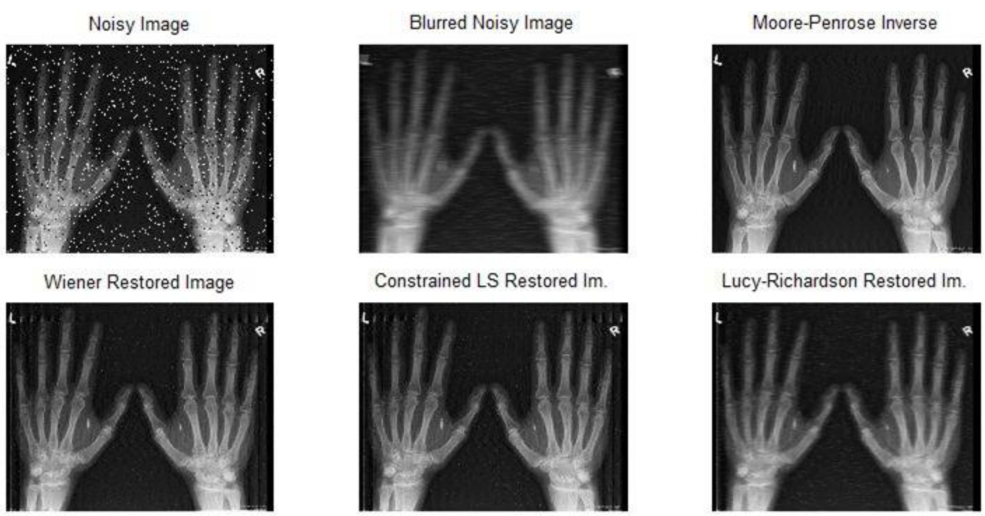


Figure 4.1.4: Removal of blur, caused by noise and uniform horizontal motion with $r = 843, n = 1050$ and $l = 40$.

The results for $ISNR$ and the peak signal-to-noise ratio ($PSNR$) [6] for the original image given in Figure 4.1.1, are presented in Figure 4.1.5 and Figure 4.1.6. The original image is degraded in two ways: by "salt and pepper" (white and black) noise with noise density of 0.03 and after that it is blurred by a uniform linear motion with l pixels. The 2-D median filtering is used for the image restoration. The image restoration procedures based on the Moore-Penrose inverse, applied to images degraded by both the motion blur and noise, are more reliable and accurate compared to other image restoration methods.

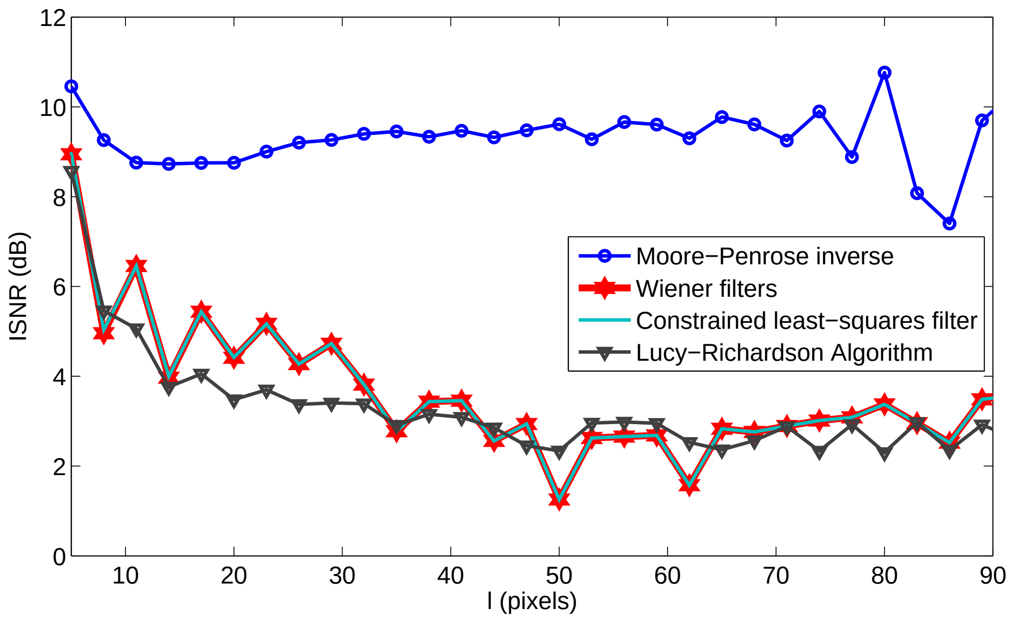


Figure 4.1.5: Improvement in signal-to-noise-ratio vs. length of the blurring process.

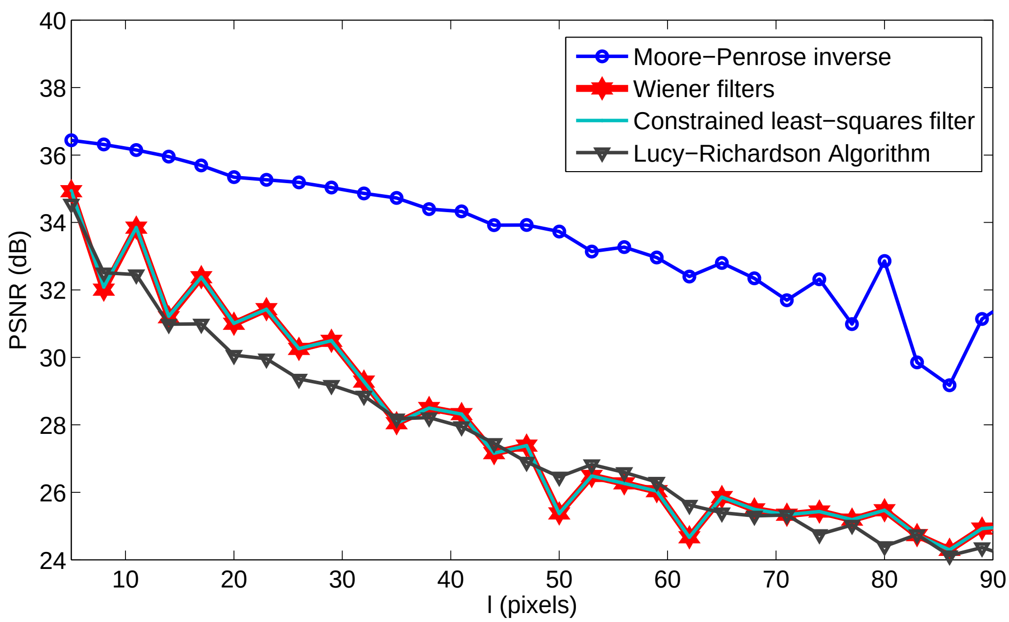


Figure 4.1.6: Peak signal-to-noise ratio vs. length of the blurring process.

The $ISNR$ and $PSNR$ values presented in Figure 4.1.5 and Figure 4.1.6 indicate that MP method is the best among the other methods.

Similar numerical results are generated when the image is blurred by a non-uniform motion detrmined with the relations (3.1.10) and (3.1.11). The case when the image is blurred by a non-uniform blurring with parameters $l_c = 35$ and $l_r = 25$, is presented in Figure 4.1.7.

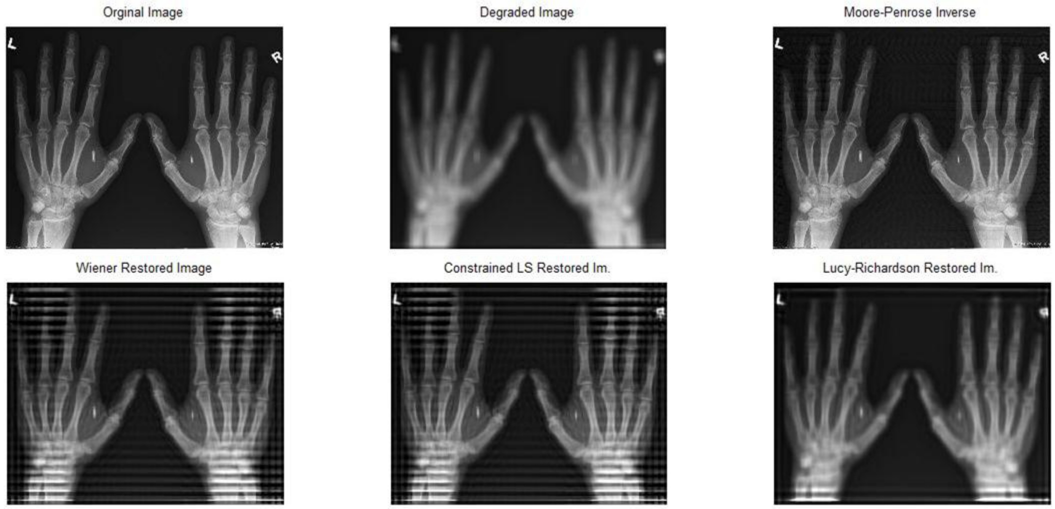


Figure 4.1.7: Removal of blur, caused by the non-uniform blurring model with $l_c = 35$ and $l_r = 25$.

A confirmation that the proposed MP method is faster than the other methods for computing the Moore-Penrose inverse is illustrated in Figure 4.1.8.

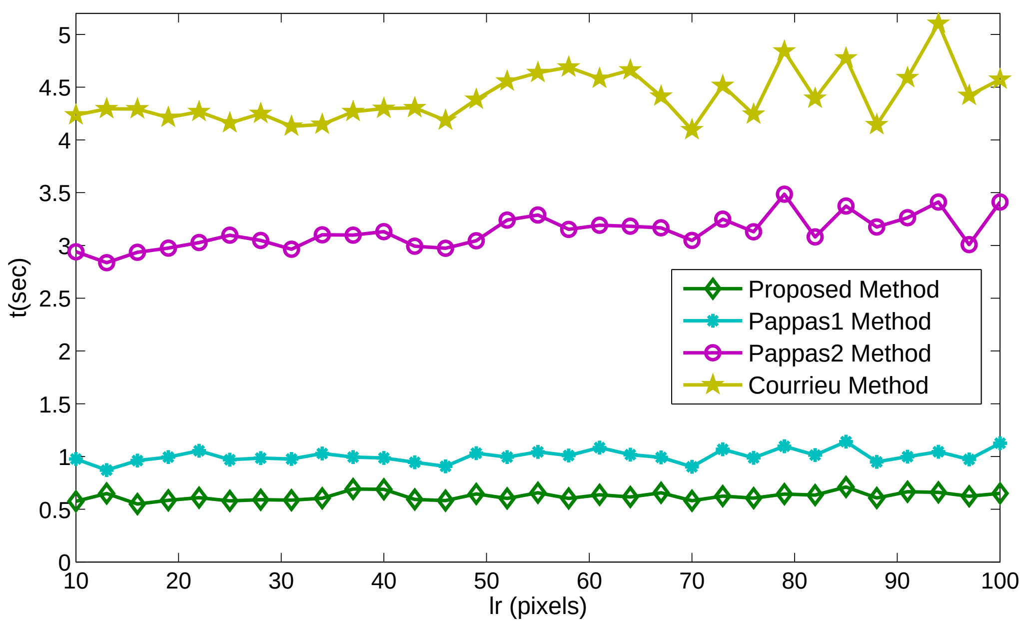


Figure 4.1.8: Computational time versus variable length l_r and $l_c = 25$ of the blurring process.

The results presented in Figure 4.1.8 are made on an Intel(R) Core(TM) i5 CPU M430 @ 2.27 GHz 64/32-bit system with 4 GB RAM memory.

4.2 Application of partitioning method on specific Toeplitz matrices

We compared the CPU time required for computation of the Moore–Penrose inverse of Toeplitz matrix H resulting from linear systems convolution kernel. Experiments are done using `MATLAB` programming package [38] on an Intel(R) Core(TM) i5 CPU M430 @ 2.27 GHz 64/32-bit system with 4 GB RAM memory. Since the algorithms we compared with are implemented in `MATLAB`, we also chose `MATLAB` as a framework for the implementation of proposed algorithms.

In Fig. 4.2.1 and Fig. 4.2.2 we present the results which refer to the computational time t(sec) needed to compute the Moore-Penrose inverse $H^{\dagger}$ as a function of the length of the blurring process $l \leq 90$ (pixels). Values incorporated in these figures are obtained for randomly generated matrix of dimensions 1000×1200, which corresponds to a randomly generated image of the same dimensions, which is blurred by Gaussian function. CPU times illustrated on Fig. 4.2.1 and Fig. 4.2.2 confirm that the proposed BPM method for computing $H^{\dagger}$ is faster than the other considered methods.

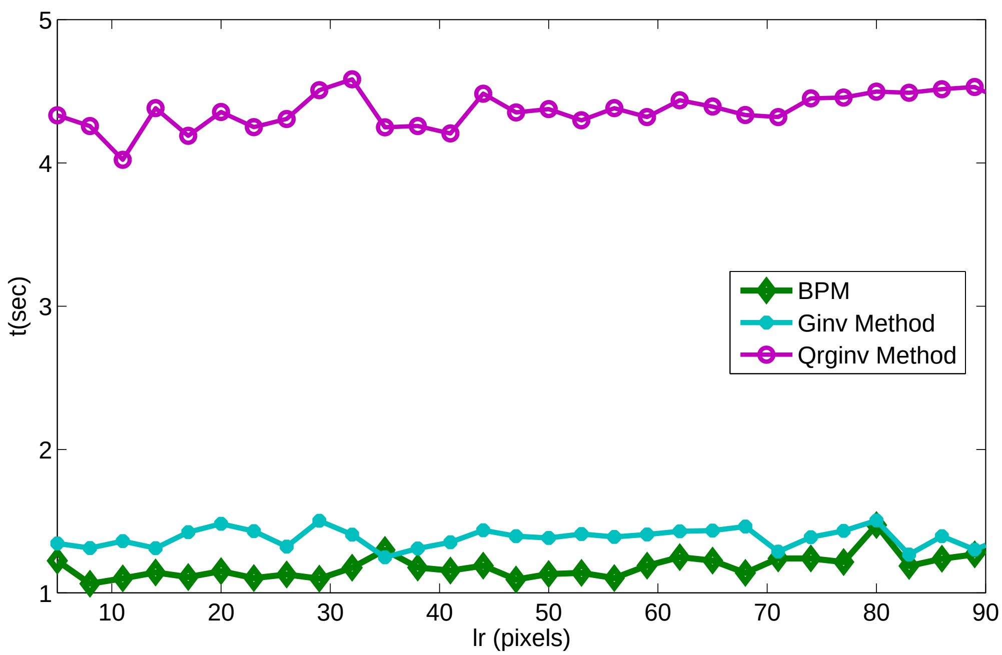


Figure 4.2.1: CPU time for computing MP inverse of the random matrix H versus l_r ($l_c = 30$).

It is easy to observe that the Block Partitioning Method overcomes *Ginv* and *QrGinv* methods. On the other hand, the authors in [46] concluded that the method *Ginv* is faster with respect to *Courrieu* method. Thus, after the modifications described before, partitioning method becomes the fastest compared to three considered methods for computing the Moore–Penrose inverse.

In addition we compare the accuracy of the results of the our method with the other three methods. We use *ginvtest* function from [46] and the accuracy was examined with the matrix 2-norm in error matrices corresponding to the four Moore-Penrose equations. In Table 4.2.1 we present average errors for different values of the parameter s, regarding the four Penrose equations. The presented average values were generated by varying the parameter l from 5

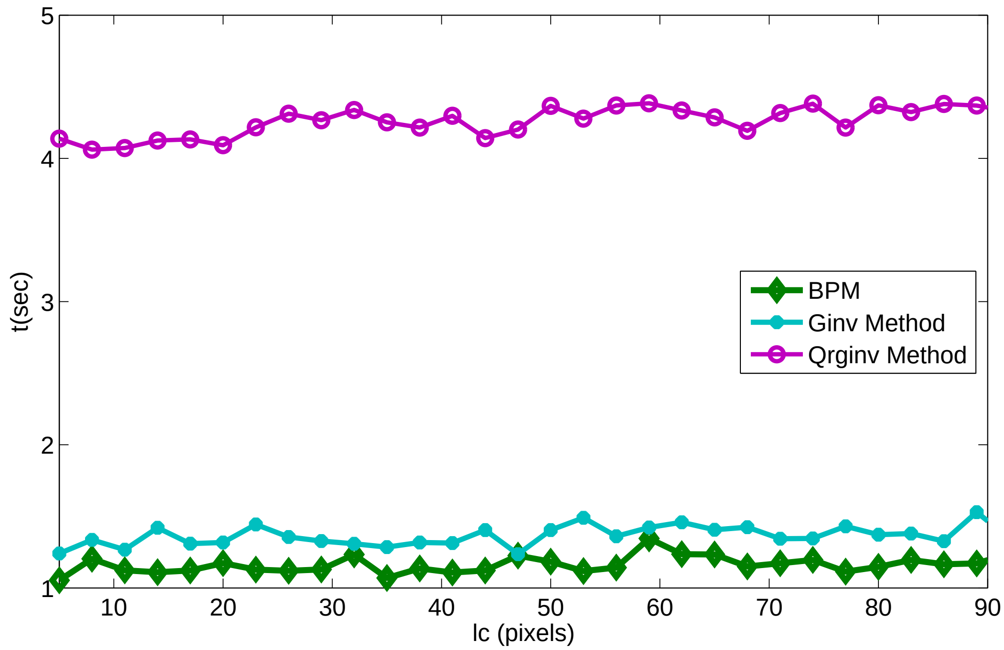


Figure 4.2.2: CPU time for computing MP inverse of the random matrix H versus l_c ($l_r = 25$).

to 90 with step 1. Based on the results shown in Table 4.2.1 the following conclusions are imposed. Greatest norms and thus the worst results are generated by *Courrieu Method*. *BPM* produces the best results for the Penrose equations (1), (2) and (3). *Ginv* and *Qrginv* methods give slightly better results regarding the matrix equation (4) with respect to *BPM* method.

Despite the differences in the matrix norms that can be observed from Table 4.2.1, differences in ISNR and PSNR values are negligible for all considered methods for computing the Moore–Penrose inverse. Therefore, the CPU time for computing $H^\dagger$ is only reliable criterion for the comparison of different methods for computing the Moore–Penrose inverse.

4.2.1 Restoring blur and noise

In this subsection attention has been paid to the model images degraded by a sequence of mutually independent operations. First, the noise is imposed to the image and after that the noisy image is blurred by the non-uniform Gaussian function. In this case the mathematical model of the non-uniform blurring process presented by (3.2.1) becomes

$$G_N = H_C(F + N)H_R^T = H_C F_N H_R^T, \tag{4.2.1}$$

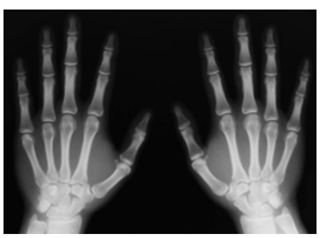
where G_N is blurred noisy image and N is an additive noise. Two steps are used to restore the original image:

1. Calculate the restored matrix $\widetilde{F}_N = H_C^\dagger G_N (H_R^\dagger)^T$ of F_N;

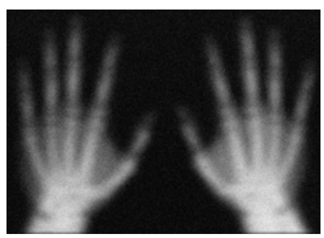
2. Generate image $\widetilde{F}$ by applying the filtering process on the image $\widetilde{F}_N$. Depending on the type of noise we use a rotationally symmetric Gaussian lowpass filter or two dimensional median filter.

Table 4.2.1: Average error results regarding the four Moore-Penrose equations for $5 \leq l \leq 90$.

s	2-norm in error matrices	*BPM*	*Ginv*	*Qrginv*	*Courrieu Method*
100	$\|TT^{\dagger}T - T\|_2$	1.4630×10^{-14}	1.9417×10^{-14}	1.1482×10^{-14}	1.0614×10^{-10}
	$\|T^{\dagger}TT^{\dagger} - T^{\dagger}\|_2$	3.0370×10^{-12}	7.8286×10^{-9}	1.6753×10^{-11}	6.3123×10^{-8}
	$\|TT^{\dagger} - (TT^{\dagger})^*\|_2$	9.8858×10^{-14}	1.1352×10^{-11}	9.9422×10^{-13}	5.7959×10^{-8}
	$\|T^{\dagger}T - (T^{\dagger}T)^*\|_2$	3.0602×10^{-13}	1.8363×10^{-14}	1.0475×10^{-13}	5.4743×10^{-11}
200	$\|TT^{\dagger}T - T\|_2$	1.4206×10^{-14}	1.9703×10^{-14}	1.1243×10^{-14}	1.0124×10^{-10}
	$\|T^{\dagger}TT^{\dagger} - T^{\dagger}\|_2$	2.9713×10^{-12}	7.9161×10^{-9}	1.7050×10^{-11}	5.9056×10^{-8}
	$\|TT^{\dagger} - (TT^{\dagger})^*\|_2$	9.6917×10^{-14}	1.1066×10^{-11}	9.9735×10^{-13}	5.4223×10^{-8}
	$\|T^{\dagger}T - (T^{\dagger}T)^*\|_2$	2.8939×10^{-13}	1.8014×10^{-14}	1.0298×10^{-13}	5.2727×10^{-11}
300	$\|TT^{\dagger}T - T\|_2$	1.3947×10^{-14}	1.9369×10^{-14}	1.1166×10^{-14}	1.0056×10^{-10}
	$\|T^{\dagger}TT^{\dagger} - T^{\dagger}\|_2$	2.9527×10^{-12}	7.7109×10^{-9}	1.6426×10^{-11}	5.8595×10^{-8}
	$\|TT^{\dagger} - (TT^{\dagger})^*\|_2$	9.5872×10^{-14}	1.0968×10^{-11}	9.5566×10^{-13}	5.3950×10^{-8}
	$\|T^{\dagger}T - (T^{\dagger}T)^*\|_2$	2.8583×10^{-13}	1.7938×10^{-14}	1.0321×10^{-13}	5.2229×10^{-11}
400	$\|TT^{\dagger}T - T\|_2$	1.4104×10^{-14}	1.9736×10^{-14}	1.1157×10^{-14}	1.0066×10^{-10}
	$\|T^{\dagger}TT^{\dagger} - T^{\dagger}\|_2$	2.9505×10^{-12}	7.8669×10^{-9}	1.7028×10^{-11}	5.8529×10^{-8}
	$\|TT^{\dagger} - (TT^{\dagger})^*\|_2$	9.6935×10^{-14}	1.0957×10^{-11}	9.8089×10^{-13}	5.3809×10^{-8}
	$\|T^{\dagger}T - (T^{\dagger}T)^*\|_2$	2.9034×10^{-13}	1.7916×10^{-14}	1.0235×10^{-13}	5.2246×10^{-11}
500	$\|TT^{\dagger}T - T\|_2$	1.4107×10^{-14}	1.9210×10^{-14}	1.1180×10^{-14}	9.9086×10^{-11}
	$\|T^{\dagger}TT^{\dagger} - T^{\dagger}\|_2$	2.8297×10^{-12}	7.2584×10^{-9}	1.6637×10^{-11}	5.7496×10^{-8}
	$\|TT^{\dagger} - (TT^{\dagger})^*\|_2$	9.4099×10^{-14}	1.0957×10^{-11}	9.6248×10^{-13}	5.2747×10^{-8}
	$\|T^{\dagger}T - (T^{\dagger}T)^*\|_2$	2.9047×10^{-13}	1.7940×10^{-14}	1.0084×10^{-13}	5.2127×10^{-11}

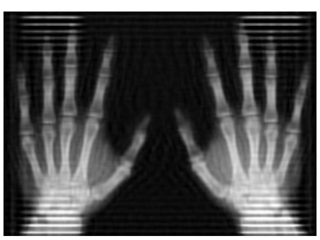


(a) Original image



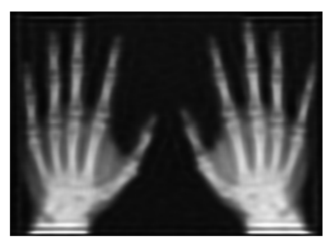
(b) Blurred noisy

(c) Moore-Penrose restore



(d) Wiener filter restore

(e) Tikhonov Regularization restore



(f) Lucy-Richardson restore

Figure 4.2.3: Removal of blur, caused by Gaussian model with $l_c = 25$ and $l_r = 45$, on a X-ray image.

Fig. 4.2.3 presents practical example of restoring blurred X-ray image. The image is taken from the results obtained after the Google Image search with the keyword "X-ray image" and location of the image is on
`http://www.tetburyhospital.co.uk/admin/subroutines/blob.php?id=13&blob=ablob`.

Fig. 4.2.3 demonstrates the efficiency of four different methods for the image restoration: the Moore-Penrose inverse based approach, Wiener filter (WF), Lucy-Richardson (LR) algorithm and Tikhonov Regularization (TR) method. For the implementation of the Wiener filter and Lucy-Richardson algorithm we used incorporated built-in functions from the `MATLAB` package. For Wiener filter we use `MATLAB` function with two input parameters: blurred noisy image G_N and point-spread function PSF. For Lucy-Richardson algorithm we have used additional parameter for the number of iterations with constant value 10. Implementation of Tikhonov Regularization method is based on Kronecker decomposition and the code presented in [32].

On Fig. 4.2.3 we illustrate the original, the blurred noisy and the restored images obtained by different methods. Figure denoted by *Original image* from Fig. 4.2.3 shows the original X-ray image. The image is divided into $r = 750$ rows and $m = 1050$ columns. To prevent loosing information from the boundaries of the image, we assumed zero boundary conditions, which implies that values of the pixels of the original image F outside of the image window are zero. This choice is natural for X-ray images since the background of these images is black. The pixels of the original image are degraded by the Gaussian white noise of mean 0 and variance 0.01 and later blurred by non-uniform Gaussian function according to model (4.2.1). For filtering we use a rotationally symmetric Gaussian low pass filter of size 3 with standard deviation 45.

The difference in quality of the restored images regarding three methods (Moore-Penrose, Wiener and the Tikhonov) is insignificant, and can hardly be seen by a human eye. For this reason, we use a common method for comparing restored images, i.e. we analyze the so called, improved signal-to-noise ratio (*ISNR*). The results for the parameters $ISNR$ [6], presented on Fig. 4.2.4, show that the restoration of the serial degraded images with the Moore-Penrose inverse is more reliable and accurate than restoration with other mentioned methods. In the Fig. 4.2.4 we used the $s = l_r/2$ and $s = l_c/2$ for non-uniform blurring process. The graph marked with *MP inverse* illustrates numerical values generated by an arbitrary method for computing the Moore–Penrose inverse.

Similar results are generated for another values of the parameter s. To illustrate this fact, in the Fig. 4.2.5 are presented results for (*ISNR*) corresponding to the choice $s = 30$.

For confirmation of our results on the next two figures we present results for the Dice Coefficient (DC) as a measure of the similarity between sets. Rang of the DC is from 0 to 1, where 0 indicates the sets are disjoint and 1 indicates the sets are identical [20, 18]. Parameters used in the Fig. 4.2.6 and Fig. 4.2.7 are the same with Fig. 4.2.4 and Fig. 4.2.5, appropriate.

Since we know that the deblurring process is in favor of the Moore-Penrose inverse, in the sequel we only compare the computational time of the methods which are based on the Moore-Penrose inverse approach. Thus, we compare the CPU time required by our method with the CPU time required by previously mentioned methods. The computational time needed to restore the degraded X-ray image by means of these methods which use the Moore-Penrose inverse approach, is shown in the Fig. 4.2.8 and the Fig. 4.2.9. For a given image the varying parameter is the parameter l_r (l_c) that takes values between 5 and 90.

As it is expected, the proposed method shows better performances with respect to the other tested methods.

Obviously, the proposed method is not only restricted to restoration of blurred X-ray images, but also can be used for other practical implementations, such as deblurring images arising in Automatic Number Plate Recognition (ANPR) systems. We assume that the blur that appears in images from ANPR systems is caused by "salt and paper" noise and non-uniform Gaussian model, given by (4.2.1). In "salt and pepper" noise, the corrupted pixel may take just one of two different values: black or white. These values are taken randomly with noise density equal to 0.05.

To prevent loosing information from the boundaries of the image, we assumed periodic

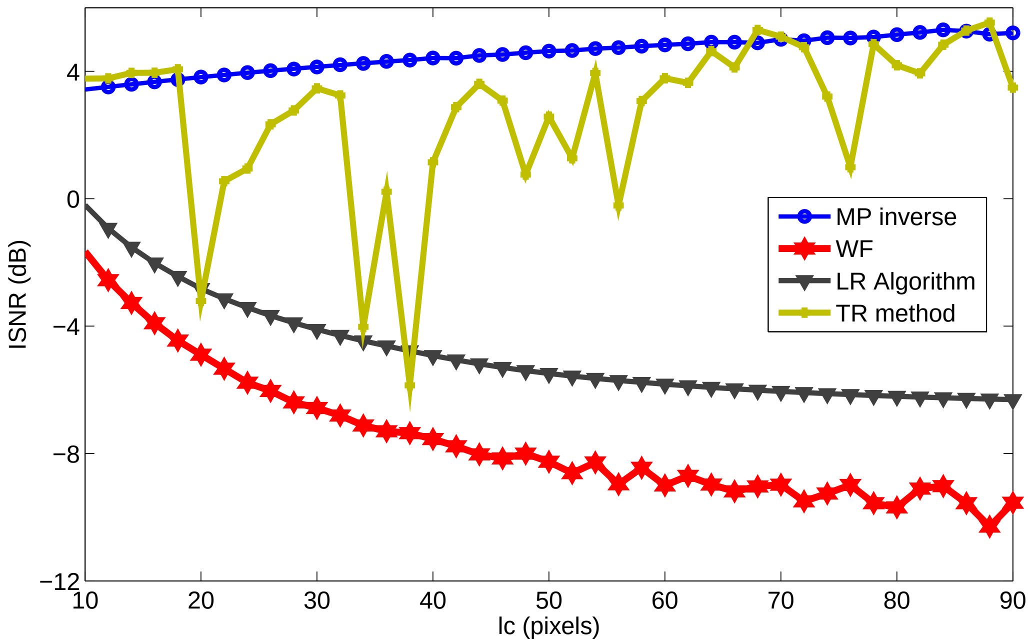


Figure 4.2.4: ISNR versus l_c for the removal of blur given by the model (4.2.1) ($l_r = 35$).

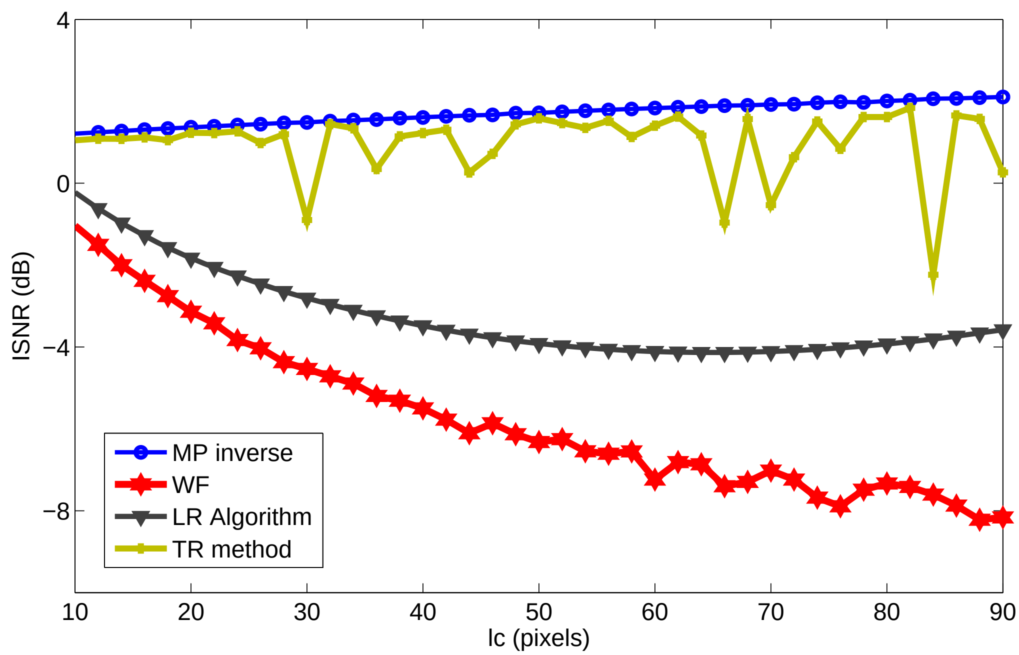


Figure 4.2.5: ISNR versus l_c for the removal of blur given by the model (4.2.1) ($l_r = 35$ and $s = 30$).

boundary conditions for ANPR images. Fig. 4.2.10 presents the results obtained by restoring an image from the ANPR system with dimensions 1023×1250.

ANPR image is taken from implemented system of automatic recognition of license plates of the Customs Administration in Serbia.

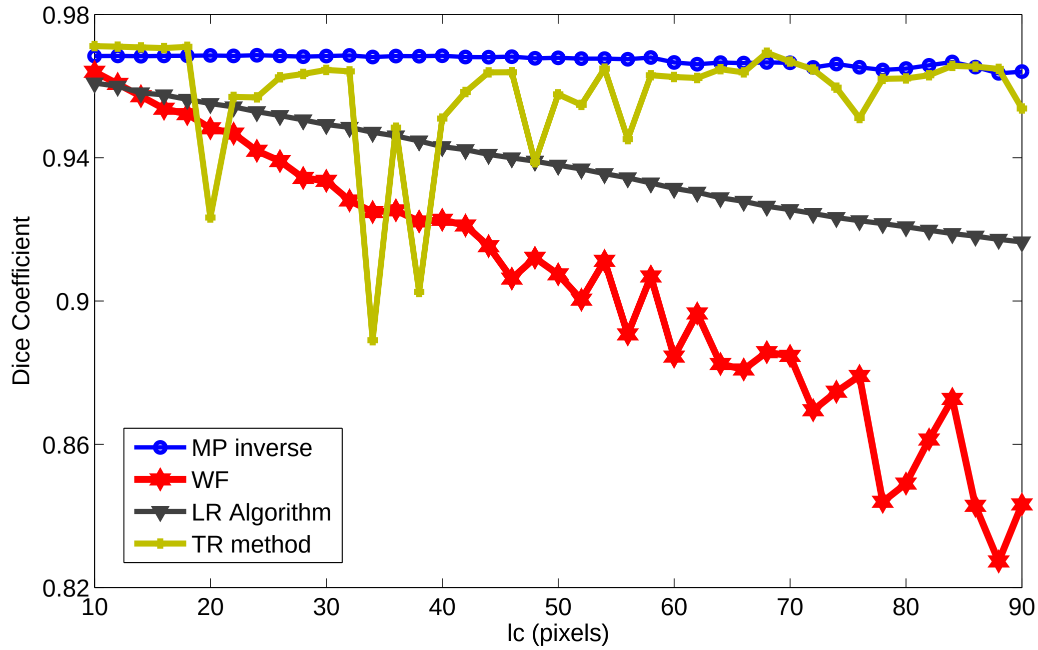


Figure 4.2.6: DC versus l_c for the removal of blur given by model (4.2.1) ($l_r = 35$).

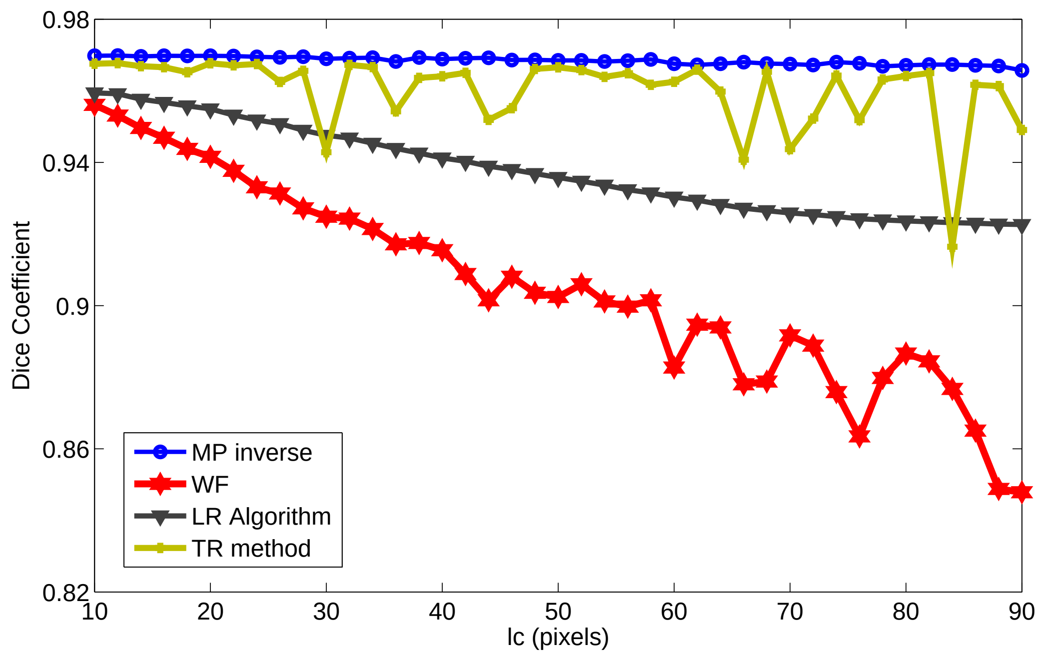


Figure 4.2.7: DC versus l_c for the removal of blur given by model (4.2.1) ($l_r = 35$ and $s = 30$).

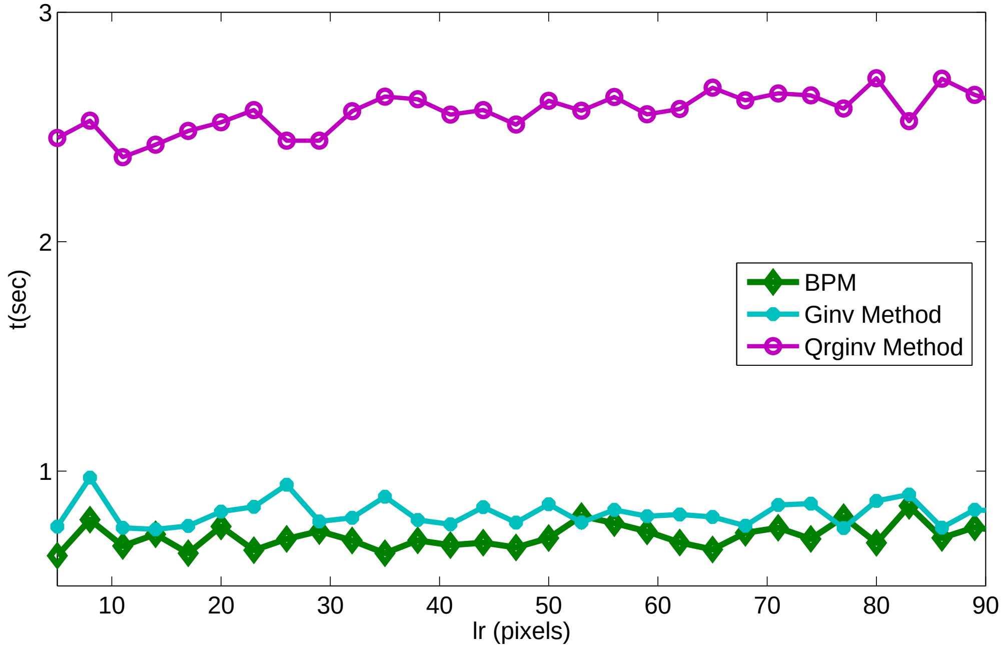


Figure 4.2.8: CPU time for removing of blur caused by Gaussian function and noise versus l_r ($l_c = 25$).

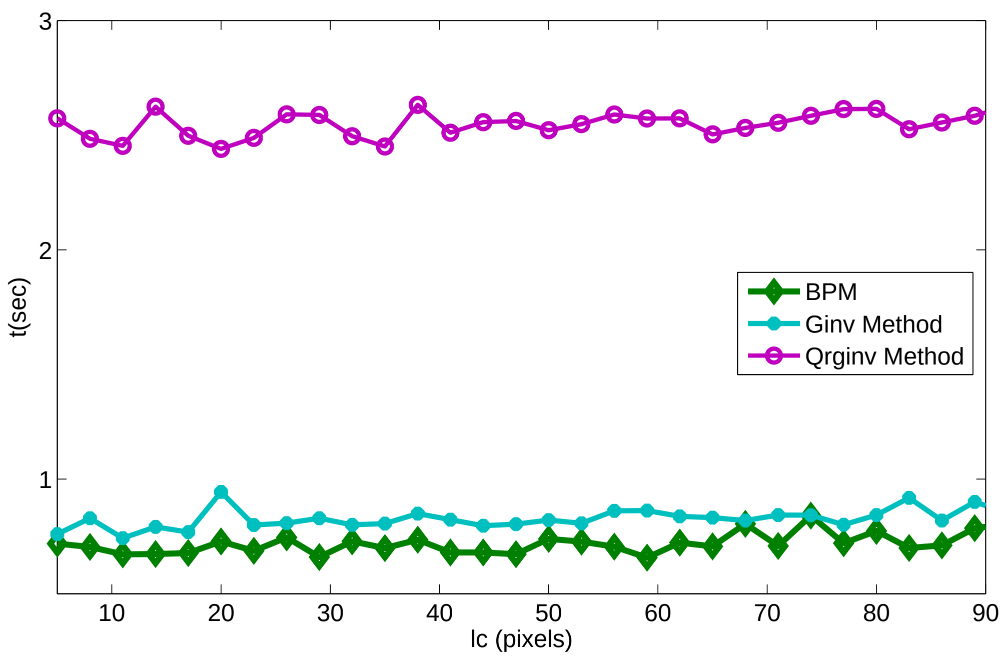


Figure 4.2.9: CPU time for removing of blur caused by Gaussian function and noise versus l_c ($l_r = 35$).

(a) original image (b) blurred noisy image (c) Moore-Penrose restored image

Figure 4.2.10: Removal of blur, caused by "salt and paper" noise and Gaussian function ($l_c = 25$, $l_r = 40$).

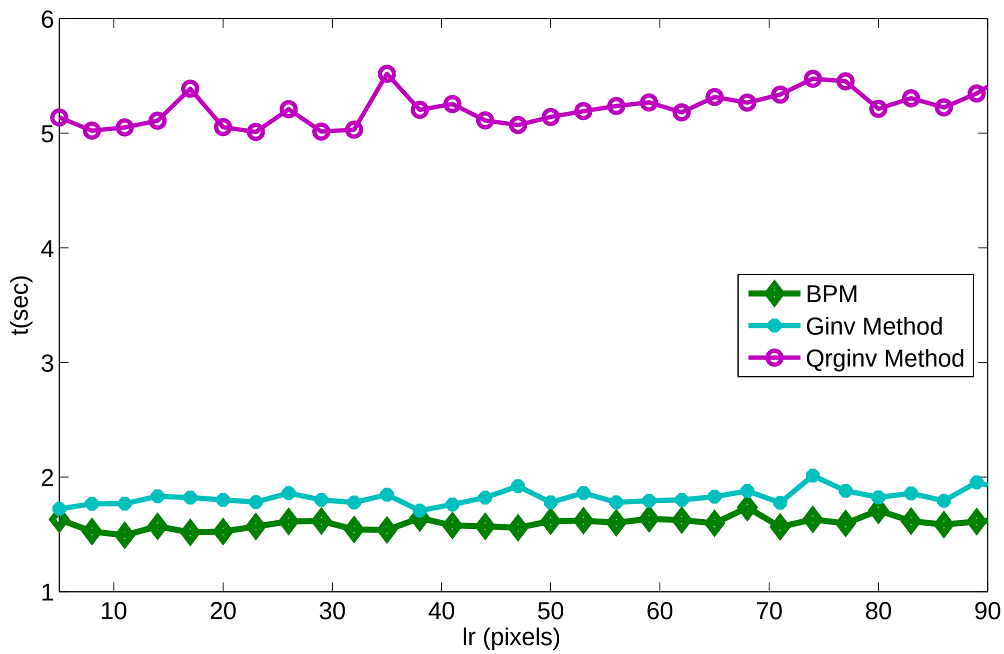


Figure 4.2.11: Time versus l_r in removal of blur given by the model (4.2.1) and "salt and paper" noise ($l_c = 35$).

Fig. 4.2.11 and Fig. 4.2.12 decidedly approves that the time required to obtain a restored ANPR image using the proposed method is again smallest with respect to other considered methods.

Also we wanted to confirm our previous findings in terms of speed for our method and other methods on a standard `MATLAB` image Lena. On Fig. 4.2.13 and Fig. 4.2.14 are presented results for Lena image and periodic boundary condition, which degraded by the Gaussian white noise of mean 0 and variance 0.01 and blurred by non-uniform Gaussian function (4.2.1). Rotationally symmetric Gaussian low pass filter of size 3 with standard deviation 45 is used for filtering.

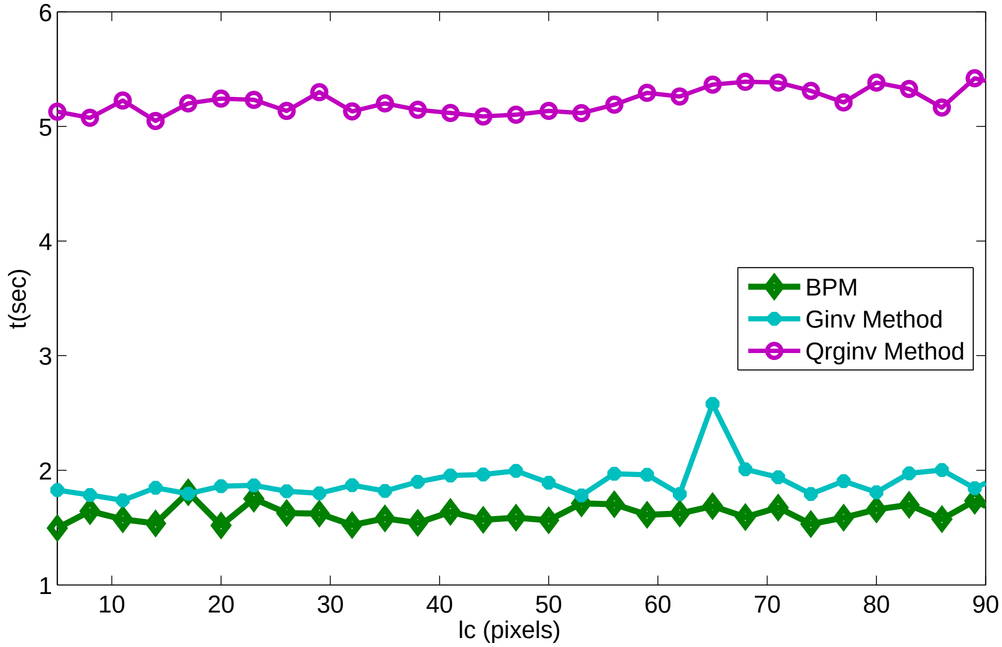
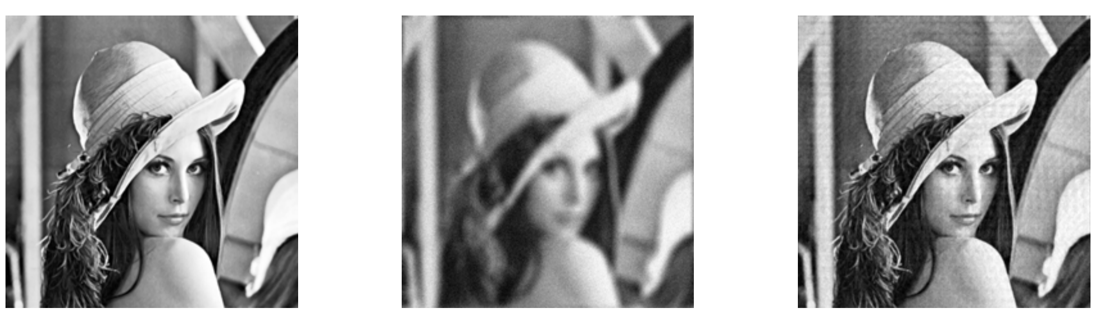


Figure 4.2.12: Time versus l_c in removal of blur given by the model (4.2.1) and "salt and paper" noise ($l_r = 40$).



(a) original image (b) blurred noisy image (c) Moore-Penrose restored image

Figure 4.2.13: Removal of blur, caused by Gaussian model with $l_c = 40$ and $l_r = 35$, on a Lena image.

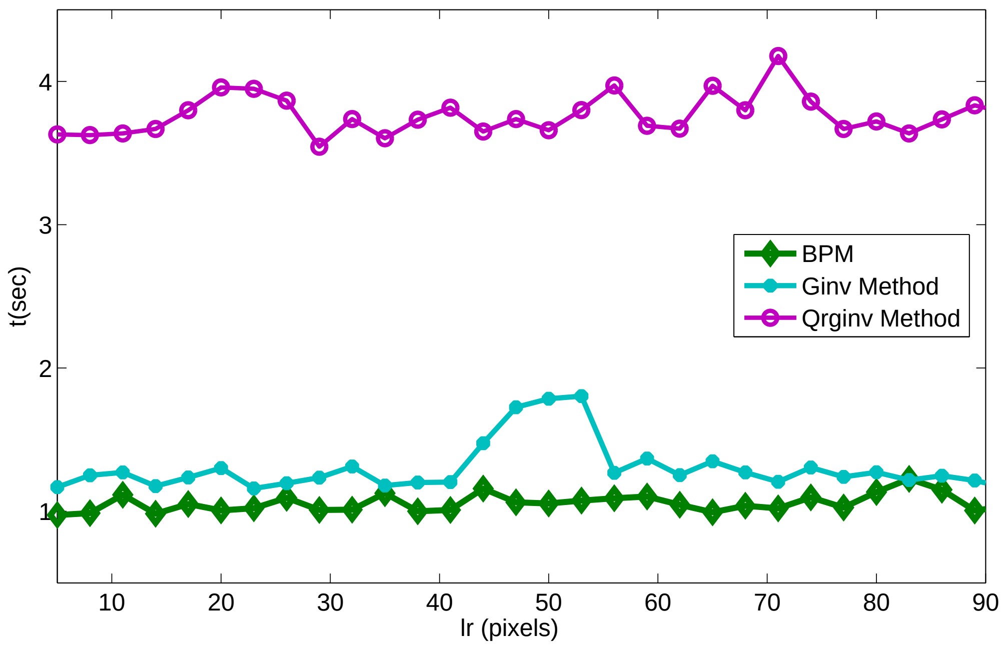


Figure 4.2.14: Time versus l_r in removal of blur given by model (4.2.1) and Gaussian noise ($l_c = 50$), for a Lena image.

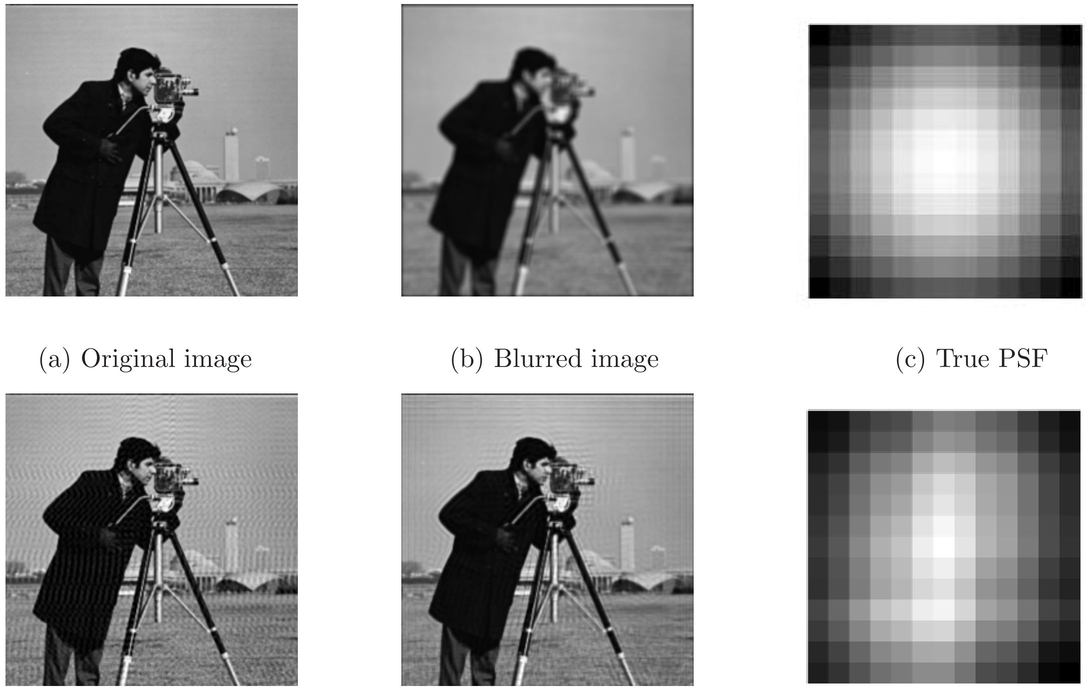
4.2.2 Blind deconvolution

So far in the section we have assumed that the convolution kernel PSF (in our case matrices H_r and H_c) is known. Let us suppose that the PSF is unknown or, at best, that it can be poorly determined. The technique of recovering the target scene from a blurred image in the presence of a poorly determined or unknown PSF is also known as blind deconvolution.

In the following, we show how the proposed block partitioning method can be applied in the case when the PSF (or matrices H_r and H_c) is unknown. Therefore, the first task is to see how to estimate PSF, and later to obtain the unknown matrices H_r and H_c, from the real-life blurred image. For the sake of simplicity, in the sequel, instead of matrices H_r and H_c (which are given by (3.2.9)) we consider only matrix H. Since, our main goal is not to deal with blind deconvolution process we only present one illustrative example. In order to estimate the unknown PSF from the given blurred image we use the built-in `MATLAB` function `deconvblind()`. This function deconvolves given image F using the maximum likelihood algorithm, returning both the deblurred image G and a restored PSF. Once we find a two-dimensional PSF we need to compute one-dimensional vector h which completely determines the matrix H.

We start with standard image "cameraman.tif". Similarly, as in example presented by `MathWorks`, in order to make the process of restoring the image easier, we assume the presence of the blur without additional presence of the noise. The example can be found on the following address: `http://www.mathworks.com/products/image/`
`examples.html?file=/products/demos/`
`shipping/images/ipexblind.html`.

The blur is simulated by convolving a Gaussian filter with the true image. The Gaussian filter then represents a point spread function.



(a) Original image (b) Blurred image (c) True PSF

(d) Moore-Penrose restore (e) Restored by deconvblind (f) Restored PSF

Figure 4.2.15: Removal of blur, based on convolution estimation approach, caused by Gaussian PSF on Cameraman image.

```
F = imread('cameraman.tif');
PSF=fspecial('gaussian',11,10);
Blurred = imfilter(I,PSF);
INITPSF = ones(size(PSF));
WT = zeros(size(I));
WT(5:end-4,5:end-4) = 1;
[G P] = deconvblind(Blurred,INITPSF,40,[],WT);
```

We took the standard values for the input parameters of function `deconvblind()`. Last parameter `WEIGHT` (shorter `WT`) specifies which pixels in the input image F are considered in the restoration. We assumed that the boundary pixels are excluded from consideration, by assigning it a value of 0 in the `WT` array. In Fig. 4.2.15 (c) and (f) we can see the original PSF and also the estimated PSF (denoted by P). As we mentioned, from two-dimensional matrix P we have to find the one-dimensional vector h. The only pixels that we need, for this purpose, are situated on the diagonal of the matrix P. Thus, in the process of image reconstruction by our method the only information that is relevant to us is the main diagonal of the matrix P. What we are, actually, interested in is how good pixels on the diagonal are estimated, while for the rest of the estimated matrix P we don't care. In Fig. 4.2.15 (d) and (e) one can also found the restored images obtained by our method and by function `deconvblind()`, respectively.

If we compare the restored image from Fig. 4.2.15 (d) and (e) with two parameters $ISNR$ and $PSNR$ (peak signal-to-noise ratio) [6], the value for the Moore-Penrose restore is higher than obtained by the function `deconvblind()`. These results are presented in the following Table 4.2.2.

Table 4.2.2: Results for $ISNR$ and $PSNR$ for Fig. 4.2.15 (d) and (e).

Parameters	Moore-Penrose restore	`deconvblind()` restore
ISNR (dB)	5.2906	3.8689
PSNR (dB)	26.2945	24.8728

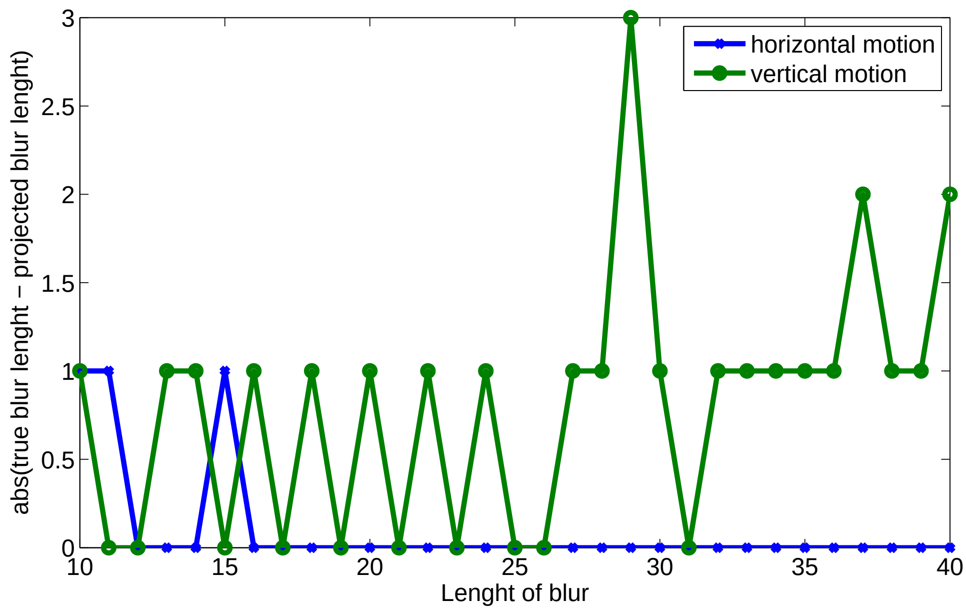


Figure 4.2.16: Error of the estimated blur length with 2D cepstral method for the standard MATLAB image Cameraman.

Motion blur is an effect you will see in images of scenes where objects are moving. It is mostly noticeable when the exposure is long, or if objects in the scene are moving rapidly. Our method we can implement for restore an image that has been blurred by uniform linear motion. In the next part we present the results when is implement the full system of image restoration which is consist of two phase: motion estimation and image deblurring with the parameter estimate in the first phase. To estimate the length of motion blur l, 2-D cepstral methods are employed. This is how we represent Cepstrum Domain [48]:

$$Cep(g(x,y)) = infFT\{\log(FT(g(x,y)))\} \tag{4.2.2}$$

The Figure 4.2.16 illustrate that when we used 2-D cepstral methods for motion blur estimation we have low level of error for estimation of the parameter l.

The restored images produced from the two phase image restoration system and by the function `deconvblind()` are presented on the Fig. 4.2.17 for Cameraman image and Fig. 4.2.18 for Lena image. Results for parameters $ISNR$ and $PSNR$ are presented in the Table 4.2.3.

(a) Blurred image

(b) Moore-Penrose restored image

(c) deconvblind restored image

Figure 4.2.17: Removal of blur, caused by horizontal motion and length of blur ($l = 49$) for Cameraman image.

(a) Blurred image

(b) Moore-Penrose restored image

(c) deconvblind restored image

Figure 4.2.18: Removal of blur, caused by horizontal motion and length of blur ($l = 28$) for Lena image.

Table 4.2.3: Results for $ISNR$ and $PSNR$ for Fig. 4.2.17 and Fig. 4.2.18.

Image	Parameters	Moore-Penrose restore	`deconvblind()` restore
Cameraman	ISNR (dB)	8.8557	3.4342
Cameraman	PSNR (dB)	27.9935	22.5720
Lena	ISNR (dB)	8.2999	3.5891
Lena	PSNR (dB)	29.6519	24.9411

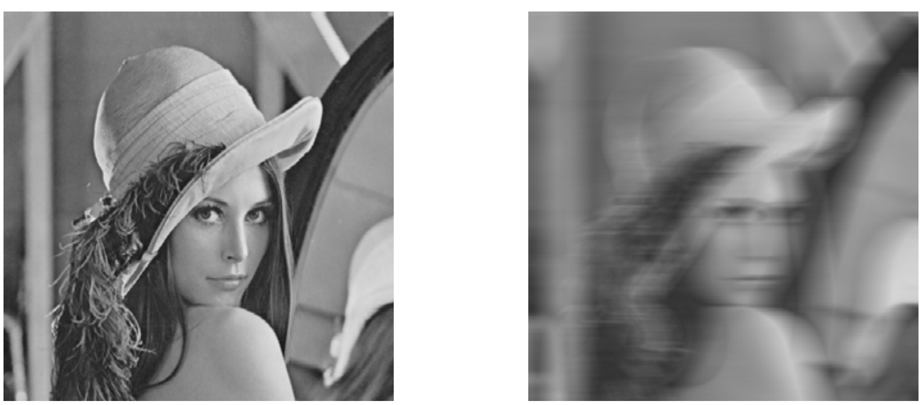
4.3 Removal of blur in images based on least squares solutions

In this section we have applied the proposed least squares transformation, given by equations (3.5.2), on the image restoration process. Our basic idea is to use the output (2.6.1) as a candidate for the matrix Y. We present numerical results for Y and $E(Y)$, compare the obtained results with each other and with results generated by the Moore–Penrose solution F, defined in (3.4.8), of the matrix system (3.5.2). The uniform linear motion is assumed in the experiments, which implies $h_i = 1/n$, $i = -u, \ldots, u$. The experiments have been performed using Matlab programming language [38] on an Intel(R) Core(TM)2 Duo CPU T5800 @ 2.00 GHz 32-bit system with 2 GB of RAM memory running on the Windows Vista Business Operating System.

4.3.1 Application of the method on blurred images

Several tests were performed in order to evaluate the effectiveness of the method. The first set of tests are aimed at the calculation of the improvement and the peak signal-to-noise ratio for the reconstruction of image for various values n of the blurring process. The second set of tests is related to the same criteria of simulating the reconstruction method but this time against different values of moments. Finally, the last set of tests aimed at accenting the reconstruction error between the original image and the reconstructed image for various values of Haar coefficients

The standard test image 'Lena' is shown in Figure 4.3.1(a). The blurred image that has been degraded by a uniform linear motion in the horizontal direction (that is usually results of camera panning or fast object motion and modeled by the matrix equations (3.3.1) and (3.3.3)) is presented in Figure 4.3.1(b).



(a) Lena (b) Blurred image of Lena

Figure 4.3.1: (a) Image of Lena, (b) Blurred image of Lena by the uniform linear motion of length $n = 50$.

The blurred images of Lena, where the periodic and reflective boundary conditions are imposed to the original image, are shown in figures 4.3.2(a) and (b), respectively.

(a) Blurred image of Lena under periodic BCs

(b) Image of Lena under reflective BCs

Figure 4.3.2: Blurred image of Lena for length $n = 50$ and different BCs.

Figures 4.3.3(a), and 4.3.3(b) show the Haar basis moment reconstructed image for the cases of $k = l = 55$ and $k = l = 195$, respectively.

(a) $k = l = 55$

(b) $k = l = 195$

Figure 4.3.3: Reconstruction of Lena from Haar basis moments.

We consider the $ISNR$ and the $PSNR$ values for increasing values $n = 10, 12, 14, \ldots, 100$ for the cases of the zero, periodic and the reflective boundary conditions. Graphical representations on figures 4.3.4–4.3.9 correspond to the following values:
(a) $ISNR(E(B_{195,195}))$, where $B_{195,195}$ is the Haar based reconstructed image for $k = l = 195$;
(b) $ISNR(E(B_{55,55}))$, where $B_{55,55}$ denotes the Haar based reconstructed image for $k = l = 55$;
(c) $ISNR(E(O))$;
(d) Wiener filter;
(e) Lucy-Richardson algorithm.
The Wiener filter and the Lucy-Richardson algorithm are two standard methods for image restoration. Wiener filter is selected from the family of least-squares filters and the Lucy-Richardson algorithm is iterative method [6, 26, 38].

Figures 4.3.4, 4.3.5 and 4.3.6 illustrate the $ISNR$ values for increasing values n of the blurring processes, in the case when the zero, periodic and reflective boundary conditions, respectively, are assumed.

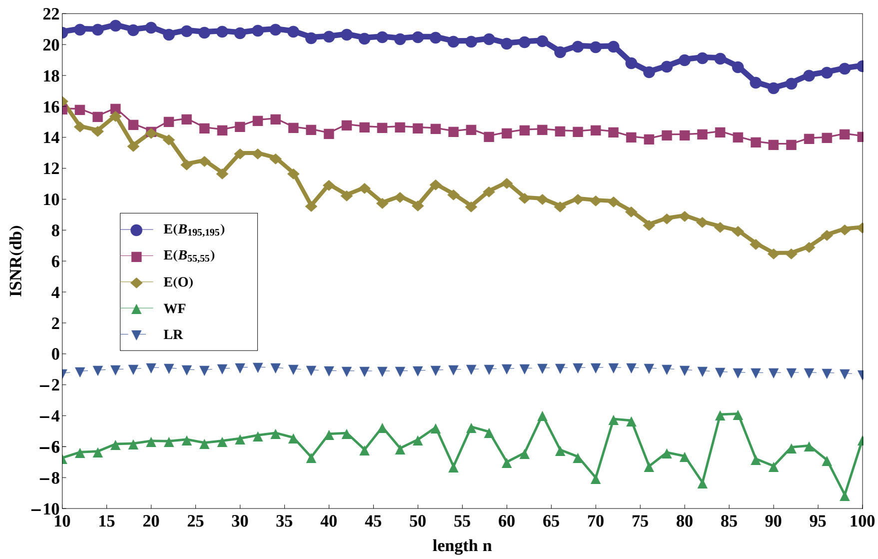


Figure 4.3.4: $ISNR$ versus length n of the blurring process under zero BCs.

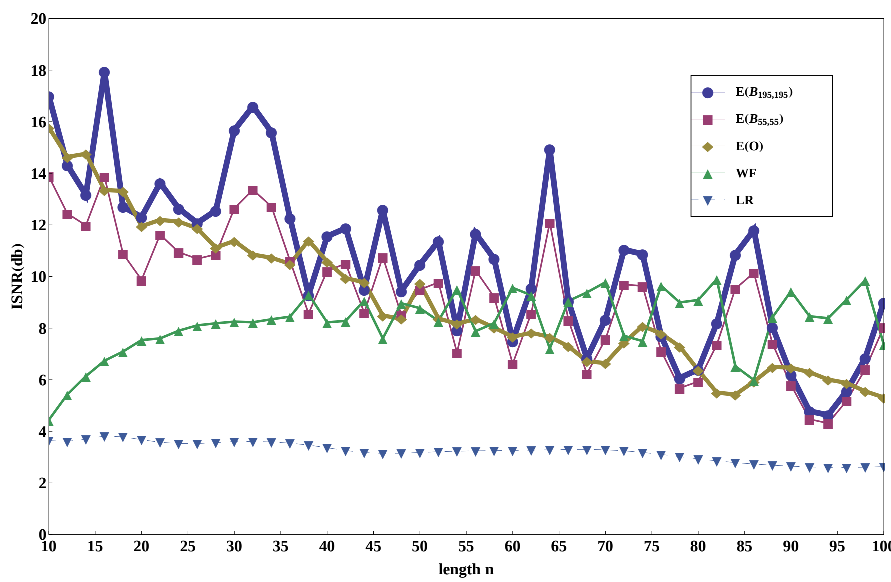


Figure 4.3.5: $ISNR$ versus length n of the blurring process under periodic BCs.

The quantity

$$r_{n,s} = \frac{ISNR_{n,s}}{\max\{ISNR_{n,s} : s \in \{E(B_{195,195}), E(B_{55,55}), E(O), WF, LR\}\}}$$

is called the performance ratio. Finally, the performance of the solver s is defined by the following cumulative distribution function

$$\rho_I(s) = \frac{n_p}{\sum r_{n,s}},$$

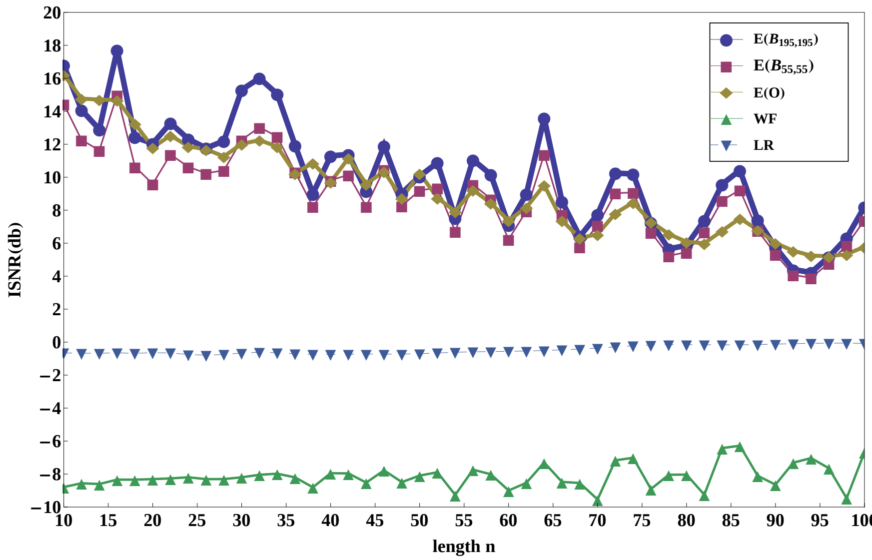


Figure 4.3.6: $ISNR$ versus length n of the blurring process under reflective BCs.

where $n_p = 46$ is the number of numerical experiments.

Similar notations, replacing $ISNR$ values by corresponding $PSNR$ values, could be defined for the performance metric $PSNR$. By $PSNR_{n,s}$ we denote the number of iterations required to solve the blurring problem with the length n by the solver s. The performance ratio is now defined by

$$r_{n,s} = \frac{PSNR_{n,s}}{\max\{PSNR_{n,s} : s \in \{E(B_{195,195}), E(B_{55,55}), E(O), WF, LR\}\}}.$$

The performance of the solver s is defined by the quotient

$$\rho_P(s) = \frac{n_p}{\sum r_{n,s}}.$$

Corresponding $PSNR$ values are illustrated on figures 4.3.7, 4.3.8 and 4.3.9.

Clearly that $r_{n,s} \leq 1$ and greater values for $r_{n,s}$ are desirable, so that the best possible property for a solver s is $\rho_I(s) = 1$. We will say that the solver s_1 overcomes the solver s_2 if the inequality $\rho_I(s_1) \geq \rho_I(s_2)$ is satisfied. This situation will be marked by $s_1 \succeq s_2$.

Several conclusions related to the operator E, Wiener filter and Lucy-Richardson algorithm are observed.

Conclusion 1. Values for $\rho_I(s)$ are arranged in Table 4.3.1.

Table 4.3.1: Values for $\rho_I(s)$.

	$\rho_I(E(B_{195,195}))$	$\rho_I(E(B_{55,55}))$	$\rho_I(E(O))$	$\rho_I(WF)$	$\rho_I(LR)$
Zero Bcs	1	0.732026	0.524752	-0.299391	-0.052279
Periodic Bcs	0.909776	0.797877	0.781241	0.756565	0.287797
Reflective Bcs	0.974235	0.859957	0.903068	-0.873373	-0.044270

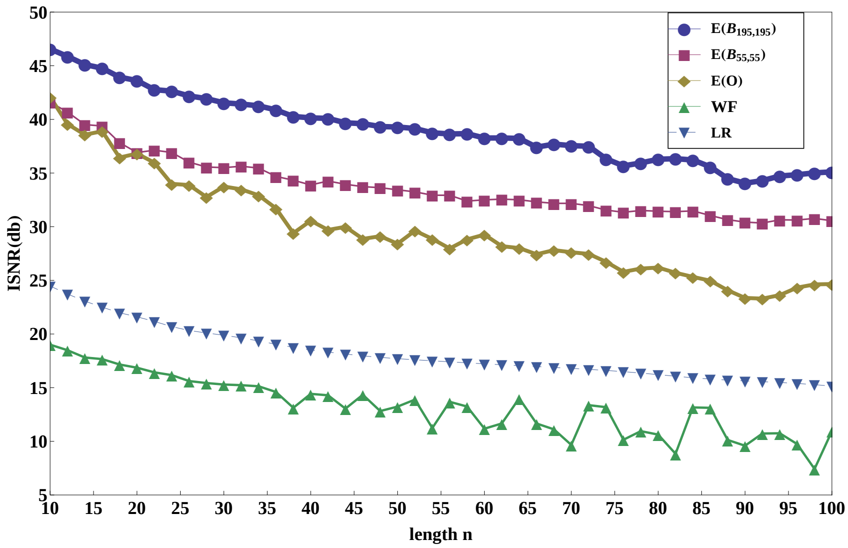


Figure 4.3.7: $PSNR$ versus length n of the blurring process under zero BCs.

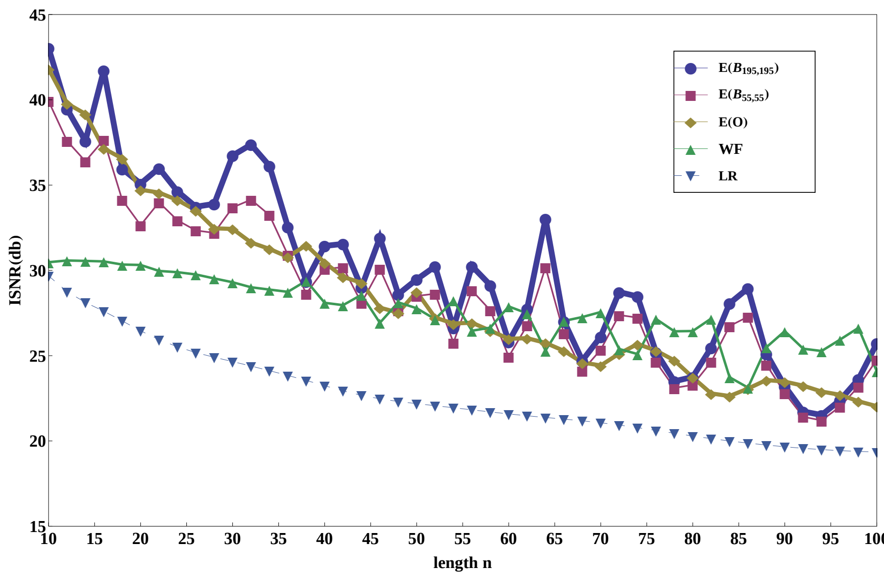


Figure 4.3.8: $PSNR$ versus length n of the blurring process under periodic BCs.

The comparison between considered image restoration solvers could be presented in Table 4.3.2.

Table 4.3.2: Comparison of considered solvers with respect to $ISNR$.

	Comparison
Zero Bcs	$E(B_{195,195}) \succeq E(B_{55,55}) \succeq E(O) \succeq LR \succeq WF$
Periodic Bcs	$E(B_{195,195}) \succeq E(B_{55,55}) \succeq E(O) \succeq WF \succeq LR$
Reflective Bcs	$E(B_{195,195}) \succeq E(O) \succeq E(B_{55,55}) \succeq WF \succeq LR$

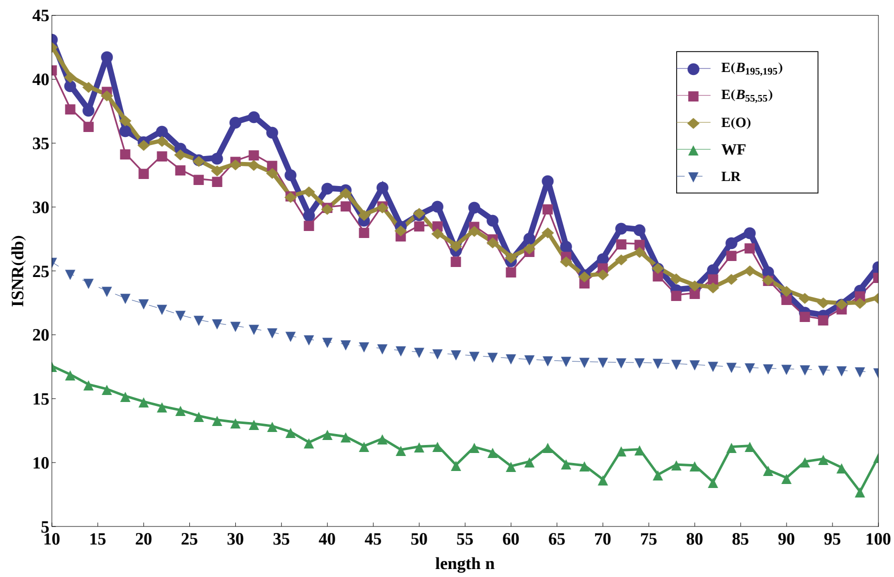


Figure 4.3.9: $PSNR$ versus length n of the blurring process under reflective BCs.

Values for $\rho_P(s)$ are arranged in Table 4.3.3.

Table 4.3.3: Values for $\rho_I(s)$.

	$\rho_P(E(B_{195,195}))$	$\rho_P(E(B_{55,55}))$	$\rho_P(E(O))$	$\rho_P(WF)$	$\rho_P(LR)$
Zero Bcs	1	0.863191	0.755812	0.336527	0.462760
Periodic Bcs	0.968595	0.926170	0.917733	0.905111	0.736051
Reflective Bcs	0.992455	0.9525998	0.964817	0.386173	0.649018

The comparison between the solvers with respect to $PSNR$ values is given in Table 4.3.4.

Table 4.3.4: Comparison of considered solvers with respect to $PSNR$.

	Comparison
Zero Bcs	$E(B_{195,195}) \succeq E(B_{55,55}) \succeq E(O) \succeq LR \succeq WF$
Periodic Bcs	$E(B_{195,195}) \succeq E(B_{55,55}) \succeq E(O) \succeq WF \succeq LR$
Reflective Bcs	$E(B_{195,195}) \succeq E(O) \succeq E(B_{55,55}) \succeq LR \succeq WF$

From tables 4.3.2 and 4.3.4 we conclude that $E(B_{195,195})$, $E(B_{55,55})$ and $E(O)$ significantly overcome two standard image restoration algorithms (Wiener filter and Lucy-Richardson algorithm).

Conclusion 2. From figures 4.3.4–4.3.9 it can be clearly seen that all the measurements taken decrease as the length of the blurring process increase. The decrement in the declination of presented graphs is self–evidence.

Conclusion 3. Figure 4.3.10 shows the $ISNR$ (line marked with squares) and the $PSNR$ (line marked with circles), for various values of moment indices k, l from values $k = l = 10$ to $k = l = 400$ with the step 5 and keeping the blurring process constant with the length $n = 50$. Also, the zero boundary conditions are assumed.

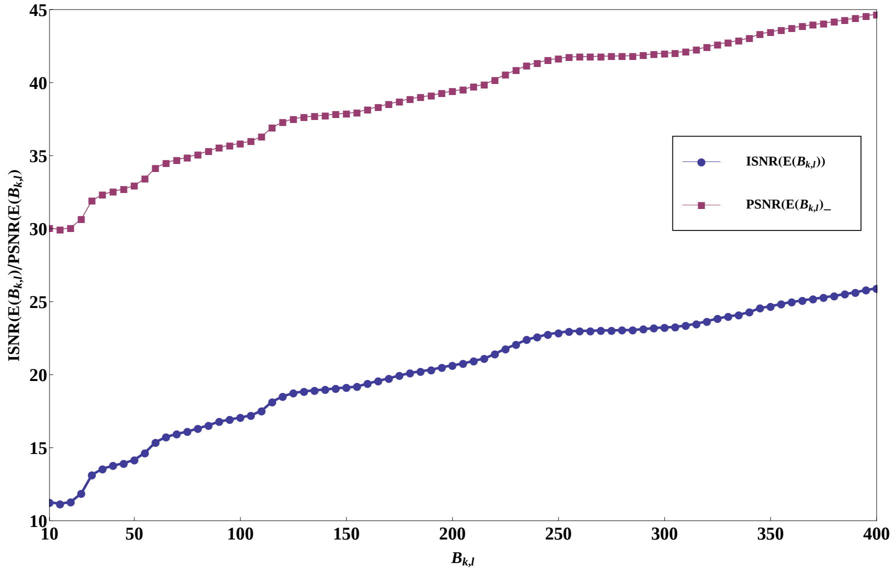
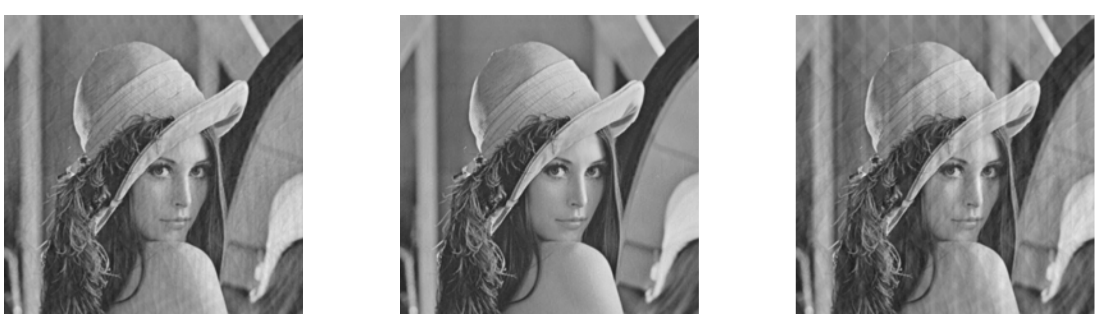


Figure 4.3.10: $ISNR$ and $PSNR$ versus values of moment, with constant n.

It is evident that the robustness of the improvement $E(Y)$ increases considerably with the increase of the number of moments. This fact could be observed on Figure 4.3.10.

In the sequel, on Figure 4.3.11 we illustrate the usefulness of the proposed method on the most obvious way. We can see clearly that both restorations $E(Y = B_{55,55})$ and $E(Y = B_{195,195})$ of the image are better than the restoration $E(Y = O)$, based on the Moore-Penrose solution. The best restoration is evidently reached in the case $E(Y = B_{195,195})$.

In what follows, we investigate an interesting property of the operator $E(Y)$, representing correlation between $ISNR(E(Y))$ and $ISNR(Y)$. For this purpose, we need some additional $ISNR$ and $PSNR$ values, for some different choices of the matrix Y. In Table 4.3.5 we present



(a) $Y = B_{55,55}$ (b) $Y = B_{195,195}$ (c) $Y = O$

Figure 4.3.11: Restoration of Lena for different Y.

generated values for $ISNR(Y)$ and $PSNR(Y)$ as well as corresponding values $ISNR(E(Y))$ and $PSNR(E(Y))$.
We used for Y the following matrices:
– $Y = rand(0,1)$, where $rand$ generates a random matrix with entries from the interval $(0,1)$ uniformly distributed;
– $Y = nctrnd(1,100)$, whose entries are random numbers chosen from the noncentral t distribution with parameters 1 and 100;
– $Y = frnd(2,2)$, which contains random numbers chosen from the F distribution with parameters 2 and 2;
– $Y = lognrnd(1,2)$ with entries from a *lognormal* distribution with parameters 1 and 2;
– $Y = randn(0,1)$, containing pseudo-random values drawn from a normal distribution with mean zero and standard deviation one;
– $Y = O$.

$ISNR$ and $PSNR$ values generated for the choice $Y = \widetilde{F}$ are denoted by $ISNR(Y)$ and $PSNR(Y)$, respectively.

Table 4.3.5: ISNR and PSNR values (in dB) for different choices of Y.

	$ISNR(Y)$	$PSNR(Y)$	$ISNR(E(Y))$	$PSNR(E(Y))$
$Y = nctrnd(1,100)$	-84.4601	15.96	-61.8865	14.83
$Y = frnd(2,2)$	-47.4319	15.16	-26.6929	18.12
$Y = lognrnd(1,2)$	-15.2974	13.07	-1.2873	17.42
$Y = randn(0,1)$	-10.6372	16.46	11.3462	20.58
$Y = O$	-10.6373	16.41	11.3497	20.58
$Y = rand(0,1)$	-10.6079	16.41	11.3639	20.59
$Y = B_{55,55}$	4.9200	21.66	16.6965	24.58
$Y = B_{195,195}$	10.6296	26.24	22.9395	30.50

Table 4.3.5 has been divided into three parts horizontally. The middle part is a row denoted by $Y = O$, which contains $ISNR$ and $PSNR$ values generated in the case $Y = O$. The first part of the table contains some of the functions where the results presented for the $ISNR$ and the $PSNR$ are worst than the case for $Y = O$. The second part includes the cases where the functions $E(Y)$, including our Haar moment based method, provide better results than the Moore–Penrose reconstruction method.

The results contained in Table 4.3.5 can be seen as one more evidence that values for Y closer to the original image F produce better improvements and when Y is far from F, the results are better for $Y = O$. More precisely, in all of the experiments, $ISNR(Y) < ISNR(O)$ implies $ISNR(E(Y)) < ISNR(E(O))$ and alternatively $ISNR(Y) > ISNR(O)$ implies that $ISNR(E(Y)) > ISNR(E(O))$. Furthermore, it is obvious that the values $ISNR(E(Y))$ (resp. $PSNR(E(Y))$) has greater values with respect to $ISNR(Y)$ (resp. $PSNR(Y)$).

Remark 4.3.1. *The operator $E(Y)$ behaves as the improvement for each Y which is not of the form $Y = E(X)$, for a matrix X. More precisely, our experience and Proposition 3.3.2 indicate*

$$ISNR(E(Y))\ \rho\ ISNR(Y), \quad \text{where } \rho = \begin{cases} >, & Y \notin \mathcal{R}(E), \\ =, & Y \in \mathcal{R}(E). \end{cases}$$

Therefore, it is seems reasonable to assume that there is a threshold real value t *which guarantees that* $ISNR(E(Z)) > ISNR(E(Y = O)) = 11.3497$, *for each* Z *which satisfies* $ISNR(Z) > t$. *The case* $Y = O$ *is especially separated because it has already been studied in the literature. According to results presented in Table 4.3.5, we conclude that*

$$t = \begin{cases} ISNR(O) = -10.6373, & Z \notin \mathcal{R}(E), \\ ISNR(E(O)) = 11.3497, & Z \in \mathcal{R}(E). \end{cases}$$

Taking into account the PSNR values from Table 4.3.5, we get

$$t = \begin{cases} PSNR(O) = 16.41, & Z \notin \mathcal{R}(E), \\ PSNR(E(O)) = 20.58, & Z \in \mathcal{R}(E). \end{cases}$$

The regression line has been approximated according to the calculated values of the $ISNR(Y)$ and $ISNR(E(Y))$ represented on Table 4.3.5. The equation of the regression line is

$$ISNR(E(Y)) = 0.9104 \cdot ISNR(Y) + 16.5881.$$

The results were plotted on Figure 4.3.12, where the isolated circles represent the calculated values and the solid line represents the approximated regression line.

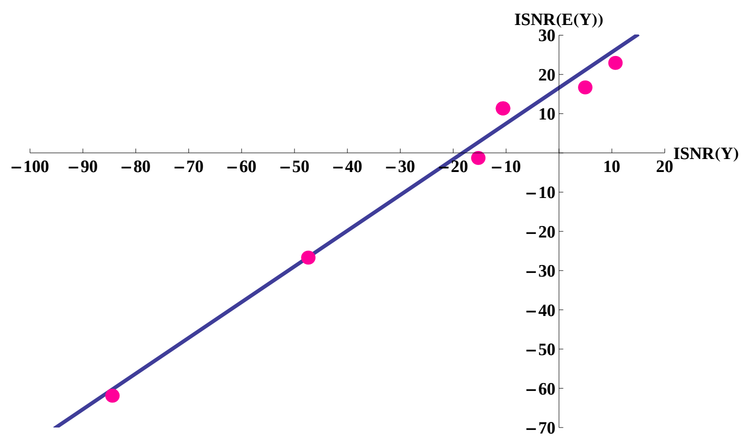


Figure 4.3.12: Linear Regression between $ISNR(Y)$ and $ISNR(E(Y))$.

4.3.2 Application of the method on blurred and noisy images

Noise is unavoidable in most of applications, so that a real observation is thus often modeled by

$$HF_{XY} + ns = F_{XY}$$

provided that the noise ns is additive, although multiplicative noise can be handled similarly. In another formulation the noise can also be simulated by rewriting the equation as

$$F_{XY} = \sum_{k=1}^{p} \sum_{l=1}^{m} F_{XY}(k,l)h(i,j;k,l) + ns(i,j) \tag{4.3.1}$$

where $i = 1, \ldots, p$ and $j = 1, \ldots, m$.

(a) Noisy degraded image (b) Noisy and motion blurred image (c) Restored image

Figure 4.3.13: Restoration of noisy and motion blurred Lena.

In Figure 4.3.13(a), we simulate a noise model where a number of pixels are corrupted and randomly take on a value of white and black (salt and pepper noise) with noise density equal to 0.1. The image that we receive from a faulty transmission line can contain this form of corruption. In Figure 4.3.13 (b) we present the original image while a motion blurred and a salt and pepper noise has been added to it. Figure 4.3.13 (c) demonstrates the application of a low pass rotationally symmetric Gaussian filter of standard deviation equal to 45 after using the moment based restoration method for $E(Y = B_{195,195})$.

Accordingly, the graphical representation of the $ISNR$ against the noise density of the salt and pepper noise interference is illustrated on Figure 4.3.14. The bottom curve (in green) represents the case where the matrix Y filled with random entries, the middle graph (in red) represents the case where the matrix Y is equal to zero and finally the upper curve (in blue) corresponds to the case where the matrix Y is a moment based image restoration matrix using as $Y = B_{195,195}$. It can be seen that the latter case provides the best obtained result, in all of the cases the number of blurring process kept a high constant value equal to 100.

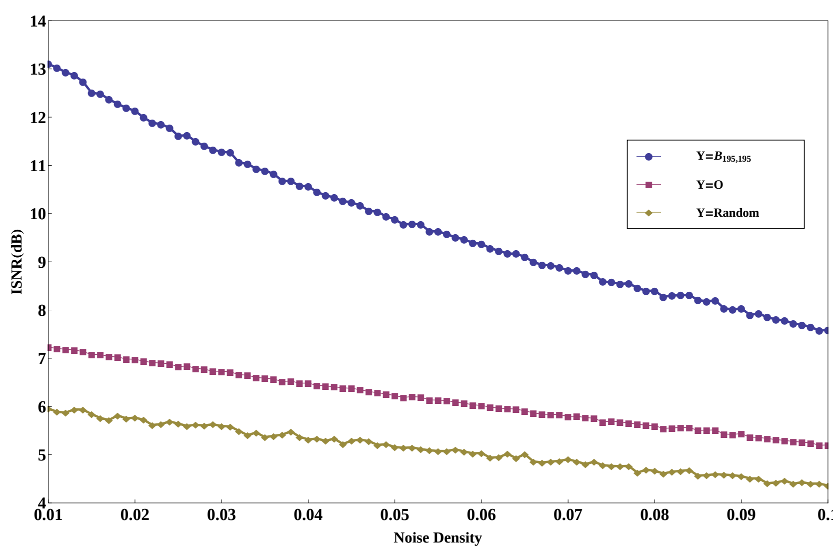


Figure 4.3.14: ISNR against the noisy density for the salt and pepper interference case.

For convenience and comparison reasons the graphical representation of the $ISNR$ against the noise density of the Gaussian noise interference is illustrated on Figure 4.3.15.

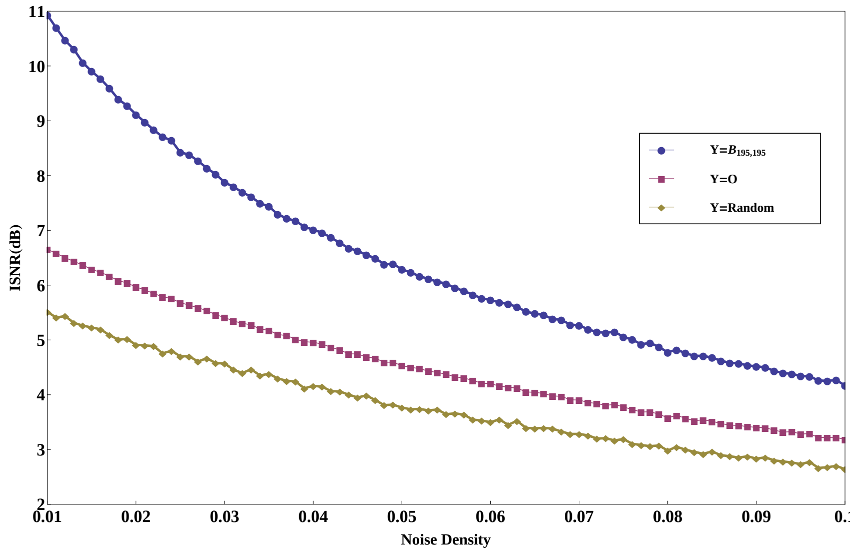


Figure 4.3.15: ISNR against the noisy density for the Gaussian interference case.

The bottom curve (in green) illustrates the data generated in the case where the matrix Y filled with random entries, the middle graph (in red) represents the case $Y = O$ and the upper graph (in blue) represents the case where the matrix Y is a moment based image restoration matrix, that is, $Y = B_{195,195}$.

In Table 4.3.6 we present the normalized reconstruction error given by

$$N_E = \frac{1}{\sqrt{\sum_{x=1}^{X}\sum_{y=1}^{Y}[F_{XY}]^2}}\sqrt{\sum_{x=1}^{X}\sum_{y=1}^{Y}[F_{XY} - F_{XY}]^2},$$

where F_{XY} and F_{XY} are the original and the reconstructed images respectively. This time the image goes through a motion - blurred and noisy process according to formula presented on eq. (4.4.1). The noise part is a Gaussian white noise with constant mean value at 0 and variance that varies from 0.01 to 0.10 as shown on Table 4.3.6. Image processing and analysis are based on filtering the content of the images in a certain way. The filtering process that is used in order to clear the image is achieved by applying a low-pass rotationally inverse Gaussian filter, in this case on the generalized inverse or Haar based reconstructed image.

After a considerable investigation and for observations where noise is present, the threshold point has been traced. That point is where the results obtained from our proposed method are identical to the Moore-Penrose reconstruction method (where $Y = O$). In our case the threshold point attained when the Haar coefficients are equal to $k = l = 30$. It is worth noticing that for the case where noise is not present, the threshold point is attained for very low values of the Haar coefficients.

Table 4.3.6: Normalized reconstruction error.

Variance	N_E (for $Y = 0$)	N_E (for $B_{55,55}$)	N_E (for $B_{195,195}$)
0.01	0.2094	0.2039	0.1848
0.02	0.2460	0.2225	0.2003
0.03	0.2491	0.2281	0.2104
0.04	0.2692	0.2532	0.2153
0.05	0.2778	0.2336	0.2363
0.06	0.2923	0.2533	0.2516
0.07	0.3029	0.2646	0.2711
0.08	0.3267	0.2713	0.2691
0.09	0.3297	0.2896	0.2848
0.10	0.3381	0.3122	0.2958

4.4 Image deblurring process based on separable restoration methods

In this section we apply the proposed transformations $E(Y)$ and $E_1(Y)$ to several types of possible values for the matrix Y. The assumption $h_i = 1/n$, $i = 1, \dots, n$ is imposed in matrices H_c and H_r. The experiments are performed using Matlab programming language on an Intel(R) Core(TM) i5 CPU M430 @ 2.27 GHz 64/32-bit system with 4 GB of RAM memory running on the Windows 7 Ultimate Operating System.

4.4.1 Results for Wiener Filter, CLS Filter and SMRS solution

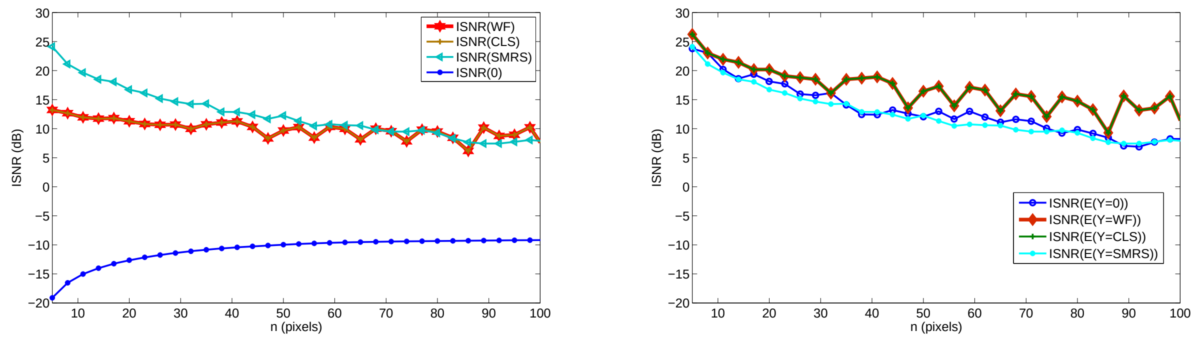
The uniform linear blur case. The tested image *Lena* is shown in Figure 4.4.1(a). The blurred image that has been degraded by a uniform linear motion in the horizontal direction is presented in Figure 4.4.1(b).

(a)

(b)

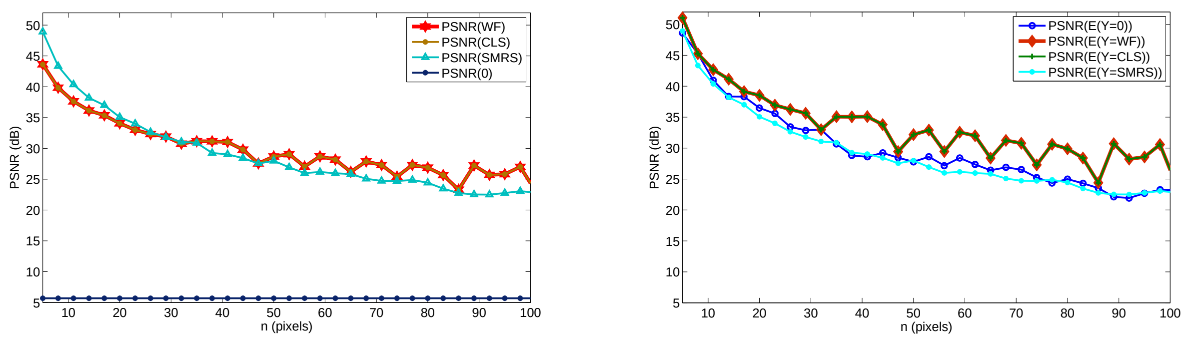
Figure 4.4.1: (a) Image of Lena, and (b) Blurred Image of Lena from uniform linear motion with $n = 62$.

The ISNR and the PSNR values are presented on the figures 4.4.2 and 4.4.3, respectively, for different values of the length n of the blurring processes. The left graphical representations on these figures show the ISNR and the PSNR values generated after the removal of blur by means of two standard image restoration methods (such as Wiener filter – WF, constrained least-squares filter –CLS), as well as the Symmetric Minimal Rank Solution ($SMRS$) of the matrix equation (3.4.3). The presentation of the ISNR and the PSNR values corresponding to $E(Y)$ for $Y = O$, $Y = WF$, $Y = CLS$ are given in the right graphs on figures 4.4.2 and 4.4.3. Because of the completeness of the comparison, values for $ISNR(O)$ and $PSNR(O)$ are also shown on the left images.



(a) (b)

Figure 4.4.2: ISNR versus length n of the blurring process for Lena and uniform blur for (a) $O, WF, CLS, SMRS$, and (b) $E(Y = O)$, $E(Y = WF)$, $E(Y = CLS)$, $E(Y = SMRS)$.



(a) (b)

Figure 4.4.3: PSNR versus length n of the blurring process for Lena and uniform blur for (a) $O, WF, CLS, SMRS$, and (b) $E(Y = 0), E(Y = WF), E(Y = CLS)$, $E(Y = SMRS)$.

We note that the results obtained applying the image restoration algorithms WF and CLS are almost identical, so that corresponding graphs overlap on both pictures. Changes from values $ISNR(Y)$ and $PSNR(Y)$ to corresponding values $ISNR(E(Y))$ and $PSNR(E(Y))$ were different.

• In the cases $Y = WF$ and $Y = CLS$ there was an identical increasing.
• The values $ISNR(E(Y = O))$ and $PSNR(E(Y = O))$ have the greatest increase compared to $ISNR(O)$ and $PSNR(O)$.
• The values $ISNR(SMRS)$ and $ISNR(E(Y = SMRS))$ are identical (as well as $PSNR(SMRS)$ and $PSNR(E(Y = SMRS))$), as it can be expected from Theorem 3.4.1.

• Hierarchy corresponding to both $ISNR(Y)$ and $PSNR(Y)$ values (observed in left graphs on figures 4.4.2 and 4.4.3) retains the same in the cases $ISNR(E(Y))$ and $PSNR(E(Y))$ in the choices $Y = WF, Y = CLS$ (present situation in right graphs on figures 4.4.2 and 4.4.3). Since the equality $ISNR(SMRS) = ISNR(E(Y = SMRS))$ evidently holds, the selection $Y = SMRS$ lose the priority with respect to the case $Y = O$, seen at the graphs shown in figures 4.4.2(a) and 4.4.3)(a).

Figures 4.4.4(a), 4.4.4(b), and 4.4.4(c) show the reconstructed image *Lena* with with three standard image restoration methods described in Chapter 2.

(a) (b) (c)

Figure 4.4.4: Reconstruction of Lena with: (a) WF, (b) CLS, (c) $SMRS$.

Figure 4.4.5 illustrates improvements in the restoration, which is obtained applying the operator $E(Y)$.

(a) (b) (c)

Figure 4.4.5: Restoration of Lena with: (a) $E(Y = WF)$; (b) $E(Y = CLS)$; (c) $E(Y = O)$.

Comparing figures 4.4.4 and 4.4.5 it is obviously clear that $E(Y)$ gives better restorations with respect to restorations generated by Y. From Figure 4.4.5 it is clearly apparent by the human eye that each of the restorations $E(WF), E(CLS)$ produces better reconstruction with respect to the choice $Y = O$, corresponding to the Moore–Penrose solution of the matrix equation (3.4.3).

In Table 4.4.1 we illustrate our hypothesis from [83] that the opposite inequality

$$ISNR(E(Y)) < ISNR(Y)$$

is satisfied in the case when the matrix Y is almost the same as the original F. Since so far no image restoration method that could generate such a big improvement is known, the matrix Y is obtained as a small degradation of F, not by applying an image restoration method. Numerical experience shows that for standard Matlab pictures, such as Lena, Barbara, Boat, Man and other $ISNR(Y) > 180$ implies $ISNR(Y) > ISNR(E(Y))$. This experience can be formally verified using the trend line $y = 0.9104 * x + 16.5881$ from [83]. The inequality $y < x$ is satisfied in the case $x > 16.588/0.09 \approx 184.31$. For example, the results for 'Lena' corresponding to values $ISNR(Y)$ and $ISNR(E(Y))$ for all values n from 5 to 101 with step 3 are arranged in Table 4.4.1.

Table 4.4.1: Values for n, $ISNR(Y)$ and $ISNR(E(Y))$.

n	$ISNR(Y)$	$ISNR(E(Y))$		n	$ISNR(Y)$	$ISNR(E(Y))$
5	249.2923918	246.5754751		56	261.1267412	245.0397281
8	252.5410863	245.8133034		59	261.3197821	244.7947034
11	254.0380289	245.4559672		62	261.5020619	244.5066636
14	255.290775	247.8147012		65	261.6693869	245.3431482
17	256.1627449	245.0849736		68	261.8267126	245.3387704
20	256.9266896	247.1001053		71	261.9720388	244.7968745
23	257.5158629	246.4096921		74	262.1082977	244.9597282
26	258.0526813	246.8480396		77	262.2343166	243.3086415
29	258.503794	245.9257102		80	262.3524128	245.0872482
32	258.9244034	245.3700613		83	262.4623246	243.7656052
35	259.2885286	244.8638084		86	262.5664801	244.6452163
38	259.6271666	245.6146567		89	262.6646006	244.0284835
41	259.9272486	245.5255758		92	262.7584279	244.6950721
44	260.2088708	243.9298914		95	262.847103	243.8340798
47	260.461986	245.844263		98	262.9314357	244.3155611
50	260.7001833	245.3140994		101	263.0108589	242.6759354
53	260.9182511	244.2328601				

The results arranged in Table 4.4.1 confirm the next facts:
– the inequality $ISNR(E(Y)) < ISNR(Y)$ holds;
– increase in values $ISNR(Y)$ implies decrease in values $ISNR(E(Y))$.

We also made numerical results for $SMRS_{\alpha,\beta}$. For selected values $n = 35, 44, 53, 62$ and $\alpha, \beta = 0, 0.1, 0.2, \ldots, 1$ derived results are arranged in Table 4.4.2.

Table 4.4.2: Selected values $ISNR(Y = SMRS_{\alpha,\beta})$ for given values n.

n	$ISNR(SMRS_{0,0})$	$ISNR(SMRS_{1,1})$	$\max(ISNR(SMRS_{\alpha,\beta}))$
35	14.10480781	14.30639718	$ISNR(SMRS_{0.5,1}) = 14.91242836$
44	13.21132067	12.42006111	$ISNR(SMRS_{0.4,1}) = 13.68685028$
53	12.97303198	11.32319729	$ISNR(SMRS_{0.3,1}) = 13.17562410$
62	11.98878458	10.60190380	$ISNR(SMRS_{0.3,1}) = 12.23963421$

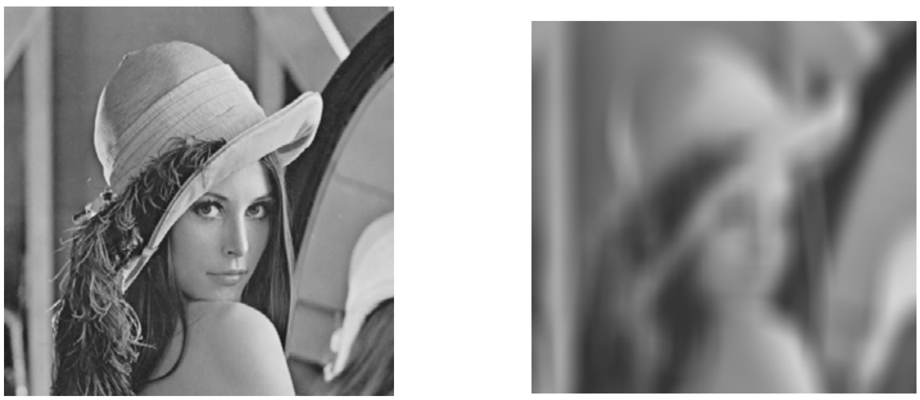
From Table 4.4.2 we can conclude that for some values of the weighting coefficients α and β we obtain higher values for the $ISNR$ in terms when coefficient have value 0 or 1. These mean that for a given value of the uniform linear motion n, we can determine the values of the coefficients $\alpha = r1$ and $\beta = r2$ for which is valid that

$$ISNR(SMRS_{\alpha,\beta}) = ISNR(SMRS_{r1,r2}) > ISNR(SMRS_{0,0}) = E(Y = 0)$$

and

$$ISNR(SMRS_{\alpha,\beta}) = ISNR(SMRS_{r1,r2}) > ISNR(SMRS_{1,1}).$$

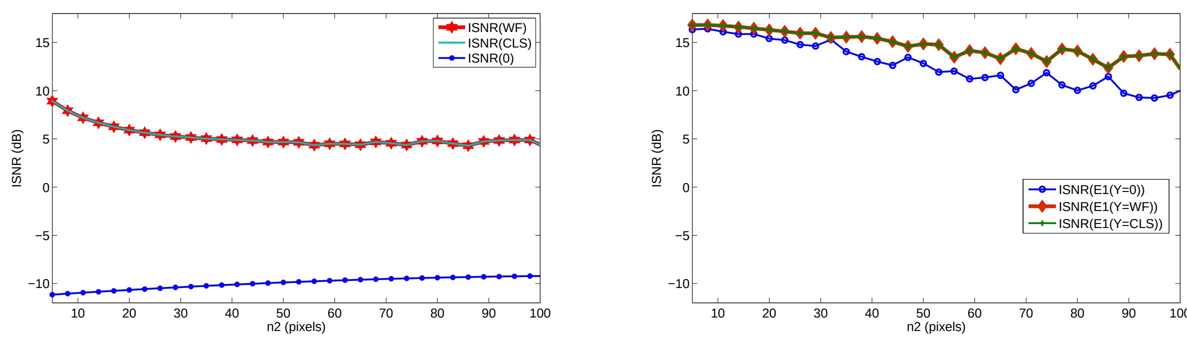
The separable motion blur case. The input image *Lena* blurred by separable motion with $n_1 = 30$ and $n_2 = 44$ is given in Figure 4.4.6.



(a) (b)

Figure 4.4.6: (a) Image of Lena, and (b) Blurred Image of Lena by separable motion with $n_1 = 30$ and $n_2 = 44$.

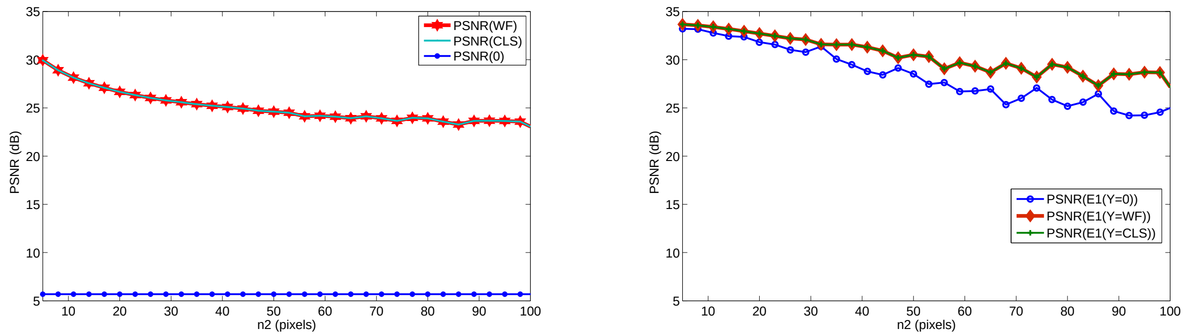
Values for ISNR and PSNR arising after the application of separable motion blurring on the input image *Lena* are illustrated in figures 4.4.7 and 4.4.8, respectively.



(a) (b)

Figure 4.4.7: ISNR versus variable length n_2 and $n_1 = 30$ of the blurring process for separable motion blur (a) O, WF, CLS and (b) $E_1(Y = O), E_1(Y = WF), E_1(Y = CLS)$.

From figures 4.4.7 and 4.4.8 we observe relations $ISNR(E_1(Y)) > ISNR(Y)$ and $PSNR(E_1(Y)) > PSNR(Y)$. As a consequence, $E_1(Y)$ produces evidently better restorations



(a) (b)

Figure 4.4.8: PSNR versus variable length n_2 and $n_1 = 30$ of the blurring process for separable motion blur (a) O, WF, CLS and (b) $E_1(Y = O), E_1(Y = WF), E_1(Y = CLS)$.

with respect to restorations defined by Y. This fact is confirmed by the next two figures 4.4.9 and 4.4.10.

(a) (b)

Figure 4.4.9: Reconstruction of Lena with: (a) $Y = WF$, (b) $Y = CLS$.

The numerical results corresponding to the restoration of the image *Barbara* degraded by the separable motion blur are identical as in the previous case, where the input picture *Lena* is chosen.

4.4.2 Results for moment based methods

In this subsection the matrix Y was substituted by the moment representation of the image. For comparison reasons the number of moments $B_{k,l}$ presented on section 2.6 was kept constant with $k = l = 55$ or $k = l = 155$. The values of Y obtained applying the Haar (resp. Fourier) basis in these cases are denoted by $Y = H_{55}$ and $Y = H_{155}$ (resp. $Y = F_{55}$ and $Y = F_{155}$).

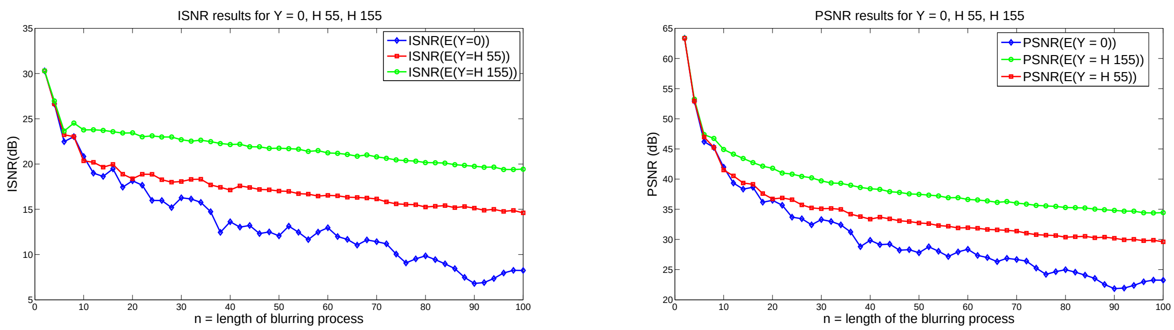
Results corresponding to the Haar basis

(a) (b) (c)

Figure 4.4.10: Restoration of Lena with: (a) $E_1(Y = WF)$; (b) $E_1(Y = CLS)$; (c) $E_1(Y = O)$.

The uniform blur case.

The ISNR and PSNR values obtained applying the operator $E(Y)$ on the blurred image *Lena* are shown on left and right graph on Figure 4.4.11, respectively. The selected values for the Y are $Y = O$, $Y = H_{55}$ and $Y = H_{155}$.



(a) (b)

Figure 4.4.11: a) ISNR values for $E(Y)$, b) PSNR values for $E(Y)$ generated using $Y = O$, $Y = H_{55}$ and $Y = H_{155}$.

Values corresponding to the cases $Y = H_{55}$ and $Y = H_{155}$ are better than the corresponding results generated by the choice $Y = O$.

The separable motion blur case.

The image *Lena* blurred by the separable motion with $n_1 = n_2 = n$ is tested. The results for the ISNR and PSNR values were plotted on Figure 4.4.12.

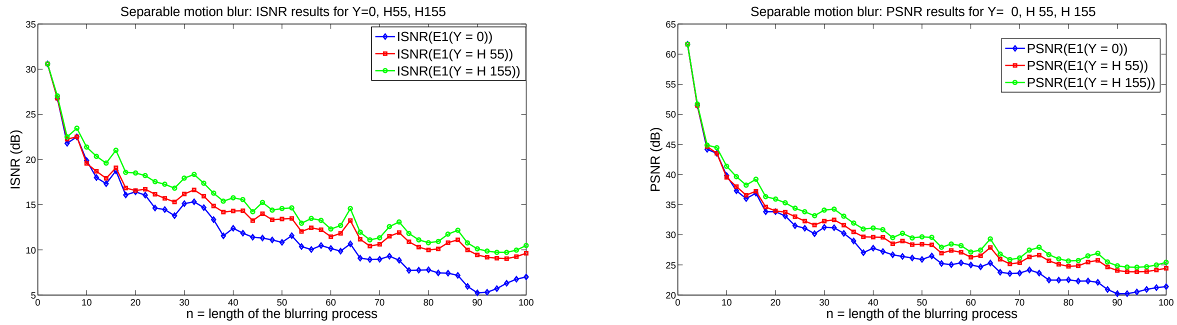
The Haar moment basis analysis provides better results for the image reconstruction when it is used as a value for Y compared to the normal restoration process where Y is equal to the zero matrix.

Results corresponding to the Fourier Transform

The uniform blur case.

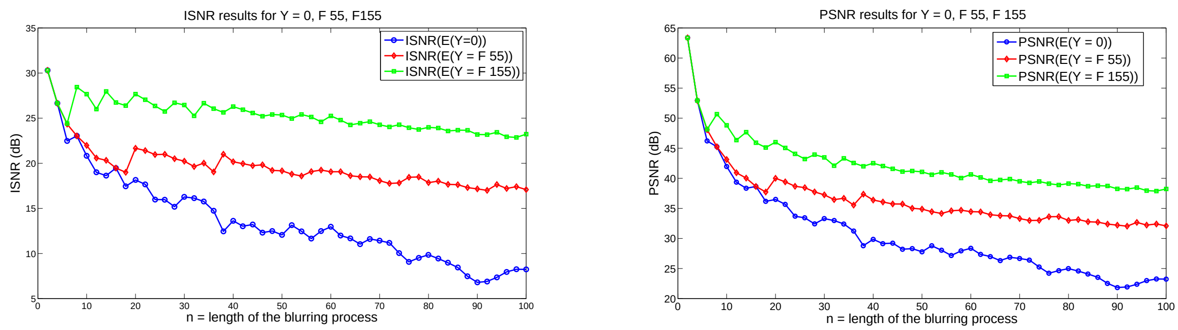
The ISNR and PSNR values for $E(Y)$ corresponding to the tested image *Lena* are shown in Figure 4.4.13. Chosen values for Y are $Y = O$, $Y = F_{55}$ and $Y = F_{155}$.

The separable motion blur case.



(a) (b)

Figure 4.4.12: a) ISNR values for $E_1(Y)$, b) PSNR values for $E_1(Y)$ generated for $n_1 = n_2$ using as $Y = O$, $Y = H_{55}$, $Y = H_{155}$.



(a) (b)

Figure 4.4.13: a) ISNR values for $E(Y)$, b) PSNR values for $E(Y)$ generated using $Y = O$, $Y = F_{55}$, $Y = F_{155}$.

The input image *Lena* blurred by the separable motion with $n_1 = n_2 = n$ is tested. The ISNR and PSNR values are shown in the following figure:

The information that can be extracted from the ISNR and PSNR figures is that the improvement on the restored image is significantly better in all of the cases where the moment basis analysis was used as a value for Y than the one used by the generalized inverse method where $Y = O$.

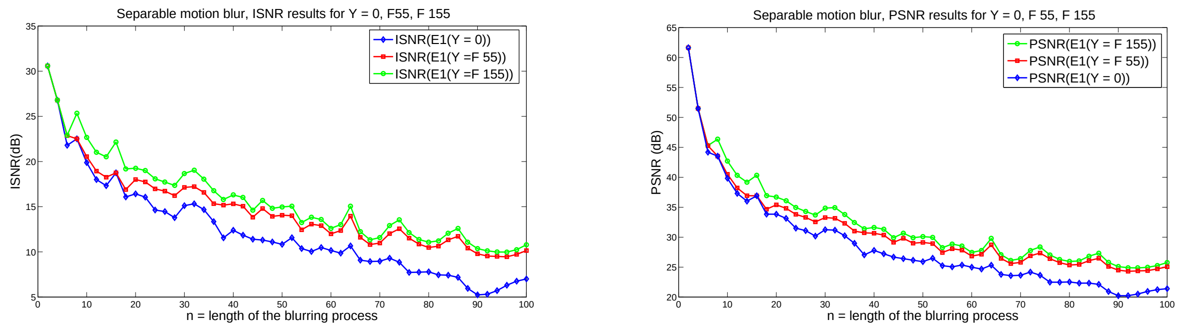
The results were similar for all other images used for numerical experiments.

4.4.3 Final comparisons

Finally, we will present an overall comparison between the methods presented in Section 3.3 in the uniform blur case. The ISNR and the PSNR values of $E(Y)$ are generated for all considered choices of the matrix Y.

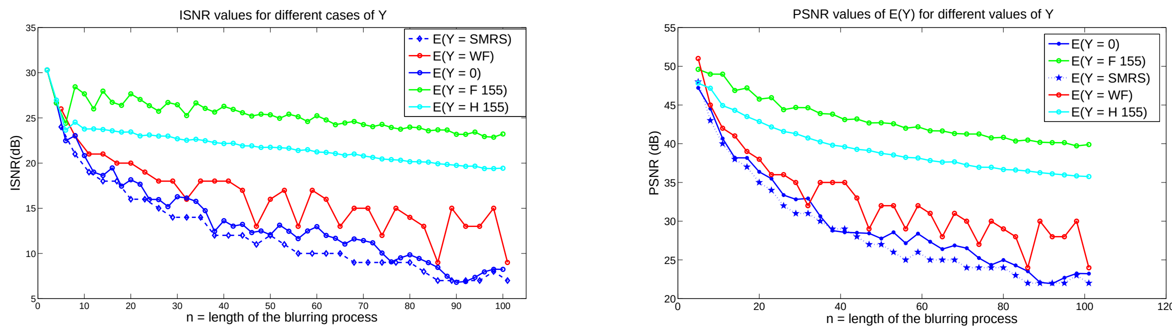
As we can see, it is clear that the moment based methods (Fourier basis and Haar basis) give much better results that all other methods tested. We can also see that among the other methods, the case $Y = WF$ gives greater ISNR and PSNR values and that the case $Y = SMRS$ is the least preferred candidate for the matrix Y. Since the values corresponding to the choice $Y = CLS$ almost coincide with those of the Weiner filter ($Y = WF$) in both criterions (ISNR and PSNR), they are not presented in these figures.

Overall, it seems that for various cases that have been tested for the matrix Y, the best



(a) (b)

Figure 4.4.14: a) ISNR values for $E_1(Y)$ and b) PSNR values for $E_1(Y)$ generated using $Y = O$, $Y = F_{55}$, $Y = F_{155}$, $n_1 = n_2 = n$.



(a) (b)

Figure 4.4.15: a) ISNR values of $E(Y)$; b) PSNR values of $E(Y)$, for all cases of Y presented.

results obtained when Fourier (spectral domain) and Haar (spatial domain) moments were used. It is therefore desirable that the original image is known. On the other hand, the moment method provides a fast recovery technique by taking the necessary coefficients each time depending on the original image. Generally moments provide information on its low frequency of an image. The higher number of moments used the more detail of the image captured. Moreover, the method is independent from the filtering process which provides a serious advantage compared to the other image processing and analysis techniques tested in this section.

4.4.4 Application of the method on blurred and noisy images

In most applications noise is unavoidable and a real observation is thus often modeled by

$$G = FH^T + N$$

where N is the noise additive matrix, provided that the noise N is additive. The multiplicative noise can be handled similarly. In another formulation the noise can also be simulated by rewriting the above equation as

$$S_{XY} = \sum_{k=1}^{p} \sum_{l=1}^{m} S_{XY}(k,l)h(i,j;k,l) + ns(i,j) \tag{4.4.1}$$

where $i = 1, \ldots, p$ and $j = 1, \ldots, m$.

As it was shown in the previous subsection, the moment based methods (Fourier basis and Haar basis) give significantly better results than all other methods tested. The presence of noise will only be investigated in such cases.

In Figure 4.4.16(a), we simulate a noise model where a number of pixels are corrupted and randomly take on a value of white and black (salt and pepper noise) with noise density equal to 0.1. The image that we receive from a faulty transmission line can contain this form of corruption. In Figure 4.4.16 (b) we present the damaged image that is obtained by a motion blur and a salt and pepper noise. Figure 4.4.16 (c) demonstrates the result of applying a low pass rotationally symmetric Gaussian filter of standard deviation equal to 45 after using the moment based restoration method for $E(Y = H_{155})$.

(a) (b) (c)

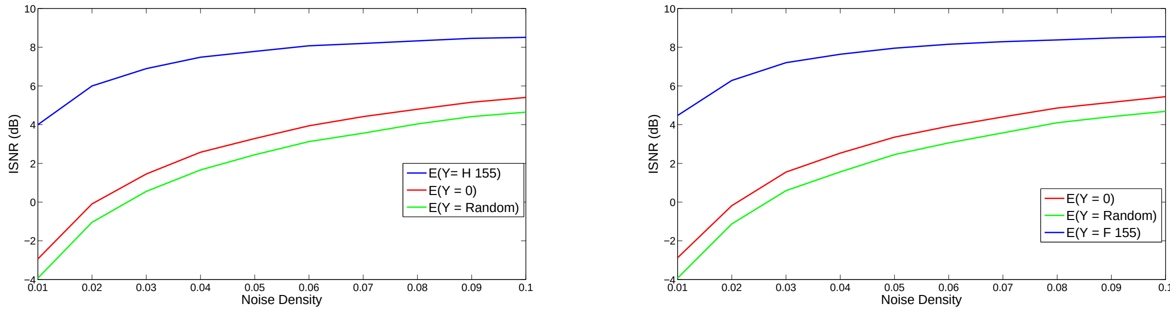
Figure 4.4.16: (a) Noisy degraded Lena image, (b) noisy and motion blurred Lena image and (c) Restored based on the proposed method and noisy filtered image using the Haar basis.

Accordingly, the graphical representation of the ISNR against the noise density of the salt and pepper noise interference is illustrated on Figure 4.4.17. The bottom curves (in green) represent the case where the matrix Y is filled with random entries, the middle ones (in red) represent the case where the matrix Y is equal to zero and finally the upper curves (in blue) correspond to the case where the matrix Y is a moment based image restoration matrix using as $Y = H_{155}$, or as $Y = F_{155}$ respectively. It can be seen that the latter case provides the best obtained result, in all of the cases the number of blurring process kept a high constant value equal to 100, for both the Haar and the Fourier basis.

For convenience and comparison reasons, the graphical representation of the ISNR against the noise density of the Gaussian noise interference is illustrated on Figure 4.4.18. The bottom curves (in green) illustrate the data generated in the case where the matrix Y filled with random entries, the middle ones (in red) represent the case $Y = 0$ and the upper graphs (in blue) represent the case where the matrix Y is a moment based image restoration matrix, that is, $Y = H_{155}$ on graph (a) and $Y = F_{155}$ on graph (b).

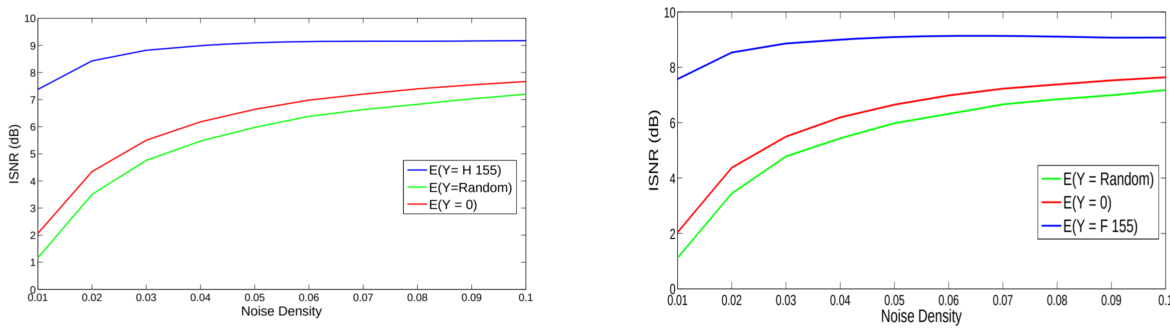
In Table 4.4.3 we present the normalized reconstruction error given by

$$N_E = \frac{1}{\sqrt{\sum_{x=1}^{X} \sum_{y=1}^{Y} [S_{XY}]^2}} \sqrt{\sum_{x=1}^{X} \sum_{y=1}^{Y} [S_{XY} - S_{XY}]^2},$$



(a) (b)

Figure 4.4.17: ISNR against the noisy density for the salt and pepper interference case: (a) The Haar Basis and (b) The Fourier Basis.



(a) (b)

Figure 4.4.18: ISNR against the noisy density for the Gaussian interference case: (a) The Haar Basis and (b) The Fourier Basis.

where S_{XY} and S_{XY} are the original and the reconstructed images respectively. This time the image goes through a motion - blurred and noisy process according to formula presented on eq. (4.4.1). The noise part is a Gaussian white noise with constant mean value at 0 and variance that varies from 0.01 to 0.10 as shown on Table 4.4.3. Image processing and analysis are based on filtering the content of the images in a certain way. The filtering process that is used in order to clear the image is achieved by applying a low-pass rotationally inverse Gaussian filter, in this case on the generalized inverse or Haar based reconstructed image.

After considerable investigation and observations where noise is present, the threshold point has been traced. That point is where the results obtained from our proposed method are identical to the Moore- Penrose reconstruction method (where $Y = O$). In our case, the threshold point is attained when the Haar coefficients are equal to $k = l = 30$, while for the Fourier coefficients the threshold point was found to be $k = l = 27$. It is worth noticing that for the case where noise is not present, the threshold point is attained for very low values of the Haar and the Fourier coefficients.

Table 4.4.3: Normalized reconstruction error.

Variance	$Y=0$	$Y=H_{55,55}$	$Y=H_{155,155}$	$Y=F_{55,55}$	$Y=F_{155,155}$
0.01	0.2094	0.2039	0.2125	0.1292	0.2135
0.02	0.2460	0.2225	0.2483	0.1394	0.2492
0.03	0.2491	0.2281	0.2571	0.1547	0.2539
0.04	0.2692	0.2532	0.2856	0.1667	0.2717
0.05	0.2778	0.2336	0.2996	0.1857	0.2785
0.06	0.2923	0.2533	0.3073	0.1925	0.2953
0.07	0.3029	0.2646	0.3102	0.1980	0.3048
0.08	0.3267	0.2713	0.3317	0.2082	0.3248
0.09	0.3297	0.2896	0.3357	0.2188	0.3379
0.10	0.3381	0.3122	0.3426	0.2252	0.3473

4.5 Improving the Tikhonov and TSVD image deblurring methods

We present numerical results for $E(Y)$, where Y take value from TSVD or TIK method, and compare the obtained results with results generated by TSVD (2.7.3) and TIK (2.7.4) method. The uniform linear motion is assumed in the experiments, which implies $h_i = 1/n$, $i = -u, \dots, u$. The experiments have been performed using Matlab programming language [38] on an Intel(R) Core(TM)i3 CPU M380 @ 2.53 GHz 64 bit system with 2 GB of RAM memory running on the Windows 7 Enterprise.

We consider the blurred image that has been degraded by a uniform linear motion in the horizontal direction that is modeled by the matrix equation (3.3.3). The length of the blurring process is denoted by l.

Data corresponding to standard test image 'Lena' are displayed on figures 4.5.1 and 4.5.2. The left (resp. right) graphic on Figure 4.5.1 displays data which are generated applying the Tikhonov (resp. TSVD) image deblurring method based on the FFT (Fast Fourier Transforms) algorithm. FFT is an efficient algorithm to compute matrix-vector multiplication [32], which is needed for spectral decomposition of the matrix A from (2.7.1). Data obtained by TIK image deblurring using the FFT algorithm and TSVD image deblurring using the FFT algorithm are denoted by $\mathrm{TIK}_{\mathrm{fft}}$ and $\mathrm{TSVD}_{\mathrm{fft}}$, respectively.

Figure 4.5.1 displays data corresponding to zero boundary conditions.

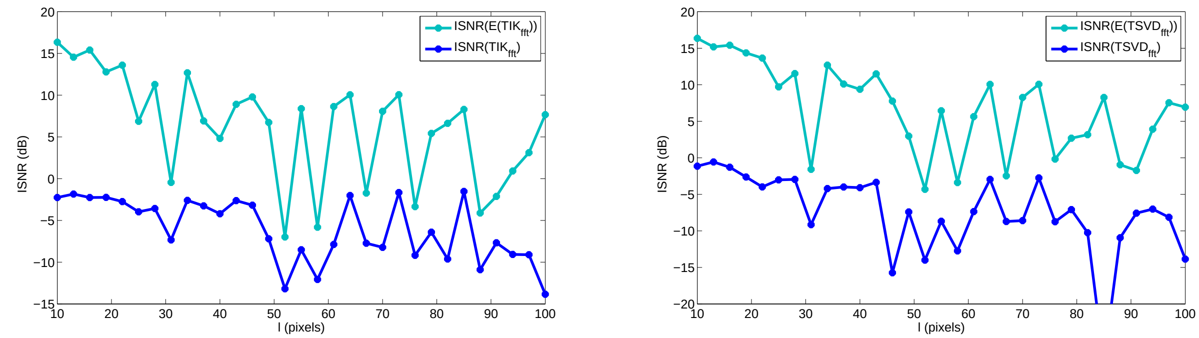
Figure 4.5.2 presents data corresponding to reflective boundary conditions.

Data obtained restoring the test image 'Barbara' are shown on figures 4.5.3 and 4.5.4.

In the following figure we show results for parameter $PSNR$ when we use periodic boundary condition in TIK and TSVD image deblurring with using of Kronecker decomposition [32]. Methods with this decomposition are presented with subscript SEP in the Figure 4.5.5.

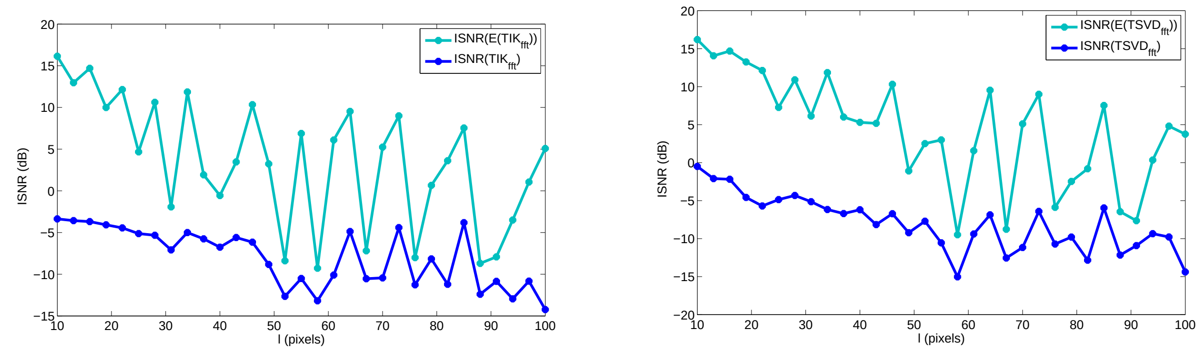
Confirmation that the image restorations obtained by (3.5.2) (i.e. $(E(TIK)$ and $E(TSVD))$ are better than the restoration produced by a straight application of the TIK and TSVD methods could be observed from figures 4.5.6, 4.5.7 and 4.5.8 for Lena image and different boundary conditions.

Results corresponding to Barbara image and periodic boundary conditions are presented on Figure 4.5.9.



(a) $\mathrm{TIK}_{\mathrm{fft}}$ (b) $\mathrm{TSVD}_{\mathrm{fft}}$

Figure 4.5.1: ISNR versus length l of the uniform blurring process for Lena and zero boundary conditions



(a) $\mathrm{TIK}_{\mathrm{fft}}$ (b) $\mathrm{TSVD}_{\mathrm{fft}}$

Figure 4.5.2: ISNR versus length l for Lena and reflective boundary conditions

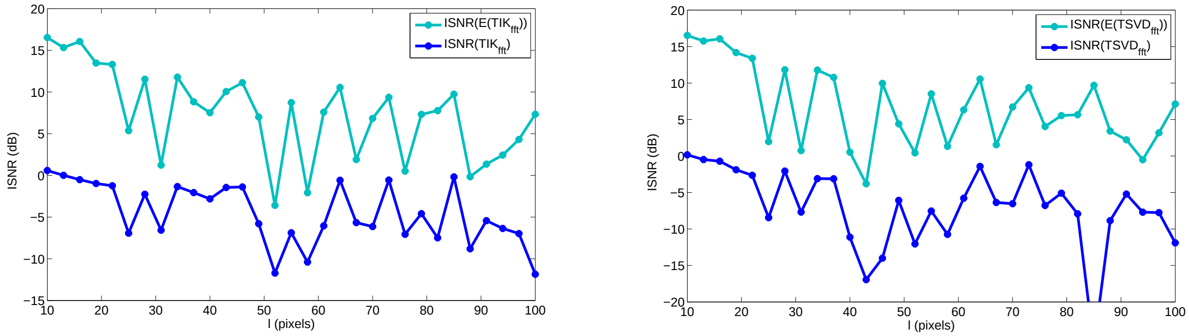
4.6 Comparison with the least squares solution

For the sake of simplicity let us again focus on the blur which is caused by a horizontal motion. In this section we compare the effectiveness of the method based on the usage of the Moore-Penrose inverse with the method which uses the least squares solution of the following linear system

$$g_{i,j} = \sum_{k=0}^{l-1} h_k f_{i,j+k}, \ \ i = 1, \dots, r, \ \ j = 1, \dots, m, \tag{4.6.1}$$

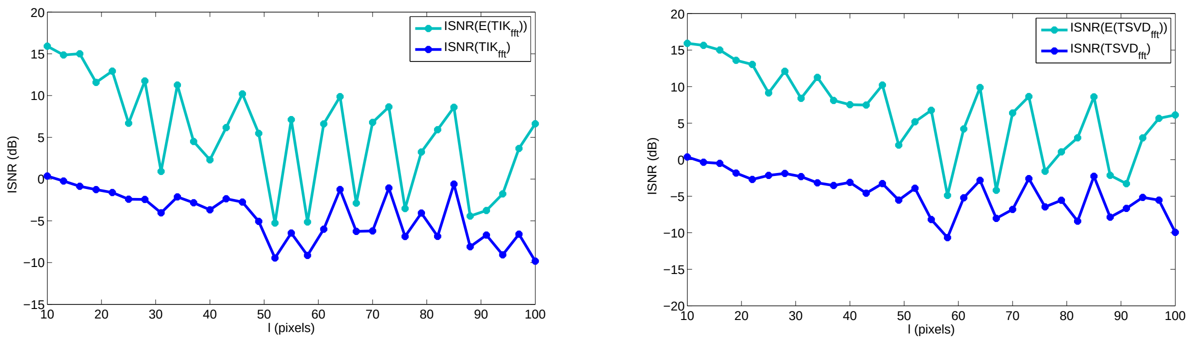
arising from (3.2.9)–(3.2.12) (with $n = m + l - 1$). Corresponding solution is derived using the standard `MATLAB` function `mrdivide()` and it will be denoted as LS solution in test examples. Function `mrdivide(B,A)` (or its equivalent `B/A`) performs matrix right division (forward slash) [56]. Matrices B and A must have the same number of columns.

If A is a square matrix, `B/A` is roughly the same as `B*inv(A)`. If A is an $n \times n$ matrix and B is a row vector with n elements, or a matrix with several such rows, then `X = B/A` is the solution of the equation $XA = B$ computed by using Gaussian elimination with partial pivoting. If B is an $m \times n$ matrix with $m \sim= n$ and A is a column vector with m components, or a matrix with several such columns, then `X = B/A` is a solution in the least-squares sense to the under- or overdetermined system of equations $XA = B$. In other words, X minimizes `norm(A*X -`



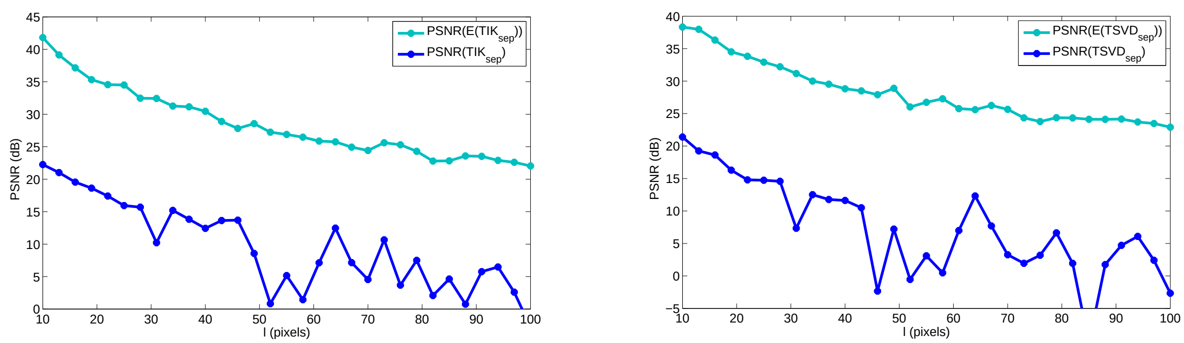
(a) $\mathrm{TIK}_{\mathrm{fft}}$ (b) $\mathrm{TSVD}_{\mathrm{fft}}$

Figure 4.5.3: ISNR versus length l for Barbara and zero boundary conditions



(a) $\mathrm{TIK}_{\mathrm{fft}}$ (b) $\mathrm{TSVD}_{\mathrm{fft}}$

Figure 4.5.4: ISNR versus length l for Barbara and reflective boundary conditions



(a) $\mathrm{TIK}_{\mathrm{sep}}$ (b) $\mathrm{TSVD}_{\mathrm{sep}}$

Figure 4.5.5: PSNR versus length l of the uniform blurring process periodic boundary conditions

`B)`. The rank k of A is determined from the QR decomposition with column pivoting. The computed solution X has at most k nonzero elements per column. If $k < n$, this is usually not the same solution as `X = B*pinv(A)`, which returns a least-squares solution with the smallest norm $\|X\|$. The presented results show that using the pseudo inverse approach leads to better improvements than solving the system directly.

In the case of the underdetermined or overdetermined system of equations $G = FH^T$ the least squares solution $F = G/H^T$ is usually not the same as the least square solutions of the

(a) Original image (b) Blurred image (c) TIK restore

(d) E(TIK) restore (e) TSVD restore (f) E(TSVD)

Figure 4.5.6: Removal of blur length $l = 34$ on a Lena image with zero boundary conditions.

(a) Original image (b) Blurred image (c) TIK restore

(d) E(TIK) restore (e) TSVD restore (f) E(TSVD)

Figure 4.5.7: Removal of blur length $l = 64$ on a Lena image with reflective boundary conditions.

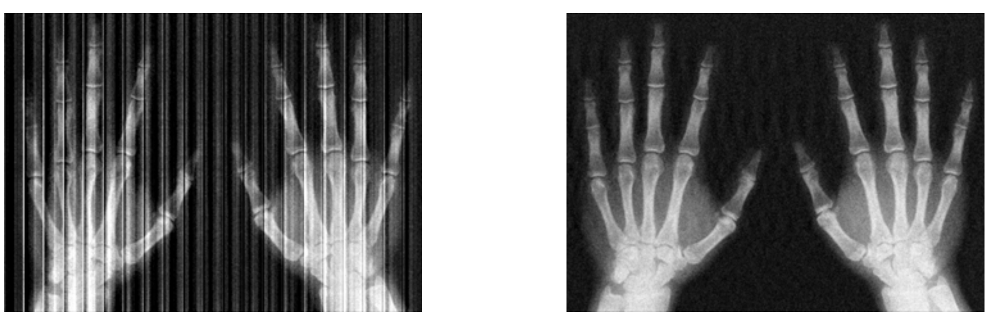
(a) Original image (b) Blurred image (c) TIK restore (d) E(TIK)

Figure 4.5.8: Removal of blur length $l = 46$ on a Lena image with periodic boundary conditions.

(a) Original image (b) Blurred image (c) TIK restore (d) E(TIK)

Figure 4.5.9: Removal of blur length $l = 46$ on a Barbara image with periodic boundary conditions.

minimal norm $F = G(H^T)^\dagger$. The comparison between the Moore-Penrose approach and least square solutions for the problem (3.2.12) is illustrated in the next figure. Blurring using in this example are uniform Gaussian with length $l = 50$ and Gaussian white noise of mean 0 and variance 0.05 . The left picture in Fig. 4.6.1 shows the restored image obtained as the direct solution of the system (4.6.1) while the right image shows the image in which the blur is restored based on the usage of the Moore-Penrose inverse.



(a) the LS solution (b) the Moore-Penrose inverse solution

Figure 4.6.1: Restoration arising from the LS solution and the Moore-Penrose inverse.

The difference in the quality of the restored images is in favor of the Moore-Penrose inverse approach and can be seen with the human eye. This illustration was confirmed by the corresponding values of the ISNR parameter, which are illustrated in Fig. 4.6.2.

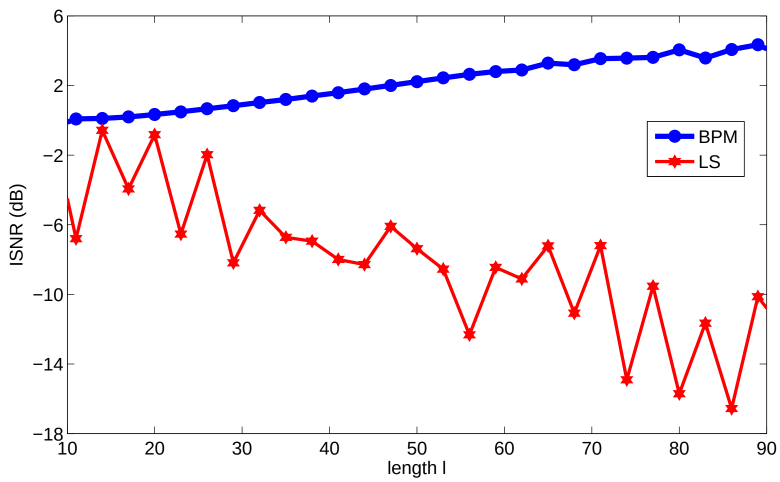


Figure 4.6.2: ISNR values arising from the Moore–Penrose inverse approach and the LS solution.

Chapter 5

Conclusion

In the Ph.D. dissertation, we analyzed, presented and developed new non-iterative methods for image restoration of images. The images are blurred or degraded by uniform or non–uniform linear motion. The methodology used is based on mathematical analysis, computer implementation and experimental tests. The environment in which the implementation has been developed and experimental tests have been made for the new image restoration methods is the software package `MATLAB`.

Conclusions for different presented non-iterative methods:

- *Application of the pseudoinverse computation in reconstruction of blurred images*: The method is based on the usage of the Moore-Penrose inverse solution of the matrix equation which presents a model of the motion blur. Our method exploits the structure of the blurring matrix and generates its pseudoinverse directly, without any iterations. The main advantage of the method is found in the decrease of the CPU time with respect to other methods for paseudoinverse computation. We illustrate the theoretical findings by comparing the Moore-Penrose inverse method against the Wiener filter, Constrained least-squares filter and Lucy-Richardson algorithm. Also, we present numerical results in which we compare our method (called MP method) with well-known $Pappas1$, $Pappas2$ and $Courrieu$ methods.

- *Application of partitioning method on specific Toeplitz matrices*: Motivated by the problem of restoring blurred images via well developed mathematical methods which are based on the Moore-Penrose inverse computation, we introduced a computational method to restore images that have been blurred by the non–uniform linear motion. The presented method is based on appropriate adaptations of well-known computational methods introduced in [93] and [29]. Using the specific structure of the matrix H as well as the fact that $H_m^\dagger$ can be computed easily, we adjust the partitioning methods in order to obtain the most efficient one.
  We compare our method with respect to two methods for fast computing the Moore-Penrose inverse introduced in [42, 46] and used in [12, 13, 14] as well as with the Courrieu method [17]. The main advantage of the proposed method is a significant reduction of the CPU time required to obtain the restored image compared to other methods based on the Moore-Penrose inverse approach. This fact agrees with the given theoretical results concerning analysis of computational complexities of algorithms.
  The introduced method can be used to restore a noisy X-ray image that has been blurred by non–uniform linear motion. Also, our method, except in radiography, can be used in

different practical realizations, such as restoration of images from ANPR systems. Additionally, the proposed method can be applied to real-life images that could be blurred due to camera motion or lack of focus.

- *Removal of blur in images based on least squares solutions*: The method is based on the least square solution of the matrix equation $XB = D$ with respect to unknown X. The objective of this method was the recovery of an image from degraded observations often caused by linear motion in several scientific areas including medical imaging and diagnosis, military surveillance, satellite and astronomical imaging, and remote sensing. By using the proposed method, the resolution of the restored image remains at a very high level, although the main advantage of the method was found on the improvements of $ISNR$ and $PSNR$ that has been increased considerably compared to the other methods and techniques. We present the results by comparing our method with the method based on the Moore-Penrose, which is a well established and tested method used for fast recovered and high resolution restored images.
The proposed method is an improvement of a method that has been tested against well known restoration methods (e.g. [12, 13] and the references within these papers) and has been found that provide better results and faster recovery against that methods. Moreover, it could find applications in more applied fields such as image watermarking (see [78]) in which case it has been also compared with well tested and current applied methods and was found to be working better against noise interference on the original image. Following that, the improvement version of the method also provides a more accurate more robust and faster recovery method.
The study shows that the proposed moments perform consistently better in terms of images reconstruction capability and invariant recognition accuracy in noisy or noise-free environment and can be potentially useful as feature descriptor in image analysis.

- *Image deblurring process based on separable restoration methods*: A new computational method is introduced that finds applications on image reconstruction and restoration. The method is based on the least square solution of the matrix equation representing an image that has been blurred by linear motion (of the general form $AXB = D$). The method has been tested on various images and measurements such as the ISNR and PSNR values which are used in order to indicate the advantage of the technique. Comparison of the method with several image restoration algorithms is also presented. The essence of the method is that it can be applied after the known image restoration algorithms. The main advantage of our method is found in the improvements of ISNR and PSNR that have been increased considerably compared to the other methods and techniques which are applied immediately before that method. In this study, we present the results by comparing our method with that of the Moore- Penrose inverse, the Wiener filter and other well established and tested methods used for fast recovering and high resolution restored images. The main difference between other image restoration methods and the proposed method, is that the already used methods use a matrix Y while our method is using $E(Y)$.
We also presented and tested the SMRS solutions although they were less preferred. The reason for this is the interesting property $E(Y) = Y$. We found that this property has been satisfied in the cases $Y = SMRS$, $Y = F$ and $Y = E_1(Y_1)$.
The proposed moments method performs consistently better in terms of image reconstruction capability and invariant recognition accuracy in a noisy or a noise-free environment

and can be potentially useful as feature descriptor in image analysis. Experimental results show that the proposed computational approach provides a better result on the image reconstruction than a blurred model, linear blur or not, especially in the case when applied for an incremental number of Haar or Fourier basis coefficients. The simulations presented in this method enable us to evaluate the current methods. Possible developments of the proposed algorithm can be generalized and find application in other scientific areas where the image restoration is needed, even if blur or noise is present.

- *Improving the Tikhonov and TSVD image deblurring methods*: We have introduced improving of the Tickhonov and TSVD method, based on the least square solution of the matrix equation $XB = D$ with respect to unknown X. By using the proposed improvement, the resolution of the restored image remains at a very high level. The main advantage of the proposed approach was found in the improvements of $ISNR$ and $PSNR$. We present the results by comparing our method with the Tickhonov and TSVD method, which is a well established and tested method used for fast recovered and high resolution restored images.
  The objective of this method was the recovery of an image from degraded observations often caused by linear motion. This approach can be applied in several scientific areas including medical imaging and diagnosis, military surveillance, satellite and astronomical imaging, and remote sensing.

In our future investigations we will try to develop and implement the methods for image restoration, when the characteristics of the degrading system or the blurring matrix is unknown. Also we will make an effort to develop mathematical methods for spatially variant out-of-focus blur removal and for spatially variant motion blur removal. Spatially variant blur is a type of deformation where different pixels in an image are blurred differently.

We consider that the results presented in the Ph.D. dissertation besides their contribution in the fields of image restoration, computational and applied mathematics will serve as a inspiration for future investigations in these fields, and implementation of the results in some practical realizations.

List of Algorithms

List of Figures

List of Tables

Bibliography

[1] P. Agrawal, P. J. Narayanan, Person De-Identification in Videos, IEEE Transactions on Circuits and Systems for Video Technology **21** No 3 (2011), 299–310.

[2] M.R. Banham, A.K. Katsaggelos, *Digital image restoration*, IEEE Signal Processing Magazine **14** (1997), 24–41.

[3] A. Ben-Israel, T.N. E. Greville, *Generalized Inverses. Theory and Applications, Second edition*, Springer-Verlag, New York, 2003.

[4] P. Bhimasankaram, *On Generalized Inverses of Partitioned Matrices*, Sankhya: The Indian Journal of Statistics, Series A **33** No 3 (1971), 311–314.

[5] J. Biemond, R.L. Lagendijk, R. M. Mersereau, *Iterative methods for image deblurring*, Proc. IEEE. **78(5)** (1990), 856–883.

[6] A. Bovik, *The essential guide to the image processing*, Academic Press, San Diego, San Francisko, New York, Boston, London, Sydney, Tokyo, 2009.

[7] A. Bovik, *Handbook of image and video processing*, Academic Press, San Diego, San Francisko, New York, Boston, London, Sydney, Tokyo, 2000.

[8] S. Chan, T.Q. Nguyen, *Fast LCD motion deblurring by decimation and optimization*, Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), (2009), 1201–1204.

[9] S. Chan, T.Q. Nguyen, *LCD Motion Blur Modeling and Simulation*, IEEE International Conference on Multimedia and Exposition, (2010), 400–405.

[10] S. Chan, D. Vo, T.Q. Nguyen, *Subpixel motion estimation without interpolation*, IEEE International Conference on Acoustics, Speech and Signal Processing, (2010), 722–725.

[11] T. F. Chan, C. K. Wong, *Total variation blind deconvolution*, IEEE Transactions on Image Processing **7** No 3 (1998), 370–375.

[12] S. Chountasis, V. N. Katsikis, D. Pappas, *Applications of the Moore-Penrose inverse in digital image restoration*, Math. Probl. Eng., Volume 2009, Article ID 170724, 12 pages doi:10.1155/2009/170724.

[13] S. Chountasis, V. N. Katsikis, D. Pappas, *Digital Image Reconstruction in the Spectral Domain Utilizing the Moore-Penrose Inverse*, Math. Probl. Eng., Volume 2010, Article ID 750352, 14 pages doi:10.1155/2010/750352.

[14] S. Chountasis, V. N. Katsikis, D. Pappas, *Image restoration via fast computing of the Moore-Penrose inverse matrix*, Systems, Signals and Image Processing, 2009, IWSSIP 2009.

[15] S. Chountasis, V.N. Katsikis, D. Pappas, *Image Reconstruction Methods for MATLAB Users A Moore-Penrose Inverse Approach*, MATLAB - A Fundamental Tool for Scientific Computing and Engineering Applications - Volume 1, ISBN: 978-953-51-0750-7, InTech, doi:10.5772/45811, (2012).

[16] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms, Second Edition*, MIT Press, Cambridge, Massachusetts London, McGraw-Hill Book Company, Boston, New York, San Francisco, St. Louis, Montreal, Toronto, 2001.

[17] P. Courrieu, *Fast Computation of Moore-Penrose Inverse Matrices*, Neural Information Processing - Letters and Reviews **8** No 2 (2005), 25–29.

[18] R. C. Craddock, G. A. James, P. E. Holtzheimer III, X. P. Hu, H. J. Mayberg, *A Whole Brain fMRI Atlas Generated via Spatially Constrained Spectral Clustering*, Human Brain Mapping **33** No 8 (2012), 1914–1928.

[19] I. Daubechies, *Ten lectures on wavelets*, Society of Industrial and Applied Mathematics, Philadelphia, 1992.

[20] L. R. Dice, *Measures of the Amount of Ecologic Association Between Species*, Ecology **26** No 3 (1945), 297–302

[21] L. Du, H. Ling, *Preservative License Plate De-identification for Privacy Protection*, In Proceedings of ICDAR, (2011), 468–472.

[22] S. A. Dudani, K. J. Breeding, R. B. McGhee, *Aircraft identification by moment invariants*, IEEE Trans. on Computers **C-26** No 1 (1977), 39–46.

[23] Y. W. (Daniel) Fan, J.G. Nagy, *Synthetic boundary conditions for image deblurring*, Linear Algebra Appl. **434** (2011), 2244–2268.

[24] N. Fisekovic, T. Nauta, H. Cornelissen, J. Bruinink, *Improved motion-picture quality of am-lcds using scanning backlight*, Proc. IDW, (2001), 1637–1640.

[25] G. Golub, C. Van Loan, *Matrix Computations*, 3nd Edition, Johns Hopkins University Press, 1996.

[26] R.C. Gonzalez, R.E. Woods, *Digital Image Processing*, 3rd Edition, Prentice-Hall, 2007.

[27] R. C. Gonzalez, R. E. Woods, S. L. Eddins, *Digital Image Processing Using MATLAB*, 2nd Edition, Prentice-Hall, 2009.

[28] F. Graybill, *Matrices and Applications to Statistics*, second ed., Wadsworth, Belmont, California, 1983.

[29] T.N.E. Grevile, *Some applications of the pseudo-inverse of matrix*, SIAM Rev. **3** (1960), 15–22.

[30] R. Gross, L. Sweeney, J. Cohn, F. d. l. Torre, S. Baker, *Face de-identification*, In A. Senior (Ed.), Protecting privacy in video surveillance, London: Springer-Verlag, (2009), 129–146.

[31] P.C. Hansen, *Regularization Tools version 4.0 for Matlab 7.3*, Numerical Algorithms, **46** (2007), 189–194.

[32] P.C. Hansen, J.G. Nagy, D.P. O'Leary, *Deblurring images: matrices, spectra, and filtering*, SIAM, Philadelphia, 2006.

[33] S. Har-Noy, T. Q. Nguyen, *LCD Motion Blur Reduction Using FIR Filter Banks*, 16th IEEE International Conference on Image Processing (ICIP), (2009), 1305–1308.

[34] F. H. Heesch, M.A. Klompenhouwer, *Spatio-temporal frequency analysis of motion blur reduction on LCDs*, International Conference on Image Processing, **4**(2007), 401–404.

[35] M. Hillebrand, C. H. Müller, *Outlier robust corner-preserving methods for reconstructing noisy images*, Ann. Statist. **35** (2007), 132–165.

[36] S. Hong, B. Berkeley, S. S. Kim, *Motion image enhancement of LCDs*, Proc. International Conference on Image Processing. IEEE, (2005).

[37] R. E. Hufnagel, N. R. Stanley, *Modulation Transfer Function Associated with Image Transmission through Turbulence Media*, Journal of the Optical Society of America **54** No 1 (1964), 52–60.

[38] *Image Processing Toolbox User's Guide*, The Math Works, Inc., Natick, MA, 2009.

[39] A. Janssons, *Deconvolution of Images and Spectra, Second edition*, Academic Press, New York, 1996.

[40] R.E. Kalaba, F.E. Udwadia, *Analytical Dynamics: A New Approach*, Cambridge University Press, Cambridge 1996.

[41] R.E. Kalaba, F.E. Udwadia, *Associative memory approach to the identification of structural and mechanical systems*, J. Optim. Theory Appl. **76** (1993), 207–223.

[42] S. Karanasios, D. Pappas, *Generalized inverses and special type operator algebras*, Facta Univ. Ser. Math. Inform. **21** (2006), 41–48.

[43] A.K. Katsaggelos, editor, *Digital image restoration*, Springer-Verlag, New York, 1991.

[44] A. K. Katsaggelos, *Iterative Image Restoration Algorithms*, Digital Signal Processing Handbook, Ed. Vijay K. Madisetti and Douglas B. Williams, Boca Raton: CRC Press LLC, 1999.

[45] V. N. Katsikis, D. Pappas, A. Petralias, *An improved method for the computation of the Moore-Penrose inverse matrix*, Applied Mathematics and Computation **217** No 23 (2011), 9828–9834.

[46] V. Katsikis, D. Pappas, *Fast computing of the Moore- Penrose Inverse matrix*, Electron. J. Linear Algebra **17** (2008), 637-650.

[47] M. Klompenhouwer, L. Velthoven, *Motion blur reduction for liquid crystal displays: Motion compensated inverse filtering*, Proceedings of SPIE-IS&T Electronic Imaging, SPIE, (2004).

[48] F. Krahmer, Y. Lin, B. Mcadoo, K. Ott, J. Wang, D. Widemann, B.Wohlberg, *Blind image deconvolution: Motion blur estimation*, In IMA Preprints Series, #2133-5, September 2006.

[49] D. Kundur, D. Hatzinakos, *Blind image deconvolution: an algorithmic approach to practical image restoration*, IEEE Signal Process. Mag. **13** No 3 (1996), 43–64.

[50] R. L. Lagendijk, J. Biemond, *Iterative Identification and Restoration of Images*, Kluwer Academic Publishers, Boston, MA, 1991.

[51] BW. Lee, K. Song, DJ. Park, Y. Yang, U. Min, S. Hong, C. Park, M. Hong, and K. Chung, *Mastering the moving image: Refreshing TFT-LCDs at 120 hz*, SID Symposium Digest of Technical Papers. SID, (2005).

[52] L. B. Lucy, *An iterative technique for the rectification of observed distributions*, Astronomical Journal **79** No 6 (1974), 745–754.

[53] P. J. Maher, *Some operator inequalities concerning generalized inverses*, Illinois J. Math. **34** (1990), 503–514.

[54] P. J. Maher, *Some norm inequalities concerning generalized inverses*, Linear Algebra Appl. **174** (1992), 99–110.

[55] P. J. Maher, *Some norm inequalities concerning generalized inverses, 2*, Linear Algebra Appl. **420** (2007), 517–525.

[56] *MATLAB 7 Mathematics*, The Math Works, Inc., Natick, MA, 2010.

[57] C. Meyer, *Matrix Analysis and Applied Linear Algebra*, SIAM, 2000.

[58] R. I. McCartney. *A liquid crystal display response time compensation feature integrated into an lcd panel timing controller*, In SID Digest, (2003).

[59] P. Milanfar, W. C. Karl, A. S. Willsky, *A moment-based variational approach to tomographic reconstruction*, IEEE Trans. on Image Processing **5** No 3 (1996), 459–470.

[60] S. Miljković, M. Miladinović, P. Stanimirović, I. Stojanović, *Application of the pseudoinverse computation in reconstruction of blurred images*, Filomat **26** No 3 (2012), 453–465, DOI: 10.2298/FIL1203453M.

[61] N. Mishima, G. Itoh, *Novel frame interpolation method for hold-type displays*, Proc. International Conference on Image Processing, **3** (2003), 1473–1476.

[62] E. Moore, *On the reciprocal of the general algebraic matrix*, Bull. Amer. Math. Soc. **26** (1920), 394-395.

[63] R. Mukundan, S. H. Ong, P. A. Lee, *Image analysis by Tchebychef moments*, IEEE Transactions on Image Processing **10** No 9 (2001), 1357–1364.

[64] R. Mukundan, *Improving image reconstruction accuracy using discrete orthonormal moments*, CISST 03 International Conference,(2003), 287–293.

[65] R. Mukundan, S.H. Ong, P.A. Lee, *Some Computational Aspects of Discrete Orthonormal Moments*, IEEE Trans. Image Process. **13** No 8 (2004), 1055–1059.

[66] T. B. Nguyen, B. J. Oommen, *Moment-Preserving piecewise linear approximations of signals and images*, IEEE Trans. on Pattern Analysis and Machine Intelligence **19** No 1 (1997), 84–91.

[67] M. T. Noda, I. Makino, T. Saito, *Algebraic methods for computing a generalized inverse*, ACM SIGSAM Bulletin **31** (1997), 51–52.

[68] H. Pan, X.-F. Feng, and S. Daly, *LCD motion blur modeling and analysis*, IEEE Int. Conf. Image Processing, (2005).

[69] H. Pan, X.-F. Feng, and S. Daly, *Quantitative analysis of LCD motion blur and performance of existing approaches*, Proc. SID Symp. Dig.Tech. Papers, **36** (2005), 1590-1593.

[70] M. Pawlak, *On the reconstruction aspects of moment descriptors*, IEEE Trans. on Information Theory **38** (1992), 1698–1708.

[71] R. Penrose, *A generalized inverse for matrices*, Proc. Cambridge Philos. Soc. **51** (1955), 406-413.

[72] R. Penrose, *On a best approximate solutions to linear matrix equations*, Proc. Cambridge Philos. Soc. **52** (1956), 17-19.

[73] V. B. Prasath, *A Well-posed multiscale regularization scheme for digital image denoising*, International Journal of Applied Mathematics and Computer Science **21** No 4 (2011), 769–777.

[74] M. A. Rakha, *On the Moore-Penrose generalized inverse matrix*, Applied Mathematics and Computation **158** (2004), 185–200.

[75] C.R. Rao, S.K. Mitra, *Generalized Inverse of Matrices and its Applications*, John Wiley and Sons, Inc, New York, London, Sydney, Toronto, 1971.

[76] C.R. Rao, *A note on a generalized inverse of a matrix with applications to problems in mathematical statistics*, J. R. Stat. Soc. Ser. B Stat. Methodol. **24** (1962), 152–158.

[77] W. H. Richardson, *Bayesian-Based Iterative Method of Image Restoration*, J. Opt. Soc. Am. **62** No 1 (1972), 55–59.

[78] M. Savelonas, S. Chountasis, *Noise-resistant watermarking in the fractional Fourier domain utilizing moment-based image representation*, Signal Processing **90** (2010), 2521–2528.

[79] D. C. Schuurman, D. W. Capson, *Video-rate eigenspace methods for position tracking and remote monitoring*, Fifth IEEE Southwest Symposium on Image Analysis and Interpretation, (2002), 45–49.

[80] Y. Shi, Q. Chang, *Acceleration methods for image restoration problem with different boundary conditions*, Appl. Numer. Math. **58** (2008), 602–614.

[81] N. Shinozaki, M. Sibuya, K. Tanabe, *Numerical algorithms for the Moore-Penrose inverse of a matrix: Direct methods*, Annals of the Institute od Statistical Mathematics **24** No 1 (1972), 193–203.

[82] A. Smoktunowicz, I. Wróbel, *Numerical aspects of computing the Moore-Penrose inverse of full column rank matrices*, BIT Numer. Math. **52** (2012), 503–524.

[83] P.S. Stanimirović, S. Chountasis, D. Pappas, I. Stojanović, *Removal of blur in images based on least squares solutions*, Mathematical Methods in the Applied Sciences (2013), DOI: 10.1002/mma.2751.

[84] P.S. Stanimirović, M. Miladinović, I. Stojanović, S. Miljković, *Application of partitioning method on specific Toeplitz matrices*, International Journal of Applied Mathematics and Computer Science, Accepted for publication.

[85] P. S. Stanimirović, I. Stojanović, S. Chountasis, D. Pappas, *Image Deblurring Process Based on Separable Restoration Methods*, Computational and Applied Mathematics (2013) DOI: 10.1007/s40314-013-0062-2.

[86] G.W. Stewart, *Introduction to Matrix Computation*, Academic Press, New York, 1973.

[87] I. Stojanović, P.S. Stanimirović, M. Miladinović, *Applying the algorithm of Lagrange multipliers in digital image restoration*, Facta Universitatis, Series Mathematics and Informatics **27** No 1 (2012), 41–50.

[88] I. Stojanović, P. Stanimirović, M. Miladinović, D. Stojanović, *Application of Non- Iterative Method in Image Deblurring*, Journal of Computer Science and Control Systems (JCSCS), ISSN 1844-6043, **5** No 1 (2012), 99–102.

[89] C. H. Teh, R. T. Chin, *On image analysis by the methods of moments*, IEEE Trans. Pattern Anal. Machine Intell. **10** (1988), 496–513.

[90] M. R. Teague, *Image analysis via the general theory of moments*, Journal of the Optical Society of America **70** No 8 (1980), 920–930.

[91] H. J. Trussell, S. Fogel, *Identification and Restoration of Spatially Variant Motion Blurs in Sequential Images.*, IEEE Trans. Image Proc **1** (1992), 123–126.

[92] F.E. Udwadia, R.E. Kalaba, *An Alternative Proof for Greville's Formula*, J. Optim. Theory Appl. **94** (1997), 23–28.

[93] F.E. Udwadia, R.E. Kalaba, *General Forms for the Recursive determination of generalized inverses: Unified approach*, J. Optim. Theory Appl. **101** (1999), 509–521.

[94] G. Wang, Y. Wei, S. Qiao, *Generalized Inverses: Theory and Computations*, Science Press, 2003.

[95] Y. Wei, J. Cai, M. K. Ng, *Computing Moore-Penrose inverses of Toeplitz matrices by Newton's iteration*, Math. Comput. Model. **40** (2004), 181–191.

[96] Q. Xiao, X. Hu, L. Zhang, *The symmetric minimal rank solution of the matrix equation $AX{=}B$ and the optimal approximation*, Electron. J. Linear Algebra **18** (2009), 264–273.

[97] Z. Xu, *On Moore-Penrose inverse of Toeplitz matrices*, Linear Algebra Appl. **169** (1992), 9-15.

[98] S.Yahyanejad, J. Ström, *Removing motion blur from barcode images*, IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), (2010), 41–46.

[99] Y. L. You, M. Kaveh, *Blind image restoration by anisotropic regularization*, IEEE Transactions on Image Processing **8** No 3 (1999), 396–407.

[100] F. Zhang, *Matrix Theory: Basic Results and Techniques*, New York: Springer-Verlag, 1999.

[101] G. Zielke, *Report on test matrices for generalized inverses*, Computing **36** (1986), 105–162.

[102] S. Zohar, *Toeplitz matrix inversion: The algorithm of W.R. Trench*, J. ACM **16** (1969), 592–601.

Biografija

Igor Stojanović rođen je 01.02.1974 godine u Kumanovu, gde je sa odličnim uspehom završio osnovnu školu i Gimnaziju "Goce Delčev". Studije na smeru za Elektroniku i telekomunikacije na Elektrotehničkom fakultetu u Skoplju upisao je školske 1992/93 godine, a završio ih je 1997 sa odbranom diplomskog rada iz oblasti digitalne obrade signala.

Postdiplomske studije na Elektrotehničkom fakultetu u Skoplju , smer elektronika, upisao je u oktobru 1997 godina. Nakon polaganja predviđenih ispita sa prosecnom ocenom 10, magistrirao je u decembru 2002 sa odbranom magistarskog rada pod nazivom "Pretraživanje JPEG slika i lociranje objekata multirezolucijskom analizom" pod mentorstvom prof. dr Momcila Bogdanova.

Govori tečno engleski, a služi se i francuskim jezikom.

Posle diplomiranja radio je u Sektoru za informacione i komunikacione tehnologije u Upravi carina Republike Makedonije od septembra 1999 do decembra 2007 godina. Radio je na različitim pozicijama: sistem inženjer za operativne sisteme, baze podataka i pomoćnik direktora IKT sektora. Imao je obuke za baze podataka (Oracle i Informix u Sloveniji i Belgiji), Unix administracija (ECS Makedonija), administriranje mreže (Cisco, Belgija, Makedonija i Slovenija). Od Januara 2008 je zaposlen kao asistent na Fakultetu za informatiku pri Univerzitetu Goce Delčev - Štip.

Aktivno je učestvovao u realizaciji projekta "Procena rizika za carinu na zapadnom Balkanu". Projekat je finansiran sa strane Evropske komisije, "Information Society Technologies" - Sixth Framework Programme (2002-2006)).

Akademske doktorske studije iz oblasti Informatike (računarskih nauka) na Prirodno-matematičkom fakultetu u Nišu upisao je školske 2010/11. godine. Oblasti njegovog naučnog interesovanja su restauracija oštećenih slika, obrada slika, digitalna obrada signala, numerička linearna algebra.

Na Fakultetu za informatiku u Štipu izvodio je nastavu iz sledećih predmeta:

1. Objektno orijentisano programiranje
2. Digitalna logika
3. Operativni sistemi
4. Arhitektura kompjutera
5. Mrežni operativni sistemi

U međuvremenu je napisao i objavio sledeće naučne radove.

(a) Radovi u časopisima koji imaju Impact Factor:

1. P. Stanimirović, M. Miladinović, **I. Stojanović**, S. Miljković, *Application of partitioning method in removal of blur in images*, International Journal of Applied Mathematics and Computer Science, (IF 2012=1.008) (Accepted).

2. P. Stanimirović, **I. Stojanović**, S. Chountasis, D. Pappas, *Image Deblurring Process Based on Separable Restoration Methods*, Computational and Applied Mathematics, ISSN: 0101-8205 (Print) 1807-0302 (Online), DOI: 10.1007/s40314-013-0062-2 (IF 2012=0.452).

3. P. Stanimirović, S. Chountasis, D. Pappas, **I. Stojanović**, *Removal of blur in images based on least squares solutions*, Mathematical Methods in the Applied Sciences, Print ISSN: 0170-4214, Online ISSN: 1099-1476, DOI:10.1002/mma.2751 (IF 2012=0.778).

4. S. Miljković, M. Miladinović, P. Stanimirović, **I. Stojanović**, *Application of the pseudoinverse computation in reconstruction of blurred images*, Filomat **26(3)** (2012), 453-465, DOI: 10.2298/FIL1203453M (IF2012=0.714).

5. **I. Stojanović**, S. Markovski, C. Martinovska, A. Mileva, *Application of the progressive wavelet correlation for image recognition and retrieval from the collection of images*, Technics Technologies Education Management, ISSN: 1840-1503, **7(4)** (2012), 1550-1560 (IF 2012=0.414).

(b) Radovi u časopisima koji nemaju Impact Factor:

6. **I. Stojanović**, A. Mileva, D. Stojanović, I. Kraljevski, *Image Recognition by Using the Progressive Wavelet Correlation*, International Journal of Image, Graphics and Signal Processing (IJIGSP), ISSN: 2074-9074(Print), ISSN: 2074-9082 (Online), **4(9)** (2012), 1-7, DOI:10.5815/ijigsp.2012.09.01.

7. **I. Stojanović**, Z. Zdravev, A. Tasevski, *Progressive Wavelet Correlation as a Tool for Recognition of the Images*, Journal of Computer Science and Control Systems (JCSCS), ISSN 1844-6043, **5(2)** (2012), 33-38.

8. **I. Stojanović**, P. Stanimirović, M. Miladinović, D. Stojanović, *Application of Non-Iterative Method in Image Deblurring*, Journal of Computer Science and Control Systems (JCSCS), ISSN 1844-6043, **5(1)** (2012), 99-102.

9. **I. Stojanović**, P. Stanimirović, M. Miladinović, *Applying the Algorithm of Lagrange Multipliers in Digital Image Restoration*, FACTA UNIVERSITATIS, Series Mathematics and Informatics, ISSN 0352-9665, **27(1)** (2012), 41-54.

Прилог 1.

ИЗЈАВА О АУТОРСТВУ

Изјављујем да је докторска дисертација, под насловом

НЕИТЕРАТИВНИ МЕТОДИ ЗА ДИГИТАЛНУ РЕСТАУРАЦИЈУ СЛИКА

- резултат сопственог истраживачког рада,
- да предложена дисертација, ни у целини, ни у деловима, није била предложена за добијање било које дипломе, према студијским програмима других високошколских установа,
- да су резултати коректно наведени и
- да нисам кршио/ла ауторска права, нити злоупотребио/ла интелектуалну својину других лица.

У Нишу, 12.11.2013

Аутор дисертације: Игор Стојановић

Потпис докторанда:

Прилог 2.

ИЗЈАВА О ИСТОВЕТНОСТИ ШТАМПАНЕ И ЕЛЕКТРОНСКЕ ВЕРЗИЈЕ ДОКТОРСКЕ ДИСЕРТАЦИЈЕ

Име и презиме аутора: Игор Стојановић

Студијски програм: РАЧУНАРСКЕ НАУКЕ

Наслов рада: НЕИТЕРАТИВНИ МЕТОДИ ЗА ДИГИТАЛНУ РЕСТАУРАЦИЈУ СЛИКА

Ментор: Проф. др Предраг Станимировић

Изјављујем да је штампана верзија моје докторске дисертације истоветна електронској верзији, коју сам предао/ла за уношење у **Дигитални репозиторијум Универзитета у Нишу.**

Дозвољавам да се објаве моји лични подаци, који су у вези са добијањем академског звања доктора наука, као што су име и презиме, година и место рођења и датум одбране рада, и то у каталогу Библиотеке, Дигиталном репозиторијуму Универзитета у Нишу, као и у публикацијама Универзитета у Нишу.

У Нишу, 12.11.2013

Аутор дисертације: Игор Стојановић

Потпис докторанда:

Прилог 3.

ИЗЈАВА О КОРИШЋЕЊУ

Овлашћујем Универзитетску библиотеку „Никола Тесла" да, у Дигитални репозиторијум Универзитета у Нишу, унесе моју докторску дисертацију, под насловом:

НЕИТЕРАТИВНИ МЕТОДИ ЗА ДИГИТАЛНУ РЕСТАУРАЦИЈУ СЛИКА

која је моје ауторско дело.

Дисертацију са свим прилозима предао/ла сам у електронском формату, погодном за трајно архивирање.

Моју докторску дисертацију, унету у Дигитални репозиторијум Универзитета у Нишу, могу користити сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons), за коју сам се одлучио/ла.

1. Ауторство
2. Ауторство – некомерцијално
3. Ауторство – некомерцијално – без прераде
4. Ауторство – некомерцијално – делити под истим условима
5. Ауторство – без прераде
6. Ауторство – делити под истим условима

(Молимо да подвучете само једну од шест понуђених лиценци; кратак опис лиценци је у наставку текста).

У Нишу, 12. 11. 2013

Аутор дисертације: Игор Стојановић

Потпис докторанда: