Charles University in Prague
Faculty of Mathematics and Physics

DIPLOMA THESIS

Jozef Sabo

Image de-blurring using two images with different exposure times

Department of Software and Computer Science Education

Supervisor: Ing. Filip Šroubek, Ph.D.

Study Program: Computer Science, Software Systems

2012

I would hereby like to thank the thesis supervisor, *Ing. Filip Šroubek, PhD.*, for his patience, goodwill and advice provided to me while writing this thesis and to my family and friends for continued support.

I declare that I wrote the thesis by myself and listed all used sources. I agree to make the thesis publicly available.

Prague, April 13, 2012   Jozef Sabo

Contents

List of Figures

List of Tables

Název práce: Odstranění rozmazání pomocí dvou snímků s různou délkou expozice

Autor: Jozef Sabo

Katedra (ústav): Kabinet software a výuky informatiky

Vedoucí diplomové práce: Ing. Filip Šroubek, Ph.D.

e-mail vedoucího: sroubekf@utia.cas.cz

Abstrakt: V předložené práci studujeme metody odstranění rozmazání pomocí dvou snímků stejné předlohy s různou dobou expozicie, přičemž se soustředíme na dvě hlavní kategorie těchto metod, tzv. dekonvoluční a nedekonvoluční. U obou kategorií rozebíráme jejich teoretické základy a zkoumáme jejich výhody a omezení. Samostatnou kapitolu věnujeme vyhodnocení a srovnání kategorií metod na testovacích datech (obrázky), k testování používáme metody implementovány v jazyku MATLAB. Účinnost zkoumaných metod srovnáváme i s vybraným odšumovacím algoritmem pracujícím s jedním vstupním obrázkem. Nesoustředíme se na výpočetní složitost srovnávaných algoritmů a pracujeme pouze se šedotónovými obrázky.

Klíčová slova: krátká a dlouhá, expozice, odstranění rozmazání, šum, odšumení, dekonvoluce

Title: Image de-blurring using two images with different exposure times
Author: Jozef Sabo
Department: Department of Software and Computer Science Education
Supervisor: Ing. Filip Šroubek, Ph.D.
Supervisor's e-mail address: sroubekf@utia.cas.cz

Abstract: In the presented work we study methods of image de-blurring using two images of the same scene with different exposure times, focusing on two main approach categories, so called deconvolution and non-deconvolution methods. We present theoretical backgrounds on both categories and evaluate their limitations and advantages. We dedicate one section to a comparison of both method categories on test data (images) for which we use our MATLAB implementation of the methods. We also compare the effectiveness of said methods against the results of a selected single-image de-noising algorithm. We do not focus at computational efficiency of algorithms and work with grayscale images only.

Keywords: long and short, exposure, image de-blurring, noise, denoising, deconvolution

Chapter 1

Introduction

Image restoration[1] techniques have always been a subject of great interest within the domain of digital image processing. Their significance increases as digital photography undergoes rapid development with many digital imaging devices becoming readily available in countless forms, such as cell-phones, cameras and video cameras, to name a few.

We know from experience that it is quite difficult to obtain a quality image especially in insufficient lighting conditions requiring longer exposure times. In other words, taking "bad images" is easy. One of the most prevalent image degrading factors in photography is *motion blur*, caused either by motion in the photographed scene, the camera itself, or both. Suppression or complete removal of motion blur is highly desirable especially in hand-held devices.

According to [1], there are essentially two categories of approaches to deal with motion blur, the so-called *in-process* and *post-process*. The former focus at improving the conditions at which the image is being taken, usually by hardware means (e.g. image stabilizers), while the latter aim to correct the effects of motion blur after the image was taken. Widespread deployment of in-process-capable devices is however limited due to their high prices and as a result, the need for effective *post-process* algorithms arises.

In image processing, motion blur is modeled by discrete convolution. If a convolution operator (usually called *point-spread function* or *PSF*, sometimes more simply *kernel*) of motion blur is known (this applies to camera motion as local blurring caused by movement of objects in the scene tend to lead to *PSF* that is location-dependent - which is beyond the scope of this thesis; however, some of

---

[1]Image restoration aims to reconstruct the original pre-degradation image as faithfully as possible as opposed to image enhancement, which aims to make desired features more visible. Contrast ehnancement, sharpening etc. are examples of image enhancement operations.

the methods to be mentioned below are capable of handling space-variant blur-ring naturally) the original image can be recovered by a deconvolution algorithm, such as Lucy-Richardson [2]. In most cases, there is little or no prior information on blur PSF which requires us to employ a *blind deconvolution* algorithm. First, a PSF is estimated from a given image or a set of images and subsequently, an existing *non-blind deconvolution* algorithm is used.

Blind deconvolution from a single image is a highly ill-posed problem that lacks a general solution. Blind deconvolution from multiple images (more commonly known as *multichannel blind deconvolution*) of the same scene subject to varying degrees of degradation (blur, noise), is better posed and leads to better results. The subject of this thesis is a special case of the multichannel blind deconvolution problem where we obtain two images of the same scene using different exposure times. The first image is taken using a long exposure time resulting in proper level of lighting but is degraded by motion blur caused by camera shake, whereas the second image is taken using a short exposure time and is not affected by motion blur, but may be darker. Both images are affected by noise (the short exposure image is more noisy), which is dependent on the *ISO* sensitivity setting of the digital camera; we elaborate on this in greater detail in section 2.4. In practice, such a pair of images is easy to obtain by using a feature called *ISO bracketing* which is commonly available in digital SLRs.

There are several approaches how to restore the original image from the short and long exposure image pairs. These broadly fall in two categories – the *non-deconvolution* and *deconvolution* methods. The former do not perform deconvolution at any stage and try to utilize the image information in other ways as opposed to the latter, where deconvolution is performed at some point. Both categories feature methods with varying complexity, computational cost and efficiency, some of which will be described below. Section 2.3 elaborates on the problem of deconvolution.

The aim of this thesis is to analyze selected methods and evaluate their degree of suitability in various circumstances in terms of exposure times, signal-to-noise ratio improvements and other indicators based on experimental results. Methods are compared against themselves and we also compare them with a chosen single-image denoising algorithm to evaluate the benefits of additional image information in the form of improved quality of the restored image. We do not attempt to implement computationally effective algorithms at all costs nor do we provide an exhaustive list of all available methods. We perform experiments on both real-world and simulated data but focus on the latter, which allow the best degree of control over the experiment. Images are assumed to be gray-scale[2]

---

[2]Methods can be extended to process color images simply by applying them to each color channel separately. This is, however, not the subject of this thesis.

and registered. Real-world experiment images are registered manually if needed.

Chapter 2

Research

In this chapter, we present a theoretical background to a wide array of subjects that are related to the problem of image restoration from the blurred - noisy (long and short exposure) image pair. We begin by presenting the model of our situation in section 2.1, which follows this introduction. We briefly describe several methods of solving the the problem in section 2.2. Since one of the methods we test makes use of deconvolution, we elaborate more on the subject in section 2.3. To be able to generate testing data, we need to understand the nature of noise in digital photography and its relationship to digital camera settings. We discuss this matter in section 2.4. The BM3D algorithm, which we use as a reference alongside the other two methods, is briefly described in section 2.6. Both BM3D denoising algorithm and our extension of the Šroubek's multichannel blind deconvolution method assume image noise to be known a priori; we have therefore studied and used several methods to estimate it from images, which are desribed in section 2.5. The chapter is concluded with the description of Tico's wavelet image fusion algorithm, which we've chosen as a representative non-deconvolution method and Šroubek's multichannel blind deconvolution method including our extension, which enables it to operate on data featuring different noise in each channel.

2.1 Model

For the purpose of de-blurring algorithms listed in the following chapters, we present a mathematical model of the short and long exposure image pair as mentioned above. Let $f(\mathbf{x})$ be the image function of the original, discrete, non-degraded grayscale image N^2 pixels in size. Let $g_2(\mathbf{x})$ be the image function of the original image subject to blurring and additive noise and finally let $g_1(\mathbf{x})$ be the image function of the underexposed image subject to additive noise only. According to Tico [4] we have:

$$\begin{aligned}
\alpha g_1(\mathbf{x}) &= f(\mathbf{x}) &&+ n_1(\mathbf{x}) \\
g_2(\mathbf{x}) &= f(\mathbf{x}) \star d(\mathbf{x}) &&+ n_2(\mathbf{x})
\end{aligned} \tag{2.1}$$

where $\mathbf{x} = (x, y)$ are pixel coordinates, $d(\mathbf{x})$ is the point-spread function describing motion blur, α is the change in brightness as a result of shorter exposure (clearly $\alpha \leq 1$), $\star$ denotes the convolution operation and n_1, n_2 are noise terms. We assume additive Gaussian noise with $\mu_1 = \mu_2 = 0$ and $\sigma_1^2 \gg \sigma_2^2$ (where μ_i and σ_i^2 are the mean and variance respectively). We assume that g_1 is not blurred; however, the multichannel blind deconvolution method by Šroubek and Flusser (see section 2.8) and our extension can also handle blurring in the noisy image. Please note that in the following text, we denote the size of image f as $m_f n_f$ and the size of blur PSF as $m_d n_d$.

It is important to note that the assumption of zero-mean Gaussian noise is only used for the de-blurring process. Experimental data are generated in accordance with Foi's [3] noise model and Petteri M. Ojala's measurements [15]. These assumptions simplify theoretical derivation of de-blurring alghorithms; however, they do not significantly affect their efficiency, as is shown in experimental results.

2.2 Blurred-noisy image pair problem

Broadly put, solving the blurred-noisy (or long and short exposure) image pair problem means restoring the best possible original image $\hat{f}$ only using the information available from both input images. This is often expressed in terms of mean square error, that is, using the notation from equation 2.1 to find an estimate $\hat{f}$ of the original image f such that $||\hat{f} - f||_2$ is minimized.

Several approaches can be taken:

1. Enhance the short-exposure (noisy) image g_1 by removing noise, not considering the long-exposure (blurred) image g_2 at all. The BM3D denoising

algorithm [20], used as a reference, fits into this category.

2. Enhance the short-exposure (noisy) image g_1 by using image statistics (standard deviation, mean) of the long-exposure (blurred) image g_2. This is the approach taken by Reinhard et al. [14]. A similar approach, aimed at enhancing image brightness and contrast by successive applications of 2 specialized functions is used by Razligh and Kehtarnavaz [5]. Since noise is present and is amplified by both methods, noise removal is necessary.

3. Regardless of α, perform blind deconvolution of the long-exposure (blurred) image g_2 making use of the image information present in short-exposure image g_1. Attempting blind deconvolution using only image g_2 is an under-determined problem yielding unsatisfactory results. This is the approach taken by Lim and Silverstein [23], Tico in his 2006 and 2007 papers [4] [11], Babacan et al. [25], Yuan et al. [10] and others. The MCAM algoritm of Šroubek and Flusser [16], used in this thesis, is a multi-channel, multi-PSF generalization of the long-short exposure image pair problem, and is extended to accomodate different levels of noise in each channel. A more detailed description of the deconvolution problem is presented in section 2.3, Šroubek and Flusser's MCAM algorithm is presented in section 2.8.

4. Fuse images g_1 and g_2 without using deconvolution. The wavelet-based image fusion by Tico [6], described in section 2.7 and used in this thesis, is a fitting example. The primary advantage of this algorithm stems from its inherent independence on the type of blurring, which can even be local (movement of individual objects) or space-variant.

Figure 2.1 presents a block diagram of the blurred-noisy image pair problem and the approaches to solving it.
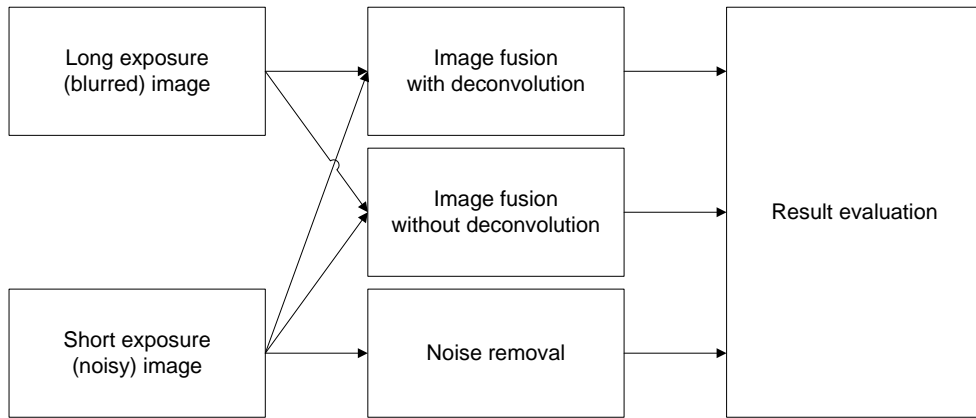

Figure 2.1: Approaches to solving the blurred-noisy image pair problem.

8

2.3 Deconvolution problem

Referring to our model in equation 2.1, we have the image g_2 corrupted by blurring and noise, which is expressed as

$$g_2(\mathbf{x}) = f(\mathbf{x}) \star d(\mathbf{x}) + n_2(\mathbf{x}) \qquad (2.2)$$

Our task is to reverse the blurring (convolution) process, remove as much noise as possible and present the best possible estimate $\hat{f}$ of the original image f. Quality of the estimation is assessed both visually and quantitatively, the usual requirement being the minimization of mean squared error between the estimate and the original image. In many cases we also need to recover the blur PSF $d(\mathbf{x})$.

Deconvolution is a fairly non-trivial task. The amount of prior information about degradation, i.e., the size or shape of blurs, and the noise level, determines how mathematically ill-posed the problem is. Even non-blind restoration, when blurs are available, is in general an ill-posed problem because of zeros in the frequency domain of the blurs [16]. Without any prior knowledge of the blur PSF $d(\mathbf{x})$, we employ a *blind deconvolution* algorithm, which either provides the estimate in a single pass, or iteratively. The algorithm, based on assumptions that can be made, first estimates the blur PSF and then proceeds to remove the blur using a known, *non-blind* deconvolution algorithm.

In the absence of noise and assuming the convolution operator (impulse response) to be known, constructing an *inverse filter* may often be the first approach. The idea for an inverse filter lends itself directly from the properties of convolution which becomes multiplication in the frequency domain. Denoting the Fourier transform of a function with a capital letter, the situation becomes $G = FD$ from which we can directly obtain $F = GD^*|D|^{-2}$, where * denotes complex conjugation.

An improved solution presented by Wiener in 1949 aims to minimize mean square error in the presence of noise and can be expressed as

$$\hat{F}(f) = G(f)\frac{1}{D(f)}\left[\frac{|D(f)|^2}{|D(f)|^2 + \frac{1}{\text{SNR}(f)}}\right] \qquad (2.3)$$

where SNR is the signal-to-noise ratio.

The requirement of mean square error minimization is often a starting point not only for the design of deconvolution algoritms, but optimization problems in general. However, due to the ill-posed nature of the problem, simple mean square error minimization results rarely prove satisfactory. Regularization, or in other

words introducing additional information in order to attain a better final solution, is often employed both to improve numerical stability of the computation and to enforce desired properties on the eventual result. Usually, regularization has the form of penalty for complexity. Regularization terms in the energy functional used by Šroubek and Flusser's MC-AM algorithm, discussed in section 2.8, serve as a good example; the term $Q(f) = \int |\nabla f|$ yields higher values, thus in effect penalizes solutions that are "rough", in other words, enforces smoothnes on the estimated image function f.

A different approach is to view deconvolution, or image restoration as a whole, in terms of Bayesian probability. Given the observed images $g_1, g_2 \ldots g_P$, the ideal restored image $\hat{f}$ is one for which the posterior probability $p(f|g_1 \ldots g_P)$ is maximized, in other words

$$\hat{f} = \arg\max_f p(f|g_1 \ldots g_P) \tag{2.4}$$

In case we also need to find blur estimates d_i, the equation becomes

$$\hat{f}, \hat{d_1} \ldots \hat{d_P} = \arg\max_{f, d_1 \ldots d_P} p(f, d_1 \ldots d_P|g_1 \ldots g_P) \tag{2.5}$$

From the properties of conditional probability, we can write

$$p(f, d_1 \ldots d_P|g_1 \ldots g_P) = \frac{p(g_1 \ldots g_P|f, d_1 \ldots d_P)p(f)p(d_1 \ldots d_P)}{p(g_1 \ldots g_P)} \tag{2.6}$$

An estimate obtained this way is called maximum a posteriori (MAP). This approach to defining the objective function definition is taken by Tico [4] [11], Jia et al. [1]

It is important to note that although taking the mean square error and probability approach to solving the problem may seem different from MSE at first, both are an expression of the same principle. Thus, in general, almost all image restoration methods define an energy or objective function of the original, ideal image function, often including additional regularization terms.

The deconvolution process can significantly benefit from the presence of noisy image g_1, as it improves initial estimate of both $\hat{f}$ and $\hat{d}$.

2.4 Image noise in photography

In conventional photography, the photographic material's sensitivity to light is determined by the size of silver halide grains embedded in its emulsion. The

larger a grain is, the more photons can it capture increasing the probability of exposure. Upon illumination, grains develop in an all-or-nothing fashion meaning that a grain decomposes into silver completely or not at all. This produces the characteristic "film grain" that is especially present in highly sensitive material used to shoot fast-moving scenes with very short shutter times.

Several systems were used to designate film sensitivity in the past. Among the most common were ASA, DIN and GOST. The ASA and DIN scales were incorporated into the new ISO standard published in 1987 (ISO 5800:1987) as the ISO arithmetic scale and logarithmic scale, respectively. The ISO logarithmic scale gradually fell out of use in favor of the ISO arithmetic scale.

The process of capturing an image in a digital camera is somewhat different. Photographic material is replaced by an electronic image sensor which consists of an array of individual cells capable of converting light into electrical signals. This signal is then amplified, digitized and stored. Known types of image sensors include the CCD (*charge-coupled devices*) or CMOS (*complementary metal-oxide semiconductors*). CCD cells store captured light as an electrical charge until they are read (one at a time) whereas circuitry attached to each CMOS cell converts light energy into voltage directly. Neither technology has a clear advantage; CMOS sensors are however cheaper to manufacture.

Digital cameras typically allow the user to select from several ISO settings. This is made possible by varying the amplification factor affecting the signal leaving the image sensor. Since no image sensor is completely free of noise, amplification of the signal also amplifies noise, resulting in a lowered signal-to-noise ratio. Foi in [3] demonstrated that it is possible to model digital camera noise as two independent Gaussian (accounting for electrical and thermal noise) and Poissonian (accounting for the photon capturing process) components as follows:

$$g(\mathbf{x}) = f(\mathbf{x}) + \eta_p(f(\mathbf{x})) + \eta_g(\mathbf{x}) \tag{2.7}$$

where $\mathbf{x}$ is the pixel coordinate, $f(\mathbf{x})$ is the original image, $g(\mathbf{x})$ is the observed image and η_p and η_g are Poissonian and Gaussian noise components, respectively. In terms of distributions, the equation becomes

$$\chi(f(\mathbf{x}) + \eta_p(f(\mathbf{x}))) \sim P(\chi f(\mathbf{x}))$$
$$\eta_g(\mathbf{x}) \sim N(0, b) \tag{2.8}$$

here $\chi \geq 0$ and $b \geq 0$ are real scalar parameters and N and P denote normal (i.e. Gaussian) and Poissonian distributions, respectively. Using $E\{\}$ for the expected value and $var\{\}$ for the variance of a random variable, we obtain

$$E\{\chi(f(\mathbf{x}) + \eta_p(f(\mathbf{x})))\} = var\{\chi(f(\mathbf{x}) + \eta_p(f(\mathbf{x})))\} = \chi f(\mathbf{x}) \tag{2.9}$$

from the properties of the Poisson distribution. Since

$$E\{\chi(f(\mathbf{x}) + \eta_p(f(\mathbf{x})))\} = \chi f(\mathbf{x}) + \chi E\{\eta_p(f(\mathbf{x}))\} \tag{2.10}$$

and

$$\chi^2 var\{\eta_p(f(\mathbf{x}))\} = \chi f(\mathbf{x}) \tag{2.11}$$

it follows that

$$E\{\eta_p(f(\mathbf{x}))\} = 0 \text{ and } var\{\eta_p(f(\mathbf{x}))\} = \frac{f(\mathbf{x})}{\chi} \tag{2.12}$$

The Poissonian η_p thus has variable variance that depends on the value of $f(\mathbf{x})$, $var\{\eta_p(f(x))\} = af(\mathbf{x})$ where $a = \chi^{-1}$. The Gaussian component η_g has constant variance equal to b. As a consequence, the total variance of the expression 2.7 can be expressed as

$$\sigma^2(f(\mathbf{x})) = af(\mathbf{x}) + b \tag{2.13}$$

Poissonian distribution can be approximated by normal distribution to a sufficient degree of accuracy by

$$P(\lambda) \approx N(\lambda, \lambda) \tag{2.14}$$

which combined with 2.12 and 2.13 yields

$$\eta_p(f(\mathbf{x})) + \eta_g(\mathbf{x}) \sim N(0, af(\mathbf{x}) + b) \tag{2.15}$$

The relationship between the ISO setting of a particular camera and parameters a and b was studied by Petteri M. Ojala in [15] who performed a set of experiments with a Nikon D300 digital camera and discovered a linear relationship between the ISO number and parameter a and a quadratic relationship between the ISO number and parameter b. Moreover, his experiments showed neglegible dependence of these parameters on shutter time and the temperature at which photos are being taken. We adopt Ojala's measurements from table 2.1 and employ polynomial fitting (MATLAB's `polyfit` and `polyval` functions) to obtain the values of a and b for other ISO values.

2.5   Estimation of noise from images

The BM3D denoising algorithm, described in section 2.6 as well as our extension of the Šroubek and Flusser's MCAM algorithm (see section 2.8.1) both require noise standard deviation(s) σ as one of its input parameters. Hence, several

ISO	a	b
200	$4.64700.10^{-5}$	$6.41233.10^{-8}$
400	$9.05967.10^{-5}$	$2.97067.10^{-7}$
800	$1.78800.10^{-4}$	$1.01390.10^{-6}$
1600	$3.57867.10^{-4}$	$3.29467.10^{-6}$
3200	$7.37933.10^{-4}$	$1.18733.10^{-6}$

Table 2.1: Parameters a and b as a function of ISO for the Nikon D300 digital camera.

methods to estimate σ from images are presented. Zero mean Gaussian noise is assumed in all cases. We present the methods in order of increasing complexity.

Immerkaer's method [21], published in 1996, is a simple and fast method that first applies the Laplacian operator on the image

$$N = \begin{pmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{pmatrix} \tag{2.16}$$

and follows with a direct estimation of the noise standard deviation σ using the formula

$$\sigma = \sqrt{\frac{\pi}{2}} \frac{1}{6(W-2)(H-2)} \sum_{\mathbf{x} \in I} |I(\mathbf{x}) \star N| \tag{2.17}$$

where W is the width of the image and H its height. Altough the method is fast, its inherent simplicity causes it to perceive fine details (lines etc.) as noise, especially in highly textured images. An adaptive approach is therefore needed.

Tai and Yang's method [22], published in 1998, is an extension of the previous approach. First, the Sobel operator

$$G_x = I(\mathbf{x}) \star \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

$$G_y = I(\mathbf{x}) \star \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \tag{2.18}$$

$$G = |G_x| + |G_y|$$

is used to detect edges. In the following stage, it is necessary to estimate which regions are consired to be edges and therefore excluded from the computation of σ. This is achieved by thresholding G using threshold value G_{thr}, which is determined adaptively as a percentage $p\%$ of the image area that is to be considered as edge information. G_{thr} is set to a value where the cumulative

histogram of G reaches $p\%$. This way, even if p is fixed, G_{thr} varies depending on the input image. After the exlusion of edge pixels, noise standard deviation is computed using Immerkaer's formula (2.17). Authors fix the p value at 10%.

Wavelet approach [7]. The wavelet approach to σ estimation uses the observation that wavelet transform distributes noise evenly across all scales, whereas image details are emphasized only at specific ones. First, the input image I is subject to discrete wavelet transform to obtain the first level, i.e. the finest detail coefficients $I_{1_{HVD}}$. The detail, horizontal and vertical transform coeffients are then grouped together to form a single array C_{HVD}. Median absolute deviation

$$\sigma = median_i(|C_{HVD}(i) - median_j(C_{HVD}(j))|) \qquad (2.19)$$

is then used to yield the σ estimate[3].

2.6   BM3D denoising algorithm

The BM3D denoising algorithm, short for block matching and 3D filtering, is a novel, highly efficient algorithm for still image denoising, introduced in 2006 by Alessandro Foi et al. in [20]. It is used in this thesis as a reference to verify the efficiency of the blurred-noisy image fusion.

Historically, still image denoising algorithms predominantly work by attenuating noise in a suitable transform domain (FFT or wavelet transformation are amongst the most used). Although transform-based methods are generally effective, they share common disadvantages such as introduction of unwanted artifacts and loss of detail often characteristic of a given approach.

The BM3D algorithm computes pixel values of the original image from image regions similar to the region centered at the estimated pixel. This approach introduces very few artifacts and, to prevent oversmoothing the image, regions are weighed adaptively. The image is processed pixel-by-pixel by means of a sliding window, regions similar to the currently processed window are stacked to form 3D arrays, inducing a high corellation between regions. A 3D decorrelating unitary transform τ_{3D} (such as DFT or DCT) which produces a sparse representation of the true signal in 3D transform domain is then performed, followed by efficient noise attenuation which is done by applying a shrinkage operator (e.g. hard-thresholding or Wiener filtering) on the transform coefficients. Blocks are then are reconstructed by an inverse 3D transform of the filtered coefficients. The

---

[3]In other words, a median of a set of values is calculated first. Then, the median value is subtracted from each value in the set. The absolute value of each subtraction in the set is calculated and finally, median of the absolute values is computed yielding the median absolute deviation.

final estimate is the weighted average of all overlapping local block-estimates. Because of overcompleteness which is due to overlapping, blocking artifacts are avoided and estimation ability is further improved.

Since the BM3D algorithm is not the focus of this thesis and has only been reused, we only present a brief description of the algorithm; more details can be found in [20]. It is important to note that the algorithm assumes noise standard deviation σ to be known a-priori; section 2.5 elaborates on methods of estimating it from images.

1. *Block matching.* For each pixel x and its corresponding block Z_x of size $N \times N$ (the pixel being located in the upper left block corner), a set S_x of similar image blocks is found. Block similarity is inversely proportional to block distance, which in turn is directly proportional to the L^2 norm of the difference between thresholded transformation τ_{2D} coefficients of a pair of evaluated blocks.

2. *Denoising in the 3D transformation domain.* Each set of blocks matching the reference block Z_x is stacked into a 3D array $\mathbf{S}_x$ (ordered by increasing block distance), subject to 3D transformation τ_{3D}, thresholding and inverse transformation, yielding the estimated array $\hat{\mathbf{S}}_x$. A weight coefficient $w_{\hat{\mathbf{S}}_x}$ is assigned to each estimate $\hat{\mathbf{S}}_x$; it is inversely proportional to the number of nonzero coefficients after τ_{3D} transformation and thresholding.

3. *Estimate aggregation.* After all reference blocks Z_x have been processed and their estimated block sets $\hat{S}_x$ computed, the final estimate is computed as a weighted average of all local ones.
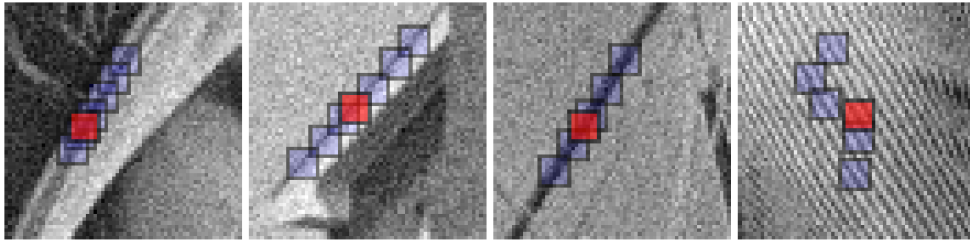

Figure 2.2: The BM3D algorithm. The images show cutaways of *Lena, House, Boats* and *Barbara* images corrupted by zero-mean additive Gaussian noise of $\sigma = 15/255$. A reference block R and a few of its matched ones illustrate block matching. Taken from [20].

2.7 Tico's wavelet method (2009)

Tico presents in [6] a relatively simple wavelet-based approach to blurred and noisy image fusion. Input (blurred and noisy) images are first decomposed into their respective wavelet coefficients by using *undecimated* (also known as *stationary*) discrete wavelet transformation. Then, multi-level coefficient blending is performed. Finally, inverse wavelet transform is performed yielding the result image.

We observe that the absolute difference between the blurred and noisy images is due to presence of noise in the short-exposed image and blurring in the other. We therefore aim for an estimator that emphasizes the short-exposed image where the absolute difference between the two images is larger, and the long-exposed image otherwise. To achieve better separation between the signal and noise, an image estimator is derived in the wavelet domain. The edge locations (i.e., large values in the difference signal), are emphasized at some scales whereas the noise variance is evenly distributed across the scale space. Considering an orthonormal wavelet transform of the two images, denoting by $G_i(k)$, the k-th wavelet coefficient and assuming the same overall brightness of both images[4], we have

$$
\begin{aligned}
G_1(k) &= F(k) &&+ N_1(k) \\
G_2(k) &= F_b(k) &&+ N_2(k)
\end{aligned}
\tag{2.20}
$$

Where F_b denotes the blurred image as a whole since the nature of blurring is not important in this case. Using the observation $\sigma_1^2 \gg \sigma_2^2$ we neglect the noisy coefficients $N_1(k)$ and the term $N_2(k)$ becomes $N(k)$.

We can now fuse the images together using different weights at different scales. Taking advantage of the de-correlation in the wavelet domain, we propose a minimum mean square error diagonal estimator of the original image in the form of a linear combination between the wavelet coefficients of the two images

$$
\widehat{F}(k) = G_1(k) + W(k)D(k)
\tag{2.21}
$$

where $\widehat{F}(k)$ stands for the wavelet coefficients of the restored image, $D(k) = G_2(k) - G_1(k)$ denotes the difference signal between the wavelet coefficients of the two observed images, and $W(k)$ are weight coefficients. We can estimate the best weight $W(k)$ for each wavelet coefficient by minimizing the mean squared error

$$
E[\|\widehat{F}(k) - F(k)\|]_2^2 = E[\|G_1(k) - F(k) + W(k)D(k)\|]_2^2
\tag{2.22}
$$

whose derivative with respect to $W(k)$ equated with zero yields

$$
W(k)E[\|D(k)\|^2] = E[D(k)N(k)]
\tag{2.23}
$$

---

[4]The same overall brightness for both long and short exposure images can in practice be achieved by proper experimental setup

The computation of the weight $W(k)$ requires an estimate of the noise variance in the short-exposed image, and an estimate of the term $E[\|D(k)\|^2]$. In order to estimate noise variance in the short-exposed image the approach presented by Mallat in [7] where noise variance is calculated by using the median absolute deviation M_x[5] of the finest-scale wavelet coefficients as $\sigma \approx M_x/0.6745$ is used. Given the fact that in practice noise is spatially variant over the image, we apply the wavelet-based noise estimate in the pixel neighborhood (e.g. $7x7$). Finally, the $E[\|D(k)\|^2]$ is approximated with $max(\sigma^2(k), avg(|D(k)|^2))$ where avg denotes local spatial average and $\sigma^2(k)$ is the noise variance at the spatial location that corresponds to the k-th wavelet coefficient.

As a consequence, the weight $W(k)$ emphasizes the short exposed image in areas of image transitions (edges, for example) whereas the blurred image is emphasized in smooth regions.

2.8   Šroubek and Flusser's method (2003)

Šroubek and Flusser's MCAM algorithm, published in 2003 [16], presents a solution to the SIMO (*single input, multiple output*) multichannel blind deconvolution in the presence of noise, which is assumed to be Gaussian of zero mean and variance σ^2. We extend this algorithm to accomodate noise of different variance per each channel and use it in our 2 channel setting. It has been selected due to its high resilience to noise, no assumptions on the input PSFs and its potential to better accomodate real-life situations, as some amount of blurring typically occurs even in the short-exposed image.

Expressed mathematically, the algorithm assumes the input to be in the form

$$g_i(\mathbf{x}) = f(\mathbf{x}) \star d_i(\mathbf{x}) + n(\mathbf{x}) \tag{2.24}$$

where g_i are observed images, f is the original image, d_i are blur PSFs, n is Gaussian zero-mean noise of variance σ^2, $i \in \{1 \ldots P\}$ is the channel number and $P \subset \mathbb{N}$ is the number of channels. As we can see, this approach is an extension of our model given by formula 2.1, albeit with the assumption of constant noise in each channel.

To achieve its task, the algorithm seeks to minimize the energy functional

$$E(f, d_1, \ldots, d_P) = \frac{1}{2} \sum_{p=1}^{P} \|g_p - f \star d_p\|^2 + \lambda Q(f) + \gamma R(d_1, \ldots, d_P) \tag{2.25}$$

---

[5]Median absolute deviation on an univariate sample of quantitative data is defined as $MAD = median_i(|X_i - median_j(X_j)|)$

with respect to f and $d_1 \ldots d_P$. $Q(f)$ and $R(d_1, \ldots, d_P)$ are regularization terms for the original image and blur PSF estimates and λ and γ are parameters affecting their influence.

The regularization term $Q(f)$ adopts the total-variation form introduced by Rudin et al. in [24]

$$Q_{TV}(f) = \int_\Omega |\nabla f| \tag{2.26}$$

which, compared to other forms of regularization such as Tichonov's $Q_{TV}(f) = \int_\Omega |\nabla f|^2$ that assume f to be smooth and introduce unwanted ringing artifacts, provides better results. The regularization term $R(d_1, \ldots, d_P)$ takes the form

$$R(d_1, \ldots, d_P) = \sum_{1 \leq i < j \leq P} ||g_i \star d_j - g_j \star d_i||^2 \tag{2.27}$$

Making use of the vector-matrix notation, the P-channel acquisition model can be expressed as

$$\mathbf{g} = \mathcal{D}\mathbf{f} + \mathbf{n} = \mathcal{F}\mathbf{d} + \mathbf{n} \tag{2.28}$$

where $\mathbf{f} = [\mathbf{f_1}^T, \ldots, \mathbf{f_P}^T]^T$ and $\mathbf{d} = [\mathbf{d_1}^T, \ldots, \mathbf{d_P}^T]^T$ are aggregate vector forms of f and d. The matrices $\mathcal{D}$ and $\mathcal{F}$ are defined as

$$\mathcal{F} \equiv \begin{pmatrix} \mathbf{F} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{F} \end{pmatrix} \text{ and } \mathcal{D} \equiv \begin{pmatrix} \mathbf{D_1} \\ \vdots \\ \mathbf{D_P} \end{pmatrix} \tag{2.29}$$

where $\mathbf{F}$ and $\mathbf{D_i}$ are vector-matrix forms of f and d_i. The modified energy functional then assumes the form

$$E(\mathbf{d}, \mathbf{f}) = \frac{1}{2}||\mathcal{D}\mathbf{f} - \mathbf{g}||^2 + \lambda \mathbf{f}^T \mathcal{L} \mathbf{f} + \gamma ||\mathcal{Z}\mathbf{d}||^2 \tag{2.30}$$

Here, the use of $\mathcal{L}$ matrix achieves the computation of total variance from equation 2.26 in discrete form. The matrix $\mathcal{Z}$ achieves the same goal for the computation of regularization term in equation 2.27. Expanding the terms $\frac{1}{2}||\mathcal{D}\mathbf{f} - \mathbf{g}||^2$ (which is equivalent to $\frac{1}{2}||\mathcal{F}\mathbf{d} - \mathbf{g}||^2$) and $||\mathcal{Z}\mathbf{d}||^2$ yields the energy functional in the form

$$E(\mathbf{d}, \mathbf{f}) = \frac{1}{2}[\mathbf{d}^T \mathcal{F}^T \mathcal{F} \mathbf{d} - 2\mathbf{g}^T \mathcal{F} \mathbf{d} + \mathbf{g}^T \mathbf{g}] + \lambda \mathbf{f}^T \mathcal{L} \mathbf{f} + \gamma \mathbf{d}^T \mathcal{Z}^T \mathcal{Z} \mathbf{d} \tag{2.31}$$

To obtain solutions for the minimal $\mathbf{d}$ and $\mathbf{f}$, we solve the system of differential equations

$$\begin{aligned} \frac{\partial E(\mathbf{d}, \mathbf{f})}{\partial \mathbf{d}} &= 2\mathcal{F}^T \mathcal{F} \mathbf{d} & - 2\mathcal{F}^T \mathbf{g} + 2\gamma \mathcal{Z}^T \mathcal{Z} \mathbf{d} &= 0 \\ \frac{\partial E(\mathbf{d}, \mathbf{f})}{\partial \mathbf{f}} &= 2\mathcal{D}^T \mathcal{D} \mathbf{f} & - 2\mathcal{D}^T \mathbf{g} + 2\lambda \mathcal{L} \mathbf{f} &= 0 \end{aligned} \tag{2.32}$$

from which we obtain the expressions for the solutions of $\mathbf{d}$ and $\mathbf{f}$

$$\begin{aligned}
(\mathcal{F}^T\mathcal{F} \quad + \gamma\mathcal{Z}^T\mathcal{Z})\mathbf{d} &= \mathcal{F}^T\mathbf{g} \\
(\mathcal{D}^T\mathcal{D} \qquad + \lambda\mathcal{L})\mathbf{f} &= \mathcal{D}^T\mathbf{g}
\end{aligned} \tag{2.33}$$

and finally we obtain the values of $\mathbf{f}$ and $\mathbf{d}$ through the alternating minimization algorithm, which consists of the following steps.

Require: Initial value $\mathbf{f}^0$, blur size (m_d, n_d) and regularization parameters γ, λ

> **for** $n \geq 1$ **do**
> $\mathbf{d}^n \leftarrow$ solve $[(\mathcal{F}^{n-1})^T\mathcal{F}^{n-1} + \gamma\mathcal{Z}^T\mathcal{Z}]\mathbf{d}^n = (\mathcal{F}^{n-1})^T\mathbf{g}$
>   set $\mathbf{h}^0 = \mathbf{f}^{n-1}$ and $\mathcal{L}^0 = \phi(\mathbf{f}^{n-1})$
>   **for** $k \geq 1$ **do**
> $\mathbf{f}^k \leftarrow$ solve $[(\mathcal{D}^n)^T\mathcal{D}^n + \lambda\mathcal{L}^{k-1}]\mathbf{f}^k = (\mathcal{D}^n)^T\mathbf{g}$
> $\mathcal{L}^k = \phi(\mathbf{h}^k)$
>   **end for**
> $\mathbf{f}^n \leftarrow \mathbf{h}^k$
> **end for**

The ϕ function in the inner loop k serves as an auxiliary function introduced to solve the minimization of $Q(u)$ as $\partial Q(u)/\partial u$ is a highly nonlinear function which is not defined for $|\nabla u| = 0$.

Parameters γ and λ providing the best results are usually obtained empirically, but can be estimated from equation 2.33 to obtain better starting values. Under the $\mathbb{L}_2$ norm, we obtain

$$\begin{aligned}
\gamma^2 &= \frac{||\mathcal{F}^{\mathcal{T}}(\mathcal{F}\mathbf{d} - \mathbf{g})||^2}{||\mathcal{Z}^T\mathcal{Z}\mathbf{d}||^2} \\
\lambda^2 &= \frac{||\mathcal{D}^{\mathcal{T}}(\mathcal{D}\mathbf{f} - \mathbf{g})||^2}{||\mathcal{L}\mathbf{f}||^2}
\end{aligned} \tag{2.34}$$

Since $F\mathbf{d} - \mathbf{g} = D\mathbf{f} - \mathbf{g}$ is equal to white Gaussian noise $\mathbf{n}$ and $||\mathbf{n}||^2 \approx Pm_fn_f\sigma^2$, it is easy to observe from the properties of vector-matrix notation that $||\mathcal{F}^T(\mathcal{F}\mathbf{d} - \mathbf{g})||^2 \approx Pm_dn_d||\mathbf{f}||^2\sigma^2$. Given the fact that $\mathbf{d}$ stands for the correct PSFs, it must be a linear combination of $\mathcal{Z}^T\mathcal{Z}$ eigenvectors that correspond to a set of minimal eigenvalues, which can be expressed as $||\mathcal{Z}^T\mathcal{Z}\mathbf{d}||^2 = \lambda_1^2||\mathbf{d}||^2$. From the definition of $\mathcal{Z}$ it follows that $\lambda_1 \approx \sigma^2(P-1)m_fn_f$, yielding the final form of approximation

$$|\gamma| = \frac{1}{P-1}\frac{\sqrt{Pm_dn_d}}{||\mathbf{d}||}\frac{||\mathbf{f}||}{m_fn_f}\frac{1}{\sigma} \tag{2.35}$$

The $\mathbb{L}^2$ norms of $\mathbf{f}$ and $\mathbf{d}$ are not known in advance, but the former can be approximated by $\mathbf{g}$ and the latter, given that elements in each PSF must sum up to 1, is bound within $(P/(m_dn_d)^2) \leq ||\mathbf{d}||^2 \leq 1$.

A similar procedure is applied to the estimate of the λ parameter, but only a bottom limit can be obtained. The uncertainty resides in the term $||\mathcal{L}\mathbf{f}||^2$ which cannot be further simplified since it is completely dependent on the local behavior of f. The bottom limit in general is zero; the upper limit can be generously defined as $||\mathcal{L}\mathbf{f}||^2 \leq c^2||\mathbf{f}||^2$. The c constant is dependent on the used regularization term, that is $c = 4$ for the $\mathcal{L}_4$ and $c = 4 + 4(1/\sqrt{2})$ for the $\mathcal{L}_8$ TV. Given the fact that $||\mathcal{D}^T(\mathcal{D}\mathbf{f} - \mathbf{g})||^2 \approx ||\mathbf{d}||^2\sigma^2 m_f n_f$, the bottom limit is expressed as

$$|\lambda| \gtrsim \frac{||\mathbf{d}||\sigma}{c} \frac{\sqrt{m_f n_f}}{||\mathbf{f}||} \tag{2.36}$$

The product of the parameters

$$|\gamma||\lambda| \geq \frac{1}{c(P-1)} \sqrt{\frac{P m_d n_d}{m_f n_f}} \tag{2.37}$$

only depends on the dimension of the problem and establishes a fixed relation between them.

2.8.1 Algorithm extension

We propose and implement an extension to the MC-AM algorithm to allow for variable amount of noise in each channel. We assume each channel $i \in 1 \dots P$ to be corrupted by zero mean white Gaussian noise of variance σ_i^2, in other words

$$g_i(\mathbf{x}) = f(\mathbf{x}) \star d_i(\mathbf{x}) + n_i(\mathbf{x}) \tag{2.38}$$

To achieve this task, we need to balance the contribution of each channel to the overall minimization process. To this end, we need to analyze and assess the influence of variable noise on each of the terms in the energy fuctional described by equation 2.25 and modify it accordingly.

The first term

$$\sum_{p=1}^{P} ||g_p - f \star d_p||^2 \tag{2.39}$$

can be expanded as

$$\sum_{p=1}^{P} ||g_p - f \star d_p||^2 = ||n_1||^2 + ||n_2||^2 + \cdots + ||n_P||^2 \approx m_f n_f \sigma_1^2 + m_f n_f \sigma_2^2 + \cdots + m_f n_f \sigma_P^2 \tag{2.40}$$

from which we can directly observe that each term needs to be multiplied by σ_p^{-2} to balance its contribution, we thus obtain

$$\sum_{p=1}^{P} \frac{||g_p - f \star d_p||^2}{\sigma_p^2} \tag{2.41}$$

as the modification of the first term.

The second, total variation term will not be modified as it is a function of the original image f. Its initial value for the algorithm is provided by g_i as is the case in the unmodified version. It is however important to note that the value of the parameter λ will change as a result of our modification, as will be described below.

The third, blur PSF regularization term $R(d_1 \ldots d_P)$ is a sum of terms $||g_i \star d_j - g_j \star d_i||^2$, where $1 \leq i < j \leq P$. Each term can be expanded as

$$
\begin{aligned}
||g_i \star d_j - g_j \star d_i||^2 &= ||(f \star d_i + n_i) \star d_j - (f \star d_j + n_j) \star d_i||^2 \\
&= ||f \star d_i \star d_j - f \star d_j \star d_i + n_i \star d_j - n_j \star d_i||^2 \tag{2.42} \\
&= ||n_i \star d_j - n_j \star d_i||^2.
\end{aligned}
$$

The term $||n_i \star d_j - n_j \star d_i||^2$ can be simplified further. When we think of convolution, we can visualize it as α-blending of $m_d n_d$ shifted copies of the original image, where $\sum \alpha_i = 1$. Doing the same with noise results also in noise with approximately the same characteristics. We can therefore write

$$||n_i \star d_j - n_j \star d_i||^2 \approx ||n_i - n_j||^2 \tag{2.43}$$

Since terms n_i and n_j represent uncorrelated noise and $|| \ldots ||^2 = (\ldots)^T (\ldots)$, we obtain

$$
\begin{aligned}
||n_i - n_j||^2 &= (n_i - n_j)^T (n_i - n_j) \\
&= n_i^T n_i - n_j^T n_i - n_i^T n_j + n_j^T n_j \\
&= n_i^T n_i + n_j^T n_j \tag{2.44} \\
&\approx m_f n_f (\sigma_i^2 + \sigma_j^2).
\end{aligned}
$$

The regularization term $R(d_1 \ldots d_P)$ thus assumes the form

$$R(d_1, \ldots, d_P) = \sum_{1 \leq i < j \leq P} \frac{1}{\sigma_i^2 + \sigma_j^2} ||g_i \star d_j - g_j \star d_i||^2 \tag{2.45}$$

In discrete form, weights are introduced into the first term by matrix multiplication using a block diagonal matrix Σ that consists of P^2 blocks of size $m_d n_d$.

All blocks contain zeros except for the diagonal ones (Σ_{ii}), which are themselves diagonal matrices containing inverse values of noise standard deviations σ_i^{-1}.

$$\left\| \begin{pmatrix} \Sigma_1^{-1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \Sigma_P^{-1} \end{pmatrix} \left[\begin{pmatrix} \mathbf{F} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{F} \end{pmatrix} \begin{pmatrix} \mathbf{d_1} \\ \cdots \\ \mathbf{d_P} \end{pmatrix} - \begin{pmatrix} \mathbf{g_1} \\ \cdots \\ \mathbf{g_P} \end{pmatrix} \right] \right\|^2 . \tag{2.46}$$

Rather than altering the matrix $\mathcal{Z}$ which is never used directly, we modify the matrix product $\mathcal{Z}^T \mathcal{Z}$. This consists of P^2 square blocks of size $m_d n_d$ and is originally defined as follows:

$$\mathcal{Z}^T \mathcal{Z} = (\mathcal{Z}^T \mathcal{Z})_{ij} = \begin{cases} \mathbf{G}_j^T \mathbf{G}_i, & i \neq j \\ \sum_{k \neq i} \mathbf{G}_k^T \mathbf{G}_k, & i = j \end{cases} \tag{2.47}$$

Our modification introduces channel weights in the form

$$\mathcal{Z}^T \mathcal{Z} = (\mathcal{Z}^T \mathcal{Z})_{ij} = \begin{cases} \frac{1}{\sigma_i^2 + \sigma_j^2} \mathbf{G}_j^T \mathbf{G}_i, & i \neq j \\ \sum_{k \neq i} \frac{1}{\sigma_i^2 + \sigma_k^2} \mathbf{G}_k^T \mathbf{G}_k, & i = j \end{cases} \tag{2.48}$$

It is necessary to determine how our modification will affect the estimation of γ and λ parameters. We can easily observe that the introduction of the Σ weight matrix will modify the numerator in both γ^2 and λ^2 estimation (see equation 2.34) as $||\mathcal{F}^T \Sigma^T \Sigma (\mathcal{F} \mathbf{d} - \mathbf{g})||^2$ and $||\mathcal{D}^T \Sigma^T \Sigma (\mathcal{D} \mathbf{f} - \mathbf{g})||^2$, respectively. Since

$$||\mathcal{F}^T (\mathcal{F} \mathbf{d} - \mathbf{g})||^2 \approx P m_d n_d ||\mathbf{f}||^2 \sigma^2 \tag{2.49}$$

and

$$||\mathcal{D}^T (\mathcal{D} \mathbf{f} - \mathbf{g})||^2 \approx ||\mathbf{d}||^2 \sigma^2 m_f n_f \tag{2.50}$$

and σ_i^{-2} emerge from the $\mathbb{L}_2$ norm as σ_i^{-4}, we obtain the modified numerators in the form

$$||\mathcal{F}^T (\mathcal{F} \mathbf{d} - \mathbf{g})||^2 \approx m_d n_d ||\mathbf{f}||^2 \sum_{i=1}^{P} \frac{1}{\sigma_i^2} \tag{2.51}$$

and

$$||\mathcal{D}^T (\mathcal{D} \mathbf{f} - \mathbf{g})||^2 \approx m_f n_f \sum_{i=1}^{P} \frac{||\mathbf{d}_i||^2}{\sigma_i^2} \tag{2.52}$$

The denominator $||\mathcal{L}\mathbf{f}||^2$ in the λ^2 estimation equation remains unaffected. To evaluate the change in the $||\mathcal{Z}^T \mathcal{Z} \mathbf{d}||^2$ denominator, let us first assume that each of the P channels is corrupted by Gaussian zero-mean noise of the same variance σ^2. This will, according to our modification of the $\mathcal{Z}^T \mathcal{Z}$ matrix in equation 2.46, lead to a matrix in which each element is multiplied by $1/2\sigma^2$. Thus, in this

special case, we obtain the modified $\mathcal{Z}^T\mathcal{Z}$ matrix simply by multiplying the original by $1/2\sigma^2$. Under the L_2 norm, this term will cancel out the σ^2 from the above estimation of eigenvalue λ_1, yielding the final form of the gamma estimate

$$|\gamma| = \frac{2}{P-1}\sqrt{\frac{m_d n_d}{||\mathbf{d}||^2}\frac{||\mathbf{f}||^2}{m_f^2 n_f^2}\sum_{i=1}^{P}\frac{1}{\sigma_i^2}} \tag{2.53}$$

and of the lambda estimate

$$|\lambda| \gtrapprox \frac{1}{c}\sqrt{\frac{m_f n_f}{||\mathbf{f}||^2}\sum_{i=1}^{P}\frac{||\mathbf{d}_i||^2}{\sigma_i^2}} \tag{2.54}$$

We use the theoretically computed values of γ and λ in experiments for both simulated and real-world data.

Since mathematical properties of the original algorithm permit solutions to both point spread function(s) and the original image to be shifted by a constant ξ, we need to modify the algorithm further to prevent the solutions to PSFs from gradually "running away". This was achieved by computing the center of gravity of blur PSFs after each iteration by using the formula

$$M_{\mathbf{d}} = \frac{1}{\sum_{x,y\in\Omega}\mathbf{d}(x,y)}\sum_{x,y\in\Omega}xy\mathbf{d}(x,y) \tag{2.55}$$

and shifting them so that computed centers of gravity match the centers of blur PSF support Ω.

Chapter 3

Results

In this chapter, we present the results of tests that have been performed on both simulated and real-world data. We begin by describing the experimental setup used to obtain the data in section 3.1. This is followed by a separate presentation of results for simulated and real-world data in sections 3.2 and 3.3, respectively. The chapter concludes with an analysis and discussion of our findings in section 3.4.

3.1 Experimental setup

3.1.1 Setup for simulated data

We have used a set of 4 point spread functions (see figure 3.1) and 4 testing images (see figure 3.2). These were selected in order to reflect various environments and conditions in which taking images could take place.
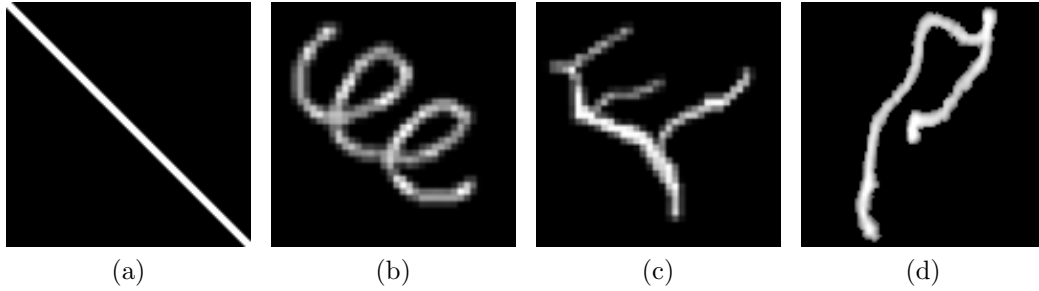

(a)   (b)   (c)   (d)

Figure 3.1: Point-spread functions selected for use in testing. 3.1a) 1 pixel thick diagonal line-shaped blur kernel (the line shown is thicker in order to be visible in print). 3.1b) Blur kernel 7. 3.1c) Blur kernel 8. 3.1d) Blur kernel 10. Blur kernels 3.1b) through 3.1d) were taken from [10].
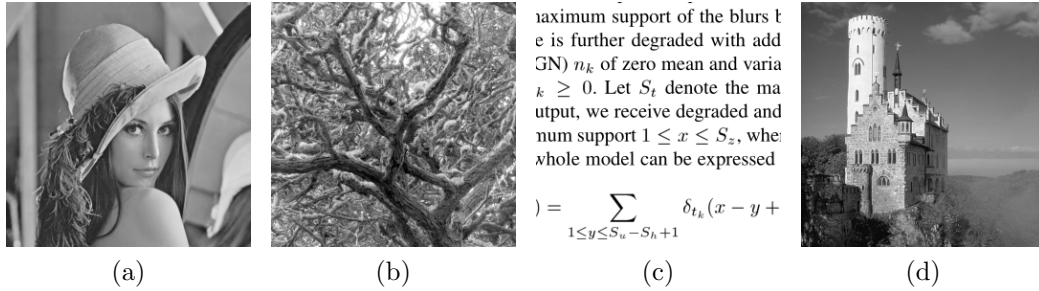

(a)   (b)   (c)   (d)

Figure 3.2: Images selected for use in testing. 3.2a) Lena image, taken from [18]. 3.2b) Tree image, provided by the thesis supervisor. 3.2c) Printed text, a screenshot from one of the cited papers. 3.2d) Castle image. The images are henceforth referred to as Lena, Tree, Text and Castle, respectively.

Testing images used were 256 by 256 pixels in size. We avoided the use of larger resolutions since they tend to be computationally expensive, especially in case of Šroubek's algorithm. The PSF sizes were $2^n - 1$, with n being an integer in range $\{2 \ldots 5\}$. Odd-sized PSFs are preferred due to the fact that they have a well-defined center. The range of used PSF sizes corresponds to results attained in real-world experiments which, although greatly dependent on

the individual photographer's steadiness while taking pictures, did not exceed 31 pixels. This is described in section 3.3. We've performed experiments for both simulated ISOs (the "ISO mode") and additive white Gaussian noise (the "variance mode").

Since Ojala in [15] demonstrated that temperature and exposure time have very little influence on the level of noise present in a digital photograph, we chose to abandon these parameters completely, leaving ISO as the only parameter. ISO 100 was chosen as the base value for both generated and real-world data.

In case of generated data, the pair of ISO values is chosen in advance, one of the values being 100 for the blurred, less noisy image. The other value is set to $100 * 2^i$, where i is an integer greater than 1. This corresponds to ISO settings usually present in digital cameras; however, as of 2011, ISO settings over 25600 are rarely encountered in commercially available products, requiring us to use a value of i no greater than 8. We have, however, also used the i value of 10 (ISO 102400) to generate noise which cannot be simply approximated as a zero-mean Gaussian to observe its effect on the outcome of individual methods.

Fixing the low ISO value at 100, the choice of the high ISO value depends on the level of illumination we wish to emulate. The lower the amount of light in the photographed scene, the higher ISO we need to compensate for the potentially longer exposure time (and thus larger blurring) needed for the camera to capture enough light. For example, given a 2 second exposure time to achieve a proper level of lighting at ISO 100, we might even need to go to ISO 12800 to achieve a potentially blur-less exposure at 1/60 seconds. To achieve approximately the same level of illumination in both short and long exposure images s and l, their ISO and shutter speed settings need to satisfy the equation $ISO_s t_s = ISO_l t_l$. Dynamic ranges of images taken in real-world experiments have confirmed this rule with a sufficient degree of precision.

In total, we've used 4 different images, 4 different blur PSFs at 4 different sizes and 9 different ISO settings, totalling 576 testing scenarios.

For each experiment, the input is a tuple (*input image*, *input PSF*, *input PSF size*, *noisy image ISO*). The ISO value for the blurred image is fixed at 100. We first read the input image f, create two copies, one of which is subject to ISO simulation at the *noisy ISO* value yielding the noisy image g_1, the other is blurred using the *input PSF* resized to *input PSF size* (we denote this intermediate image as f_b) and subject to ISO simulation at ISO 100 to obtain the blurred image g_2. In the next step, noise standard deviation for the noisy image is estimated as the standard deviation of the difference signal between g and f, that is $\hat{\sigma}_1 = std(g - f)$; standard deviation of noise in the blurred image is estimated as standard deviation of the difference signal between g_2 and f_b, or in

other words $\hat{\sigma}_2 = std(g_2 - f_b)$. We then proceed to remove noise from the noisy image using the BM3D algorithm with the estimated noise variance $\hat{\sigma}_1^2$ and to execute both Šroubek's and Tico's algorithms on the blurred - noisy image pair. The noisy and blurred images and the images resulting from the three other methods are written into separate files, as is the initial estimate of the blur PSF and the resulting blur PSFs from Šroubek's algorithm. We then compute signal to noise ratios from all images. A record of each experiment is also written to an output *.csv file. The record features the name of the image and PSF used, the PSF size, the ISO value used for blurred and noisy image, the values of estimated noise standard deviations, the λ and γ parameters used by Šroubek's algorithm and signal-to-noise ratios for the blurred, noisy and other resulting images.

As we've mentioned earlier, we had also performed experiments using the same images and blur PSFs in variance mode, where the images were corrupted by additive white Gaussian noise of variances 10^{-i}, i being in range $\{1 \ldots 4\}$. The base image was corrupted by additive white Gaussian noise of variance 10^{-5}. Again, there were 4 input images, 4 different blur PSFs at 4 different sizes and 4 noise levels, bringing the total to 256 experiments. Similarly to the ISO mode, the input of each experiment is a tuple (*input image*, *input PSF*, *input PSF size*, *noisy image variance*) and noise variance of the blurred (base) image is fixed at 10^{-5}. We do not need to estimate noise standard deviation in any of the images as it is already given. The format of output files and the record written to *.csv file differs only in ISO being replaced by noise variance and the estimates of noise standard deviation not being present at all.

The resulting *.csv report files are then processed to obtain useful data. The images and *.csv records from all experiments are included on the CD in their entirety.

We adopt Šroubek's method to evaluate signal-to-noise ratios of resulting images. Given the original image f and its estimate $\hat{f}$, a linear transform

$$\hat{f}_t(x,y) = \hat{f}(x,y) \frac{1}{N\sigma_{\hat{f}}^2} \sum_{x,y \in \Omega} (\hat{f}(x,y) - \mu_{\hat{f}})(f(x,y) - \mu_f) \qquad (3.1)$$

is first applied on the estimated image $\hat{f}$ yielding the transformed estimate $\hat{f}_t$. We then compute the signal-to-noise ratio using the formula

$$SNR(f,\hat{f}) = 10\log_{10}\sigma_f^2/\sigma_{f\hat{f}_t}^2 \qquad (3.2)$$

where σ_f^2 is the variance of the original image f and $\sigma_{f\hat{f}_t}^2$ is the variance of the difference between the original image f and the transformed estimate $\hat{f}_t$. In the first formula, μ_f and $\mu_{\hat{f}}$ are the means of the original and estimated image,

respectively; $\sigma_{\hat{f}}^2$ is the variance of the estimated image, N is the number of pixels assumed to be the same in both f and $\hat{f}$ and Ω is the image support.

Due to the fact that the output of Šroubek's as well as Tico's algorithms can be shifted compared to the original image (the amount largely depends on the shape of blur PSF) we employ a sliding window in order to compute the best value of the signal-to-noise ratio in a matching position.

3.1.2 Setup for real-world data

Real-world experiments were performed by projecting (using the Acer X110P projector) the Lena image on a vertical white wall 2.6 by 2.6 metres in size and taking images of it with a Canon EOS 5D digital SLR camera using an 85 mm lens at a distance of 5 meters. The projector's optical axis was not perpendicular to the projecting wall; however, a dedicated setting of the projector was used to compensate. The projected image was 70 by 70 centimeters in size and was carefully measured vertically, horizontally and diagonally to rule out any spatial distortion. Sources of light other than the projector itself were blocked out to prevent light pollution and color interference. Control over the level of illumination was achieved by multiplying the image function by a constant; 0.5, 0.25 and 0.125 were the values used (the resulting images were too bright for multiplier 1 - the original Lena image). Projector settings were kept unchanged throughout the experiment. Since the projector's output resolution was 800 by 600 pixels and the used image was 256 by 256 pixels in size, it was possible to insert control features (pixels and crosses) in the otherwise unused part of the projected space (see 3.4). These features were used to align the obtained images and to directly observe PSF shapes.

Three experiments were carried out, one for each multiplier value. In each experiment, several (about 20) images were taken at a progressively greater camera ISO setting until the images produced contained little or no blurring. All images were taken in aperture priority mode with aperture set to the lowest possible value, 1.8. For each experiment, a set of two images was selected; the first taken with ISO set to 100 (the blurred image) and the second taken at the lowest ISO setting that produced little or no blurring (the "noisy" images). These images were processed as follows:

1. Images are converted from 24 bits per pixel color JPEG to 8 bits per pixel PNG.

2. Images are semi-manually registered in Matlab using the `cpselect`, `cp2tform` and `imtransform` routines. The registration process (see figure 3.6) also

Figure 3.3: Experimental setup for real-world testing.

downsamples the images from their original 4368 by 2912 pixels resolution to less than 800 by 600 needed for further processing.

3. Registered images are manually aligned (Paint.NET image editor was used). Figure 3.7 illustrates the result of one such alignment.

4. PSF size for the blurred image is determined by fitting the blur kernel in the smallest possible odd-sized square region with a small margin.

5. A region of the same size as the input image is cut from the blurred and noisy image in their aligned position.

6. Noise standard deviation is estimated from both images as an average value of the results of the 4 available methods (see section 2.5).

7. The noisy image is denoised by the BM3D algorithm using the estimated noise standard deviation value.

8. The noisy and blurred images are subject to Tico's wavelet fusion and Šroubek's MC-AM deconvolution.

9. Resulting images are evaluated in terms of SNR and visual quality.

Figure 3.4: An example of a testing image with control features.

Apart from the estimation of noise using the four methods (see section 2.5) we also estimate noise directly from the original, registered blurred-noisy image pairs for additional verification. Values obtained this way are not used for the computation of Šroubek's or the BM3D algorithm, however. Noise is estimated from three 20 by 100 pixel regions adjacent to the image, and for each region, noise standard deviation is computed as the standard deviation of the region's pixel values after subtracting the region's mean value from them. Resulting values from each region are averaged, yielding the final estimate. The image 3.5 illustrates the regions used.
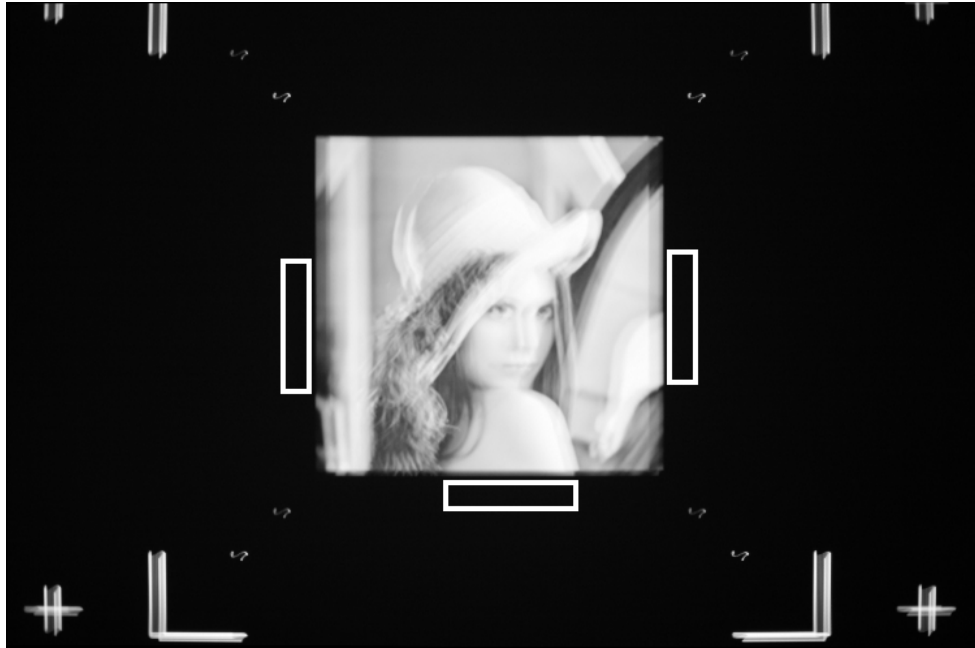
Figure 3.5: Regions used for direct noise estimation from experimental images. Pictured is the blurred image from experiment 3.
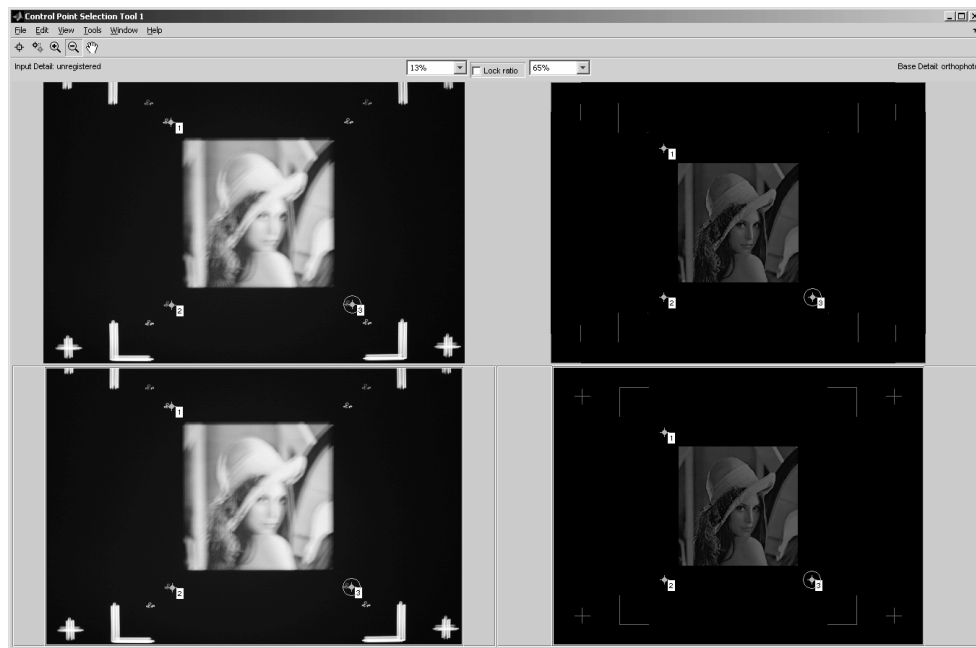

Figure 3.6: Image registration process in Matlab, feature point selection.

Figure 3.7: The result of image alignment for experiment 1.

3.2 Results for simulated data

From the total of 576 experiments performed in ISO mode, our modification of the Šroubek's algorithm provided the best results in terms of SNR in 182 cases. Of those 182 cases, 122 were marked with a difference greater than 1 dB from the result of the BM3D algorithm, which provided the best result in 393 cases (277 of which with a difference greater than 1 dB from the result of Šroubek's algorithm). Tico's algorithm provided the best result in a single case (Lena image, diagonal-line shaped PSF of size 3 for the blurred image and ISO 102400 for the noisy image), which is illustrated by figure 3.8. The SNR of Tico's result was 14.2735 compared to Šroubek's 13.813 and 13.7881 for the BM3D algorithm. A breakdown of individual methods' success rates can be found in table 3.1, which provides the number of successful results for Tico's, Šroubek's and the BM3D algorithm for a given combination of input ISO and PSF size, regardless of the type of image or blurring. The most successful algorithm for a given input combination is emphasized in bold. The same layout is used for variance mode test results, to be found in table 3.2. A detailed overview of the most successful methods for every image, blur kernel, kernel size and input ISO is provided by figures 3.10, 3.11, 3.12 and 3.13. Blue color is reserved for Šroubek's, green for Tico's and red for the BM3D algorithm.

	3	7	15	31
200	0 - 0 - **16**	0 - 0 - **16**	0 - 0 - **16**	0 - 0 - **16**
400	0 - 3 - **13**	0 - 0 - **16**	0 - 0 - **16**	0 - 0 - **16**
800	0 - **8** - **8**	0 - 0 - **16**	0 - 0 - **16**	0 - 0 - **16**
1600	0 - **15** - 1	0 - 0 - **16**	0 - 0 - **16**	0 - 0 - **16**
3200	0 - **15** - 1	0 - 0 - **16**	0 - 0 - **16**	0 - 0 - **16**
6400	0 - **16** - 0	0 - 2 - **14**	0 - 0 - **16**	0 - 0 - **16**
12800	0 - **16** - 0	0 - **10** - 6	0 - 4 - **12**	0 - 0 - **16**
25600	0 - **16** - 0	0 - **15** - 1	0 - **10** - 6	0 - 4 - **12**
102400	1 - **15** - 0	0 - **13** - 3	0 - **12** - 4	0 - **8** - **8**

Table 3.1: A detailed breakdown of tested methods' success rates in ISO mode. For a given combination of input ISO (rows) and input PSF size (columns), the table lists the aggregate number of times Tico's, Šroubek's and the BM3D algorithm achieved the best result, regardless of the input image or input PSF. The most successful algorithm is written in bold.

Of the total of 256 experiments performed in variance mode, Šroubek's algorithm provided the best results in terms of SNR in 126 (88) cases, the BM3D was more successful in 128 (85) cases and Tico's algorithm in 2. The relatively higher success rate of Šroubek's algorithm can be explained by lower performance
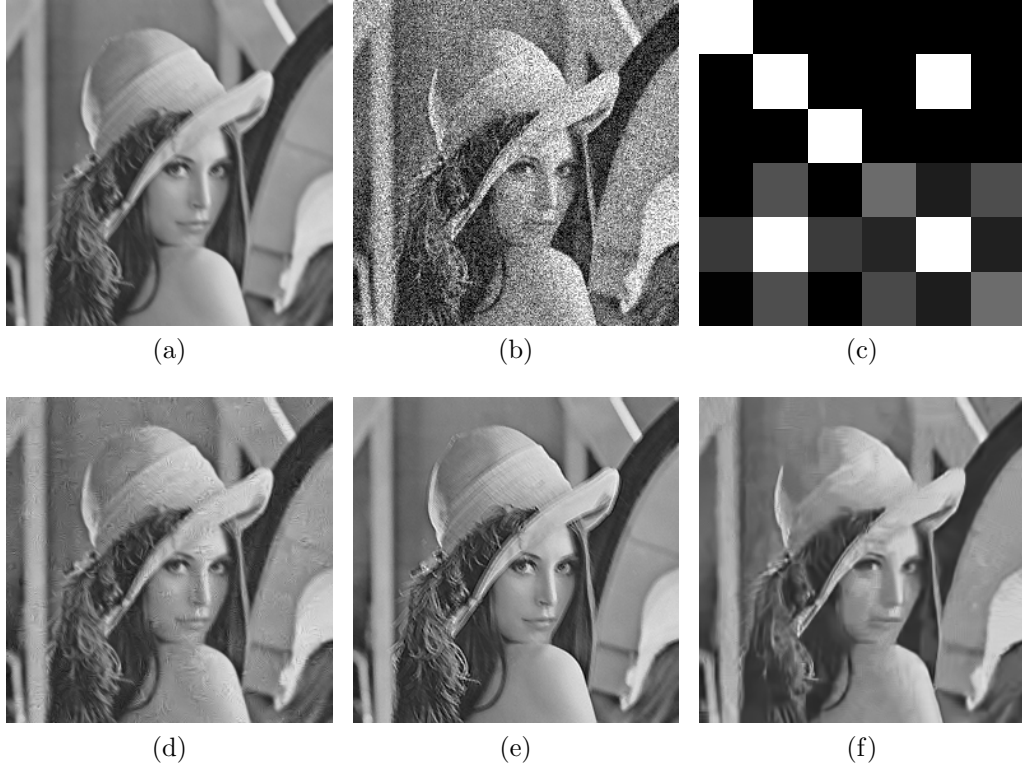
Figure 3.8: The successful result of Tico's algorithm in ISO mode. Figures 3.8a and 3.8b show the blurred and noisy image pair. Figures 3.8d, 3.8e and 3.8f display the results of Tico's, Šroubek's and the BM3D algorithm, respectively. Figure 3.8c shows the enlarged original (top row) and recovered (bottom row) blurred and noisy image PSFs.

of the BM3D algorithm in extremely noisy conditions (variances 0.01 and 0.1) which were not achieved by ISO simulation. In comparison, the simulated ISO of 25600 corresponds to AWGN variance of about 0.003 for all images, the extreme simulated ISO value of 102400, which is only encountered in high end digital SLRs, if at all, corresponds to AWGN[6] variance of about 0.02, which is still an order of magnitude less than the highest value used in variance mode. Similarly to ISO mode, a more detailed breakdown of individual methods' success rates is given in table 3.2. A complete overview of the most successful methods for every image, blur kernel, kernel size and input variance is shown in figures 3.14, 3.15, 3.16 and 3.17. The color scheme used is the same as in ISO mode.

Both of the two marginally successful results of Tico's algorithm in variance mode were achieved in conditions of extreme noise in the noisy image (variance 0.1) and extreme blurring in the other (PSF sizes 15 and 31), one of which

---

[6]Additive white Gaussian noise.

	3	7	15	31
0.0001	0 - **11** - 5	0 - 0 - **16**	0 - 0 - **16**	0 - 0 - **16**
0.001	0 - **15** - 1	0 - 5 - **11**	0 - 1 - **15**	0 - 0 - **16**
0.01	0 - **16** - 0	0 - **13** - 3	0 - **12** - 4	0 - 6 - **10**
0.1	0 - **16** - 0	0 - **12** - 4	1 - **13** - 2	1 - 6 - **9**

Table 3.2: A detailed breakdown of tested methods' success rates in variance mode. For a given combination of input variance (rows) and input PSF size (columns), the table lists the aggregate number of times Tico's, Šroubek's and the BM3D algorithm achieved the best result, regardless of the input image or input PSF. The most successful algorithm is written in bold.

is illustrated by figure 3.9. The SNR of Tico's result was 5.3947 compared to Šroubek's 2.5659 and 5.0676 for the BM3D algorithm.

The rather unexpected single case of Tico's method success for the Lena image, a high ISO of 102400 and a small PSF 3 pixels wide (where Šroubek's algorithm typically provides the best result) can be explained by the shape of the recovered blur PSF which in turn negatively affected the shape of the resulting image, causing it to "jitter" and yield a lower SNR value. The other three experiments that used the same input image, ISO value and PSF size, albeit different PSF shapes, led to Šroubek's algorithm providing the best result.

(a)   (b)   (c)
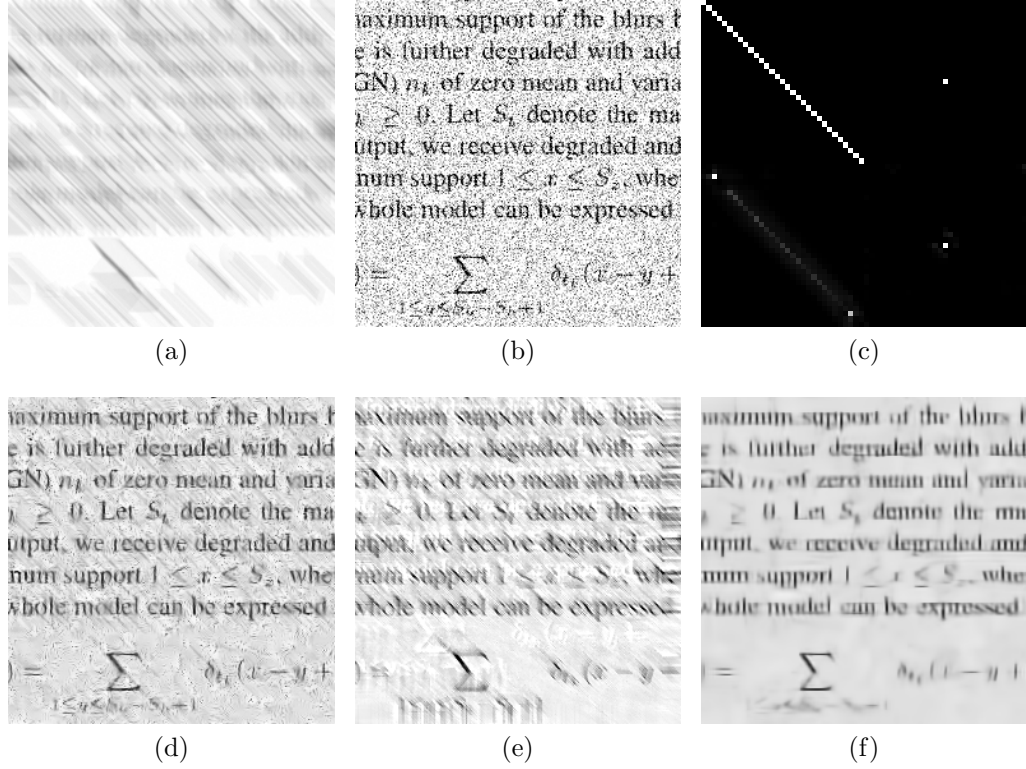
(d)   (e)   (f)

Figure 3.9: The successful result of Tico's algorithm in variance mode. Figures 3.9a and 3.9b show the blurred and noisy image pair. Figures 3.9d, 3.9e and 3.9f display the results of Tico's, Šroubek's and the BM3D algorithm, respectively. Figure 3.9c shows the enlarged original (top row) and recovered (bottom row) blurred and noisy image PSFs.
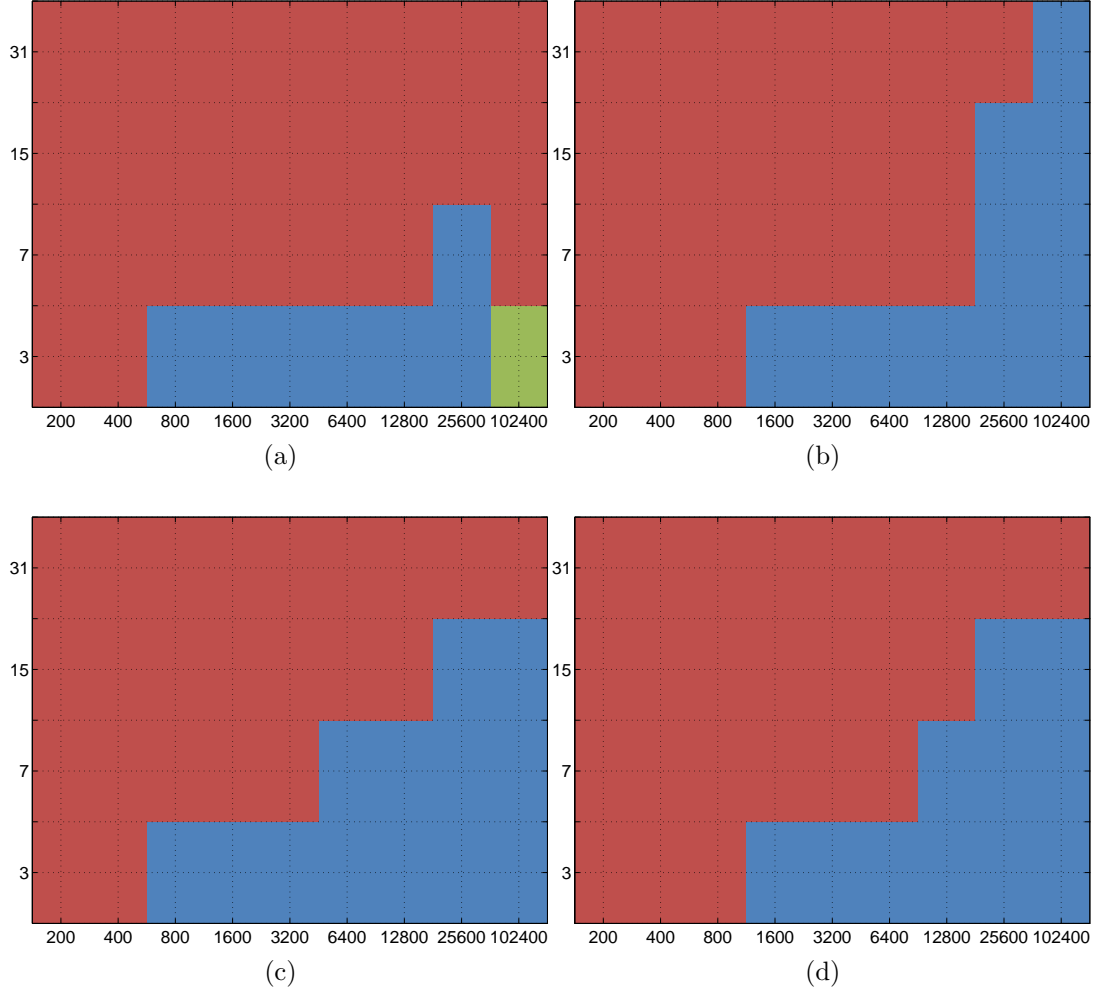
Figure 3.10: Results of the simulated data experiment on the Lena image in ISO mode. For a given combination of input ISO (x axis) and PSF size (y axis), the image shows the method that provided the best result. Red color denotes the BM3D algorithm, green and blue denote Tico's and Šroubek's algorithm, respectively. 3.10a Results for diagonal line-shaped blur kernel. 3.10b Results for blur kernel 7. 3.10c Results for blur kernel 8. 3.10d Results for blur kernel 10.
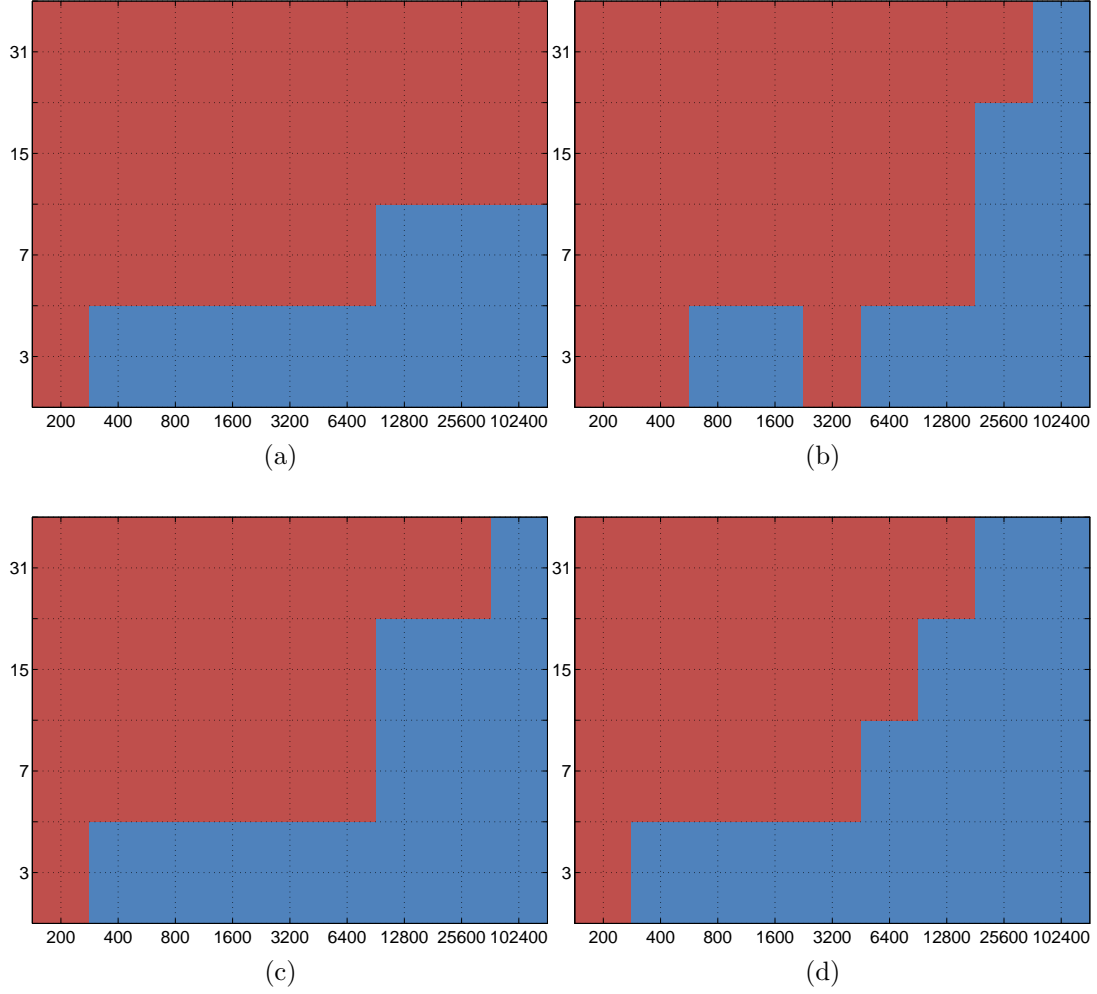
Figure 3.11: Results of the simulated data experiment on the Tree image in ISO mode. For a given combination of input ISO (x axis) and PSF size (y axis), the image shows the method that provided the best result. Red color denotes the BM3D algorithm, green and blue denote Tico's and Šroubek's algorithm, respectively. 3.10a Results for diagonal line-shaped blur kernel. 3.10b Results for blur kernel 7. 3.10c Results for blur kernel 8. 3.10d Results for blur kernel 10.
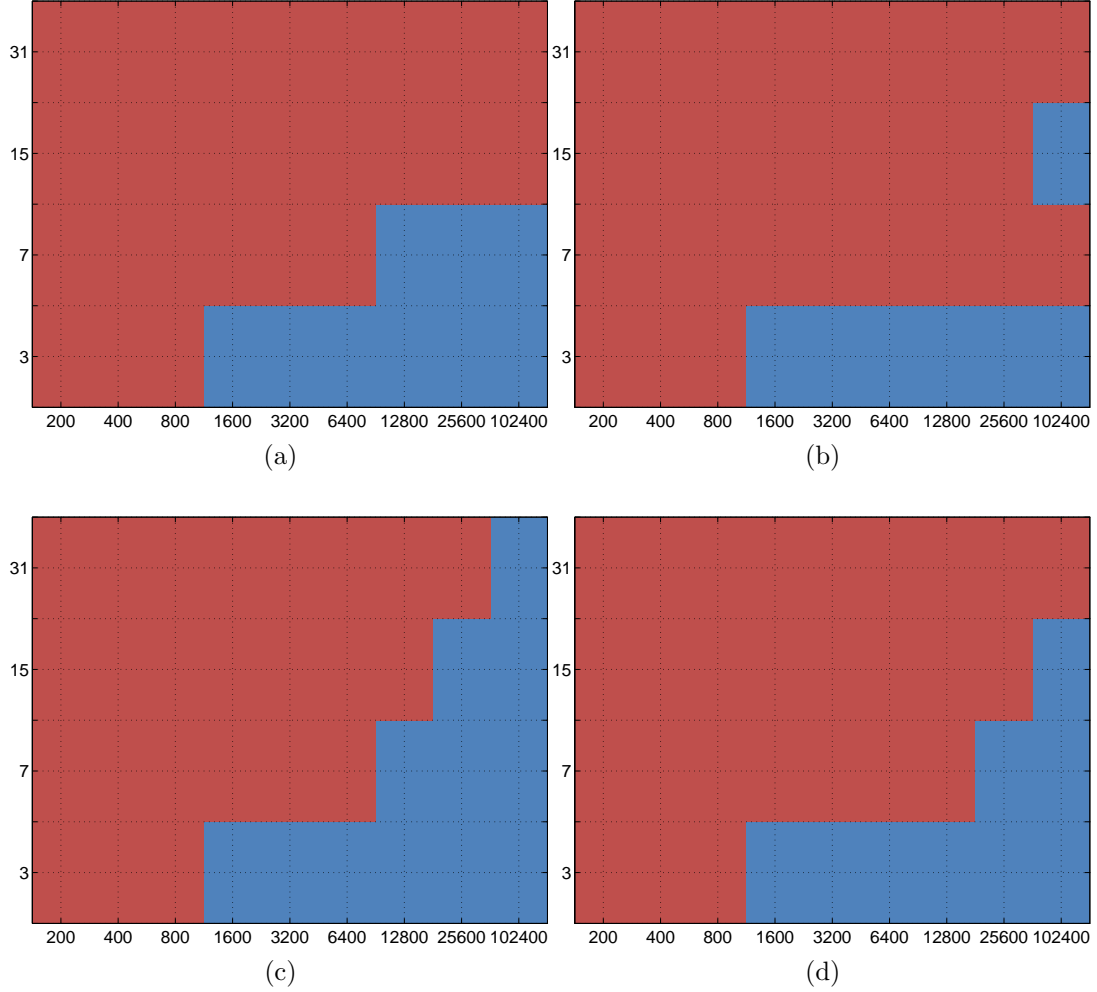
Figure 3.12: Results of the simulated data experiment on the Text image in ISO mode. For a given combination of input ISO (x axis) and PSF size (y axis), the image shows the method that provided the best result. Red color denotes the BM3D algorithm, green and blue denote Tico's and Šroubek's algorithm, respectively. 3.10a Results for diagonal line-shaped blur kernel. 3.10b Results for blur kernel 7. 3.10c Results for blur kernel 8. 3.10d Results for blur kernel 10.
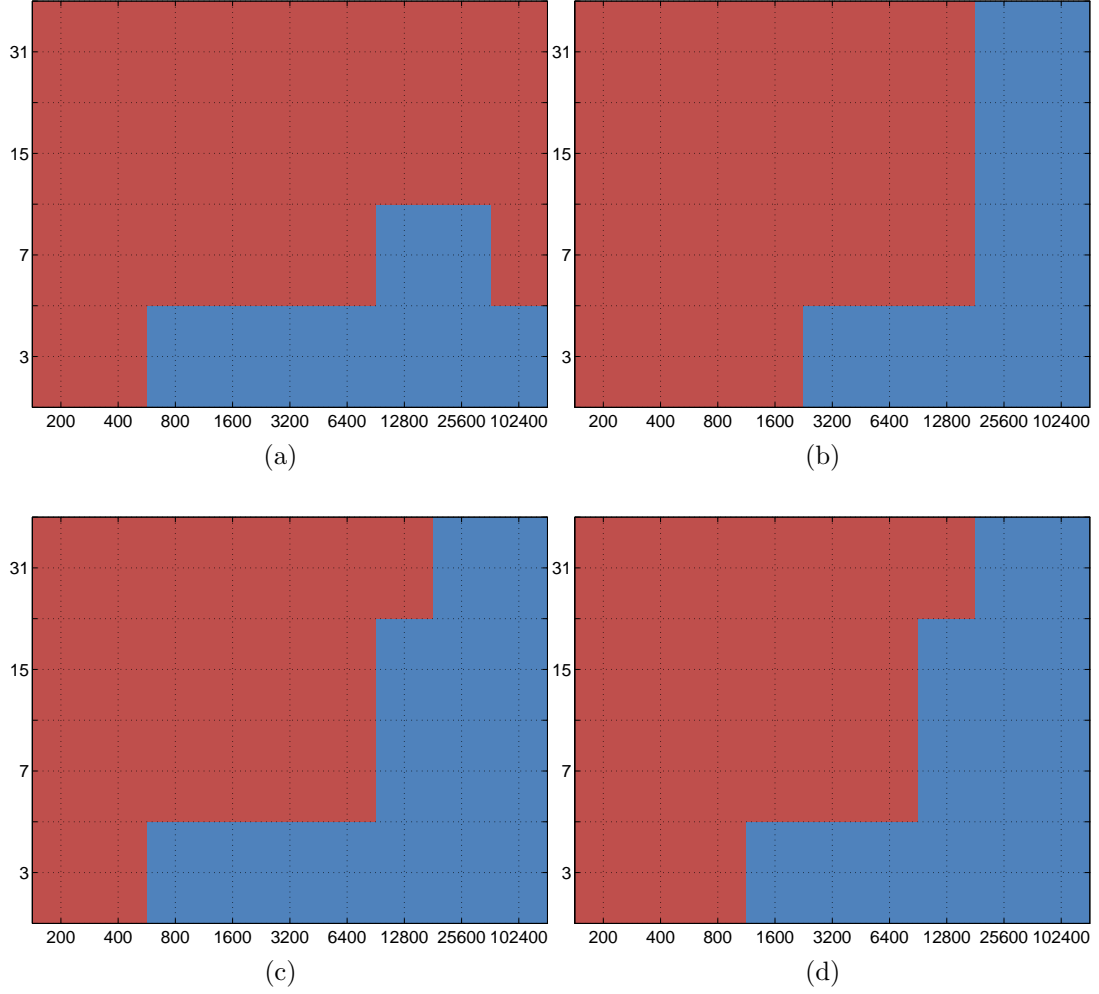
Figure 3.13: Results of the simulated data experiment on the Castle image in ISO mode. For a given combination of input ISO (x axis) and PSF size (y axis), the image shows the method that provided the best result. Red color denotes the BM3D algorithm, green and blue denote Tico's and Šroubek's algorithm, respectively. 3.10a Results for diagonal line-shaped blur kernel. 3.10b Results for blur kernel 7. 3.10c Results for blur kernel 8. 3.10d Results for blur kernel 10.
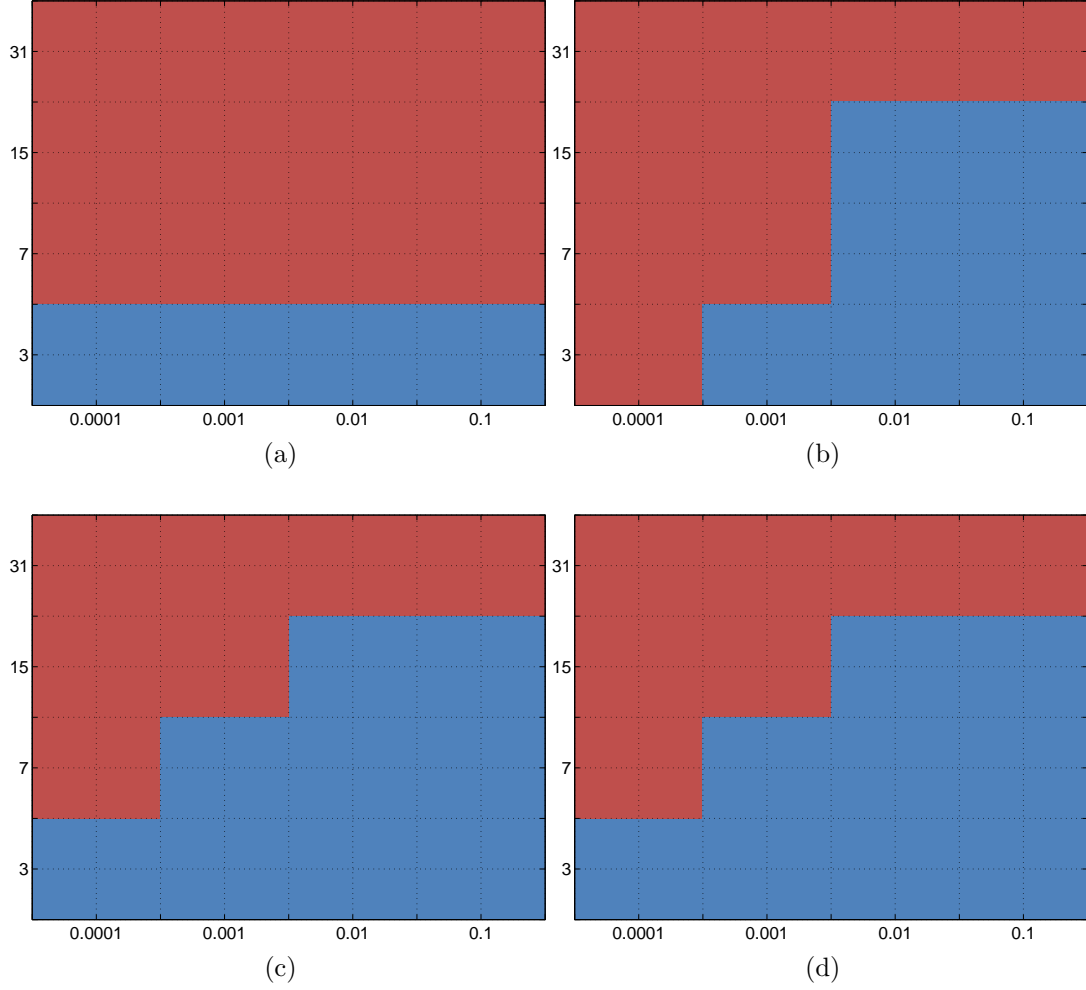
Figure 3.14: Results of the simulated data experiment on the Lena image in variance mode. For a given combination of input variance (x axis) and PSF size (y axis), the image shows the method that provided the best result. Red color denotes the BM3D algorithm, green and blue denote Tico's and Šroubek's algorithm, respectively. 3.10a Results for diagonal line-shaped blur kernel. 3.10b Results for blur kernel 7. 3.10c Results for blur kernel 8. 3.10d Results for blur kernel 10.
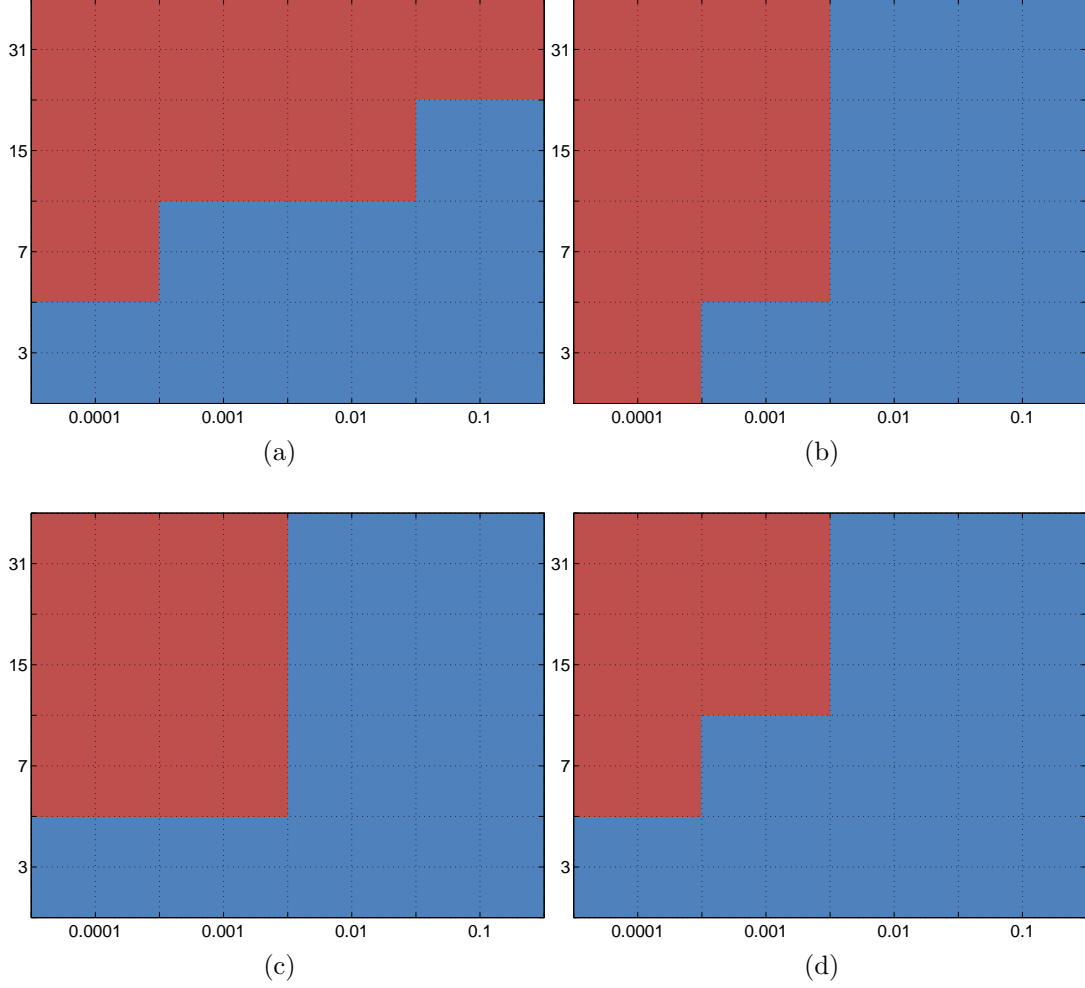
Figure 3.15: Results of the simulated data experiment on the Tree image in variance mode. For a given combination of input variance (x axis) and PSF size (y axis), the image shows the method that provided the best result. Red color denotes the BM3D algorithm, green and blue denote Tico's and Šroubek's algorithm, respectively. 3.10a Results for diagonal line-shaped blur kernel. 3.10b Results for blur kernel 7. 3.10c Results for blur kernel 8. 3.10d Results for blur kernel 10.

Figure 3.16: Results of the simulated data experiment on the Text image in variance mode. For a given combination of input variance (x axis) and PSF size (y axis), the image shows the method that provided the best result. Red color denotes the BM3D algorithm, green and blue denote Tico's and Šroubek's algorithm, respectively. 3.10a Results for diagonal line-shaped blur kernel. 3.10b Results for blur kernel 7. 3.10c Results for blur kernel 8. 3.10d Results for blur kernel 10.
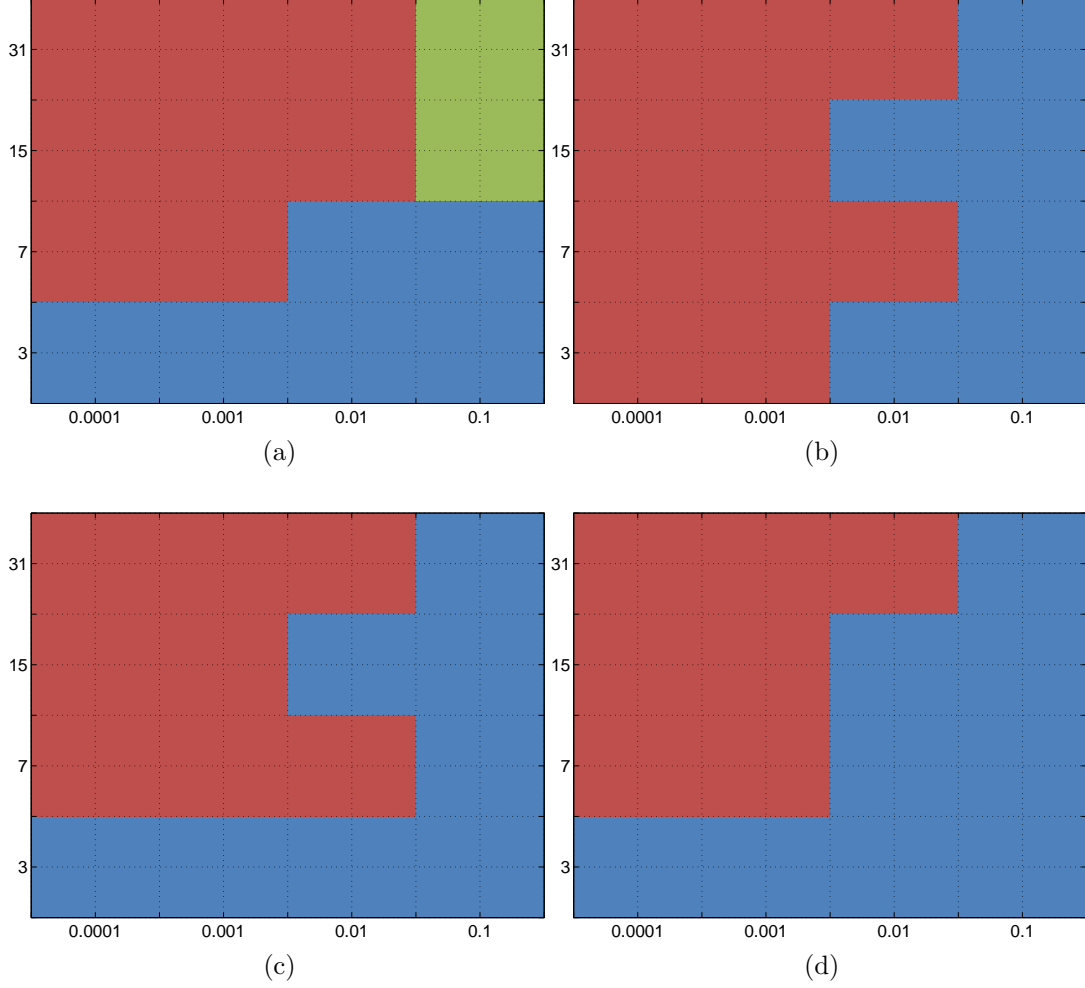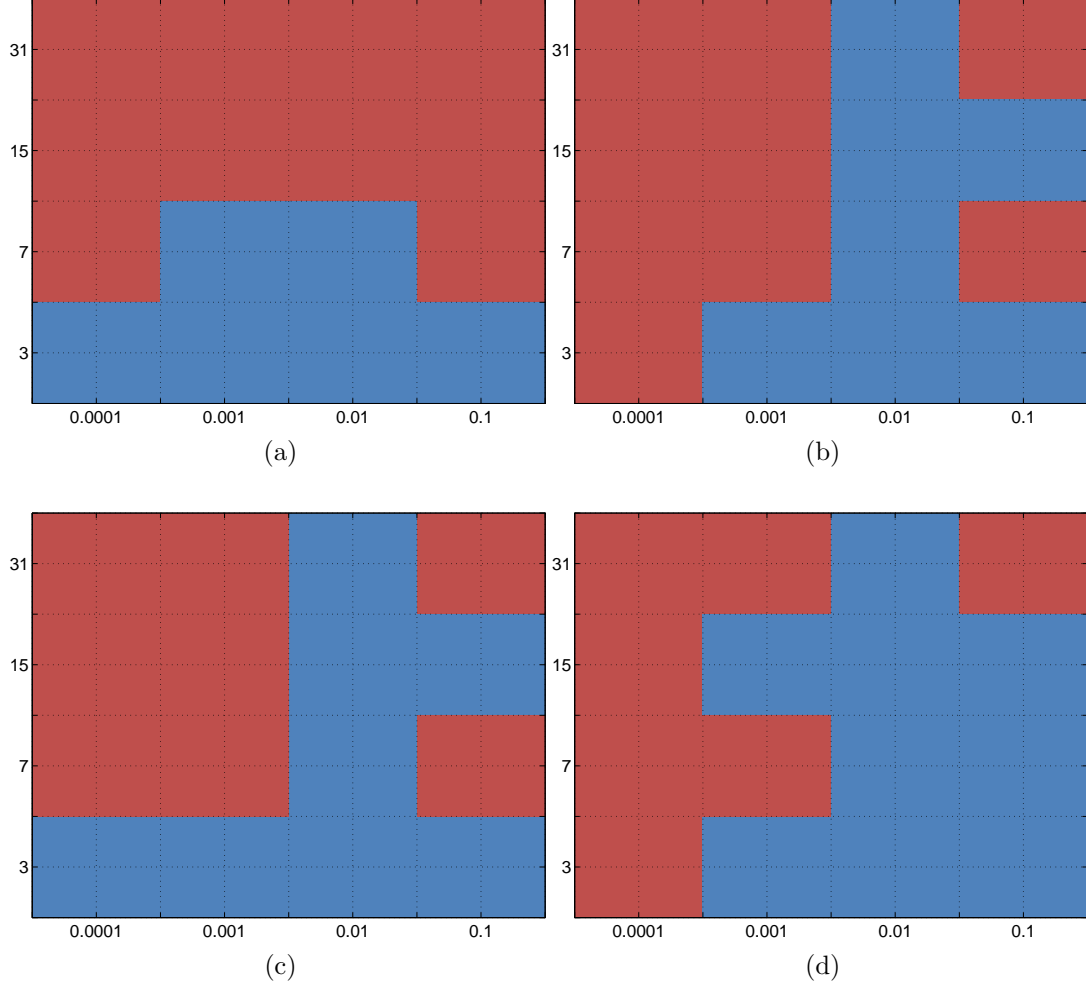
Figure 3.17: Results of the simulated data experiment on the Castle image in variance mode. For a given combination of input variance (x axis) and PSF size (y axis), the image shows the method that provided the best result. Red color denotes the BM3D algorithm, green and blue denote Tico's and Šroubek's algorithm, respectively. 3.10a Results for diagonal line-shaped blur kernel. 3.10b Results for blur kernel 7. 3.10c Results for blur kernel 8. 3.10d Results for blur kernel 10.

3.3 Results for real-world data

As we've mentioned earlier, we'd performed 3 experiments, one for each value of the image multiplier that was used to control the level of illumination for a particular experiment.

Experiment 1, multiplier 0.125.

In this experiment, we simulated extremely low-light conditions, which is evident from the 2.5 second exposure time needed to obtain the blurred image with ISO set to 100. The noisy image was obtained with the ISO setting of 1600 and 1/6 second exposure time. The blurred image's blur kernel size was determined to be 31 by 31 pixels, blurring was also present in the noisy image with blur kernel size 9 by 9 pixels. Setting the ISO to a higher value would have eventually removed blurring in the noisy image completely, we were however limited by the highest possible ISO setting of the camera, which was 1600. The original unmodified blurred and noisy image pair, albeit converted to grayscale, as well as the experiment's result images are shown in figure 3.18. Table 3.3 features the experiment's statistics.

	Original	Blurred	Noisy	Šroubek	Tico	BM3D
Mean	0.4884	0.5751	0.5256	0.5697	0.5256	0.5256
St. dev.	0.1859	0.1466	0.1903	0.2059	0.1890	0.1899
IMM96 st. dev.	0.0093	0.0032	0.0090	0.0043	0.0041	0.0019
TAI98 st. dev.	0.0062	0.0029	0.0079	0.0023	0.0035	0.0013
DWT07 st. dev.	0.0087	0.0058	0.0145	0.0109	0.0083	0.0062
SWT07 st. dev.	0.0087	0.0058	0.0145	0.0109	0.0085	0.0063
Mean st. dev.	0.0082	0.0044	0.0115	0.0071	0.0061	0.0039
Exper. st. dev.	-	0.0058	0.0097	-	-	-
SNR	-	3.4323	11.0442	9.5644	11.5003	11.7639
Lambda	-	-	-	25.0000	-	-
Gamma	-	-	-	0.0005	-	-

Table 3.3: Image statistics for real-world experiment 1. Featured are the experiment's statistics for the original Lena image, the (registered and cropped) blurred and noisy images and the results of Šroubek's, Tico's and the BM3D algorihm, respectively. *Mean* and *st. dev.* are the mean and standard deviation for a particular image, *IMM96 st. dev.*, *TAI98 st. dev.*, *DWT07 st. dev.* and *SWT07 st. dev.* are the results of noise estimation methods described in section 2.5, while *mean st. dev.* is the average of those results. The *exper. st. dev.* is the noise standard deviation determined directly from the uncropped, albeit registered, blurred and noisy images. *SNR* is the signal-to-noise ratio with respect to the original Lena image and *lambda* and *gamma* are the parameters used for the computation of Šroubek's algorithm.

Experiment 2, multiplier 0.25.

In this case, the blurred image was obtained using the ISO setting of 100 and 1 second exposure time. The noisy image was again obtained with the ISO setting of 1600 and 1/15 second exposure time. The blurred image's blur kernel size was determined to be 21 by 21 pixels, blurring in the noisy image was also present, at 5 by 5 pixels. The original blurred and noisy image pair (converted to grayscale) and the resulting images are shown in figure 3.19, their statistics are listed in table 3.4.

	Original	Blurred	Noisy	Šroubek	Tico	BM3D
Mean	0.4884	0.6560	0.6488	0.6665	0.6488	0.6488
St. dev.	0.1859	0.1977	0.2422	0.2393	0.2401	0.2419
IMM96 st. dev.	0.0093	0.0031	0.0095	0.0073	0.0052	0.0042
TAI98 st. dev.	0.0062	0.0026	0.0077	0.0048	0.0041	0.0027
DWT07 st. dev.	0.0087	0.0058	0.0116	0.0095	0.0079	0.0067
SWT07 st. dev.	0.0087	0.0058	0.0116	0.0095	0.0081	0.0067
Mean st. dev.	0.0082	0.0043	0.0101	0.0078	0.0063	0.0051
Exper. st. dev.	-	0.0040	0.0080	-	-	-
SNR	-	4.8812	10.5967	10.4295	10.7121	10.7885
Lambda	-	-	-	50.0000	-	-
Gamma	-	-	-	0.0010	-	-

Table 3.4: Image statistics for real-world experiment 2. Shown are the statistics for the original Lena image, the (registered and cropped) blurred and noisy images and the results of Šroubek's, Tico's and the BM3D algorihm, respectively. *Mean* and *st. dev.* are the mean and standard deviation for a particular image, *IMM96 st. dev.*, *TAI98 st. dev.*, *DWT07 st. dev.* and *SWT07 st. dev.* are the results of noise estimation methods described in section 2.5, while *mean st. dev.* is the average of those results. The *exper. st. dev.* is the noise standard deviation determined directly from the uncropped, albeit registered, blurred and noisy images. *SNR* is the signal-to-noise ratio with respect to the original Lena image and *lambda* and *gamma* are the parameters used for the computation of Šroubek's algorithm.

Experiment 3, multiplier 0.5.

The blurred image was taken using the ISO setting of 100 and 1/3 second exposure time, resulting in a blur kernel 17 by 17 pixels in size. The noisy image, obtained with the ISO setting of 1000 and 1/30 second exposure time, was blurred by a kernel 5 by 5 pixels in size. The original grayscale blurred and noisy image pair along with resulting images are shown in figure 3.20. Experiment statistics are listed in table 3.5.

	Original	Blurred	Noisy	Šroubek	Tico	BM3D
Mean	0.4884	0.7700	0.6879	0.7425	0.6879	0.6879
St. dev.	0.1859	0.1798	0.2304	0.2109	0.2286	0.2302
IMM96 st. dev.	0.0093	0.0030	0.0074	0.0042	0.0049	0.0032
TAI98 st. dev.	0.0062	0.0025	0.0057	0.0021	0.0037	0.0019
DWT07 st. dev.	0.0087	0.0058	0.0116	0.0069	0.0072	0.0056
SWT07 st. dev.	0.0087	0.0058	0.0116	0.0069	0.0074	0.0057
Mean st. dev.	0.0082	0.0043	0.0091	0.0051	0.0058	0.0041
Exper. st. dev.	-	0.0045	0.0055	-	-	-
SNR	-	3.8498	9.1393	8.2007	9.8502	9.8939
Lambda	-	-	-		-	-
Gamma	-	-	-		-	-

Table 3.5: Image statistics for real-world experiment 3. The table features statistics for the original Lena image, the (registered and cropped) blurred and noisy images and the results of Šroubek's, Tico's and the BM3D algorihm, respectively. *Mean* and *st. dev.* are the mean and standard deviation for a particular image, *IMM96 st. dev.*, *TAI98 st. dev.*, *DWT07 st. dev.* and *SWT07 st. dev.* are the results of noise estimation methods described in section 2.5, while *mean st. dev.* is the average of those results. The *exper. st. dev.* is the noise standard deviation determined directly from the uncropped, albeit registered, blurred and noisy images. *SNR* is the signal-to-noise ratio with respect to the original Lena image and *lambda* and *gamma* are the parameters used for the computation of Šroubek's algorithm.

(a)　　　　　　　　　　　　　　(b)

(c)　　　　　　　(d)　　　　　　　(e)

(f)　　　　　　　(g)　　　　　　　(h)

Figure 3.18: A complete visual record of experiment 1. Figures 3.18a and 3.18b show the original, unmodified blurred and noisy image pair. Figures 3.18c and 3.18d show the registered and cropped images which were used as algorithm inputs. Figures 3.18f, 3.18g and 3.18h display the results of Tico's, Šroubek's and the BM3D algorithm, respectively. Figure 3.18e shows the original (top row) and recovered (bottom row) blur PSFs from the blurred and noisy image. The output of Šroubek's algorithm is smaller by the size of the blur PSF diminished by 1, which was 35, overestimated over the PSF size detected from input images, which was 31.

Figure 3.19: A complete visual record of experiment 2. Figures 3.19a and 3.19b show the original, unmodified blurred and noisy image pair. Figures 3.19c and 3.19d show the registered and cropped images which were used as algorithm inputs. Figures 3.19f, 3.19g and 3.19h display the results of Tico's, Šroubek's and the BM3D algorithm, respectively. Figure 3.19e shows the original (top row) and recovered (bottom row) blur PSFs from the blurred and noisy image. The output of Šroubek's algorithm is smaller by the size of the blur PSF diminished by 1, which was 25, overestimated over the PSF size detected from input images, which was 21.
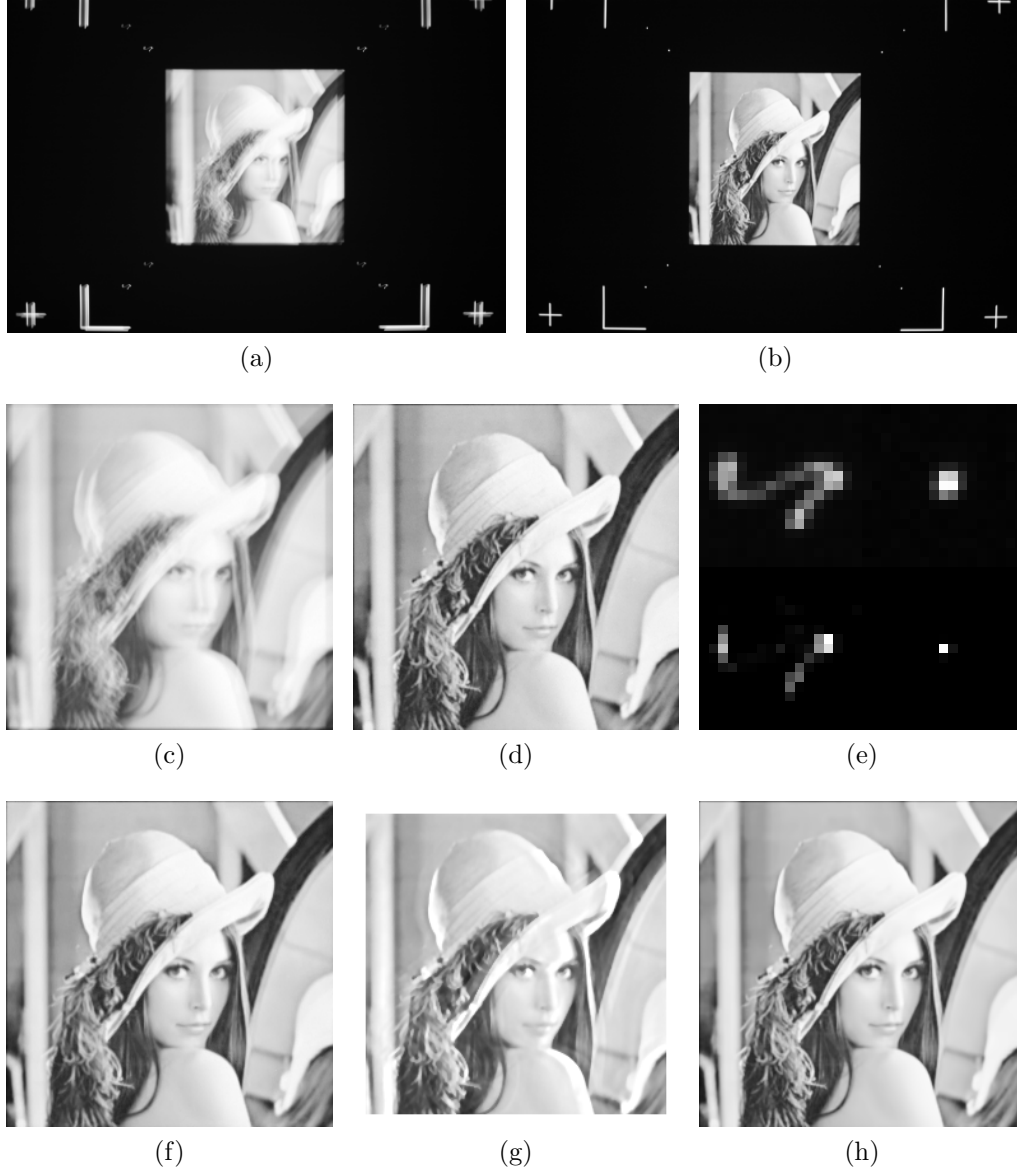
Figure 3.20: A complete visual record of experiment 3. Figures 3.20a and 3.20b show the original, unmodified blurred and noisy image pair. Figures 3.20c and 3.20d show the registered and cropped images which were used as algorithm inputs. Figures 3.20f, 3.20g and 3.20h display the results of Tico's, Šroubek's and the BM3D algorithm, respectively. Figure 3.20e shows the original (top row) and recovered (bottom row) blur PSFs from the blurred and noisy image. The output of Šroubek's algorithm is smaller by the size of the blur PSF diminished by 1, which was 21, overestimated over the PSF size detected from input images, which was 17.

3.4 Evaluation

From simulated experimental data, we observe that on average, the presence of noise has a greater influence on the eventual outcome of a particular method than the extent of blurring does. This is evident from figures 3.21, 3.22, 3.23 and 3.24. Results of the BM3D denoising algorithm are completely independent of the blur PSF size, since the algorithm is executed solely on the noisy, blurless image.
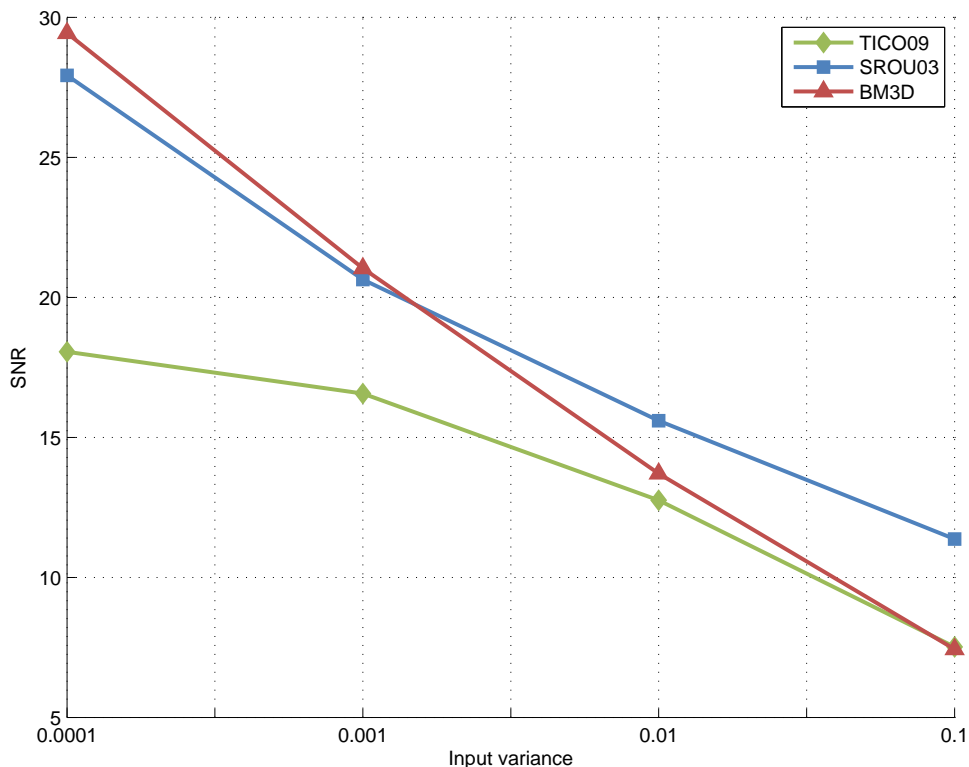

Figure 3.21: Average dependence of tested algorithms on the level of noise.

We've observed that the BM3D denoising algorithm provides consistent results irrespective of the underlying image (also of the type and size of the blur PSF, since blurred images were not denoised); the only exception to this rule are images that exhibit self-similar (fractal) behavior, such the as tree image used in our tests. These features disrupt the block matching process as more matching blocks are found all across the image than is normally the case. The performance of this algorithm is largely independent on the type of noise present and the precision of the estimation of noise standard deviation σ. The algorithm performs very well in case of zero mean-Gaussian additive noise (even for variances as high as 0.01) and also for most ISO values, where the noise roughly
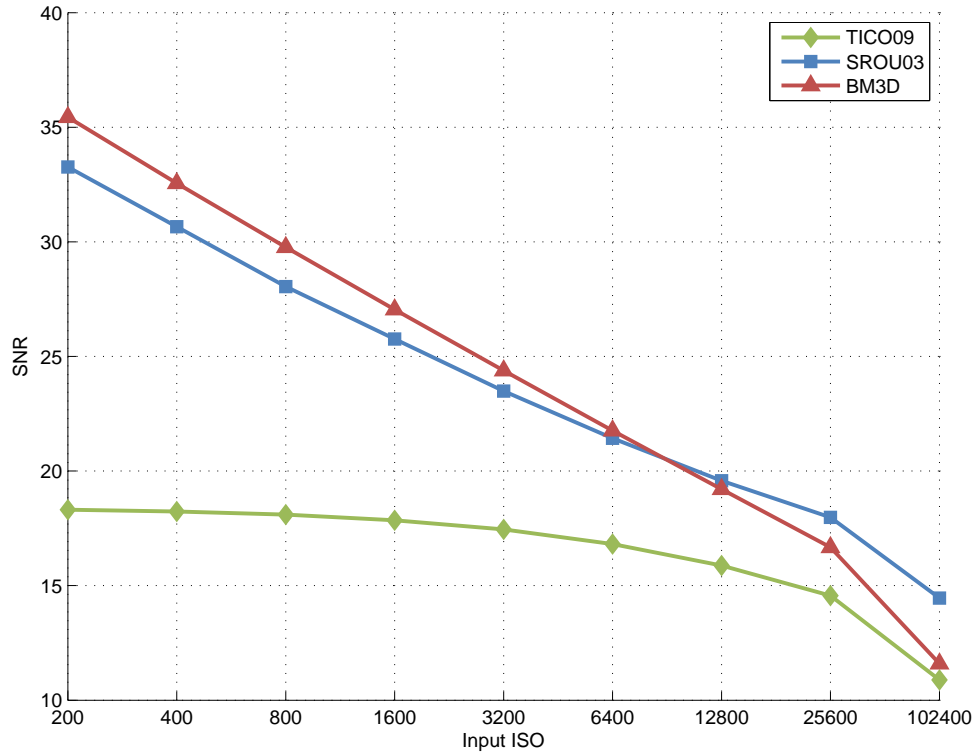
Figure 3.22: Average dependence of tested algorithms on the input ISO.

corresponds to Gaussian. In case of high variance, or non-Gaussian noise, block-matching artifacts become more evident, resulting in reduced performance, as can be seen figure 3.25, which also illustrates the lowered performance on an image containing self-similarity.
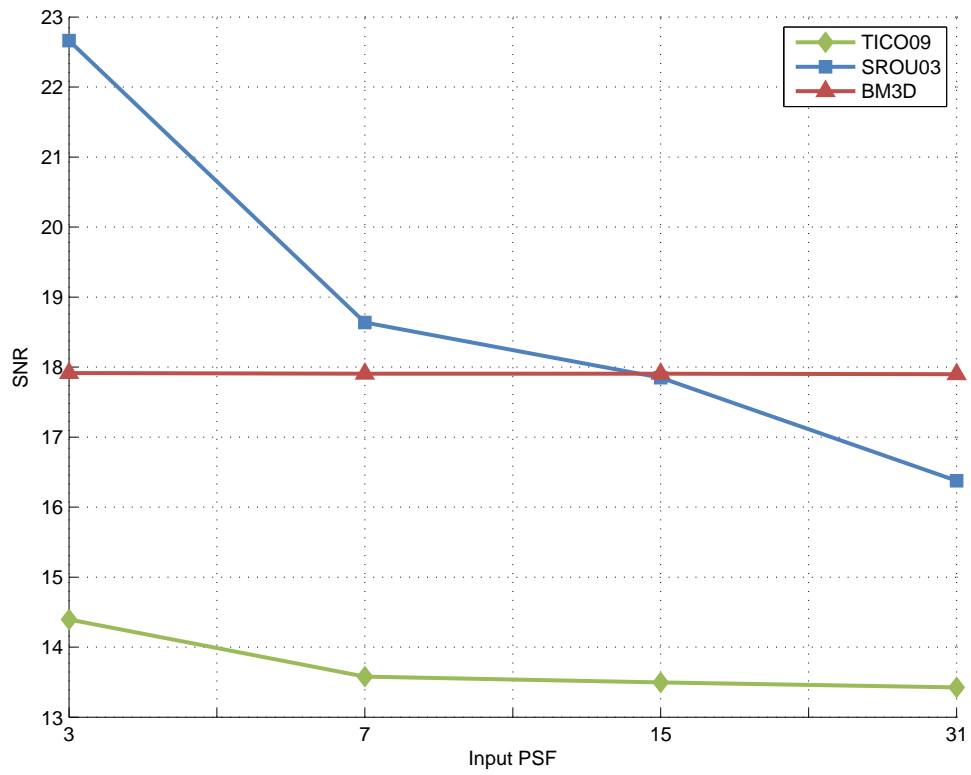
Figure 3.23: Average dependence of tested algorithms on the size of blurring (for variance data).
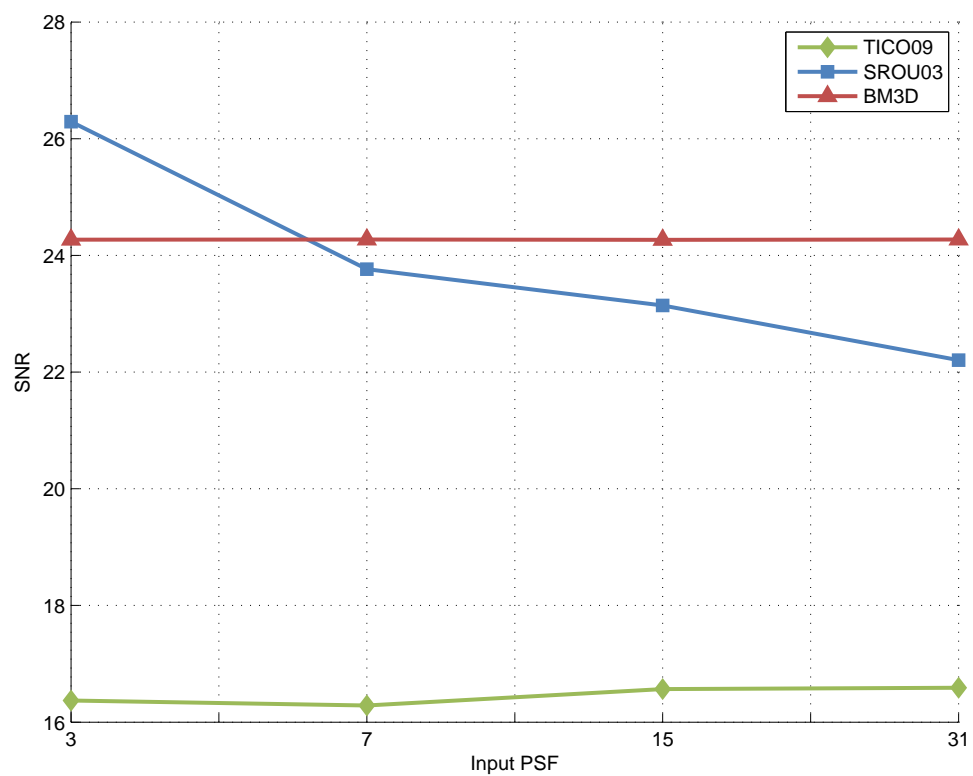
Figure 3.24: Average dependence of tested algorithms on the size of blurring (for ISO data).

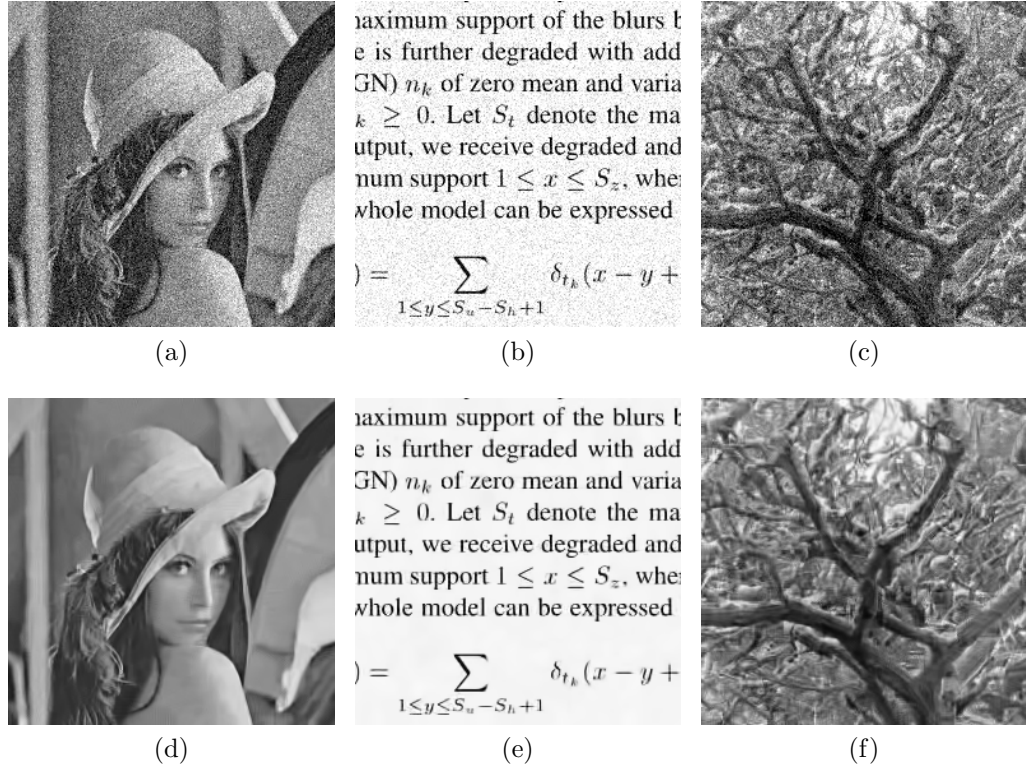|		
(a)	(b)	(c)
(d)	(e)	(f)

Figure 3.25: Failure of the BM3D algorithm. 3.25a, 3.25b and 3.25c are images of Lena, Text and Tree corrupted by additive white Gaussian noise of $\mu = 0$ and $\sigma^2 = 0.01$. Images 3.25d, 3.25e and 3.25f are the respective results of denoising using the BM3D algorithm. While 3.25d and 3.25e prove visually satisfactory, most of the fine detail is lost from image 3.25f. This is due to a relatively larger amount of self-similar regions being present in the Tree image which affect the block-matching process, causing more matching blocks to be found than is normally the case, negatively affecting the outcome. The SNR of the denoised Lena image is 15.8378 versus 15.027 and 9.8666 for the Text and Tree image, respectively.

Although simulated experimental data suggest a certain degree of dependence on the type of input image and input PSF, we observe that Šroubek's algorithm is more successful in the setting of little to moderate blurring in the long-exposed image and high noise in the short exposed image. Here, the blurred image contributes information that the BM3D algorithm simply cannot obtain from a . In rare cases, Šroubek's algorithm outperforms the BM3D algorithm even for highly blurred images. However, a significant disadvantage of Šroubek's algorithm is the need to fine-tune the λ and γ parameters, since the theoretically computed ones do not produce satisfying results, as we've observed in real-world data. There are, however, other parameters to be tweaked (such as the number of iterations, both external and internal) that expand the space that could potentially be searched for the best result.

Although the output of Tico's algorithm is plagued by artifacts at high noise values which lower its potential, the algorithm itself has outperformed either Šroubek's or the BM3D algorithm on only 3 occasions, none of which occured in the real-world setting. This may at first seem statistically insignificant and lead us to quick conclusions; however, the gap between the results of Šroubek's, BM3D and Tico's algorithm is much narrower in real-world experiments, where it even outperforms Šroubek's algorithm. This can be attributed to its internal design which makes it less sensitive to space-variant blurring and statistical variations, which did not occur in simulations yet can never be completely avoided in real-world experiments. These properties would make Tico's algorithm well suited for lower-fidelity equipment, such as camera phones, where high noise levels could pose diffulties for the BM3D algorithm. On the other hand, obtaining a suitable image pair with low fidelity equipment presents yet another obstacle. The better the available equipment is, the lower the noise levels are which again makes the BM3D algorithm a more suitable candidate, especially if the long-exposure image is highly blurred.

Chapter 4

Conclusion

We have studied and described several methods of image deblurring using two images with different exposure times. We have established two categories, the *deconvolution* and *non-deconvolution* methods. We have selected one method from each category, tested them on generated as well as real-world data and compared them to each other and to the results of a state-of-the-art denoising algorithm (BM3D) performed on the noisy image (g_1 from equation 2.1, specifically). We have, given the experimental results, demonstrated that:

- In a majority of cases, using two images with different exposure times provides no additional benefit over single-image denoising

- Two-image deblurring algorithms "defeat" single image denoising in rare cases and in conditions too extreme to be attained by conventional photography

- In low-noise conditions, using a *non-deconvolution* method generally provides better results than using a *deconvolution* method

- All tested methods generally fail on data exhibiting self-similarity and other non-standard behavior rarely present in conventional photographs

This leads us to a general conclusion that an efficient single image denoising algorithm provides better results than using two images with different exposure times.

4.1 Possible future improvements

A particular area for improvement could be the implementation of methods. Due to the fact that this thesis did not focus at computational efficiency and partially due to the fact that MATLAB was used, the algorithms' performance could be significantly improved by re-writing them in a programming language such as C or C++. This would nonetheless be necessary for potential future deployment, especially on devices with limited resources. There is room for improvement and optimization in Matlab source code as well.

For a better overall picture, simulated experiments could be extended to include more image fusion methods. Moreover, Šroubek's algorithm offers potential for more tweaking beyond the number of iterations and the λ and γ parameter, exploring its potential exhaustively could effectively constitute a thesis on its own.

Real-world experiments would benenefit from the inclusion of lower-end imaging devices, such as cellphones; these however often lack even the simplest of controls, such as ISO or exposure time; placing serious restrictions on a reasonable analysis of results. It would also be experimentaly valuable to obtain a digital SLR capable of very high ISO settings, such as 25600, to achieve conditions similar to the simulated ones.

Although real-world experiments were carefully designed and performed, we were unable to completely avoid optical distortion further away from the optical axis. Some degree of inaccuracy was introduced by the analog projector and the interaction between the projected image and the process of taking an image of it. It is however important not to grow too distant from real-world photography, of which these imperfections are a daily part.

Appendix A

Contents of the CD

Structure of the CD corresponds to the structure of the author's working directory and the structure of the online Git repository, where the thesis was maintained.

- **Docs** Miscellaneous thesis-related documents, stored herein is the previous version of the thesis written in Microsoft Word, the Visio diagram illustrating a general overview of the blurred-noisy image pair overview solution as well as an Excel file used for a computation of average parameters for Ojala's noise model

- **Images** Images selected for use in testing, blur kernels and their working versions

- **Input** The source and processed images of the 3 real-world experiments, stored in 3 subdirectories named 001, 002 and 003.

- **Matlab** Matlab source files

- **Output**

- **TeX** The TeX source file along with the included images.

Appendix B

Implementation and testing notes

Methods tested in this thesis have been implemented using Matlab version 7.4.0 running on Microsoft Windows XP Professional SP 2. Apart from standard Matlab functions, the *Wavelet Toobox* and *Image Processing Toolbox* were used.

Tests were performed on two computers, the first being a single-core 1.8 GHz PC running 32-bit Microsoft Windows XP Professional SP 2 with 1792 MB of RAM and the second was a quad-core hyperthreaded laptop PC running 64bit Windows XP 7 Professional with 8192 MB of RAM, on which up to 4 virtual Windows XP Professional PCs configured to have a single processor and 1024 MB of RAM were run in parallel. Virtual Windows XP PCs were required due to the available Matlab version, which is older and incompatible with Windows 7.

The total time needed to generate all BM3D data was approximately 45 minutes, which was due to the fact that the BM3D denoising algorithm was written in C++ and provided in binary form with a MATLAB interface. In case of Tico's wavelet fusion, processing of one blurred - noisy image pair was independent of the blur size and took about 2 minutes, bringing the total time close to 28 hours. The running time of Šroubek's MC-AM algorithm was highly dependent on PSF size and doubling the PSF size approximately quadrupled the running time, which was approximately 20 - 40 (sometimes even an hour) for PSF 31 pixels in size, 5 - 10 minutes for 15, about 2 minutes for 7 and less than a minute for 3, bringing the total time to about 5 and a half days, which was effectively halved due to computations being performed on 2 PCs in parallel. Searching for the best λ and γ parameters for the computation of Šroubek's algorithm in case of real-world data required its complete execution for each pair; since there were approximately 200 λ and γ pairs tested for each

experiment, the total running time was approximately 500 hours, effectively about one quarter of time was needed due to parallelization.

No GUI was implemented and tests were performed in batch mode, results being written to a *.csv file. This is readable in plain text, but is better wieved in a spreadsheet processor, such as Microsoft Excel or Libre Office. Graphs were created in both MATLAB and Microsoft Excel 2007, both of which allow them to be created in the eps format, convertible to pdf.

Image registration was performed directly in MATLAB using control point selection routine `cpselect` which produces a set of corresponding control points, the `cp2tform` routine, which takes the set of control points and computes an image transformation structure from them depending on the desired transformation type and the routine `imtransform`, which performs the transformation itself. Projective transformation was used as the desired transformation to correct for possible perspective deformation that could occur when projecting the image or taking picture of it.

The thesis was developed and maintained using the Git versioning system and placed (with the exception of generated image data) into a public Git repository at `https://github.com/jozefsabo/Thesis` to facilitate accessibility from multiple locations.

Image editing tasks were handled in Paint.NET and IrfanView.

Bibliography

[1] Jia J., Sun J., Tang C.K., Shum H.Y.: *Bayesian correction of image intensity with spatial consideration*, Proceedings of the eighth European Conference on Computer Vision, 256-266, Praha, 2004

[2] Richardson, W. H.: *Bayesian-Based Iterative Method of Image Restoration*, JOSA, Volume 62, 2007

[3] Foi A., Trimeche M., Katkovnik V., Egiazarian K.: *Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data*, IEEE Transactions, 2007

[4] Tico M., Vehvilainen M.: *Estimation of motion blur point spread function from differently exposed images*, Nokia Research Center, Tampere, Finland, 2006

[5] Razligh Q.R., Kehtarnavaz N.: *Image blur reduction for cell-phone cameras via adaptive tonal correction*, Department of Electrical Engineering, University of Texas at Dallas, 2007

[6] Tico M., Pulli K.: *Image enhancement via blur and noisy image fusion*, Nokia Research Center, Palo Alto, CA, USA, 2009

[7] Mallat, S.: *A wavelet tour of signal processing, the sparse way*, Elsevier, Burlington, MA, 2009

[8] Rao, A.R.: *A Taxonomy for Texture Description and Identification*, Springer Verlag, 1990

[9] Liu X., Gamal A.: *Simultaneous image formation and motion blur restoration via multiple capture*, Proceedings of the International Conference on Acoustics, Speech, Signal Processing, 2001

[10] Yuan L., Sun J., Quan L., Shum H.Y.: *Image Deblurring with blurred/noisy image pairs*, Computer Science Department, Hong Kong University of Science and Technology, 2007

[11] Tico M., Vehvilainen M.: *Image stabilization based on fusing the visual information in differently exposed images*, Nokia Research Center, Tampere, Finland, 2007

[12] Beneš B., Felkel P., Sochor J., Žára J.: *Moderní počítačová grafika, 2. vydání*, Computer Press, 2004

[13] Biedl T., Chan T., Demaine E., Fleischer R., Golin M., King J., IanMun J.: *Fun Sort - or the Chaos of Unordered Binary Search*, Discrete Applied Mathematics, vol. 144, p. 231-236, 2004

[14] Gooch M., Reinhard E., Ashikmin E., Shirley P.: *Color transfer between images*, IEEE Computer Graphics and Applications, 34-40, 2001

[15] Petteri, O. M.: *Dependence of the parameters of digital image noise model on ISO number, temperature and shutter time*, Tampere University of Technology, Department of Automation Science and Engineering, Tampere, Finland, 2008

[16] Šroubek F., Flusser J.: *Multichannel blind iterative image restoration*, Proceedings of IEEE Transactions on Image Processing, vol. 12, no. 9, September 2003

[17] Šroubek F., Flusser J.: *Multichannel blind deconvolution of spatially misaligned images*, Proceedings of IEEE Transactions on Image Processing, vol. 14, no. 7, July 2005

[18] *The USC-SIPI Image Database*
http://sipi.usc.edu/database/

[19] *Wikipedia*, miscellaneous articles.
http://en.wikipedia.org/

[20] Dabov K., Foi A., Katkovnik V., Egiazarian K.: *Image denoising with block-matching and 3D filtering*, Institute of Signal Processing, Tampere University of Technology, PO BOX 553, 33101 Tampere, Finland, 2006

[21] Immerkær J.: *Fast noise variance estimation*, Computer Vision and Image Understanding, Vol. 64, No. 2, 300-302, 1996

[22] Tai S. Ch., Yang S. M.: *A fast method for image noise estimation using laplacian operator and adaptive edge detection*, Proceedings of the second International Symposium on Communications, Control and Signal Processing, 1077-1081, 2008.

[23] Lim S. H., Silverstein A.: *Estimation and removal of motion blur by capturing two images with different exposures*, Hewlett-Packard Laboratories, 2008.

[24] Rudin L., Osher S., Fatemi E.: *Nonlinear total variation based noise removal algorithms*, Phys. D, vol. 60, 259–268, 1992.

[25] Babacan S.D., Wang J., Molina R., Katsaggelos A.K.: *Bayesian blind deconvolution from differently exposed image pairs*, Proceedings of the 2009 IEEE International Conference on Image Processing, 133-136, 2009.

[26] Vekkot S., Shukla P.: *A novel architecture for wavelet based image fusion*, World Academy of Science, Engineering and Technology 57, 373-377, 2009

[27] Portilla J.: *Image restoration through L0 analysis-based sparse optimization in tight frames*, Proceedings of the 2009 IEEE International Conference on Image Processing, 3909-3912, 2009.

[28] Kano H., Hatanaka H., Fukumoto S., Murata H.: *Motion blur estimation of handheld camera using regular- and short-exposure image pair*, Proceedings of the 2009 IEEE International Conference on Image Processing, 1317-1320, 2009.