Technical Report
CVL-TR-6

Automated Inpainting of Unknown Palimpsest Regions

Fabian Hollaus and Robert Sablatnig

Computer Vision Lab
Institute of Computer Aided Automation
Vienna University of Technology

September 19, 2011

Abstract

This work deals with the automated reconstruction of modern English handwritings and ancient Greek texts. The Greek writings were overwritten in the past and the presented system utilizes multi-spectral images, in order to identify the overwritten regions and to reconstruct them. The detection of the overwritten regions is fulfilled by a binarization algorithm, which considers the differences between intensity minima and maxima in a local neighborhood. The differences found are afterwards used for the determination of local thresholds, which are utilized in the foreground segmentation step.

The foreground regions found are subsequently filled by a statistic inpainting approach. The inpainting technique chosen is part of the Fields of Experts (FoE) framework. This approach is based on Markov Random Field modeling and allows for a learning of statistical image models, which model long range dependencies between pixels. The models are defined by several parameters, which are all learned from a training database. Such a trained model is used in an inference task in order to maximize the probability of a restored image. FoE image models are used in this work in order to recover the occluded parts of ancient Greek texts. Additionally, the FoE algorithm is applied on natural images and modern handwritings.

A further inpainting technique has been developed for the reconstruction of modern writings: Contrary to the statistical approach, the second algorithm is based on a heuristic one that is called tensor voting framework. This approach is based on the Gestalt principles and allows for a reconstruction of missing or damaged image data. The technique is applied on handwritings in order to restore missing curve points, which belong to character boundaries.

The recovery of modern writings, which were partially overlapped by ruling lines, was analyzed in one experiment. The heuristical approach gained an average PSNR value of 15.58 dB in this test. The statistical inpainting technique was able to reconstruct more missing text regions and hence it yielded a higher PSNR value of 18.24 dB. The proposed system gained an average PSNR value of 20.34 dB on the dataset, which was comprised of the Greek writings.

Contents

Chapter 1

Introduction

This thesis deals with digital image inpainting, which denotes the automated recovery of unknown image regions. Inpainting is not defined for a particular group of images and the majority[1] of the inpainting techniques listed below are concerned with the restoration of natural images. While the restoration of such images is also investigated in this work, the main focus lies on the recovery of images containing handwritings. The retouching of recent handwritten text is examined in this thesis, as well as the recovery of ancient documents.

The analysis of such historic writings can be facilitated by multi-spectral imaging [RB05a]. Multi-spectral imaging has also proven its usefulness for the examination of a specific group of documents, named palimpsests [RB05a], [Kno08]. The term palimpsest stems from the Greek word *palimpsestos* meaning scraped again. Palimpsests are historic manuscript pages that were reused, because parchment was a costly material in former times [RB05a]. The original texts in such palimpsest were scraped off and overwritten by younger writings. Hence, the former text is often barely legible under visible light, which makes a text analysis under this lighting condition difficult or even impossible [FK06]. Multi-spectral imaging can be used in order to enhance the visibility of ancient underwritings [RB05a], [EKCB03]. The system, which will be introduced in the following, aims at retouching such underwritings in palimpsests. The proposed method makes use of Multi-Spectral Images (MSI) in order to separate the overwritings from the underwritings and to restore the older texts.

The palimpsests, which are used in this thesis, are generously provided by the Archimedes Palimpsest Project. The Archimedes Palimpsest is a copy of treatises created by the Greek scientist Archimedes (287 - 212 BC). The copy was created in the tenth century in Constantinople, but in the thirteenth century the manuscript was taken apart and reused as a prayer book, also called Euchologion. The original text was erased and overwritten [Kno08].

The Danish scholar Heiberg discovered in 1906 that the Euchologion contained overwritten treatises by Archimedes and transcribed those treatises. Heiberg was not able to transcribe the entire underwritings, because they are barely visible under tungsten illumination. Shortly after Heiberg's finding, the manuscript disappeared in 1908 for yet

[1] All of the inpainting methods, which will be introduced in the following, are designed for natural images, except for: [CG09], [BGS05], [BEG07].

unknown reasons and remained missing till 1998. The book was sold in this year and lent to the Walter Art Museum in Baltimore. Since then, the team from the Archimedes Palimpsest Project put effort into making the ancient texts legible [Ber07].

1.1 Motivation

The project team discovered that the original text is most visible under UltraViolet (UV) illumination. Under this lighting condition the overwriting is unfortunately still visible. If a manuscript leaf is in contrast illuminated with red light, the ancient text almost disappears, while the younger text is still present. Knox [Kno08] used this fact to recover the underwritings, by subtracting the red light images from the UV images. He applied multi-spectral unmixing techniques in order to separate the underwriting from the residual image information. Furthermore, he proposed a pseudo-color approach, which visualizes the over- and underwriting.

Figure 1.1 illustrates the two processing techniques that have been suggested by Knox. Figure 1.1 (a) and Figure 1.1 (b) show two photographs of a parchment portion, taken under different lighting conditions. The different illuminations exhibit the overwriting and the underwriting, respectively. The underwriting is visible under UV illumination, as it is shown in Figure 1.1 (b). In Figure 1.1 (c) the corresponding pseudo-color image is presented. Figure 1.1 (d) shows an image, which exhibits purely the older text. Such images are called difference images, since they are obtained by subtracting red light images from UV images.

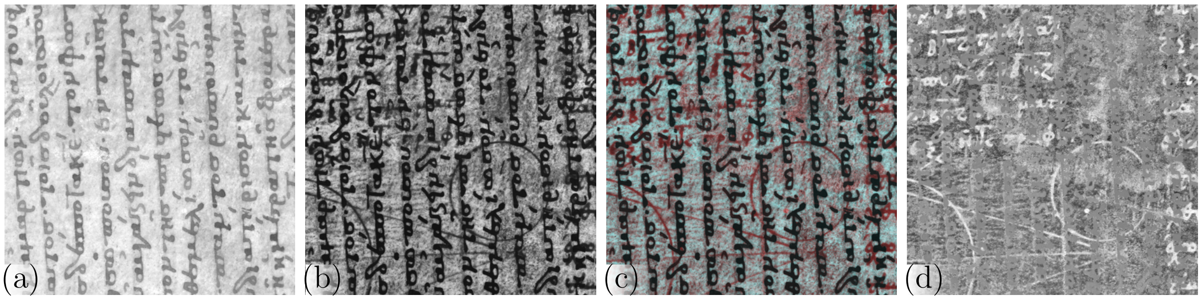


Figure 1.1: Multi-spectral imaging reveals the underwritings in leaf 40r. (a) Only the overwriting is recognizable under red light. (b) The underwriting is most visible under UV illumination. (c) The pseudo-color image exhibits the older and younger text. (d) Through multi-spectral unmixing it is possible to reveal the underwriting [Kno08].

The example images exhibit that multi-spectral imaging enhances the contrast of underwritings and facilitates a text analysis by philologists. The project team found that scholars favored pseudo-color images over difference images. The scholars disliked the latter ones, because such images contained characters that were interrupted by gaps. The philologists could not determine, whether a gap was caused by an occlusion or stemmed from a region, where no text was present. Hence, the scholars preferred images, which exhibit the overwritings and underwritings [Kno08].

The readability of the texts in such images is nevertheless limited, since the ancient writings are partially occluded by younger ones. The aim of this work is to detect the occluded regions and to restore the former writings automatically. The presented system consists of two stages: Firstly, the overwritings are detected by applying a binarization algorithm on red light images. Secondly, the retouching of the detected regions is fulfilled by applying an image inpainting algorithm on UV images [KDH$^+$11].

While the main focus of this thesis lies on the reconstruction of ancient Greek handwritings, the recovery of modern writings is also treated in the course of this work. The restoration of such texts is only demonstrated on synthetic data, but it is remarkable that inpainting can also be used for the retouching of recent handwritings, which may suffer from degradations like show-through in scans. A restoration of such damaged regions can increase the performance of an Optical Character Recognition system (OCR), as it is shown in [TGM10]. Inpainting can also be used for the recovery of handwritings that are partially occluded by ruling lines. It is stated in [CG09] that such a measurement improves the capability of an OCR system. Furthermore, Chen et al. [CLK10] have shown that the performance of a writer identification system can be improved significantly by a preceding ruling line removal step.

The papers mentioned show that a text analysis of modern writings can be facilitated, if damaged text regions are retouched before the analysis step is carried out. Hence, the inpainting algorithm, which is used for the palimpsest recovery, is also applied on modern handwritings. Additionally, the retouching of such recent writings is fulfilled with another recovery method that relies on heuristics. The algorithm is based on the tensor voting framework, suggested by Medioni and Kang [MK04].

State of the Art

The term digital inpainting was introduced by Bertalmio et al. [BSCB00] in 2000 and describes the automated filling of a specific image region. The region is called inpainting region, or domain, and is provided in the form of a binary mask. The inpainting mask may mark damaged regions - like scratches in photographs - or occluded areas - like superimposed logos in television broadcasts. Regardless of the underlying problem, the masked area is always assumed to be unknown and is solely restored based on the information that is contained in the unmasked image region. One example for an inpainting problem is given in Figure 1.2 (left), where the superimposed inpainting mask[2] is colored red. The output of the investigated inpainting method is shown in Figure 1.2 (right).

During the last decade diverse inpainting problem solutions were suggested. In the pioneering work of Bertalmio et al. [BSCB00], the inpainting task is fulfilled by solving PDE's. Thus, isophote lines are continued from the known into the unknown image regions. Another PDE based algorithm is proposed by Chan and Shen [CS00]: They suggest solving the inpainting problem with the minimization of an energy function, namely TV. Similar inpainting strategies have been proposed in [BLC08], [BBC$^+$01] and [ES02]. Common to the mentioned algorithms is that anisotropic diffusion is applied during the inpainting sequence.

A major drawback of these methods is that they are only capable of filling thin and narrow domains in an appropriate manner, while large inpainting domains are blurred out [CPT04], [TLD06], [Fid08]. The algorithms are additionally only able to restore non-

[2] The text serving as the mask was taken from: http://de.wikipedia.org/wiki/Stuttgart, last accessed on 11. August 2011.

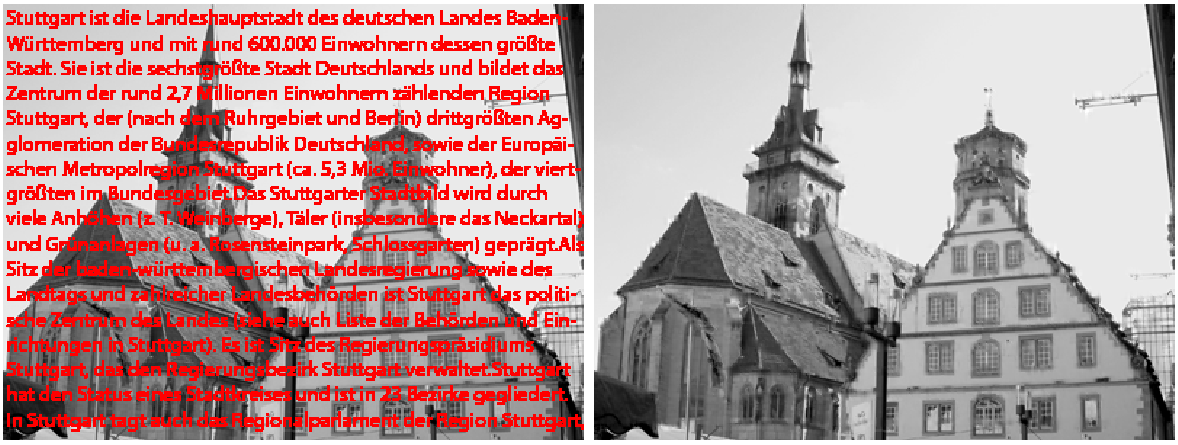

Stuttgart ist die Landeshauptstadt des deutschen Landes Baden-
Württemberg und mit rund 600.000 Einwohnern dessen größte
Stadt. Sie ist die sechstgrößte Stadt Deutschlands und bildet das
Zentrum der rund 2,7 Millionen Einwohnern zählenden Region
Stuttgart, der (nach dem Ruhrgebiet und Berlin) drittgrößten Ag-
glomeration der Bundesrepublik Deutschland, sowie der Europäi-
schen Metropolregion Stuttgart (ca. 5,3 Mio. Einwohner), der viert-
größten im Bundesgebiet.Das Stuttgarter Stadtbild wird durch
viele Anhöhen (z. T. Weinberge), Täler (insbesondere das Neckartal)
und Grünanlagen (u. a. Rosensteinpark, Schlossgarten) geprägt.Als
Sitz der baden-württembergischen Landesregierung sowie des
Landtags und zahlreicher Landesbehörden ist Stuttgart das politi-
sche Zentrum des Landes (siehe auch Liste der Behörden und Ein-
richtungen in Stuttgart). Es ist Sitz des Regierungspräsidiums
Stuttgart, das den Regierungsbezirk Stuttgart verwaltet.Stuttgart
hat den Status eines Stadtkreises und ist in 23 Bezirke gegliedert.
In Stuttgart tagt auch das Regionalparlament der Region Stuttgart,

Figure 1.2: Example for image inpainting. (Left) Input image and corresponding inpainting mask. (Right) Output of the examined inpainting algorithm.

textured images in a satisfying way [CPT04], [TLD06], [Fid08]. Criminisi et al. [CPT04] suggest overcoming these drawbacks by copying entire image patches from the known into the unknown image regions. This strategy prevents a blurring of large domains and produces more convincing resulting images than the methods in [BSCB00] and [CS00]. Several authors [CZL07], [WR09], [SJZW07], [HHL+08], [CX10], [ZRK10] have picked up the basic idea of the algorithm by Criminisi et al. and suggested extensions to the original algorithm.

The papers mentioned rely on heuristics, whereas other papers formulate the inpainting task as a global optimization problem. Komodakis and Tziritas [KT07] suggest posing the inpainting problem as a labeling problem, which is solved by an MRF. The algorithm fills the unknown image region with patches that correspond to labels. Wexler et al. [WSI07] have suggested finding the maximum in a global objective function, which rates the quality of the inpainting solutions. The algorithms, which are based on global optimization, are able to produce convincing recoveries of complex image regions, as will be shown in Section 2.

The papers mentioned above are concerned with the retouching of natural images. Cao and Govindaraju [CG09] propose instead an algorithm that is exclusively designed for the restoration of handwritings. The authors suggest training an image model on handwritings. The model is subsequently utilized for the binarization and restoration of images, which are containing handwritten characters that are overlapped by ruling lines. The ruling lines are detected automatically and their locations are encoded into inpainting masks.

The generation of the mask is not considered in the formal inpainting definition and the mentioned inpainting papers - except the last one - utilize user-supplied masks. Our motivation is similar to that in [CG09], since overwritten palimpsest regions are automatically determined by the system presented. Another similarity to the method in [CG09] is that an MRF is utilized in order to learn the statistics of handwritings. Contrary to [CG09], the proposed system is applied on gray-scale images. While Cao and Govindaraju utilize a patch-based and pairwise MRF, the palimpsest recovery system developed makes use of a pixel-based and high-order MRF.

Objective

It is intended to develop a system, which improves the readability of ancient underwritings and thus facilitates a subsequent text analysis. The readability should be improved by applying an image inpainting technique, which propagates interrupted strokes in a likely manner. Since the palimpsest recovery is fulfilled with an inpainting technique, solely the regions inside the inpainting mask are manipulated, whereas the unmasked regions are left unchanged. The over- and underwritings are in diverse conditions and the system proposed should be capable of handling the following difficulties successfully.

Before the inpainting step can be accomplished, it is necessary to identify the overlapping areas. The overwritings contain partially faded out characters, which aggravates the binarization. The mask generation is fulfilled on red light images, which are comprised of characters with diverse contrasts to the background. The first column of Figure 1.3 shows an example for a character with a low background contrast. It can be seen that the letter in the corresponding red light image - given in the second row - has a lower background contrast than the remaining overwritings in Figure 1.3.

Furthermore, the binarization is aggravated by the fact that parts of the ancient underwritings are visible under tungsten illumination. Such regions should not be incorporated into the mask, since the restoration algorithm might fill them with the background color. One example for ancient characters that are visible under tungsten illumination is given in the second column of Figure 1.3.

Similar to the overwritings, the underwritings contain strokes that are partially faded out. The restoration is further compounded by the circumstance that background clutter is present in the photographs utilized. This can be seen in the third column of Figure 1.3, where the ancient character in the image center is not entirely distinguishable from the background. The aforementioned difficulties aggravate the restoration task. In contrast, the fourth and fifth column of Figure 1.3 show palimpsest portions that are easier to solve, since the contrast between fore- and background is considerably higher than in the example images previously shown.

This work does not aim at restoring any kind of occlusions. The system is in fact only capable of restoring partially occluded letters. The reconstruction of entirely occluded characters is not possible with image inpainting algorithms, since such methods propagate surrounding image structures into unknown image regions. One example for such a problem that cannot be solved by the system is given in the sixth column of Figure 1.3.

The recovery of totally missing characters would only be possible by analyzing the contents of the ancient texts, which is not considered in this work. Instead, the main goal of this thesis is to examine the applicability of a particular inpainting algorithm for the restoration of ancient palimpsest texts. Since the system is not designed for a successful handling of all possible occlusions, the output of the proposed method has to be surveyed by a scholar.

Main Contribution

The main contribution of this work is to apply an image inpainting algorithm for the restoration of occluded handwritings in palimpsests. The system introduced makes use

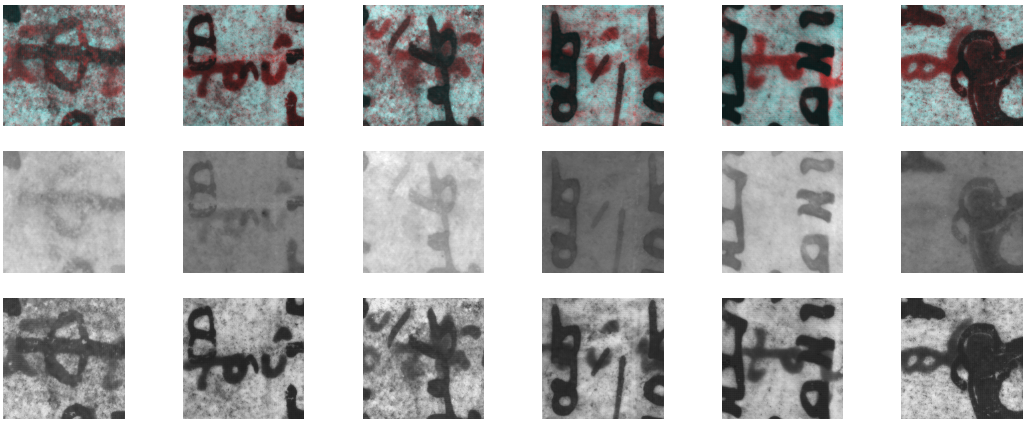


Figure 1.3: Greek letters extracted from different palimpsest folios. The first row shows pseudo-color images and the corresponding tungsten illuminated photographs are given in the second row. The UV photographs of the Greek characters are presented in the last row.

of MSI, in order to detect occlusions and to reconstruct them automatically. The determination of the occluded regions is conducted on photographs of parchments that have been exposed to tungsten light. The occluded regions are detected by a binarization algorithm, which considers the difference between a maximum and a minimum intensity value within a local neighborhood. Regions, where the differences exceed a certain threshold are assumed to be character boundaries.

The inpainting process is fulfilled on photographs of parchments, which have been illuminated with UV light. The reconstruction is accomplished by an inpainting algorithm that is named FoE approach. The FoE algorithm is based on MRF modeling and was suggested by Roth and Black in [RB05b]. The basic idea of the method is to train an image model offline and to use this model for a subsequent inference task.

Roth and Black use models that have been trained on natural images. To the best of my knowledge, this thesis is the first work in which FoE priors are trained on handwritings. It will be evaluated in Section 4.2.1 if such models are more suited for the recovery of handwritten text than models, which have been trained on natural images.

The inpainting algorithm is also evaluated on modern writings, which consist of Latin letters. Thus, it is analyzed if the FoE inpainting algorithm is suitable for the retouching of not degraded data. For this kind of data a second recovery algorithm was developed. In contrast to the FoE approach, the second method makes use of heuristics. The performances of both methods are compared in Section 4.2.3.

1.2 Scope of Discussion

The original FoE algorithm is only applied on natural images, whereas the main focus of the actual work lies on the recovery of handwritings. Nevertheless, the inpainting of natural images is also evaluated. This gives the ability to compare the inpainting technique chosen to other algorithms that are designed for natural images. Therefore, selected inpainting results, found in related literature, are compared to outputs, which are produced by the FoE algorithm. However, it is not intended to give an exhaustive comparison to other inpainting techniques, but rather to enable a basic assessment of the inpainting performance gained.

This thesis is mainly concerned with the question, if it is possible to recover handwritings successfully by applying inpainting techniques. In order to answer this question, the recovery of ancient and degraded handwritings is evaluated, as well as the inpainting of recent and undamaged text. Two inpainting methods are applied on modern handwriting data. These writings are originally undamaged and artificially created masks were utilized in the experiments. Hence, this kind of data is also referred to as synthetic data. In the tests that are conducted on synthetic data, the locations of the occluded regions are given and no mask generation step is fulfilled. Such a mask generation step would of course be inevitable in a practical application. Nevertheless, it is not intended to create an entire restoration system for recent writings, but instead to evaluate the capabilities of the inpainting techniques.

The performance of the handwriting recovery is analyzed by evaluating the similarity between ground truth images and inpainting results. It would also possible to evaluate the text recovery by comparing the performances gained by an OCR system that is applied on partially occluded writings and on the corresponding recoveries. This approach would allow for an assessment of the practical effect of the text recovery, but was not considered due to time restrictions.

The palimpsest recovery is also evaluated by comparing ground truth images to restoration results. In contrast to synthetic data, the ground truth images of the real world data had to be created manually based on transcriptions that are provided in [NAW04]. It has to be mentioned that an exact manual reconstruction of the occluded image regions is not possible, since the missing intensity values can only be estimated. Furthermore, the similarity measurement used is not a sophisticated measurement of the human perception of similarity [BLLC02]. Hence, a qualitative assessment of the inpainted images by scholars would be more appropriate. This was not realized, since no scholar was involved in this work. Instead, numerical results are presented along with exemplary images in order to enable a subjective evaluation by the reader.

1.3 Definition of Terms

In this section terms, which are frequently used in the thesis, are briefly introduced. These terms will be explained in more detail in the course of this work.

FoE **Fields of Experts**: An MRF approach that is proposed by Roth and Black [RB05b]. The main advantage over earlier MRF models is the fact that all parameters are learned from a training set and that long-range correlations between pixels can be modeled.

MAP **Maximum A Posteriori**: An inference technique that is used in Bayesian image analysis. The method is the most often used inference technique in MRF modeling [Li09]. The MAP - MRF framework was introduced by Geman and Geman [GG84] for low level vision [Pér98].

ML **Maximum Likelihood**: A statistical method that aims at learning the parameters of an MRF model. The parameters are estimated by maximizing the likelihood of the training data.

MRF **Markov Random Field**: A graphical model that allows for a modeling of contextual dependencies. The underlying theory is based on the Bayes theorem and can be used for image restoration tasks [Li09].

PDE **Partial Differential Equation**: A type of differential equations that is used by PDE based inpainting algorithms. This kind of equations allows for a smoothing of data, while preserving image discontinuities [Tsc06]. Bertalmio et al. [BSCB00] were the first who proposed solving inpainting problems with this kind of equations [KT07].

TV **Total Variation**: Mathematical problem formulation, which can be used for image denoising and inpainting tasks. Chan and Shen [CS00] introduced a TV formulation for image inpainting that is based on the TV restoration model suggested by Rudin et al. in [ROF92].

1.4 Results

The performance of the FoE algorithm was evaluated on three different types of images: Natural images, recent writings and underwritings in palimpsests. The tests conducted on natural images showed that the inpainting performance depends on the structure of the inpainting mask. Three different mask types have been restored and each mask occluded about 0.25 % of the input image. The best performance - in terms of Peak Signal-to-Noise-Ratio (PSNR) - was gained on mask regions with an average width of 3 pixels: The performance achieved was 33.48 dB. The recovery of 5 pixels wide regions led to a similarity of 32.25 dB. The worst result, namely 31.37 dB, was achieved on 11 pixels wide regions. While the restorations of the 3 pixels wide regions look visually plausible, the recoveries of the 11 pixels thick regions look implausible, since texture is not reproduced and the inpainting regions are blurred out.

In the first test conducted on recent handwritings, the performance of three different FoE priors was evaluated: One prior captured the statistics of natural images. The remaining two priors were trained on diverse handwritings: The first model was trained mainly on cursive written characters, whereas the second one was trained on uppercase letters. The first test set contained 100 cursive written words, which were overlapped by other words. The model trained on natural images gained an average similarity value of 18.06 dB. The prior which captured the statistics of cursive written characters achieved a performance of 19.03 dB and the prior that was trained on uppercase letters yielded a PSNR value of 18.78 dB. The second test set was comprised of 100 words, written in uppercase, and the performance of the aforementioned FoE models was: 16.34 dB, 17.08 dB, 17.37 dB. The experiment showed that the handwriting models capture the main orientation of the strokes, which are contained in the training database. The aforementioned dependency on the mask width was also recognized on recent handwritings: 100 handwritten words were overlaid with horizontal ruling lines, whereby the lines were 3, 5 and 7 pixels wide. The corresponding similarity values gained by the statistical based algorithm are 22.59 dB, 20.03 dB and 18.24 dB. The heuristical based approach was clearly overwhelmed, since it achieved the following lower performances: 19.20 dB, 17.17 dB and 15.58 dB.

The palimpsest recovery was analyzed on four panels extracted from different folios belonging to the Archimedes palimpsest. It turned out that the system proposed is not capable of restoring the palimpsest images at the original resolution, since the mask centers are not filled adequately. Hence, the images used had to be resampled using a resize factor of 0.25. The presented system gained the following PSNR values inside the inpainting regions: 18.52 dB, 19.46 dB, 20.36 dB and 23.04 dB. The corresponding similarity values of the entire panels are: 23.13 dB, 24.44 dB, 25.42 dB and 26.97 dB. The results for the palimpsest recovery are only modest, since the panels had to be downsized and noise was often reinforced during the inpainting task.

1.5 Organization of this Work

The basic ideas of a few inpainting algorithms have been mentioned already. A more detailed explanation of these and other inpainting techniques is given in Chapter 2. The first part of this chapter provides a survey of algorithms that aim at restoring image geometry. The second part details instead inpainting methods, which are able to reproduce texture.

In Chapter 3 the investigated algorithms are depicted. At first, the mask extraction step is described in Section 3.1. Secondly the FoE framework is discussed in Section 3.2. The underlying theory of MRF modeling in general is introduced as well as the FoE framework in particular. In Section 3.3 the FoE inpainting algorithm is applied on natural images and images containing handwritten text. Typical characteristics of the examined restoration technique are discussed based on examples. Finally, the tensor based text recovery is introduced in Section 3.4. At first the basic principles of the tensor voting framework are explained. Subsequently the restoration algorithm, which is based on tensor voting, is presented.

Chapter 4 provides experiments that demonstrate the capabilities and limits of the algorithms examined. The chapter is divided into three parts: The first part deals with the inpainting of natural images, which is fulfilled with the FoE inpainting approach. The recovery of synthetic handwriting data is depicted in Section 4.2. Both restoration techniques are evaluated and their performances are compared at the end of the section. The retouching of real world data is analyzed in the last section and in [HS11]. The mask generation step and the inpainting sequence are analyzed by means of manually created ground truth images.

The thesis is concluded by Chapter 5, where the strengths and limitations of the evaluated methods are summed up. Possible improvements are discussed at the end of this work.

Chapter 2

Related Work

This chapter provides an overview on inpainting algorithms. An exhaustive explanation of the outlined methods is beyond the scope of this thesis. Instead, this chapter is intended to give an overview over different problem solutions. Therefore the basic ideas of the approaches introduced are summed up and their advantages and disadvantages are discussed.

The inpainting techniques presented in the following can be categorized into two different groups: Geometric methods and textural inpainting approaches. The chapter structure follows this categorization. In Section 2.1 the inpainting problem is defined and geometric restoration algorithms are reviewed. The FoE framework is a geometrical inpainting technique that is based on MRF modeling. Therefore, the main principles of other MRF based algorithms will be discussed in an own subsection. The second section of the chapter deals with textural inpainting methods. Those algorithms combine the strengths of geometrical methods and texture synthesis. A few, namely 7, texture synthesis algorithms will be introduced in a separate subsection, since they are utilized by outstanding textural inpainting methods. Those methods are either based on ad-hoc principles or on global problem formulations and both categories will be depicted at the end of this chapter. Additionally, textural inpainting methods are compared to geometrical restoration algorithms.

2.1 Geometric Inpainting

In this section geometric inpainting techniques are introduced. Those methods aim at recovering image geometry and are also called structural inpainting methods. The first geometrical methods have been published, before textural inpainting techniques were introduced. This subsection starts with a description of selected works that are based on heuristics. Later on, structural inpainting techniques, which are based on MRF modeling, will be depicted.

2.1.1 Heuristical Inpainting Techniques

A pioneering approach, falling into this category, was published by Nitzberg et al. in [NMS93]. In this work the authors describe an image segmentation algorithm, which is able to connect edges that are occluded by an object. They accomplish this task by searching for T-junctions in an edge map and connecting those T-junctions based on an energy formula that minimizes the length of a connection.

Masnou and Morel [MM98] suggest using level lines instead of edges for the disocclusion task, since edges are more sensitive to contrast changes [TLD06]. The proposed method also solves the inpainting problem with the minimization of an energy formula. The energy formula used is based on the TV norm. Bertalmio et al. [BBS01] note that this inpainting method requires a simple topology and that angles are not well preserved. Another disadvantage of the algorithm is the fact that it is only able to create straight level lines, which is not suitable for the inpainting of large areas.

The term digital image inpainting was coined by Bertalmio et al. in [BSCB00]. In this work image inpainting is defined as the automated filling of an image region. The region is called the inpainting region and is provided in the form of a binary mask. Bertalmio et al. suggest fulfilling the inpainting task by solving PDEs. The basic idea of the algorithm is to smoothly complete isophote lines[1], arriving at the border of the inpainting region, from the outside of the mask into the inner region. The direction of the propagated isophote lines is orthogonal to their gradients and points into the inside of the inpainting region. The propagated information is an estimation of the variation of the smoothness which is implemented with a two-dimensional Laplace operator. This propagation is repeated iteratively and anisotropic diffusion is applied after a certain amount of iterations. This anisotropic diffusion preserves the sharpness of the edges and avoids a crossing of lines that are not connected. Figure 2.1 shows an inpainting result of the work of Bertalmio et al.

A major drawback of the method described above is its slow convergence time, which may take about a few minutes - depending on the image size. Bertalmio et al. propose a faster algorithm in [BBS01]. This technique borrows ideas from computational fluid dynamics, which is a well-studied research area [Fid08]. The authors propose to interpret an image as a stream function for an incompressible flow. The propagated information is the Laplacian - as it is in [BSCB00] - and can be seen as the vorticity of a fluid. Therefore the image defines a vector field and the Laplacian is transported into the inpainting region. With this analogy in mind, inpainting problems can be solved with Navier-stokes equations. The results of the inpainting process based on Navier-stokes equations are similar to the results in [BSCB00], but the algorithm needs less time to converge, namely a few seconds.

Another inpainting algorithm that is based on a physic theory is described in [BGS05]. This method uses the Ginzburg-Landau equation, which models phase transitions in superconductors. While the previously described methods are only able to inpaint two-dimensional images, the Ginzburg-Landau equation may be used for inpainting in a three-dimensional space. An example of a three-dimensional inpainting result is shown in Figure 2.2.

Chan and Shen [CS00] propose a TV inpainting method that is inspired by the work of Masnou and Morel [MM98]. In contrast to [MM98], the authors explicitly define the inpainting process and discuss principles for satisfying inpainting results. In order

[1] The term isophote line is a synonym for level line.

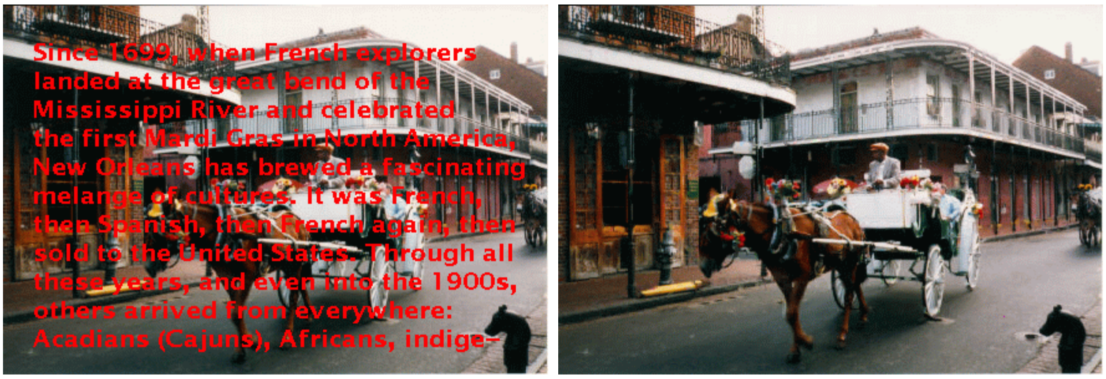

Since 1699, when French explorers
landed at the great bend of the
Mississippi River and celebrated
the first Mardi Gras in North America,
New Orleans has brewed a fascinating
melange of cultures. It was French,
then Spanish, then French again, then
sold to the United States. Through all
these years, and even into the 1900s,
others arrived from everywhere:
Acadians (Cajuns), Africans, indige-

Figure 2.1: (Left) Color image and a superimposed text, which serves as the inpainting mask. (Right) The resulting inpainted image. [BSCB00]

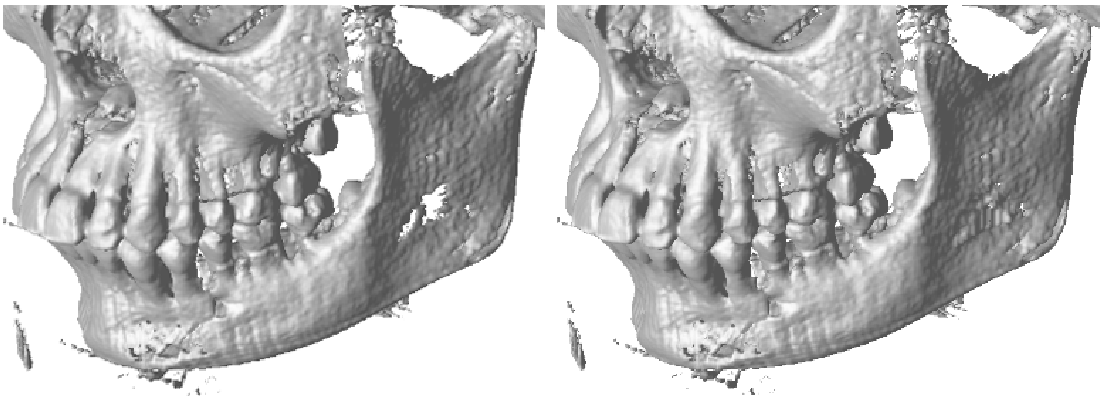


Figure 2.2: (Left) Three-dimensional input data. (Right) The hole in the cheekbone is inpainted. [BGS05]

to preserve the propagated level lines, anisotropic filtering is applied in the inpainting process. The strength of the anisotropic diffusion depends only on the strength of the isophotes as Chan and Shen note in their follow-up paper in [CS01]. In this follow-up paper the authors suggest that the diffusion strength should depend on the strength of the isophotes and additionally on the geometric information of the isophotes. The suggested formula for the diffusion strength penalizes level lines with a high curvature and instead favors level lines with a small curvature. The proposed method is able to inpaint larger regions than the previously described method in [CS00].

Recently the algorithm in [CS01] was extended by Loeza and Chen [BLC08]. The authors notice that the original algorithm requires much time for convergence. The extended algorithm overcomes this drawback with the use of a multigrid algorithm. Loeza and Chen report that their method produces similar results as the algorithm in [CS01], while needing considerably less time to converge.

Chan et al. [CKKS02] interpret the inpainting process as an application of Bayesian inference: Satisfying inpainting algorithms must take care about the prior model of images and the data model. The term data model stands here for the known regions of the inpainted image. The prior model contains a priori knowledge about curves like smoothness and convexity. The authors propose a prior model that is based on plane curves Euler's elastica. Furthermore the authors show that their Euler's elastica inpainting model is a generalization of the methods in [CS01] and [BSCB00].

Another TV inpainting algorithm is described in [BBC$^+$01] by Ballester et al. The authors present an energy function that has to be minimized w.r.t. to a vector field, which is obtained from level lines. Ballester et al. state that the proposed energy formula is based on an interpretation of Gestaltist's good continuation. This means that the algorithm attempts to connect level lines like it would be done by a human inpainter. This can be seen in Figure 2.3: The image in Figure 2.3 (b) is produced by an algorithm that minimizes the connection lengths. Ballester et al. propose an energy formula that additionally takes care about the surrounding structure. This consideration of nearby structures leads to the connection of the line in Figure 2.3 (c).

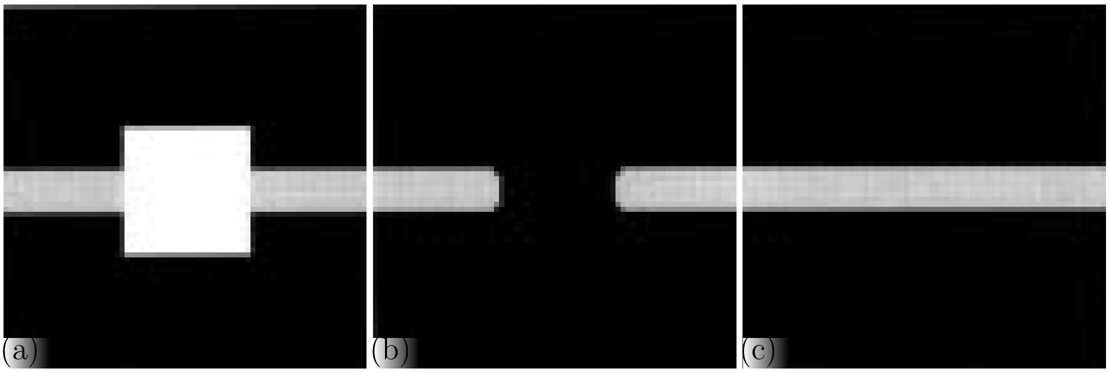


Figure 2.3: (a) Input image. The white rectangle in the middle of the picture is the inpainting mask. (b) Result obtained by minimizing the connection length. (c) Image inpainted considering surrounding structures. [BBC$^+$01]

Esedoglu and Shen [ES02] propose a TV inpainting method that is based on the Mumford-Shah image model defined in [MS88], where the model is used for segmentation. Esedoglu and Shen [ES02] note that the original model is insufficient for inpainting, since it favors straight lines. Therefore the Mumford-Shah image model is extended with the Euler's elastica curve model, which produces images with a higher quality than the original model.

While the previously described methods propagate level lines, Rareş et al. propose a method in [RRB05] that connects edges. The authors state that edges separate regions with different contents and those edges are more robust against intensity changes. The proposed method consists of three steps: In the first step the image is segmented and the edges of the contours, belonging to the segmented objects, are extracted. In the second step pairs of corresponding edges are found. In the final step the corresponding edge pairs are connected with the arcs of fitted circles. The regions that are bounded by the connecting edges are filled with interpolated color values.

Tschumperle [Tsc06] presents an inpainting algorithm that is based on curvature-preserving PDE. The author suggests a framework based on tensor driven PDE, which regularizes images and preserves the curvature of curves. The inpainting task is fulfilled with anisotropic smoothing. The geometry of an input image is encoded into a field of structure tensors. Those structure tensors are used for the computation of a field of diffusion tensors, which determines the strength of the anisotropic diffusion. An inpainting result of the algorithm is shown in Figure 2.4. According to [ALM10], Tschumperle's method is one of the most efficient PDE based inpainting algorithms and produces satisfying results, if the parameters are set correctly.

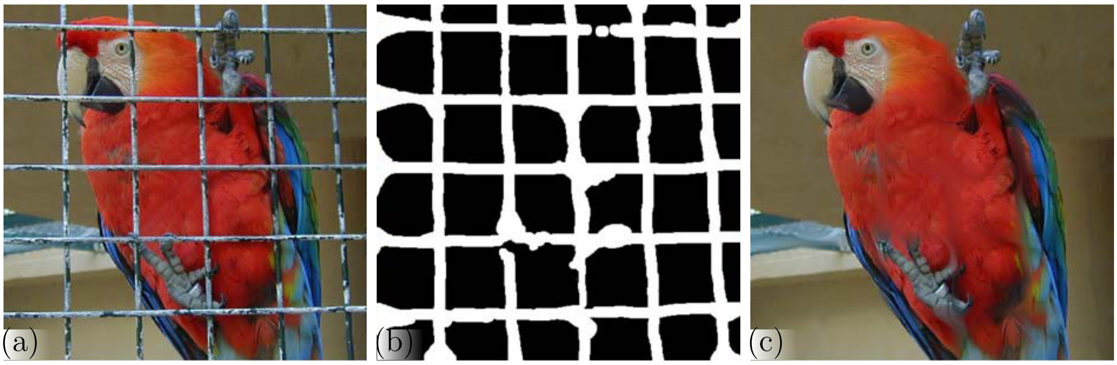


Figure 2.4: (a) Undamaged input image. (b) Mask. (c) Final inpainting result. [Tsc06]

Oliveira et al. suggest a further efficient algorithm that is based on diffusion. Contrary to [Tsc06] and [BSCB00] the authors suggest performing isotropic diffusion instead of anisotropic diffusion. The authors suggest the usage of user-defined diffusion barriers, which mark high contrast regions. The diffusion process is stopped at the barriers, in order to preserve strong edges. The semi-automatic method produces results similar to the algorithm in [BSCB00], if the barriers are set adequately.

Another inpainting method that is faster than other established PDE based methods like [BSCB00], [ES02], [CS01] is proposed by Bertozzi et al. in [BEG07]. This method is based on a slightly modified Cahn-Hilliard equation, which models the phase separation in binary fluids. The suggested inpainting algorithm works only on binary images, since the Cahn-Hilliard equation can only be applied to binary liquids. In the case of a binary image - like a binarized text - the algorithm produces similar or even better results as the methods in [BSCB00], [ES02], [CS01]. The result of an experiment conducted on synthetic data is shown in Figure 2.5. In [BHS09], the Cahn-Hilliard inpainting method is generalized to gray-value images.

Telea presents an inpainting method in [Tel04], which is inspired by the one in [BSCB00].

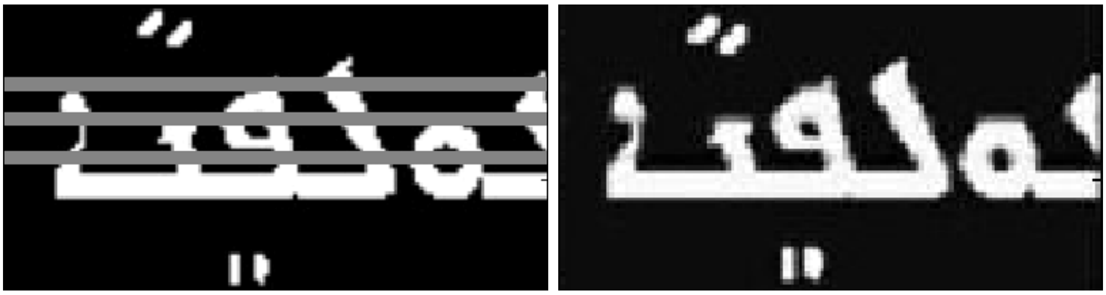


Figure 2.5: (a) Input image with superimposed mask (marked gray). (b) Restored text. [BEG07]

Pixels inside the inpainting region are inpainted successively and a fill order ensures that the most confident pixels are inpainted first. The fill order is calculated with the fast marching method that is introduced in [Set98]. The gray value that is assigned to an inpainted pixel is interpolated from the gray values of its known neighbors. The method described produces similar results as the method in [BSCB00], but it needs less time to converge. In [ALM10], this method is described as a reliable and very fast inpainting method.

2.1.2 Inpainting based on the MRF Model

The previously described methods are based on heuristics. Nevertheless, those approaches solve the inpainting problem by making assumptions about image statistics and therefore they are based on implicit statistical models. In contrast to those heuristics, statistical-based inpainting methods use an explicit model of image statistics, which is learned from a set of training images.

Levin et al. suggest learning the image statistics from the known regions of the input image. The authors propose to compute the marginal statistics of features, belonging to the known image areas. In the algorithm the gradient magnitude and the gradient angle are chosen as significant features. The inpainting is achieved by searching for a gradient field, inside the inpainting domain, which has similar statistics as the known image regions. Figure 2.6 shows how the statistics of the input image influence the inpainting result. The inpainting algorithm uses an MRF framework to find a solution, which maximizes the probability of the filling. It is notable that Levin et al. use a pairwise MRF, whose potential functions are approximated.

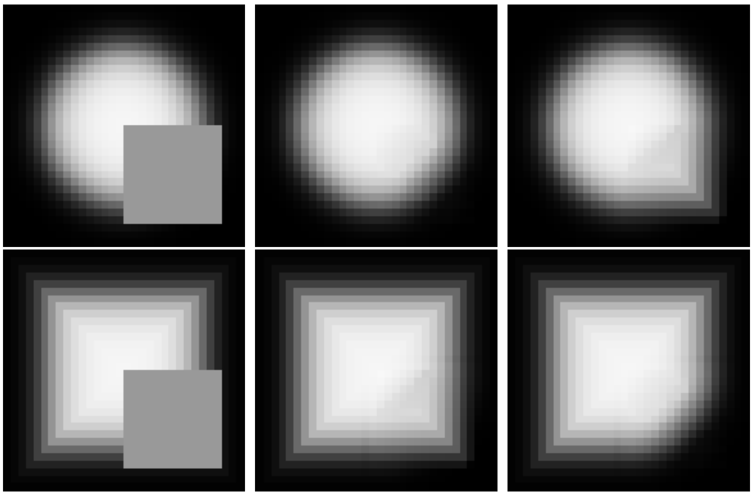


Figure 2.6: Inpainting results of two input images. The input images and the superimposed masks are shown in the left column. The images in the middle column are inpainted using image statistics that were learnt from the original input images. Each inpainted image in the right column is in contrast obtained with statistics from the other image, which leads to incorrect results. [LZW03]

Yasuda et al. [YOT05] propose another inpainting algorithm that utilizes a pairwise

MRF. In contrast to Levin et al. [LZW03] the image gradient is not regarded by Yasuda et al. The energy function of the MRF just considers the gray value differences of direct neighbors inside the inpainting domain and the differences between unknown pixels and their nearest neighbors in the known image area. The authors of the method note that the parameters of the potential functions are not learnt and that the algorithm tends to blur out edges.

Another interesting image restoration method is published by Cao and Govindaraju in [CG09]. This procedure uses a pairwise MRF for the modeling of handwritten text. The image statistics that are contained in the prior are learnt from a training set and the observation model is learned from the input image. The proposed MRF model is able to binarize handwritings and to inpaint missing text parts that are occluded by a ruling line, as it is shown in Figure 2.7.

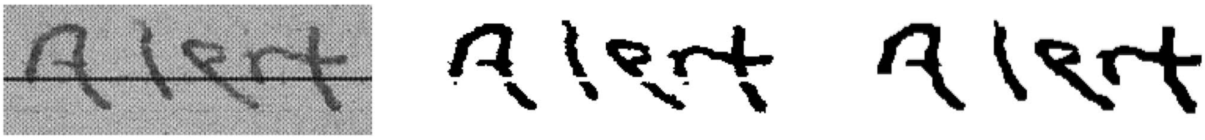


Figure 2.7: Binarization of a degraded handwriting and subsequent inpainting of missing regions. [CG09]

2.2 Textural Inpainting

The geometric inpainting techniques that were described in the previous section are only able to fill in thin, elongated inpainting regions. Additionally, those methods only produce satisfying results when they are applied on non-textured images. If geometric inpainting techniques are used for the filling of textured images they tend to oversmooth the inpainting region [KT07]. On the contrary, textural inpainting methods are designed to fill in larger and more heterogeneous regions, compared to geometric inpainting methods. In this section, important textural inpainting approaches are discussed. In some of the related literature textural inpainting is also called image completion and geometric inpainting is defined as image inpainting. However in this thesis the term image inpainting is used as a generic term for geometric and textural inpainting.

2.2.1 Texture Synthesis

The goal of texture synthesis is to produce a larger texture from a smaller input sample. The synthesized image should look visual plausible, which means that it should have similar appearance and similar structural properties as the input image. Texture synthesis is related to the inpainting field, but it is not a special kind of inpainting technique, because it is not able to fill in holes inside the input image. This task can be accomplished by textural inpainting methods, which make use of texture synthesis algorithms. In this section, established texture synthesis techniques are discussed, hence texture synthesis is an essential part of textural inpainting.

Texture synthesis algorithms can be categorized into two groups: Statistical-based approaches and image-based approaches. Statistical-based methods assume that a texture is generated by a stochastic process and the overall goal is to produce textures that seem to stem from this process. To the best of my knowledge there is just one inpainting algorithm existing, which makes use of statistical-based synthesis: In [IP97], Igehy and Pereira propose a method that is able to fill in texture, if the region around the hole belongs to the same kind of texture. However, the algorithm does not continue linear structure and therefore statistical-based techniques are not explained any further in this thesis.

Image-based approaches generate an output texture by directly copying pixels or patches from the input texture. Those methods preserve image details better than stochastic approaches, because they maintain the local neighborhood of a pixel [LHW$^+$06]. Since image based approaches simply copy gray or color values from the input texture, they are also called exemplar-based methods.

One of the first image-based approaches is published by Efros and Leung in [EL99]. In this work the output image is modeled as an MRF. This means that the value that is assigned to an unknown pixel depends on its local neighborhood. For each pixel, which is going to be filled, the algorithm searches for a patch in the input sample that is similar to the patch around the actual pixel. The similarity is measured with the Sum of Squared Differences (SSD) of the already filled pixels in the two patches. When the most similar patch is found, the value of the pixel centered on the patch is assigned to the unknown pixel. The quality of the synthesized images is better compared to statistical-based methods. However the trade-off for the more satisfying results is an increased time complexity, since searching takes place in the whole input sample.

In [WL00] Wei and Levoy propose a more efficient technique for texture synthesis. They use the same distance function as in [EL99] but the searching is accelerated with the use of tree-structured vector quantization. According to [Fid08], this acceleration leads to a decreased convergence time, but also to a reduced quality compared to [EL99].

Ashikhmin [Ash01] extends the texture synthesis by Wei and Levoy [WL00]: He suggests reducing the search space, by only allowing a few candidate positions, from which sampling may take place. The basic idea of the search space reduction is to store the coordinates from which each pixel was sampled. When it comes to the filling of a pixel, the algorithm considers the sampling coordinates of the already filled pixels in a local neighborhood. Searching takes place in the near of those coordinates. Thereby, the search time is reduced, compared to the method in [WL00]. Additionally local structures - like edges - are better preserved.

The texture synthesis methods discussed so far are pixel-based techniques, since they fill one pixel at a time. In the following, patch-based methods are explained. Those approaches copy entire patches from an input sample. Patch-based algorithms produce textures with a higher quality, compared to pixel-based techniques, since they preserve the global texture structure in a better way [KEBK05], [BSFG09].

One of the first patch-based methods is proposed by Efros and Freeman in [EF01]. The algorithm starts with a tile of the output texture with equal sized blocks randomly extracted from the input texture. In a second step smaller blocks are pasted into the boundary regions between the blocks, which were inserted in the previous step. The smaller blocks are not extracted randomly, but instead are chosen in such a way that the difference between neighboring pixel values belonging to different blocks is minimized. The authors state that their method produces similar or even better results than the method in [EL99], while needing less time to converge.

A similar algorithm is published by Liang et al. in [LLX$^+$01]. The authors mention that the search for appropriate patches is a search for the nearest neighbors. To speed up the synthesis task, Liang et al. propose to search for the Approximate Nearest Neighbors (ANN). The search is accelerated with the use of an optimized kd-tree and principal components analysis. The authors report that their generated textures overwhelm results that are obtained by [EL99] and [WL00].

Kwatra et al. [KSE$^+$03] suggest an improved version of the algorithm in [EF01]. Instead of copying entire patches the authors propose to select a candidate patch and to copy an irregularly shaped region from this candidate patch. A graph cut algorithm is used for the determination of the shape of the copied region. In Figure 2.8 (d) an image is shown that was generated with the graph cut synthesis method. Additionally the outputs of the methods in [EF01] and [WL00] are presented in Figure 2.8 (b) and Figure 2.8 (c).

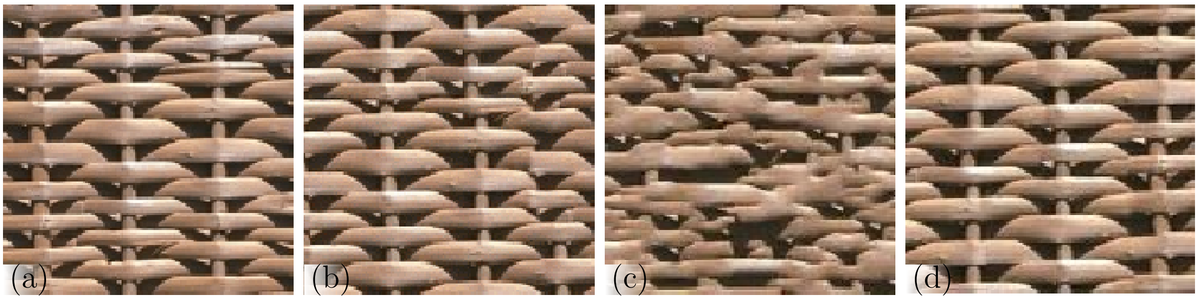


Figure 2.8: Comparison of different synthesis methods. (a) Input sample (b) Result obtained using the patch-based method in [EF01]. (c) Texture generated with the pixel-based approach in [WL00]. (d) Output of the graph cut synthesis algorithm in [KSE$^+$03]. It is remarkable that the patch-based methods clearly overwhelm the pixel-based synthesis. [KEBK05]

2.2.2 Image Completion

Image Completion is an inpainting category that combines the strengths of the just described texture synthesis methods and geometrical inpainting approaches. As with structural inpainting, image completion techniques pay special attention to the continuation of linear structures. Textural inpainting methods are additionally able to overcome the major drawback of structural inpainting techniques, which is that they tend to blur out large inpainting regions.

One of the first attempts to fulfill the inpainting task with the use of texture synthesis is proposed by Bornard et al. in [BLLC02]. Their texture synthesis algorithm extends the algorithm in [EL99] by using a synthesis order. The proposed fill order depends on the number of the already filled pixels in the neighborhood. Thus pixels, having more reliable information in their neighborhood, are filled in an early stage of the inpainting procedure. The suggested fill sequence is more appropriate for image inpainting than the sequential fill order used in [EL99]. However, the circumstance that the synthesis order does not depend on the image content may lead to an unsatisfying continuation of image features.

Another early work on textural inpainting is published in [BVSO03] by Bertalmio et al. In this method the input image is decomposed into two images, one that contains the basic image structure and one that captures texture. The image, which contains the image structure, is inpainted with the PDE based algorithm in [BSCB00] and the inpainting region of the second image is filled with the exemplar-based synthesis method in [EL99]. The two manipulated images are then added back together, which produces the final output image. The proposed method is able to inpaint texture, but it is still limited to small and narrow inpainting regions and tends to oversmooth the inpainting area [CPT04], [TLD06]. A schematic illustration of the algorithm steps is presented in Figure 2.9.

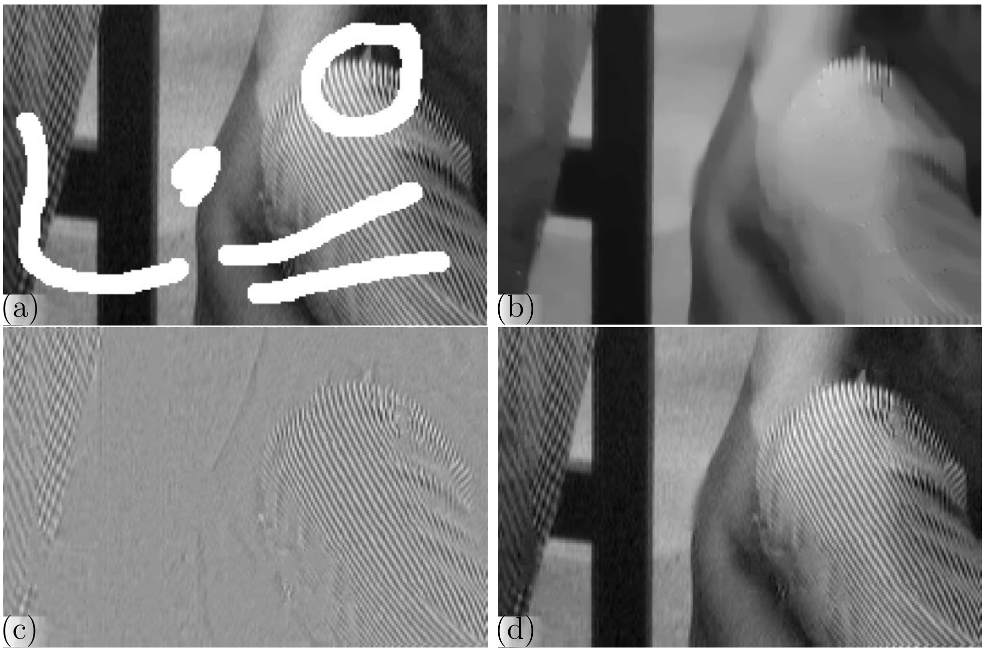


Figure 2.9: Textural inpainting result achieved by Bertalmio et al. (a) Input image with superimposed, white mask. (b) The structure of the image is inpainted separately. (c) The texture is inpainted independently too. (d) The final output is the combination of the inpainted structure and texture images. [BVSO03]

Drori et al. propose an image completion algorithm in [DCOY03] that uses an extended version of the patch-based algorithm in [EF01]. The authors suggest copying circular fragments from the known region into the unknown region. The fragment radius is not fixed and allows the copying of large fragments. In order to avoid smearing artifacts, large fragments are favored over small fragments. Similar to [BLLC02] a fill order is defined that favors pixels that have more filled pixels in their local neighborhood.

A textural inpainting method that pays special attention to curve connection is proposed in [JT03] by Jia and Tang. The first step in the algorithm is a texture based segmentation. The boundary curves of the segmented regions are afterwards encoded into 2D stick tensors and 2D tensor voting is applied. Once the voting procedure is finished, connecting curves are extracted from the voting field inside the inpainting region. At this time the whole image - including the inpainting area - is segmented. Afterwards the texture is synthesized for each segmented region separately with the use of ND tensor voting. Figure 2.10 shows an example for object removal obtained by tensor voting. The proposed method maintains curvature and is capable of filling large holes. However, the algorithm relies on image segmentation, which is an expensive task [CPT04] and hard to achieve correctly [TLD06].

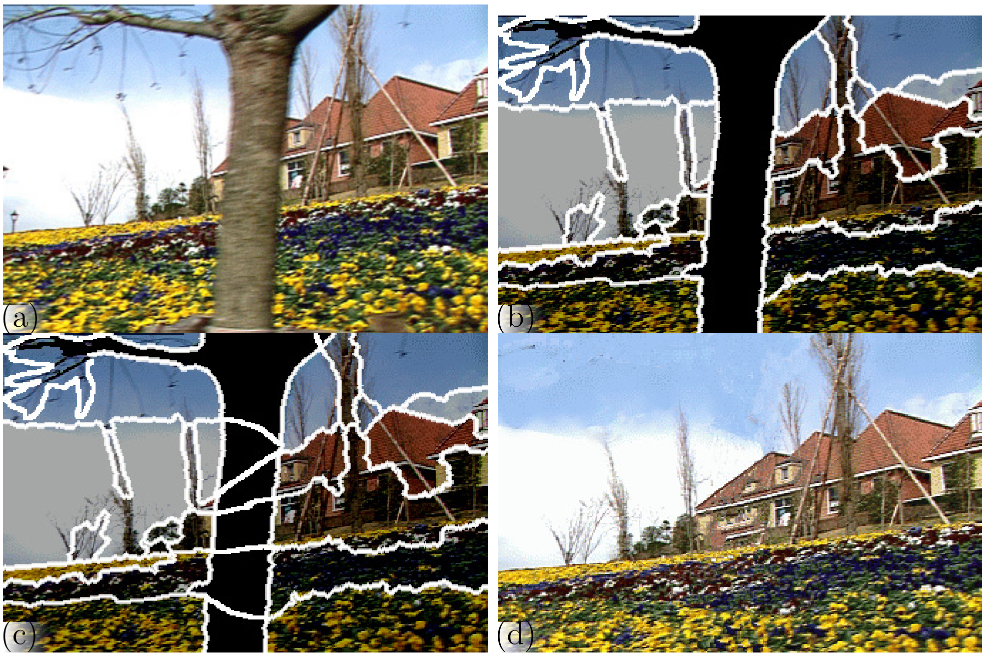


Figure 2.10: Object removal based on tensor voting. (a) The tree that partially overlaps the house and the grassland should be removed. (b) Segmentation of the regions, which are not occluded by the tree. (c) The partitioning curves are connected with the use of tensor voting. (d) In the final step, the segmented regions are inpainted with texture synthesis. [JT03]

Criminisi et al. propose a method - called exemplar-based inpainting - in [CPT04] that combines the strengths of PDE based inpainting methods and patch based texture synthesis in [EF01]. The main focus of PDE-based image inpainting methods lies on the preservation of linear structures (e.g. edges). The authors take this circumstance into account, by defining a fill order, which gives pixels belonging to edges a higher priority, than pixels belonging to homogeneous regions. The priority of one pixel is also determined by its confidence, which is a measurement of the reliability of the information surrounding the pixel: The confidence is directly proportional to the number of the pixels in the neighborhood that have been known from the beginning, or that have already been filled. Once the fill order is calculated, the patch with the highest priority is filled, which means that a similar patch from the outside of the inpainting hole is copied into the hole. The similarity measure used is the SSD.

The authors provide an exhaustive comparison between their method and the methods in [BSCB00], [DCOY03], [JT03] and [BVSO03]. The juxtaposition shows that just the algorithm based on tensor voting in [JT03] is able to produce images of comparable quality, while the other algorithms are outperformed. One example for the outstanding performance of the proposed method is shown in Figure 2.11, where a large object is removed. The result of the exemplar-based method in Figure 2.11 (b) is relatively satisfying, whereas the result in (c) is visually implausible. The latter one was obtained using the structural inpainting method by Bertalmio et al. in [BSCB00]. The patch-based synthesis approach leads to a reduction of the convergence time, compared to pixel-based methods.

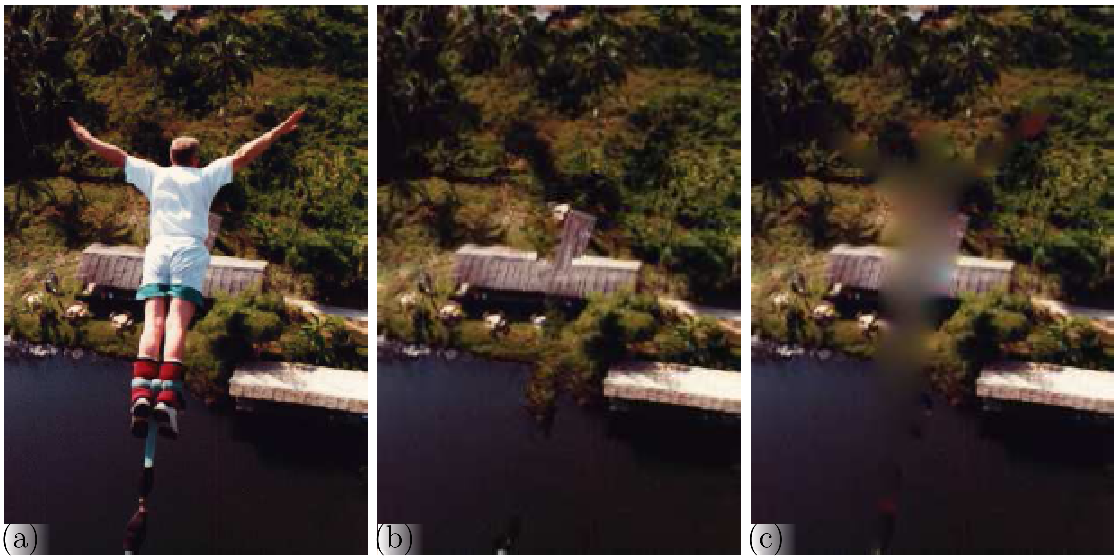


Figure 2.11: Comparison of textural and structural inpainting methods. (a) The bungee jumper should be removed. (b) Output of the algorithm published by Criminisi et al. (c) Result obtained with the structural inpainting method proposed by Bertalmio et al. in [BSCB00]. It is noticeable that the second result is clearly inferior, since it introduces blur and does not inpaint texture. [CPT04]

Although the algorithm by Criminisi et al. produces similar or even better results than previous methods it still has some drawbacks: The algorithm is not able to connect curves [Fid08]. Since the algorithm uses a greedy search for a similar patch in the whole known image, the synthesis is a time-consuming task [CZL07]. Another disadvantage is the fact that a patch is directly copied into a target region, without any modification, and once it has been synthesized it cannot be altered anymore [Cuz09].

Chen et al. reduce the convergence time of Criminisi's approach in [CZL07] by restricting the search range to a window centered on the target patch. They further propose to transfer the copied patch based on the color information in the local neighborhood of the patch, which is going to be filled. Another extension to the method [CPT04] is described in [WR09]: Instead of using a fixed patch size, the algorithm makes use of a dynamic patch size that is adapted to the size of a specific texture pattern. In [SJZW07] Shen et al. state that the SSD is not always suffice to achieve satisfying completion result. Shen et al. propose instead a similarity measurement which is based on color and gradient differences. In the proposed framework the use of this similarity measurement leads to better results than the use of the SSD. Another extension to the algorithm by Criminisi et al. is described in [HHL+08]. In this paper the authors pay special attention to the maintaining of curvature. Therefore the image is segmented and contour lines that belong together are connected with fitted Bezier curves. The following texture synthesis step is similar to the one used by Criminisi et al.

Chen and Xu propose a new exemplar-based algorithm in [CX10]. Similar to [JT03] the input image is segmented and the contour lines are connected with tensor voting. The following texture synthesis is inspired by the graph cuts texture synthesis algorithm in [KSE+03]. Instead of copying an entire patch - like in [CPT04] - the authors propose to copy an optimal portion of a patch, which leads to a reduction of artifacts that stem from rectangular patches. A comparison of the described method and the one by Criminisi et al. [CPT04] is given in Figure 2.12.

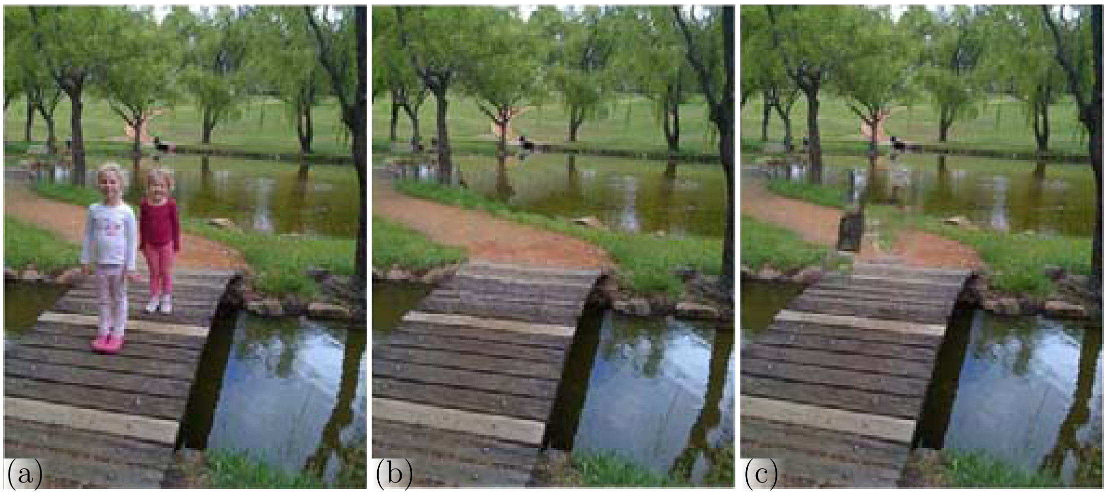


Figure 2.12: Comparison of object removal algorithms. The result obtained by Chen and Xu in (b) outperforms the outcome of the method proposed by Criminisi et al. in (c). The curvature in (b) is better maintained, since the curves are connected with tensor voting. [CX10]

Recently a further extension to the algorithm proposed by Criminisi et al. was published by Zhou and Robles-Kelly in [ZRK10]. The authors stress out that the basic fill order is not sufficient, since two neighboring pixels may have the same priority. If this is case the algorithm by Criminisi et al. has no clear answer which pixel should be filled first. Zhou and Robles-Kelly propose instead, to compute a list of candidate patches for an unknown pixel. Further patch lists, for the unknown neighboring pixels, are generated in a subsequent step. Afterwards the final patch, which maximizes the local consistency with respect to neighboring candidate patches, is selected. Figure 2.13 shows that the proposed method may overtop the approach by Criminisi et al. The basic exemplar-based method produces defects, which stem from the fact that only the current unknown pixel is considered in the inpainting process.

Contrary to the introduced exemplar-based methods is the attempt to fulfill inpainting with the use of sparse representations. In [FSM09], [ESQD05], [YSM10] and [Gul06] sparse

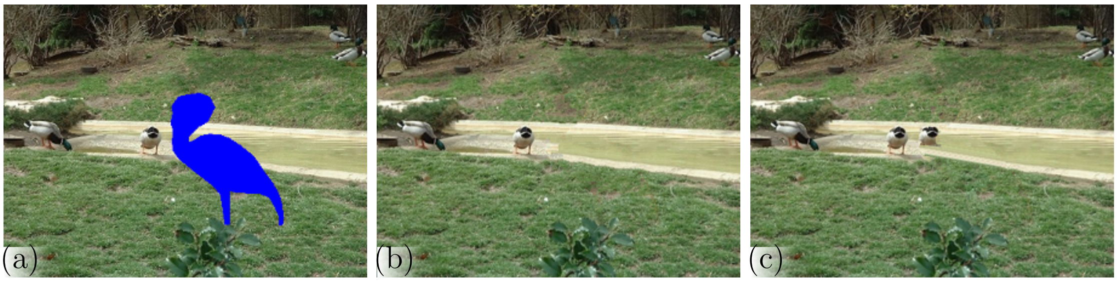


Figure 2.13: Comparison of two exemplar-based methods. (a) Input image with overlying mask (blue). (b) Outcome of the extended algorithm proposed by Zhou and Robles-Kelly. (c) Output of the basic exemplar-based algorithm published by Criminisi et al. It is obvious that the basic algorithm produces artifacts, which are avoided by the extended version. [ZRK10]

coding techniques are applied to recover unknown image data. The proposed methods are able to reproduce fine textures in a satisfying manner.

Textural Inpainting as a Global Optimization

Komodakis and Tziritas [KT07] stress out that the exemplar-based methods in [DCOY03] and [CPT04] are based on heuristics and ad hoc principles. Komodakis and Tziritas propose instead to formulate the image completion task as a global optimization problem. Therefore they suggest the use of an MRF, which poses the filling task as labeling problem. A label in this MRF framework corresponds to a patch and the labeling problem is solved with an extension of Belief Propagation (BP), which is called Priority-BP. This extension reduces the number of labels and leads to a decreased computation time (namely up to two minutes for a 256x170 image). The proposed algorithm produces images with a higher quality as in [CPT04] - as it is shown in Figure 2.14. It is stated in [BSFG09] that although the described method produces excellent results; the presented resulting images are small.

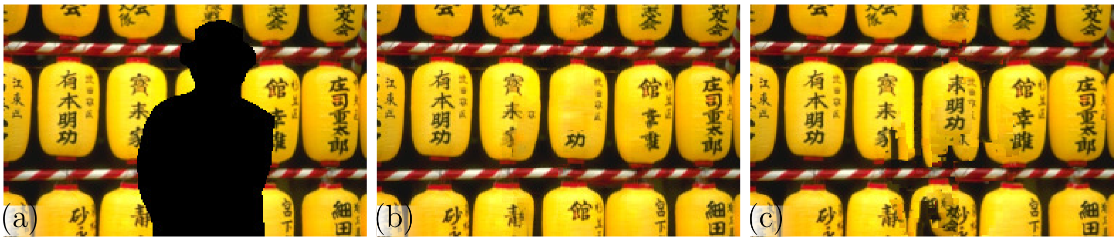


Figure 2.14: Resulting image of the algorithm, suggested by Komodakis and Tziritas. (a) Input image with superimposed, black mask (b) Output of the approach proposed by Komodakis and Tziritas (c) Inpainted image achieved by Criminisi et al. [KT07]

Lately Hsin et al. [HLLC10] proposed an algorithm similar to the just depicted MRF approach by Komodakis and Tziritas [KT07]. Like in [KT07] the labeling problem is solved with Priority-BP. The label set in the work of Hsin et al. contains labels of the known image regions. Additionally the set is enlarged with transformations of the original labels. Those transformations include scaling, vertical and horizontal flipping and rotations in 90° step. Figure 2.15 shows how the enlargement of the label set enhances the inpainting result. In this example the symmetric Eiffel tower is effectively reconstructed, because the label set contains flipped patches of the known image regions.

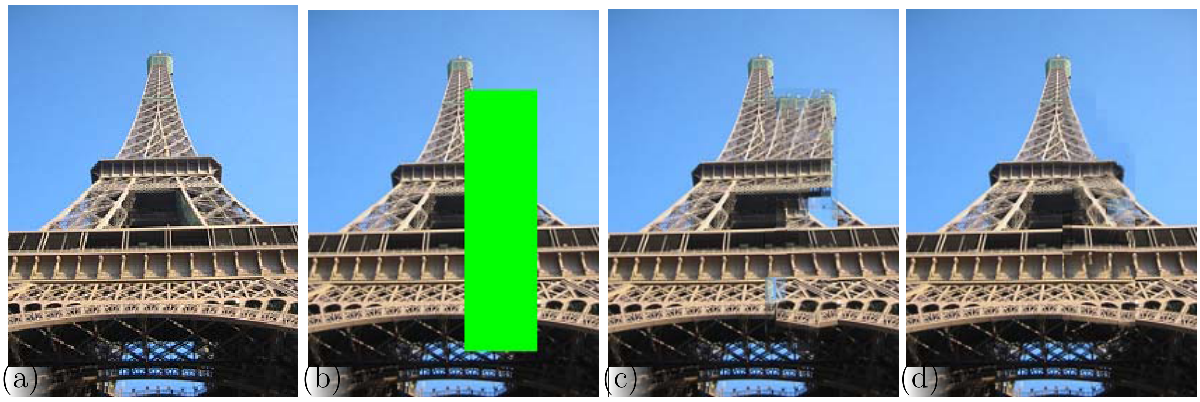


Figure 2.15: Comparison of global optimization methods: (a) Original image. (b) Masked image. (c) Output of the method in [KT07]. (d) Output of Hsin et al. [HLLC10]

Another global optimization approach is published by Wexler et al. in [WSI07]. Although the main purpose of this paper is the completion of video, the presented method is capable of filling holes in images. Image completion is achieved with a search for similar patches which are copied into the unknown image region. The method described uses a global objective function, which ranks the quality of the filling. The algorithm searches for a maximum in this function, which denotes the best solution for the completion task. Wexler et al. propose to fulfill the completion in an iterative process which starts with a low resolution and the found solution is afterwards propagated to a finer resolution level. The output images in Figure 2.16 show that the proposed method is able to produce images with higher quality as the exemplar-based method in [CPT04].

Recently an approach similar to [WSI07] was published by Barnes et al. in [BSFG09]. The algorithm by Wexler et al. searches for nearest neighbor patches and is typically slow [BSFG09]. Barnes et al. suggest a faster search method inspired by the ANN method in [Ash01]. The core idea of this method is that for a patch, which is going to be synthesized, there are several patches in its local neighborhood existing that are already synthesized. The locations from which those neighboring patches were sampled are stored and the search for the new patch takes place in regions around those locations. This search space reduction - along with other extensions - makes the algorithm by Wexler et al. applicable to greater images. The proposed algorithm is faster than the methods in [KT07] and [WSI07] and the speed up allows the usage of the algorithm in interactive editing tools. Figure 2.17 shows an image completion result, which was obtained in a few seconds.

2.3 Summary

In this chapter, algorithms found in the inpainting related literature, have been explained. It was shown that textural inpainting algorithms are better suited for the restoration of large unknown regions than structural techniques. The term digital image inpainting

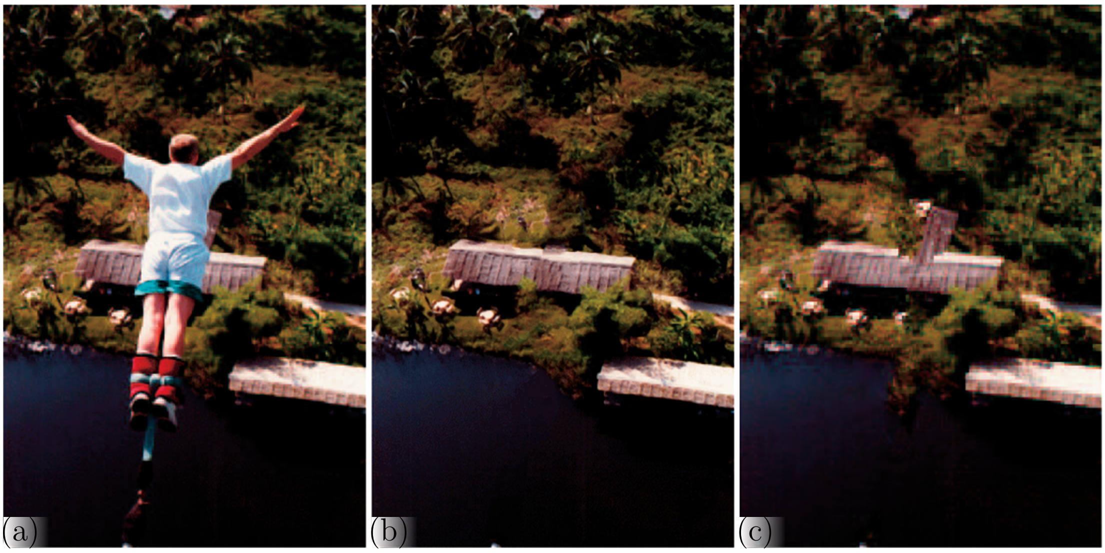


Figure 2.16: Object removal comparison: The result obtained by Wexler et al. in (b) looks more visual plausible than the output of the algorithm proposed by Criminisi et al. in (c). [WSI07]

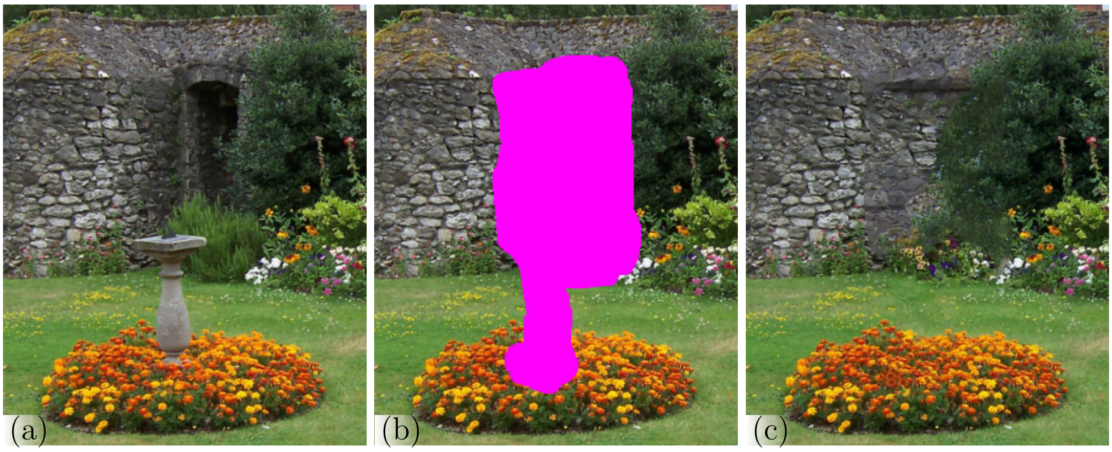


Figure 2.17: Inpainting of a large area. (a) Input image. (b) The inpainting mask overlaps various texture regions. (c) Inpainted result. It is remarkable that all textured regions are restored in a satisfying manner. [BSFG09]

exists now for about a decade, although a related algorithm [MM98] was published in 1993. Nevertheless, the majority of the papers were introduced in the last 10 years and the capabilities of the algorithms increased with time. The state-of-the-art restoration algorithms - depicted at the end of this chapter - are able to restore complex image scenes, but inpainting remains an open field of research, since the inpainting problem is certainly ill-posed [FSM09], [LZW03].

Chapter 3

Methodology

The main focus of this work lies on the reconstruction of unknown text portions. Two techniques have been examined for this purpose and they will be discussed in the following. It was depicted in the previous section that textural inpainting outperforms structural techniques, when both are applied on textured images. Nonetheless, two geometrical inpainting algorithms have been chosen, because images containing text are far less textured than natural images.

The proposed approaches differ greatly on their underlying concepts. The first inpainting algorithm makes use of the FoE framework, which is based on Bayesian image analysis. Since the underlying theory of the FoE framework is relatively complex, it will be explained separately in Section 3.2. Subsequently inpainting with the FoE model is detailed in Section 3.3. The first part of this section deals with the inpainting of natural images, while the second one is dedicated to text reconstructions.

The second proposed method is depicted in Section 3.4. The algorithm is based on an ad-hoc principle - namely tensor voting - and is exclusively designed for the inpainting of text characters. The heuristical approach considers image gradients - in contrast to the FoE algorithm. The inpainting quality of the heuristic method relies strongly on an edge detection step. Thus, this approach is only applicable to images, where the handwritings are clearly identifiable. In the case of the palimpsests that are considered in this thesis, the contrast between the older text and the background is not sufficient enough to achieve satisfying inpainting results. Therefore the second approach is only used for the restoration of non-degraded texts.

Inpainting of palimpsest texts is fulfilled with the FoE model. Before inpainting may take place, it is necessary to extract an inpainting mask from the overwritten text. Although the overwriting is generally in a better condition than the older text, it still suffers from certain degradation processes. In order to extract a reliable inpainting mask, a binarization algorithm for degraded documents was implemented. The binarization step is briefly explained in Section 3.1.

3.1 Extracting Inpainting Masks from Palimpsests

The underwriting is not visible under red light, whereas the overwriting is still present [Kno08]. Therefore the inpainting masks are generated from photographs of parchments, which have been exposed to tungsten illumination. The contrast between the foreground and the background of those documents is strongly varying, which complicates the binarization step. Recently Su et al. proposed a simple but yet effective binarization approach in [SLT10]. Their method is especially designed for historical documents, which makes it suitable for our needs.

The binarization algorithm starts with the construction of a contrast image. Each pixel in this image encodes the difference between the maximum and the minimum intensity value in a local neighborhood. The contrast is higher at the border of character strokes, compared to the homogeneous image regions. However, Su et al. note that taking the absolute difference is insufficient, because the high background variation in historical documents may lead to false foreground classifications. In order to suppress the background variation, the absolute difference is normalized and the contrast image is defined by:

$$D(x,y) = \frac{f_{max}(x,y) - f_{min}(x,y)}{f_{max}(x,y) + f_{min}(x,y) + \epsilon} \tag{3.1}$$

where $f_{max}(x,y)$ and $f_{min}(x,y)$ denote the maximum and minimum gray value in a local neighborhood. In this work a window size of 3×3 is used. $\epsilon > 0$ is an infinitely small value that avoids a division by zero.

The denominator is smaller if a dark stroke is surrounded by slightly brighter background pixels, compared to a character that has a higher background contrast. However, in the former mentioned case the nominator is also smaller - compared to the latter case - and thus the influence of the dark background region is lowered. If the absolute contrast value is higher, the greater value of the denominator diminishes the overall image contrast. Thus, the normalization reduces the influence of the background variation. In Figure 3.1 the contrast image of a parchment part is shown.

Once the contrast image is calculated, pixels are marked as high contrast pixels if they exceed a global threshold. This threshold is determined with Otsu's [Ots79] thresholding approach. At this juncture the pixels around stroke borders are classified as high contrast pixels. What needs to be remained is a classification of the strokes. A pixel is classified as a foreground pixel, if two requirements are met: First, the pixel should be in the near of high contrast pixels. Second, the pixel intensity must be smaller or equal than the mean intensity of the high contrast pixels in a local neighborhood window N_w. This considerations lead to the following equation:

$$R(x,y) = \begin{cases} 1 & N_e \geq N_{min} \& I(x,y) \leq E_{mean} + E_{std}/2 \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

where N_e is the number of high contrast pixels in a local neighborhood window. N_{min} is the minimum number of high contrast neighbors. The mean and the standard deviation of the intensity of the high contrast pixels in a local neighborhood are denoted as E_{mean}

and E_{std}. The neighborhood sizes of both conditions in Equation 3.2 are equal. The choice of the local neighborhood window depends on the stroke width. Figure 3.1 (c) shows the binarization result of the contrast image in Figure 3.1 (b). For this example a local neighborhood window size of 11 × 11 was used and N_{min} was set to 4.

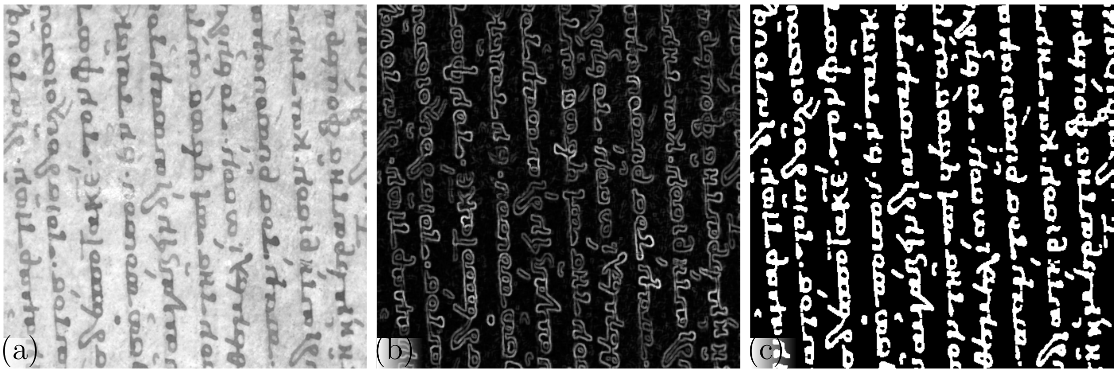


Figure 3.1: Binarization of an image part belonging to leaf 40r. (a) Input image, which was illuminated with tungsten light. (b) Contrast image. (c) Binarized output, obtained with the method proposed by Su et al.

3.2 Fields of Experts framework

In order to understand the FoE model - which is a kind of MRF - it is necessary to understand the concepts of MRF models. Therefore, the basic concepts of MRF and the related terminology will be explained in the first part of this section. A detailed discussion of MRF models is far beyond the scope of this thesis and the interested reader is referred to Li's [Li09] exhaustive monograph on MRFs in computer vision. Once MRFs in general are introduced, the FoE model will be explained in detail.

3.2.1 Markov Random Fields

Bayesian image analysis

Bayesian statistics play an important role in the FoE-framework, since the learning and inference algorithms are based on the Bayes theorem, which is defined as:

$$p(x|y) = \frac{p(y|x) \cdot p(x)}{p(y)} \tag{3.3}$$

where the posterior $p(x|y)$ depends on the likelihood $p(y|x)$ and the priors $p(x)$ and $p(y)$. The posterior probability $p(x|y)$ is the conditional probability that an event x occurs after another event y has already occurred. The likelihood $p(y|x)$ describes a model of the observation process. The prior $p(x)$ is a prior probability which represents an a-priori knowledge about an event x that is defined without a concrete observation. Finally, $p(y)$ acts as a normalizing constant.

The Bayes theorem may be used for image restoration, where the aim is to restore an ideal image x from an observed image y. In the work of Roth and Black [RB05b] the FoE model is used for two image restoration tasks: Image denoising and image inpainting. Image denoising is a more generic case of image restoration, while the latter task is more specialized - as we will see later on. Therefore, a short description of image denoising with Bayesian statistics may provide a more comprehensible explanation of Bayesian image analysis: The goal in image denoising is to restore a noise free image x from an observed image y that is corrupted with noise. The likelihood $p(y|x)$ contains information about the noise that corrupted the ideal image x. For example the noise may be assumed to be Gaussian. The prior $p(x)$ contains information about natural image statistics [Rot07]. With the use of Bayesian inference it is possible to maximize the posterior $p(x|y)$ in Equation 3.3, which leads to the desired noise reduction. Since $p(y)$ is a constant it can be omitted and Equation 3.3 becomes:

$$p(x|y) \propto p(y|x) \cdot p(x). \tag{3.4}$$

In image restoration one is seeking for the optimal solution x^* that maximizes the posterior probability $p(x|y)$. This inference technique is named MAP estimation and is formally defined by:

$$x^* = \arg\max p(x|y) \tag{3.5}$$

The MAP criterion is the most commonly used inference technique in MRF modeling [Li09]. In this field the MAP technique is called the MAP-MRF framework, which was advocated by Geman and Geman [GG84] for low level vision [Pér98]. Since then MRFs have been utilized for certain low level vision applications, including image restoration [Bes86], [GG84], [Rot07], optical flow estimation [LH08], [RG00], image interpolation [GYZ$^+$09] etc..

MRF modeling consists of two main parts: The inference technique and learning of the MRF parameters Θ. The parameters of an FoE model are learnt with the ML approach. This method is - like the MAP approach - based on the Bayes theorem. The ML approach estimates the optimal parameters Θ^* by maximizing the likelihood of a training dataset $\mathcal{X}$:

$$\Theta^* = \arg\max p(\mathcal{X}; \Theta). \tag{3.6}$$

In general there is no closed form solution for the ML equation existing [Rot07]. Therefore, the parameters Θ^* have to be approximated.

Graphical Models

An MRF is an undirected graphical model. Such a graph $G = (V, E)$ consists of nodes $v \in V$ and edges $e \in E$ that connect the nodes. With this graph G it is possible to model a d-dimensional random vector $\mathbf{x}$. Each random variable x_v is represented by a node v. The edges describe the relationships between the nodes - or respectively the random variables. A clique $c \in C$ is a subset of neighboring nodes v. For each clique c there is a potential function f_c existing that assigns a positive value to the clique. The joint distribution is defined as:

$$p(\mathbf{x}) = \frac{1}{Z} \prod_{c \in C} f_c(\mathbf{x}_{(c)}) \tag{3.7}$$

where

$$Z = \int_{\mathbf{x}} \prod_{c \in C} f_c(\mathbf{x}_{(c)}) \, d\mathbf{x} \tag{3.8}$$

is a normalizing constant. Z is also named partition function and assures that $p(\mathbf{x})$ integrates to 1. Usually the computation of the partition function is intractable and it has to be approximated. This circumstance is a main difficulty in MRF modeling. Most MRF models - including the FoE model - are said to be homogeneous, which means that the potential functions are independent of their position on the graph [Li09]. Thus, the prior depends only on the actual configuration of a clique and is translation invariant.

A graph G is an MRF if the following two conditions are met:

$$p(\mathbf{x}) > 0 \tag{3.9}$$

$$p(x_v | \mathbf{x}_{V \backslash \{v\}}) = p(x_v | \mathbf{x}_{\mathcal{N}(v)}) \tag{3.10}$$

where $V \backslash \{v\}$ is the set of all nodes except the node v and $\mathcal{N}(v)$ is a subset, which contains the nodes in a local neighborhood of v. Equation 3.9 is called positivity and is usually fulfilled [Li09]. Equation 3.10 is named Markovianity, which describes the fact that v depends only on its local neighborhood $\mathcal{N}(v)$ and is conditionally independent of all the other nodes in the graph. In the FoE framework, a graph models a two dimensional image. Each node of a graph corresponds to an image pixel. The Markovianity depicts the assumption that the value of an actual pixel is conditionally independent of the other pixels in the image, given its neighboring pixels. Figure 3.2 illustrates an MRF as a graphical model that is used for image denoising. It is notable that the observed pixels are only connected to their noise free counterparts and not to other observed pixels, while each restored pixel is connected to its four neighbors. This models the assumption that each restored pixel is dependent on its restored neighbors and the associated observation. The edges between the observed and restored pixels represent the observation model, which is described by the likelihood.

Depending on the maximal clique size there are two different MRF types existing: Pairwise MRFs and high-order MRFs. An example for a pairwise MRF is shown in Figure 3.2. In such a pairwise MRF the largest clique size is two. Thus, the prior is obtained, by connecting each node to its four direct neighbors. If the clique size is larger than two, the graphical model is called a high-order MRF.

3.2.2 Fields of Experts

In the past MRFs were typically pairwise and the parameters for the potential functions were often chosen by hand [Rot07]. The MRFs used for geometrical inpainting - outlined in Section 2.1.2 - are for instance pairwise models. Roth and Black propose a high-order

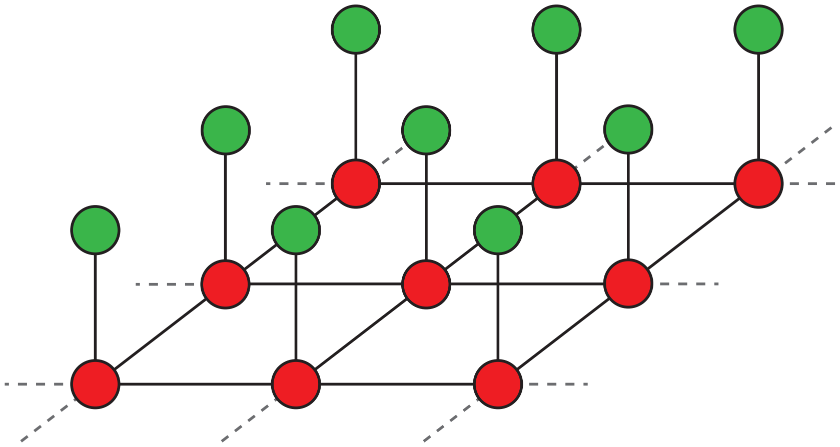


Figure 3.2: An MRF that is used for image denoising. The green nodes represent the observed, noisy pixels, while the red ones denote the restored, noise free pixels.

MRF in [RB05b]. The authors show that the FoE model outperforms simple pairwise graphical models, because it is able to model long-range correlations in images, while pairwise models are only able to model dependencies between direct neighboring pixels. Schmidt [Sch10] reasserts this hypothesis in an exhaustive comparison between the FoE model and pairwise MRF models.

Basic Model

The potential functions that are used in the FoE framework are called experts and were introduced by Hinton in [Hin99]. The experts are defined as:

$$f(\mathbf{x}_{(k)}) = f_{\mathrm{PoE}}(\mathbf{x}_{(k)}; \Theta) = \prod_{i=1}^{N} \phi(\mathbf{J}_i^{\mathrm{T}} \mathbf{x}_{(k)}; \alpha_i), \tag{3.11}$$

where $\mathbf{x}_{(k)}$ is the k-th patch of an image $\mathbf{x}$. The image patch $\mathbf{x}_{(k)}$ is projected onto a linear filter $\mathbf{J}_i$. Thus, the maximal clique size is equal to the size of the filters. In this thesis 3×3 filters are evaluated, although a higher performance can be gained by using larger filters [Rot07]. Filters with a larger size are nevertheless not used in this work due to time restrictions, since the usage of larger filters leads to an increased learning and inference time.

There are N filters and expert parameters existing. It is worth noting that this number of experts N is not predefined, but is instead chosen by the designer of the MRF. For the inpainting results presented in this work, N was set to 8, as it is recommended by Roth [Rot07] for the clique size chosen.

What needs to be defined is an appropriate expert function ϕ. According to Roth and Black the expert functions should be heavy tailed functions, since those functions are able to model the statistics of natural images. The marginal distributions of the derivates of such natural images are strongly non Gaussian. This can be seen in Figure 3.3, where the blue line denotes the intensity differences of neighboring pixels (in horizontal and vertical directions). The intensity differences are computed from a database that consists of natural images (see Section 3.3.1 for details). The dashed red line in Figure 3.3 denotes a Gaussian fit of the intensity differences. The Gaussian distribution is not able to model the heavy-tailed property of the intensity differences. Instead of using a Gaussian distribution, Roth and Black suggest to use Student t-distributions as expert functions. Those functions are heavy-tailed distributions, as it can be seen in Figure 3.3, where the green line is a Student t-distribution. The utilized expert functions are in fact simplified Student t-distributions, which are defined by:

$$\phi(\mathbf{J}_i^{\mathrm{T}}\mathbf{x}; \alpha_i) = \left(1 + \frac{1}{2}(\mathbf{J}_i^{\mathrm{T}}\mathbf{x})^2\right)^{-\alpha_i} \tag{3.12}$$

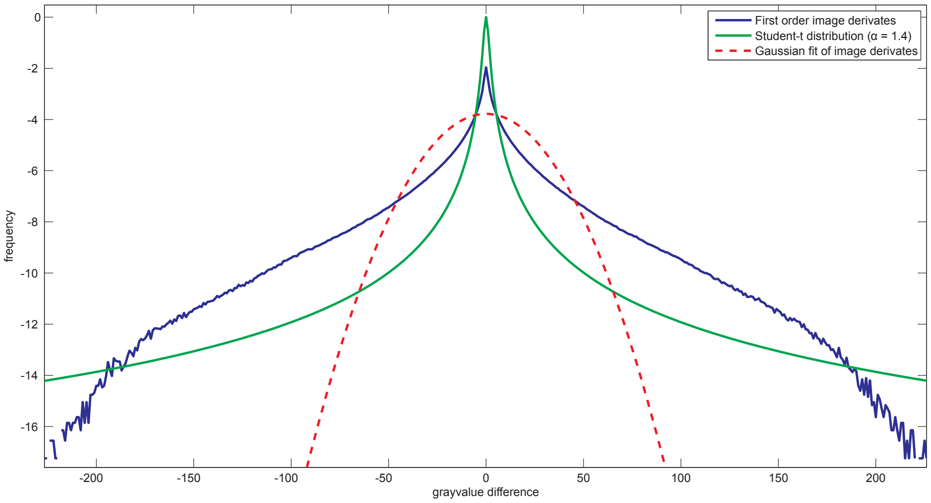


Figure 3.3: Log plot of the first order image derivates. The fitted Gaussian distribution (dashed red line) is not able to model the images derivates (blue line) sufficiently. The Student t-distribution (green line) models the heavy-tailed property of the intensity differences more accurately.

Since the potential functions are now defined, the generic MRF prior in Equation 3.7 can be reformulated into the FoE prior:

$$\begin{aligned} p_{\mathrm{FoE}}(\mathbf{x}; \Theta) &= \frac{1}{Z(\Theta)} \prod_{k=1}^{K} f_{\mathrm{PoE}}(\mathbf{x}_{(k)}; \Theta) \\ &= \frac{1}{Z(\Theta)} \prod_{k=1}^{K} \prod_{i=1}^{N} \phi(\mathbf{J}_i^{\mathrm{T}}\mathbf{x}_{(k)}; \alpha_i), \end{aligned} \tag{3.13}$$

where K is the number of nodes - or respectively pixels[1] - of an image $\mathbf{x}$, $Z(\Theta)$ is the partition function for a concrete parameter set Θ and the other variables are defined as

[1] The number of pixels where the prior is evaluated is smaller than the number of pixels of an image, strictly speaking. This stems from the required boundary handling in the FoE framework: A pixel is only evaluated if its corresponding clique has the maximal clique size.

above. The learning procedure is fulfilled on the log of the FoE and therefore the prior is reformulated as:

$$\begin{aligned} p_{\mathrm{FoE}}(\mathbf{x};\Theta) &= \frac{1}{Z(\Theta)}\exp\{-E_{\mathrm{FoE}}(\mathbf{x};\Theta)\} \\ &= \frac{1}{Z(\Theta)}\exp\left\{\sum_{k=1}^{K}\sum_{n=1}^{N}\psi(\mathbf{J}_i^{\mathrm{T}}\mathbf{x}_{(k)};\alpha_i)\right\}, \end{aligned} \tag{3.14}$$

where $\psi(\mathbf{J}_i^{\mathrm{T}}\mathbf{x}_{(k)};\alpha_i) = \log\phi(\mathbf{J}_i^{\mathrm{T}}\mathbf{x}_{(k)};\alpha_i)$. The above formulation is in fact a Gibbs distribution. The equivalence of the probability density of an MRF and a Gibbs distribution is established in the Hammersley-Clifford theorem [HC71].

Training

The filters and the corresponding expert parameters are learned from a dataset $\mathcal{X}$ that consists of D image patches. Since the training of an FoE model is a highly time-consuming task, the model is trained on 15×15 patches instead of whole images - as it is suggested by Roth and Black [RB05b]. Although this patch size is dramatically smaller than the size of the inpainted images, it nevertheless affords a modeling of the overlapping cliques in the FoE framework. In the case of 3×3 filters each training patch still consists of 13×13 cliques.

The training algorithm is based on the ML approach defined in Equation 3.6. Thus, the likelihood is maximized w.r.t. the parameter set Θ. Unfortunately there is no closed form solution for the optimal parameters Θ^* existing and the parameters have to be estimated in an approximate manner: In an iterative process a gradient ascent is performed on the log of the likelihood. In each iteration all parameters θ_j - this includes the filter coefficients and the expert parameters - are updated. The update formula that is used during the training sequence is defined by:

$$\delta\theta_j = \eta\left\{-\left\langle\frac{\partial E_{\mathrm{FoE}}}{\partial\theta_j}\right\rangle_{\mathcal{X}} + \left\langle\frac{\partial E_{\mathrm{FoE}}}{\partial\theta_j}\right\rangle_{p_{\mathrm{FoE}}}\right\}, \tag{3.15}$$

where η is a learning rate and the energy E_{FoE} is defined as in Equation 3.14. In the present work η is set to 0.01 - as it is proposed in [Rot07]. The partial derivates of the energy w.r.t. to the parameters are presented in [Rot07]. $\langle . \rangle_{\mathcal{X}}$ represents the expectation over the training data and $\langle . \rangle p_{\mathrm{FoE}}$ denotes the expectation w.r.t. the model distribution $p(\mathbf{x})$.

While the computation of the average over the training data $\mathcal{X}$ is feasible, the expectation over the model distribution has to be approximated. The expectation over the model distribution is approximated by taking the average of samples that are drawn from $p_{\mathrm{FoE}}(\mathbf{x};\Theta)$. The samples are generated using a Markov Chain Monte Carlo (MCMC) method. MCMC methods are useful when complex or high-dimensional probability functions cannot be analyzed directly. The aim of MCMC techniques is to sample from a distribution that is as close as possible to the demanded distribution. Such a distribution is constructed with the help of a Markov Chain (MC) [Nea93]. The MCMC method that is used in the FoE framework is called Hybrid Monte Carlo (HMC) sampler, which is an efficient sampler and was proposed by Duane et al. in [Dua87]. An exhaustive explanation of the HMC sampler is given in [Nea93].

Although an efficient sampler is used, there is still a need for a further decrease of the learning time: To reduce the computational burden, Roth and Black suppose to make use of the Contrastive Divergence (CD) [Hin02] criterion. The idea of CD is to initialize the MC with the training data $\mathcal{X}$. Thereby only a few iterations are necessary until the MC converges to the desired model distribution. In this thesis only one MCMC iteration is used. Roth [Rot07] demonstrates that this is sufficient for the parameter estimation.

20.000 patches are used for the training task - as it is recommend in [RB05b]. In order to lower the training time a further measure is taken: Instead of processing the whole training set in one iteration, 100 patches are selected randomly from the training set. This random selection and a randomized process, required by the HMC sampler, lead to non converging parameter updates. Thus, the parameter updates are manually observed. It was found that around 5.000 iterations are necessary to obtain mainly stabilized parameters. The manual monitoring of the parameter updates is also proposed by Roth.

Inference

The inference in the FoE framework is based on the MAP approach that was discussed in Section 3.2.1. Unfortunately, for graphical models the computation of the exact inference in Equation 3.5 is NP hard in general [Rot07]. Therefore - like in the training algorithm - an approximate technique is applied: In an iterative procedure, a gradient ascent is performed on the log of the posterior:

$$\mathbf{x} \leftarrow \mathbf{x} + \eta[\nabla_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x}) + \nabla_{\mathbf{x}} \log p_{\mathrm{FoE}}(\mathbf{x})], \tag{3.16}$$

where η is a user defined step size. The likelihood term in Equation 3.16 describes the observation model and has not been defined yet. In the case of image denoising, the likelihood contains information about the noise that corrupted the image. The likelihood for inpainting is different, because of two reasons: On the one hand, the area that is not part of the inpainting mask $\mathcal{M}$ must not be altered, because it is not damaged. On the other hand, we assume that no observation is made in the inpainting region. This considerations lead to the following likelihood formula:

$$p(\mathbf{y}|\mathbf{x}) = \prod_{j=1}^{M} p(y_i|x_i) = \prod_{j=1}^{M} \begin{cases} 1 & j \in \mathcal{M} \\ \delta(y_j - x_j) & j \notin \mathcal{M} \end{cases} \tag{3.17}$$

where the Dirac delta is defined by $\delta(f) = 0$ for all $f \neq 0$. The Dirac delta assures that any inpainting operation, which alters the known image area, has a probability of 0. Since the likelihood is now defined, we can formulate the overall inpainting algorithm, which is defined by:

$$\begin{aligned} \mathbf{x}^{(t+1)} &= \mathbf{x}^{(t)} + \eta M[\nabla_{\mathbf{x}} \log p_{\mathrm{FoE}}(\mathbf{x}^{(t)})] \\ &= \mathbf{x}^{(t)} + \eta M \left[\sum_{i=1}^{N} \mathbf{J}_i^{-} * \boldsymbol{\psi}'(\mathbf{J}_i * \mathbf{x}^{(t)}; \alpha_i) \right]. \end{aligned} \tag{3.18}$$

For the sake of simplicity the derivation of $\nabla_{\mathbf{x}} \log p_{\mathrm{FoE}}(\mathbf{x})$ is not explained here. The filter $\mathbf{J}_i^-$ is obtained by mirroring $\mathbf{J}_i$ around its center and t is the actual iteration of the inpainting process. The derivate of the log of the Student t-distribution is defined as:

$$\psi'(y;\alpha) = -\frac{\alpha y}{1+\frac{1}{2}y^2}, \tag{3.19}$$

where the scalar input y may be replaced with the dot product $\mathbf{J}_i^{\mathrm{T}}\mathbf{x}_{(k)}$. However, the input of $\boldsymbol{\psi}'$ in Equation 3.18 is not a scalar value, but instead an image, which results from the convolution $\mathbf{J}_i * \mathbf{x}^{(t)}$. In Equation 3.18, $\boldsymbol{\psi}'(\mathbf{x})$ is a vector, which contains all $\psi'(\mathbf{x}_k)$. This element-wise evaluation allows a simple and efficient implementation of the inpainting algorithm, with the use of convolutions [Rot07].

The models that are used in this thesis are all trained on grayscale images. If the input image is a color image, each channel is convolved separately. It is possible to train an FoE prior successfully on color images, as it is reported by McAuley et al. [MCSF06]. In their proposed extension the FoE parameters are learnt for each color channel separately, which leads to a tripling of the parameter number. Although this might enhance the inpainting of color images, the approach is not investigated here, since the increased parameter number would lead to a raised learning time.

It is notable that the convolutions, which are performed during the restoration task, are very similar to anisotropic diffusion. It was shown in Section 2.1 that the structural inpainting algorithms proposed in [BSCB00], [CS00], [CS01] and [Tsc06] make use of nonlinear diffusion. The main difference between the FoE approach and the aforesaid methods is that the FoE inpainting technique requires no computation of the image gradient. Another advantage over usual diffusion techniques is the fact that the FoE framework allows the usage of more filters.

FoE Applications

Beside image inpainting and denoising the FoE framework has also been applied successfully to novel view synthesis [WRTF06], image interpolation [GYZ$^+$09] and optical flow estimation [Rot07]. In the case of optical flow estimation the statistical model is trained on a database that contains scene depth information. In the other papers named above always the same database [MFTM01] is used in the training task. To the best of my knowledge there is just one work existing [Gui09] that uses an FoE prior for the inpainting of a specific type of images, namely building images. Images belonging to this group are used in the learning and the inference algorithm - as it is done in this thesis for handwritten characters. In [Gui09] the model that was trained on a specific image type did not outperform the model that was trained on natural images. It will be discussed later, whether a specialized prior overtops a general prior in the case of text reconstruction. Before, FoE inpainting of usual images will be described in the subsequent section.

3.3 Inpainting using FoE priors

The following section is divided into two parts: The inpainting of natural images will be explained in the first part, while the second part deals with the restoration of characters.

It was found that similar parameter settings may be used for the two applications. Some properties of the FoE inpainting process are better observable in the case of text restoration. Those characteristics are convergence and rotation variance, which will be detailed at the end of this section.

3.3.1 Natural Images

It is necessary to define the meaning of natural images in the context of this thesis: The term natural images describes images that contain scenes that one can find in real life. Those scenes may contain humans, animals, landscapes, plants, architectures etc. The Berkley segmentation dataset [MFTM01] is comprised of such natural images. This database is used in the work of Roth and Black [RB05b] and in the present work it is utilized for the training of priors that model natural image statistics. The database consists of 200 color images, which are used for the training of generic FoE priors.

Inpainting Characteristics

Inpainting, using FoE models, exhibits two characteristics of structural inpainting methods: On the one hand inpainting regions belonging to large and homogeneous areas are inpainted successfully. On the other hand, if the inpainted area is surrounded by textured regions, the inpainted region tends to blur out. This can be seen in Figure 3.4, where a wire that partially occludes the bull and the surrounding is removed. The inpainted regions belonging to the bull are successfully inpainted. A human observer, who has no prior knowledge of the inpainting mask, would probably primarily identify the restored grass as an inpainted region. It can also be seen that the FoE model connects strong edges (like the silhouette of the bull) and maintains curvature.

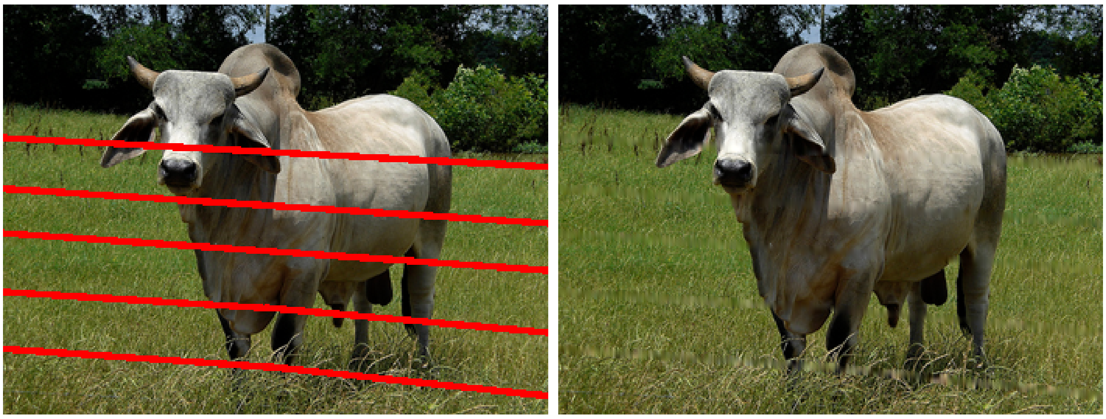


Figure 3.4: Removal of a wire. (a) Image with superimposed mask (colored red). (b) Restored image. The input image is taken from [Gre08].

However, this is only true, if the unknown region is thin and elongated. The thicker the mask gets, the lower the quality of the resulting image gets. This is illustrated in Figure 3.5: For this example an artificial mask was generated. Although the total number of unknown pixels is nearly the same as in Figure 3.4, the quality of the output is certainly lower. Both images were initialized with the same color value, but in the case of the latter one, the centers of the mask regions are not altered adequately. This stems from the fact that the FoE prior favors large and homogeneous areas. The example images show that the inpainting quality is not only dependent on the number of unknown pixels, but also on the structure of the mask - and of course on the complexity of the input image.

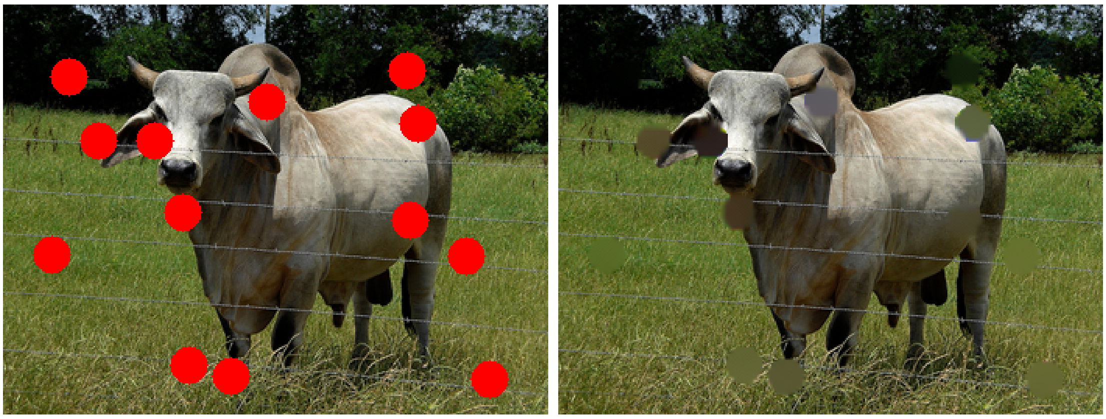


Figure 3.5: A large mask leads to an inappropriate inpainting result. (a) Inpainting problem. (b) Resulting image. The original photography is taken from [Gre08].

The choice of the inpainting step size has not been discussed yet. In most experiments a step size of 100 was used in the first 2300 iterations of the restoration task. This relatively large step size causes numerical instabilities, which become apparent in the form of artifacts. In order to get rid off those artifacts, which make the inpainted image look noisy, η is set to 1 for the last 450 iterations. This cleanup step is a reliable measure to remove the undesired artifacts. It was found that the inpainting result is not very sensitive on the step size. However, a well-founded statement about the step size influence can only be made with the help of a similarity measurement. Therefore, various step sizes will be compared in a quantitative evaluation in the result section.

Learnt filters

It is not straightforward to interpret a learnt FoE model. The structure of the learnt filters often seems to be irregular [Rot07]. In Figure 3.6 the filters of a learnt model are shown, along with their responses on the input image in Figure 3.4. The filter responses provide a more comprehensible way to interpret the learnt filters. It can be seen that the filter responses are larger at high-frequencies, compared to pixels belonging to homogeneous regions, e. g. the back of the bull. This becomes obvious if we consider the strong edges - like the silhouette of the bull - in the input image and the corresponding filter responses. The responses on such edges are diverse, which means that the filters encourage edges with different directions. The filter in the upper right corner encourages vertical edges, for instance. The student t-distribution dislikes large filter responses and favors responses close to zero. Thus, homogeneous regions become very likely for the FoE prior. This circumstance explains the poor quality of the inpainting result in Figure 3.5: Since the FoE model favors large homogeneous regions, it tends to inpaint large areas with one dominant color.

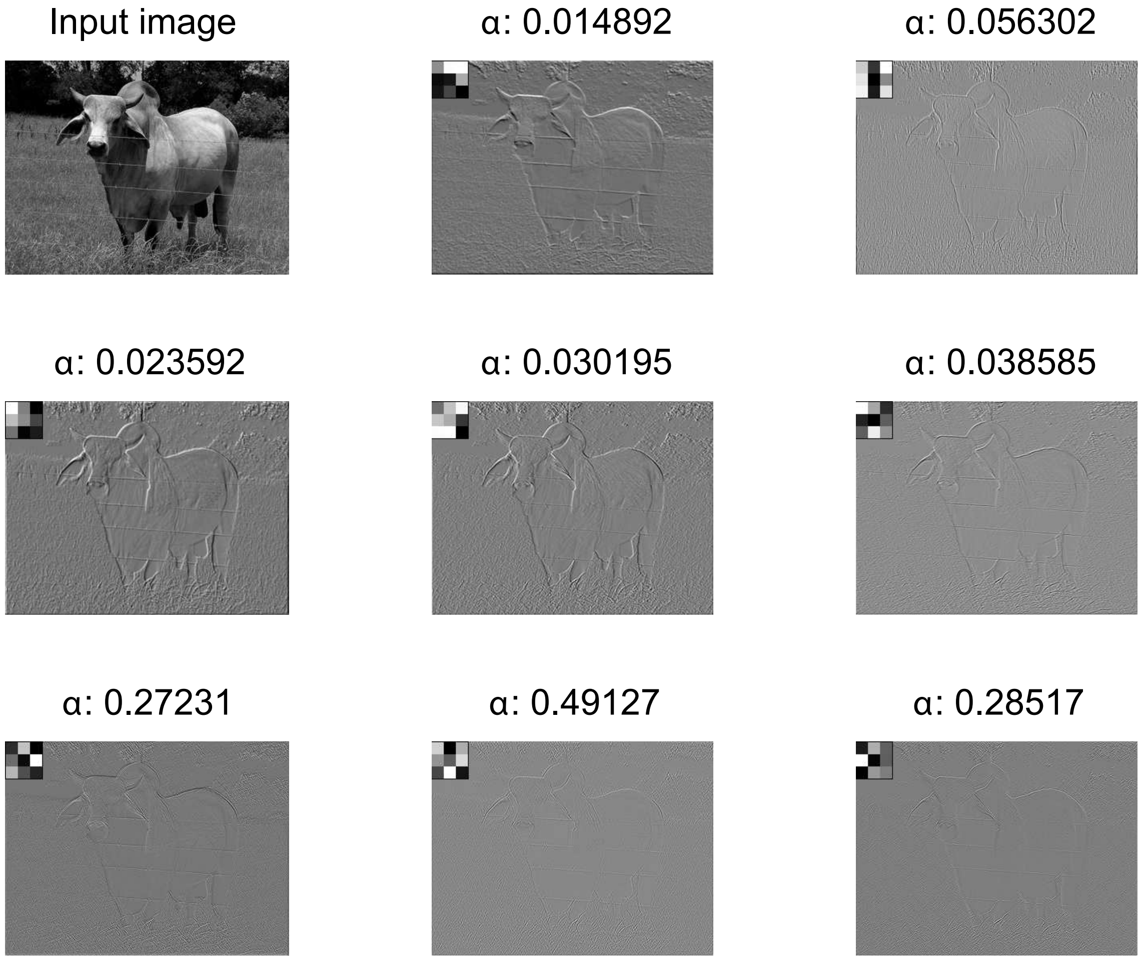


Figure 3.6: Filter responses of a learnt FoE model. The first image is the input image. The filter responses are given subsequently, along with the filters, which are shown in the upper left corner of each response image.

Initialization

Roth [Rot07] does not mention that the initialization of the inpainting area has an influence on the overall restoration quality. In the inpainting code provided by Roth, each unknown pixel is initialized with zero. During the tests conducted for this thesis, it was still observed that the inpainting result is dependent on the initialization. Small inpainting areas tend to converge to the same output, independent of the assigned gray values. The inpainting quality of larger regions on the contrary, depends on the initialization values. The restoration quality can be improved by initializing the inpainting region with a gray value that is similar to the known gray values in the surrounding of the unknown region. It was also found that less iterations are required until the inpainted image stabilizes, if the inpainting area is initialized appropriately.

Figure 3.7 shows an example for the initialization influence. In this experiment the inpainting domains are initialized with zero or the mean gray value in a band around the mask region. Two different mask sizes are evaluated. In the case of the smaller mask width, the inpainting algorithm produces the same outputs, independent of the initialization value. However, if the mask size is slightly increased, diverse inpainting results are obtained. In the depicted images it is not evident which initialization should be used overall, since both produce convincing images and implausible results as well. In the result section the two mentioned initialization types will be compared, along with other initialization values.

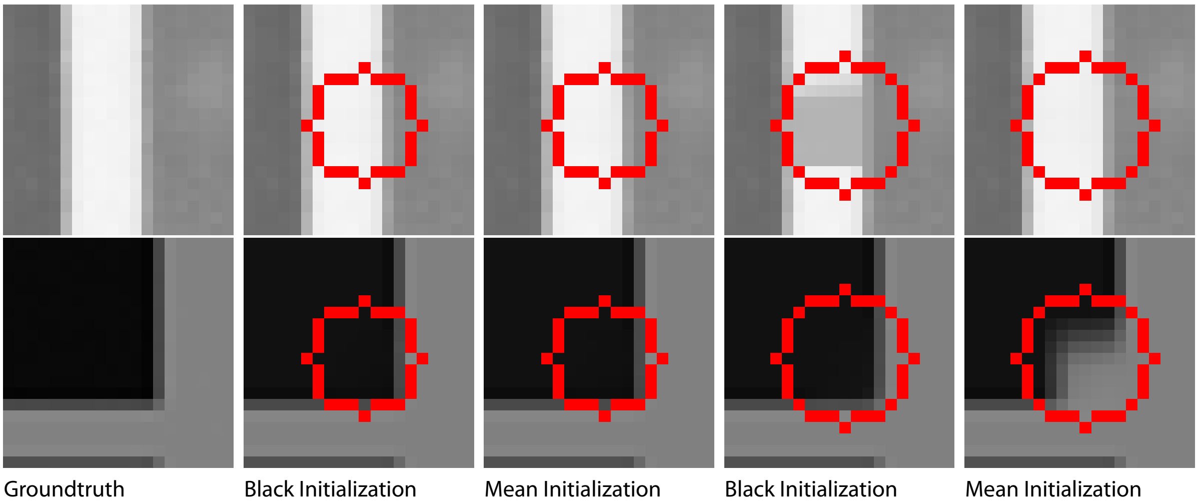


Figure 3.7: Comparison of inpainting domain initializations. In the case of mean initialization, the inpainting region is filled with the mean intensity of the pixels that are contained in a 3 pixel wide band around the unknown region. (Left column) Groundtruth images. (Second and third column) Black and mean initialization of the mask region. The mask perimeter is 11 pixels. (Fourth and fifth column) The inpainting area is again initialized with zero and the mean gray value. The unknown region has a perimeter of 13 pixels.

3.3.2 Text Restoration

In this section properties of text restoration, based on image priors, are provided. The priors that are used for text reconstruction are trained on a handwriting database, which is introduced in [MB02]. The FoE models, which are trained on handwritten text, overtop image priors that are learnt from natural image statistics. This can be seen in Figure 3.8: In Figure 3.8 (b) the output of a generic prior is given. The image in Figure 3.8 (c) is inpainted with a model that was trained on text images. It can be seen that the second prior introduces more blur than the generic model. Nevertheless, the second inpainting result is more satisfying compared to the image in Figure 3.8 (b), since the large inpainting regions, which partially occlude the characters 'u' and 'e', are inpainted with connecting strokes. The generic FoE model fills those holes with the background color.

The main disadvantage of structural inpainting approaches, compared to textural methods, is that the inpainted areas are smeared, if they belong to heterogeneous re-

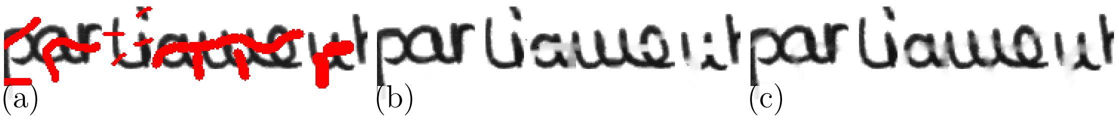


Figure 3.8: Text restoration using different FoE priors. (a) Input image with superimposed mask. (b) Image inpainted with a prior that models natural image statistics. (c) Inpainting result obtained with a model, which is trained on handwritings.

gions. This is also true in the case of the FoE approach - as it is demonstrated in Figure 3.8. It was found that the blurring is reinforced if the unknown area is surrounded by a region that is corrupted with noise. The image inpainting definition by Bertalmio et al. [BSCB00] assumes that the known image area is not damaged. This requirement is not met, when it comes to the inpainting of degraded documents - like palimpsests. Figure 3.9 shows how noise impairs the restoration quality. The noise in Figure 3.9 (c) is actually enhanced and smearing artifacts are present in the inpainting domain. This example suggests that a preceding noise removal step is necessary, if the image is corrupted with noise.

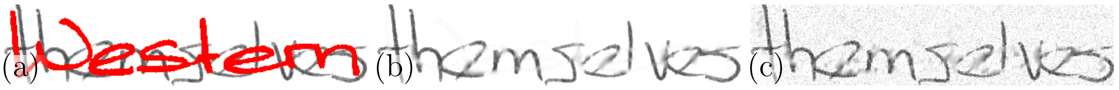


Figure 3.9: Influence of noise. (a) Noise free input image and inpainting mask. (b) Inpainted image. (c) Gaussian noise - with a variance of 0.1 - is added to the input image in (a). The noise is reinforced during the inpainting process.

The training patches are randomly extracted from an image database. In the case of text priors, the patches are also randomly chosen. Although the patches are selected randomly for both applications, the sampling strategy differs significantly in one aspect: Each patch, which is sampled from a handwriting database, must contain at least one pixel that belongs to a character stroke. Otherwise the patch is discarded. Thus, the prior learns the statistics of the characters, in place of the uniform background.

The rest of this section explains typical properties of text reconstruction. Those characteristics become also apparent if an FoE prior is used for the restoration of natural images. However, those properties are easier observable in the case of character inpainting, due to the reduced complexity of the input images.

Convergence

Several inpainting stages of a single character are shown in Figure 3.10. Additionally the respective gradients inside the inpainting regions are presented in the second row. One could expect that the gradients converge towards zero during the gradient ascent procedure, since the posterior values should reach maximas after a certain amount of time.

Figure 3.11, which shows the sum of the absolute gradient values, implies that this is not the case. It can be seen that the sum of the gradient converges to a certain value till the cleanup - with a smaller step size - is started at iteration 2301. The smaller step size causes an increase of the absolute gradient values. During the cleanup process, the

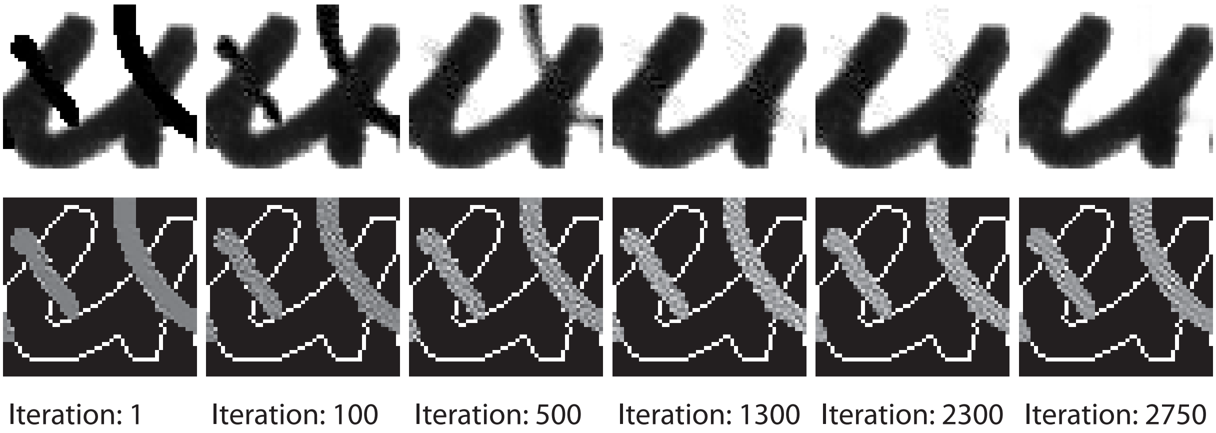


Figure 3.10: Six stages of the inpainting progress. In the upper row the inpainted images are shown. The corresponding gradients of the unknown areas are given in the lower row.

gradient sum converges also to a certain value. Although the gradient does not converge to zero, the overall inpainting result stabilizes during time, because the mean of the gradient values moves towards zero. The mean gradient value is shown in Figure 3.12. In the depicted example mainly positive gradient values are added in the first 1300 iterations, because the inpainting areas are initialized with zeros. Afterwards minor changes are made, until the cleanup process is started. The smaller step size leads to a removal of numerical instabilities, which are caused by the relatively high step size used in the first 2300 iterations. It is notable that the mean gradient value gets stabilized during the cleanup process. The main image structure is propagated in the first 1300 iterations, as it can be seen in Figure 3.10. This circumstance is also observable in Figure 3.12, where the mean intensity inside the inpainting domain, is depicted by the green line. The mean gray value stabilizes around iteration 1300, but it is still varying due to the large step size.

Rotation Variance

It is stated in [Bov05] that MRFs are inherently rotation variant. The FoE model is also rotation variant. In order to evaluate the influence of the stroke orientation, the characters in Figure 3.13 (a) are inpainted with different FoE priors. The lower image in (a) is the 270 degree rotated version of the upper image. The orientation of the upper image is typical for the character orientation in the training database. The images in Figure 3.13 (b) are inpainted with a prior that is trained on the original database. The stroke gap in the lower left corner of the upper image is inpainted successfully. If we consider the bottom image in (b), we can see that the same gap is not restored in an appropriate manner. The reason for this circumstance is the fact that the prior is learnt from a database, in which the majority of the strokes has an orientation that is similar to the long stroke in the upper row. In order to prove this claim, the training patches are rotated through 270 degrees and a new prior is trained on the rotated database. The inpainting results obtained with the modified prior are presented in Figure 3.13 (c). The rotation of the training patches leads to a decreased inpainting quality of the upper image, because the aforesaid stroke gap gets blurred out. The quality of the bottom image is in

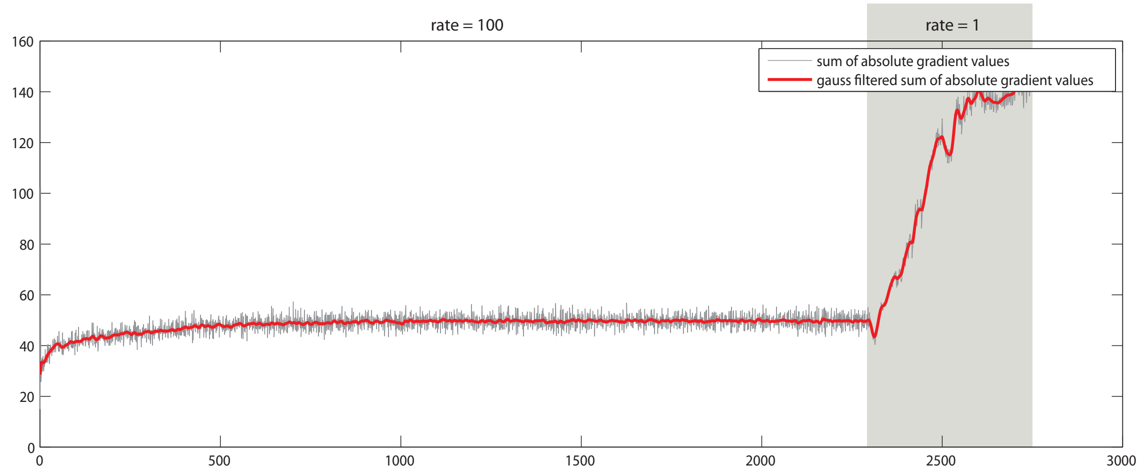


Figure 3.11: Sum of the absolute gradient values in each iteration. It is notable that the smaller step size (starting at iteration 2301) causes higher gradient values than the greater step size. The gray line shows the absolute gradient values and the red line is a Gauss filtered function of the absolute gradient. The size of the Gaussian window is 21.

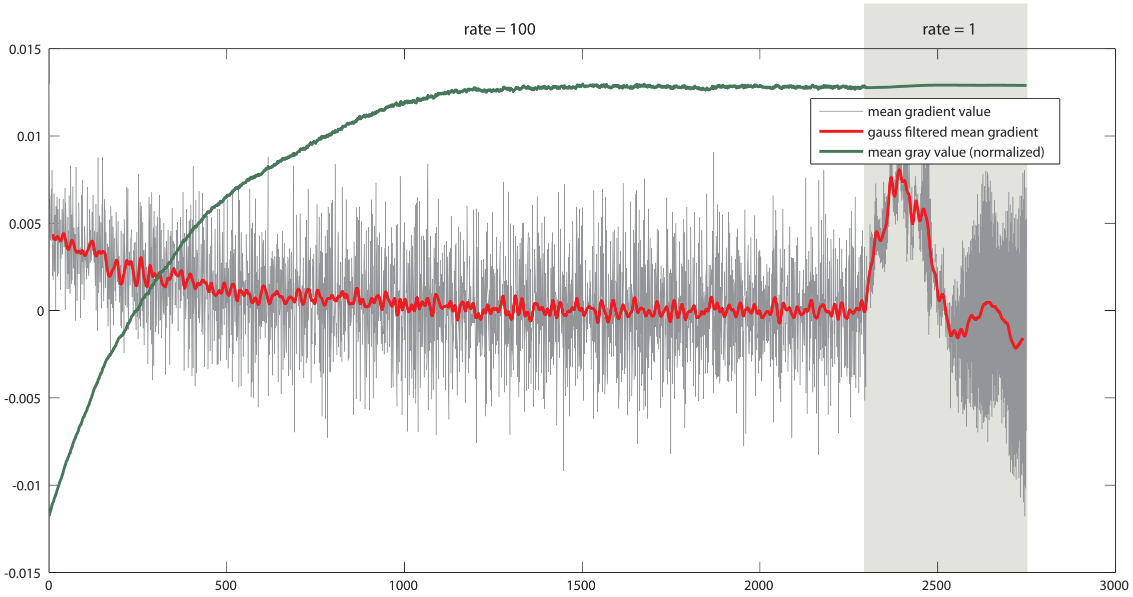


Figure 3.12: Mean of the gradient values during inpainting. As the inpainting procedure proceeds, the mean gradient value moves towards zero. Once the cleanup takes place the mean changes, because the artifacts are removed. It can be seen that the mean gray value (green line) stabilizes about iteration 1300. Due to the large step size the mean gray value is still varying, until the cleanup is started.

contrast more satisfying.

The depicted example images show that the FoE prior is rotation variant. This rotation variance becomes clearly evident if we consider the structure of the learnt filters, which are given in Figure 3.14. The structure of the majority of the filters is similar, putting aside the fact that the filters on the right side are rotated through 90 or 270 degrees. This rotation stems obviously from the rotation of the database. The expert parameters learned are also similar. The differences between the corresponding α values are small: The largest absolute difference is 0.0021 and the mean difference is 0.000588. These differences are caused by the random selection of the training patches that are used in one learning iteration.

The filters shown prove that the FoE prior learns the main orientation of the strokes, which is a major characteristic of text characters. Nevertheless, the blurred out inpainting areas shown in Figure 3.13 are certainly not desirable. For the purpose of reducing the influence of the stroke orientation, another experiment was conducted: The images from which the training patches have been extracted were rotated randomly, before the sampling was carried out. The inpainted images in Figure 3.13 (d) exhibit that this modification results in a trade-off between the satisfying inpainting of common strokes and the inadequate restoration of seldom strokes. It will be evaluated in the next chapter, if a random rotation of the training images leads to improved inpainting results.

3.4 Text Restoration based on Tensor Voting

The main goal of tensor voting is to perform perceptual grouping of elements that encode image primitives. In the case of image inpainting this perceptual grouping leads to a connection of geometric structures. Unlike the FoE approach, tensor voting is not based on image statistics. Tensor voting is instead a heuristical approach, which aims at preserving local structures. Those structures are formed by curves or points, which are called tokens. Basically each token propagates its information to nearby tokens and accumulates the information that it receives from the neighborhood. "Tokens with compatible orientations that can form salient structures reinforce each other. The support of a token for its neighbors is expressed by votes that are cast according to the Gestalt principles of proximity, co-linearity and co-curvilinearity."[MK04] (Medioni and Kang, 2000, p. 195) In this thesis only two-dimensional tensor voting is explained, but it is notable that the tensor voting framework can be extended to any dimensionality. For this reason tensor voting is also called ND tensor voting.

3.4.1 2D-Tensor Voting Framework

Tensor Encoding

A tensor encodes the saliency that he belongs to a certain perceptual structure, which may be a region or a curve. If a tensor belongs to a curve, it additionally encodes the orientation of the vector perpendicular to the curve tangent. A tensor T can be expressed as a 2×2 matrix:

$$T = \lambda_1 \hat{e}_1 \hat{e}_1^T + \lambda_2 \hat{e}_2 \hat{e}_2^T = (\lambda_1 - \lambda_2)\hat{e}_1 \hat{e}_1^T + \lambda_2(\hat{e}_1 \hat{e}_1^T + \hat{e}_2 \hat{e}_2^T) \qquad (3.20)$$

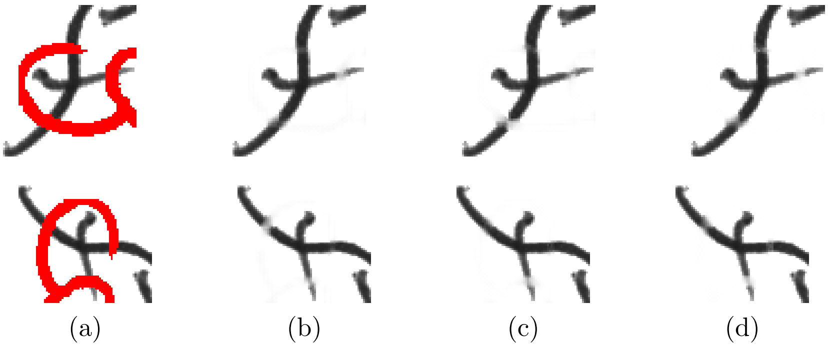


Figure 3.13: Influence of the stroke orientations in the learning database. The input images are shown in the first column, whereby the orientation of the upper image is typical for the character orientation in the training database. The lower image is the 270 degree rotated version of the upper image. (b) The images are inpainted with a model that is trained on the original dataset. The major orientation of the characters in the original database is similar to the orientation of the large stroke in the first upper image. (c) The learning patches used for the training of the prior used in (b) have been rotated through 270 degrees. The model that is trained on the rotated patches, gains a successful closing of the most gaps in the rotated lower image. (d) The utilized FoE prior is trained on patches that were sampled from randomly rotated images.

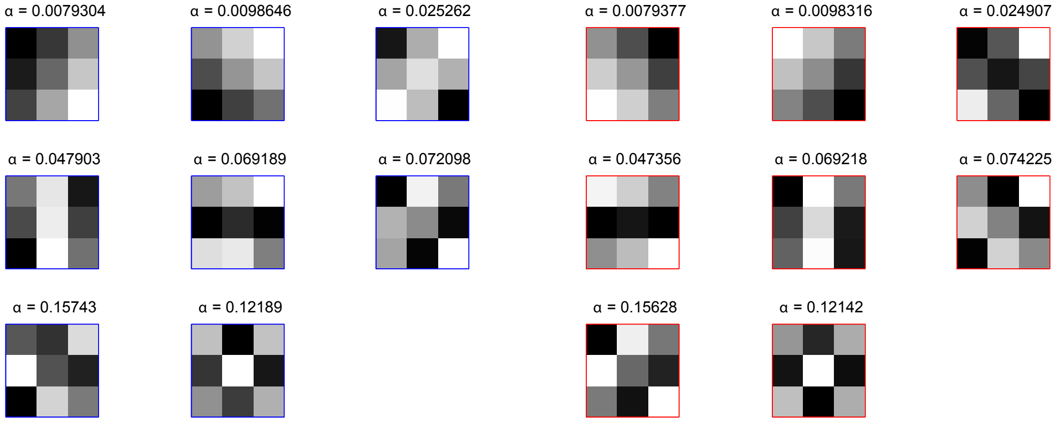


Figure 3.14: The character orientation affects the orientation of the learnt filters. The shown filters are used for the inpainting of the images in Figure 3.13 (b) and (c). The filters on the left side (bordered blue) are learnt on training patches that are randomly extracted from the database. Those training patches have been rotated through 270 degrees and were used for the training of the filters on the right (bordered red).

where $\lambda_1 \geq \lambda_2 \geq 0$ are the eigenvalues and $\hat{e}_1$ and $\hat{e}_2$ are the corresponding eigenvectors. The first term of Equation 3.20 may be interpreted as a stretched ellipsoid, which encodes the orientation of the curve normal. This tensor is named the stick tensor. The second part of Equation 3.20 describes a circle, which is called the ball tensor. A ball tensor has no clear orientation. The size of a tensor indicates the saliency about the geometric structure. In the encoding step the perceptual structure is known and therefore each tensor has a size of 1. Table 3.1 illustrates how the tensors are initialized.

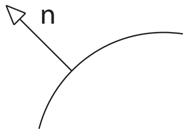
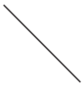
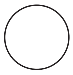
Geometric structure	Tensor visualization	Eigenvalues	Tensor
 Curve		$\lambda_1 = 1, \lambda_2 = 0$	$\begin{bmatrix} n_1^2 & n_1 n_2 \\ n_1 n_2 & n_2^2 \end{bmatrix}$
 Unorientated		$\lambda_1 = \lambda_2 = 1$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

Table 3.1: Tensor encoding of two-dimensional structures

Tensor voting

Once the tensors are encoded, tensor voting takes place. In this process each token refines its information contained by receiving the information, which is casted by its local neighbors. In the following the voting scheme for stick tensors is explained. Voting of ball tensors is not discussed here, since the implemented inpainting algorithm makes only use of stick tensor voting. In the original tensor voting framework ball tensors are useful if a structure is damaged and its orientation is unknown. Although ball tensors might help to restore a degraded structure, they are not encoded, because the known image area is assumed to be undamaged.

The basic idea of stick tensor voting is illustrated in Figure 3.15. Suppose that we are searching for the most likely curve that connects the origin O and the point P. Medioni and Kang claim that "the arc of the osculating circle (the circle that shares the same normal as a curve at the given point) at O that goes through P is the most likely smooth path, since it maintains constant curvature." [MK04] (Medioni and Kang, 2000, p. 205) In Figure 3.15 the center of the osculating circle is the point C. The vote that is casted by O to P is also a stick tensor.

Since every tensor in the neighborhood of P casts a vote to the token in P, it is important to weight the votes. The weight of a vote is inversely proportional to the

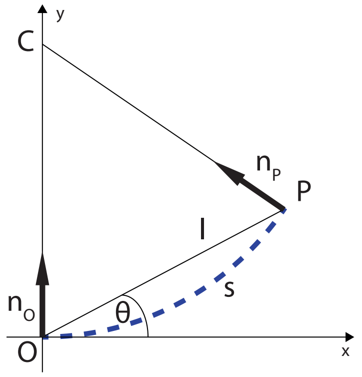


Figure 3.15: The stick tensor in O casts a vote to the tensor in P.

length of the connection. Thus, nearby tokens have a greater influence than faraway tokens. The weighting function suggested by Medioni and Kang is named saliency decay function and is defined by:

$$DF(s, \kappa, \sigma) = e^{-}(\frac{s^2 + c\kappa^2}{\sigma^2}) \tag{3.21}$$

where s is the arc length, κ the corresponding curvature and σ defines the scale of voting. σ is the only parameter that is chosen by the user and controls the neighborhood size and the smoothness of the connecting curve. A large σ causes a higher degree of smoothness, which leads to the grouping of compatible structures even if s is relatively great. Hence, the tensor voting approach is able to restore strokes that are interrupted by relatively large gaps. A small σ favors the votes of tokens in the local neighborhood and thereby preserves local structures. c is a constant that depends on σ:

$$c = \frac{-16\log(0.1)(\sigma - 1)}{\pi^2} \tag{3.22}$$

Since the saliency decay function has now been defined, we can move on with the definition of the stick vote. The vote that is casted from a stick tensor located at point O to a tensor located at P is defined by:

$$\begin{gathered}\mathbf{S}(l, \theta) = DF(s, \kappa, \sigma) \begin{bmatrix} -\sin(2\theta) \\ cos(2\theta) \end{bmatrix} [-\sin(2\theta)\cos(2\theta)] \\ \kappa = \frac{2\sin(\theta)}{l}, \qquad s = \frac{\theta l}{\sin(\theta)}\end{gathered} \tag{3.23}$$

where l is the distance between O and P and θ is the angle enclosed by the vector $\vec{OP}$ and the tangent of the curve in O (see Figure 3.15). The votes that are casted by O to each token in the neighborhood are also stick tensors. These votes build up the voting field of O. The field is restricted by the constraint that the enclosed angle θ is less or equal than 45 degrees.

In Figure 3.16 two partially overlapping voting fields are shown. It is obvious that the size of the tensors decreases with the distance to the voter, which stems from the explained saliency decay function. Each tensor accumulates the votes by simply adding it to itself. This process is called tensor addition. If a stick tensor receives various stick votes with parallel curve normals its saliency increases. If a stick tensor receives in contrast votes with varying normal vectors its saliency decreases. An extreme example for this circumstance is a tensor that receives two votes with perpendicular vectors. In this case the receiving tensor is transformed into a ball tensor, which means that this tensor has no preferable orientation.

3.4.2 Inpainting Algorithm

In the previous section the general 2D-Tensor voting framework was discussed. The result of the voting process is a field of tensors which encode orientated or un-oriented structures. It has not been explained yet how features are extracted from the final tensor field. This feature extraction step and the overall inpainting algorithm will be explained in the following.

Tensor Encoding and Voting

Before the tensor encoding may take place a few preprocessing steps are necessary. Those steps are shown in Figure 3.17. In the first preprocessing step the unknown pixels are initialized with the background color (see Figure 3.17 (b)). In the second step the horizontal and vertical image gradients of the image are computed with a first order edge detector - namely the Canny [Can86] edge detector. In Figure 3.17 (c) the corresponding edge image is shown. The image gradients that exceed a certain threshold serve as the curve normals in the tensor encoding step. However, before the encoding may be conducted a further preprocessing step is required: Since false edges are introduced at the mask border, the corresponding gradients must be deleted. The unwanted edges are caused by character strokes that immediately end at the mask border. In the removal step (illustrated in

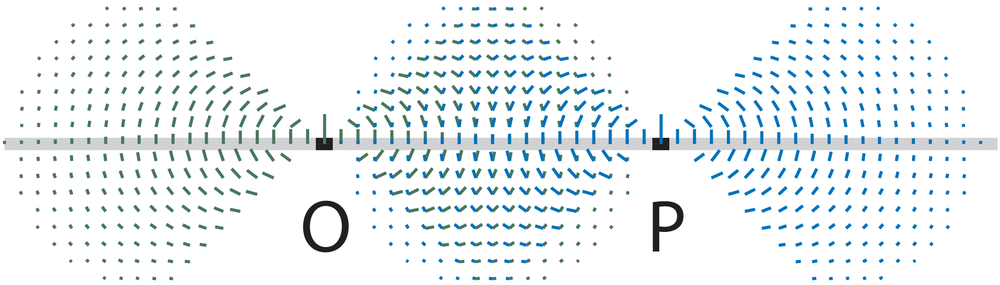


Figure 3.16: Voting fields of two tensors located at O and P. The tensors belong to the same structure, namely the gray line that should be restored. A tensor that receives votes by O and P accumulates the votes by simply summing them up. It is notable that from the subset of tensors, which receive two votes, only tensors located at the straight line are still stick tensors after the voting process. Hence the connecting line is the most likely connection.

Figure 3.17 (d)) the mask is slightly dilated and the gradient values inside the dilated mask are set to zero. The remaining image gradients are afterwards encoded into tensors. The points, where the gradients do not exceed a global threshold, are initialized with zero matrices.

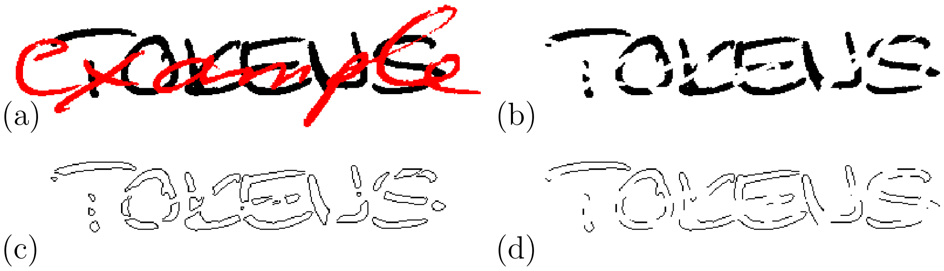
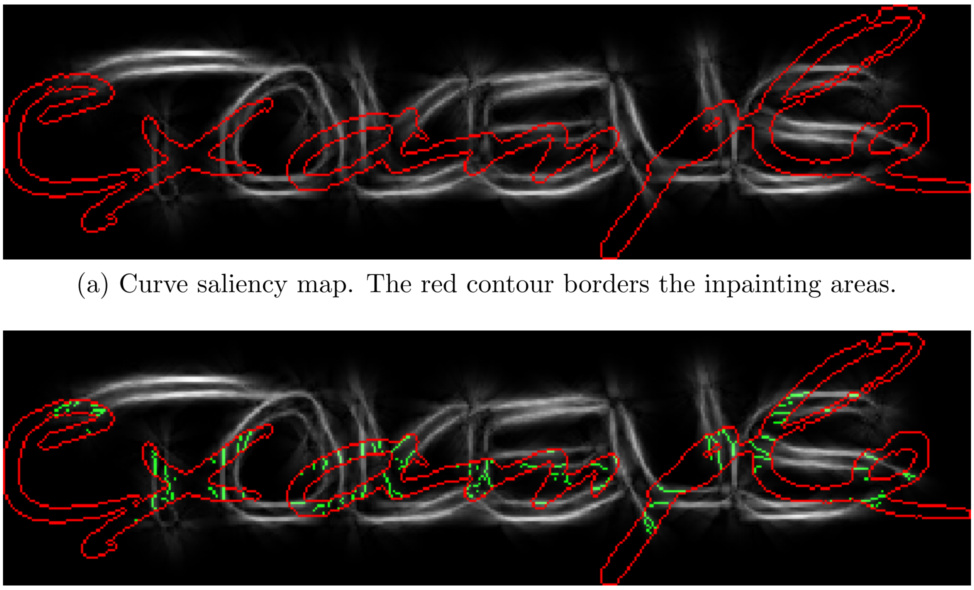


Figure 3.17: Necessary preprocessing steps before tensor encoding. (a) Input image with overlapping mask. (b) Initialization of the inpainting region. (c) Result of edge detection. (d) A dilation of the mask removes the unwanted edges.

Once the stick tensors are encoded, tensor voting takes place. The voting procedure was already discussed in Section 3.4.1. However, the voting formula in Equation 3.23 is only defined for a voter, whose curve tangent is a horizontal vector. If a curve tangent has another orientation, the corresponding voting field must be rotated. The interested reader is referred to [MK04] for an explanation of this rotation.

Feature Extraction

Once the voting process is finished, the restored image is encoded as a field of tensors. To extract the saliency and the normal vector from a tensor, the corresponding eigenvalues $\lambda_1 \geq \lambda_2 \geq 0$ and eigenvectors $\hat{e}_1$ and $\hat{e}_2$ are calculated. The eigenvalues are needed for the construction of a curve saliency map. This map encodes the saliency that an actual tensor is a stick tensor, which is defined by $\lambda_1 - \lambda_2$. Figure 3.18 (a) shows the saliency map for the input image in Figure 3.17 (a). In the illustrated example it is obvious that tensors belonging to large linear structures reinforce themselves, which results in a high saliency. In the inpainting regions the saliency is generally smaller than in the known image regions, since those areas are initialized with zero matrices. This circumstance is unapparent in the shown saliency map, because the saliency inside the known image region is normalized in order to provide a better visualization.



(a) Curve saliency map. The red contour borders the inpainting areas.

(b) Result of feature extraction. The green marked pixels are incorporated into the edge image.

Figure 3.18: The curve saliency map in (a) is used in the edge detection step illustrated in (b).

Before the edge extraction is started, unlikely tensors are eliminated with a simple thresholding technique: A tensor is declared as unlikely, if the following condition is met:

$$\lambda_1 - \lambda_2 \leq t, \tag{3.24}$$

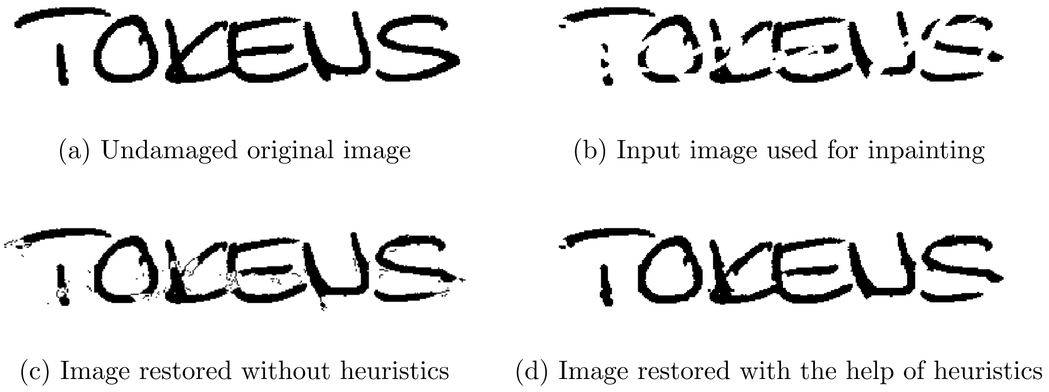
where t is a user defined threshold. The curve saliency map serves as the basis for the extraction of the edges inside the inpainting area. It can be seen in Figure 3.18 (a) that the connecting edges assume local maximas. A simple thresholding method is insufficient for the edge extraction, since the saliency of local non-maximas at the mask border is higher than the saliency of more distant local maximas. Therefore the non-maximum suppression method that is introduced in [Can86] is applied. In the Canny edge detection algorithm this method is used for the detection of local maximas in the direction of the image gradient. It is straightforward to adapt the non-maximum suppression method to our needs, since the eigenvector $\hat{e}_1$ of a tensor describes the image gradient. In Figure 3.18 (b) the local maximas of the saliency map are marked green.

Region Filling and Postprecessing

Before region filling takes place the edge image is post processed. The post processing is necessary since edges are partially interrupted, which stems mainly from great distances between the voting and the receiving tensors. Small interruptions are eliminated by bridging unconnected pixels. This means that a pixel is incorporated into the edge image, if it has two not connected neighboring edge pixels.

Each pixel in the restored edge image is then set to the foreground text color, if it belongs to an edge, or otherwise to the background color. The unknown regions of the input image are afterwards replaced with the adequate parts of the edge image. Now holes inside the inpainting areas are filled with the foreground color. A hole is a region that is enclosed by a set of 8 connected foreground pixels. Although the character strokes are filled at this time, there is still a need for a further manipulation: Usually there are edges in the filled image that do not enclose a hole. To get rid of those edges, a pixel is set to the background color if there are only 4 or less foreground pixels in its 9×9 neighborhood. Finally all inpainted foreground pixels that are not connected to the known foreground are set to the background color.

The proposed post processing steps are heuristics, since they do not rely on the information encoded in the final tensor field. Nevertheless the use of those methods leads to an improved result, as it can be seen on the final inpainting results provided in Figure 3.19. The restored text in Figure 3.19 (c) is obtained without using the discussed post processing steps, while the inpainted image in Figure 3.19 (d) is created with the help of the heuristics. Apparently the elimination of isolated edges erases mainly visual implausible edges. Those unlikely edges are caused by votes casted by tensors belonging to large, linear structures that do not immediately touch the inpainting mask. While the erasing of such edges mainly acts as a noise removal step, the bridging of the edge image may introduce noise in the resulting image. Although this noise might introduce not existing strokes, the bridging step is useful, because it helps to wipe out small interruptions in existing structures. In the example image presented especially the character 'K' benefits from the bridging step. It is notable that the connection of unconnected neighboring pixels results partially in thickened strokes. In the post processed image in Figure 3.19 the characters 'N' and 'S' contain such thickened strokes.



(a) Undamaged original image

(b) Input image used for inpainting

(c) Image restored without heuristics

(d) Image restored with the help of heuristics

Figure 3.19: Final inpainting results.

3.5 Summary

In this chapter the principles of the methods, which have been investigated, were explained. First it was depicted how the overwritten palimpsest regions are detected. The detection of these areas is fulfilled on tungsten illuminated parchments, which exhibit mainly the overwritings. The binarization algorithm used relies on a contrast image, which encodes the difference between the maximum and minimum intensity value within a local neighborhood. Secondly, the concept of the FoE algorithm was outlined. The underlying MRF theory was briefly introduced and the learning and inference equations, which are used in the FoE framework, were explained. Characteristics of the inpainting technique were illustrated using example results that were gained on natural images and images containing handwritings. Finally the heuristical restoration method was depicted. This method is based on tensor voting, which performs a grouping of elements. The general two-dimensional tensor voting framework was explained and the additional steps, which are part of the inpainting algorithm, were described.

Chapter 4

Results

This chapter provides an evaluation of the methods that were depicted in the last section. The inpainting of natural images will be analyzed as well as the restoration of handwritings. Since the tensor voting algorithm is restricted to the recovery of high contrast letters, the majority of the following experiments shows characteristics of the FoE inpainting algorithm.

The chapter is structured as follows: In Section 4.1 the inpainting of natural images is evaluated. The first test shows how the performance of the FoE approach is affected by the inpainting parameters. Additionally the influence of the mask extent is analyzed. In the subsequent experiment the FoE algorithm is applied to the recovery of masks, which are generated from salt & pepper noise. The section is concluded by a comparison to other inpainting techniques.

In Section 4.2 the recovery of synthetic handwriting data is analyzed. The section starts with an analysis of the statistically based method. First it is shown how the training data affects the text restoration performance of the FoE algorithm. Therefore, the performances of four priors, which were trained on different data sets, will be compared. Furthermore, a parameter evaluation of the statistical approach is given by means of two experiments. In Section 4.2.2 the parameter dependency of the text recovery based on tensor voting is analyzed. The performances of both restoration algorithms are compared in Section 4.2.3. Therein the strengths and weaknesses of the investigated systems are depicted and special attention is paid to the question, which occlusions can be restored in a satisfying manner.

The recovery of historical underwritings is evaluated in Section 4.3. A ground truth set was created manually in order to provide an assessment of the inpainting results. At first it is argued, why it was necessary to shrink the utilized images. The parameter dependency of the mask extraction step and the inpainting sequence is evaluated. The proposed method suffers from various drawbacks, which will be explained in the course of the section. Finally the outputs of a prior trained on synthetic data are contrasted to restorations produced by a prior, which was trained on underwritings.

Similarity Measurement

In the subsequent experiments the PSNR is used to measure the similarity between an inpainted image I' and the corresponding ground truth image I. The PSNR is defined as:

$$PSNR = 20log_{10}\left(\frac{255}{\sqrt{MSE}}\right), \tag{4.1}$$

where the Mean Squared Error (MSE) for two equally sized images I and I' is given by:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - I'(i,j)]^2 \tag{4.2}$$

A PSNR value is given in decibels (dB) and a high PSNR value indicates a high restoration quality. It is noteworthy that the MSE and PSNR are not sophisticated measures of the human perception of similarity and that no other similarity measurement has yet been established for image inpainting [BLLC02]. This explains why the majority of the papers, introduced in Section 2, do not make use of any error measurement. Only a minority of the introduced papers[1] provide PSNR values to measure the inpainting quality. Nevertheless, the PSNR is utilized in this section, because it allows for a statistical evaluation of multiple inpainting results. The numerical results are presented along with selected inpainted images, in order to allow for a subjective assessment by the reader.

The aforementioned papers provide PSNR values of entire images, regardless of the amount of unknown pixels. In order to enable comparability to this works, this notation kind is also used in the following section, which deals with the recovery of natural images. The masks that are used in these experiments have similar ratios between known and unknown pixels and the PSNR values are not biased by varying mask structures or sizes.

In the experiments, which are conducted on handwritings, the mask areas are in contrast varying. Therefore, it is necessary to provide the PSNR values inside the inpainting domains, because otherwise the results for different handwritings are not comparable.

4.1 Experiments on Natural Images

The following two tests are conducted on the Miscellaneous volume of the USC-SIPI image database [Web97]. The volume consists of 44 images, which are frequently used in image processing - like Lena, mandrill monkey, etc. The volume contains 16 color images, which have been converted to grayscale images in order to reduce the computational burden. Additionally four large pictures - with a size of 1024 × 1024 px - have been down sized by a factor of two. The modified data set contains 14 images with a size of 256 × 256 px and 30 images with a size of 512 × 512 px.

4.1.1 Parameter Evaluation

The first experiment demonstrates how the inpainting rate η and the extent of the inpainting mask influence the restoration quality. The utilized masks can be categorized

[1] Those papers are namely: [YSM10], [Gul06], [FSM09], [ESQD05], [Rot07], [Gui09] and [BLC08].

into three domain types: Hand drawn strokes, circles and text masks. Common to all domains is the ratio between the unknown and known pixels: Each inpainting mask occludes about 25 % of the corresponding input image. Two different masks have been generated for each domain type. This was necessary, because the dataset is comprised of images with two different sizes. Masks belonging to the same group have similar attributes, e.g. the same font size or diameter, regardless of the mask size.

14 different inpainting rates - ranging from 20 to 140 - have been evaluated and the gained mean PSNR values are provided in Figure 4.1. For the moment only the black initialization is discussed, while the mean initialization will be evaluated later on.

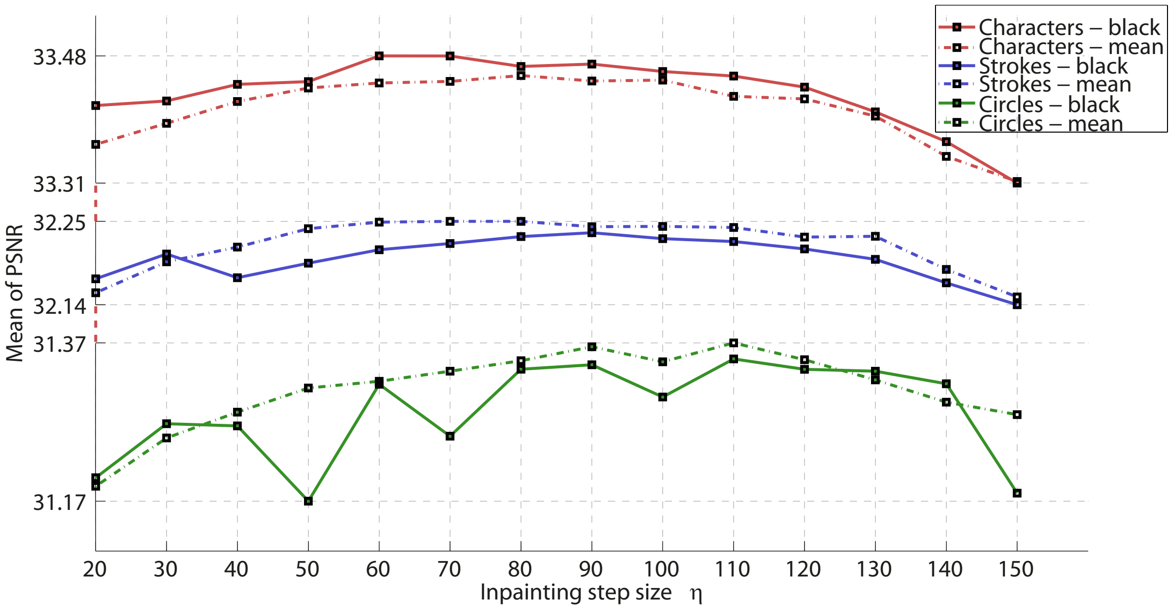


Figure 4.1: Evaluation of mask types and inpainting rates η. The y-axis, which shows the mean PSNR values, is partially shrinked in order to provide a detailed representation for each mask type. The dotted red lines on the y-axis mark the decreased ranges.

Although the percentage of the occluded area is nearly the same for each mask type, the numerical results gained are varying: The maximum PSNR values of the text, strokes and circles masks are 33.48 dB, 32.23 dB and 31.35 dB. A qualitative analysis by the author confirmed that the recoveries of the text masks look most pleading, while the restorations of the circles domains are less convincing than the retouchings of the remaining mask types. This is attributed to the fact that the widths of the distinct mask regions are varying: The most character lines are 3 pixels wide, a stroke has a width of 5 pixels and the diameter of a circle is 11 pixels. It can be concluded that the width of the mask regions has a direct impact on the inpainting quality. The mask lengths have less influence on the restoration, since the stroke lengths are larger than the circle diameters. This behavior is common to most geometric inpainting methods, which “are known to work well with ’narrow’ inpainting domains.” [CK06] (Chan and Kang, 2006, p. 87)

The width of the distinct inpainting regions must be considered when it comes to the selection of the inpainting rate η. In the case of the narrow text characters the best result

- 33.48 dB - is achieved if η is set to 60. It turned out that local structures are preserved by this inpainting rate. If a higher rate is applied, fine structures are more likely to get vanished, because of the influence of nearby homogeneous regions.

An example for the text mask recovery is shown in Figure 4.2. The input image and the inpainting domain are shown on left. The image on the right is obtained by setting η to 60. The PSNR value of the restored image is 34.88 dB. A lower similarity, namely 34.85 dB is gained, if η is set 110. The differences between both images become only apparent at a closer look. Therefore, a detail view of the yellow bordered square in Figure 4.2 (left) is given in Figure 4.3. The image in Figure 4.2 (b) is restored with $\eta = 60$, whereas the image in Figure 4.2 (c) is inpainted with $\eta = 110$. The edges at the top of the jet fighter are better preserved by the smaller inpainting rate. However, the images exhibit only minor differences, which shows that the inpainting rate has hardly any influence on the restoration of small mask regions.

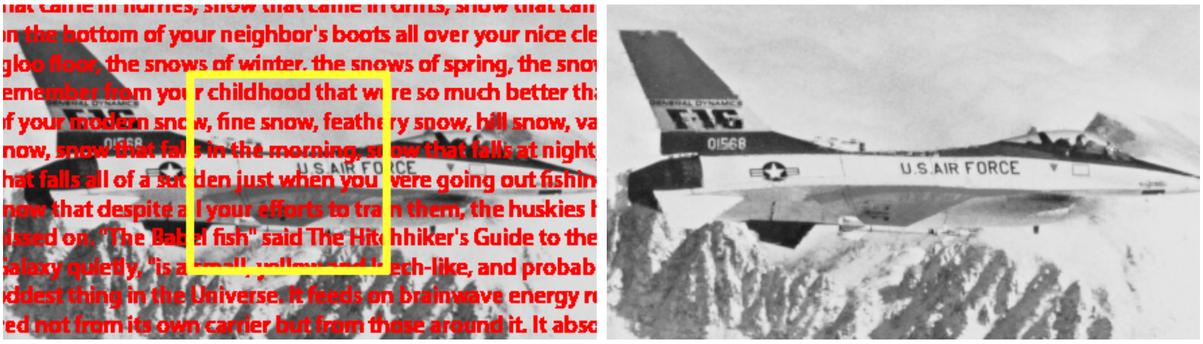


Figure 4.2: Recovery of a text mask, shown left. The image on the right is restored, by using an inpainting rate of 60.

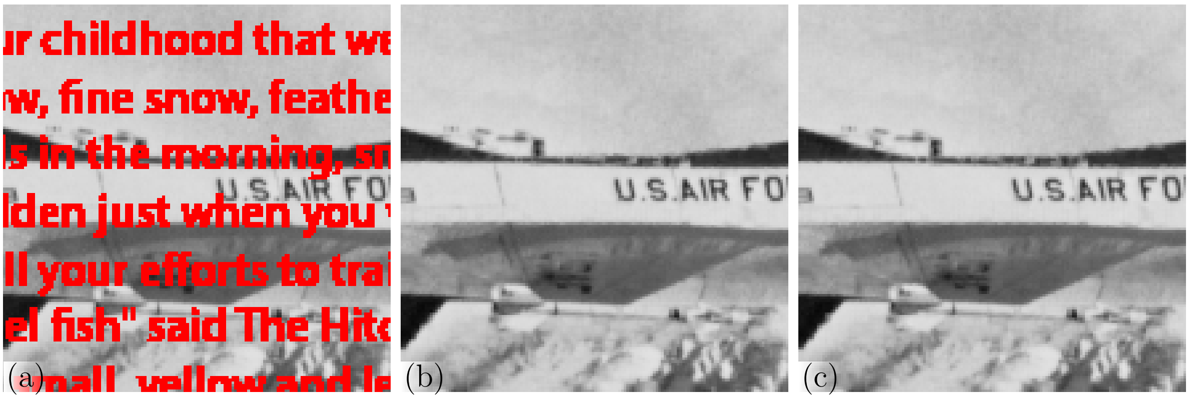


Figure 4.3: Detail view. (a) Ground truth image. (b) Result of $\eta = 60$. (c) Result of $\eta = 110$.

It turned out that the inpainting rate has also a minor effect on the restoration quality of the strokes masks. The highest similarity is gained, if η is set to 90. The corresponding mean PSNR value is 32.23 dB. The inpainting rate has to be enlarged, compared to the text masks, in order to ensure propagation into the mask centers. Otherwise the unknown regions are not altered sufficiently.

However, the differences are again only recognizable, if very different rates are applied. One example for the strokes mask recovery is given in Figure 4.4, where the right image is restored with $\eta = 90$. The similarity to the ground truth image is 39.16 dB. A considerably lower performance, namely 38.09 dB, is gained, if η is set to 30. Portions of the inpainted images are shown in Figure 4.5. The image in Figure 4.5 (b) is retouched, using a rate of 30, whereas the photography in Figure 4.5 (c) is inpainted with $\eta = 90$. The image, produced with the lower rate, looks visually implausible, since regions, which are located at the boundary of the milk, are partially filled with the background color.

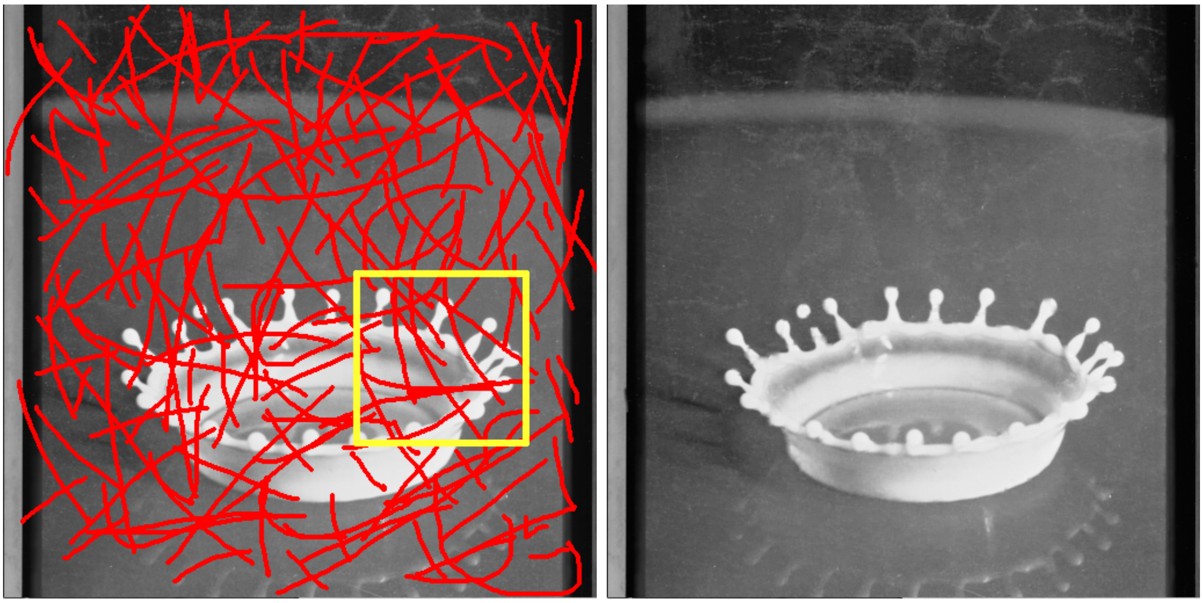


Figure 4.4: Restoration of the strokes mask. The utilized inpainting rate is 90.

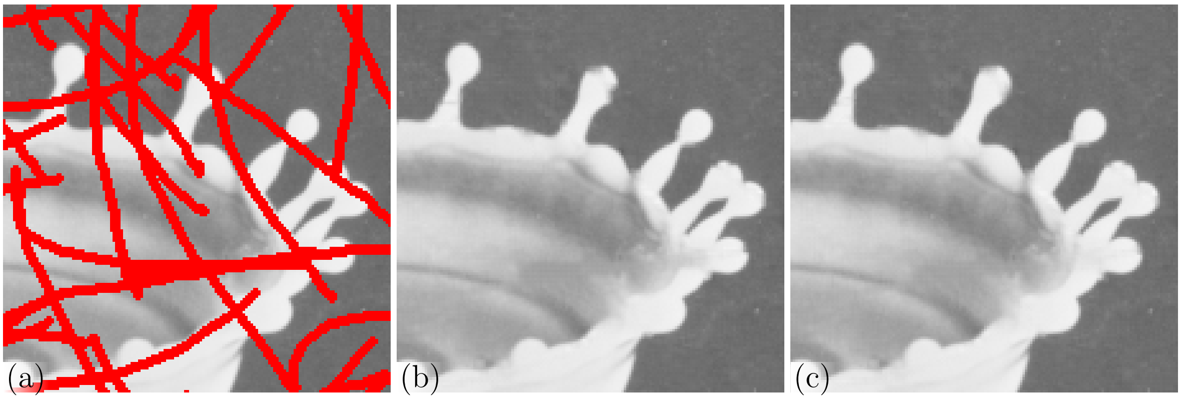


Figure 4.5: Detail view. (a) Inpainting problem. (b) Result of $\eta = 30$. (c) Result of $\eta = 90$.

The circles masks are the most challenging inpainting domains in this experiment. The highest mean PSNR value - 31.35 dB - is gained, if η is set to 110. This relatively large rate has to be applied, in order to accomplish an alteration of the mask centers. However, if a mask region is surrounded by heterogeneous regions, the inner pixels are oversmoothed and the inpainting process is evident. An extreme example for the weak performance on the circles masks is shown in Figure 4.6. The image exhibits that the inpainting algorithm is incapable of reproducing texture in a satisfying manner, as it is also noted in [Rot07].

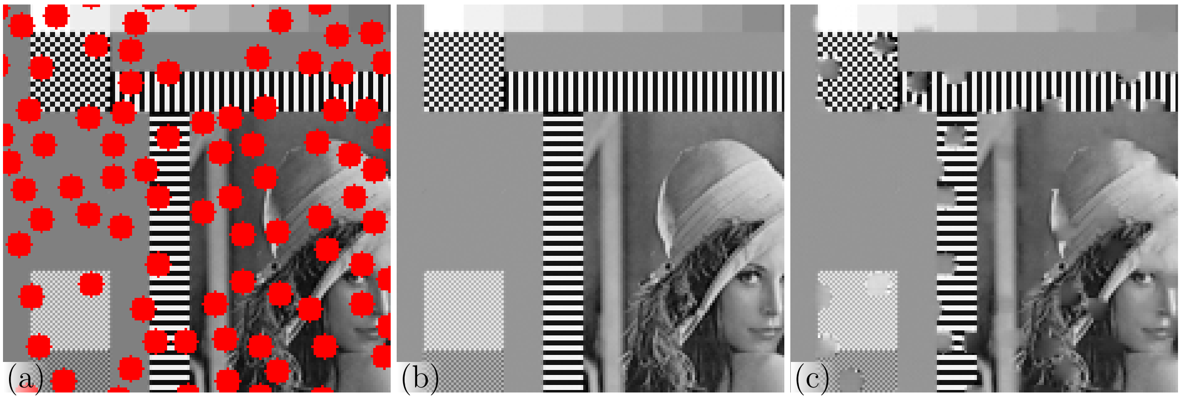


Figure 4.6: Limitation of the inpainting algorithm. (a) Input image. (b) Ground truth image. (c) Image inpainted with $\eta = 110$. The PSNR value of the inpainting result is low, namely 24.93 dB.

Initialization Dependency

It has not been discussed yet, whether the mean initialization improves the inpainting performance. The term mean initialization describes the initialization with the mean intensity of the known pixels that are contained in a 3 pixel wide band around a distinct inpainting region. The PSNR values that are gained with this initialization kind are given in the already presented Figure 4.1. In order to provide a more comprehensible comparison of the two initialization kinds, the highest PSNR values - regardless of the inpainting rate - are given in Table 4.1.

Mask type	Text	Stroke	Circle
Black initialization:			
Optimal rate	60	90	110
PSNR	33.48	32.23	31.35
Mean initialization:			
Optimal rate	80	80	110
PSNR	33.45	32.25	31.37

Table 4.1: Influence of the initialization value.

It can be seen that the text masks are better restored, if the unknown regions are initialized with zero. The PSNR values, obtained with a black initialization, are greater for each examined rate, compared to the mean initialization. On the contrary, the inpainting quality of the other mask groups is most of the times higher, if the unknown areas are initialized with mean gray values. In the case of the stroke masks, only for $\eta = 20$ and $\eta = 30$ a higher similarity is achieved with a zero initialization.

The recovery of the circle masks benefits also most of the times from the mean initialization. Only four inpainting rates - namely 20, 30, 130 and 140 - gain a greater PSNR value with the black initialization. It can be seen in Figure 4.1 that the graph, which depicts the black initialization (solid green line) is more varying than the graph, which shows the mean initialization (dashed green line). The mean initialization enhances therefore slightly the restoration result, but the results for the circles masks are in general modest, as it was previously shown.

If η is set to 110 the highest PSNR values for the circles masks are gained by both initializations. This rate was used for the restoration of the image that is shown in Figure 4.7. The inpainting regions of the image shown in Figure 4.7 (b) were initialized with zeros. The result of the mean initialization is given in Figure 4.7 (c). The corresponding detail images are given in Figure 4.8. It can be seen that the input image in Figure 4.8 (a) contains 3 strong, vertical edges, which are better restored by the mean initialization. This effect is also indicated by the higher PSNR value of the image that was initialized with mean values, but it has to be mentioned that both restorations look visually implausible, since blurred out areas are present in both images.

The plot in Figure 4.1 exhibits that the initialization value has a certain effect on the inpainting quality. However, the investigated mean initialization is not a guarantor for improved results, since the results for the character masks are worsened by the mean initialization. The quality of the restoration of the larger masks is slightly enhanced, but the investigated initialization kind is not a generally applicable extension of the FoE inpainting algorithm.

4.1.2 Removal of Salt & Pepper Noise

In this experiment, the inpainting of salt & pepper noise masks is analyzed. The removal of salt & pepper noise is usually accomplished by image denoising algorithms - like [Rot07] and [Tsc06] -, which make no assumption about the location of the perturbation. This is contrary to the conducted experiment, where the position of the noise is encoded in the inpainting domain. Therefore, this experiment is artificial, because in practical applications the location of the disturbance is unknown. Masnou [Mas02] has suggested to identify damaged pixels with a so-called grain filter and to generate an inpainting mask from the found noise locations. This approach is not investigated here, but it is remarkable that image inpainting can be used for noise removal.

The domains that are used in this experiment have been generated from salt & pepper noise and occlude 40%, 60% and 80% of the input images. The input images that were used in the preceding experiment are also utilized in actual experiment. The images were restored by setting η to 70. The influence of the initialization value is again analyzed, but this time different initialization values are used. The mean initialization is not investigated, since a dilated mask would cover the majority of the corresponding input image.

Instead, the inpainting regions are initialized with statistical measures that are computed from the entire known image. The investigated measures are the mean, the median and the Otsu threshold of the known pixels. The mean PSNR values that are achieved by the different initialization values are provided in Table 4.2. It can be seen that relatively

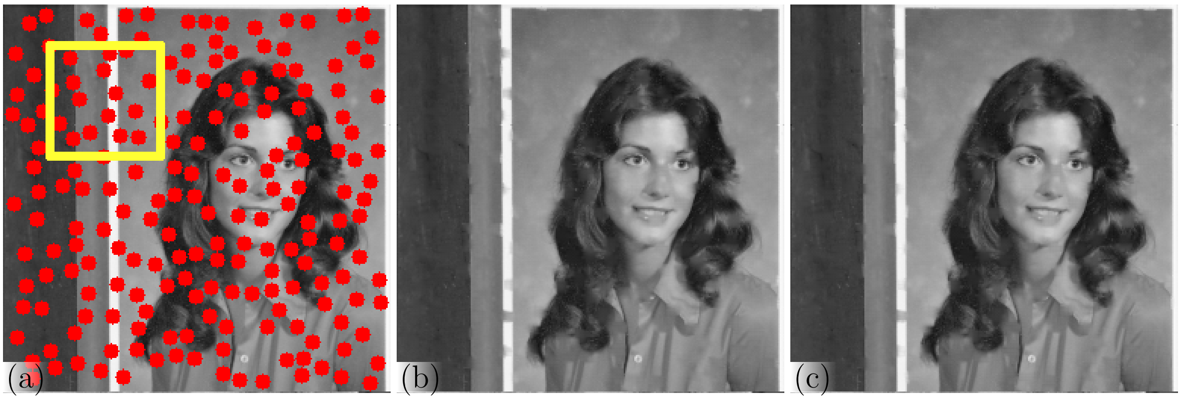


Figure 4.7: Inpainting of the circle mask, shown in (a). (b) Black initialization result: PSNR = 33.54 dB. (c) Mean initialization result: PSNR = 34.30 dB.

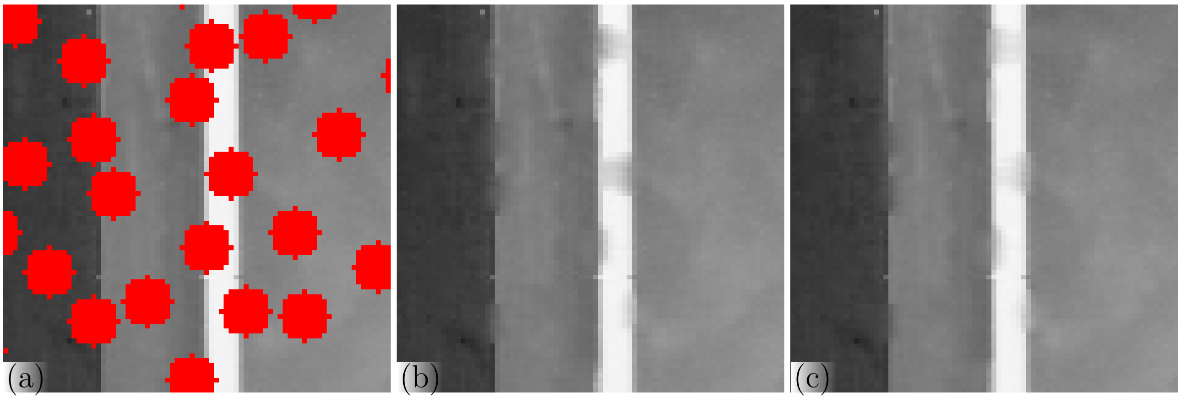


Figure 4.8: Detail view. (a) Image with superimposed mask. (b) Result of the black initialization. (c) Result of the mean initialization.

similar performances are gained. This indicates that the restored regions converged to similar intensity values, independent of the initial values.

The stable convergence is attributed to the irregular structure of the masks. In the masks utilized the distances from the unknown pixels to the closest known pixels, are short, compared to the previously used masks. Even if 80% of the pixels are unknown, the maximal distance is still just 4.12 pixels and the average distance is 1.04 pixels. Therefore, the performance gained on the noise masks, which occlude 40% of the input

Initialization	40 %	60 %	80 %
Zero	34.13	31.27	28.02
Mean	34.14	31.25	28.01
Median	34.14	31.24	28.01
Otsu threshold	34.12	31.26	28.01

Table 4.2: Performances gained in the noise removal experiment. The initialization values are computed from the entire known image.

images, is higher than the performances that were achieved in the last experiment. The masks that were used in the preceding experiment cover just 25% of the input images, which confirms the statement that the inpainting performance is highly dependent on the structure of the distinct mask regions.

If the noise level is set to 40% or 60%, the restoration with an FoE prior produces convincible results. If 80% of the pixels are missing, the recovery becomes apparent, particularly in textured regions. This can be seen in Figure 4.9, where the three different disturbances are removed from an input image. The restorations of the masks that occlude 40% and 60% (shown in the first two rows of Figure 4.9) look plausible. On the contrary, the recovery of the remaining domain (given in the last row of Figure 4.9) looks unnatural, since edges are frayed and image details, like parts of the man's mouth, are lost.

4.1.3 Comparison with other Inpainting Algorithms

This section provides a comparison between results found in the inpainting literature and outputs that are obtained with a generic FoE prior. The basic ideas of the algorithms selected were already introduced in Chapter 2. The majority of the compared algorithms presents no numerical results. Therefore, a side-by-side comparison is given and numerical results are provided if they are available. The presented FoE outputs have been inpainted with $\eta = 100$ and the inpainting regions were initialized with zero.

The first image compared is produced by the PDE based algorithm suggested by Bertalmio et al. in [BSCB00]. The input image and the associated inpainting mask are shown in Figure 4.10 (a). The inpainting result of the PDE based method is presented in Figure 4.10 (b) and the output of the FoE algorithm is given in Figure 4.10 (c). The output produced by the FoE prior, is slightly inferior compared to the inpainting result of the gradient based approach, since the inpainting regions are more blurred out by the statistically based method. The blurring becomes especially apparent in the inpainting region that is partially occluding the hauberk of the knight.

In the previous experiment an artificial mask was restored. A more practical application field of image inpainting is scratch removal. One example for this application field is given in Figure 4.11, where a scratch is removed from a photography of Lincoln. The inpainted result in Figure 4.11 (b) is obtained by the diffusion based approach that is presented in [OBMC01]. The output of the FoE approach - given in Figure 4.11 (c) - is slightly inferior, because the region, which is covering the hair of Lincoln, is oversmoothed.

Figure 4.12 provides a comparison to the TV based inpainting algorithm that is suggested by Loeza and Chen [BLC08]. The image in Figure 4.12 (b) is produced by this method. The output of the FoE algorithm is given in Figure 4.12 (c). It is not evident, which restoration result should be favored, because of two reasons: One the hand, it is noticeable that the FoE result looks smoother than the image in Figure 4.12 (b), in which curvature is often not maintained. On the other hand, the FoE algorithm is incapable of reconnecting the thin hair part (2 pixels wide), whereas the TV based method fills the unknown area.

A further practical application for inpainting is object removal. A challenging problem, belonging to this category, is shown in Figure 4.13 (a). The birdcage is covered by an inpainting mask, which is considerably thicker than the domains that were used in the

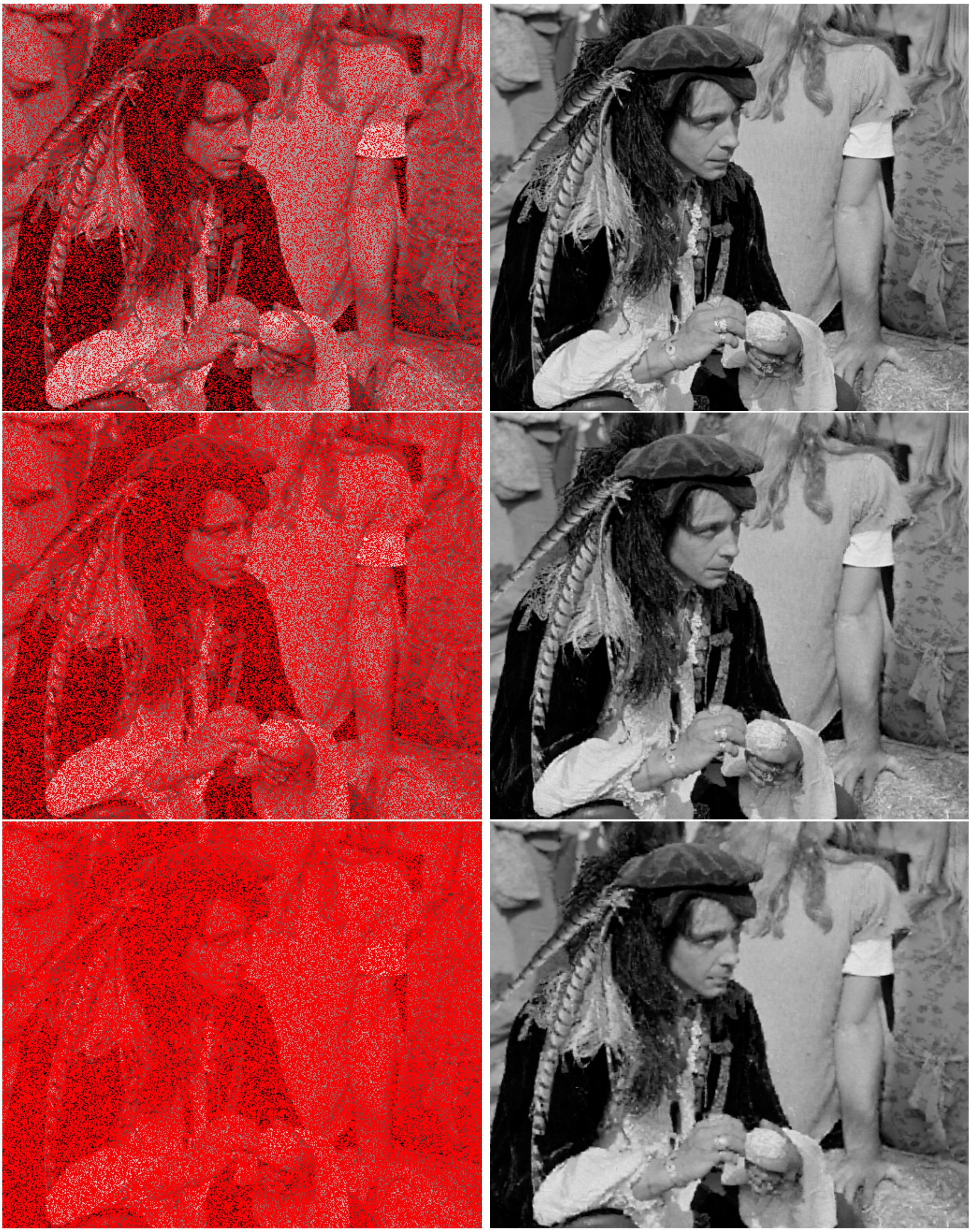


Figure 4.9: Inpainting of three different random masks. In the left column the input image and the superimposed masks are shown. In the right column the corresponding results are given. The PSNR values of the restorations - from top to bottom - are: 33.25 dB (40% masked), 30.17 dB (60% masked) and 26.90 dB (80% masked).

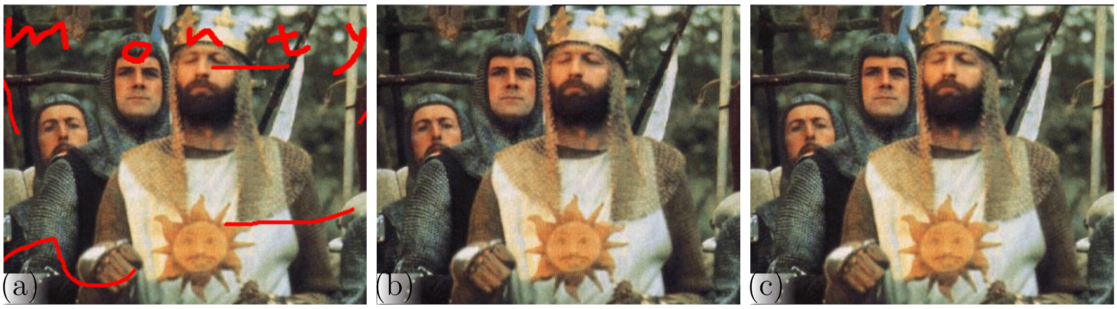


Figure 4.10: Inpainting of a hand drawn mask. (a) Frame of the movie “Monty Python and the Holy Grail” and superimposed mask. (b) Inpainting result of [BSCB00]. (c) Output of the examined algorithm.

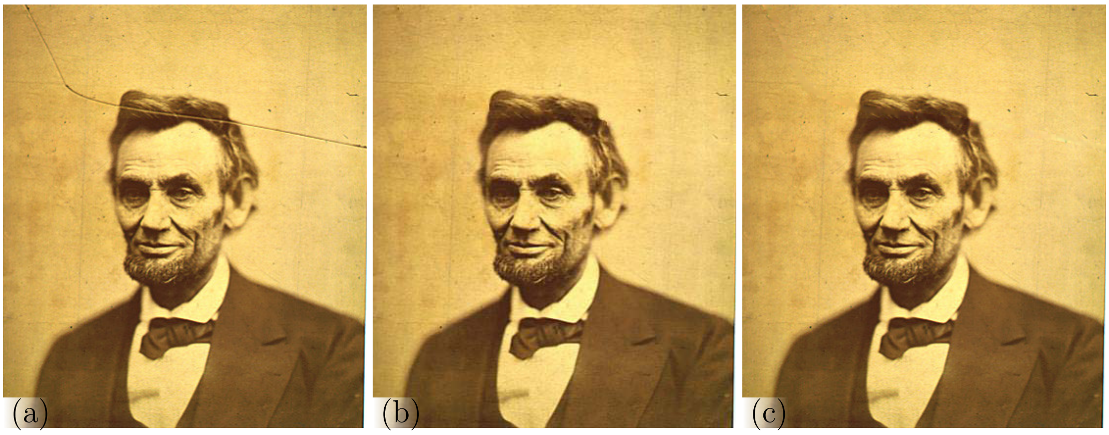


Figure 4.11: Scratch removal. (a) The photography of Lincoln is damaged by a scratch. (b) Restoration result of [OBMC01]. (c) Image recovered by the FoE algorithm. The images in (a) and (b) are taken from [OBMC01].

previous examples. The inpainting task is further complicated by the fact that the input image contains textured regions. The tensor-driven method proposed by Tschumperle [Tsc06] is capable of overcoming those difficulties, since the resulting image in Figure 4.13 (b) is inpainted in a satisfying manner. On the contrary, the quality of the FoE restoration, shown in Figure 4.13 (c), is certainly lower since image details are not recovered and the silhouette of the bird is blurred. The example shows that the main weakness of the FoE inpainting approach is the inability to reproduce large and textured regions.

Finally, the FoE method is compared to four algorithms, which use sparse representations for the recovery of missing image regions. The four algorithms have been applied for the retouching of four noise masks that are similar to the domains, which were used in Section 4.1.2. The utilized masks cover 60% of the input images.

The PSNR values that are gained by the compared algorithms are given in the first four columns of Table 4.3. The performance of the FoE approach is presented in the last column. Especially noticeable is the weak performance that is gained by the FoE algorithm on the Barbara image. This can be attributed to the fact, that this image contains fine image structures, which are not reproduced by the investigated inpainting method. The similarities that are gained on the three residual images are in contrast

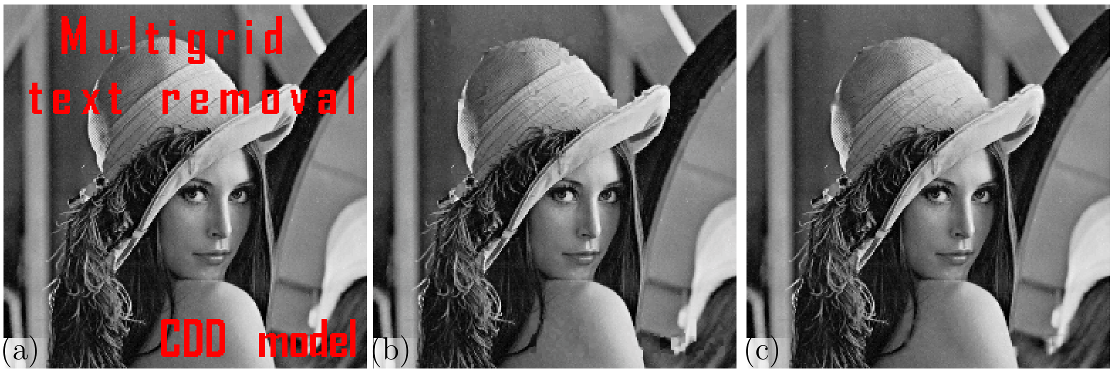


Figure 4.12: Text removal example. (a) Lena image and inpainting mask (b) Result obtained by the TV based algorithm in [BLC08]. (c) Image inpainted with the help of an FoE prior.

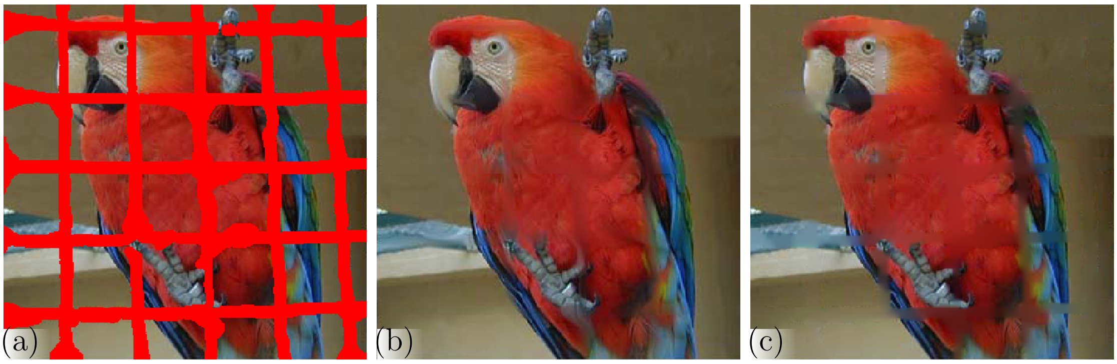


Figure 4.13: Object Removal. (a) The parrot is partially occluded by a cage, which serves as the inpainting mask. (b) The PDE based algorithm in [Tsc06] is able to fill the relatively wide mask in an adequate manner. (c) The output of the FoE approach is clearly inferior to the result of the PDE based method.

significantly higher.

The algorithm in [YSM10] achieves the highest performance on each image. Two resulting images of this approach are shown in Figure 4.14, along with the outputs that were produced by the FoE approach. The method in [YSM10] is able to inpaint the challenging Barbara image in a satisfying manner, while the FoE algorithm fails in the reproduction of the textured regions. The restoration of the Lena image looks in contrast more natural, but it is still inferior to the output of the method proposed in [YSM10].

Algorithm	[YSM10]	[Gul06]	[FSM09]	[ESQD05]	FoE
Lena	36.02	33.43	29.91	33.95	34.34
Barbara	34.05	29.77	28.52	32.33	26.60
House	37.41	34.10	29.56	35.23	33.86
Boat	32.53	30.40	27.58	31.00	31.74

Table 4.3: Performance (in PSNR) of the FoE approach compared to other state-of-the-art algorithms. The inpainting regions cover 60% of the images. The masks used for the images 'Lena' and 'Barbara' are obtained from [YSM10]. The other inpainting masks are not provided in [YSM10] and are therefore randomly generated.

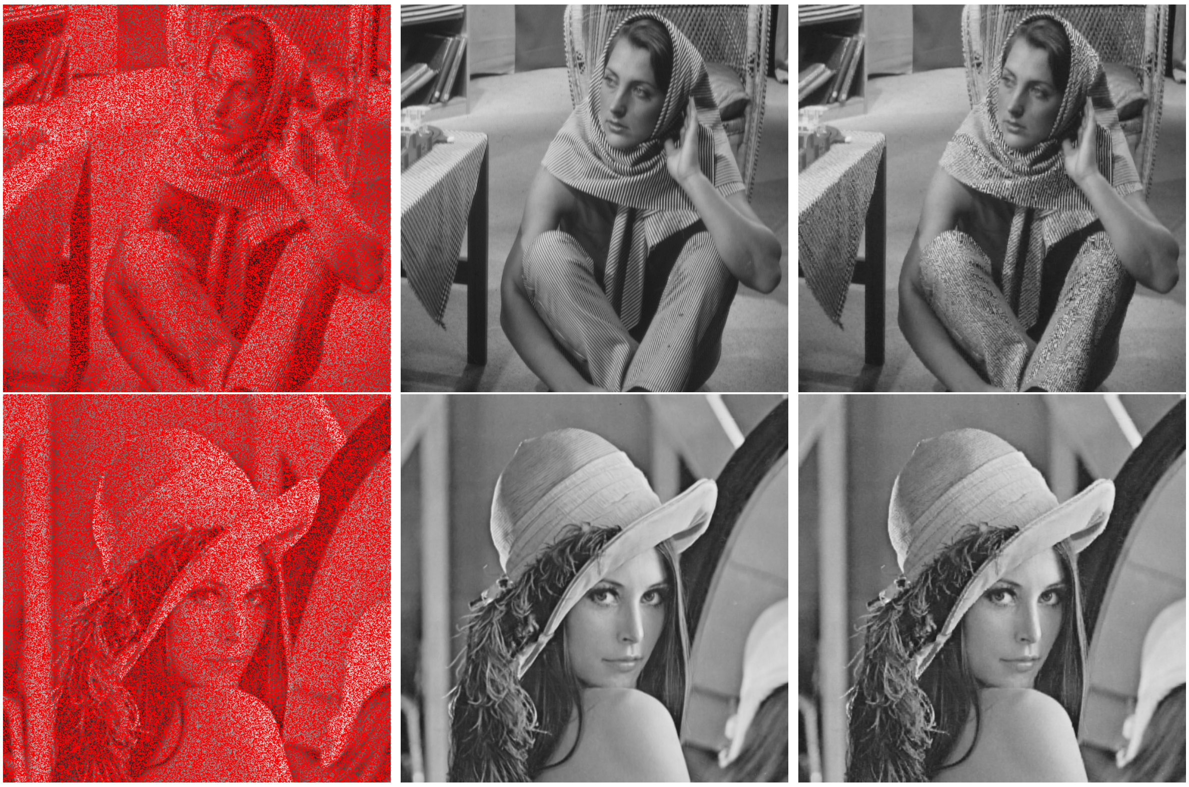


Figure 4.14: Inpainting of a randomly generated mask. (Left) Input images with superimposed masks. (Middle) Resulting images produced by the method in [YSM10]. (Right) Images recovered with an FoE prior.

4.2 Experiments on Synthetic Text Data

This section deals with the restoration of words that consist of Latin letters. The utilized words are taken from the IAM handwriting database [MB02], which contains handwritings that were scanned at a resolution of 300 Dots per Inch (DPI). The stroke widths and colors are varying and the text orientations are additionally diverse, since the database contains writings from over 600 writers.

The majority of the subsequent experiments are carried out on images containing a word that is overlapped by another word. Such test sets will be herein also referred to as “artificial palimpsests”. All input images, belonging to such sets, are randomly chosen from the database. (Except the input images that are utilized in the first experiment, which is conducted on a subset of the database.)

The inpainting masks are created from images, which have a similar size as the corresponding input images. If a randomly picked mask was greater or smaller than its assigned input image, the mask was cropped or zero-padded, respectively. It is noteworthy that from now on only the similarity (in terms of PSNR) inside the mask regions is provided. This is necessary due to the fact that the utilized masks have various extents.

The section starts with experiments that are performed using FoE priors. Afterwards a parameter evaluation of the tensor voting approach is given. The section ends with a comparative analysis of both examined approaches.

4.2.1 FoE Text Recovery

In the following, three experiments are conducted, which demonstrate the characteristics of the FoE based handwriting recovery. Firstly, it is analyzed how the statistics of the training database influences the inpainting result. Secondly, an evaluation of two parameters, namely the inpainting rate and the initialization value, is given.

Learning the Statistics of Handwritings

In the following experiment the performances of four different FoE models are compared. One prior captures the statistics of natural images. This image model was used for the recovery of the images that were presented in Section 4.1. The remaining three priors model the statistics of handwritings. The first handwriting model is trained on images that are randomly extracted from the IAM database. This database contains mainly words that are written in cursive lowercase. The second prior is instead trained on a subset of the database. The subset contains only words, which consist of uppercase letters. Those letters have another stroke orientation, than the cursive written words. The third handwriting model is again trained on images, extracted from the entire database. The images were randomly rotated before the sampling of the training patches was carried out.

Two different sets, containing artificial palimpsests, have been generated: The first set is comprised of 100 input images, which contain words that are written in lowercase. The second set contains instead 100 words that are written in uppercase.

Table 4.4 shows the average PSNR values that are gained inside the inpainting masks. It can be seen that the handwritings models gain a higher performance if the statistics of the test images are similar to the statistics of the patches, which have been used in the learning sequence. Hence, the weakest performance is achieved by the model that is trained on natural images.

It can be concluded that a handwriting prior favors the propagation of strokes, which have orientations that are similar to the main orientation of the training set. Connections with different orientations are more likely to get vanished, due to the rotation variance of the image model. The random rotation of the training images is not sufficient to overcome this drawback, as it is indicated by the low PSNR value.

Test set \Training set	Mixed letters	Capitals	Rotated patches	Natural images
Lower case letters	19.03 dB	18.78 dB	18.86	18.06 dB
Upper case letters	17.08 dB	17.37 dB	16.78	16.34 dB

Table 4.4: Restoration performance of four different image priors.

One example for the influence of the training data is given in Figure 4.15. The image in Figure 4.15 (c) was restored by the prior that is trained on the entire dataset. Figure 4.15 (d) shows the output of an FoE prior that models the statistics of uppercase letters. It is obvious that the cursive written word is better recovered by the former mentioned image model. This prior is used in the following experiments.

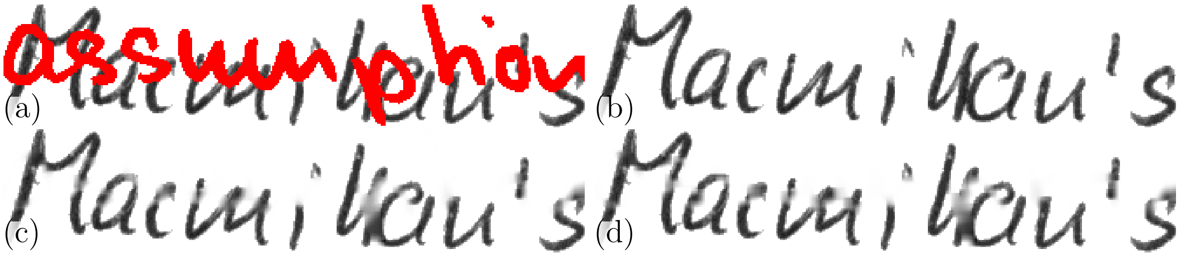


Figure 4.15: Text restoration using different FoE priors. (a) Input image and corresponding mask. (b) Ground truth image. (c) Restoration result of a model, which has been trained mainly on cursive words (PSNR: 17.47 dB). The retouched image looks more natural than the image in (d), which is produced by a prior that captures the statistics of uppercase letters (PSNR: 16.84 dB).

Parameter Evaluation

In this experiment the influence of the initialization value is evaluated. The analyzed dataset contains 100 artificial palimpsests and the utilized inpainting rate is 100. The background of the inpainted images is uniformly white and the text color is varying. 11 initialization values in the range between 0 and 1 are evaluated, whereby 1 represents the maximum intensity value. The average PSNR values achieved are presented in Figure 4.16.

The plot reveals that an initialization, which is similar to the background color, improves the inpainting result. The highest similarity is gained, if the unknown regions are initialized with 0.9. The initialization with the dominant background color prevents an introduction of smearing artifacts. Those artifacts are present in large inpainting regions if they are initialized with a gray value that is at least smaller than 0.5.

One example is shown in Figure 4.17. The results shown depict that the restoration process benefits from an initialization with the dominant background color. From this point on, only the initialization with the dominant background color is used in the subsequent experiments.

In the next experiment, the FoE inpainting algorithm is used for the removal of ruling lines that are partially occluding handwritings. The impact of the inpainting rate η is analyzed. The utilized test set includes 100 input images. Three masks have been

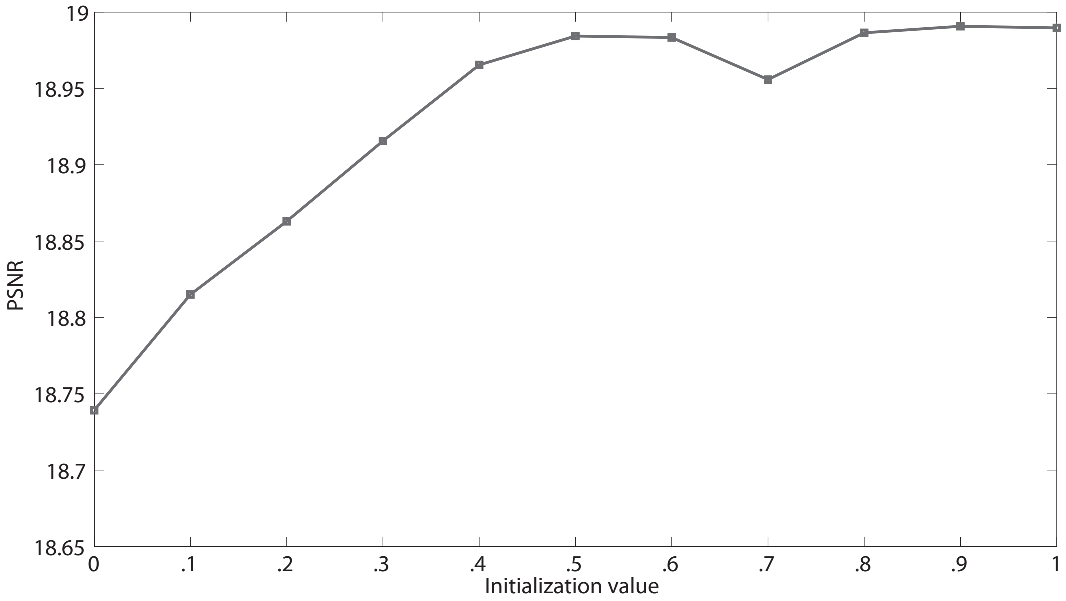


Figure 4.16: Evaluation of the initialization value.

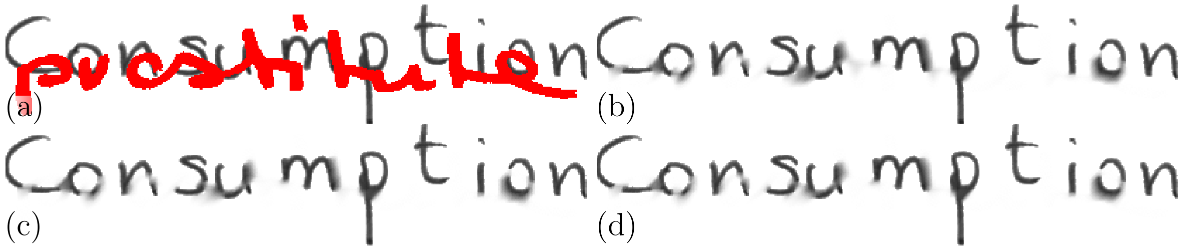


Figure 4.17: Handwriting restoration, using different initialization values. (a) Disocclusion task. (b) Initialization value: 0, PSNR: 15.48 dB. (c) Initialization value: 0.5, PSNR: 16.18 dB. (d) Initialization value: 1, PSNR: 16.48 dB.

generated for each input image. The masks contain horizontal ruling lines, which are evenly spread over the input images. The vertical distance between two line centers is 20 pixels and the number of lines depends on the height of the input image. The three masks that belong to an input image contain lines with different thicknesses: The ruling lines have vertical extents of 3, 5 and 7 pixels. One example for the ruling line removal scenario is given in Figure 4.18.

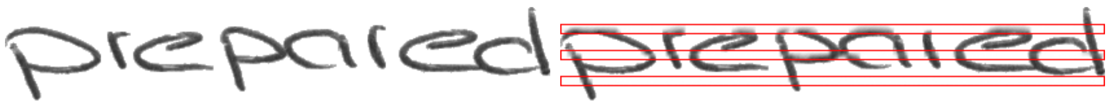


Figure 4.18: Ruling line removal example. (a) Ground truth image. (b) Restoration result (PSNR: 19.67 dB). The borders of the ruling lines are illustrated by the red boxes. The lines have a thickness of 7 pixels.

The results gained are presented in the plot, shown in Figure 4.19. The broadening of the ruling lines leads to a reduction of the PSNR values. The maximum PSNR value is 22.59 dB. This similarity is scored, if an inpainting rate of 110 is utilized for the restoration of the three pixels wide lines. The highest PSNR value gained on the 5 pixels wide mask regions is 20.03 dB. This similarity is obtained by applying an inpainting rate of 130. The worst results are obtained in the case of the restoration of the 7 pixels wide lines. The maximum PSNR value for this mask type is 18.24 dB. This similarity value is achieved, if η is set to 120.

The inpainting result is not sensitive on the inpainting rate, since there are only minor performance differences for the rates in the range from 80 to 140. This insensitivity to the step rate was already observed on natural images and will not be evaluated any further. In the following experiments η will always be set to 100.

4.2.2 Tensor Voting

In the following experiment the performance of the tensor voting approach is evaluated. The dataset, which was utilized in the ruling line removal experiment, is also used in this test. It was mentioned in Section 3.4 that the tensor voting approach is exclusively designed for the restoration of binarized text images. The preceding test was instead carried out on gray scale images. Therefore it was necessary to convert the binary images, produced by the tensor voting approach, into gray scale images, in order to enable a comparison between the heuristical and the statistical approach. The conversion is fulfilled by using the following, simple technique:

At first the color of each distinct foreground region in the inpainting domain is determined. Therefore, the known foreground pixels in a three pixel wide band around the

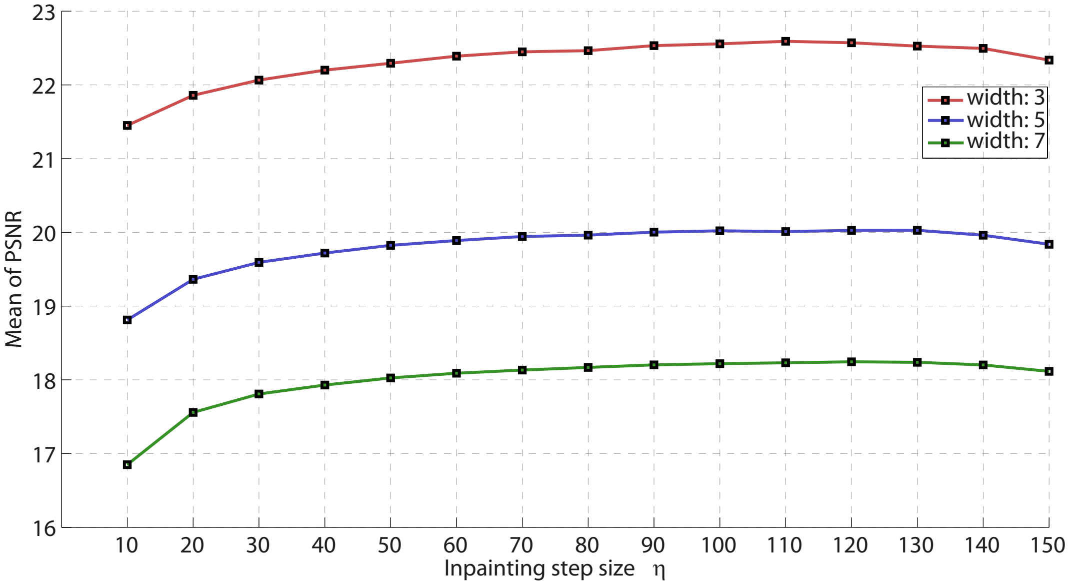


Figure 4.19: Performance of the FoE approach gained on three different mask types. The masks contain horizontal ruling lines with various thicknesses (see legend for details).

restored region are considered. The recovered foreground region is filled with the median of the gray values belonging to the considered pixels. In the second step, the fading of the character strokes is imitated. The inner regions of the utilized handwritings are typically darker than the outer regions, where the ink is faded out. Therefore, a second post processing step is applied, in order to approximate this property: The restored gray scale image is filtered with a 5×5 Gaussian filter. The color recovery technique explained is not capable of filling in color gradients. This drawback becomes apparent, if the restored region is relatively large.

The result of the tensor voting approach depends mainly on the parameter σ, which defines the scale of voting. This means that a relatively high σ allows for a connection of faraway tokens, while a small σ preserves local structure. In this experiment it is evaluated, how σ affects the recovery of the regions that are occluded by ruling lines.

The mean similarities that are gained for each ruling line type are presented in Figure 4.20. The investigated σ values are given on the horizontal axis and the PSNR values are depicted on the vertical axis. Two remarkable circumstances can be seen in the figure: Firstly, the inpainting performance is inversely proportional to the line width and considerably lower than the results, which are scored by the FoE approach. Secondly, the maximum similarity values for each mask class are obtained with diverse σ values: In the case of the 3 pixels wide lines, the highest similarity is 19.20 dB. This PSNR value is gained if σ is set to 4. A higher σ is more appropriate for the reconstruction of the 5 pixels wide mask regions: If σ is set to 5, the highest similarity, namely 17.17 dB, is gained. Finally, the maximum PSNR value inside the 7 pixels wide mask regions is 15.85 dB. This similarity is gained, if σ is set to 6.

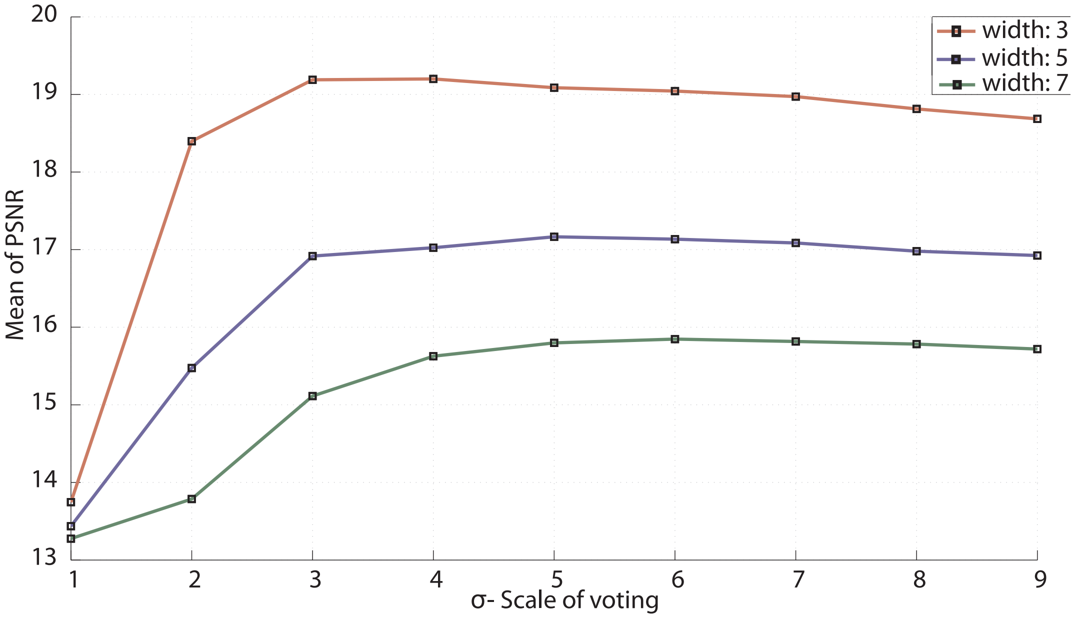


Figure 4.20: Performance of the tensor voting based approach. The utilized masks contain horizontal ruling lines with varying thicknesses.

In the case of the 3 pixels wide mask regions, a small σ enables a satisfying filling of the relatively small gaps. By the use of this small scale of voting, the influence of faraway tensors is prevented and only nearby tensors cast votes into the inpainting domain. Nevertheless, if the inpainting area is thicker (like the 5 and 7 pixels wide regions), σ must be increased in order to enable a connection of matching edges. By raising σ, the impact of distant tensors is increased and tensors in the inpainting area receive votes from multiple tokens, with different orientations. Thus, the saliency of tensors inside the unknown area is lowered and local structures are often not preserved.

In Figure 4.21 an example is given, which illustrates how σ affects the restoration result. The input image and the 5 pixels thick mask are shown in Figure 4.21 (a) and the ground truth image is given in Figure 4.21 (b). The image in Figure 4.21 (c) is inpainted with $\sigma = 3$. This value is too small for a successful closing of the gaps, since the votes are not casted into the mask centers and the extracted edges are therefore interrupted. The image in Figure 4.21 (d) is restored with a more appropriate σ value, namely 5. The image in Figure 4.21 (e) is inpainted with $\sigma = 9$.

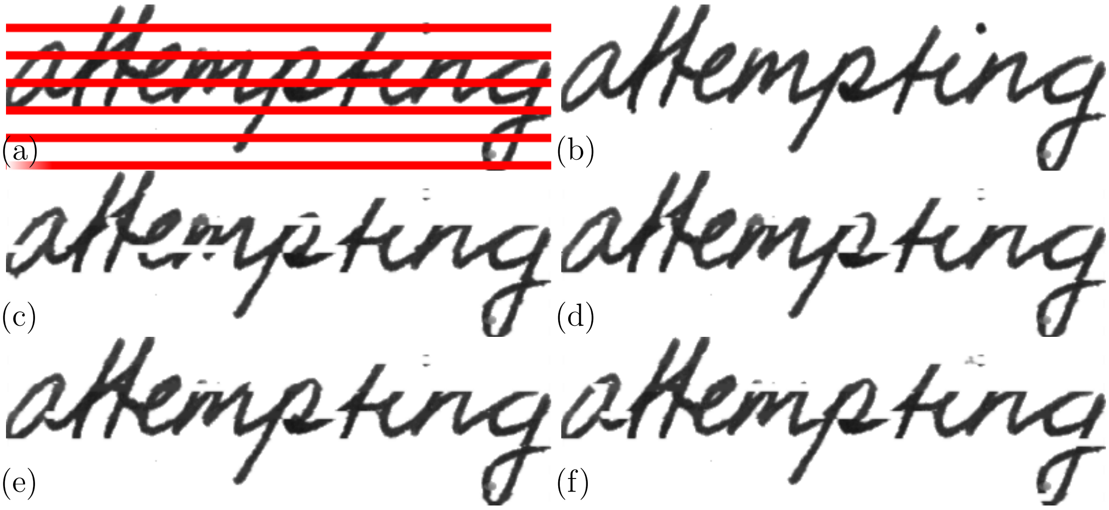


Figure 4.21: Impact of σ. (a) Input image with superimposed mask. The mask regions are 5 pixels wide. (b) Ground truth image. (c) $\sigma = 3$: 14.83 dB. (d) $\sigma = 5$: 15.83 dB. (e) $\sigma = 9$: 16.14 dB. (f) $\sigma = 12$: 15.03 dB.

If we compare Figure 4.21 (d) and Figure 4.21 (e), we can see that the hole in the letter 'p' is better reconstructed with the greater σ value, since it is located at the end of a large, linear structure. By contrast, the letters 'a' and 'm' in Figure 4.21 (e) contain small holes, which are not present in Figure 4.21 (d). Those local structures are better preserved by the smaller scale of voting. Finally, the output of $\sigma = 12$ is given in Figure 4.21 (f). The extracted edge image contains interrupted edges, caused by votes of faraway tensors. Thus, only large structures with very similar orientations are preserved, while edges with a high curvature are not propagated.

The images in Figure 4.21 demonstrate that a single σ value is suitable for the closing of particular inpainting regions, while other σ values are more appropriate for the recovery of the remaining holes. Hence, the performance can be increased by applying various σ values during the inpainting sequence. The scale of voting can be adapted to the extent of the holes or to the curvature of the nearby edges. This extension has not been investigated due to time restrictions, but will be explored in the future work.

In the current implementation only a fixed σ value is used for a particular test set. Figure 4.22 shows an example for the restoration of line regions with increasing extents. The 3 pixels thick regions are inpainted with $\sigma = 4$ (Figure 4.22 (a)), the 5 pixels thick lines are restored with $\sigma = 5$ (Figure 4.22 (b)) and the largest inpainting regions are recovered by setting σ to 6 (Figure 4.22 (c)). All images contain small disturbing artifacts that are either caused by votes of multiple tokens or by the morphological operations, which were depicted in Section 3.4.2. Those artifacts are especially apparent in the border regions of the inpainting domains.

Furthermore, it can be seen that the broadening of the domains leads to a reduction of the similarity. While most gaps in the first restoration given are successfully closed, the remaining images contain more interrupted strokes. The tensor voting based approach is generally only capable of restoring small gaps. However, the term small is imprecise and therefore it will be clarified in the following section, which gaps can be restored successfully by both handwriting recovery methods.

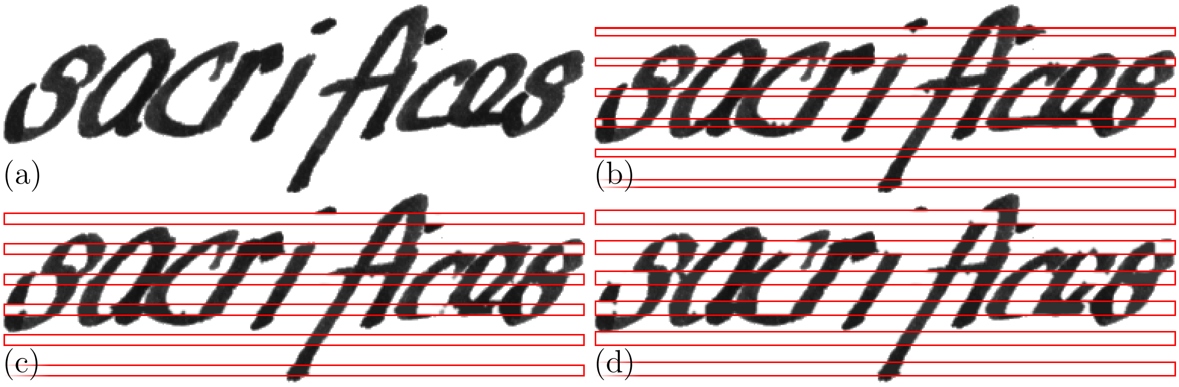


Figure 4.22: Restoring regions that are occluded by ruling lines. (a) Ground truth image. (b) 3 pixels wide ruling lines. $\sigma = 3$: 17.79 dB. (c) 5 pixels wide ruling lines. $\sigma = 4$: 14.55 dB. (d) 7 pixels wide ruling lines. $\sigma = 7$: 12.90 dB.

4.2.3 Comparative Analysis

In the last experiment that is conducted on synthetic data, the performances of both restoration techniques are compared. The actual test addresses the issue which kind of occlusions, caused by overwritings can be restored in an appropriate manner. Therefore, it is necessary to characterize the masks or occlusions, respectively. It was already shown that it is inappropriate to classify the masks based on the number of unknown pixels. Therefore the masks are instead categorized by the mean width of the overlapping strokes[2].

The classification described does not consider whether the foreground or the background is occluded. The occluded regions should also be regarded, since the recovery of

[2] The mean stroke width of a mask is determined as follows: At first the skeleton of the binary image is computed. Afterwards the distances between the skeleton pixels and the nearest background pixels in the mask image are computed. The average of this distances serves as the mean stroke width.

the uniform background is trivial, whereas an appropriate restoration of the foreground is more complicated than a correct reconstruction of the uniform background. Hence, another classification kind is also utilized in this experiment: The occlusions are described with the ratio between the amount of occluded foreground pixels and the overall number of foreground pixels. This categorization takes into account that the restoration quality depends not only on the extent of the inpainting mask, but also on the occluded image content.

The investigated dataset is comprised of 1000 artificial palimpsests. The retouching with the FoE algorithm took about 11 hours, whereas the tensor voting based approach needed considerably less time for convergence, namely about one hour. The experiment was performed on a single PC, which was equipped with an Intel Core 2 Duo processor (2.4 GHz) and 4 GB RAM.

Firstly, the test set is divided into 8 groups, based on the occlusion ratio (ranging from 10% to 50%). For the statistical approach an inpainting rate of 100 was applied, while tensor voting was fulfilled by setting the scale of voting to 6. The similarity values gained are presented in Table 4.5 and in Figure 4.23. Additionally, a further series of PSNR values is provided in order to allow for a better evaluation of the gained results. These similarity values are achieved, by simply filling the inpainting regions with the background color.

Both inpainting approaches produce results with a higher similarity than the background filling. Furthermore, the similarity values gained by the statistical inpainting approach are superior to the results obtained by the heuristic method. This can be attributed to the fact that the tensor voting approach is only capable of filling holes that are surrounded by edges with compatible orientations. Other inpainting regions are left untouched, whereas the FoE tends to alter such regions, if the occlusion ratio does not exceed a certain value. It is difficult to draw a general conclusion about this value, since the results depend on the complexity of the input image and the mask. However, it can be seen in Table 4.5 that the PSNR values, gained by the FoE approach, are at least around 20 dB, if the occlusion ratio is not greater than 20%. The heuristical inpainting approach gains a similar PSNR value only if less than 10% of the foreground pixels are occluded.

The performance of both methods is monotonically decreasing with respect to the occlusion ratio. The difference of both algorithms is 3.379 dB on average. It can be concluded that the FoE approach restores handwritings in a more appropriate way than the method that is based on tensor voting.

Table 4.6 and Figure 4.24 present the results that are gained by grouping the masks based on the width of the strokes contained. The mean performance difference of both restoration approaches is 3.71 dB. The difference between the tensor voting approach and the background filling is 3.28 dB on average. Contrary to the preceding results, the PSNR values of the tensor voting approach are not monotonically decreasing: The mean PSNR value gained on the 8 pixels wide mask regions is 14.00 dB. A considerably smaller similarity value, namely 13.31 dB, is in contrast achieved on the 7.5 pixels broad regions. If we consider the performance gained by the background filling, we can see that the last PSNR value is also considerably higher than the second last similarity value. This indicates that the last subset of the test data contains more occluded background regions

Occlusion (in %)	10	15	20	25	30	35	40	50
FoE	23.12	20.78	19.34	18.03	17.12	16.42	15.58	14.84
Tensor voting	20.01	17.46	15.82	14.50	13.65	13.04	12.04	11.67
Background	15.36	13.14	11.69	10.80	10.48	10.24	9.67	9.71
# of samples	67	135	224	263	168	95	31	17

Table 4.5: Performance comparison of the examined inpainting techniques. 1000 artificial palimpsests are grouped based on their occlusion ratio. The upper bounds of the particular ratios are given in the first row.

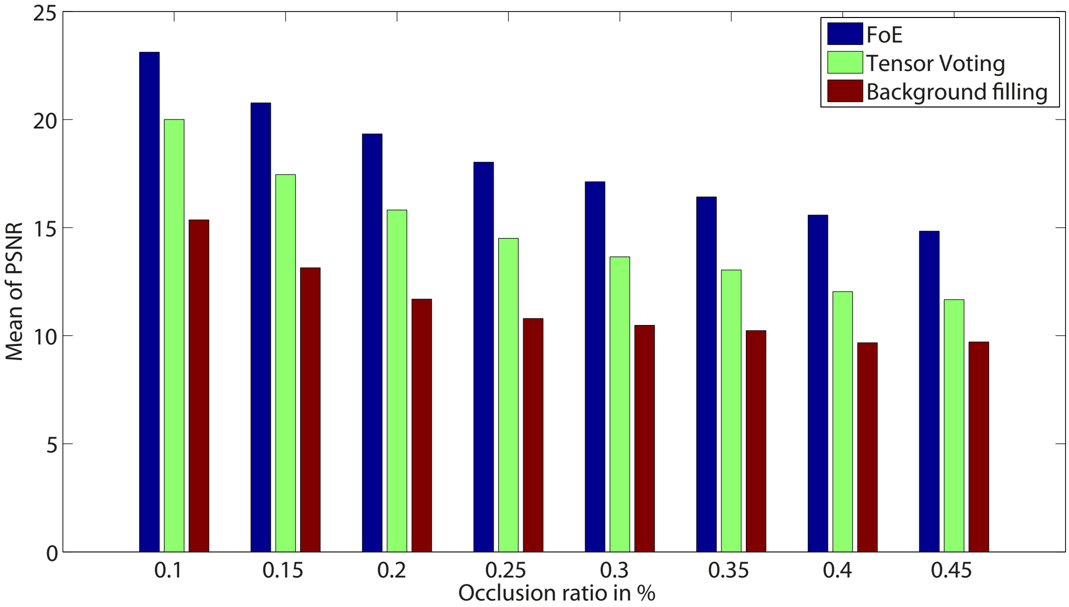


Figure 4.23: Performances gained by both systems. The utilized test set is divided based on the occlusion ratio. The red bars depict, which similarity values are gained by filling the mask regions with the background color.

than the second last subset.

In order to provide an idea of how different inpainting problems are solved by both approaches, four representative examples are given in Figure 4.25 to Figure 4.28. Additionally, the results of the background filling are provided. Figure 4.25 and Figure 4.26 show the recoveries of masks with a relatively small extend. Both restoration techniques produce relatively convincing results. Nevertheless, the numerical results are diverse which can be attributed to the fact that the tensor voting based approach fills distinct regions with a solid color, while the FoE approach is able to fill in color gradients. Furthermore, it can be seen that the statistical approach tends to over smooth the inpainting areas, whereas the heuristical approach generates strokes with a higher background contrast.

Figure 4.27 and 4.28 illustrate the restoration of greater masks. The tensor voting algorithm is not capable of connecting most matching edges and does not alter the majority of the holes. The statistical algorithm alters more inpainting regions, but the inpainting

Mask stroke width (in px)	4.5	5.0	5.5	6.0	6.5	7.0	7.5	8.0
FoE	20.82	20.15	19.62	19.12	18.14	17.24	17.00	16.50
Tensor voting	16.96	16.43	15.67	14.73	13.91	13.87	13.31	14.00
Background	11.71	11.65	11.65	11.65	10.95	11.65	10.35	13.00
# of samples	56	135	190	184	186	125	57	67

Table 4.6: Performance evaluation of both recovery methods. The used masks are classified by their mean stroke widths. The upper bounds of the chosen stroke widths are given in the first row.

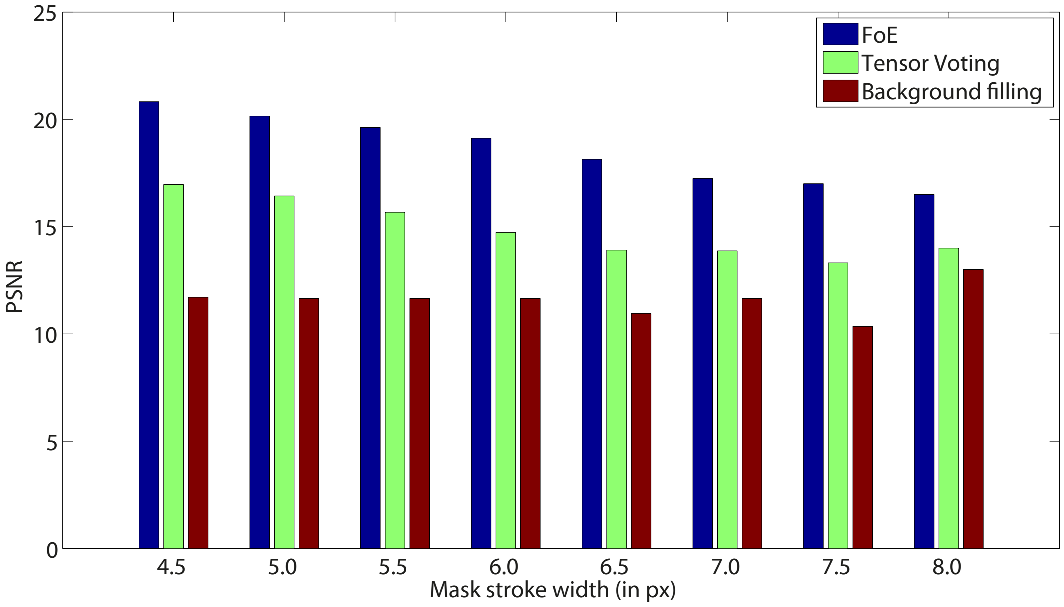


Figure 4.24: Evaluation of both systems. The masks are grouped based on their mean stroke width.

process is clearly observable, since the restored strokes are blurred out. The greatest occluded regions are also filled with the background color.

The numerical results and images presented exhibit that the FoE algorithm produces more satisfying restorations than the tensor voting method. The latter algorithm is based on the assumption that edges, which are belonging together, are detected trough the voting process. However, it turned out the usage of a single σ value is not appropriate, since small values are on the one hand not suitable for the filling of large holes. On the other hand, smaller regions require a small σ in order to restore local structures. The FoE approach is able to fill larger regions, but the inpainting process becomes more visible with the mask extent.

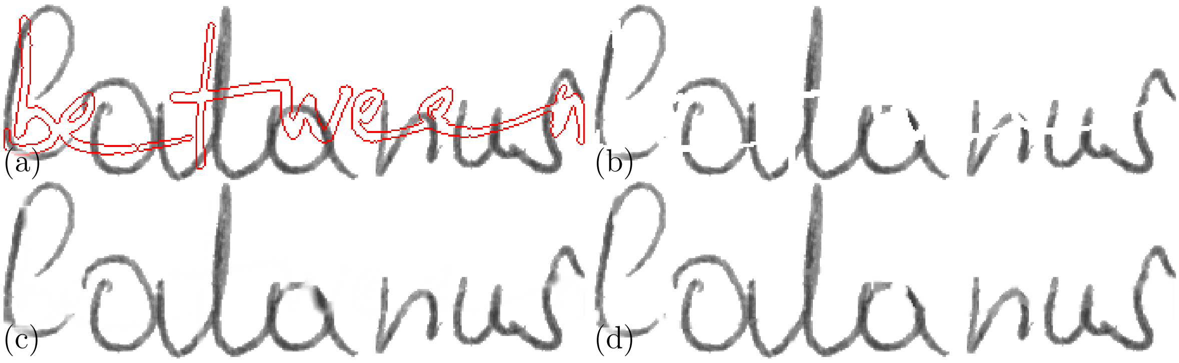


Figure 4.25: Inpainting of an overwritten area (Occlusion ratio: 10.6 %, mask width: 4.35 pixels) (a) Inpainting problem. (b) Background filling result (PSNR = 13.77 dB). (c) FoE result (PSNR = 21.72 dB). (d) Tensor voting result (PSNR = 19.69 dB).

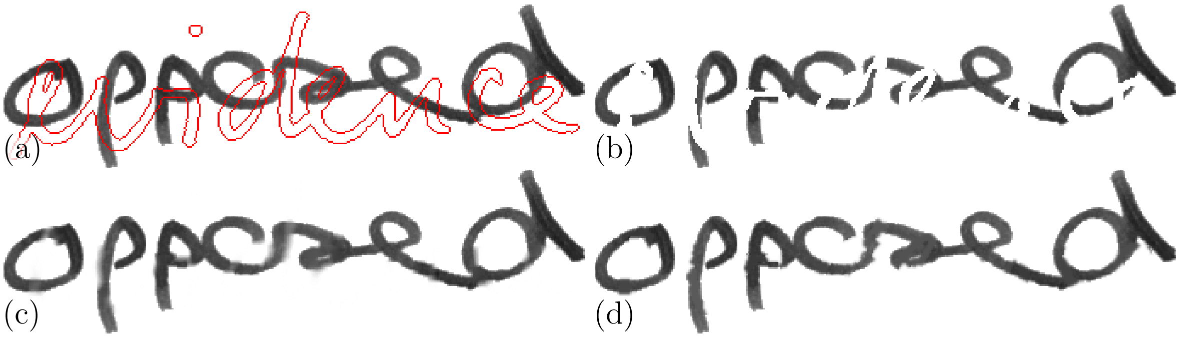


Figure 4.26: Inpainting of an overwritten area (Occlusion ratio: 18.9 %, mask width: 5.4 pixels) (a) Inpainting problem. (b) Background filling result (PSNR = 13.02 dB). (c) FoE result (PSNR = 20.18 dB). (d) Tensor voting result (PSNR = 18.21 dB).

4.3 Palimpsest Reconstruction

In this section the recovery of the utilized palimpsests is evaluated by means of manually created ground truth data. Four test panels have been extracted from different leaves belonging to the Archimedes palimpsest. The corresponding ground truth images have been created with the help of transcriptions that are provided from the Archimedes Palimpsest Project [NAW04]. It has to be mentioned that the ground truth was not generated by a philologist, but instead by the thesis author. Hence, the presented numerical results should be treated with caution.

The section is structured as follows: At first it is argued, why the palimpsest images used were downsized before the binarization and inpainting steps took place. Secondly, the mask generation step and its parameter sensitivity are analyzed. It turned out that it is necessary to enlarge the generated masks in order to prevent a propagation of overwriting vestiges that are located at the mask boundary. The necessity for this mask enlargement is shown in a separate experiment, where the restorations of untampered and enlarged masks are contrasted. Furthermore, a test is conducted that demonstrates the insensitivity to the inpainting rate. Finally, the outputs produced by a prior, which was trained on

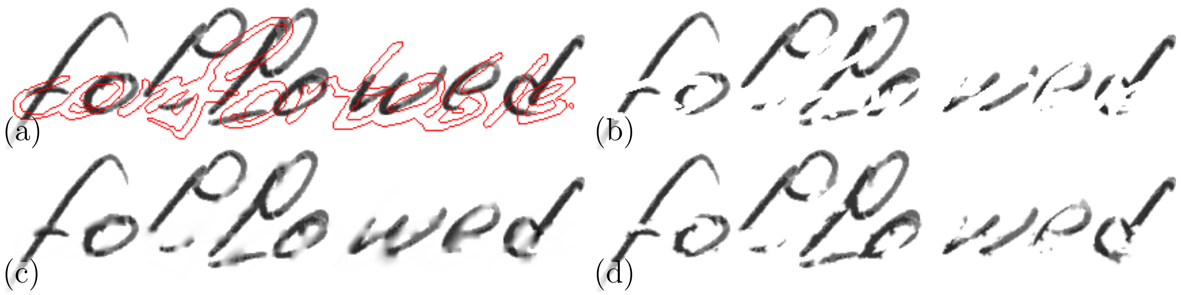


Figure 4.27: Inpainting of an overwritten area (Occlusion ratio: 31.2 %, mask width: 6.4 pixels) (a) Inpainting problem. (b) Background filling result (PSNR = 9.00 dB). (c) FoE result (PSNR = 14.47 dB). (d) Tensor voting result (PSNR = 11.41 dB).

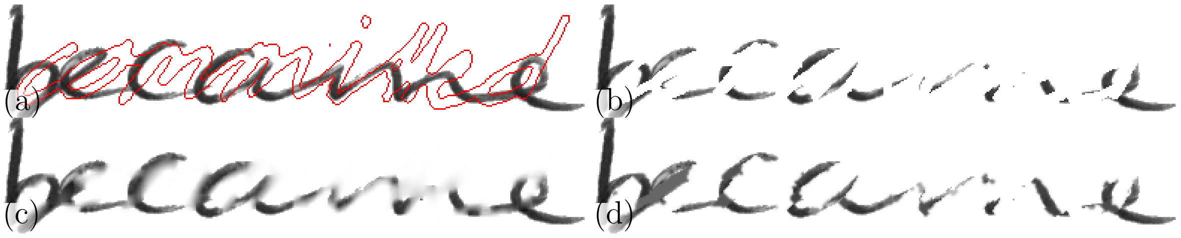


Figure 4.28: Inpainting of an overwritten area (Occlusion ratio: 45.8 %, mask width: 7.3 pixels) (a) Inpainting problem. (b) Background filling result (PSNR = 9.83 dB). (c) FoE result (PSNR = 16.47 dB). (d) Tensor voting result (PSNR = 12.24 dB).

underwritings, are compared to restorations that are generated by an image model that was trained on the IAM database.

4.3.1 Influence of the Gap Width

The statistical inpainting method is only capable of filling thin inpainting regions in a satisfying manner. Large regions are not altered sufficiently, which becomes also apparent when it comes to the restoration of the investigated historical writings. The parchments containing these writings, have been imaged at 700 DPI. While this resolution was considered to be adequate by scholars[3], it is too high for an adequate filling of the occluded areas. Therefore, the four test panels were downsized from 2001 × 2001 pixels to 501 × 501 pixels, which represents a resize factor of 0.25. The images were resized using bicubic interpolation.

The resize factor was chosen, based on a qualitative assessment of the recovered underwritings and based on the results that were obtained on synthetic data. The IAM database is scanned at a resolution of 300 DPI and the ratio between the resolution of the synthetic and the real world data is therefore 0.43 : 1. This indicates that the palimpsest images have to be downsized, but the ratio is not used as downsampling factor, since it cannot be assumed that both datasets have similar mask widths.

[3] From: http://www.archimedespalimpsest.org/imaging_production1.html, last accessed on 11. August 2011

Instead, a resize factor of 0.25 was chosen. The masks that are generated from such resized images have an average mask width of 7.2 pixels[4]. The preceding test showed that the recovery of mask regions with a width of 7 pixels or more leads to relatively poor results, because the unknown regions are often not altered sufficiently or blurred. Nevertheless, the parchment images were not downsized any further, since this would lead to an increased loss of image details.

One example for the inability to restore the panels at a high resolution is given in Figure 4.29: The image in Figure 4.29 (b) is restored at the original resolution of 700 DPI. The inpainted image in Figure 4.29 (c) was resized by a factor of 0.7, before inpainting took place. The restoration in Figure 4.29 (d) was obtained by applying a resize factor of 0.25. We can see that the usage of the latter factor leads to a worsened legibility of the underwritings. Nevertheless, a drastic downsizing of the input images is necessary, because otherwise the occluded areas are not altered sufficiently during the inpainting process.

4.3.2 Evaluation of the Mask Generation Step

In the following the sensitivity to the binarization parameters is analyzed. It was found that it is essential to dilate the binarized images by one pixel, because otherwise remaining overwritings vestiges are propagated into the mask region. Therefore, the masks that are presented in the following were dilated by one pixel. It will be depicted later on, why this measurement is inevitable.

Evaluation of the Binarization Parameters

The parameters, which are used in the utilized binarization algorithm, have to be adjusted according to the width of the strokes that are contained in the ancient documents. The segmentation method relies mainly on two parameters, which are used in Equation 3.2: The first parameter is N_{min}, which defines the minimum number of high contrast pixels in a local neighborhood window that are necessary for a foreground classification. The size of the neighborhood window is defined by the second parameter N_w.

Since Equation 3.2 depends both on N_{min} and N_w, it is essential to find an appropriate parameter combination. Therefore, it has been analyzed how several combinations of N_{min} and N_w affect the restoration results. A binarization evaluation in terms of recall and precision is not made, since we are interested in the overall recovery performance, which depends also on the inpainting step. Hence, the influence of the binarization parameters is analyzed by comparing the restored images with their ground truth counterparts.

The restored images have been inpainted with a rate of 100. The performance of 8 N_{min} values and 6 neighborhood sizes has been analyzed. In contrast to the experiments conducted on synthetic data, it is necessary to compare entire image domains, since the mask is not always covering the overwritings.

The mean similarities - in terms of PSNR - between the inpainted images and the corresponding ground truth images are given in Table 4.7. Each column shows PSNR

[4] This mask width is achieved under the following conditions: The utilized binarization parameters are: $N_{min} = 4$ and $N_w = 9 \times 9$. The masks, which have been generated with these parameters, are afterwards dilated by one pixel. (See Section 4.3.2 for a justification of the chosen settings.)

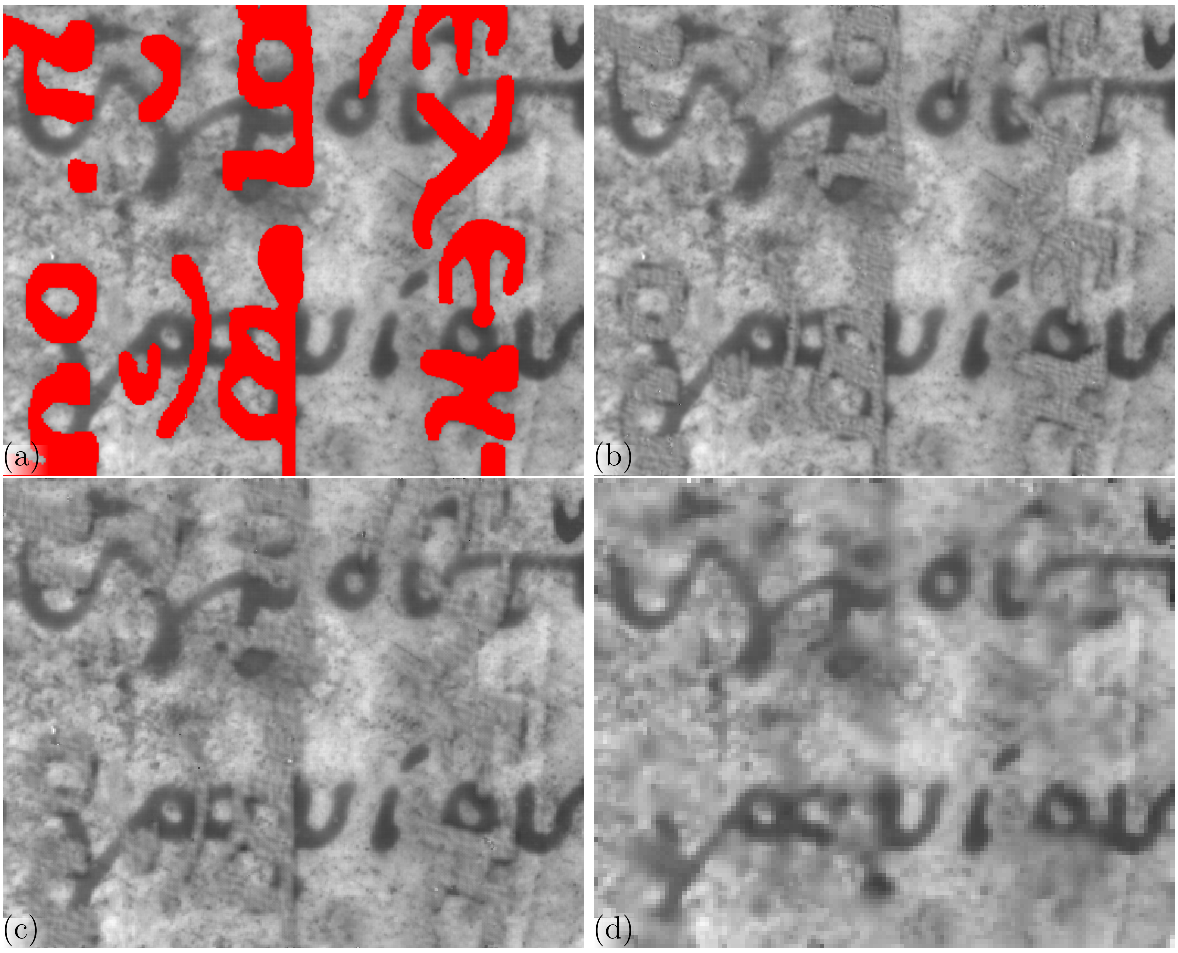


Figure 4.29: Restoring underwritings at different resolutions. (a) Input image. (b) Image inpainted at the original resolution of 700 DPI. (c) The input image was resized by a factor of 0.7. (d) Restoration obtained with the chosen resize factor of 0.25.

values for a fixed neighborhood size and the utilized thresholds are given in the table rows. The maximum PSNR value for a particular neighborhood size is printed in bold. The highest mean similarity is 24.99 dB, which is gained, if $N_w = 9 \times 9$ and $N_{min} = 4$.

The maximum PSNR values in the table columns indicate that an increasing of N_w must go in hand with a raising of N_{min}. This statement is based on the consideration of the given mean PSNR values. In order to substantiate this claim, the performance gained on panel 58 verso, is presented in the plot that is given in Figure 4.30. The horizontal axis depicts the investigated window sizes and the ordinate shows the scored similarity values. Each N_{min} value is represented by a single graph.

It can be seen that for a window size of 5×5 the highest PSNR value is achieved if the minimum number of high contrast pixels is set to 1. This N_{min} value is represented by the blue graph. If a higher N_{min} is combined with this small window size, the performance is significantly worsened. The performance for $N_w = 5 \times 5$ is in general low. There are two reasons for the weak performance caused by the small window size: On the one hand, text pixels, which are not marked as high contrast pixels, are often misclassified because

$N_{min} \backslash N_w$	5×5	7×7	9×9	11×11	13×13	15×15
1	**24.4848**	**24.9050**	24.8882	24.7317	24.6022	24.5934
2	24.2419	24.8749	24.9352	24.8062	24.6602	24.6620
3	23.9097	24.7803	24.9881	24.8598	24.7396	24.6947
4	23.5745	24.6677	**24.9885**	24.9011	24.7930	24.7480
5	20.6459	24.5347	24.9162	**24.9475**	24.8370	24.7911
6	22.5438	24.3276	24.9038	24.9301	24.8375	24.8104
7	22.0258	24.1440	24.8438	24.9385	**24.8739**	24.8313
8	19.4116	23.9658	24.7772	24.9415	24.8763	**24.8494**

Table 4.7: Mean PSNR values that are gained using various parameters combinations. The maximum PSNR value for each neighborhood size is written in bold.

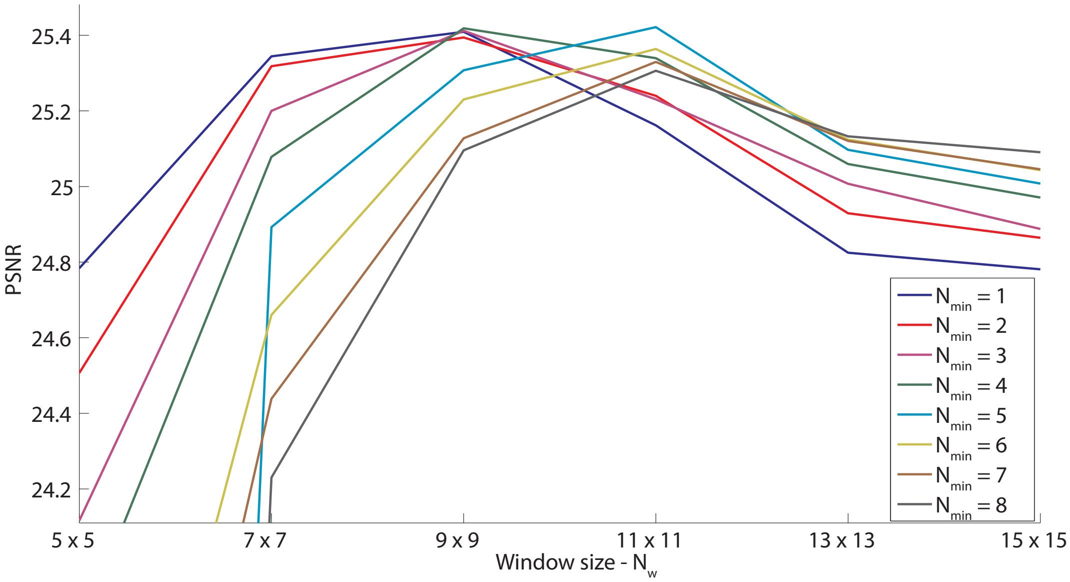


Figure 4.30: Restoration of panel 58 verso with varying binarization parameters. The restoration quality is shown on the vertical axis and the utilized window sizes are presented on the horizontal axis. The applied N_{min} values are illustrated by separate graphs.

their distance to high contrast pixels is mostly larger than the utilized window size. On the other hand, a relatively low N_{min} value must be used in combination with the small neighborhood window, because otherwise the image is under-segmented. However, by lowering the threshold, the sensitivity to noise gets enlarged. Thus, varying background regions are incorporated into the inpainting mask.

A higher similarity is instead achieved, if the window length is set to 7, 9 or 11. However, a high performance is only gained if N_{min} is chosen according to the window size. Appropriate N_{min} values for $N_w = 7 \times 7$ are 1 (blue graph) and 2 (red graph). For a window length of 9, the highest similarity is gained if the number of high contrast pixels is set to 4 (green graph). It can be seen in Figure 4.30 that if N_{min} is set to 1, 2 or 3 similar, but slightly decreased PSNR values are achieved. If these thresholds are in contrast used in combination with greater window sizes, the performance is significantly worsened. This stems from the fact that the binarized images are over-segmented. The over-segmentation becomes especially apparent, if a N_{min} value that is smaller than 4 is used in combination with a window length that is larger than 11. By making use of a greater N_{min} value, the number of false positives is decreased, but the performance is still weak, compared to more appropriate parameter combinations, e. g. $N_w = 9 \times 9$ and $N_{min} = 4$.

The dependency between N_{min} and N_w was also observed on the other panels. In order to enable a vivid comparison of the performances gained on the four test panels, a subset of the parameter combinations presented is used in the following analysis. For each neighborhood size a fixed N_{min} value was chosen: Each value was selected in order that the highest mean similarity - already presented in Table 4.7 - is gained for a particular window size. The resulting similarity values are presented in Figure 4.31. It can be seen that the overall results for different panels are diverse. While the reason for this circumstance will be explained later on, we will focus on the results for each panel separately for the moment.

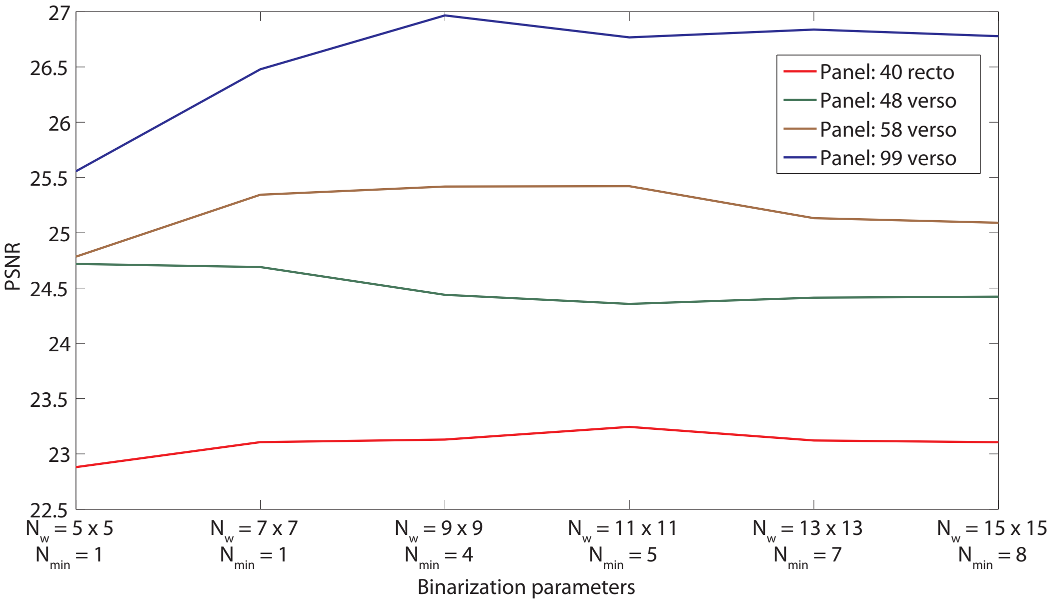


Figure 4.31: Restoration of four panels, using various binarization parameter combinations. The N_{min} values are given on the abscissa, along with the utilized window sizes.

It is conspicuous that the highest similarity for panel 48 verso is yield if N_w is set to 5×5, whereas the remaining panels are better restored by higher N_w values. This can be attributed to the fact that the tungsten illuminated photography of leaf 48 verso contains characters with a high background contrast, compared to the other investigated parchments. Since panel 48 verso contains less faded overwriting characters than the remaining panels, the panel is best restored with $N_w = 5 \times 5$, whereas the other panels require a greater window size.

A portion of the tungsten illuminated photography of this parchment is shown in

Figure 4.32 (a). In Figure 4.32 (b) the corresponding part of the UV photography is shown, along with the inpainting mask that is generated by applying $N_w = 5 \times 5$ and $N_{min} = 1$. The mask in Figure 4.32 (c) is obtained by setting N_{min} to 4 for a neighborhood size of 9×9. It can be seen that the latter inpainting mask contains more regions that stem from a high background variation. The resulting images are given underneath and the corresponding ground truth image is shown in Figure 4.32 (d). The restoration quality is mainly affected by the false positives found, since similar true positives are detected. Therefore, the usage of $N_w = 5 \times 5$ leads to a PSNR value of 24.72 dB, whereas a similarity of 24.44 dB is gained for $N_w = 9 \times 9$.

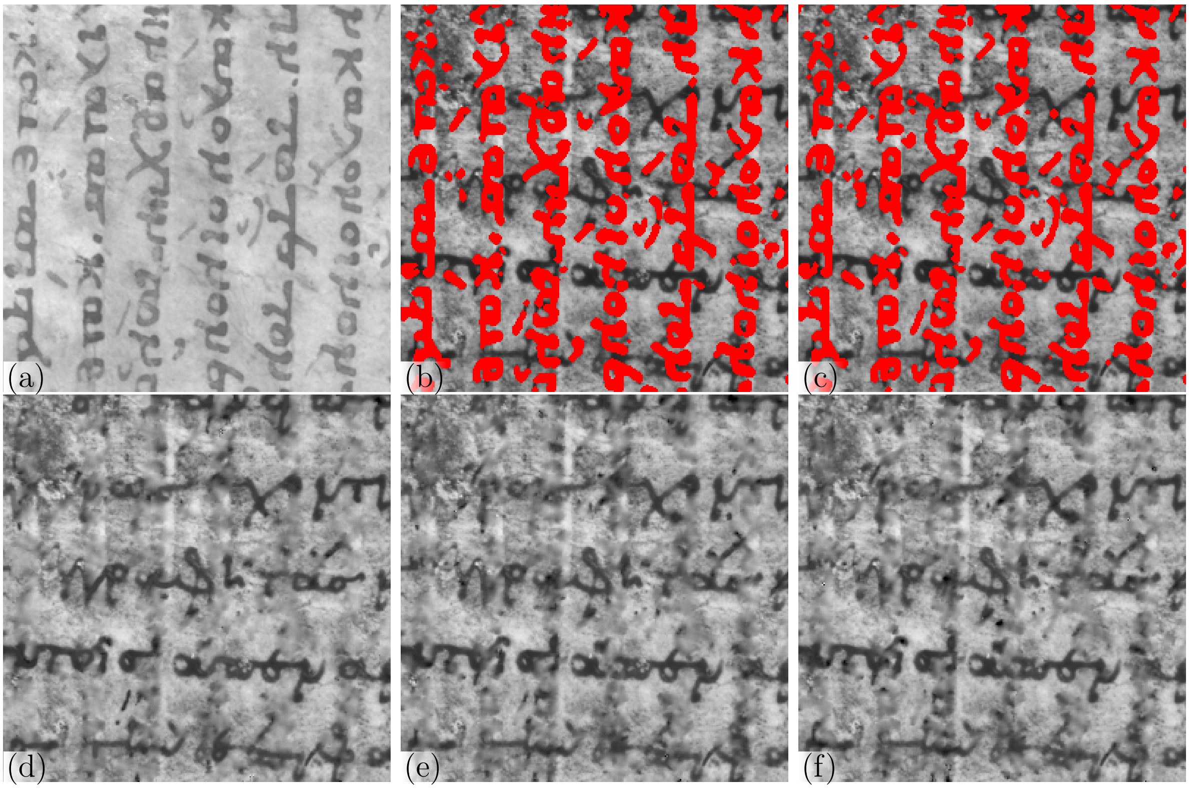


Figure 4.32: Retouching panel 48 verso with different binarization parameters. (a) Tungsten illuminated photography. (b) UV image with superimposed inpainting mask. The mask was generated using $N_w = 5 \times 5$ and $N_{min} = 1$. (c) Inpainting mask obtained by applying $N_w = 9 \times 9$ and $N_{min} = 4$. (d) Ground truth image. The restorations of the inpainting problems, shown in (b) and (c), are given underneath in (e) and (f).

Contrary to panel 48 verso, the other panels benefit from an increased window size, due to the more varying contrast between foreground and background pixels. A portion of leaf 58 verso is shown in Figure 4.30. The images in the middle column are generated by applying $N_w = 9 \times 9$ and the images in the right column are produced by utilizing a window length of 15. The tungsten illuminated photography of the parchment is given in Figure 4.30 (a). Therein a narrow stroke is marked by a green circle. It can be seen in Figure 4.30 (b) that only the upper part of this stroke is segmented, if N_w is set to 9×9. This is because the boundary pixels of the lower stroke part are not high contrast pixels. Those pixels are only marked as foreground pixels if a relatively large neighborhood - e.

g. $N_w = 15 \times 15$ - is considered, as it can be seen in Figure 4.30 (c). However, the usage of the larger neighborhood size leads to an increased sensitivity to noise, as it can be seen in the bottom left of Figure 4.30 (c). This image part contains background pixels that are wrongly classified, because they are in the near of high contrast pixels, which are located at the stroke boundaries. The recovery with $N_w = 9 \times 9$ yields a PSNR value of 25.42 dB, while for $N_w = 15 \times 15$ a similarity of 25.09 dB is gained.

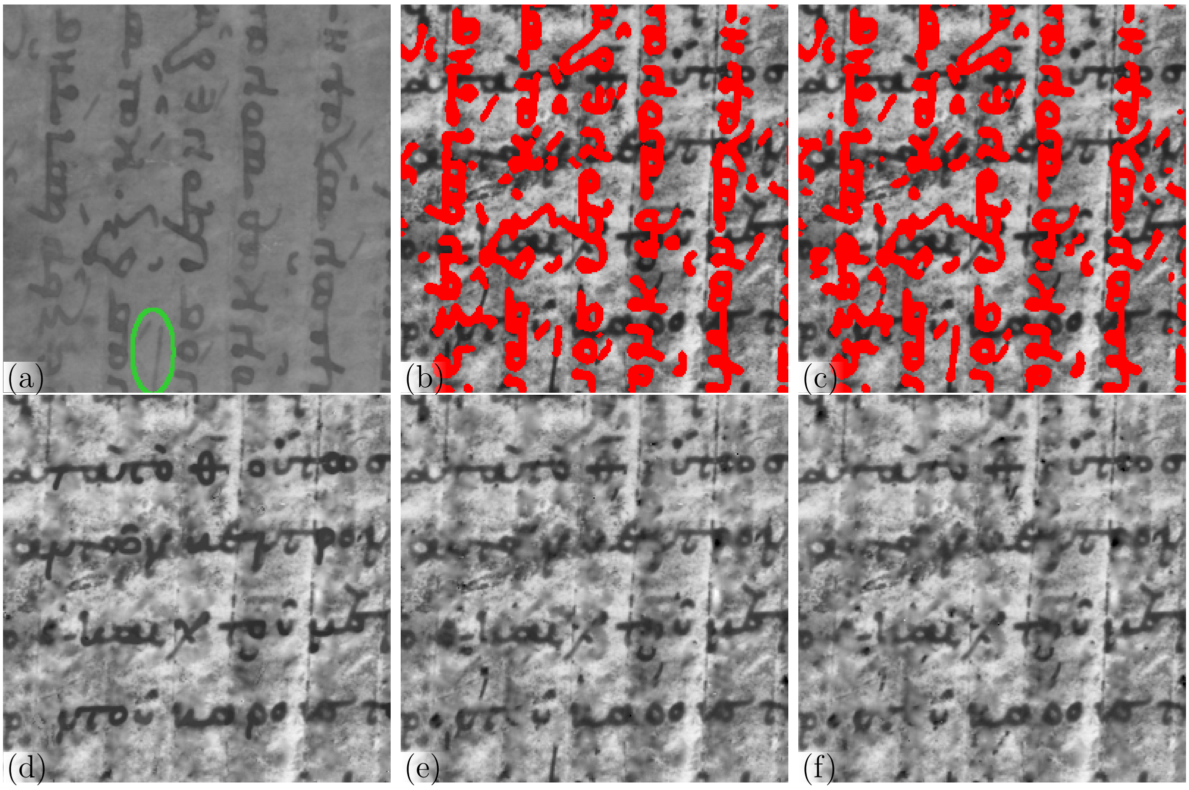


Figure 4.33: Retouching panel 58 verso with different binarization parameters. (a) Tungsten illuminated photography. (b) UV image with superimposed inpainting mask. The mask was generated using $N_w = 9 \times 9$ and $N_{min} = 4$. (c) Inpainting mask obtained by applying $N_w = 15 \times 15$ and $N_{min} = 8$. (d) Ground truth image. The restorations of the inpainting problems, shown in (b) and (c), are given underneath in (e) and (f).

The recovery of panel 99 verso is extremely dependent on the binarization parameters: The lowest similarity, namely 25.56 dB, is achieved for a window length of 5. For $N_w = 9 \times 9$ a PSNR value of 26.97 dB is gained. Portions of the generated masks and inpainted images are given in Figure 4.34. The mask in Figure 4.34 (b) is generated by using a window length of 5, whereas the domain in Figure 4.34 (c) is obtained by utilizing a window length of 9. The former domain is inappropriate, since it contains more false negatives, which are located at the centers of stroke regions[5]. Those foreground pixels are misclassified, because they are not in the immediate near of high contrast pixels. Therefore, more vestiges of the overwritings are remaining in the inpainted image. Those

[5] Figure 4.34 (b) contains 33 holes located at stroke centers, while in Figure 4.34 (c) only 3 holes are present.

remaining parts lead to the introduction of non existing structures.

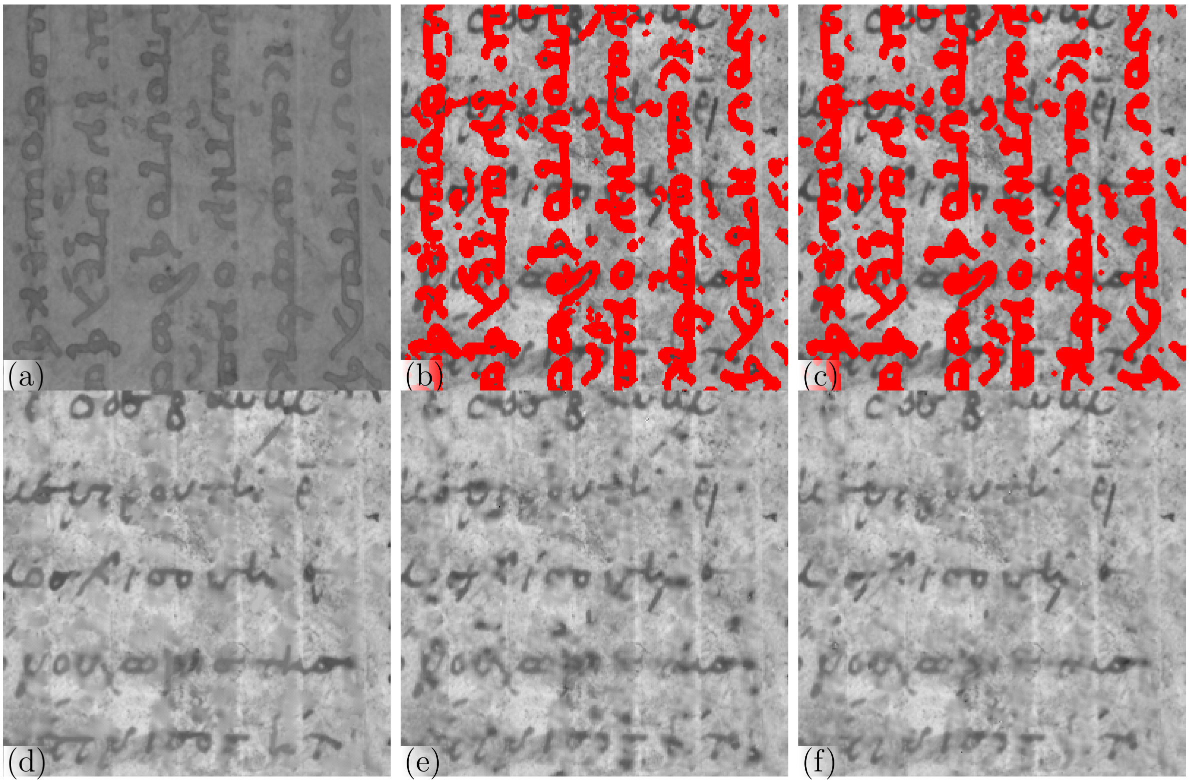


Figure 4.34: Retouching panel 99 verso with different binarization parameters. (a) Tungsten illuminated photography. (b) UV image with superimposed inpainting mask. The mask was generated using $N_w = 5 \times 5$ and $N_{min} = 1$. (c) Inpainting mask obtained by applying $N_w = 9 \times 9$ and $N_{min} = 4$. (d) Ground truth image. The restorations of the inpainting problems, shown in (b) and (c), are given underneath in (e) and (f).

The results show that the binarization algorithm suggested by Su et al. [SLT10] is capable to identify strokes with a relatively low contrast to the background. The implemented algorithm depends heavily on the parameter combination of N_w and N_{min}. The parameters have to be chosen according to the stroke width. The best overall performance is gained by setting N_w to 9×9 and N_{min} to 4. Although the highest mean PSNR value is achieved by this parameter setting, it has to be mentioned that the binarization algorithm is still sensitive to varying background regions. Especially noisy regions, which are in the near of high contrast pixels, are likely to get classified as foreground regions. It was also found that faded-out stroke endings are wrongly classified, if they are not in the near of high contrast pixels. Therefore, the usage of the aforementioned parameters leads to a trade-off between noise insensibility and the ability to identify foreground pixels. Since the former mentioned parameter combination gains the highest average performance - in terms of PSNR - it will be used for the generation of the inpainting masks that are utilized in the following experiments.

Mask Dilation

The masks, which were utilized in the preceding experiment, were dilated by a circular Structuring Element (SE). This measurement causes a loss of known pixels, if a foreground region found in the binarization step is already covering an entire character. Nevertheless, it is indispensable to enlarge the mask as it is shown in the following experiment, where the inpainting masks have been modified in the following ways: Firstly, the binary images, which were produced by the binarization algorithm, have been used as inpainting masks. Secondly, the binary images have been dilated by one pixel. Furthermore, the binarization outputs were dilated by two pixels. The dilatations were fulfilled by using disk-shaped SE. Each image was inpainted with η set to 100 and the resulting PSNR values are given in Table 4.8.

Mask dilation \Panel	40 recto	48 verso	58 verso	99 verso
No dilation	17.05 dB	16.85 dB	17.76	20.98 dB
SE radius = 1	23.13 dB	24.44 dB	25.42	26.97 dB
SE radius = 2	23.76 dB	24.10 dB	25.08	24.08 dB

Table 4.8: Evaluation of the mask dilation step.

The restoration of masks, which have not been post processed, leads to the worst results for each panel. The similarity is significantly increased, if the domains are enlarged by one pixel. This can be attributed to the fact that pixels, which are immediately touching the inpainting mask, have the largest influence on the inpainting result. Therefore, strokes belonging to the younger text are often propagated, if they are adjoining the mask regions. The dilation step helps to avoid this introduction of not existing structures.

Table 4.8 shows that a dilation with a SE, which has a radius of one, leads to the highest performance in three of the four cases investigated. Only for panel 40 recto a higher similarity is gained by a dilation with the larger SE. It was already shown above that the lowest inpainting performance is gained for this panel. The low performance can be attributed to the fact that the parchment is in a poorer condition than the remaining folios, since the contrast between fore- and background is smaller than in the other investigated parchments. The bad state of folio 40 recto reduces the inpainting quality because of two reasons: On the one hand, the background in the UV image is strongly varying. On the other hand, the image, which is used in the mask generation step, contains faded-out characters that are not fully segmented. Hence, less overwritten areas are correctly classified, compared to the other panels. The remaining vestiges of the overwritings and the dark background regions are propagated during the inpainting process. In the case of panel 40 verso, a dilation by one pixel is not sufficient to avoid the introduction of artifacts and a dilation by two pixels is more appropriate, as it is indicated by the higher PSNR value. Contrary, the highest performances for the other panels are gained, if the masks are enlarged by one pixel.

Figure 4.35 illustrates, why it is necessary to enlarge the mask. The restoration of panel 40 recto, given in Figure 4.35 (e), is produced by using the binary image that is produced by the segmentation method. The restoration of the dilated mask, shown in Figure 4.35 (f), is superior to the former mentioned image. The recovery of the dilated mask contains fewer artifacts than the restoration of the untampered mask. However, it can also be seen that remaining overwritings vestiges are even propagated, if the mask regions are dilated by one pixel.

4.3.3 Evaluation of the Inpainting Step

This section provides an evaluation of the inpainting sequence. In the following experiment the sensitivity to the inpainting rate is evaluated. The second part of this section provides a comparison to inpainting results that are obtained by a prior, which was trained on synthetic data. The main characteristics of the palimpsest recovery were already explained in the preceding experiments. This section is shortened, compared to the previous section, since it was already shown that the proposed system fails at recovering palimpsests in an adequate manner. Two possibilities of improvement are discussed at the end of this section.

Inpainting Rate

The four test panels have been restored using 10 different inpainting rates - ranging from 75 to 120. The PSNR values inside the inpainting regions are presented in Table 4.9. It turned out that the inpainting rate has only a minor impact on the restoration results. This property was already observed on natural images and on synthetic text data. It is not evident from Table 4.9 which inpainting rate should be used in general, since the highest performances are gained by diverse rates. The inpainted images were qualitatively analyzed by the thesis author, but it was also not found that a particular inpainting rate should be favored, since the outputs look akin, regardless of the inpainting rate utilized.

Rate	75	80	85	90	95	100	105	110	115	120
40r	18.53	18.56	**18.56**	18.56	18.56	18.52	18.51	18.51	18.46	18.42
48v	19.48	19.48	**19.49**	19.46	19.45	19.46	19.44	19.45	19.45	19.43
58v	20.32	20.33	20.35	20.37	20.37	20.36	20.38	**20.39**	20.36	20.35
99v	23.05	23.07	23.08	**23.10**	23.07	23.04	23.01	23.02	22.97	22.90

Table 4.9: Varying the inpainting rate η. The maximum PSNR value in each row is printed in bold.

The greatest performance difference - namely 0.2 dB - arises for panel 99 verso. A portion of this panel is given in Figure 4.36. The image in Figure 4.36 (c) is inpainted using a rate of 75, while the image in Figure 4.36 (d) was restored with $\eta = 120$. The differences between both panels become only apparent at a closer look. Therefore, a region, which is restored differently, is marked by the green circle in Figure 4.36 (b). However, the images exhibit that the inpainting rate has hardly any influence on the recovery performance. Therefore, it can be concluded that the restoration quality depends mainly on the binarization step and on the trained prior. The influence of the image prior is evaluated in the following and concluding experiment.

Comparison to a Prior trained on Synthetic Data

The inpainted underwritings that have been presented so far were produced by a prior, which was trained on historical handwritings. It was shown that this prior produces modest results, because gaps are only partially closed and remaining parts of overwritings are propagated into the inpainting domain. Several FoE priors, which model the statistics of the underwritings, have been trained during the course of this thesis. Those models are not able to overcome the aforementioned drawbacks and their inpainting performances are similar to the performance of the presented prior.

The restorations produced by the image model, which was utilized for the recovery of the artificial palimpsests, are on the contrary different. This image model was applied for the recovery of the four real palimpsests. The inpainting rate was set to 100 and the gained results are presented in Table 4.10.

Training set \Panel	40 recto	48 verso	58 verso	99 verso
IAM database	19.76	20.64	21.18	23.94
Palimpsest	18.52	19.46	20.36	23.04

Table 4.10: Palimpsest reconstruction using priors that trained on diverse handwriting datasets.

The results are counter-intuitively, since the model that was trained on the IAM database gains higher PSNR values than the model, which was trained on the ancient writings. The higher performances can be attributed to the circumstance that the former mentioned FoE prior produces smoother inpainting results. The model is less sensitive to varying background regions and residuals of overwritings, but it tends to oversmooth the inpainting regions. This can be seen in Figure 4.37 and Figure 4.38, where parchments portions are given that are restored by both image models. It is apparent that the prior, which is trained on ancient underwritings tends to favor a propagation of dark regions. This leads to an introduction of not existing structures. The FoE prior that was instead trained on the IAM database produces fewer artifacts. The inpainting regions that are restored by this model are blurred out and clearly distinguishable from the surrounding regions.

The circumstance that the prior, which is trained on underwriting patches, introduces more artifacts indicates that the FoE models are extremely sensitive on the nature of the utilized training patches. The background in the underwriting patches is highly varying, while the background of the synthetic patches is uniformly white. It can be assumed that the model, which was trained on the IAM database, favors therefore homogeneous inpainting regions, while the other prior favors varying regions.

The numerical results suggest that the underwriting model is overwhelmed by the prior trained on the IAM database. However, a qualitative analysis shows that the performances of both models are limited. Neither the prior that models Latin letters, nor the model, which was trained on the ancient underwritings, are able to produce convincing palimpsest restorations.

It has to be mentioned that the inpainting quality can be increased by making use of the full potential of the FoE framework. Roth and Black suggest the following measures in order to enhance the denoising and inpainting performance:

- Roth [Rot07] reports that the usage of 5 × 5 models leads to a higher performance compared to 3 × 3 filters. The greater filters capture long-range dependencies in a more accurate way.

- Roth and Black [RB05b] propose a whitening transformation of the training data. This measurement allows for a better recovery of high-frequency domains. Additionally, an insensibility to brightness changes is achieved. This step can reduce the sensitivity to varying background regions.

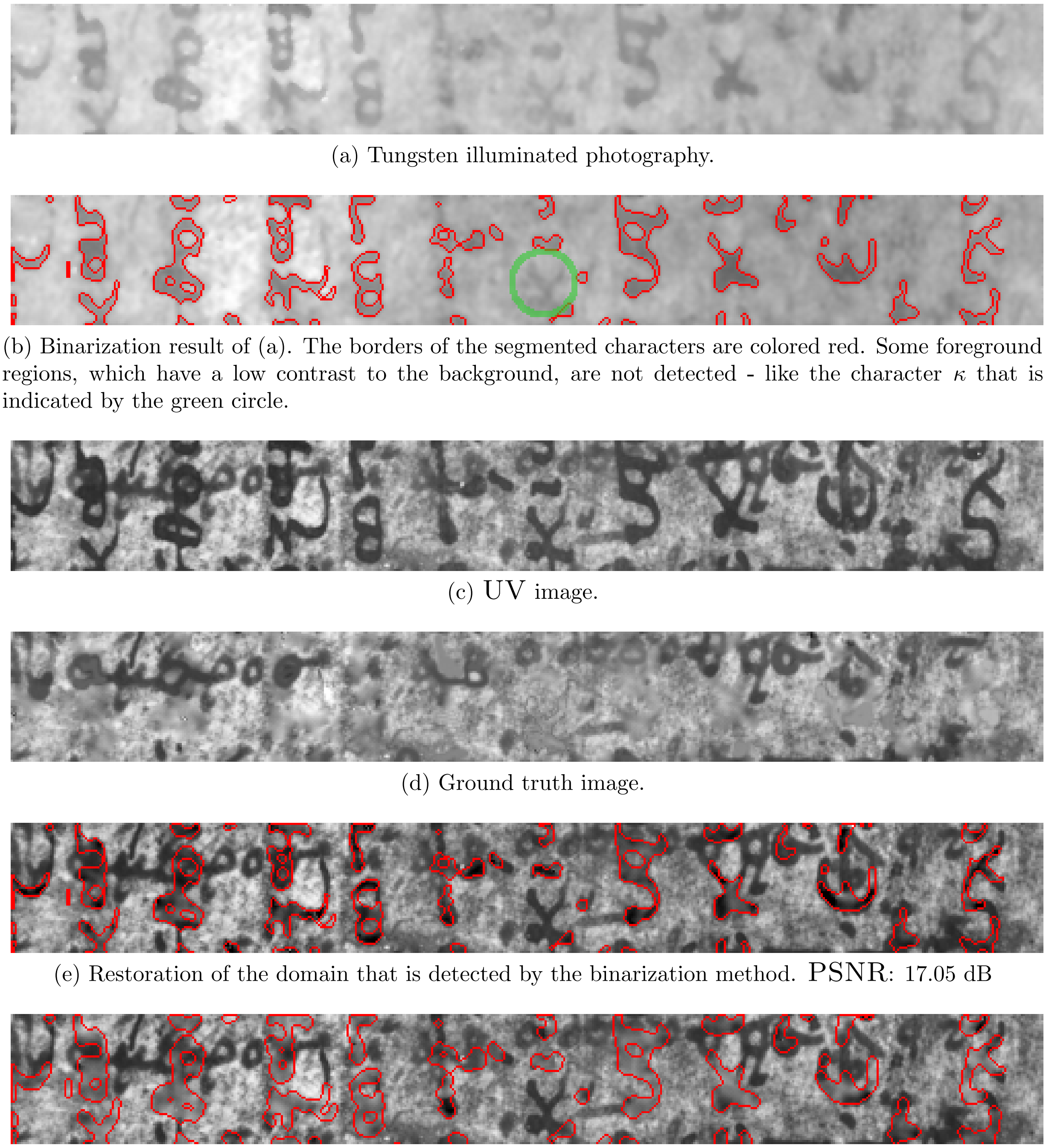
The extensions described were not successfully implemented, due to time restrictions. The capacity of the system would certainly benefit from a realization of the mentioned possibilities of improvement. However, it can be assumed that the FoE framework is generally not suitable for the reconstruction of the utilized palimpsests at the original resolution. This statement is based on the circumstance that the FoE algorithm is designed for small inpainting regions [Woh09], which is typical for geometrical inpainting methods. Since these inpainting techniques are in general limited to small and narrow inpainting domains [CPT04], [KT07], [CK06], a textural inpainting algorithm is more appropriate for the automated recovery of unknown palimpsest regions.

4.4 Summary

The performances of the investigated methods were evaluated in this chapter. The first subsection dealt with the inpainting of natural images. It turned out that the FoE inpainting technique is capable of filling thin and narrow domains successfully. This is a typical property of geometrical inpainting techniques [CK06] and the algorithm produces similar or slightly inferior results than other algorithms - [BSCB00], [OBMC01] and [BLC08] - falling in this category, as it was shown at the end of the subsection. The recovery of larger mask regions is in contrast evident in the restored images, especially if the surrounding image regions are textured.

Secondly the recovery of modern handwritings was analyzed. It turned out that the statistical based inpainting method favors a propagation of strokes, which have a similar orientation as the main orientation of the strokes contained in the training database. The performance of the statistically based technique was compared to the performance of the heuristical based algorithm. The latter one needs less time for the reconstruction task, but the retouched images are less visually plausible than the images that are produced by the FoE algorithm.

Finally, the recovery of ancient underwritings was analyzed. It was shown that the FoE algorithm is incapable of restoring the palimpsest images at the original resolution and that a down sampling of the images used is necessary. The parameter dependency of the binarization algorithm was evaluated by comparing restored images with their ground truth counterparts. It was shown that the trained FoE prior is sensitive to dark image regions and that a dilation of the inpainting mask is needed in order to avoid a propagation of overwriting vestiges into the inpainting domain. An FoE prior, which was in contrast trained on synthetic data is less sensitive to such vestiges, but blurs out the inpainting region. The outputs produced by both priors are generally not satisfying and hence the system is currently inapplicable for a subsequent analysis by scholars.



(a) Tungsten illuminated photography.

(b) Binarization result of (a). The borders of the segmented characters are colored red. Some foreground regions, which have a low contrast to the background, are not detected - like the character κ that is indicated by the green circle.

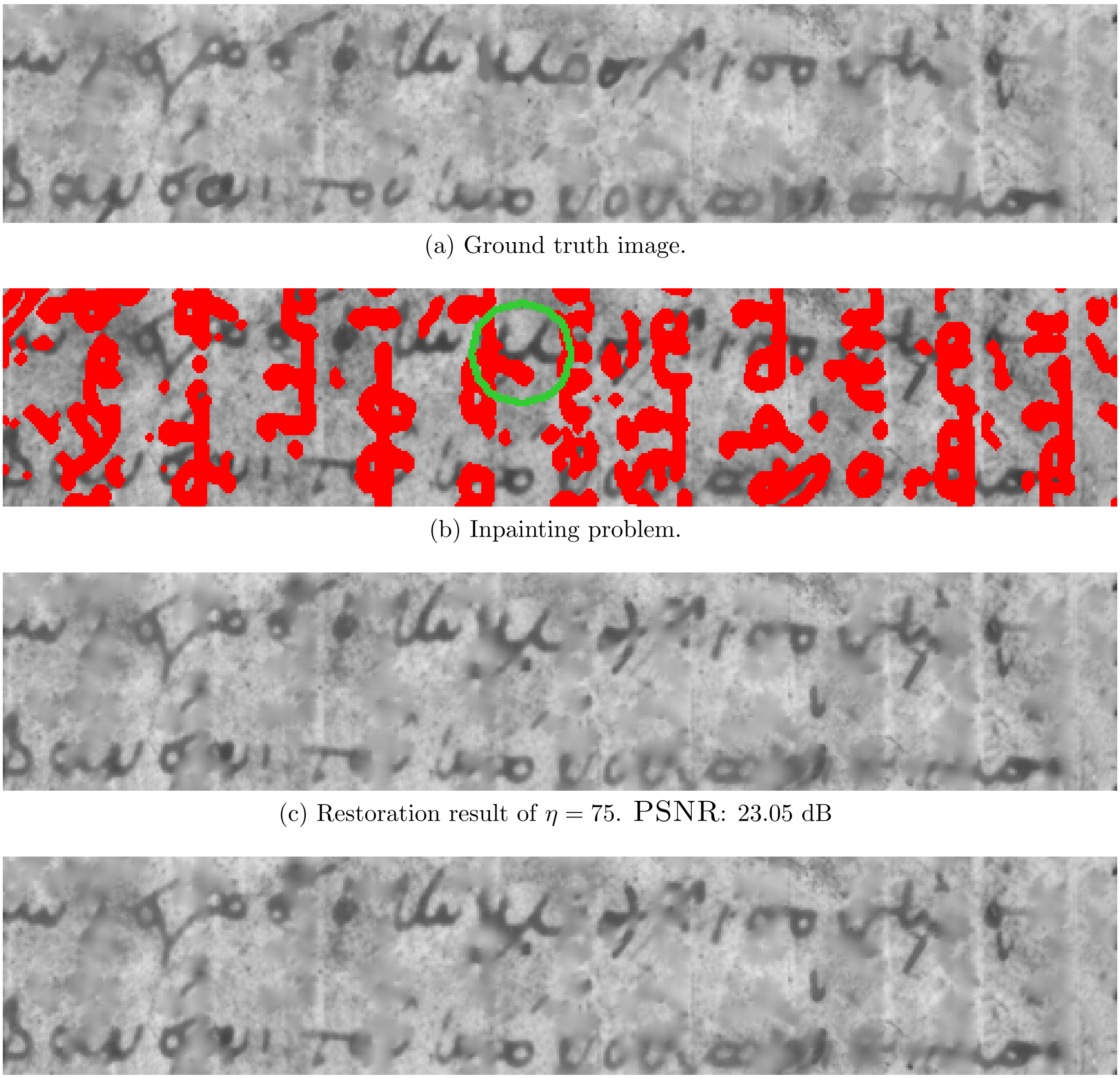
(c) UV image.

(d) Ground truth image.

(e) Restoration of the domain that is detected by the binarization method. PSNR: 17.05 dB

(f) The segmentation result was dilated by one pixel, before the inpainting task took place. PSNR: 23.13 dB

Figure 4.35: Restorations of leaf 40 recto, showing the necessity for the mask enlargement step. The retouching of the mask, which is detected by the binarization algorithm, is provided in (e). This mask was dilated by one pixel and the resulting recovery is given in (f).



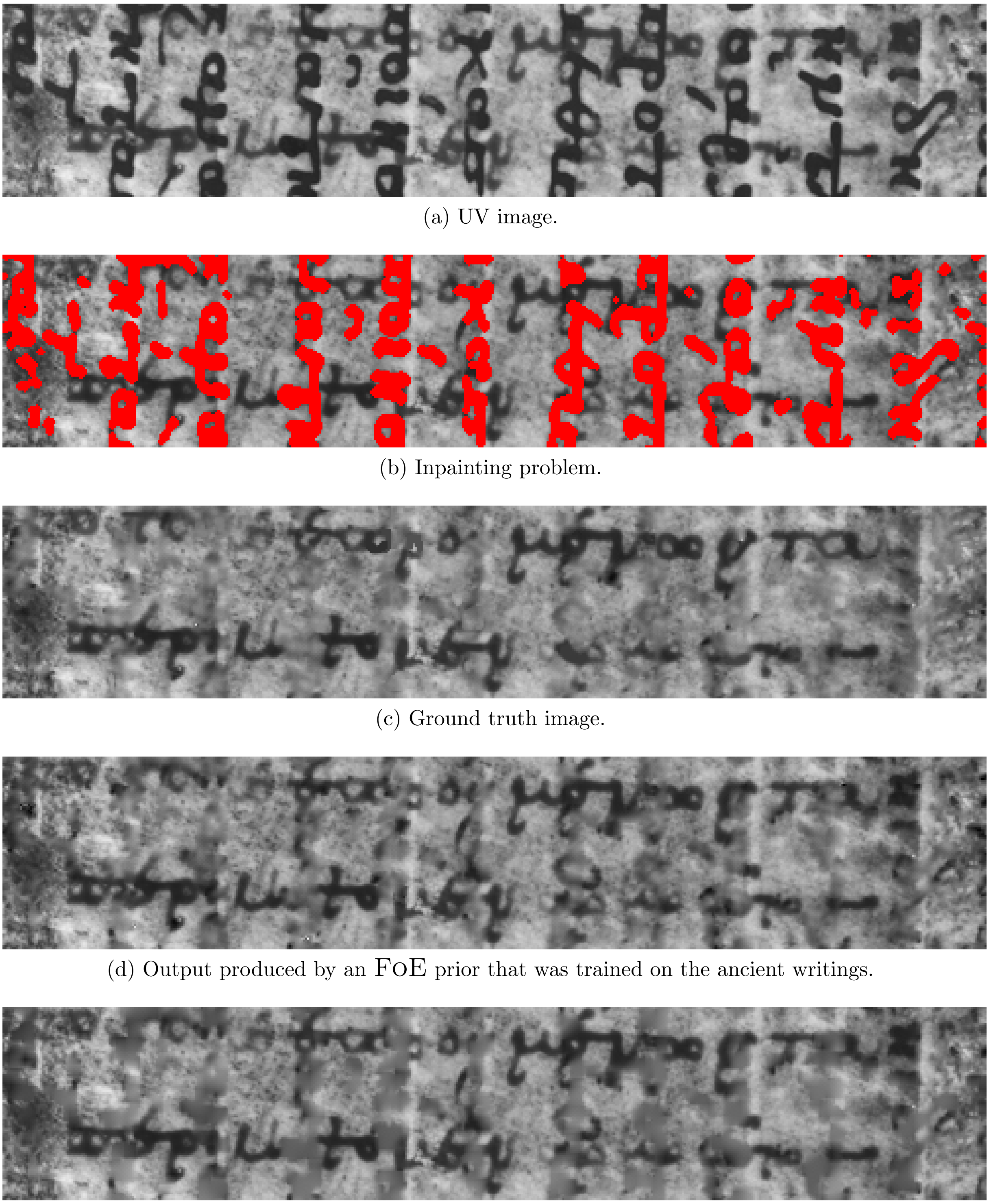
(a) Ground truth image.

(b) Inpainting problem.

(c) Restoration result of $\eta = 75$. PSNR: 23.05 dB

(d) Inpainting fulfilled with $\eta = 120$. PSNR: 22.90 dB

Figure 4.36: Influence of the inpainting rate. There are only minor differences between (c) and (d). A region, where the difference is recognizable, is marked by the green circle in (b).



(a) UV image.

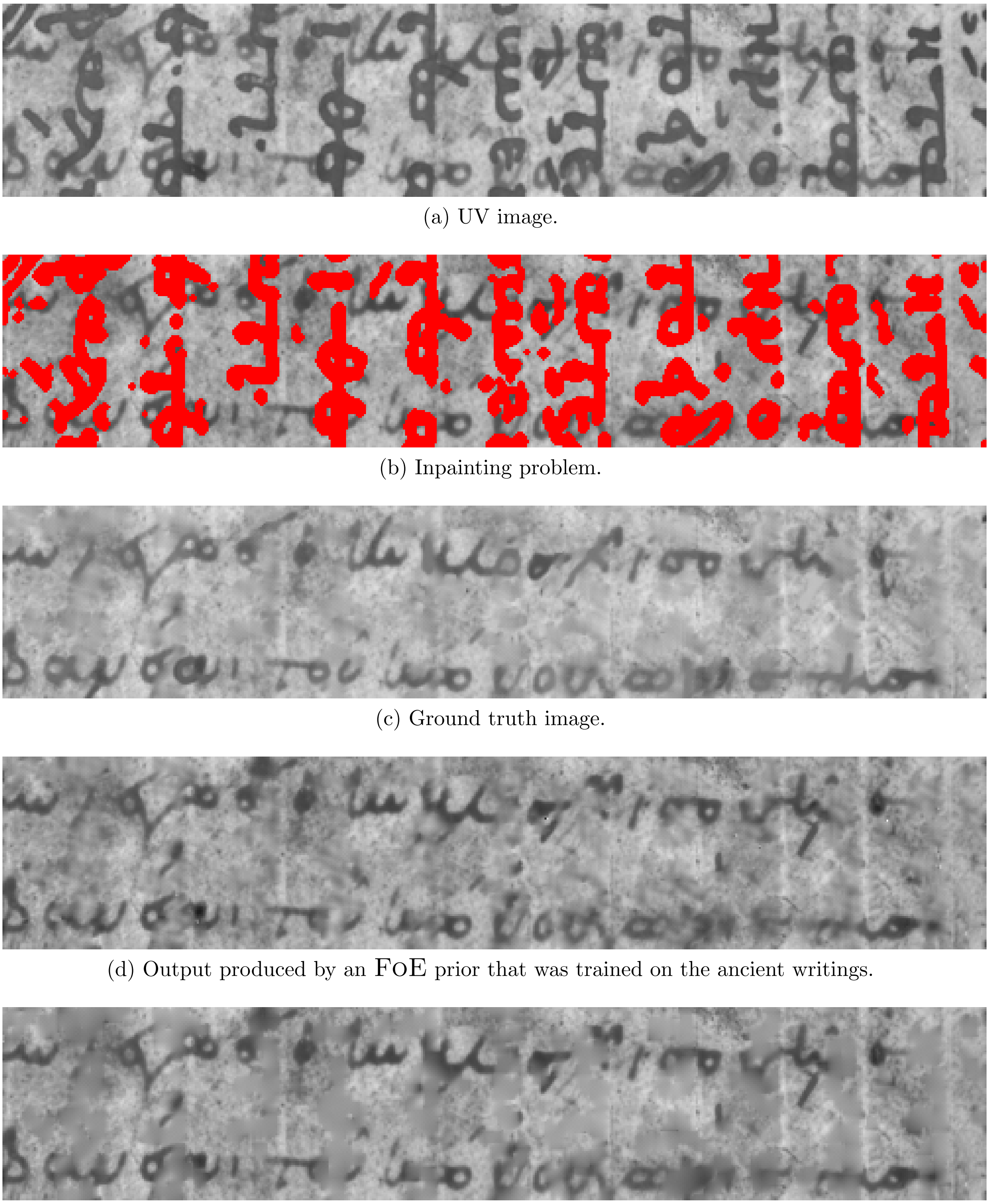
(b) Inpainting problem.

(c) Ground truth image.

(d) Output produced by an FoE prior that was trained on the ancient writings.

(e) Inpainting result of an FoE prior, which was trained on the IAM database.

Figure 4.37: Restorations of panel 48 verso, produced by different handwriting models.



(a) UV image.

(b) Inpainting problem.

(c) Ground truth image.

(d) Output produced by an FoE prior that was trained on the ancient writings.

(e) Inpainting result of an FoE prior, which was trained on the IAM database.

Figure 4.38: Restorations of panel 99 verso, produced by different handwriting models.

Chapter 5

Conclusion

This work was concerned with the automated filling of unknown regions in natural images and handwritings. The restoration of modern handwritings was investigated, as well as the recovery of palimpsests. Palimpsests are ancient writings, which were overwritten and the system proposed aims at restoring the overwritten regions automatically in order to enhance the legibility of the underwritings.

This task is stated as an inpainting problem and hence it is necessary to generate an inpainting mask, which encodes the locations of the overwritten image regions. The mask is generated by applying a binarization method on photographs of parchments that have been exposed to tungsten illumination. Such photographs exhibit only the overwritten regions, while the underwritings are mostly visible under UV light. Therefore, the inpainting process is carried out on photographs of leafs, which have been illuminated with UV light.

The inpainting of handwritings and natural images is fulfilled with the FoE framework, which is an approach for MRF modeling. The FoE framework allows for an offline learning of image statistics that can be used in inference tasks, including image inpainting. The equation, which is used in the inpainting sequence, is based on the MAP approach and makes use of a trained FoE prior. The priors, which were used in the presented experiments, have been trained on natural images and modern and ancient handwritings.

A second algorithm was evaluated, which is exclusively designed for the restoration of recent handwritings. The proposed algorithm relies on an ad-hoc principle, namely tensor voting: The essential idea of two-dimensional tensor voting is to encode the direction of a vector perpendicular to a curve tangent into a corresponding tensor and to subsequently cast the information of this tensor into its local neighborhood. Thus, a grouping of together belonging tokens is achieved. The presented method makes use of two-dimensional tensor voting in order to propagate interrupted edges into the unknown image regions.

The performances of both algorithms were analyzed in several experiments. Firstly, the inpainting of natural images - with the FoE approach - was investigated. Special attention was paid to the question, which kind of inpainting masks can be restored in an adequate manner. Hence, three different mask sets have been generated for 44 input images. The performance was evaluated by comparing undamaged input images with the corresponding restoration results. The similarity was measured with the PSNR. It turned out that the capacity of the inpainting approach is dependent on the width of the distinct mask regions: The highest performance was gained for the mask type, which contained the thinnest regions, while the restoration of the thickest regions led to the lowest similarity values. The tests showed that the FoE approach is capable of producing convincing inpainting results if the mask regions are thin: For 5 pixels thick mask regions an average PSNR value of 32.25 dB was gained. Contrary, the inpainting process becomes evident if the inpainting regions are thicker: The recovery of 11 pixels wide regions led to a mean PSNR value of 31.37 dB. The recovery of large and textured regions is better fulfilled by the inpainting techniques proposed in [Tsc06] and [YSM10] - as it was shown in Section 4.1.3.

Secondly the recovery of synthetic handwriting data was analyzed. It was shown that FoE priors learn the main orientation of writings, which can be attributed to the rotation variance of MRF models. Furthermore it was demonstrated that priors, which model the statistics of handwritten text, are better suited for the recovery of images containing handwritings than models that have been trained on natural images. The performance of the tensor voting based recovery system was also investigated. It was shown that the performance of the approach is dependent on a parameter, which defines the scale of voting. The current implementation suffers from the drawback that the same parameter value is used for one input image, while it would be more appropriate to adapt the scale of voting to the extent of the inpainting region. The section, which deals with the recovery of modern writings, was concluded by a comparison of both restoration techniques. It turned out that the tensor voting based approach is clearly overwhelmed by the statistical based technique: For example, the heuristical approach gained an average similarity of 16.43 dB for the recovery of 5 pixels wide overwritings, while the FoE approach achieved an average PSNR value of 20.15 dB for the same class of inpainting problems. The performances of both techniques are indirectly proportional to the width of the overwritings. Hence, the restoration of the thickest overwritings - with an average width of 8 pixels - led to the worst results: The heuristical approach gained a similarity value of 14.00 dB for this overwriting class, whereas the statistical based inpainting method achieved a performance of 16.5 dB. Although the FoE approach gained a considerably higher performance for this mask class, the inpainting process is observable, since mask centers are not altered and the inpainted strokes are blurred.

The circumstance that the FoE is only capable of restoring small mask regions in a convincing manner became also apparent, when the recovery of palimpsest underwritings was analyzed. The resolution of the photographs utilized is around 700 dpi[1]. The photographs had to be resampled using a factor of 0.25 in order to assure that the mask centers are altered adequately. The results gained on the downsized images showed that some occluded character portions are successfully restored by the inpainting technique, while others are not altered adequately. A drawback of the proposed method is the circumstance that it cannot distinguish between underwritings and vestiges of overwritings that have not been detected in the mask generation step. If such a leftover of the younger text is bordering the inpainting region it is likely that the vestige is propagated into the inpainting region. Therefore, it was necessary to enlarge the binarization result by one pixel, before the inpainting process was started. Otherwise pixels belonging to the

[1] From: http://www.archimedespalimpsest.org/imaging_production1.html, last accessed on 11. August 2011

overwriting boundaries might not be covered by the inpainting mask and the FoE prior would continue this vestiges into the inpainting regions. The average PSNR value that was gained inside the inpainting regions is 20.34 dB and the mean similarity of the entire test images is 24.99 dB. Because of the disadvantage that the images had to be downsized, the palimpsest recovery system is currently inapplicable for a subsequent text analysis by philologists.

Future Work

The performance of the system could be improved by using larger clique sizes and by a whitening of the training data. A further measure of improvement is suggested in [HWH09]: Heess et al. suggest using bimodal potential functions instead of unimodal Student-t distributions. The authors report that such functions are more accurate for the modeling of natural images and that the inpainting performance can be increased significantly. The recovery of handwritings could also benefit from the use of bimodal potential functions.

The restoration of handwritings could also be improved by using a textural inpainting technique instead of a geometrical inpainting approach. It was depicted in Section 2 that algorithms falling into the former mentioned group are not limited to narrow inpainting domains. Hence, a preceding downsizing of the images could be avoided. This would meet the requirements by scholars, who prefer to analyze high-resolution images [FK06].

List of acronyms

ANN	Approximate Nearest Neighbors
BP	Belief Propagation
CD	Contrastive Divergence
DPI	Dots per Inch
FoE	Fields of Experts
HMC	Hybrid Monte Carlo
MAP	Maximum A Posteriori
MC	Markov Chain
MCMC	Markov Chain Monte Carlo
ML	Maximum Likelihood
MRF	Markov Random Field
MSE	Mean Squared Error
MSI	Multi-Spectral Images
OCR	Optical Character Recognition system
PDE	Partial Differential Equation
PSNR	Peak Signal-to-Noise-Ratio
SSD	Sum of Squared Differences
SE	Structuring Element
TV	Total Variation
UV	UltraViolet

Bibliography

[ALM10] Jean-François Aujol, Saïd Ladjal, and Simon Masnou. Exemplar-based inpainting from a variational point of view. *SIAM J. Math. Analysis*, 42(3):1246–1285, 2010.

[Ash01] Michael Ashikhmin. Synthesizing natural textures. In *SI3D*, pages 217–226, 2001.

[BBC$^+$01] Coloma Ballester, Marcelo Bertalmío, Vicent Caselles, Guillermo Sapiro, and Joan Verdera. Filling-in by joint interpolation of vector fields and gray levels. *IEEE Transactions on Image Processing*, 10(8):1200–1211, 2001.

[BBS01] M. Bertalmio, A.L. Bertozzi, and G. Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages I:355–362, 2001.

[BEG07] Andrea L. Bertozzi, Selim Esedoglu, and Alan Gillette. Inpainting of binary images using the Cahn-Hilliard equation. *IEEE Transactions on Image Processing*, 16(1):285–291, 2007.

[Ber07] Uwe Bergmann. Archimedes brought to light. *Physics World*, 20:39–42, 2007.

[Bes86] Julian Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, 48(3):259–302, 1986.

[BGS05] Alfio Borzì, Harald Grossauer, and Otmar Scherzer. Analysis of iterative methods for solving a Ginzburg-Landau equation. *International Journal of Computer Vision*, 64(2-3):203–219, 2005.

[BHS09] Martin Burger, Lin He, and Carola-Bibiane Schönlieb. Cahn–Hilliard inpainting and a generalization for grayvalue images. *SIAM J. Imaging Sciences*, 2(4):1129–1167, 2009.

[BLC08] Carlos Brito-Loeza and Ke Chen. Multigrid method for a modified curvature driven diffusion model for image inpainting. *J. Comp. Math.*, 26, Number 6:856–875, 2008.

[BLLC02] Raphaél Bornard, Emmanuelle Lecan, Louis Laborelli, and Jean-Hugues Chenot. Missing data correction in still images and image sequences. In *ACM Multimedia*, pages 355–361, 2002.

[Bov05] Alan C. Bovik. *Handbook of Image and Video Processing (Communications, Networking and Multimedia).* Academic Press, Inc., Orlando, FL, USA, 2005.

[BSCB00] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *SIGGraph-2000*, pages 417–424, 2000.

[BSFG09] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28(3), August 2009.

[BVSO03] Marcelo Bertalmío, Luminita A. Vese, Guillermo Sapiro, and Stanley Osher. Simultaneous structure and texture image inpainting. *IEEE Transactions on Image Processing*, 12(8):882–889, 2003.

[Can86] J Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986.

[CG09] Huaigu Cao and Venu Govindaraju. Preprocessing of low-quality handwritten documents using markov random fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(7):1184–1194, 2009.

[CK06] Tony F. Chan and Sung Ha Kang. Error analysis for image inpainting. *Journal of Mathematical Imaging and Vision*, 26(1-2):85–103, 2006.

[CKKS02] Tony F. Chan, Sung Ha Kang, Kang, and Jianhong Shen. Euler's elastica and curvature based inpaintings. *SIAM J. Appl. Math*, 63:564–592, 2002.

[CLK10] Jin Chen, Daniel P. Lopresti, and Ergina Kavallieratou. The impact of ruling lines on writer identification. In *ICFHR*, pages 439–444, 2010.

[CPT04] A. Criminisi, P. Perez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Trans. Image Processing*, 13(9):1200–1212, September 2004.

[CS00] T.F. Chan and J.H. Shen. Mathematical models for local deterministic inpaintings. Cam report 00-11, UCLA, 2000.

[CS01] T.F. Chan and J.H. Shen. Nontexture inpainting by curvature-driven diffusions. *Journal of Visual Communication and Image Representation*, 12(4):436–449, December 2001.

[Cuz09] Anne Cuzol. Stochastic methods for inpainting. Unpublished: `http://perception.inrialpes.fr/visiontrain/reports/Posdocreports/ACuzol/Postdoc-report-ACuzol.pdf`, 2009.

[CX10] Xiaowu Chen and Fang Xu. Automatic image inpainting by heuristic texture and structure completion. In *MMM*, pages 110–119, 2010.

[CZL07] Qiang Chen, Yingxiang Zhang, and Yuncai Liu. Image inpainting with improved exemplar-based approach. In *MCAM*, pages 242–251, 2007.

[DCOY03] Iddo Drori, Daniel Cohen-Or, and Hezy Yeshurun. Fragment-based image completion. *ACM Trans. Graph.*, 22(3):303–312, 2003.

[Dua87] S. Duane. Hybrid monte carlo. *Physics Letters B*, 195(2):216–222, September 1987.

[EF01] Alexei A. Efros and William T. Freeman. Image quilting for texture synthesis and transfer. In *SIGGRAPH*, pages 341–346, 2001.

[EKCB03] R.L. Easton, K.T. Knox, and W.A. Christens-Barry. Multispectral imaging of the Archimedes palimpsest. In *32nd Applied Imagery Pattern Recognition Workshop*, pages 111 – 116, 2003.

[EL99] Alexei A. Efros and Thomas K. Leung. Texture synthesis by non-parametric sampling. In *ICCV*, pages 1033–1038, 1999.

[ES02] S. Esedoglu and J.H. Shen. Digital inpainting based on the Mumford-Shah-Euler image model. *European Journal on Applied Mathematics*, 13:353–370, 2002.

[ESQD05] M. Elad, J.-L. Starck, P. Querre, and D.L. Donoho. Simultaneous cartoon and texture image inpainting using Morphological Component Analysis (MCA). *Applied and Computational Harmonic Analysis*, 19(3):340 – 358, 2005. Computational Harmonic Analysis - Part 1.

[Fid08] Işık Barış Fidaner. A survey on variational image inpainting, texture synthesis and image completion. Unpublished: `http://www.scribd.com/doc/3012627/`, 2008.

[FK06] C. Fischer and I. Kakoulli. Multispectral and hyperspectral imaging technologies in conservation: current research and potential applications. *Reviews in Conservation*, 7:3–16, 2006.

[FSM09] M.J. Fadili, J.-L. Starck, and F. Murtagh. Inpainting and zooming using sparse representations. *Comput. J.*, 52:64–79, January 2009.

[GG84] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, November 1984.

[Gre08] Benton Greene. Website, 2008. Image taken from `http://www.flickr.com/photos/beezum88/3028183796/`.

[Gui09] Richard Guillaume. Automatic inpainting of with fields of experts. Master's thesis, University of Manchester, 2009.

[Gul06] O.G. Guleryuz. Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising-part ii: adaptive algorithms. *IEEE Transactions on Image Processing*, 15(3):555 –571, 2006.

[GYZ$^+$09] Kai Guo, Xiaokang Yang, Rui Zhang, Songyu Yu, and Hongyuan Zha. Interpolating fine texturess with fields of experts prior. In *ICIP*, pages 353–356, 2009.

[HC71] John Hammersley and Peter Clifford. Markov field on finite graph and lattices. 1971.

[HHL$^+$08] Jason C. Hung, Chun-Hong Huang, Yi-Chun Liao, Nick C. Tang, and Ta-Jen Chen. Exemplar-based image inpainting base on structure construction. *JSW*, 3(8):57–64, 2008.

[Hin99] Geoffrey Hinton. Products of experts. In *ICANN*, volume 1, pages 1–6, 1999.

[Hin02] Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

[HLLC10] Heng-Feng Hsin, Jin-Jang Leou, Cheng-Shian Lin, and Hsuan-Ying Chen. Image inpainting using structure-guided priority belief propagation and label transformations. In *ICPR*, pages 4492–4495, 2010.

[HS11] Fabian Hollaus and Robert Sablatnig. Inpainting of Occluded Regions in Handwritings. In *OAGM/AAPR Workshop, Austrian Computer Society*, 2011.

[HWH09] Nicolas Heess, Christopher K. I. Williams, and Geoffrey E. Hinton. Learning generative texture models with extended fields-of-experts. In *BMVC*. British Machine Vision Association, 2009.

[IP97] Homan Igehy and Lucas Pereira. Image replacement through texture synthesis. In *ICIP (3)*, pages 186–189, 1997.

[JT03] Jiaya Jia and Chi-Keung Tang. Image repairing: Robust image synthesis by adaptive ND tensor voting. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages I: 643–650, 2003.

[KDH$^+$11] Florian Kleber, Markus Diem, Fabian Hollaus, Martin Lettner, Robert Sablatnig, Melanie Gau, and Heinz Miklas. Technical Approaches to Manuscript Analysis and Reconstruction. In *New Approaches to Book and Paper Conservation-Restoration*, pages 533–558. Verlag Berger Horn / Wien, 2011.

[KEBK05] Vivek Kwatra, Irfan A. Essa, Aaron F. Bobick, and Nipun Kwatra. Texture optimization for example-based synthesis. *ACM Trans. Graph.*, 24(3):795–802, 2005.

[Kno08] K. T. Knox. Enhancement of overwritten text in the Archimedes Palimpsest. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 6810 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, March 2008.

[KSE$^+$03] Vivek Kwatra, Arno Schödl, Irfan A. Essa, Greg Turk, and Aaron F. Bobick. Graphcut textures: image and video synthesis using graph cuts. *ACM Trans. Graph.*, 22(3):277–286, 2003.

[KT07] Nikos Komodakis and Georgios Tziritas. Image completion using efficient belief propagation via priority scheduling and dynamic pruning. *IEEE Transactions on Image Processing*, 16(11):2649–2661, 2007.

[LH08] Yunpeng Li and Daniel P. Huttenlocher. Learning for optical flow using stochastic optimization. In *ECCV (2)*, pages 379–391, 2008.

[LHW$^+$06] Wen-Chieh Lin, James Hays, Chenyu Wu, Yanxi Liu, and Vivek Kwatra. Quantitative evaluation of near regular texture synthesis algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition (1)*, pages 427–434, 2006.

[Li09] Stan Z. Li. *Markov Random Field Modeling in Image Analysis.* Springer Publishing Company, Incorporated, 2009.

[LLX$^+$01] Lin Liang, Ce Liu, Ying-Qing Xu, Baining Guo, and Heung-Yeung Shum. Real-time texture synthesis by patch-based sampling. *ACM Trans. Graph.*, 20(3):127–150, 2001.

[LZW03] Anat Levin, Assaf Zomet, and Yair Weiss. Learning how to inpaint from global image statistics. In *ICCV*, pages 305–312, 2003.

[Mas02] S. Masnou. Disocclusion: a variational approach using level lines. *Image Processing, IEEE Transactions on*, 11(2):68 –76, February 2002.

[MB02] U.-V. Marti and H. Bunke. The iam-database: an english sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, 5:39–46, 2002. 10.1007/s100320200071.

[MCSF06] Julian John McAuley, Tibério S. Caetano, Alex J. Smola, and Matthias O. Franz. Learning high-order mrf priors of color images. In *ICML*, pages 617–624, 2006.

[MFTM01] David R. Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. Technical Report UCB/CSD-01-1133, EECS Department, University of California, Berkeley, Jan 2001.

[MK04] Gerard Medioni and Sing Bing Kang. *Emerging Topics in Computer Vision.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 2004.

[MM98] S. Masnou and J. Morel. Level line(s) based disocclusion. In *ICIP-C 98*, pages 259–263, 1998.

[MS88] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and variational problems. *Communications on Pure and Applied Mathematics*, XLII(5):577–685, 1988.

[NAW04] Reviel Netz, Fabio Acerbi, and Nigel Wilson. Towards a reconstruction of Archimedes' Stomachion. *SCIAMVS*, 5:67–99, 2004.

[Nea93] Radford M. Neal. Probabilistic inference using Markov Chain Monte Carlo methods. Technical report, Dept. of Computer Science, University of Toronto, 1993.

[NMS93] M. Nitzberg, D. Mumford, and T. Shiota. *Filtering, Segmentation, and Depth.* Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1993.

[OBMC01] Manuel M. Oliveira, Brian Bowen, Richard McKenna, and Yu-Sung Chang. Fast digital image inpainting. In *VIIP*, pages 261–266, 2001.

[Ots79] N. Otsu. A threshold selection method from grey-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, January 1979.

[Pér98] P. Pérez. Markov Random Fields and images. *CWI Quarterly*, 11(4):413–437, 1998.

[RB05a] K. Rapantzikos and C. Balas. Hyperspectral imaging: potential in non-destructive analysis of palimpsests. In *IEEE International Conference on Image Processing*, volume 2, pages II – 618–21, 2005.

[RB05b] Stefan Roth and Michael J. Black. Fields of Experts: A framework for learning image priors. In *CVPR (2)*, pages 860–867, 2005.

[RG00] Sébastien Roy and Venu Govindu. MRF solutions for probabilistic optical flow formulations. In *ICPR*, pages 7053–7059, 2000.

[ROF92] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. In *Proceedings of the eleventh annual international conference of the Center for Nonlinear Studies on Experimental mathematics : computational issues in nonlinear science*, pages 259–268. Elsevier North-Holland, Inc., 1992.

[Rot07] Stefan Roth. *High-order Markov Random Fields for low-level vision.* PhD thesis, Providence, RI, USA, 2007.

[RRB05] Andrei Rares, Marcel J. T. Reinders, and Jan Biemond. Edge-based image restoration. *IEEE Transactions on Image Processing*, 14(10):1454–1468, 2005.

[Sch10] Uwe Schmidt. Learning and evaluating markov random fields for natural images. Master's thesis, Darmstadt University of Technology, 2010.

[Set98] J. A. Sethian. *Level Set Methods and Fast Marching Methods - Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press, 1998.

[SJZW07] Jianbing Shen, Xiaogang Jin, Chuan Zhou, and Charlie C. L. Wang. Gradient based image completion by solving the poisson equation. *Computers & Graphics*, 31(1):119–126, 2007.

[SLT10] Bolan Su, Shijian Lu, and Chew Lim Tan. Binarization of historical document images using the local maximum and minimum. In *Document Analysis Systems*, pages 159–166, 2010.

[Tel04] Alexandru Telea. An image inpainting technique based on the fast marching method. *journal of graphics, gpu, and game tools*, 9(1):23–34, 2004.

[TGM10] Anna Tonazzini, Ivan Gerace, and Francesca Martinelli. Multichannel blind separation and deconvolution of images for document analysis. *IEEE Transactions on Image Processing*, 19(4):912–925, 2010.

[TLD06] Zinovi Tauber, Ze-Nian Li, and Mark S. Drew. Review and preview: Disocclusion by inpainting for image-based rendering. Technical Report TR 2006-04, School of Computing Science, Simon Fraser University, Burnaby, BC, Canada, January 2006.

[Tsc06] David Tschumperlé. Fast anisotropic smoothing of multi-valued images using curvature-preserving PDE's. *International Journal of Computer Vision*, 68(1):65–82, 2006.

[Web97] Allan G. Weber. The USC-SIPI Image Database. Technical report, University of Southern California, Signal and Image Processing Institute, Department of Electrical Engineering, Los Angeles, October 1997.

[WL00] L.Y. Wei and M. Levoy. Fast texture synthesis using tree-structured vector quantization. In *SIGGraph-00*, pages 479–488, 2000.

[Woh09] Brendt Wohlberg. Inpainting with sparse linear combinations of exemplars. In *ICASSP*, pages 689–692, 2009.

[WR09] Jiying Wu and Qiuqi Ruan. A novel exemplar-based image completion model. *J. Inf. Sci. Eng.*, 25(2):481–497, 2009.

[WRTF06] Oliver J. Woodford, Ian D. Reid, Philip H. S. Torr, and Andrew W. Fitzgibbon. Fields of experts for image-based rendering. In *BMVC*, pages 1109–1118, 2006.

[WSI07] Yonatan Wexler, Eli Shechtman, and Michal Irani. Space-time completion of video. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(3):463–476, 2007.

[YOT05] M. Yasuda, J. Ohkubo, and K. Tanaka. Digital image inpainting based on markov random field. In *CIMCA/IAWTIC*, pages 747–752, 2005.

[YSM10] Guoshen Yu, Guillermo Sapiro, and Stéphane Mallat. Image modeling and enhancement via structured sparse model selection. In *ICIP*, pages 1641–1644, 2010.

[ZRK10] Jun Zhou and Antonio Robles-Kelly. Image inpainting based on local optimisation. In *ICPR*, pages 4440–4443, 2010.