UNIVERSITÄT
KOBLENZ · LANDAU
Fachbereich 4: Informatik

AGAS

Object Removal from Still Images Employing Inpainting Techniques

Diplomarbeit
zur Erlangung des Grades
DIPLOM-INFORMATIKER
im Studiengang Computervisualistik

vorgelegt von

Sebastian Vetter

Betreuer: Dipl.-Inform. Peter Decker, Institut für Computervisualistik, Fachbereich Informatik, Universität Koblenz-Landau
Erstgutachter: Dipl.-Inform. Peter Decker, Institut für Computervisualistik, Fachbereich Informatik, Universität Koblenz-Landau
Zweitgutachter: Prof. Dr.-Ing. Dietrich Paulus, Institut für Computervisualistik, Fachbereich Informatik, Universität Koblenz-Landau

Koblenz, im Mai 2009

Acknowledgement

The last six months have been full of joy, sweat, success and disappointment which all contributed to this work and eventually lead to its completion. Everything that follows on the next pages, however, would not have been possible without the help, support and understanding of all the people around me and thinking of me from a distance. I am thankful for all the thoughts and encouragements that helped me standing it out until the end.

A few deserve a special thanks due to their active contribution. Mike, his wife Sarah and Ali, as well as the rest of the “Stilbruch” & “Mephisto” staff, for making their venues my home/office/lounge with royal food and beverage supply at any day and night-time, never short of a *chilled brew* be it regular or “vegetarian”.

The members of the working group *Active Vision* for their advice on subject-specific matters and their help on clearing technical obstacles.

Furthermore, my fellow sufferer Christian Latsch who shared the same lab, saved me from actually going crazy, at times, and offered his expertise on the graphical design and layout.

In addition, I would like to set the proofreading squad apart. I would like to thank Marco Thum for his insightful advise on the structure and flow of this work and for the numerous hours he spent on the subject. I also thank Martin Birmingham and Justin Finkelstein for sharing their language skills and helping me with pimping my eloquence.

The greatest involvement, however, is attributed to Judith Haas who work on her own thesis [Haa09] at the same time. She helped me tremendously with her moral support spiced with the occasional motivational speech. Her uplifting feedback on my work counterbalanced her helpful though discomfiting nitpicking. I sincerely apologise for the hard times and sinusoidal moods and thank you dearly for being the technical adviser you were and, more importantly, the irreplaceable friend you are!

And finally, I would like to thank my parents, Bernd and Betina Vetter, for their unconditional love and support that I enjoyed over the years. Your encouragement and endorsement has been vital for me to be where I am now. I am grateful for all the wisdom you shared and advise you gave me. Having you as my parents makes me a rich person as I am and I love you very much!

Kurzfassung

In diese Arbeit werden drei Verfahren zur Objektentfernung aus Bildern einander gegenübergestellt. Zwei der ausgewählten Verfahren stammen aus dem Bereich der sogenannten Inpainting-Verfahren, während das Dritte aus dem Forschungsgebiet der medizinischen Bildverarbeitung entnommen ist. Die Evaluation dieser Verfahren zeigt ihre jeweiligen Vor- und Nachteile auf und prüft ihre Anwendbarkeit auf das spezifische Problem, ein Farbkalibiermuster aus strukturdominierten Bildern zu entfernen. Auf der Grundlage dieser Eigenschaften werden schlussendlich mehrere Erweiterungen vorgestellt, die eine verbesserte Anwendbarkeit auf das gestellte Problem erreichen.

Abstract

This work evaluates automated techniques to remove objects from an image and proposed several modifications for the specific application of removing a colour checker from structure dominated images. The selection of approaches covers the main research field of image inpainting as well as an approach used in medical image procesing. Their results are investigated to disclose their applicability to removing objects from struture-intense images. The discovered advantages and disadvantages are then used to propose several modifications for an adapted inpainting approach suitable for removing the colour checker.

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Die Vereinbarung der Arbeitsgruppe für Studien- und Abschlussarbeiten habe ich gelesen und anerkannt, insbesondere die Regelung des Nutzungsrechts.

Mit der Einstellung dieser Arbeit in die Bibliothek bin ich einverstanden.	ja ☒	nein □
Der Veröffentlichung dieser Arbeit im Internet stimme ich zu.	ja ☒	nein □

Koblenz, den 14. Mai 2009

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In the last few years, images have become more and more important in our society and its every day communication. Pictures are taken anywhere, at any time and for an infinite number of reasons. It is not surprising that these images offer an equally growing number of new applications and areas of processing, improved and modified. One of many applications is the recovery of unknown pixel intensities. This can be applied to the removal of objects from an image where the area occupied by the object is first erased and then reconstructed using an available source to sample from. This is a task that occurs frequently in every day life when an image is taken inattentive and shows objects that were not desired to be in the picture, e. g. a couple making out in the background on your nephew's birthday party. Which is why many image applications already implement tools that allow the removal of those objects.

The focus of this work is a similar application where the removal of a specific object is desired. The object is a colour checker that is placed somewhere in an image taken on campus at the University of Koblenz as shown in Figure 1.1. This scenario is borrowed from a deprecated project named *Enhanced Reality* [3] that aimed at enriching the real world with artificial content. A person is equipped with a see-through display hidden in a pair of glasses that allows her to see the real world as well as displayed content. One of the big challenges in this sort of application is to re-create the exact same lighting conditions and corresponding colour impressions for artificial objects. This is required to integrate them well into the real world. One possible solution to extract reproducible colour information from an image is to place a colour checker inside the image. It contains a set of normed colours arranged in a checker pattern. Knowing the correct colour values of each square on the colour checker makes it possible to normalise the colours in the image and adapt the colouring of an object according to the lighting conditions in the image.

The unfortunate side effect of this setup however is that the colour checker is not only visible to the application but also by the person that uses the see-through display. It is, therefore, evaluated if it is possible to achieve an automated solution to remove the

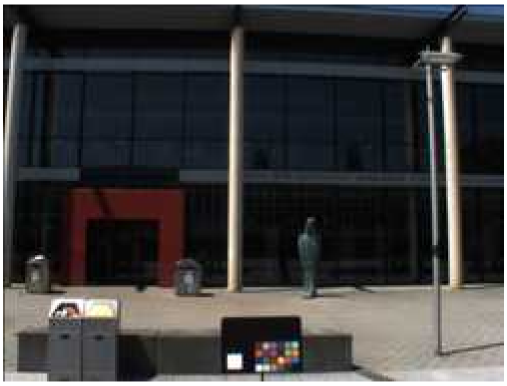


Figure 1.1: Colour checker in the campus image depicting the assumed scenario of the *Enhanced Reality* project [3].

colour checker, assuming that its location inside the image is known or provided by a previous object recognition algorithm.

The removal of objects from images is not a development or discovery of the modern age of digital imaging. It goes far back into the early days of analogue photography where image retouching was used to alter images. Stalin, for instance, is well-known for letting professional retouching artists remove political opponents from important images [Kin99]. The used techniques required highly experienced professionals to obliterate all indicators that could expose its manipulation. Image retouching became much easier when digital images started to take over and image processing applications evolved quickly. Professional tools such as Adobe® Photoshop® [5] and free open-source derivatives such as *The Gimp!* [7] provide the instruments required for digitally retouching images. Complemented with application specific [Bra98, Fit08] or more general handbooks [Bus03] this provides the foundation for manually removing objects from images. Nonetheless, professional skills are still required to obtain a high quality of the altered image.

Automating this process is as challenging as it is helpful for inexperienced users or automated procedures where no user-interaction is desired. The latter is the solution required for an automated removal of the colour checker. A whole field of research has developed over the last years trying to find a solution for this problem. An approach that first introduced the term *inpainting* was presented at the SIGGRAPH [6] conference in 2000 by its authors Bertalmio and Sapiro [BSCB00]. They tried to actually replicate the manual process used by retouching artists and other related professions. They consolidated conservators at the Minneapolis Institute of Arts and based their method on their manual workflow. Its imitation lead to coining the term "image inpainting" for this type of research. A variety of approaches succeeded providing similar or extending techniques which range from texture-based inpainting to the exploitation of spectral properties. This provides a wide-ranging repertoire of potential solutions to choose from.

The objective of this work is to investigate an extract of these approaches and identify the most promising works. In addition, an unusual candidate is included from an entirely different field of research focussing on a related topic. The reconstruction of defect pixels in medical images has been extensively researched and intriguing solutions have been provided for medical applications. The intriguing question of its applicability to real images lead to a more comprehensive evaluation and comparison of promising approaches in both areas to determine their qualitative eligibility for the removal of the colour checker. Their advantages and disadvantages are interrogated to generate a comprehensive understanding for their requirements and preconditions. The associated discovery of existing weaknesses concerning the high complexity of images with architectural content eventually motivated the proposal of several modifications to adjust the inpainting quality.

The Outline

An overview on the research in image inpainting and related topics is provided in Chapter 2 where the approaches mentioned above are put into the context of other works. This is followed by a comprehensive description of these approaches in Chapter 3. Their respective quality is then evaluated using a variety of test images. The discovered weaknesses of either approach are summarized in Chapter 4 and starting points for own modifications are defined. The actual modifications are then described and compared to the results of the major approaches in Chapter 5. A summary of the achieved results and a few notions for future modifications conclude this work in Chapter 6.

Chapter 2

State of the Art

The task of removing an object from a still image has been the subject of many research projects that collected a broad knowledge of the characteristics of this problem and its possible solutions. The very first approaches to remove damages from images simply had to deal with missing data with the size of a few pixels. Introduced by image processing operations such as resizing, skewing and rotating some pixel intensities on the grid where lost and could only be filled by *interpolation*. Consequently, many different methods were developed such as nearest neighbour, bilinear and bicubic interpolation [GW06]. Almost any image processing application today incorporates them.

Another field of research that might seem helpful at first but cannot be applied is noise removal. It deals with information that is lost through the insertion of noise such as *Salt and Pepper* or *Gaussian* noise. The intensity of a pixel, however, is not completely lost. It is contained in a mixture of colour or grey level intensity and noise. Therefore, the noise has to be mathematically modelled before it can be removed from the image. *Inpainting*, however, deals with the reconstruction of unknown pixel intensities containing *no information at all*. This requires the extraction of intensity information from known intensities in undamaged regions of an image.

Another technique [KMFR95a, KMFR95b], developed to restore film material, extracts this missing information from neighbouring frames in the sequence. They often contain the additional data required to reconstruct missing intensities. Nowadays, more sophisticated techniques [PSB05, PSB07] allow the removal of large objects from a sequence extrapolating information from its temporal neighbourhood. This requirement remains unmet in a single frame environment. A different route has to be taken to find the most suitable intensities in the known region and use them to fill the missing regions.

In summary, the major difficulty in image inpainting is that the amount of reliable information in an image is restricted by its size and the amount of missing data. These difficulties are addressed in a wide range of approaches concerning inpainting research. This chapter provides an insight into this field of research to be able to justify the selection of approaches that are investigated more comprehensively.

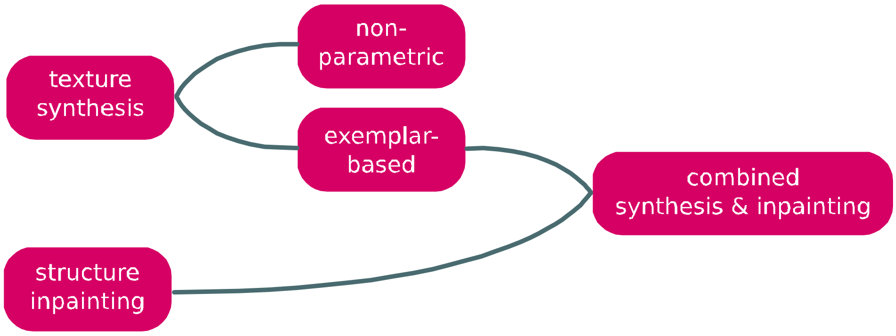


Figure 2.1: Recent techniques in image inpainting try to combine texture synthesis and structure inpainting. They are both research fields that originally attempted to solve different problems. Texture synthesis can still be partitioned into non-parametric and exemplar-based synthesis. Combined approaches eventually used ideas from both exemplar-based synthesis and structure inpainting.

2.1 Texture Synthesis and Structure Inpainting

Automated algorithms that allow inpainting of large unknown regions have slowly emerged from two fields of research, namely texture synthesis and structure inpainting. Research concerned with texture synthesis expects a small texture sample and tries to automatically create a much larger texture image. In contrast, structure inpainting initially focusses on the removal of small damages by reproducing intensities based on neighbouring pixels. The connection between texture synthesis and structure inpainting is illustrated in Figure 2.1 which shows that *exemplar-based* synthesis and structure inpainting eventually lead to combined approaches in image inpainting. This chapter provides the required insight into preceding and related research.

2.1.1 Texture Synthesis

It has been explained that texture synthesis describes the practise of creating new synthesised textures from a small texture sample. Typically, textures are classified as either stochastic or regular, although most textures in real images are located somewhere in between. The application of synthesising near-regular textures [LLH04] in real images is a challenge pointing in the same direction as image inpainting. The approaches that emerged to provide sufficient solutions are generally divided into *parametric* and *non-parametric* or exemplar-based techniques. The latter have contributed a lot to the research in image inpainting.

A very early attempt to tackle the problem of purging an item based on exemplar-based synthesis has been developed by Harrison [Har01]. It is closely based on the acquired findings by Graber [GS81] and Efros et al. [EL99]. They proposed algorithms to create a new intensity value by sampling it from its best-matching neighbouring pixels. Although [EL99] is a very important approach in texture synthesis, the size of a neighbourhood of sample pixels is restricted to the size of the largest textural pattern. Therefore, it is extended in [WL00] to combine a texture- and an exemplar-based technique working through different levels of coarseness in a pyramid scheme.

The procedure Harrison describes compares neighbouring pixels in two images, the sample texture and the input image. An output image contains the synthesised result where the unknown region is filled with data from the sample texture. He is the first to introduce the concept of a priority value to define the fill-order of this process. The similarity in the filling process is calculated using a normalised weight of a pixel and its neighbourhood. It relates the amount of information carried by a pixel in respect to its neighbours. As opposed to many others, the distance measure for patches is calculated using the *Manhattan* instead of the *Euclidean* distance because "it is more forgiving of outliers" [Har01].

0ne outcome of his research was a plugin for the image processing application The Gimp! [7] called "Resynthesizer" [4] that provides the described functionality in a ready-to-use environment. The approach adds another important piece to the puzzle that was assembled in [CPT04], where the priority value is adapted and improved.

2.1.2 Structure Inpainting

In contrast to texture synthesis, structure inpainting looks at the problem of filling in missing data from an artistic point of view. As part of their work, Bertalmio et al. [BSCB00] consolidated inpainting artists at the Minneapolis Institute of Arts to learn how they proceed when restoring a painting or drawing. They described in [BSCB00] that there is no such thing as the "right" way for inpainting. It is simply an attempt to conserve the unity of the painting by adapting the style as closely as possible. Considering these observations, they extracted four steps a conservator would take to restore the integrity of a masterpiece:

1. The unity of the picture determines its conservation.
2. Structures surrounding the target are continued into it.
3. Newly created regions are filled with appropriate colours.
4. Small details are recovered to add texture to the image regions.

With these guidelines in mind, they tried to replicate the methodical procedure in manual inpainting. Their idea was to propagate the known intensity information into the region of unknown intensities. The propagation direction of a pixel location $\boldsymbol{p}$ is given by its corresponding vector in the field $N(\boldsymbol{p}) = \nabla^{\perp}I(\boldsymbol{p})$. The direction represents the smallest spatial change of intensities. Filling the unknown region step by step along the isophote is completed by an image diffusion step after every few iterations. They use the anisotropic diffusion equation from [PM90] to ensure the propagation directions are correct. An additional problem is inherent in handling colour planes individually. They inhibit associated artefacts using their own perception oriented colour space similar to *CIELUV* [KA08]. This algorithm forms the foundation for research in image inpainting and is well-cited by many research groups who have adapted and improved their ideas. In particular, the idea of isophotes are a central aspect of more recent approaches such as [CPT04] and [IJ07].

2.1.3 Combined Inpainting and Synthesis

The work on the previously described method of structure inpainting [BSCB00] resulted in the discovery that "different techniques work better for different parts" [BVSO03]. Hence, they started to develop a new approach [BVSO03] based on the decomposition of an image into two parts. One part represents the structure, the other part, the texture of the damaged image. They use the decomposition techniques proposed by Vese and Osher [VO03], a total variation minimisation [ROF92] to extract the structure component and the space of oscillating functions [Mey01] for the texture component. Thereafter, structure inpainting and texture synthesis are applied to the separated components respectively. The image inpainting process is based on their own approach, whereas texture synthesis is borrowed from [EL99].

Drori et al. [DCOY03] headed into a similar direction trying to combine texture synthesising and structure inpainting. Their work, however, is based on the idea of capturing both global and local structures of various sizes using fragments. These circular sets of pixel locations, as opposed to the usual rectangular ones, adapt to different structures in an image by varying their sizes. The main idea of their approach is to operate on different scales, from coarse to fine. On each level, they approximate the missing region from its coarser predecessor. This is followed by computing a *confidence map* of the intensities and a *level set* that defines the filling order of pixels. Based on these values, the pixels in the unknown region are iteratively filled depending on an adaptive neighbourhood and their most suitable sample consisting of known intensities. The results of this approach stand out considering the size of the removed object relative to the size of the image. Its major drawback, however, is the high computation times that lay "between 83 and 158 minutes for 384 by 256 images" [DCOY03].

Similar to the fragments used in [DCOY03], Criminisi et al. [CPT04] proposed an approach employing rectangular patches that are iteratively filled. Their main concept is the calculation of priorities for each patch. Along the boundary between the regions of known and unknown pixel intensities, each patch is assigned a priority value depending on the reliability of its contained pixel intensities, the confidence of a pixel, and its structural information denoted by its gradient. The resulting values determine the order in which these patches are filled. In each iteration, the patch with the highest priority is selected, the best-matching sample is retrieved and corresponding pixel intensities are copied from the sample to the search patch. It is, therefore, a very simply and effective approach that became well-cited in image inpainting followed by several extension and adaptations.

2.2 Inpainting in the Wavelet Domain

Over the last years, wavelets have made their way into a lot of different fields in image processing. Image compression is the most established one. In 2000, the JPEG Committee, former creator of the JPEG standard, introduced a new format called JPEG 2000 [2] that was meant to supersede the old standard. One of the improvements was that they used a compression method based on wavelets instead of the discrete cosine transform. This drew a wider attention to the potential of wavelets for applications in image processing. Works in denoising and reconstructing damaged images emerged. Various researchers started addressing the problem of image data reconstruction exploiting the properties of wavelet coefficients. The total variation minimisation proposed in [CSZ06], uses a wavelet framework, which was a novel approach at the time. In contrast, Rane et al. related to the block-shaped origin of JPEG images and developed a patch-based approach [RRS02] implementing the wavelet transform. They use the *detail coefficients* in the wavelet domain to classify each missing block. The classification relies on the high-frequency response encoded in the horizontal, vertical and diagonal subband coefficients. They allow to identify so called *edgy* and *non-edgy* blocks which then can be handled differently according to their characteristics. These blocks are reconstructed based on the high-frequency data encoded in the coefficients of neighbouring blocks. Similar neighbours are used to interpolate the missing data in the block where similarity is measured by minimising the sum of squared errors.

A very recent approach [OH07], in contrast to most other wavelet-based approaches, attempts to fulfil the task of reconstruction using the continuous wavelet transform (CWT). Its authors exploit the properties of magnitude and phase exhibiting strong geometrical regularities around edges and within texture areas. These are then used separately in an iterative process to reconstruct missing data with respective properties. A limitation is, however, that each iterative step requires a forward and inverse transform. Although they describe it as an image inpainting technique they simply focus on small blocks of missing data with block size 16×16, similar to [RRS02]. In summary, both techniques do not qualify for actual inpainting due to their focus on small blocks.

Much of this work, however, has been crucial to develop approaches that are directed towards inpainting and texture synthesis. A proposal [PS03] to recover missing pixel intensities uses a strategy similar to [HT96], employing bandwidth limitations on frequencies. The major difference is the use of a different frequency space. [PS03] operates in the spatio-frequency domain of wavelets, as opposed to the Fourier space in [HT96]. However, both share the use of a technique known as *projection onto convex sets* (POCS). Considering [PS03], the image is first transformed using a specific wavelet function. Thereafter, each coefficient corresponding to a damaged or unknown pixel intensity is restricted by the minimum and maximum coefficient of its neighbours. Inverting the transform, a new estimate is generated and used to update the inpainting region of the input image. This process is repeated iteratively until the desired quality of the inpainted image is reached. This technique illustrates the potential of wavelet-based inpainting. Artefacts in the results demonstrate, however, that further improvements are needed to suffice the needs of image inpainting.

An additional very recent approach was proposed by Ignácio and Jung [IJ07] deriving a wavelet-based inpainting method from [CPT04]. They transform the image and the provided binary mask into the wavelet domain and apply an inpainting technique similar to the one proposed in [CPT04]. A similar fill-order guided by priorities is used, where the gradient information is extracted from high-frequency wavelet coefficients and an additional structure component is introduced. The latter also exploits structure-related information incorporated in the detail coefficients. The priority is completed by the confidence defined in [CPT04] to balance the structure components by representing texture reliability.

2.3 Supplementary Approaches

The border between two distinguishable objects or textures in real images is the most important element to reconstruct, yet the most difficult to reproduce. Examining these edges leads to multiple approaches focussing on the completion of edges before filling the remaining unknown regions. Sun and his colleagues proposed a graph-based approach [SYJS05] that separates the propagation of texture and structure. The general idea is similar to [BVSO03], where structure inpainting and texture synthesis are addressed individually. However, the solution in [SYJS05] differs in two aspects. First, the approach requires initial user interaction to define the structures that should be continued into the unknown region. Second, user-defined lines are generated into a graph that determines the fill-order of its patches. After this step is completed, the remaining regions are filled using a combination of texture propagation and photometric correction.

An inpainting method titled *sketch-guided inpainting* [CLLA06] implements the concept of [SYJS05] and aims at eliminating the user-based component. The attempt is based on the mathematical theory of *primal sketch models* proposed in [eGZW03]. Sketches in the target region are reconstructed with this model and pixels along these sketches are filled first, followed by the remaining regions. Similar results can be obtained using a tensor voting algorithm [JT03] to reconnect structures that are likely to run through the unknown region. The expenses of the required segmentation steps is the downside of the algorithm. In fact it is another restriction that can be avoided in [CPT04].

A solution that makes extensive use of the internet service *Flickr* [9] is described in [HE08]. The algorithm compares an image with an area marked for removal to a large quantity of images from the database and tries to find the most suitable image to sample the data from. They use a *gist descriptor* from [RFMT03, OT06] to recognise image similarities and find the best matching image in a large set of samples.

With such a sample subset, they have restricted the type of images to semantically similar ones that are then matched with the inpainting image based on contextual properties. The most similar region is then extracted from the selected image and inserted using *Poisson blending* [PGB03]. In contrast to most approaches they loosen the restriction of never altering known pixel intensities. They actually allow overwriting known intensities similar to [WBTC05] but employ a confidence value discouraging it the further away a pixel is located from known and already filled pixel locations.

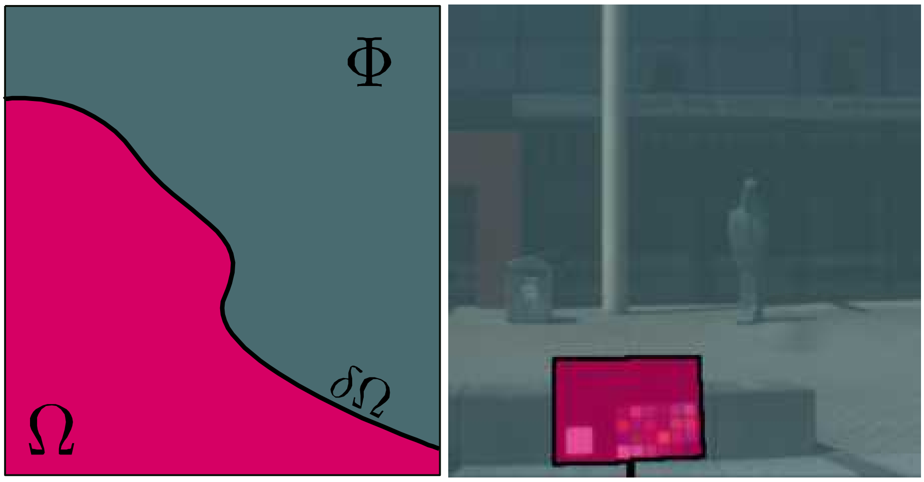
Chapter 3

Description of Inpainting Techniques

The search for possible solutions to the problem of removing the colour checker from an image was to evaluate two different approaches from unrelated fields of research. Since medical image processing is the subject of several lectures and courses at the University of Koblenz, it was an intriguing idea to test the applicability of a technique developed to recover unknown pixel intensities in X-ray images. The approach proposed by Aach et al. [AM01] exploits the periodicity of spectral lines in the Fourier space and iteratively estimates a corrected frequency spectrum. Fourier analysis as an established instrument in image processing and the usually exceptional quality of algorithms used in image processing made it the first choice.

The existing and ongoing research in the field of image inpainting suggested the consideration of an additional approach that has been developed to recover unknown pixel intensities from real images. The outline of approaches concerning this topic, described in Chapter 2, offered a wide range of techniques to choose from for a closer examination. One approach, however, stood out. It was proposed by Criminisi et al. [CPT04] as a combination of texture synthesis and structure inpainting. Researching the inpainting literature revealed that it is a well-cited approach and many alternative proposals test and validate their results against it. Furthermore, the concept of using a patch-based algorithm that tries to replicate the manual workflow, as it was first proposed in [BSCB00], appeared sensible and also straight-forward to implement. Additionally, the use of the image gradient in determining the order of propagating patches suggested a good response to dominantly structured images and presented promising results in the proposal itself. These initial assumptions where substantiated by numerous extension, such as [CHL+05], providing an improved algorithm and further insights into the properties and limitations of the approach.

The extended research also revealed another approach that employs the same concept as [CPT04], using a patch-based inpainting technique which is guided by priorities determining the order of inpainting patches in the unknown region. The novelty, however, was to transform the image into the wavelet domain and exploit its spectral prop-



(a) (b)

Figure 3.1: The partitions of an image as used in inpainting: source region (pink), target region (green) and boundary (blue). A schematic representation is shown in (a) and applied to a real image in (b).

erties. Its conceptual similarity to both [CPT04] and [AM01] made it a suitable third contestant in a comparative evaluation. Furthermore, the representation of edge-related information in the detail coefficients of the wavelet transform suggested an even better applicability to structure-rich images then [CPT04].

3.1 Defining Terms and Symbols

At first, a look at the terminology of this topic is required. Image inpainting has been the subject of research for several years and has therefore an established set of terms that are used throughout literature and research documents. Some of them might be used differently in various contexts, which makes it even more important to clarify the meaning that will be used in this work. In general, it is compliant with the definitions used in [CPT04] and [IJ07]. This is, except for the excursion into medical image processing, where a different terminology applies.

Inpainting is generously used in literature covering any topic that is related to the removal of objects from and filling in of missing regions in an image. It was originally coined in [BSCB00] where the authors borrowed it from the profession of conservators to *inpaint* restorable art work, thus, leading to the title of *structure inpainting* for their type of approach explained in Chapter 2.1. The inconsistent use throughout the literature can cause confusion where structure inpainting is combined with other techniques. To avoid this, the term structure inpainting is only be used, in this work, when the specific practice of structure inpainting is addressed. Image inpainting, or simply inpainting, applies to the general idea of filling unknown image data and is used accordingly.

The image is the most important element of image inpainting. A two-dimensional image I is defined as a mapping $I : Loc \rightarrow Val$ that maps coordinates (x, y) from Loc to intensity values in Val. Considering only colour images in this work, Loc is given by $Loc = [0, W-1] \times [0, H-1]$, where W and H denote its width and height respectively. A colour intensity is represented by a vector in $Val = [0, 2^n[^3$. Therefore, a *pixel* is defined as a tuple $P \in Loc \times Val$. Its two components can be described separately as $loc(P)$ to denote its position and $val(P)$ to denote its intensity value in position space and its coefficient in the wavelet domain. The latter are addressed as *intensity* and *coefficient* respectively, whereas the location is represented by a position vector $\boldsymbol{p} = (x, y)^{\mathrm{T}}$.

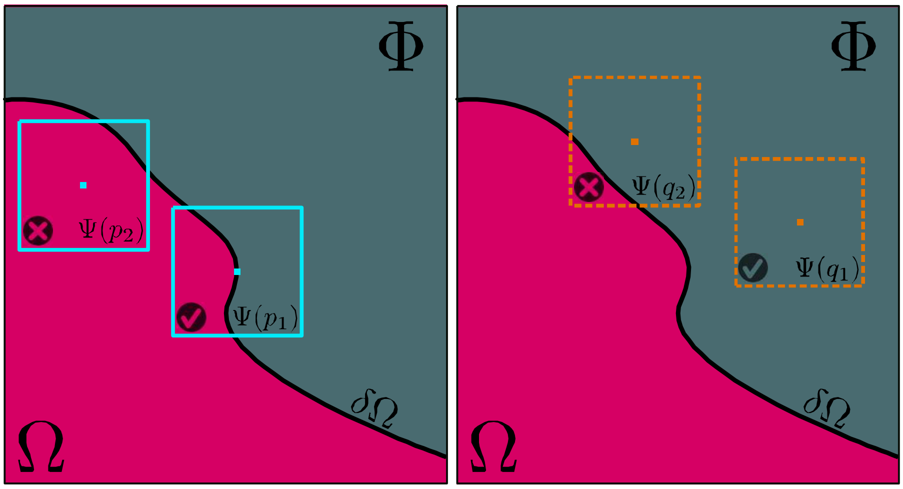
A neighbour pixel is a pixel P that is adjacent to P' in I. Two pixels are considered neighbours if the difference in location holds $||loc(P)-loc(P')|| \leq \sqrt{2}$ with $loc(P), loc(P') \in Loc$. This defines that each pixel P is surrounded by 8 neighbouring pixels, i.e. an 8-neighbourhood. The only exception applies to border pixel locations where the neighbourhood amounts to number of pixels not exceeding the border.

A region R in image processing usually denotes a *connected set* of pixel locations in an image I. Such a set of pixels is called connected when any two pixels P and P' in R are connected by a path X that lays always in R. A region can therefore be defined as $R \subseteq I$ where $\forall P, P' \in R : \exists X \subseteq R$ with $P = P_0$ and $P' = P_n$. In image inpainting, an image I is partitioned into two disjointed sets of pixel locations named *target* and *source region*. Therefore, the definition of a region to be a connected set is softened to allow disconnected pixels as well. For these two regions, it holds that $\Omega \cup \Phi = I$.

The ***target region*** Ω can be described as the area of the image that is covered by the object to be removed and can be specified in several different ways. The target region is assumed to be provided as a user-defined binary mask but could also be given by segmentation or object recognition routines. A pixel location in the binary mask is assigned to the *target region* if $val(P) = 0$. All pixel locations in the target region are considered unknown and will be filled during inpainting.

The ***source region*** Φ provides the data used to fill the target region. It includes all pixels that are not part of the target region, which translates to $\Phi = I - \Omega$. A pixel location in the source region is considered a known pixel. In general, it remains unchanged throughout the inpainting process. It will be explicitly stated otherwise.

The boundary $\delta\Omega$ denotes the border between the source and target region. A pixel location P is considered a boundary pixel if its location $loc(P)$ is in the source region



(a) Search patch restrictions. (b) Sample patch restrictions.

Figure 3.2: A search patch is always centred at a boundary pixel as shown in (a), whereas a sample patch has to lay completely in the source region, as shown in (b).

and at least one of its neighbouring pixels P' is part of the target region. This can be formalised as:

$$\delta\Omega = \left\{ loc(P) \in \Phi \mid \exists\, P' \in \Omega : \|loc(P) - loc(P')\| \leq \sqrt{2} \right\}$$

Therefore, the boundary is a subset of the source region, formally $\delta\Omega \subset \Phi$. This set of pixel locations is of special interest in multiple inpainting approaches, most importantly in [CPT04] and [IJ07].

A *patch* is defined as a set of pixel locations $\Psi(\boldsymbol{p})$ centred at location $\boldsymbol{p} \in Loc$. It has a square size $M \times M$ where M denotes the width and height in number of pixels. The only exception for a patch to have a non-square shape is to prevent the violation of the image border where clipping is applied. There are two types of patches with a specific meaning in [CPT04, IJ07]. In the following, they are defined as *search* patch denoted $\Psi(\boldsymbol{p})$ and *sample* patch denoted $\Psi(\boldsymbol{q})$. The search patch defines a patch centred on the boundary that contains unknown pixel intensities for $val(P)$ as Figure 3.2(a) shows. Figure 3.2(b) shows a sample patch that contains *only* known pixels and functions as a template for the missing values in $\Psi(\boldsymbol{p})$. In addition, the search patch with the highest priority at a given iteration is defined as $\Psi(\hat{\boldsymbol{p}})$ with its best-matching sample patch denoted $\Psi(\hat{\boldsymbol{q}})$.

Corresponding pixel locations in a search patch $\Psi(\boldsymbol{p})$ and a sample patch $\Psi(\boldsymbol{q})$ are pixels R and S which hold $loc(R) \in \Psi(\boldsymbol{p})$ and $loc(S) \in \Psi(\boldsymbol{q})$ respectively. Their correspondence is determined by their relative location to their respective patch centres $\boldsymbol{p}$

and $\boldsymbol{q}$. That is, the pair of corresponding pixels hold $loc(R) - loc(P) = loc(S) - loc(Q)$. Their locations are denoted as position vectors $\boldsymbol{r}$ and $\boldsymbol{s}$, as defined above. These correspondences imply that $\Psi(\boldsymbol{p})$ and $\Psi(\boldsymbol{q})$ have the same size $M \times M$.

The ***search window*** is defined as a set of pixel locations $\Theta(\hat{\boldsymbol{p}})$ centred at location $\hat{\boldsymbol{p}} \in Loc$ with $\hat{\boldsymbol{p}} \in \Psi(\hat{\boldsymbol{p}})$. It contains all pixel locations $\boldsymbol{q}$ that correspond to possible sample patches $\Psi(\boldsymbol{q})$ to select the best-matching patch $\Psi(\hat{\boldsymbol{q}})$ from. The size of $\Theta(\hat{\boldsymbol{p}})$ is defined as $\theta = aM \times aM$ with $a \in \mathbb{N}$.

The ***fill-order*** denotes, in a completed inpainting process, the order in which search patches have been filled. In each iteration, the search patch $\Psi(\hat{\boldsymbol{p}})$ that is to be filled is determined by the highest priority value for each corresponding pixel location $\hat{\boldsymbol{p}} \in \delta\Omega$.

3.2 Interpolation in the Fourier Domain

The advent of digital technology into medicine started a new field of research, medical image processing. A wide range of applications emerged from developments such as image intensifiers, and later, flat panel detectors [BKK⁺00]. Their application in radiography, however, requires extended image processing algorithms to deal with their technical shortcomings. Several of these works are focussing on the reconstruction of missing data in images acquired by faulty flat panel detectors, a problem which is caused during the manufacturing process of the panels. Although a new panel would be expected to be immaculate, the reality is different. Companies define that a small amount of defects are acceptable and sell them without further ado. Looking at medical imaging, it had disastrous consequences for patients if such a faulty panel would be used, as is, in an X-ray imaging device. The medical software, therefore, had to provide solutions and reconstruct the values of defect pixels. One of the major ideas for a solution was to extrapolate frequencies instead of intensities, exploiting their periodic properties in frequency space.

The preconditions for defect pixel interpolation in medicine correspond to those in image inpainting, as formally described in Chapter 3.1. The formal notation differs, however. The observed image is g with a defect window w and the desired undistorted image, or estimate, is formalised as f. w is equivalent to the binary mask that defines the target region in Chapter 3.1 partitioning the captured image into *active* and *inactive* pixels. The set of inactive pixels resembles Ω, whereas Φ corresponds to all active pixels. To retrieve w, an initial calibration of the panel is required which identifies the condition of each element. In summary, it can be rephrased as recovering an image f by inpainting the target region in g, defined in the binary mask w.

In an early attempt, an approach proposed by Papoulis introduced the concept of exploiting the periodicity of frequencies in Fourier space [Pap75]. It is assumed that

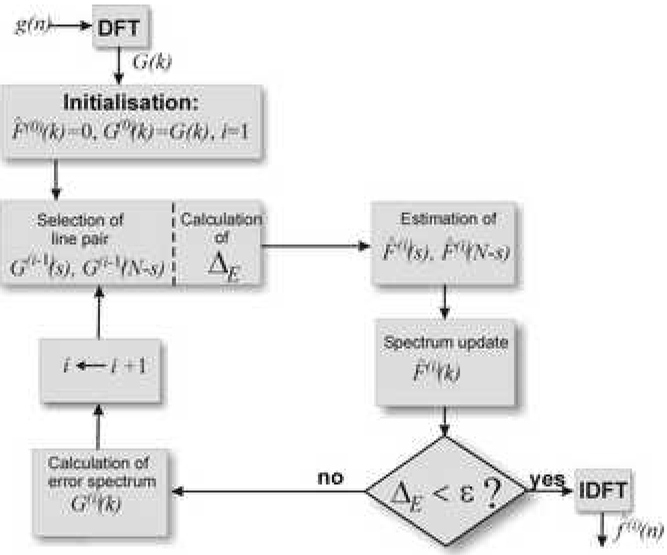


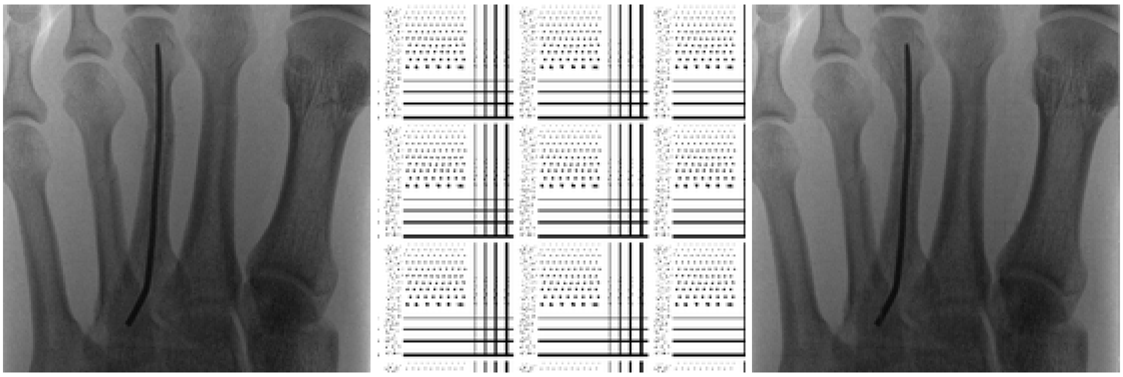
Figure 3.3: Illustration of the deconvolution algorithm by Aach et al. [AM01]. It shows that only a single transform is required in each direction while the iterative process operates completely in the spectral domain. This is the original illustration used in [AM01].

a missing intensity in the spatial domain yields frequencies that exceed a known bandwidth. Missing values can, therefore, be recovered by recursively enforcing a bandwidth limitation. This is applied to an image by decomposing it into its frequencies using Fourier analysis. Their bandwidth is then limited to the known frequency band before the inverse Fourier transform returns a new estimate $\hat{f}$. Afterwards, g is updated using intensities in $\hat{f}$ for elements that are considered inactive in w. Recursively repeating these steps converges the optimal undistorted image.

The downside of [Pap75] is that a large amount of spectral decompositions are required, which highly increases computational expenses. Aach et al. [AM01] developed an approach to succumb this limitation. A single transform into frequency space with its inverse correspondence are sufficient since the extrapolation is carried out entirely in the Fourier domain. In general, the undistorted image is recovered by deconvolving the defect window and the acquired image based on the concept of selective deconvolution [SIA90]. This concept exploits the convolution theorem, a description can be found in [PH03], modelling the observed signal as a convolution of the undistorted image and the mask, which is formally defined as

$$g(n) = f(n) \cdot w(n) \circ\!\!-\!\!\bullet\; G(k) = \frac{1}{N} F(k) * W(k) \quad 0 \leq n, k \leq N \tag{3.1}$$

where a univariate signal is assumed for simplicity.



(a) (b) (c)

Figure 3.4: The reconstructed image in (c) shows an X-ray of a hand after applying [AM01]. The defect input image was generated by applying the binary mask (b) to an optimal X-ray in (c). The black pixel intensities in (b) denote inactive elements.

The deconvolution of the observed signal $G(k)$ is broken down into the iterative correction of spectral lines. In each iteration, the most dominant pair of spectral lines are selected and a corrected frequency is estimated. The selection of line pairs $G(s)$ and $G(N - s)$ are justified by the symmetric property of a real-valued signal in Fourier space. A dominating pair of corresponding frequencies in a iteration i are identified by maximising the error function

$$\Delta_E = N \cdot \frac{2 \cdot |G^{i-1}(s^i)|^2 \; W(0) - 2 \cdot Re \; \{(G^{i-1}(s^i))^2 \cdot W^*(2s^i)\}}{|W(0)|^2 - |W(2s^i)|^2} \tag{3.2}$$

and, therefore, produces the highest error in $G(k)$. The error can be reduced by estimating new lines $\hat{F}(s)$ and $\hat{F}(N - S)$ from the selected pair and updating the estimate $\hat{F}(k)$. Adding the difference of the current to the previous estimate reduces the frequency of the dominant pair. Thereafter, the quality of the modified signal $\hat{F}(k)$ is measured using equation (3.2). If the minimal acceptable error threshold ε is smaller then the error Δ_E another correcting iteration starts. Otherwise, the estimate $\hat{F}(k)$ is inverse transformed and the undistorted signal $\hat{f}(n)$ is recovered. The whole process is illustrated in Figure 3.3.

As a well-established algorithm in medical image processing, the results satisfy the high standards required in the medical field as the example in Figure 3.4 shows. The fact that the results suffice medical requirements for X-ray images, qualifies it for further investigation to its adaptation to real images.

3.3 Exemplar-Based Inpainting

Chapter 2.1.1 describes the process of using texture synthesis instead of image inpainting to fill holes in an image. It also shows that there are some drawbacks to these methods that made further research necessary to reach a higher quality of results. Methods like [BVSO03] and [DCOY03] addressed this issue by trying to combine the advantages of texture synthesis and structure inpainting. Criminisi and two colleagues at "Microsoft Research" adapted this idea and created their own approach [CPT04] conjoining a patch-based technique with a defined fill-order.

Intensive evaluation of existing works in structure inpainting and texture synthesis unfolded interesting observations regarding the quality of either field. One discovery concerned promising results of synthesising texture exceeding its main purpose of generating large textures. They realised that applying it to real images provided results promising enough to serve as a basis for their approach. In addition, they observed that patch-based approaches, such as [DCOY03] and [LLX$^+$01], seemed to represent structural properties much better than operating on a single pixel. Using an exemplar-based texture synthesis approach, therefore, seemed to be the most appropriate choice.

A downside of many approaches, however, turned out to be the order in which pixels or patches were filled. [CPT04] indicates that the fill-order was usually carried out by an ad-hoc iteration over all boundary pixel locations. The result was a spiralling order omitting to take structural or textural properties into account. Hence producing disruptive artefacts such as disconnected edges, displaced patterns and structural discontinuities. To incorporate a more structure-oriented fill-order, [CPT04] looks at the concept of *isophotes* represented by their gradient vectors denoted ∇I. This structural representation is implemented into the texture synthesis by defining the fill-order of patch-based synthesis.

3.3.1 Determining the Fill-Order

The control over the fill-order is the main objective in [CPT04] for the reasons described above. The translation of the idea into an algorithmic description motivated a priority to determine the rank of a patch in the filling process. Priorities have been used before [Har01], however, the assembly of a balanced priority value is a novelty in [CPT04]. It is calculated for each pixel location $\boldsymbol{p}$ of the boundary $\delta\Omega$. The highest priority determines the search patch $\Psi(\hat{\boldsymbol{p}})$ that is processed in this iteration, filling its unknown pixel intensities.

The most accurate fill-order, according to [CPT04], can only be retrieved by balancing textural information and isophote direction. The priority $P(\boldsymbol{p})$ of a pixel located at $\boldsymbol{p}$ is, therefore, defined as

$$P(\boldsymbol{p}) = K(\boldsymbol{p}) \cdot D(\boldsymbol{p}) \tag{3.3}$$

where the *confidence term* $K(\boldsymbol{p})$ and the *data term* $D(\boldsymbol{p})$ weigh these characteristics respectively.

The Confidence Term

This term measures the reliability of pixel intensities in a patch-sized neighbourhood. The required confidence is an additional value assigned to each pixel location. It indicates the probability of a colour intensity at a location $\boldsymbol{p}$ being correct. A high confidence corresponds to a high probability with its maximum value defined as 1. At the other end of the scale, the lowest value is defined as 0 for absolute uncertainty. In the beginning, the confidence values are initialised by assigning the maximum confidence to source pixels and the minimum to target pixels. As the target region shrinks, confidences of newly filled pixels are updated to resemble the decreasing confidence when moving away from the initial source.

The confidence term for a patch is computed based on these values and can be described as the average reliability of source pixels within $\Psi(\boldsymbol{p})$. It is formally defined as

$$K(\boldsymbol{p}) = \frac{\sum_{\boldsymbol{r} \in \Psi(\boldsymbol{p}) \cap \Phi} C(\boldsymbol{r})}{|\Psi(\boldsymbol{p})|} \tag{3.4}$$

with $|\Psi(\boldsymbol{p})|$ denoting the number of pixel locations in the patch and Φ being the source region of the image at the current iteration. $C(\boldsymbol{r})$ is the confidence value at a pixel location $\boldsymbol{r}$. It, therefore, reflects only values that have already been filled. This implies that not only high confidence neighbourhoods are favoured but also that patches with more filled pixels are rewarded. In summary, a patch with more known information and a high reliability is filled before patches with less confidence or less known intensities.

The Data Term

The term formulates the counterweight to the confidence term and measures the strength and direction of an isophote relative to the boundary $\delta\Omega$. It can be interpreted as numerical representation of the angle between the isophote direction $\nabla I^{\perp}(\boldsymbol{p})$ and the corresponding boundary normal $\boldsymbol{n_p}$ originating at $\boldsymbol{p}$. This structural weight is defined as

$$D(\boldsymbol{p}) = \frac{\nabla I^{\perp}(\boldsymbol{p}) \cdot \boldsymbol{n_p}}{\alpha} \tag{3.5}$$

where α is a simple normalisation value that depends on the range of colour intensities in the image, e.g. $\alpha = 255$ in a grey level image. Its direction is perpendicular to the connecting line between the preceding and successive pixel locations of $\boldsymbol{p}$. The counterpart vector $\nabla I^{\perp}(\boldsymbol{p})$ is orthogonal to the gradient of the isophote, where the vector $\nabla I(\boldsymbol{p})$ is the maximum image gradient in $\Psi(\boldsymbol{p})$.

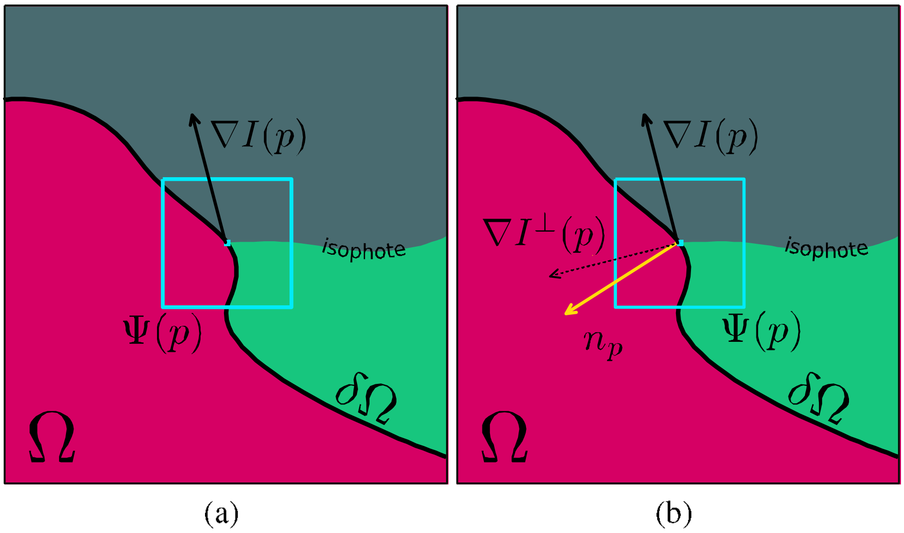


Figure 3.5: The data term is defined by the gradient of the patch as illustrated in (a). Vector $\nabla I^{\perp}(\boldsymbol{p})$ is orthogonal to the gradient and represents the direction of the isophote. The angle between its direction and the boundary normal depicted in (b) provides the scalar weight $D(\boldsymbol{p})$.

Figure 3.5 shows that the two vectors in (3.5) are parallel if an isophote is perpendicular to the boundary and, therefore, runs right into the unknown region. This results in a high value when calculating the inner product and in smaller values when closer to the orthogonality. Consequently, an isophote that hits the boundary in an obtuse angle produces a high data term value increasing its likelihood to be propagated into the target region.

The Priority

The priority value is eventually composed of these two terms balancing the textural and structural characteristics of each patch. The data term ensures that isophotes abutting the boundary are continued into the target region to avoid discontinuities, a typical artefact in texture synthesis. Its dominance however is limited by the confidence term to prevent incorrect isophote continuation. Together, they define a well-balanced priority that indicates the fill-order. The highest priority $P(\boldsymbol{p})$ then denotes the next patch $\Psi(\hat{\boldsymbol{p}})$ to be processed.

3.3.2 Finding the Best Match

In the next processing step, the best-matching sample $\Psi(\hat{\boldsymbol{q}})$ for the selected search patch has to be determined. Possible samples suffice an additional condition to improve the matching results. All values within the patch have to lay completely in the source region to be considered a valid sample patch. Its data in this patch will then be used to fill the missing pixels in the search patch, preventing the search patch from being filled with values from an incomplete patch.

The best sample can now be retrieved by comparing each sample to the search patch. Restrained by the presence of unknown values in the search patch, only the set of pre-filled pixels in the search patch can be compared to the corresponding set in each sample. Formally, this can be described as

$$\Psi(\hat{\boldsymbol{q}}) = \underset{\Psi(\boldsymbol{q}) \in \Phi}{\operatorname{argmin}} \quad d(\ \Psi(\hat{\boldsymbol{p}}), \Psi(\boldsymbol{q})\ ) \tag{3.6}$$

where $\Psi(\hat{\boldsymbol{p}})$ and $\Psi(\boldsymbol{q})$ denote search and sample patches respectively. The distance function $d(\Psi(\hat{\boldsymbol{p}}), \Psi(\boldsymbol{q}))$ incorporates the sum of squared differences of each two patches, which is similar to [EL99]. Although this seems to be a simple but effective solution, the RGB colour space does not resemble perceptual distances. Instead, this is achieved by computing the differences in the *CIE Lab* colour space described in [SH98]. It adds some additional computations, the distance calculation, however, is kept simple.

3.3.3 Filling the Search Patch

The retrieved best-matching sample is used in a final filling process to update the unknown pixel intensities. Criminisi et al. [CPT04] pointed out that simply updating the missing value in $\Psi(\hat{\boldsymbol{p}})$, with its correspondences in $\Psi(\hat{\boldsymbol{q}})$, produces the best results. In fact, limitations of other approaches, e.g. blurriness as result of diffusion [BSCB00], are prevented and texture, as well as structure, are maintained. Therefore, they even discourage further altering of the sample intensities to preserve these qualities.

Updating the Confidence

At the end of each iteration, the confidences are updated. It reflects the changes made to $\Psi(\hat{\boldsymbol{p}})$ by filling its unknown pixels in the previous step. The certainty of their colour intensities has decreased in respect to their samples. Modelling this decay of confidence involves only the pixels altered in the filling process. The update process involves only the previously filled pixel location in $\Psi(\hat{\boldsymbol{p}})$. Their new confidence value $C(\boldsymbol{r})$ is set to the average confidence $K(\hat{\boldsymbol{q}})$ given by (3.4). The update is formally defined as

$$C(\boldsymbol{r}) = K(\hat{\boldsymbol{q}}) \quad \forall \boldsymbol{r} \ \in \ \Psi(\hat{\boldsymbol{p}}) \cap \Omega \tag{3.7}$$

This ensures that the confidences dissolve as the boundary moves towards the centre of the target region resembling the reduced reliability of colour values for those pixel locations.

The major improvement to [EL99, LLX^{+}01] is the introduction of a fill-order that is automatically generated throughout the process compared to approaches where the fill-order is predefined and hard-coded into the process. In contrast, the priority value described in this chapter provides a fill-order determined by the characteristics of texture and structures. The data term especially contributes to an improvement in results because it reduces block-shaped and misaligned artefacts by enforcing the early propagation of edges into the target region. Hence, no additional processing is required to decrease the effect of those artefacts.

3.3.4 Important Extensions

After the work in [CPT04] was presented, it became a very well-cited approach in texture synthesis and related topics. Numerous adaptations and modifications were published proposing improvements to the original approach. In the following, two extensions are described that proposed successive works. The first is a simple extension that tries to prevent repetitive sampling [WO08]. The second [CHL^{+}05] investigated its predecessor to identify its weaknesses and proposed improvements to circumvent them.

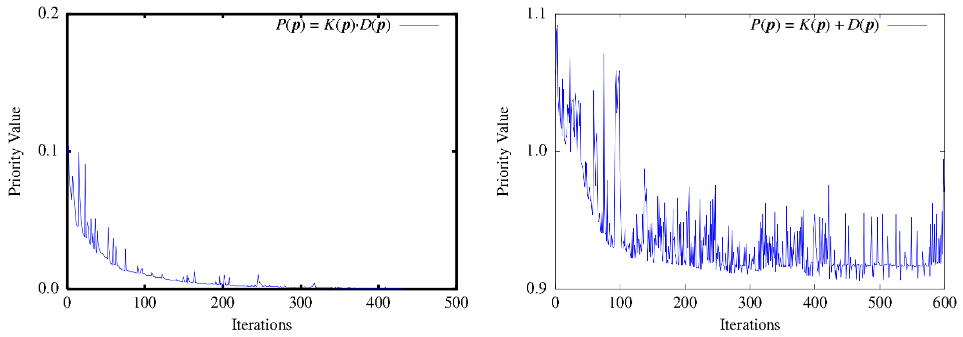
Sampling from Multiple Patches

A very straight-forward extension was developed by Wong and Orchard [WO08] integrating the generation of new pixel intensities instead of copying existing intensities. It is reasoned that the target region would not have exactly the same intensities as they can be found in the source region. The solution that is proposed describes the "mixing" of a patch with new intensities from a set of sample patches acquired by the same routine described in Chapter 3.3.2.

Instead of searching for *the* best-matching sample, a set of N samples with the highest similarities to $\Psi(\hat{\boldsymbol{p}})$ is returned, where the similarity is calculated using equation (3.6). These patches are then mixed together to create a template patch with colour intensities that are composed of corresponding pixel values in all N patches. A straightforward method would be to calculate the average of all patches and use it as a template. This would, however, result in a smearing of colours with a blurring target pixel intensities. Avoiding such blurry artefacts is achieved by weighting the samples in respect to their similarity to $\Psi(\hat{\boldsymbol{p}})$. The weight of patch $\Psi(\boldsymbol{q}_i)$ is, therefore, defined as

$$w(\ \Psi(\boldsymbol{q}_i)\ ) = exp\left(-\frac{c(\ \Psi(\hat{\boldsymbol{p}}), \Psi(\boldsymbol{q}_i)\ )}{h}\right) \tag{3.8}$$

where $c(\Psi(\hat{\boldsymbol{p}}), \Psi(\boldsymbol{q}_i))$ resembles the similarity distance between the search patch $\Psi(\hat{\boldsymbol{p}})$ and the sample $\Psi(\boldsymbol{q}_i)$. h controls the influence of a patch $\Psi(\boldsymbol{q}_i)$ on the mixed sample depending on its similarity to $\Psi(\hat{\boldsymbol{p}})$. A patch's influence decays slowly when h is small and decreases faster when a greater value is chosen for h. Although the weighing of the



(a) Criminisi Priorities

(b) Cheng Priorities

Figure 3.6: Plot (a) shows the highest priority at each iteration using [CPT04]. They drop quickly to zero as opposed to the priorities in (b) when calculating the priority proposed by [CHL$^+$05]. The plots are produced using a modification of the code provided by [1] and replicates those depicted in [CHL$^+$05].

composing sample patches apparently reduces wrong intensities, it still suffers from an increased energy in the patch. Hence, the sum of all weights is used to normalise each colour intensity, leaving the following formal definition:

$$\Psi(\hat{\boldsymbol{q}}) = \frac{\sum\limits_{i=1}^{n} w(\Psi(\boldsymbol{q}_i))\ \Psi(\boldsymbol{q}_i)}{\sum\limits_{i=1}^{n} w(\Psi(\boldsymbol{q}_i))} \tag{3.9}$$

The result is a sample patch that is a composition of intensities that prevents repetitive insertion as it occurs when sampling from similar nearby source regions. The plausibility of the generated colours, however, depends highly on the decay h. The quality of the inpainting results depends on a suitable h.

Modified Fill-Order Calculation

Cheng et al. [CHL$^+$05] chose a more investigative path in their work. They evaluated the fill-order priority and its composing terms by recording the highest priority in each iteration. The first observation was that the confidence term “drops rapidly to zero” [CHL$^+$05] during the whole process. This highly effects the priority value as a result of confidence and data term in (3.3) being multiplied. It, therefore, decays almost as quickly as the confidence term, which is illustrated in the plot in Figure 3.6(a).

This motivated a slight change of the priority equation (3.3). The multiplication is replaced by an addition to reduce the unproportional influence of either component on the overall priority. The new priority is defined as

$$P(\boldsymbol{p}) = K(\boldsymbol{p}) + D(\boldsymbol{p}) \tag{3.10}$$

which is the first step towards an improved progression of the priority. The dominance of the confidence value is reduced, which is depicted in Figure 3.6(b).

Another observation concerned the data term which is more consistent throughout the iterations than the confidence. This makes it difficult to compare or even balance them in a composed priority. Making the confidence more consistent also lead to a regularisation controlled by a single weighting factor ω, empirically set to 0.7. It is formulated as

$$R_K(\boldsymbol{p}) = (1 - \omega) \cdot K(\boldsymbol{p}) + \omega \quad \text{with } 0 \leq \omega \leq 1 \tag{3.11}$$

where ω denotes the value that is approximated by the decreasing curve. This avoids its decay to zero which, in turn, prevents the priority to be dragged towards zero as well.

Finally, they observed a discrepancy in quality when applying their method to images with very different texture and structure characteristics. A look at the original approach shows this effect was inevitable. Confidence and data term are supposed to balance each other out and are modelled into the priority calculation accordingly. This creates a rigid relation between both values. To achieve good inpainting results an image has to resemble this correlation between texture and structure.

A solution to reduce this limitation is to replicate the irregular correlation of structure and texture in a weighted priority function $R_P(\boldsymbol{p})$. The new priority is assembled from the regularised confidence (3.11) and data terms (3.5) multiplied by the weighting factors ω_C and ω_D respectively. Formally, it is defined as

$$R_P(\boldsymbol{p}) = \omega_C \cdot R_K(\boldsymbol{p}) + \omega_D \cdot D(\boldsymbol{p}) \tag{3.12}$$

with the weights justifying $\omega_C + \omega_D = 1$ and $0 \leq \omega_C, \omega_D \leq 1$. Indeed, these weights make it possible to adjust the fill-order to the distinct properties of an image, but they also increase the amount of required user interaction.

In summary, [CHL$^+$05] presents a modified calculation of the priority value reducing the domination of the confidence and improving the flexibility of the modelled balance between texture and structure properties. As a result, the priorities are more consistent with a better response to the data term.

3.4 Inpainting in the Wavelet Domain

The review of inpainting approaches in section 2.2 showed that several works attempted to exploit spectral properties of images for better results. A new idea, however, was to apply an exemplar-based technique in the frequency domain. Ignácio and Jung [IJ07] proposed an adjusted approach closely related to [CPT04], which processed wavelet coefficients instead of colour intensities. The image is transformed into the wavelet domain, the inpainting process is completed using frequency information and the inpainted result is retrieved by applying the inverse transform. A basic introduction to wavelets and the wavelet transform can be found in Appendix A.

As usual, an input image I and a binary mask defining Ω are expected. The first step is to transform them both into the wavelet domain using a *Haar* wavelet. This wavelet was chosen because of its *small support*. Applying wavelets to image and binary mask results in artificial edges especially at the boundary of the masked area in I and the target region in the binary mask. They result from high-pass filtering in the discrete wavelet transform described in Chapter A.3. The detail coefficients respond to those edges and produce a high-frequency response appearing as artificial edges. The extent of those undesired frequency responses depends on the wavelet and its characteristics. Choosing a wavelet with a small support restricts the effect locally to nearby coefficients. The Haar wavelet provides a very small support and is, therefore, the most suitable candidate for this application, although other wavelets, e.g. *Daubechies4*, could be used as well.

An additional precondition limits the wavelet transform to an one-level decomposition. It produces four subbands, illustrated in Figure A.6(a), containing the approximation coefficients in W_a and corresponding detail coefficients in horizontal W_h, vertical W_v and diagonal W_d direction respectively. The advantage of this constraint is that it is not required to correlate the coefficients in subbands of different levels. The algorithm can, therefore, operate on all four subbands simultaneously with corresponding coefficients identified by their location in respect to the top-left corner of their respective subbands. In addition to the implied simplifications, it reduces the inpainting domain to the size of a single subband.

A final preparation is required before the main procedure can start. It involves the generation of the binary mask in the wavelet domain denoted as Ω_W, with its boundary $\delta\Omega_W$. After it is decomposed into four wavelet subbands, the approximation coefficients in W_a almost represent a binary mask except for the previously explained spreading of coefficients along the edges. The same applies to the detail subbands W_h, W_v and W_d, where non-zero coefficients depict artificial edge information. As a result, a new subband-sized binary mask has to be created to define the target region. An element in this mask is described as part of the target region Ω_W if at least one of the coefficients in all four subbands is different to zero. The retrieved mask then provides the required data to start the iterative inpainting process by determining the fill-order.

3.4.1 Determining the Fill-Order

In equivalence to [CPT04], the key concept of this approach is the fill-order of its boundary patches controlled by a priority value assigned to each patch. The highest priority in each iteration denotes the patch centre of the next search patch $\Psi(\hat{\boldsymbol{p}})$. The priority value is composed of three terms, denoted as *components*, reflecting similar properties to those described in the data and confidence term. More precisely, the confidence component represents the reliability of the coefficients enclosed in a patch, whereas its structural properties are measured by *structure significance* and *structure orientation*, which were originally named *edge strength* and *edge orientation* in [IJ07]. This work refrains from using these terms for two reasons. Firstly, the same terms have a different meaning in image processing techniques concerned with edge detection. Secondly, the structural components are composed of detail coefficients in W_h, W_v and W_d responding to high frequencies due to high-pass filtering. Although these high frequencies are related to edge direction and edge strength [Wal08], they are not equivalent to edges in the spatial domain. Their exact meaning is described as follows.

The Structure Significance

The structure significance is a measure introduced to provide information on the *intensity* of structural information. Under the previous assumption of edge-related information being implicitly incorporated into the coefficients, this can be described as an indicator for the strength of an edge in a patch-sized neighbourhood. Considering a search patch $\Psi(\boldsymbol{p}) \in \delta\Omega_W$, the structural significance is described as the energy of its highest detail coefficient. Formally, it is written as

$$S_S(\boldsymbol{p}) = \max_{\boldsymbol{r}\in\Psi(\boldsymbol{p})} \quad \max_{b=\{h,v,d\}} \{(W_b)^2(\boldsymbol{r})\} \tag{3.13}$$

where W_b denotes the coefficients in the horizontal, vertical and diagonal detail subbands respectively. In short, all squared coefficients in the three subbands are compared and the highest energy is selected to represent the structural energy of the patch.

The Structure Orientation

The structure orientation is the equivalent to the data term that is used in [CPT04]. It represents the direction of the linear structure, or isophote, in a patch. Once again, the orientation is calculated as the inner product of two vectors $\nabla I^{\perp}(\boldsymbol{p})$ and $\boldsymbol{n_p}$. $\nabla I^{\perp}(\boldsymbol{p})$ denotes the direction of the structure in $\Psi(\boldsymbol{p})$ orthogonal to its gradient $\nabla I(\boldsymbol{p})$, whereas $\boldsymbol{n_p}$ is orthogonal to the boundary $\delta\Omega_W$ at the patch centre. In contrast to [CPT04], where $\nabla I(\boldsymbol{p})$ is the most significant gradient vector in the patch, [IJ07] exploits the properties of the wavelet coefficients in the horizontal and vertical subbands, W_h and W_d. Both subbands contain coefficients representing the high-frequency response in horizontal and vertical direction respectively [Wal08]. Consequently, the coefficients in W_h and W_v can be used to approximate the gradient vector of the patch centre $\boldsymbol{p}$ as follows

$$\nabla I(\boldsymbol{p}) = \begin{bmatrix} W_h(\boldsymbol{p}) \\ W_v(\boldsymbol{p}) \end{bmatrix} \tag{3.14}$$

The retrieved approximated gradient $\nabla I(\boldsymbol{p})$ has to be normalised before the computation of the structure orientation is concluded. This ensures that the strength of an isophote is exclusively encoded in the structure significance of a patch $\Psi(\boldsymbol{p})$, whereas the orientation is provided by the structure orientation. The representative scalar value, denoting the orientation, is, therefore, defined as

$$S_O(\boldsymbol{p}) = |\nabla I_n^{\perp}(\boldsymbol{p}) \cdot \boldsymbol{n_p}| \tag{3.15}$$

where the normalised gradient approximation is described by

$$\nabla I_n^{\perp}(\boldsymbol{p}) = \frac{\nabla I^{\perp}(\boldsymbol{p})}{|\nabla I^{\perp}(\boldsymbol{p})|} \tag{3.16}$$

The structure orientation takes on a high value when the structure direction is almost parallel to the boundary normal and decays to zero as the vectors become orthogonal. The correlation between the two vectors is illustrated in Figure 3.7.

The Confidence

The confidence value expresses the same reliability for coefficients, in the same way as the confidence value in [CPT04] for pixel locations. The confidence in [IJ07] is not assigned to a single coefficient but to a pixel position in the subband, assigning the same confidence $C(\boldsymbol{r})$ to the four coefficients at position $\boldsymbol{r}$ in the respective subbands. This is justified by inpainting the image on a subband-level, as described at the beginning of this chapter.

Apart from this minor difference, the confidence is initialised in the same way setting it to 0 for each pixel position in Ω, whereas all other positions are set to 1. Whenever a patch $\Psi(\hat{\boldsymbol{p}})$ is updated with coefficients from $\Psi(\hat{\boldsymbol{q}})$, new confidences are assigned to these positions based on the average confidence in $\Psi(\hat{\boldsymbol{q}})$. Its degeneration is induced by a decay parameter α_C leading to the update equation

$$C(\boldsymbol{r}) = \alpha_C \; K(\hat{\boldsymbol{q}}) \tag{3.17}$$

The decay is limited to $0 \leq \alpha_C \leq 1$ with a suggested value of 0.8 in [IJ07]. The confidence value for a search patch $\Psi(\boldsymbol{p})$ is then calculated according

$$K(\boldsymbol{p}) = \frac{\sum\limits_{\boldsymbol{r} \in \Psi(\boldsymbol{p}) \cap \Omega_W} C(\boldsymbol{r})}{|\Psi(\boldsymbol{p})|} \tag{3.18}$$

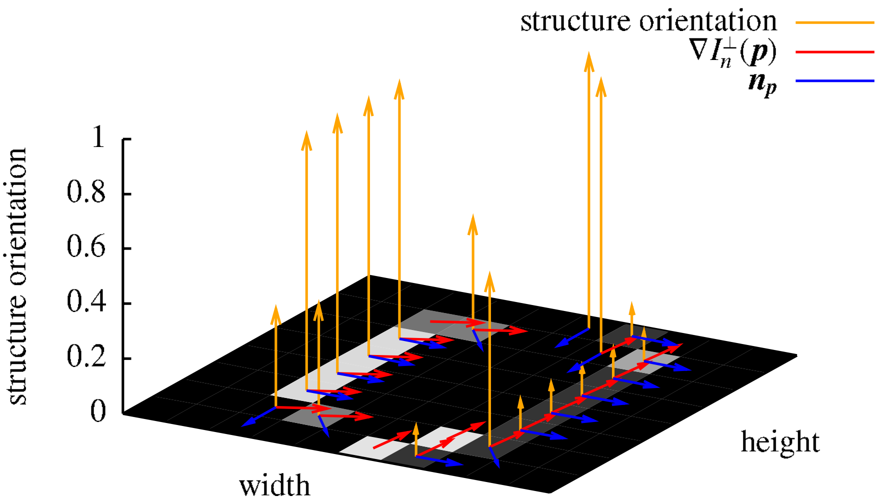


Figure 3.7: The small example image in the x-z-plane is a small example depicting the priorities as grey level values. The calculated orientation values are equal to the length of the vertical vectors (orange). The orthogonal gradient $\nabla I_n^{\perp}(\boldsymbol{p})$ (red) and the boundary normal $\boldsymbol{n_p}$ (blue) are their compositing vectors depicting the relationship between their respective angles and the resulting orientation.

where $|\Psi(\hat{\boldsymbol{p}})|$ is the number of confidences in search patch $\Psi(\hat{\boldsymbol{p}})$.

The Priority

The priority for a patch centre $\boldsymbol{p}$ is now determined by the product of these three components. The structure orientation and structure significance provide the weight on structural information. The confidence weighs the reliability of coefficients to promote patches with a smaller amount of unfilled coefficients. Combining the weights produces the following equation to compute the priority:

$$P(\boldsymbol{p}) = S_S(\boldsymbol{p}) \cdot S_O(\boldsymbol{p}) \cdot K(\boldsymbol{p}) \tag{3.19}$$

The fill-order is then determined by calculating the priority for each patch centre $\boldsymbol{p} \in \delta\Omega_W$ and retrieving the highest priority. The patch with the highest priority is the next search patch, which is then compared to sample patches to find the best-matching sample $\Psi(\hat{\boldsymbol{q}})$ for filling the missing region.

3.4.2 Finding the Best Match

The best-matching sample patch $\Psi(\hat{\boldsymbol{q}})$ is the subject of the next step in the iterative process. It is a patch of the same size and shape as $\Psi(\hat{\boldsymbol{p}})$, as defined in section 3.1, that is the best match for the search patch $\Psi(\hat{\boldsymbol{p}})$. Ignácio and Jung defined "best match" as the smallest difference between patches $\Psi(\hat{\boldsymbol{p}})$ and $\Psi(\boldsymbol{q})$. The similarity measure they used minimises the *sum of squared differences* (SSD) between corresponding coefficients in a search and a sample patch. It is the same metric as in [TWJ06, TWJ05], where it defines the distance for synthesising a large texture from a sample. Although good results for plain texture synthesis can be produced with this approach, it is limited by its fixed patch size. A similar observation was demonstrated in [DCOY03] justifying the use of fragments (the circular equivalent to patches) of variable size. It is argued that in real images the textural and structural patterns vary from image to image and even within the same one.

In [IJ07] this idea is apprehended by using an adaptive matching algorithm to find the best sample. Instead of comparing only samples of the same size M, they proposed to compare a range of sizes $M \in \{M_{min}, \ldots, M_{max}\}$. $\Psi(\hat{\boldsymbol{p}})$ is resized in respect to its centre $\hat{\boldsymbol{p}}$ for each size M and compared to all candidates $\Psi(\boldsymbol{q})^M$ of equal size. This results in a best match for each size M. Finding the most similar patch between those of different sizes requires a normalisation of the similarity. Otherwise, the smallest would always provide the shortest distance to $\Psi(\hat{\boldsymbol{p}})$. The distance of two patches $\Psi(\boldsymbol{q})$ and $\Psi(\hat{\boldsymbol{p}})$ is then given by

$$\Delta_D = \sum_{b \in \{a,h,v,d\}} \sum_{(x,y) \in \Psi(\hat{\boldsymbol{p}}) \cap \Phi_W} (\Psi_b(\boldsymbol{q})(x,y) - \Psi_b(\hat{\boldsymbol{p}})(x,y))^2 \tag{3.20}$$

which is, in turn, normalised to

$$|\Delta_D| = \frac{\Delta_D}{|\,\Psi(\hat{\boldsymbol{p}}) \cap \Phi_W\,|} \tag{3.21}$$

The best sample $\Psi(\hat{\boldsymbol{q}})$ is then defined as the patch minimising $|\Delta_D|$ for all sizes M. The search for $\Psi(\hat{\boldsymbol{q}})$ with a variable size, however, comes at a price. The computation is increased by $M - 1$ searches, compared to a fixed-size search. As a counterweight, the search is restricted to a search window $\Theta(\hat{\boldsymbol{p}})$. Its size is motivated by the connectivity principle assuming that neighbouring patches are more likely to minimise the distance to $\Psi(\hat{\boldsymbol{p}})$ then others. The search window $\Theta(\hat{\boldsymbol{p}})$, centred at $\hat{\boldsymbol{p}}$, has the defined size $3M \times 3M$ in [IJ07].

3.4.3 Filling the Search Patch

The sample patch $\Psi(\hat{\boldsymbol{q}})$, retrieved by the previous search, provides the required coefficients to fill the unknown regions in all four subbands. Earlier in this chapter it was stated that the subbands are processed simultaneously. The same applies to the filling of the search patch. The patches $\Psi(\hat{\boldsymbol{p}})$ and $\Psi(\hat{\boldsymbol{q}})$ provide a centre position $\hat{\boldsymbol{p}}$ and $\hat{\boldsymbol{q}}$ respectively as well as a patch size. This information is used to identify the corresponding regions in the subbands. The search patch $\Psi(\hat{\boldsymbol{p}})$, for instance, encloses unknown coefficients in W_a and $\Psi(\hat{\boldsymbol{q}})$ marks corresponding sample coefficients in the same subband. The missing coefficients in $\Psi(\hat{\boldsymbol{p}})$ are then simply copied from $\Psi(\hat{\boldsymbol{q}})$. The detail subbands are processed in the same way.

An update of the components is the last step to be carried out in an iteration after $\Psi(\hat{\boldsymbol{p}})$ was filled. At first, the new boundary positions are computed by updating Ω_W and $\delta\Omega_W$. For each new patch centre $\boldsymbol{p}$, the confidence is updated according to (3.17) before structure significance and structure orientation are calculated. Finally, the priority values are recomputed from equation (3.19). This whole process is iteratively repeated until all unknown coefficients are filled.

Chapter 4

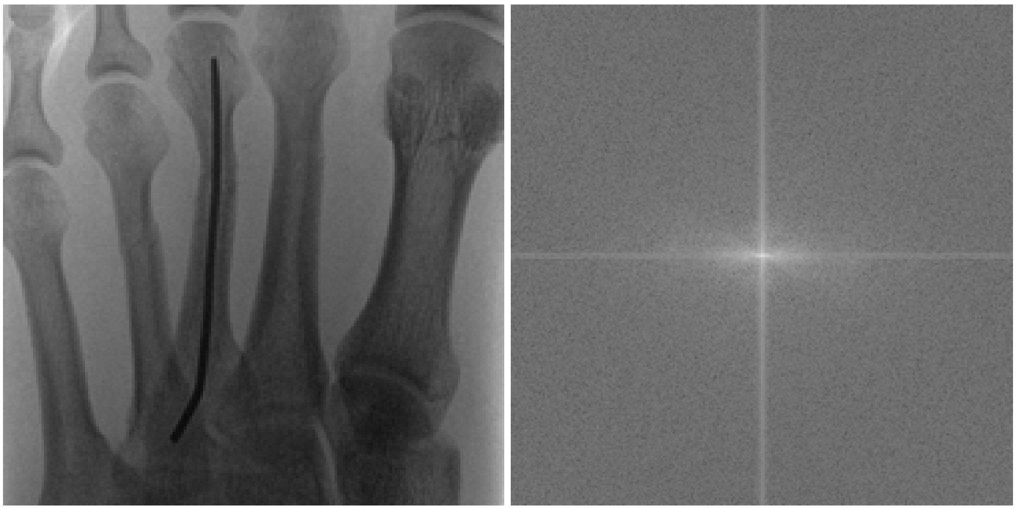
Evaluation of Inpainting Techniques

The comprehensive explanation of the three approaches in Chapter 3 provided an insight into their concepts without considering their respective quality. These properties are the subject of the comparison carried out in this chapter. The inpainting quality as well as problems and limitations are described and illustrated for either approach. The defect pixel interpolation [AM01] is examined in section 4.1, followed by the exemplar-based image approach [CPT04] in section 4.2. Finally, the wavelet-based approach [IJ07] is investigated in section 4.3.

The three approaches are examined according to the scenario described in Chapter 1, where a colour checker has to be removed. This defines the environment as the campus of the University of Koblenz which implies that images contain mostly structure-rich and complex architectural content. A representative set of images has been selected containing different views of the campus and resembling the characteristic content. These images and their respective binary masks are shown in Appendix B and are used throughout the following sections.

4.1 Interpolation in the Fourier Domain

The concept of the Aach approach to reconstruct missing pixel intensities in X-ray images is described in section 3.2. It is widely used in medical applications, implying that it provides exceptional interpolation results for its specific operative range. This is a basic requirement for any algorithm used in the medical field to ensure highly reliable images and prevent misdiagnosis through malicious image processing. An example image in Figure 3.4(c) shows a regular X-ray image of a foot. The interpolation result in Figure 3.4(c) depicts the result of the unoptimised MATLAB® [8] implementation. It has to be noted, however, that the interpolation quality differs slightly from the results presented in [AM01]. The reconstructed images are nonetheless sufficient for this investigation.



(a) (b)

Figure 4.1: Image (a) shows a common X-ray images of a foot as it was used in [AM01] to illustrate the result of interpolation. Its Fourier spectrum is depicted in (b). Its dominantly low frequencies are represented by the horizontal and vertical frequency bands crossing at the center.

It was already stated that the main application of [AM01] is the interpolation of X-ray images. Before this algorithm is applied to real images it is helpful to look at some general characteristics of its input data and the spectral decomposition carried out by Fourier analysis. The input data is an X-ray image similar to that in Figure 4.1(a). It exposes a general property of these images that is closely related to their method of acquisition. The X-rays penetrating the patients body, in this case the foot, are absorbed by the tissue and bones. On the other side, the remaining energy of each ray is measured and produces the output image. The degree of detail in such an image is, therefore, very limited. It is slightly noisy with multiple regions of homogeneous intensity. Furthermore, the number of edges is limited to the outlines of the bones which are less distinctive and more transcending into the background. In terms of Fourier analysis, the lack of edges is reflected in a dominantly low-frequent representation. It can be substantiated based on the spectral representation in Figure 4.1(b). The response to low frequencies towards the centre of the spectrum is much denser then in the outer areas of the image. In fact, the spectral information is mostly restricted to a narrow vertical and horizontal band as well as a circular region in the centre. These correspond mainly to low frequencies with a few higher ones at their outer ends. X-ray images in general can, therefore, be expected to be low-frequency dominated.

It is important to be aware of this spectral characteristic because the main concept of [AM01] and [Pap75] builds on exploiting the periodicity of frequencies in Fourier space.

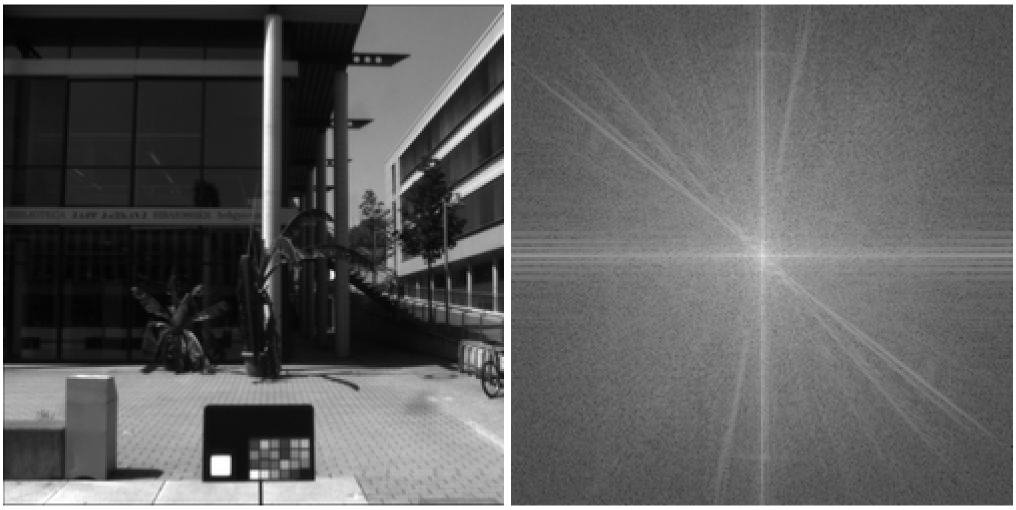
Section 3.2 describes that missing pixel intensities are reconstructed by estimating a corrected frequency spectrum for the X-ray image. This can be achieved by exploiting the property of Fourier analysis that its frequency representation discards spatial information and exhibits only periodic frequencies, implying that each frequency has a global influence on the image intensities in position space. Frequency coefficients, therefore, correspond to a combination of a certain vertical and horizontal periodic frequency whereas a pixel intensity is composed of different magnitudes of all these frequencies. On the one hand, this dependency allows recovery of unknown pixel intensities from its known frequencies. The estimation of a corrected spectrum on the other hand has a global influence on all pixel intensities. Although this effect is limited in [AM01] by including the information about unknown intensities into the interpolation, this fact has to be kept in mind when applying the procedure to real images.

Application to Real Images

The application of [AM01] to a real image is evaluated using an example image taken on campus at the University of Koblenz, as shown in Figure 4.2(a). It is a frontal view of the library with a colour checker located at its bottom centre. The target region is provided by a user-defined binary mask identifying the pixels belonging to the colour checker. The image itself is converted to grey level intensities due to the limitation of the common Fourier transform to single channel images. Alternatively, a *quaternion*-based Fourier transform [SH98] would be able to handle colour images directly, its use, however, would exceed the scope of this work. The colour intensities are, therefore, removed from the image by desaturating it using the *HSV* colour space. The obtained grey level image can then be processed by the examined approach with its iterative interpolation controlled by selecting a fixed number of iterations. The results produced by applying the interpolation algorithm with 100, 400 and 2000 iterations are depicted in Figure 4.3 using the image in Figure 4.2(a) as input.

Looking closer at these results reveals the shortcomings of the algorithm when applied to real images. The image in Figure 4.3(a), obtained by iterating 100 times, illustrates the impact of estimating the unknown intensities using periodic frequencies. Reconstructed intensities in the target region indicate a horizontal and vertical pattern introduced by composing them from multiple frequencies in the same directions. The most dominant frequencies in the image are dominating the target regions as well which leads to the blurred region resembling the shape of the pillar in the background. The influence in the frequency domain is very high and, therefore, contributes predominantly to the estimated frequencies. Increasing the iterations to 400 in Figure 4.3(b) or 2000 in Figure 4.3(c) distorts the regular pattern in the target region of Figure 4.3(a), the quality of the reconstructed intensities, however, are not improved at all.

In addition, all three results in Figure 4.3 show that the interpolated target regions are smoothed as if they were processed using a low-pass filter. Sharp and distinctive

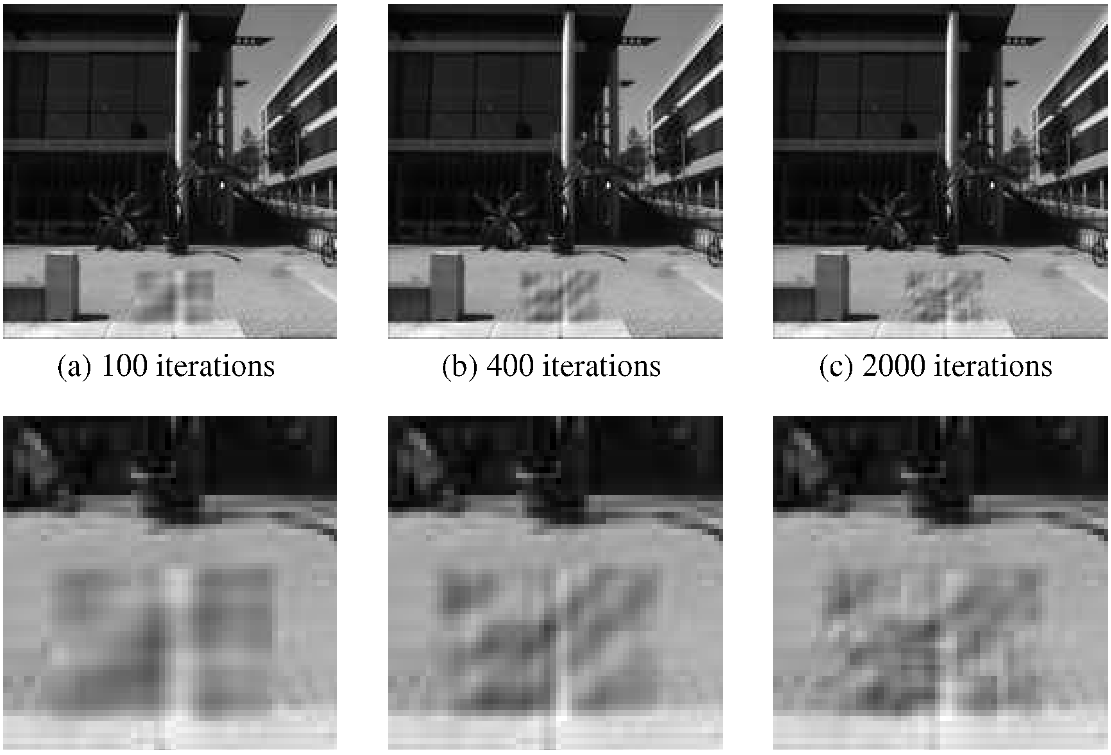


(a) (b)

Figure 4.2: An exemplar image taken on campus at the University of Koblenz-Landau is shown in (a). It is a frontal view of the library with a colour checker positioned at the bottom centre. Its Fourier spectrum is shown in (b) illustrating the diversity of frequencies incorporated in the image. Compared to 4.1(b), it has an increased response to higher frequencies.

edges are almost non-existent. The reason for such a smooth reconstruction is the better response of [AM01] to low frequencies which is a result of the periodic properties of the Fourier transform described above. Low frequencies have a large impact on pixel intensities because their resemblance of large wavelengths dominates the shorter wavelengths represented by higher frequencies. This is then reflected in the recombination of frequencies when applying the inverse Fourier transform. Real images, however, contain more high-frequency information as Figure 4.2(b) illustrates compared to Figure 4.1(b). Consequently, it is more difficult if not impossible to reconstruct all details in a real image by applying [AM01].

In summary, it can be stated that this approach is highly effective at interpolating images with dominantly low frequencies as in X-ray images. Its capabilities of reproducing high frequencies, however, are limited which makes it unsuitable for real images, especially for the campus images since they contain even more edges and structural data due to their architectural content. This approach is, therefore, not considered any further due to its poor results and restricted potential for this application.



(a) 100 iterations

(b) 400 iterations

(c) 2000 iterations

(d) 100 iterations (detail)

(e) 400 iterations (detail)

(f) 2000 iterations (detail)

Figure 4.3: Images in the first row show the results of frequency interpolation using 100 (a), 400 (b) and 2000 (c) iterations. The colour checker in the original image, shown in Figure 4.2(a), was removed by a provided binary mask corresponding to the colour checker. (d), (e) and (f) exhibit detail excerpts from their respective original above.

4.2 Exemplar-Based Inpainting

The realisation that frequency interpolation fails on real images lead to the conclusion that more subject-related research in the field of image inpainting should be considered. Chapter 2 describes that recent developments in texture synthesis and structure inpainting produced promising techniques. A major approach in this field of research is [CPT04] which is explained in section 3.3. As an algorithm that is well-cited and produces widely respected results it provides a more suitable solution for the described problem. In addition, it has been extended by [CHL$^+$05] described in section 3.3.4 with suggested improvements on the inpainted results. They will, therefore, be subject to a combined evaluation in this chapter.

The application used to generate the inpainted images for both approaches relies on a MATLAB® [8] implementation by Sooraj Bhat [1]. It provides the code to generate images as described in [CPT04]. In additon, it was extended as part of this work to include the modifications proposed in [CHL+05]. It is limited to a non-adjustable priority calculation as opposed to the description in section 3.3.4. It implements the regularised confidence equation (3.11) and additive priority equation (3.10) which provides sufficient results for the following comparison.

4.2.1 Investigating the Fill-Order

One of the main ideas introduced by [CPT04], as described in section 3.3, is to control the order in which the patches are filled. It is stated that the order is a very crucial element of the filling process in general and their algorithm in particular. This makes it the first step in the algorithm to investigate. It is very difficult, however, to measure the quality of the fill-order since there is no "correct" order of filling unknown intensities. Too many unknown parameters influence the outcome of a particular order such as the number of isophotes abutting the boundary, different textures and varying structures. This makes a direct comparison of the fill-order in [CPT04] and [CHL+05] impossible. It can only be evaluated by comparing the resulting images of both approaches.

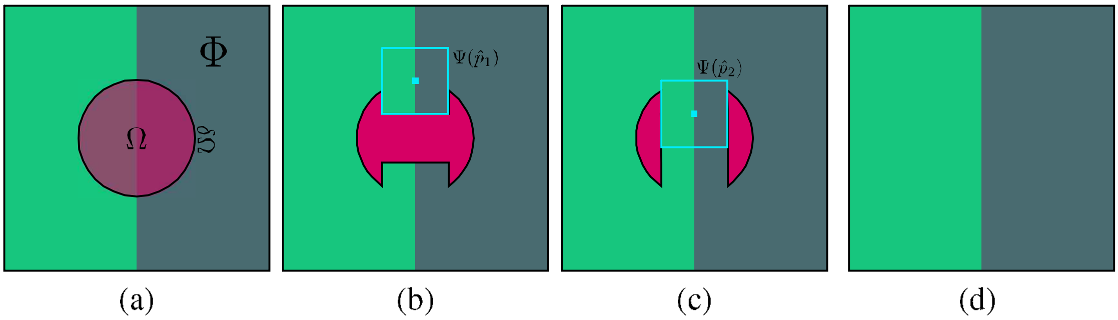
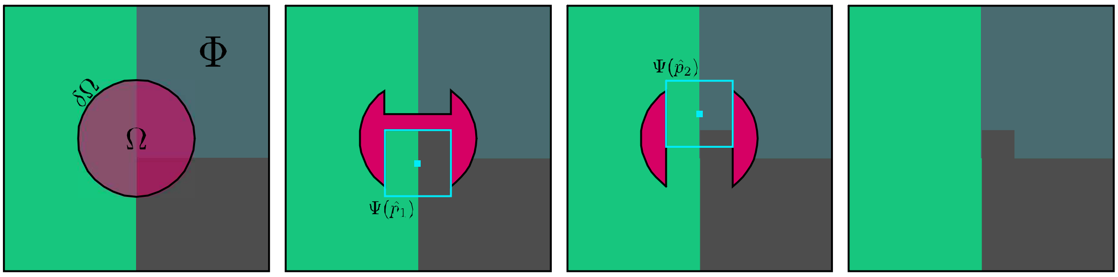


Figure 4.4: A circular target region is removed from the ideal image (a) containing two textures of different intensities. The successive images (b), (c) and (d) show the process of filling the unknown pixel intensities along the separating isophote first and then the remaining textures. The result is a perfect reconstruction of the ideal image.

A schematic example illustrates the difficulty of comparison. The main idea behind the Criminisi approach is to reconnect isophotes before filling in the unknown pixel intensities in the remaining textured areas. This is a very straightforward idea looking at a synthetic image consisting of two regions as shown in Figure 4.4(a). Each region has one intensity assigned to each of its pixels which creates a border between the regions that is the only edge in the image. A certain amount of the image is now removed and the missing intensities are to be reconstructed by one of the inpainting algorithms. The perfect way of inpainting this specific image would be to follow the routine of conservators as described in section 3.3. At first, the edge is continued through the target region and connected to its counterpart, as shown in Figure 4.4(b) and Figure 4.4(c). Afterwards, the remaining regions are filled with the intensity of their respective region. The result is a perfectly reconstructed image, illustrated in Figure 4.4(d).

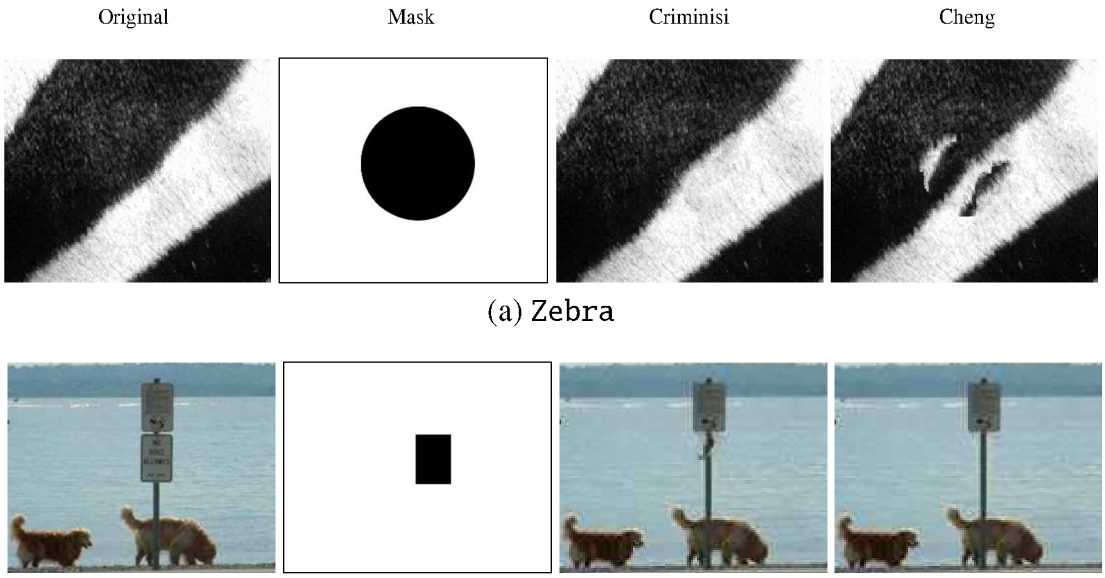


(a) (b) (c) (d)

Figure 4.5: Image (a) depicts three textures of different intensities creating two isophotes intersecting in the target region. Applying the isophote-driven inpainting procedure continues the vertical isophote first, as shown in images (b) and (c). The reconstructed image (d) exposes the problem of intersecting isophotes with a displaced horizontal isophote.

Adding another texture and, therefore, another isophote to such an image, shown in Figure 4.5(a), reveals the difficulty when inpainting real images. Applying the same strategy as above struggles with this type of image as shown in the images in Figure 4.5(b) and Figure 4.5(c). The reconstructed image in Figure 4.5(d) contains a displaced isophote as opposed to its original in Figure 4.5(a). To reconstruct it correctly a more sophisticated fill-order has to be determined that enforces a simultaneous inward motion of all isophotes to enhance the possibility of reconnection towards the centre. Real images with a multitude of isophotes that have to be connected within the target region make it even more difficult to find a suitable fill-order. The Criminisi approach [CPT04] and the Cheng extension [CHL$^+$05] each define such a fill-order that is tailored to the complexity of real images.

Looking at the results in Figure 4.6 shows that altering the fill-order produces very different results indeed. The close-up of a zebra skin in Figure 4.6(a) shows that the Criminisi algorithm reconnects the border separating the black and white segments and fills the remaining areas convincingly. The extended version [CHL$^+$05] using additive priority calculation, however, suffers from artefacts that are introduced in the textured areas. Interestingly, the isophote is still connected correctly but the remaining pixels are filled wrongly due to their fill order. The second image in Figure 4.6(b) reveals the exact opposite. The obtained pixel intensities using the original approach introduce wrong intensities that are sampled from one of the dogs' tails whereas the Cheng extension provides a good result connecting the edges of the pole correctly. A direct comparison of



(a) `Zebra`

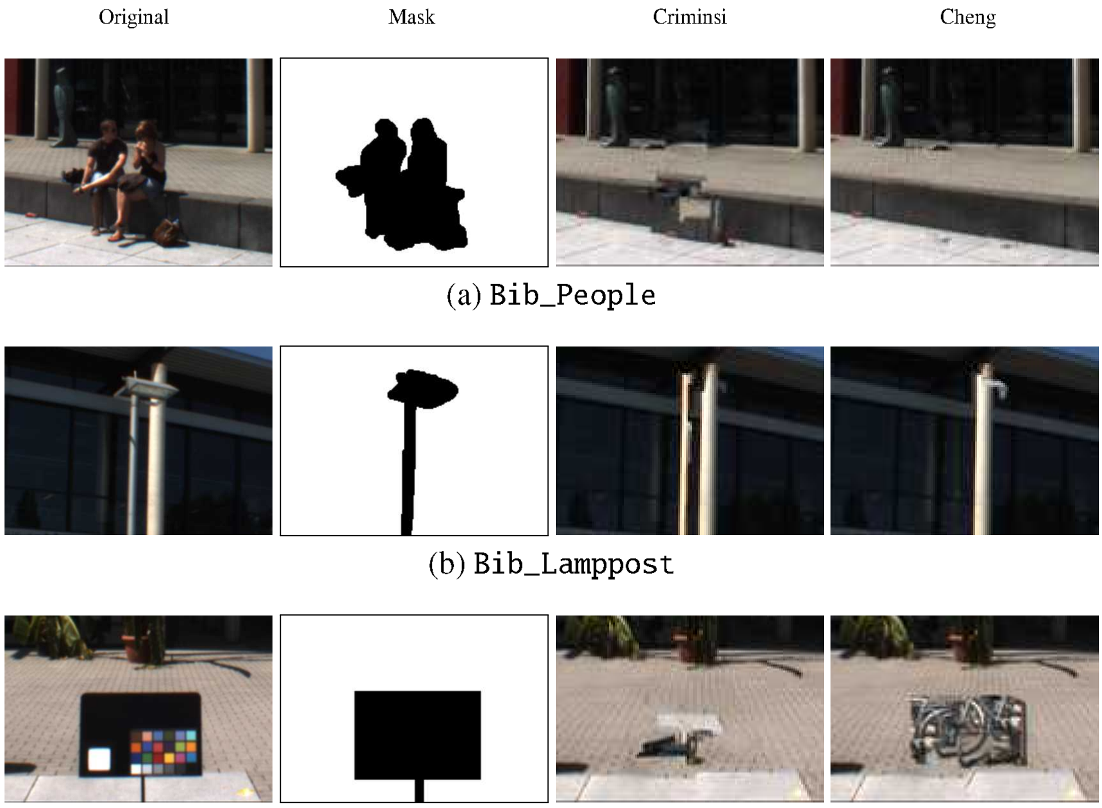
(b) `LakeWithDogs`

Figure 4.6: Inpainting results obtained by the Criminisi approach and the Cheng extension. Both examples are used in the original Criminisi proposal [CPT04].

these two images exhibits the importance of the fill-order as it was described in [CPT04]. Simply changing the priorities and, therefore, the fill-order produces different inpainting results. Their quality depends on the specific image and the suitability of the fill-order to reconstruct its structural and textural regions. This observation is emphasised in [CHL+05] where it is assumed that adjusting the correlation between the confidence and data term individually for an image is required to obtain sufficient inpainting results. It allows to make the fill-order more dependent on its image-specific characteristics.

In the case of Figure 4.6(b), the fill-order produced by the Cheng extension is better as opposed to the images in Figure 4.6(a), where the one calculated by the Criminisi approach is more suitable. This illustrates very well the problem of statically modelling the correlation between data and confidence terms. The outcome of an algorithm is bound to the conformity of its modelled fill-order calculation and the corresponding correlation between textural regions and isophotes in the input image.

A more challenging set of images is provided by those taken on campus. Their architectural content makes them more structured and detailed with a large number of isophotes running in different directions. Three examples are given in Figure 4.7 where the leftmost image is the original, the second depicts the target region and the following two are the results produced by the Criminisi approach [CPT04] and the Cheng extension [CHL+05] respectively. A look at these images reveals that the Cheng extension [CHL+05] provides much better results for this specific type of images with highly



(a) `Bib_People`

(b) `Bib_Lamppost`

(c) `Bib_ColourChecker_BikeStand`

Figure 4.7: Details of inpainted images of the campus image test set. The inpainting results are obtained using the Criminisi and Cheng approaches repsectively.

structural contents. Applying the Criminisi approach [CPT04] in contrast introduces visible artefacts in all three images. The excerpt in Figure 4.7(a) shows, for instance, that edges confining the dark stone wall are not continued correctly and pixel intensities belonging to the pavement are inserted into the wall stones. The same image computed using the Cheng extension [CHL$^+$05] exposes correctly continued and connected edges. Although the pavement in the front and the back of the wall are distorted the overall impression is much improved. Similar observations apply to the second set of results in Figure 4.7(b). [CPT04] inserts an artificial pillar next to the existing one whereas the extended algorithm creates a much better result with only a few artefacts at the top of the pillar. These two examples indicate the superiority of [CHL$^+$05] in this specific application due to a fill-order that is more suitable to their structure and contents. The respective fill-order in [CPT04] fails because its priority calculation does not resemble their image-specific correlation of structure and texture. [CHL$^+$05], however, is not a perfect solution. It also struggles to provide good results for some images such as the one in Figure 4.7(c). It can be observed that the priority calculation used in [CHL+05] does not respond well to the small isophotes in the pavement. Adding confidence and data terms in equation (3.10) allows textural information incorporated in the confidence to dominate the inpainting process. This corresponds to an unsuitable fill-order which encourages additional problems in the sampling step.

4.2.2 Investigating the Sample Search

The main problem with the search for the best-matching sample is that it returns a sample patch $\Psi(\hat{q})$ that is wrongly considered to be the best match. This becomes an even more important issue when looking at the implications. Using an unsuitable $\Psi(\hat{q})$ to fill unknown intensities in $\Psi(\hat{p})$ distorts the priority calculation in the next iteration and, therefore, has a negative influence on the fill-order. This, in turn, increases the likelihood of further errors in finding the next sample patch. Hence, it is impossible to pinpoint the reason for a bad inpainting result on either of the two steps. This makes it important to understand the problems that can occur when searching for $\Psi(\hat{q})$.

Patch Distance Calculation

As described in section 3.3.1, a best-matching sample $\Psi(\hat{q})$ for the selected search patch $\Psi(\hat{p})$ is determined by their minimum distance based on the SSD of pixel intensities. This distance is computed for each potential sample located anywhere in the source region and the search patch. In [CPT04] and [CHL+05] this leads to the problem of finding a sample that has the smallest distance but is contextually different to the search patch. This can be observed in the images in Figure 4.7(c) where fragments of a bicycle wheel are inserted into the pavement. The corresponding intensities are sampled from somewhere else in the image which is the result of a search for samples in the whole source region.

The miscalculation of the distance between $\Psi(\hat{p})$ and $\Psi(\hat{q})$ can have one of two reasons. Assuming the selected $\Psi(\hat{p})$ contains pixel locations that correspond to homogeneous texture as depicted in the images in Figure 4.8(a) and 4.8(c). The search patch in both images contains known pixel intensities in the right half and unknown ones in the left half. Searching for the best-matching sample for $\Psi(\hat{p})$ is then carried out by comparing the distances of all possible sample patches. The first problem that can occur is illustrated in Figure 4.8(a). A sample patch located at an isophote contains intensities equal to those that are known in $\Psi(\hat{p})$. Their distance will, therefore, be zero and the unknown intensities in $\Psi(\hat{p})$ are filled with corresponding intensities in $\Psi(\hat{q})$. This produces an isophote in the resulting image depicted in Figure 4.8(b). Furthermore, this artificial isophote has a negative influence on the updated fill-order in the next iteration and on the corresponding search for a sample patch. The second image 4.8(c) shows a

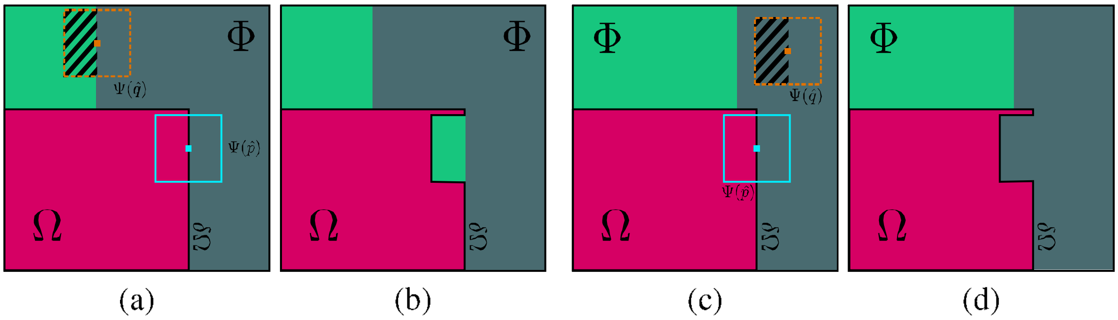


Figure 4.8: Image (a) illustrates a search patch $\Psi(\hat{\boldsymbol{p}})$ and a best-matching sample $\Psi(\hat{\boldsymbol{q}})$ that is a perfect match for the known pixel locations but introduces an artificial isophote in (b) due to its location on an isophote. (c) shows the contrasting example where the isophote is interupted in (d).

similar artefact created by a sample patch that is completely located in one of the textured regions. The right half of the patch contains intensities comparable to $\Psi(\hat{\boldsymbol{p}})$. The intensities enclosed in the other half of the patch are once again used to fill $\Psi(\hat{\boldsymbol{p}})$. In contrast to the previous example the isophote between the two textures is disrupted by unsuitable intensities in the sample patch. The lamppost image in Figure 4.7(b) exhibits these artefacts when [CPT04] is used for inpainting. The selection of suboptimal sample patches create an artificial pillar next to the real one with a frayed right edge. It illustrates the effect caused by an insufficient sample search and its recursive influence on the inpainting quality.

Patch-Shaped Artefacts

Another sampling artefact that can be observed in both algorithms concerns patch-shaped artefacts. They are a direct result of the fill-order that is defined to propagate the source region from the boundary inwards towards the centre of the target region. Multiple textures are, therefore, simultaneously growing into the target region which leads to the necessity of neighbouring textures to be adjoined at some point in the process. Depending on the similarity of these textures this can result in a visible seam that is usually rectangular due to the shape of the patches. Figure 4.9(a) shows a peninsula of unknown pixels that extends into the source region. The neighbouring texture regions contain different intensities and are separated by a distinguishable isophote. Assuming that the selected search patch $\Psi(\hat{\boldsymbol{p}})$ is placed on this peninsula, the locations denoting unknown intensities should continue the isophote abutting the boundary from the top. The sample patch $\Psi(\hat{\boldsymbol{q}})$ with the smallest distance to $\Psi(\hat{\boldsymbol{p}})$ is located in the rightmost texture and contains only homogeneous intensities. $\Psi(\hat{\boldsymbol{p}})$ is then only filled with intensities of this texture and an artefact is created corresponding to the shape of the peninsula depicted in Figure 4.9(b). The isophote in the first image is now displaced due to the wrong sample patch providing the unknown intensities. It shows that even in a very simple example the quality of the algorithm can be reduced by erroneous filling of unknown intensities.

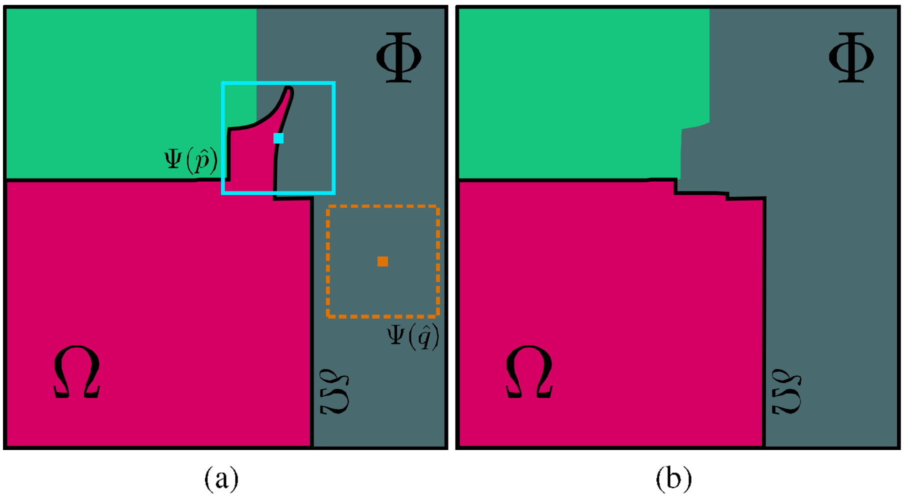


Figure 4.9: Image (a) shows a search patch $\Psi(\hat{\boldsymbol{p}})$ enclosing a peninsula of target pixels between two different textures. An insufficient sample patch $\Psi(\hat{\boldsymbol{q}})$ selected from the right-most textures procudes the displaced isophote in (b).

A correspondence in a real image can be observed in the image in Figure 4.7(a) inpainted using [CPT04]. The inward growing textures adjoin close to the stone wall where step-like seams occur to the left and right of the bright spot in the centre of the target region. The amount of artefacts, in this case, reflects the increased complexity of the textures compared to the example above. Similar to the previous sampling problem, these artefacts can result in a snowball effect where they put negative influence on the fill-order which reflects in suboptimal selection of a search patch. Hence, encouraging similar artefacts.

4.3 Inpainting in the Wavelet Domain

An alternative approach [IJ07] trying to overcome some of the described limitations was introduced in section 3.4. Its conceptual similarity to [CPT04] and [CHL$^+$05] and the exploitation of spectral information as in [AM01] triggered the interest in this approach. It is, therefore, evaluated in this chapter to provide an insight into its qualities. Since it uses a priority-based fill-order that is closely related to [CPT04] and [CHL$^+$05] it will be compared to the results obtained in the previous chapter. The inpainting results used in this evaluation are provided by the software framework developed as part of this work and are solely based on the description provided in [IJ07].

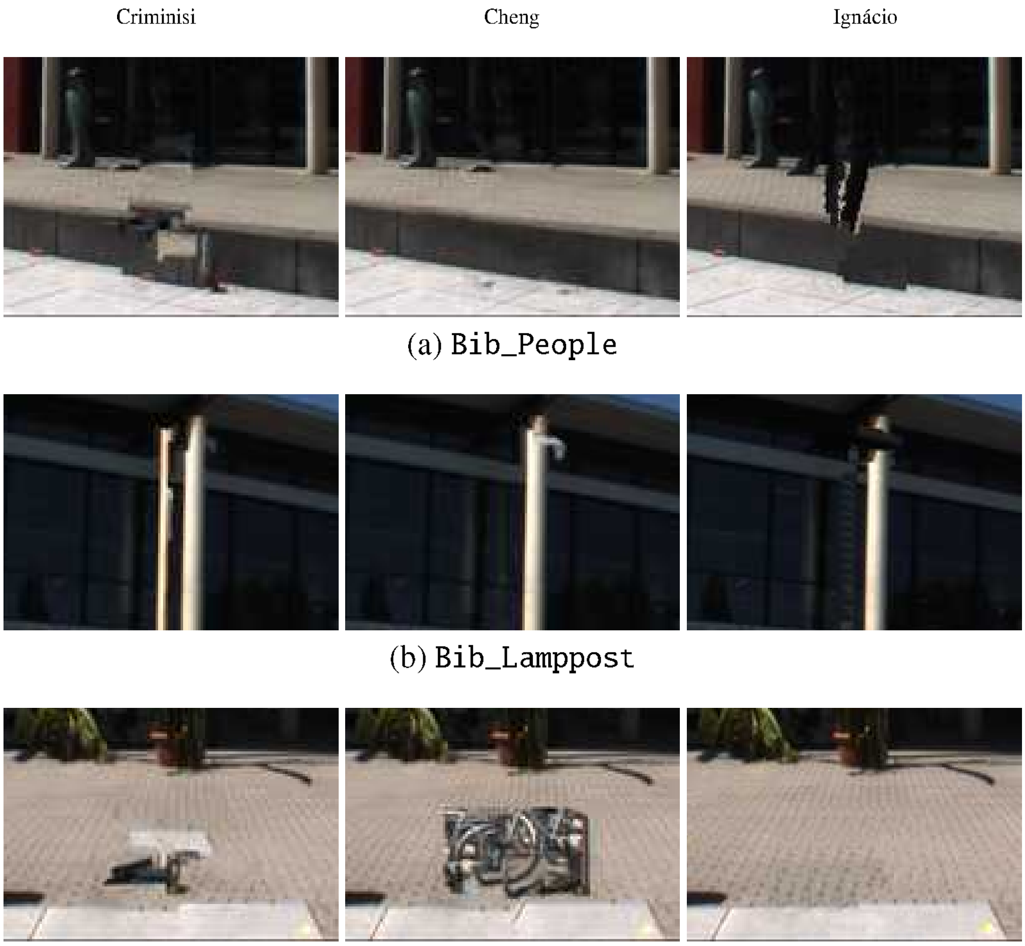
4.3.1 Investigating the Fill-Order

The fill-order described in section 3.4 calculates a priority from a confidence value and two structural components, namely structure orientation and significance. The confidence is similar to the value described in [CPT04] whereas the latter two components can be interpreted as the separation of the data term into two distinctive values denoting the orientation and significance of isophotes. Consequently, the priority value is balanced by confidence and structural information as in [CPT04]. The actual fill-order determined using wavelet coefficients, however, is very different. The corresponding inpainted images in Figure 4.10 show the influence of the new priorities on the results obtained by wavelet inpainting where the first two images in each row correspond to the results of [CPT04] and [CHL$^+$05] respectively. The last inpainted images are generated using the wavelet-based approach [IJ07].

As shown in the rightmost image in Figure 4.10(c), the new fill-order reconstructs the target region much better then the two previous approaches. The isophotes in the neighbouring source region are continued more accurately and the misplaced pixel intensities have disappeared. The lamppost image in Figure 4.10(b) is still a much better result then [CPT04] but introduces a repetitive pattern that does not occur in the corresponding result of the Cheng extension [CHL$^+$05] whereas the last image in Figure 4.10(a) is as bad as the one produced by [CPT04].

These results demand the examination of the modified priorities and the difference to the fill-order in the previous chapter. The plot in Figure 4.11 shows the development of the highest priority value during the inpainting process of the image in Figure 4.7(b) for [CPT04] and [IJ07] respectively. The curve generated by the Ignácio approach uses fewer iterations since it operates on the wavelet subbands simultaneously which reduces the processed image by a fourth in size. This, however, does not affect the considered curve behaviour. The curve in Figure 4.11(a) corresponds to the priorities computed using the Criminisi approach. The priority values are highly fluctuating for the first 100 iterations and are then starting to decay quickly. This complies with the observations described in [CHL$^+$05]. In contrast, the priorities calculated using wavelet coefficients are more consistent for the whole iterative process and do not decay as quickly. This results in a better comparability of the priority values over time and prevents the structure-related terms from becoming less relevant. [CHL$^+$05] stated that this increases the fill-order and in turn the quality of the obtained results.

This observation leads to the conclusion that determining the fill-order using wavelet coefficients in the corresponding priority equation (3.19) improves the result compared



(a) `Bib_People`

(b) `Bib_Lamppost`

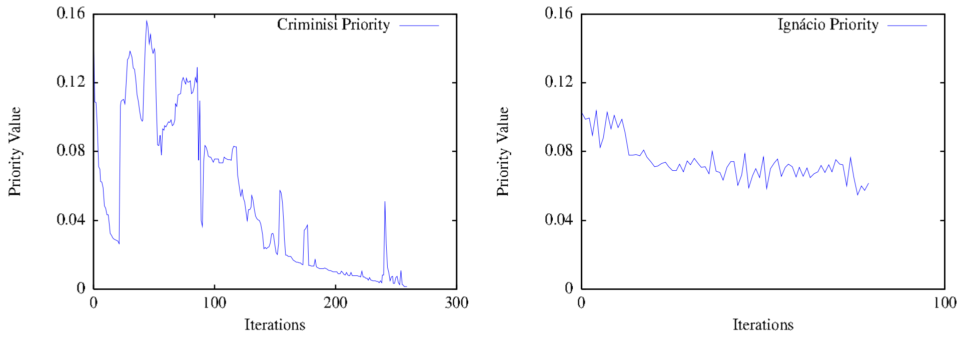
(c) `Bib_ColourChecker_BikeStand`

Figure 4.10: Excerpts from the inpainted images using Criminisi, Cheng and Ignácio approaches respectively.

to [CPT04] and stabilises the development of priority values over time. The actual result of the inpainting process, however, still depends on the correlation texture and structure represented in the input image and how closely it is matched by the modelled correlation in the priority equation. In an example such as Figure 4.10(a) where this condition is not satisfied the same weaknesses as in section 4.2 are implied.

4.3.2 Investigating the Sample Search

The behaviour of observable artefacts in the search for the best-matching patch are also very similar to those described in section 4.2. The patch-shaped artefacts as depicted



(a) Criminisi priorities

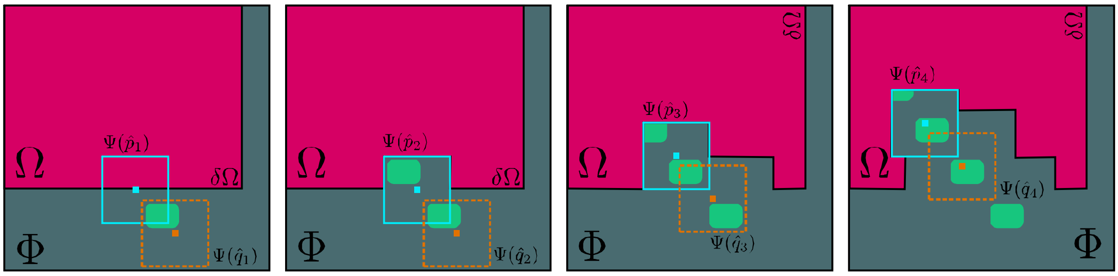
(b) Ignácio priorities

Figure 4.11: Plot (a) depicts the development of the highest priority in each iteration during a complete inpainting process using the Criminisi approach. (b) shows the corresponding priority development using the Ignácio method.

in Figure 4.9 are still a problem when seams between different texture regions occur. An inpainting example using [IJ07] is shown in Figure 4.10(a), where the isophote at the bottom of the wall is displaced. This is still a problem that depends mainly on the fill-order to prevent it.

Another weakness of [CPT04] and [CHL+05] was stated to be the search for sample patches in the whole source region. This leads to filled patches that are completely misplaced, as Figure 4.7(c) shows. [IJ07] reduces these misplacement artefacts by restricting the search region. Section 3.4 describes that sample patches are only considered in a small search window $\Theta(\hat{\boldsymbol{p}})$. It is centred at the corresponding patch centre $\hat{\boldsymbol{p}}$ of $\Psi(\hat{\boldsymbol{p}})$. This constraint is justified by the connectivity principle [PTN98] and assumes that the most suitable sample patch $\Psi(\hat{\boldsymbol{q}})$ can be found in the neighbourhood of $\Psi(\hat{\boldsymbol{p}})$. Figure 4.10(c) emphasises the correctness of the assumption. The reconstructed target region only contains pixel intensities related to the paving stone pattern as opposed to a malicious reconstruction produced by [CPT04] or [CHL+05]. Its quality has increased compared to these two approaches.

Unfortunately, this advantage comes at a price. The inpainted lamppost image in Figure 4.10(b), although a qualitative improvement to [CPT04], contains repetitive patterns in the target region which are not corresponding to any neighbouring texture. This is caused by the limited number of sample patches $\Psi(\boldsymbol{q})$ available in $\Theta(\hat{\boldsymbol{p}})$. A small set of samples obviously provides a higher risk of picking a less suitable sample. It is even increased by a search window $\Theta(\hat{\boldsymbol{p}})$ that contains a large amount of target pixels since this reduces the amount of samples even more. For instance, the synthetic image in Figure 4.12(a) shows a small textured area within an otherwise uniform texture. Assuming the depicted selection of $\Psi(\hat{\boldsymbol{p}})$ and $\Psi(\hat{\boldsymbol{q}})$ results in a repetition of the small



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

Figure 4.12: The images illustrate the cause of repeating texture during the search for the best-matching sample $\Psi(\hat{\boldsymbol{q}})$ and successive filling of $\Psi(\hat{\boldsymbol{p}})$. (a) shows a found $\Psi(\hat{\boldsymbol{q}})$ containing a small texture that is copied into $\Psi(\hat{\boldsymbol{p}})$ in (b). This happens again in (c) and (d), resulting in a repetitive pattern.

texture in Figure 4.12(b). The successive sampling steps, illustrated in Figure 4.12(c) and 4.12(d), then retrieve sample patches containing the same small texture. It is, in turn, repeated multiple times and propagated into the target region. The emerging pattern is easily recognised as an artificial pattern. Transferring this to real images results in similar repetitive patterns that are obstructive to a human recipient. Figure 4.10(b) is an example of such an artificial pattern. Its mainly vertical filling direction results in repeating the same small texture over and over. As an additive deficiency, it encourages the patch-shaped artefacts described in section 4.2.2. The overall artefacts and weaknesses introduced by finding the best-matching sample are, therefore, mainly similar to those in [CPT04] and [CHL$^+$05].

Chapter 5

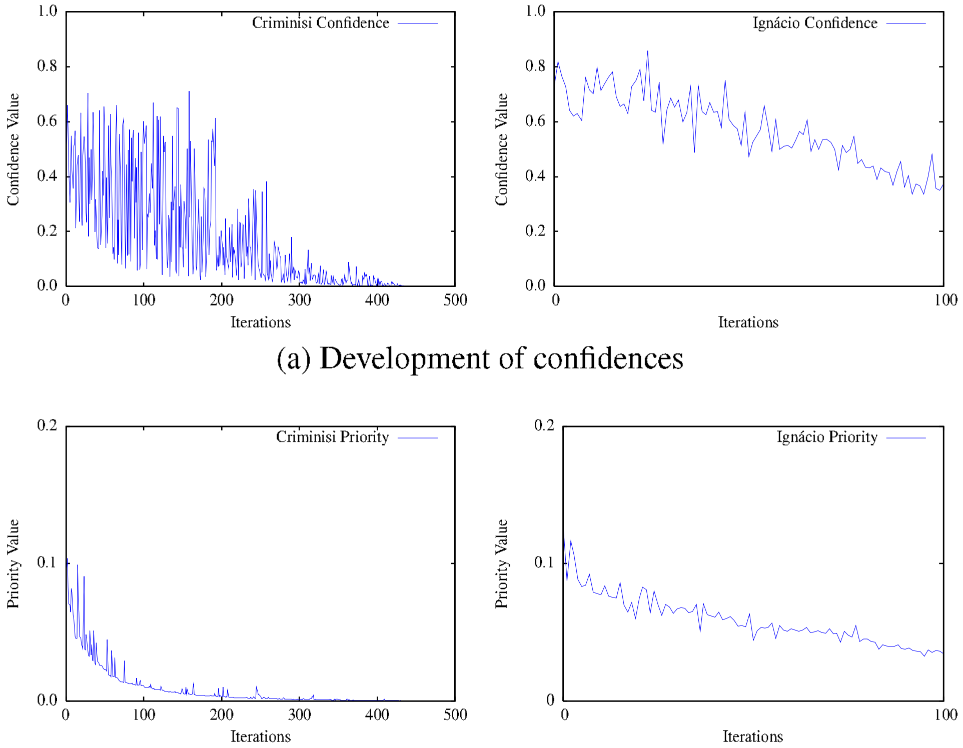
Inpainting Modifications and Results

The investigation of the approaches in the previous chapter shows that their results are very sufficient for some images and really poor for others. Unfortunately, the images taken on campus pertain to the second set of images where their target regions are generally inpainted insufficiently. Several causes have been identified limiting the quality of the approaches which makes it possible to adjust these limitations for a better applicability to the specific set of images that is used in this work. The Ignácio approach is the subject of these adjusting modifications since it is suspected to have the highest potential. A description as well as the reasoning for their implementation is the focus of section 5.1. It is followed by their evaluation in section 5.2 of the inpainting results that can be obtained with these modifications.

5.1 Wavelet-Based Inpainting Modifications

The description of the Ignácio approach in section 4.3 already discussed the modified fill-order in comparison to the Criminisi approach and the Cheng extension. Its resulting priority values behave very similar to the proposed modification in [CHL$^+$05] which aimed at a reduction of the quickly decaying priority values in [CPT04]. The priority equation (3.19) used in the Ignácio approach is defined as a combination of three components where the structure orientation and structure significance are separated from one another and counterbalanced by the confidence component. Since the equation uses wavelet coefficients instead of pixel intensities a difference in the resulting fill-order is obvious. It can be observed that the influence of the confidence component on the priority has increased in equation (3.19). This results in a fill-order that is more circular and less isophote-driven. The confidence plots in Figure 5.1(a) illustrate this change and show a more constant decay of the confidences in comparison to the Criminisi approach.

The increased influence of confidences maps this behaviour of the curve onto the corresponding priority curve as shown in the second plot in Figure 5.1(b). They rep-



(a) Development of confidences

(b) Development of priorities

Figure 5.1: The two plots in (a) show the development of the confidence value during the inpainting process using the Criminisi and Ignácio approach. The confidence value in each iteration contributes to the highest selected priority. The corresponding priority values and their development over the required amount of iterations are displayed in (b).

resent an inpainting procedure that fills high-reliability patches first responding less to structural influences along the boundary. This reduces the impact of noise and unrepresentative occurrences of either component on the priority. The gained influence of the confidence value however comprises a new deficiency of the fill-order. The inpainted image in Figure 5.2(a) shows that the confidence outweighs the structure components and mainly inpaints the textures. The isophotes incorporated in the structural components are disregarded which leads to displaced and disconnected structures and eventually poor inpainting results. The following modifications proposed an improved priority equation to reduce this effect.

5.1.1 Weighted Additive Priority

The superiority of the confidence is amplified by the multiplicative calculation of the priority in equation (3.19). Limiting this unproportional response is similar to the proposal in [CHL+05] where the multiplicative priority in [CPT04] is transformed into the additive equation (3.10). This makes the equation less prone to extreme values in either component. The same strategy can be used to reduce the impact of the confidence on the priority in equation (3.19). The modified priority equation obtains priority values by adding the respective components. It is formally defined as

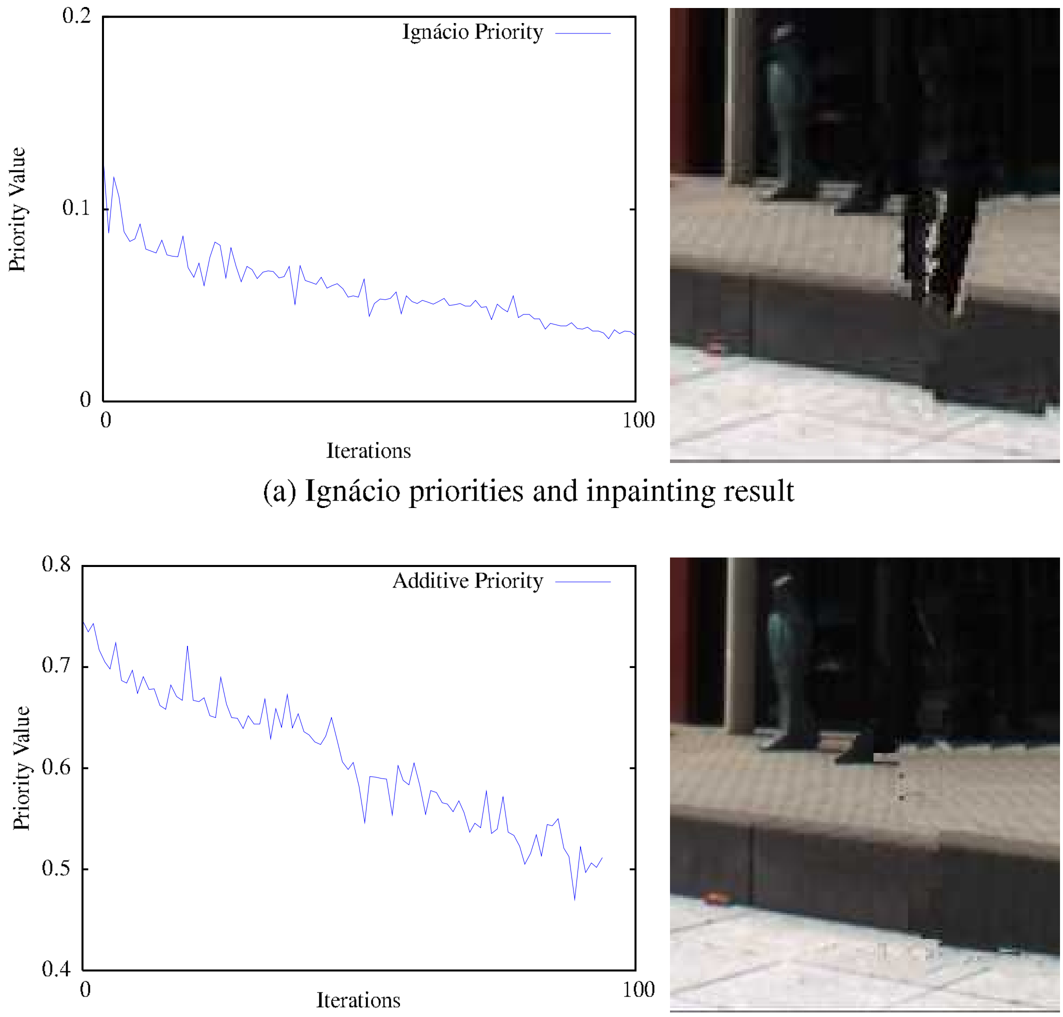
$$P(\boldsymbol{p}) = K(\boldsymbol{p}) \ + \ S_S(\boldsymbol{p}) \ + \ S_O(\boldsymbol{p}) \tag{5.1}$$

The inpainting result responds immediately to these changes. Figure 5.2 substantiates that inpainting the same image using the weighted additive priority increases the continuation of isophotes and reduces the amount of artefacts. The corresponding priority plots show that the highest priorities obtained by equation (5.1) span a larger interval of values using the additive equation instead of the equation proposed in [IJ07]. The fluctuation of the curve remains minimal. The slope of the curve however is steeper produced by more distinguishable values. It is the direct result of adding the components that produce the increased range of priority values which corresponds to a better balance of confidence and structure components. The influence of either component is equalised by computing normalised values as described in section 3.4.

A remaining shortcoming of the additive priority equation concerns the diversity of image contents and the corresponding correlation between its texture and structure. [CHL+05] described that the quality of the fill-order depends on the correct combination of texture and structure in the input image. Since the correlation of these two elements is hard-wired into equation (3.19) a large amount of images with unsuitable characteristics cannot be inpainted correctly. The Cheng extension addresses this by applying another change to their priority equation (3.12) introducing two parameters to weight the structure and texture correlation. This concept is once again adapted by modifying equation (5.1). Two parameters ω_S and ω_C are implemented for adjusting the proportional dominance of the components. ω_S is the *structure weight* and controls the relative importance of structure orientation and significance resulting in a weighted structure value $S(\boldsymbol{p})$ defined as

$$S(\boldsymbol{p}) = (1 - \omega_S)\, S_O(\boldsymbol{p}) \ + \ \omega_S\, S_S(\boldsymbol{p}) \tag{5.2}$$

with $0 \leq \omega_S \leq 1$. Setting the structure weight to $\omega_S = 0.5$ levels both components corresponding to the normalised sum of both values. Increasing ω_S stresses the structure significance S_S which resembles the strength of isophotes in a patch $\Psi(\boldsymbol{p})$. $S(\boldsymbol{p})$ is therefore dominated by the strength of these structures with less influence of their orientation relative to the boundary of the target region. On the contrary, reducing ω_S boosts



(a) Ignácio priorities and inpainting result

(b) Additive priorities and inpainting result

Figure 5.2: The plots show the development of the highest priority value during the inpainting process and the obtained inpainting results. The curve produced by the Ignácio approach, shown in (a), consists of less distinguishable priorities than (b), displaying the additive priorities described in equation (5.1).

the orientation-related characteristic of isophotes disregarding their possible difference in significance. Adjusting this weight provides enhanced control over the influence of either structural property.

Assuming an adjusted ω_S with the obtained $S(\boldsymbol{p})$ is then weighted against the confidence value. The second weight parameter ω_C is therefore similar to the proposed modification in [CHL$^+$05] where two parameters are used to balance their confidence and data term in equation (3.12). Formalising this returns the final additive priority equation

$$P_A(\boldsymbol{p}) = (1 - \omega_C)\, K(\boldsymbol{p}) \; + \; \omega_C\, S(\boldsymbol{p}) \tag{5.3}$$

where $S(\boldsymbol{p})$ is the weighted structure value in equation (5.2) and ω_C holds $0 \leq \omega_C \leq 1$. Obtaining a normalised and unweighted priority the confidence weight can be set to $\omega_C = 0.5$. In case the settings hold $\omega_S = \omega_C = 0.5$, the calculated priority corresponds to an unweighted additive priority that has been normalised in two separate steps. Altering the fill-order by using a higher ω_C results in a dominance of the structure component where inpainting is mainly isophote-driven. Accordingly, setting ω_C to a value lower then 0.5 exaggerates the propagation of texture into the target region with less influence of structure properties. This approximates a similar behaviour of the priority curve in the top plot in Figure 5.2.

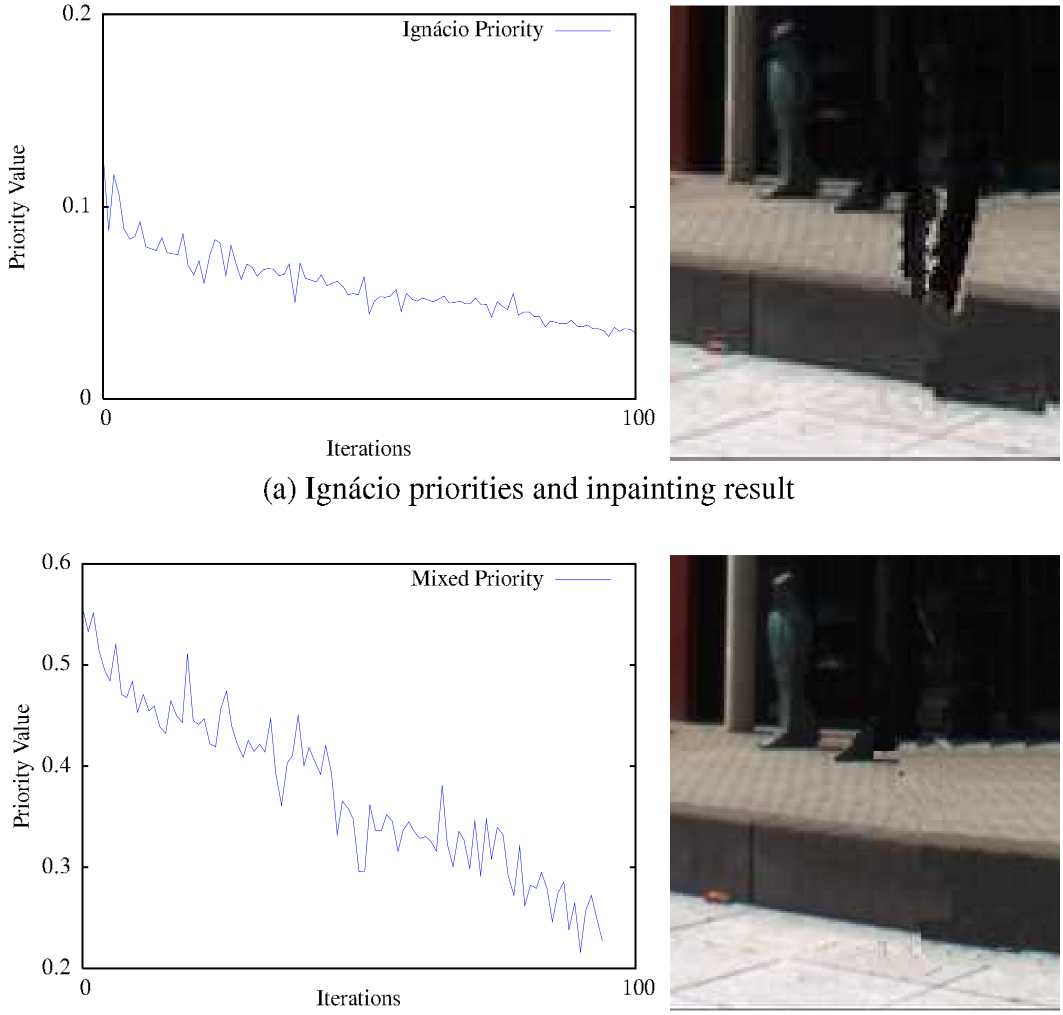
5.1.2 Weighted Mixed Priority

Experimenting with the additive priority equation above inspired a second concept to modify the priority. It aims at achieving an equivalent solution with only a single parameter ω_C. It recombines the structure components for orientation and significance similar to the data term used in [CPT04]. The data term incorporates orientation and significance of an isophote in a single value, whereas structure orientation and significance separate this information into two distinguished components, as described in section 4.3.1. The structure orientation is therefore normalised to only represent the direction of an isophote in respect to the boundary. Reverting this separation is very simple and just a multiplication away. This weights the direction of an isophote with its respective strength and makes the previous structure weight ω_S obsolete. The priority equation, referred to as *mixed priority* in the following, is formally defined as

$$P_M(\boldsymbol{p}) = (1 - \omega_C)\, K(\boldsymbol{p}) \; + \; \omega_C\; (\, S_O(\boldsymbol{p}) \cdot S_S(\boldsymbol{p}) \,) \tag{5.4}$$

with $0 \leq \omega_C \leq 1$. The plots and inpainted images in Figure 5.3 substantiate that its results are similar to those produced using the additive priority.

The exemplar result shows that the most important isophotes are continued correctly with only a few exposed artefacts. The priority plot reveals that the curve behaviour using a mixed priority implies a similar behavioural change to the curve. Priority values in successive iterations are more distinguishable than in the Ignácio approach, although remaining less fluctuating. The added advantage towards the additive priority equation (5.3) is the reduction of the adjustable parameters with a remaining quality of the obtained results. This increases the usability of the inpainting procedure in opposition to the previous modifications and maintains the possibility to adjust the fill-order to specific image characteristics.



(a) Ignácio priorities and inpainting result

(b) Additive priorities and inpainting result

Figure 5.3: The plots illustrate the development of the highest priority value during the inpainting process using the Ignácio approach in (a) and the mixed priority method in (b). The curves illustrate that equation (5.4) produces more distinctive values resulting in better inpainting results as the corresponding inpainted images show.

5.1.3 Blended Filling of Patches

The implementation of an adjustable fill-order improved the inpainting results but leaves three visible types of artefacts exposed in the inpainted images. The first two are displaced isophotes and patch-shaped artefacts caused by adjoining textures. The two examples in figures 5.2 and 5.3 corresponding to the respective fill-order modifications show this type of artefacts in the texture of the dark stone wall. The bottom isophote confining the texture is displaced and differently shaded textures are producing rectangular artefacts at their seam. The third type is the repetitive pattern in the texture in the background. Section 4.3 described the rationale for all of them. To reduce the impact and most importantly the visibility of these artefacts an additional processing step has to be introduced. Manual inpainting tools in image processing applications, such as the *healing* or *clone brush*, use a blending mask for the brush to provide a smooth transition between the added intensities and those in the background. The further away a pixel location is from the centre of the brush the more impact is induced by the background intensity on the new value. This concept is adapted by using a blending method when filling the search patch $\Psi(\hat{\boldsymbol{p}})$.

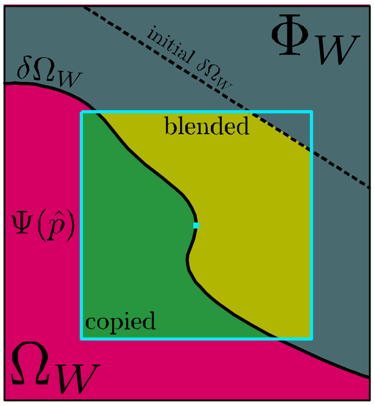


Figure 5.4: The image shows a search patch $\Psi(\hat{\boldsymbol{p}})$ centred on the propagated boundary $\delta\Omega$. The green area denoted *copied* corresponds to unknown coefficients that are filled by copying coefficients from $\Psi(\hat{\boldsymbol{q}})$. The yellow area, denoted blended, corresponds to alterable coefficients in the filled source region. The small piece of the patch pointing into the initial source region is ignored in the blending method.

Filling the unknown pixel locations in $\Psi(\hat{\boldsymbol{p}})$ using sample coefficients from $\Psi(\hat{\boldsymbol{q}})$ is described in section 3.4. It is defined that only unknown pixel locations are altered and all source pixels remain unchanged. The coefficients in $\Psi(\hat{\boldsymbol{q}})$ corresponding to these unknown locations are simply copied to fill $\Psi(\hat{\boldsymbol{p}})$. The filling of these pixel locations remains unchanged to ensure the correct filling of the target region. This leaves the known source pixel coefficients to be used in the blending process. The limitation of these coefficients remaining unaltered, therefore, has to be softened. Only pixel locations contained in the initial source region are considered unchangeable. Figure 5.4 illustrates this softened restriction and shows the parts of $\Psi(\hat{\boldsymbol{p}})$ that are *copied* or *blended*. Each pixel location in the source region is then considered in the blending process in which its coefficient is weighted against its corresponding coefficient in $\Psi(\hat{\boldsymbol{q}})$.

1	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{2}$	1
1	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{2}$	1
1	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{2}$	1
1	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{2}$	1
1	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{2}$	1

(a) Horizontal distances

1	1	1	1	1
$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
1	1	1	1	1

(b) Vertical distances

Figure 5.5: Image (a) illustrates the horizontal distances of pixel locations in respect to the patch centre for a patch of size 5 × 5. Image (b) depicts the corresponding distances in vertical direction.

It is assumed that a search patch $\Psi(\hat{\boldsymbol{p}})$ and its best-matching sample $\Psi(\hat{\boldsymbol{q}})$ have been retrieved by the sample search. Let $\boldsymbol{r}_i$ be a pixel location in $\Psi(\hat{\boldsymbol{p}})$ holding $\boldsymbol{r}_i \in \Phi_W \cap \Psi(\hat{\boldsymbol{p}})$ and its corresponding location $\boldsymbol{s}_i \in \Psi(\hat{\boldsymbol{q}})$. The new coefficient at location $\boldsymbol{r}_i$ is calculated by weighting the current coefficient at $\boldsymbol{r}_i$ with the one corresponding to $\boldsymbol{s}_i$. To ensure a smooth transition of coefficients in the patch their composition is weighted depending on the location of $\boldsymbol{s}_i$ in respect to its patch centre $\boldsymbol{s}$. The weight is formally defined as

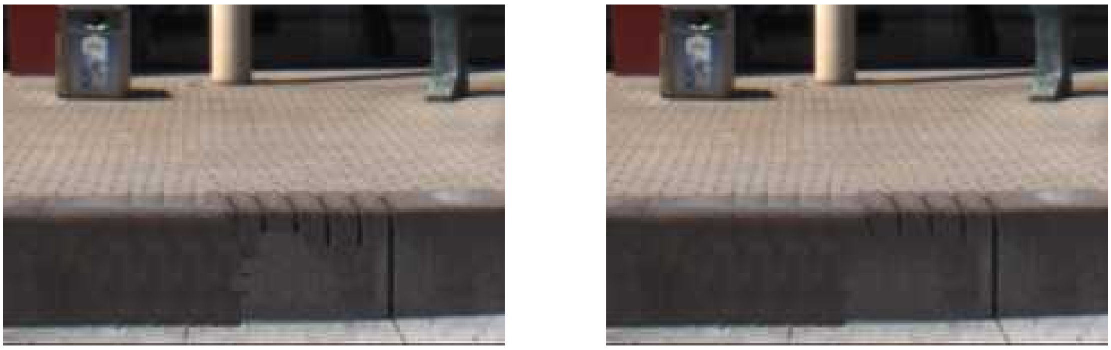
$$\omega_G = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{d_x^2 + d_y^2}{2\sigma^2}\right) \tag{5.5}$$

where d_x and d_y denote the distance of $\boldsymbol{s}_i$ to its patch centre in horizontal and vertical direction respectively. The distance is defined as a fraction of the patch size M with $\frac{2}{M+1}$ at its centre and 1 at the outer locations of the patch. Figure 5.5 illustrates the distances in both directions for an exemplar patch of size $M = 5$. The weight ω_G is then determined by a Gaussian distribution applied to these distances and is used to weight the pairs of coefficients against each other. According to the separate filling of patches in each subband, described in section 3.4 the weighting of coefficients is carried out in each subband. The blended filling for the approximation subband is therefore defined as

$$W_a(\hat{\boldsymbol{r}}_i) = (1 - \omega_G) \cdot W_a(\boldsymbol{r}_i) \; + \; \omega_G \cdot W_a(\boldsymbol{s}_i) \tag{5.6}$$

and stands representative for the three detail subbands which are calculated accordingly. The resulting coefficient maintains its energy by reducing the coefficient at $\boldsymbol{r}_i$ by the same amount the coefficient at $\boldsymbol{s}_i$ is contributing to it. This requires ω_G to hold $0 \leq$

$\omega_G \leq 1$. In the case that ω_G exceeds 1 due to the selected σ the original coefficient at $\boldsymbol{r}_i$ is maintained.



(a) No Blending

(b) Blending with $\sigma = 0.4$

Figure 5.6: Two excerpts of the same area in the inpainting result of the image `Bib_ColourChecker_Entrance`. The entire image is shown in Figure C.15. (a) shows the inpainted region without blending applied in the filling step whereas (b) uses blending. σ denotes the standard deviation used to control the slope of the Gaussian curve.

The integration of the blending method into the inpainting procedure changes the obtained result, as shown in Figure 5.6. The amount of blending can be controlled by adjusting the parameter σ in equation (5.6). It controls the decay of the Gaussian distribution that is applied to the distances in the patch described above and, therefore, affects ω_G in the same way. Using a lower value for σ results in a higher contribution of the coefficient at a location $\boldsymbol{s}_i$ close to the patch centre, decaying for outer locations. Increasing σ on the other hand determines a lower contribution of the sample coefficients close to the centre, decaying quickly towards its outer locations.

Applying the blending modification is not only reducing path-shaped artefacts it also has an influence on the fill-order. Blending source pixel coefficients changes the input to equations (5.3) or (5.4) defining a different fill-order in comparison to a non-blended inpainting procedure using the same priority equation.

5.2 Evaluation of the Modifications

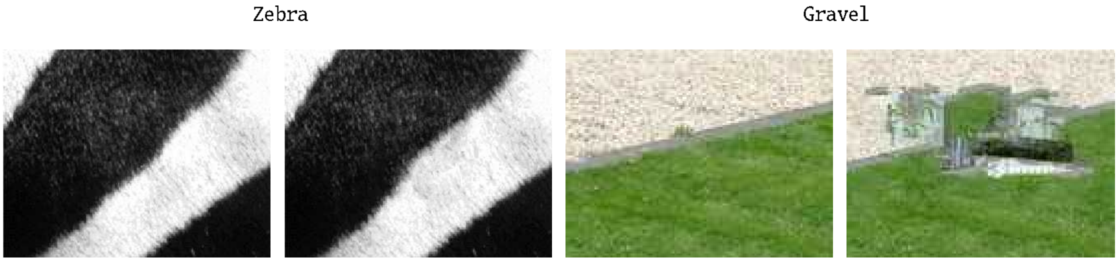
The modifications proposed in the previous section were developed to improve the inpainting quality for highly structured images based on the Ignácio approach. Although they are based on the discovered shortcomings of the approaches compared in Chapter 4 and try to eliminate the causes, their results have not been analysed yet. The following sections, therefore, evaluate the results obtained using the modified fill-orders as well as the blending method. The two fill-order modifications will be addressed as *additive*, described in section 5.1.1, and *mixed*, described in section 5.1.2. Before these results are investigated individually and compared to the Criminisi, Cheng and Ignácio approach, a short evaluation of the dynamic patch size and the dependant limitation of the search window $\Theta(\hat{p})$ is presented. It provides the justification for assumed settings in the evaluation. After these individual investigations, the modifications are compared to one another to find the most suitable combination of modifications and their respective parameters.

The evaluation uses a representative selection of campus images with corresponding binary masks which are included in Appendix B. These images contain no objects that are removed. Their binary masks specify certain target regions that are to be reconstructed and allow the comparability of the inpainted images with their respective originals, the so called *ground truths*. The *peak-signal-noise-ratio* (PSNR) is used to measure their differences including a few constraints described in the next section. Additionally, a subjective qualification of the inpainting quality is used to define the *perceptual plausibility* of the results. The characterising properties of the results are illustrated using details of the images in the test set to ensure that they are conspicuous. The most sufficient inpainted images for each modifications are depicted in Appendix C.

5.2.1 Inpainting Quality Measure

To compare the inpainting results a measure to determine their quality is required. PSNR is one existing measure to define the difference between images and is widely used in the field of image processing. In research concerning lossy image and video compression, for instance, it is important to define the quality of the decompressed result. This is achieved using PSNR which computes the pixel-wise difference of two images using their *mean squared error* (MSE) and represents it on a logarithmic scale. It is usually expressed in *decibel* (dB) and values between 30 and 50 dB are considered sufficient in this specific field of research. The preferred results in applications using PSNR is that there is no difference between the two images at all, corresponding to an infinite PSNR and an MSE equal to zero.

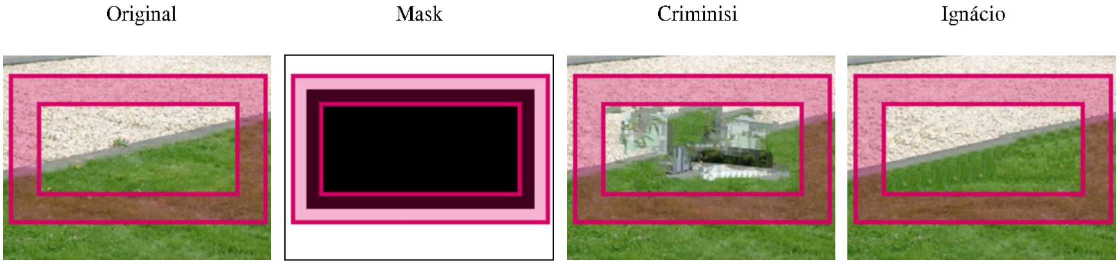
Unfortunately, it is not a well-suited quality measure for image inpainting. The main objective of image inpainting is to fill in the target region with intensities that create an authentic impression. Hence, it is not required to achieve the most accurate or even identical reconstruction of the original, which is not even possible when an object is actually removed from it. Finding a plausible inpainting result is satisfactory. This contradicts with the concept of PSNR where the highest value defines the most accurate result. This can lead to a deceptive PSNR in images that are authentically inpainted but their texture intensities are displaced by a few pixel locations in respect to the original. This can be illustrated on the inpainted image in Figure 5.7(b) which is not identical to its original in Figure 5.7(a) but can be considered a sufficient result.



(a) Original (b) Inpainted: PSNR: 23.77 dB (c) Original (d) Inpainted: PSNR: 33.88 dB

Figure 5.7: The images show inpainting results using the Criminisi approach [CPT04]. The images are compared to their respective originals using the PSNR metric calculating the pixel-wise difference.

Computing its PSNR, however, defines it as an insufficient result with a value below 30 dB. The image Figure 5.7(d) substantiates an additional problem using the PSNR measure. The inpainted target region is a complete mess. Intensities from all over the image are inserted and render it a very bad result. Considering its PSNR, however, claims otherwise by providing an even higher value than for image in Figure 5.7(b). This shows that using PSNR inconsiderately results in an usable evaluation.



(a) (b) (c) PSNR: 41.68 dB (d) PSNR: 40.55 dB

Figure 5.8: The images show details from the original image `Gravel` and the corresponding binary mask in (a) and (b) respectively. The region enclosed by the pink lines and semi-transparent background denotes the *comparable region* used to calculate the modified PSNR values. Images (c) and (d) depict the pixel locations covered by the comparable region and illustrate the problem with the modified measure.

An attempted solution to define an objective quality measure that can be used in the evaluation of inpainting results restricted the PSNR to a *comparable region*. It is a set of corresponding pixel locations in the original and inpainted image, as shown in Figure 5.8. It encloses an equivalent amount of ten pixels to each side of the boundary between the initial source and target region. The image in Figure 5.8(b) shows the binary mask with the overlaid comparable region. Only pixel locations enclosed in this region are then considered in calculating the PSNR. The main idea was to measure the correct continuation of isophotes into the target region. Their progression within the target region, however, is ignored, preventing the reduction of the PSNR if the continuation of an isophote is not pixel accurate. The aim was to provide a PSNR-based quality measure that produces values with a higher approximation to the perceived quality.

The comparison of the original PSNR with the comparable region revealed that the latter provides a more meaningful PSNR for some images. However, the problem described above remains, as shown Figure 5.8(c). Its insufficiently inpainted target region is not represented in the comparable region. The result is a higher PSNR compared to the image in Figure 5.8(d) depicting a more authentic result. Additionally, the data in Table 5.1 contrasts the PSNR for these images exhibiting the same tendencies for both measures. Using a selected amount of pixel locations to compare their corresponding intensities is, therefore, no improvement over the regular PSNR.

	PSNR (dB)				
Pixels considered in	Criminisi	Cheng	Ignácio	Additive	Mixed
Whole image	35.25	33.70	33.88	30.66	33.09
Comparable region	41.68	32.55	40.55	40.61	40.66

Table 5.1: Data generated by computing the PSNR between the respective inpainted image and it original image. The first row computes the PSNR using all pixels in the image whereas the second row uses a defined test region to measure the difference.

This constitutes that PSNR is not suitable to measure and evaluate the quality of inpainting results. Even the attempt to adapt it to this specific objective is not providing any improved outcomes. In absence of any other established measure and the use of PSNR in inpainting-related literature, the original PSNR is considered in the following evaluation. The obtained values are consulted to illustrated trends in choosing different parameters without being used to define the actual quality. This is carried out by a visual examination of the inpainting results which remains a subjective measure of quality of quality of quality of quality.

The PSNR data in the following comparison uses a colour-coding scheme to indicate the highest and lowest PSNR obtained for each test image. The lowest value is coloured in pink, whereas green denotes the highest value. If the same highest or lowest value occurs more than once both values are coloured accordingly.

5.2.2 Search Window and Dynamic Patch Size

Studying the Ignácio approach thoroughly posed a question concerning their proposal of a dynamic patch size and the dependant search window $\Theta(\hat{\boldsymbol{p}})$. It was suggested to search for a sample patch using various sizes to improve the quality of the best-matching $\Psi(\hat{\boldsymbol{q}})$. The adaptation of its size to potentially different structures and textures in the neighbourhood of $\Psi(\hat{\boldsymbol{p}})$ provided the justification for it. A result of the dynamic patch size and the search window $\Theta(\hat{\boldsymbol{p}})$ results in a noticeable increase of computational expenses. Investigating the qualitative improvement they induce was, therefore, self-evident.

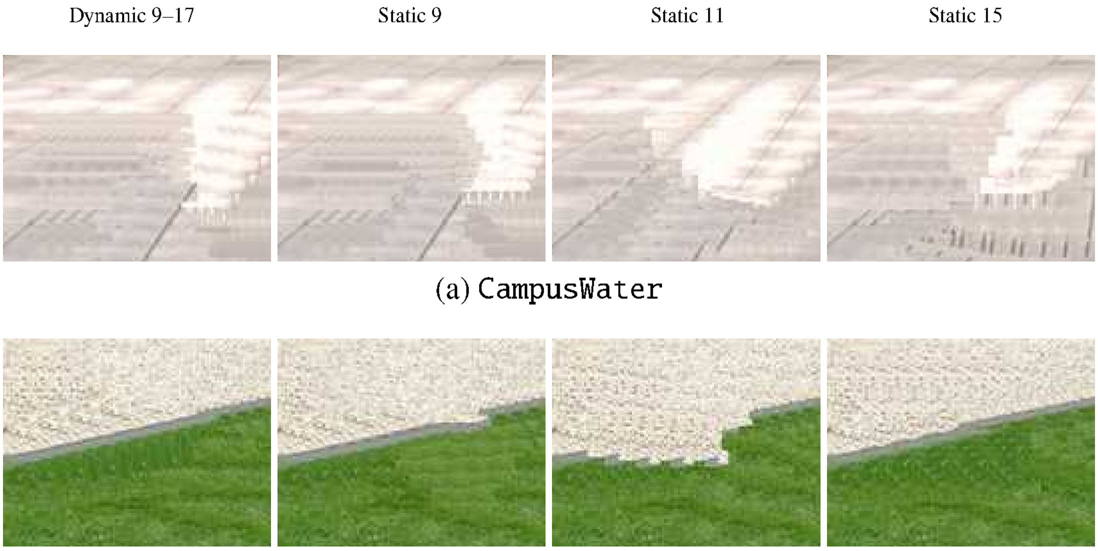
Dynamic Patch Size

The first assumption was that using a single patch size, in the following referred to as *static*, provides similar results to those obtained by a dynamic patch size. To confirm this assumption, the latter is compared to a set of static patch sizes according to the settings defined in [IJ07]. That is, a range of $M_{min} = 9$ to $M_{max} = 17$ is used with a search window $\theta = 3M$. The patch size is increased by to ensure odd-sized patches with a valid patch centre $\boldsymbol{p}$. Static and dynamic patch sizes are tested using the originally proposed algorithm in [IJ07].

	PSNR (dB)					
	Dynamic size	Static size				
Image	9–17	9	11	13	15	17
`G_Entrance_Pillar`	48.81	48.93	49.04	49.07	48.78	47.03
`G_Entrance_Top`	48.00	46.71	49.18	47.51	47.30	46.62
`G_Entrance_2sq`	43.65	44.10	42.10	39.28	39.43	39.92
`MikadoPlatz_TopLeft`	48.10	49.19	50.41	48.91	49.36	48.48
`LibFront_HorizLong`	39.33	39.91	39.25	37.35	36.04	35.59
`LibFront_RightBottom`	40.82	41.18	40.85	41.70	42.07	40.21
`Gravel`	33.88	33.95	29.99	33.05	33.67	32.65
`Menseria_LeftBottom`	42.63	42.30	42.66	42.66	42.71	42.17
`Menseria_Top`	33.65	35.80	35.29	34.81	32.47	32.90
`Obelisk_Centre`	33.82	33.60	33.75	33.66	31.12	31.47
`CampusWater`	35.14	35.17	33.85	34.30	35.20	36.26

Table 5.2: PSNR values retrieved by varying the fixed patch size. The dynamic patch size denotes the reference data corresponding to the Ignácio approach. The inpainting process is carried out using the original priority equation (3.19) and a search window size $\theta = 3M$.

Computing the PSNR for the results obtained using the different setting for the patch size provided the data shown in Table 5.3. The distribution of the highest PSNR for each image suggests an improved inpainting quality for static patch sizes which tends towards lower sizes instead of higher ones. Static patch sizes $M = 9$ and $M = 11$ are leading the table with 55% of the highest PSNR values. In addition, the data shows that the smallest patch size $M = 9$ produces a predominantly higher PSNR than the dynamic patch size for the test set. Larger static patch sizes, however, suggest a regression of the quality with mostly lowest PSNR values the patch size $M = 17$. This provides that the previous assumption is correct and static patch sizes produce equivalent or better results than their dynamic counterpart. In addition, smaller patches are preferable to larger ones.



(a) CampusWater

(b) Gravel

Figure 5.9: Exemplar excerpts from inpainting results using the patch size settings specified above each column. The inpainting process used the original fill-order calculation from the Ignácio approach with a search window size $\theta = 3M$.

Based on the description in section 5.2.1, the results have to be visually examined to substantiate these suggestions. The detail images in Figure 5.9 show exemplar results corresponding to the data in Table 5.2. The inpainted regions provide different degrees of sufficiency depending on the selected patch size. Although none of the results are sufficient and they all suffer from repetitive texture, they expose the expected properties. The target regions in Figure 5.9(a) for the dynamic patch size and static patches with $M = 9$ and $M = 15$ are poorly inpainted with repeating texture disrupting the strong isophotes traversing the image from back to front. The best results are obtained by the patch size $M = 11$ complying with the previous suggestion that static patch sizes are sufficient. The images in Figure 5.9(b) emphasis this as well. Admittedly, the results for $M = 9$ and $M = 11$ are less sufficient with disrupted isophotes and gravel wrongly inserted into the grass texture. The last image, however, shows that a similar inpainting result is obtainable using the static patch size $M = 15$. The isophotes are continued correctly and repeating texture is equivalent to result for the dynamic patch size.

Using a static patch size is, therefore, considered the sufficient choice for inpainting. The suggested superiority, assured in [IJ07], could not be reproduced and small static patch sizes provide similar results with less computational expense. In some cases, as shown in Figure 5.9(a), even better results can be obtained. Hence, the following sections assume a static patch size, generally set to $M = 9$, unless stated otherwise.

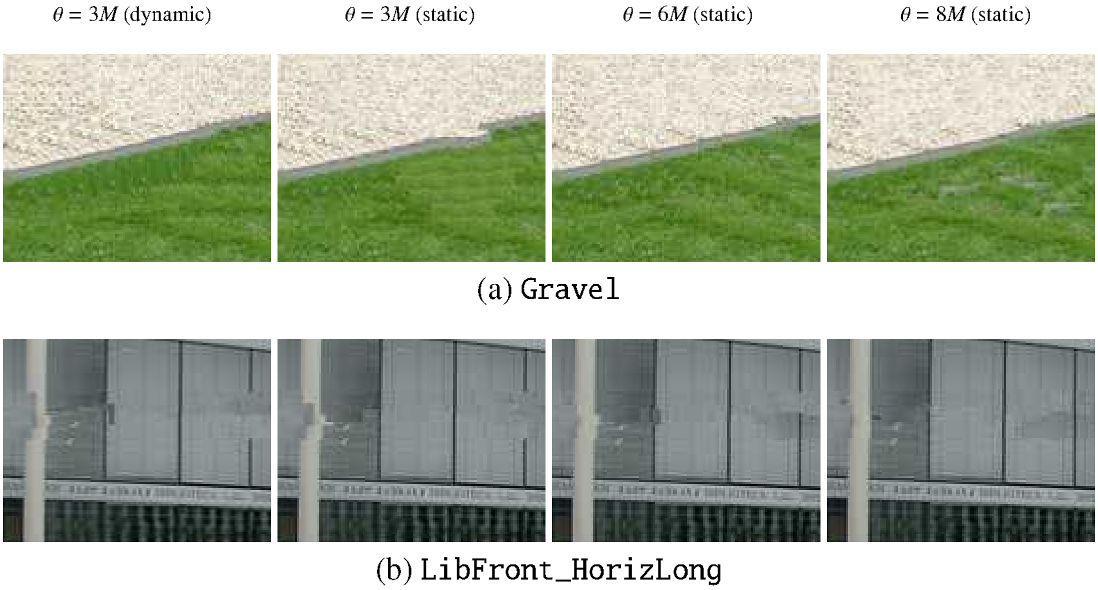
Search Window Size

It has already been stated that the target regions depicted in Figure 5.9(a) contain a large amount of repeating textures, a problem that was discussed in section 4.3.2. The cause for such disrupting patterns is suggested to be the limitation of the search region $\Theta(\hat{\boldsymbol{p}})$ to a very small size of $\theta = 3M$. The reason for introducing $\Theta(\hat{\boldsymbol{p}})$ was to reduce the high computation times accompanying the dynamic patch size. Since this is no longer a concern, the question is raised whether to eliminate this constraint. The investigation in section 4.3.2, however, described that it also prevents some artefacts inherent in the Criminisi and Cheng approaches. A compromise is to increase the search window size to reduce the repetitive patterns but refrain from abandoning it completely.

	PSNR (dB)				
	θ (Dynamic 9–17)	θ (Static 9)			
Image	$3M$	$3M$	$4M$	$6M$	$8M$
`G_Entrance_Pillar`	48.81	48.93	49.22	48.81	47.77
`G_Entrance_Top`	48.00	46.71	46.88	47.41	47.29
`G_Entrance_2sq`	43.65	44.07	41.06	42.15	42.11
`MikadoPlatz_TopLeft`	48.10	49.11	49.56	49.67	49.85
`LibFront_HorizLong`	39.33	39.91	40.47	40.59	41.45
`LibFront_RightBottom`	40.82	41.18	41.53	42.94	41.26
`Gravel`	33.88	33.95	33.85	34.49	33.82
`Menseria_LeftBottom`	42.63	42.30	42.76	42.75	42.97
`Menseria_Top`	33.65	35.80	35.68	35.62	35.11
`Obelisk_Centre`	33.82	33.60	33.73	33.74	33.82
`CampusWater`	35.14	35.17	35.71	35.53	34.99

Table 5.3: PSNR values computed from inpainting results using a varying search window size θ. Inpainting is carried out using the original fill-order from the Ignácio approach and static patch size $M = 9$. The dynamic patch size denotes the reference data set.

The impact of this change is investigated using the original settings of the Ignácio approach with a dynamic patch size and the smallest static patch size $M = 9$. The retrieved PSNR data is shown in Table 5.3. Although, it is not offering a direct insight into the increase or decrease of repeating textures, the values show a tendency for higher PSNR values towards a larger $\Theta(\hat{\boldsymbol{p}})$. More specific, the search window sizes $\theta = 6M$ and $\theta = 8M$ produce the highest PSNR for 64% of the inpainting results. The remaining 36% are almost equally distributed between the other settings. This indicates that the suggested increase of the search window size leads to better inpainting results. The reduction of repeating texture could be derived from that assuming that pixel-wise differences increase with repetitive patterns where the original image contains diffuse textures. The PSNR, however, does not provide the sufficient measure to substantiate more then just a assumable trend.



(a) Gravel

(b) LibFront_HorizLong

Figure 5.10: Excerpts from inpainting results computed with a varying search window size. The size used for the corresponding images is specified above each column. The inpainting results are obtained using [IJ07] with a patch size of 9 pixels where a static size is used.

The confirmation for this suggestion is, therefore, to be found in visually investigating the images in Figure 5.10. Comparing the images in Figure 5.10(a), in particular, expose the expected influence of $\Theta(\hat{\boldsymbol{p}})$ on the inpainted image. The excerpt of the image computed with the dynamic patch size, although it contains sufficiently continued isophotes, is dominated by repetitive grass texture close to the stone texture in its centre. Using the same setting for $\Theta(\hat{\boldsymbol{p}})$ with a static patch size $M = 9$ in the second image illustrates that reducing the patch size inpaints the target region creating an even less acceptable result. Choosing larger sizes for $\Theta(\hat{\boldsymbol{p}})$, as shown in the two successive images, shows sufficient isophotes combined with reduces repeating textures. This validates the tendency observed in Table 5.3. Choosing $\theta = 6$ or $\theta = 8$ provides a larger set of sample patches which makes it more likely to find an optimal best-matching patch $\Psi(\hat{\boldsymbol{q}})$. The similar can be observed looking at the results in Figure 5.10(b), where the first image depicts an unacceptable result with disrupted isophotes in the shutter texture and the pillar to the left. Selecting a larger $\Theta(\hat{\boldsymbol{p}})$ shows an improved image with each increment, eventually producing an improved inpainted image at $\theta = 8M$. Assuming, however, that larger and larger sizes produce better and better results was already stated to be misleading, as described in section 4.2.2. A closer look at the result in Figure 5.10(a) using $\theta = 8M$ reveals that small pieces of stone texture are placed in the grassy part of the target region.

These examples confirm the PSNR-based assumption that using an increased $\Theta(\hat{\boldsymbol{p}})$ used with static patch sizes improves the results compared to the Ignácio approach. Furthermore, it is demonstrated that abandoning $\Theta(\hat{\boldsymbol{p}})$ is not a solution since wrongly sampled textures start to occur when $\Theta(\hat{\boldsymbol{p}})$ becomes too large implying that a intermediate solution is the most sufficient. Evaluating the test set and their representative examples in Figure 5.10 lead to the conclusion that the size $\theta = 6M$ satisfies the desired quality and is, therefore, used in all further evaluations unless it is stated otherwise.

5.2.3 Weighted Additive Priority

The major realisation throughout chapters 3 and 4 was that the fill-order and, therefore, the calculation of the priority value for each search patch $\Psi(\boldsymbol{p})$ is the most important element of all three algorithms. It has the largest influence on the inpainting result. Improving these results can only be achieved by improving the quality of the fill-order. After investigating the shortcoming of the the approaches in Chapter 4, two modifications were suggested in sections 5.1.1 and 5.1.2. The first modification considers an additive priority calculation allowing to weigh the three components confidence, structure orientation and structure significance against each other using equation (5.3). This section evaluates the test results retrieved using this new calculation and shows how the weighing effects the inpainting process. The structure weight ω_S and the confidence weight ω_C are investigated independently to illustrate the impact either value has on the fill-order.

The Structure Weight

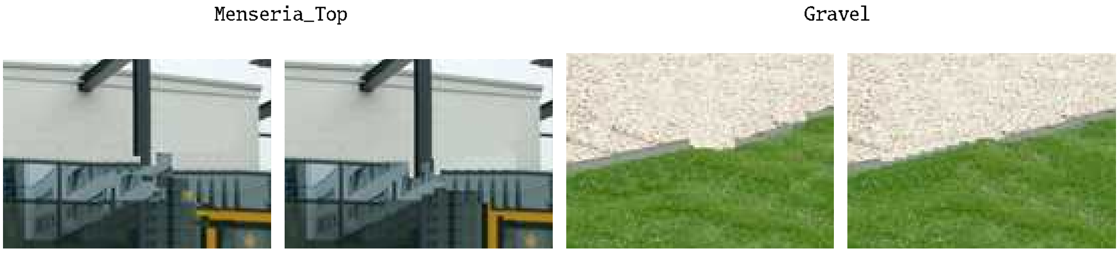
Applying the additive priority approach using only the structure weight is the first step in this evaluation. The existence of two parameters implicates a higher complexity investigating them at the same time. The confidence weight ω_C is, therefore, assume to be $\omega_C = 0.5$ to simplify the process. The structure weight ω_S, as described in section

5.1.1, is used to weigh structure orientation and the structure significance. Using a higher value for ω_S results in a dominance of the significance component, whereas lowering it increases the influence of the orientation of the structure. The PSNR data obtained by applying the inpainting method using several different settings for ω_S is shown in Table 5.4. An attempt to identify trends towards certain settings from these results is unsatisfactory. Highest and lowest PSNR values are spread over all columns with no particular response for a specific ω_S. As it was pointed out in section 3.3.4 concerning the Cheng extension, this is justified by the statement that different images required different adjusted parameter settings. Hence, the data suggests an individual adjustment of ω_S to improve the inpainting quality.

	PSNR (dB)				
	Structure weight with $\omega_C = 0.5$				
Image	0.2	0.4	0.5	0.6	0.8
G_Entrance_Pillar	48.69	49.29	49.19	49.07	49.01
G_Entrance_Top	47.75	47.32	47.41	48.01	47.19
G_Entrance_2sq	44.64	42.70	42.30	42.29	44.01
MikadoPlatz_TopLeft	49.06	48.70	49.02	49.02	48.73
LibFront_HorizLong	39.79	39.53	39.55	39.76	38.88
LibFront_RightBottom	41.23	40.74	41.76	42.60	42.34
Gravel	32.59	33.30	34.23	33.66	33.58
Menseria_LeftBottom	43.13	42.77	43.21	43.14	43.24
Menseria_Top	35.63	35.76	35.11	34.55	35.71
Obelisk_Centre	33.02	33.85	33.83	33.85	33.58
CampusWater	31.59	33.73	33.85	33.56	33.41

Table 5.4: PSNR values obtained by varying the structure weight ω_S in equation (5.2). The inpainting was carried out using a fixed confidence weight $\omega_C = 0.5$, a static patch size $M = 9$ and a search window size $\theta = 6M$.

Another visual examination of the test set is required to find a regularity of adjusting ω_S and to understand the impact of the structural weight on the results. The image in Figure 5.11(a), for instance, was generated with $\omega_S = 0.2$ defining the structure orientation as the dominant structural component. Although it does not provide a perfect result, it illustrates that the strongest isophotes in the image are continued correctly into the target region. Even their intersection in the centre of the target region is reasonably good regardless of its complexity. In contrast, a higher weight $\omega_S = 0.8$ inverts the previous ratio stressing the structural significance. Strong isophotes in Figure 5.11(b) are still continued, such as the vertical beam and the glass façade, but their intersection is less appealing. The dominant structure significance increased the priority of patches close to the beam due to its more significant coefficients. Consequently, its isophotes are propagated with higher priority and eventually displace the horizontal isophote bordering the glass façade.



(a) $\omega_S = 0.2$ (b) $\omega_S = 0.8$ (c) $\omega_S = 0.2$ (d) $\omega_S = 0.8$

Figure 5.11: Excerpts of inpainting results computed using the additive priority approach with a fixed confidence weight $\omega_C = 0.5$. The inpainting process used a static patch size $M = 9$ and a search window $\theta = 6M$.

The second image reveals a different effect that the weighing of the structural components can have. It illustrates the true complexity of finding the most suitable ω_S. Figure 5.11(c) depicts an image inpainted with a dominating orientation component. The first assumption is to have the isophotes, traversing from left to right, continued and connected. Unfortunately, the orientation of isophotes is more important than their significance and therefore the small isophotes present in the gravel texture are continued into the target region at the same pace as those framing the grey paving stone. This leads to the disrupted isophotes, where the gravel texture grows into the grass texture. Swapping the dominance for the same image produces the inpainting result in image 5.11(d). It shows a better connection of the major isophotes in the centre. However, the result is still not really sufficient.

The two exemplar images illustrate that, as suggested, the weight ω_S has to be adjusted individually to each image for satisfactory results. The magnitude of impact on the fill-order is obvious in the images in Figure 5.11. Provided the most suitable ω_S is found for an image, its inpainted quality can be increased, as shown in Figure 5.11(d). However, the complexity of adjusting ω_S, especially in images with a large number of isophotes, makes it difficult to find this most suitable value. Figure 5.11 even illustrated that opposing weights in different images can create equivalent inpainting results depending on their structural characteristics. The adjustment of this parameter is, therefore, highly empirical and requires a certain degree of experience.

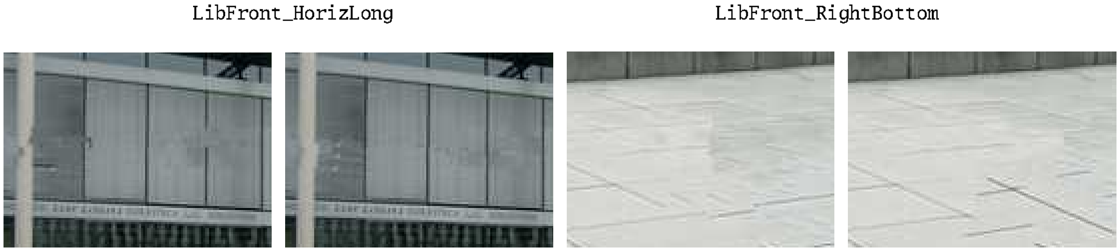
The Confidence Weight

The second weighing parameter ω_C is used in equation (5.3) where it balances the confidence component and the weighted structure obtained from equation (5.2). A low ω_C lets the priority tend towards the confidence reducing the importance of the structural components and vice versa. To evaluate only the influence of ω_C on the inpainting results the structure weight is set to $\omega_S = 0.5$. Applying the additive priority approach using several settings for ω_C obtained inpainting results corresponding to the PSNR data in Table 5.5.

	PSNR (dB)				
	Confidence weight with $\omega_S = 0.5$				
Image	0.2	0.4	0.5	0.6	0.8
G_Entrance_Pillar	49.04	49.11	49.19	48.81	49.08
G_Entrance_Top	47.24	47.14	47.41	47.28	47.89
G_Entrance_2sq	42.49	42.88	42.30	42.15	41.99
MikadoPlatz_TopLeft	48.99	48.58	49.02	49.07	49.20
LibFront_HorizLong	39.52	39.81	39.55	40.46	40.96
LibFront_RightBottom	42.14	42.15	41.76	41.01	40.88
Gravel	33.61	33.82	34.23	32.60	34.38
Menseria_LeftBottom	42.93	43.21	43.21	43.08	42.53
Menseria_Top	35.88	35.89	35.11	35.95	35.27
Obelisk_Centre	32.49	33.71	33.83	33.10	33.92
CampusWater	30.44	33.44	33.85	30.80	33.30

Table 5.5: PSNR values for inpainting results obtained by varying the confidence weight ω_C. The structure weight is set to $\omega_S = 0.5$ and a static patch size of 9 pixels is used. The search window size is set to $\theta = 6M$.

As opposed to the data shown in Table 5.4 the PSNR values in Table 5.5 are more orderly arranged. Investigating the highest and lowest values for each image reveals that a high PSNR for a low ω_C always responds with a low PSNR for the higher ω_C results. Image *LibFront_RightBottom* for instance has its highest values at $\omega_C = 0.4$ and the corresponding lowest PSNR at the other end of the tested range of settings with $\omega_C = 0.8$. This confirms the statement in section 3.3.4 as well as the assumption this modification is based on. The correlation between texture and structure in an image can be represented in the fill-order by adjusting ω_C resembling this correlation. Since an image is either texture or structure dominated, the PSNR data responds accordingly with high and low PSNR values at opposite ends of the range of settings. In addition, the PSNR data reveals another tendency for ω_C. The medium value 0.5 produces several highest PSNR values and, in average, is only 0.46 dB below the highest PSNR for each result. Therefore, it suggests that this value tends to provide good results for images regardless of their texture and structure correlation.



(a) $\omega_C = 0.2$ (b) $\omega_C = 0.8$ (c) $\omega_C = 0.2$ (d) $\omega_C = 0.8$

Figure 5.12: Excerpts of inpainting results computed with different confidence values ω_C. The structure weight in the additive priority calculation is set to $\omega_S = 0.5$. A static patch size $M = 9$ is used with a corresponding search window $\theta = 6M$.

Substantiating these observations and suggestions, once again, requires the visual investigation of examples. Figure 5.12(a) contains a high amount of structure with isophotes of different strengths and directions. Using a low ω_C in the inpainting process limits the influence of the isophotes and proceeds mainly based on high confidences. This leads to displaced and disconnected isophotes, especially, for the stronger vertical ones. Increasing ω_C inverts this behaviour and emphasises the structural information in the priority. As a result, most of the vertical isophotes are continued and connected correctly with a few artefacts near the pillar on the left side. The opposite behaviour is observable in Figure 5.12(d) where the emphasis on the structure introduces strong isophotes in the target region cutting through the texture of the pavement. The dominance of strong isophotes in the fill-order overrules smaller isophotes intersecting in the target region and disrupts them. Reducing ω_C, however, provides a more texture-driven fill-order that allows less strong isophotes to be connected as well and returns a more uniform pavement.

The results in Figure 5.12 substantiate the observations based on the PSNR data in Table 5.5. Depending on the correlation between structure and texture, the obtainable results adjusting ω_C are more satisfying than without. Furthermore, the change induced by modifying ω_C is compliant with its expected behaviour. If a higher ω_C produces an improved inpainted image, a lower value is less sufficient and vice versa. This makes the adjustment of ω_C much more intuitive and is, therefore, the preferable weight to adjust.

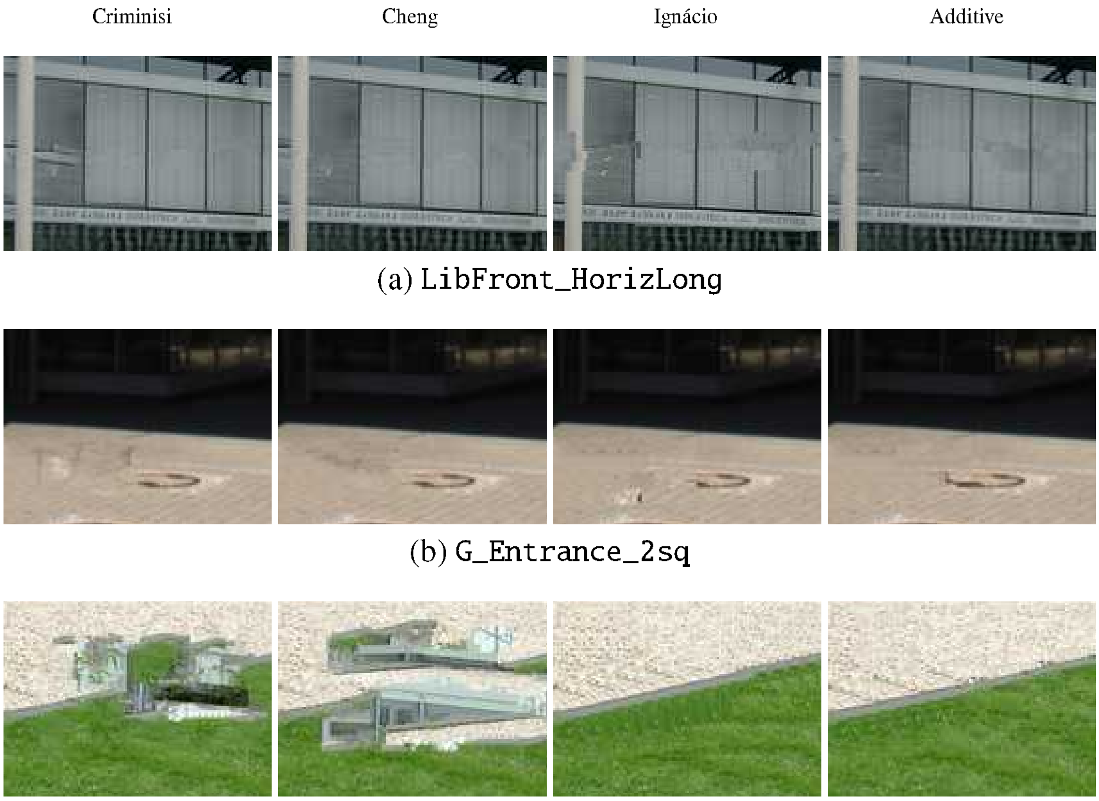
Comparing Approaches

The difficulty of finding the most suitable parameters ω_S and ω_C for each image individually poses the question if the inpainting results obtained by an optimal set of parameters provides the desired improvement compared to the three existing approaches. Empirically retrieving the most suitable settings for ω_S and ω_C and the images in the test set provides the data in Table 5.6. These data, however, suffer intensely from the problem described in section 5.2.1. The PSNR values for the image `Gravel`, for instance, represent a quality of results that is absolutely misleading and do not resemble their actual quality at all. The corresponding inpainted images in Figure 5.13(c) illustrate this. The PSNR data is presented for completeness but is ignored in favour of comparing the quality based on exemplar images.

	PSNR (dB)				
Image	Criminisi	Cheng	Ignácio	Additive	Additive (9-6)
`G_Entrance_Pillar`	44.62	49.17	48.81	48.37	49.29
`G_Entrance_Top`	44.62	48.72	48.00	49.96	48.01
`G_Entrance_2sq`	44.26	44.5	43.65	44.71	44.64
`MikadoPlatz_TopLeft`	39.55	53.23	48.10	49.61	49.20
`LibFront_HorizLong`	40.93	39.82	39.33	40.00	40.96
`LibFront_RightBottom`	31.23	40.55	40.82	41.01	42.60
`Gravel`	35.25	33.70	33.88	30.66	34.38
`Menseria_LeftBottom`	42.43	43.60	42.63	42.65	43.24
`Menseria_Top`	36.45	28.9	33.65	34.87	35.89
`Obelisk_Centre`	29.72	29.46	33.82	33.20	33.92
`CampusWater`	31.89	37.05	35.14	34.14	33.85

Table 5.6: PSNR values computed from the respective inpainting results of the denoted approaches. The additive priority approach uses a dynamic patch size with $\theta = 3M$ and weights set to $\omega_C = 0.5$ and $\omega_S = 0.5$ respectively. The second additive approach represents the highest PSNR value determined during the evaluation above. It uses a static patch size $M = 9$ and the search window size $\theta = 6M$. The abbreviation *9-6* denotes the minimum patch size and selected factor for the search window size.

The examples in Figure 5.13 compare the results created by these three approaches respectively to those created using an optimal set of parameters for equation (5.3). The patch size is set to a static size of 9 pixels. It turns out that the result produced using additive instead of multiplicative priorities are a visual improvement to the results obtained by the Ignácio approach. It suffers from visible patch-shaped artefacts in the corresponding excerpt in Figure 5.13(a) and repetitive texture in Figure 5.13(c). They can both be reduced using the additive fill-order approach. The comparison with [CPT04] and [CHL+05] is more difficult since the results vary dramatically. Their results depicted in Figure 5.13(a) are very good with the Cheng extension superseding Criminisi and producing a very plausible inpainted region. It connects the most important isophotes correctly and provides reasonable inpainting in the textured area. This can also be applied to the results in Figure 5.13(b) where the only difference is that the additive approach reconstructs the gully cover slightly better than the Cheng extension.



(a) LibFront_HorizLong

(b) G_Entrance_2sq

(c) Gravel

Figure 5.13: Excerpts of inpainting results obtained by the Criminisi, Cheng and Ignácio approaches as well as the proposed additive priority approach. The used approach is specified above the corresponding column. Additive inpainting uses a static patch size $M = 9$, the search window size $\theta = 6M$ and the respective weights $\omega_S = 0.5$ and $\omega_C = 0.5$.

In contrast, the results presented in Figure 5.13(c) are the very opposite of the previous observation. Inpainted solutions provided by [CPT04] and [CHL+05] are not even close to plausibility with elements from the surrounding building being introduced in the gravel and grass texture, as shown in Figure B.8. The Ignácio approach as well as the proposed additive modification provide much better results with the latter being, once again, the most plausible result with the least artefacts. It substantiates the suggestion that the additive priority approach produces similar or improved results compared to the most sufficient ones obtained by Criminisi, Cheng or Ignácio.

5.2.4 Weighted Mixed Priority

The second modification proposed in section 5.1.2 tries to reduce the complexity of finding the correct set of parameters for an image by using a single weight ω_C. There is no corresponding weight to ω_S, the structure components are multiplied without any additional weighing. It was explained in the previous section that obtaining good results by adjusting ω_S is difficult due to the complexity of correlation between the strength and orientation of isophotes. The mixed priority method is therefore an approach to simplify this by using a single more intuitive weight ω_C.

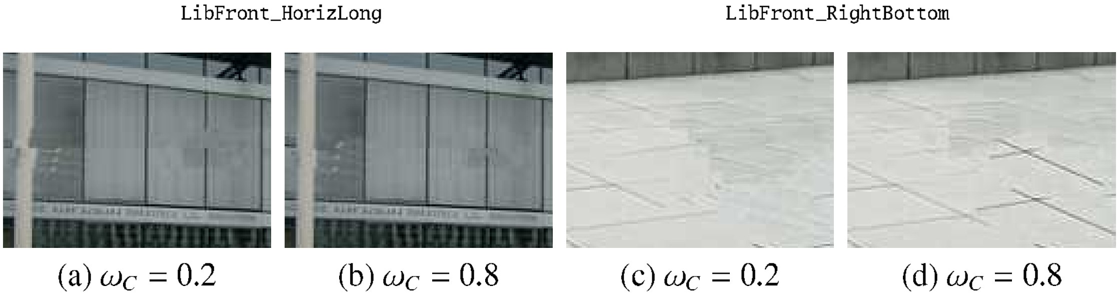
The Confidence Weight

The priority value, calculated using the ω_C in equation (5.4), is reverting the splitting of the structure value defined in [IJ07] and provides a single scalar denoting the combined structural information of orientation and significance. This value is then weighted against the confidence value as described in section 5.1.2. It is similar to the confidence weight in section 5.1.1 and is, therefore, denoted with the same symbol.

	PSNR (dB)				
	Confidence weight ω_C				
Image	0.2	0.4	0.5	0.6	0.8
G_Entrance_Pillar	49.11	49.23	49.09	49.02	48.77
G_Entrance_Top	47.08	47.28	46.61	46.56	47.95
G_Entrance_2sq	44.48	44.90	44.94	45.10	35.24
MikadoPlatz_TopLeft	49.80	49.72	49.13	48.58	47.92
LibFront_HorizLong	39.98	40.00	40.26	40.22	39.99
LibFront_RightBottom	41.58	41.91	42.07	41.54	42.23
Gravel	33.08	32.80	33.67	34.10	33.87
Menseria_LeftBottom	42.31	42.52	42.32	42.56	42.55
Menseria_Top	34.56	35.68	35.89	35.04	35.15
Obelisk_Centre	25.93	31.16	32.01	33.07	33.34
CampusWater	29.94	33.17	33.01	32.97	30.87

Table 5.7: PSNR value for different confidence weights ω_C using the mixed priority equation (5.4). The static patch size $M = 9$ and search window size $\theta = 6M$ were used for inpainting.

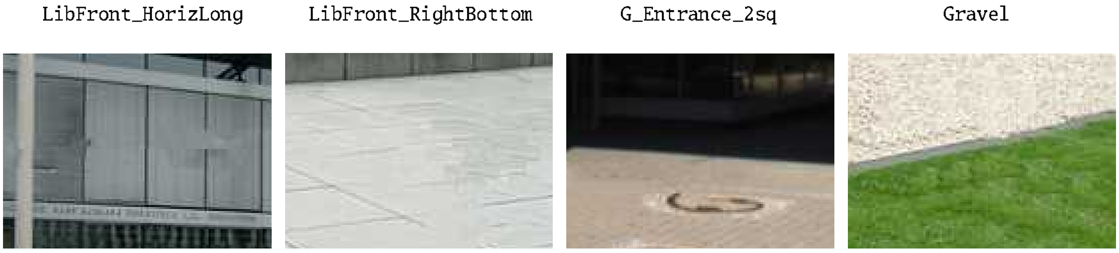
The PSNR data produced by the inpainted test set is shown in Table 5.7 and is not providing very sufficient information on the quality of mixed priority inpainting. Compared to the data in Table 5.5, there is no distinguishable tendency for a suitable ω_C per image and the pattern of highest and lowest PSNR being located at opposite ends of the table has also disappeared. The only observable property is that $\omega_C = 0.5$ produces inpainted results that tend to be close to the highest PSNR value. More precisely, their difference averages 0.33 dB. This suggests, at least, this setting is generally suitable to a larger number of images and can, therefore, be considered a good default value. This is illustrated by the inpainted images in Figure 5.15.



(a) $\omega_C = 0.2$ (b) $\omega_C = 0.8$ (c) $\omega_C = 0.2$ (d) $\omega_C = 0.8$

Figure 5.14: Inpainting results obtained by using varying confidence weights in the mixed priority approach. The inpainting procedure used a static patch size $M = 9$ with $\theta = 6M$.

Adjusting ω_C to the proportion of texture and structure in an image, however, is desired for an improved quality. Changing the values for ω_C influences the fill-order equivalently to the additive priority methods, as illustrated in the examples in Figure 5.14. The comparison of these inpainting results to those presented in Figure 5.12 exposes their high similarity. It illustrates that the impact of ω_C in both cases produces similarly sufficient inpainted images. The increase of ω_C puts emphasis on the isophotes in the image and provides their propagation into the target region. The isophotes in Figure 5.14(d) are better continued than those in Figure 5.14(c). In this image the higher amount of structural elements requires an increased ω_C to provide more suitable results for the isophotes. The image in Figure 5.14(c) in contrast has an improved result using a lower ω_C since the complexity of intersecting isophotes abutting the target region responds better to texture-driven inpainting. In equivalence to the statement in section 5.2.3, increased emphasis on inpainting the texture improves the results because of numerous isophotes crossing the target region. Their varying significances make it difficult to continue all isophotes equivalently without disrupting any of them. Emphasising the structure components in the image in Figure 5.14(d), however, decreases its quality due to several strong isophotes continued into the target region. They are cutting through it disrupting the texture of the pavement and its contained, but less significant, isophotes.



LibFront_HorizLong LibFront_RightBottom G_Entrance_2sq Gravel

(a) $\omega_C = 0.5$ (b) $\omega_C = 0.5$ (c) $\omega_C = 0.5$ (d) $\omega_C = 0.5$

Figure 5.15: Inpainting results computed using the mixed priority approach. The static patch size $M = 9$ and $\theta = 6M$ was used.

This emphasises that an adjustment of the fill-order to the correlation of texture and structure is crucial to produce sufficient results. ω_C is considered a suitable parameter that allows this exact adjustment and responds as expected in terms of empirical retrieval of suitable settings.

Comparing Approaches

Comparing the mixed priority approach to the three existing approaches allows similar observations to those provided in section 5.2.3. The PSNR data for each image and approach in Table 5.8 behaves similar to that in Table 5.6. The PSNR values are not providing enough insight into the inpainting quality to even consider it as quality measure. Looking at the data produced by the image `Gravel` defines the Criminisi approach as provider of the most sufficient result, whereas the images in Figure 5.16(b) depict the exact opposite. The only observable tendency is that the mixed priority approach with $M = 9$ and $\theta = 6M$ produces, in general, PSNR values that are close to the highest value. Although this is no proof for its superiority it suggests an improved inpainting quality.

The visual examination shows that this suggestion can be substantiated. The proposed modification performs better than the Ignácio approach with improved connected isophotes in Figure 5.16(a) and less repeating patterns in the textures of the image in Figure 5.16(b). Inpainting results computed using mixed priorities also provides an improvement to the Criminisi approach. In this specific image, however, the Cheng extension still performs better than the proposed extension. The reason for the reduced authenticity is the disrupted isophote in the right part of the inpainting region and the bright spot near the pillar repeated several times.

In summary, it can be stated that the mixed priority approach provides a simplified adjustable fill-order as suggested in section 5.1.2. Adjusting the fill-order to an image is more intuitive than fine-tuning ω_S in the additive priority approach. Texture and struc-

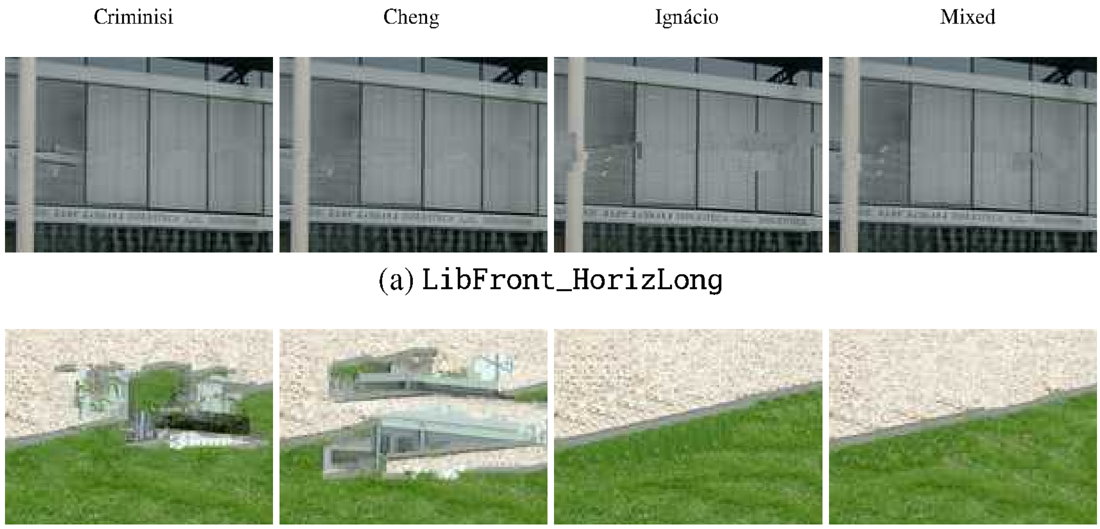
	PSNR (dB)				
Image	Criminisi	Cheng	Ignácio	Mixed	Mixed (9-6)
G_Entrance_Pillar	44.62	49.17	48.81	48.52	49.23
G_Entrance_Top	44.62	48.7	48.00	49.07	47.95
G_Entrance_2sq	44.26	44.52	43.65	44.66	45.10
MikadoPlatz_TopLeft	39.55	53.23	48.10	47.87	49.80
LibFront_HorizLong	40.93	39.82	39.33	39.25	40.26
LibFront_RightBottom	31.23	40.55	40.82	40.37	42.23
Gravel	35.25	33.70	33.88	33.09	34.10
Menseria_LeftBottom	42.43	43.60	42.63	42.63	42.56
Menseria_Top	36.45	28.90	33.65	35.19	35.89
Obelisk_Centre	29.72	29.46	33.82	32.83	33.34
CampusWater	31.89	37.05	35.14	33.75	33.17

Table 5.8: PSNR values calculated for the inpainting results of the specified approaches. The mixed priority approach uses a dynamic patch size with $\theta = 3M$ for better comparability with the three original approaches. The approach denoted Mixed (9-6) represents the highest PSNR value obtained in the preceding evaluation. It used a static patch size $M = 9$ with $\theta = 6M$.

ture are clearly recognisable in an image and weighing their correlation using ω_C is, therefore, more intuitive. Furthermore, using this single weighing parameter produces inpainted images equivalent to that additive priority approach which makes it the preferable selection when applying it to an image. The downside, however, is that it leaves a reduced margin for optimisation and adaptation than the additive priority approach.

5.2.5 Blended Filling of Patches

The last modification, proposed in section 5.1.3, focusses on the reduction of artefacts introduced during filling of the search patch, as explained in Chapter 4.3. A blending of of the sample patch coefficients with existing source region coefficients is employed to smooth the transition between the source region and the inserted coefficients. A reduction of patch-shaped artefacts is the desired result. Applying the blending method in the inpainting process together with the additive priority approach reveals an unanticipated property of the resulting procedure. Blending the sample and search patch coefficients is not only generating new coefficients in the source region depending on the blending weight ω_G, it also influences the fill-order of succeeding iterations. This influence is obvious considering that the coefficients near the boundary are altered by the blending method and, therefore, have an impact on the calculated priorities. Consequently, the blending modification has to be investigated carefully to ensure that its impact on the

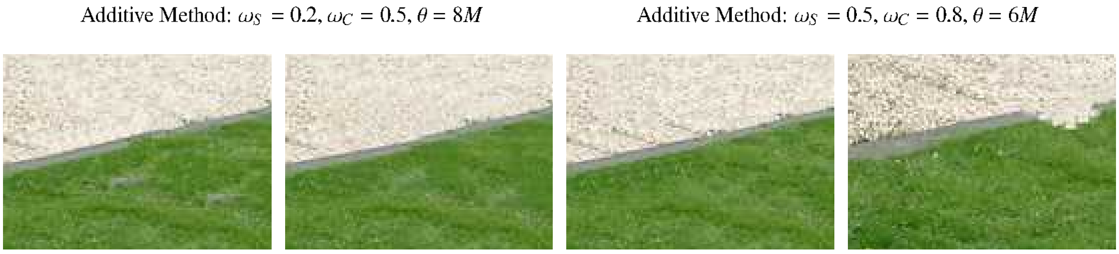


(a) `LibFront_HorizLong`

(b) `G_Entrance_2sq`

Figure 5.16: Inpainting results obtained using the Criminisi, Cheng and Ignácio approach as well as the mixed priority modification. The used approach is specified above each column respectively. The mixed priority approach uses a patch size $M = 9$ with $\theta = 6M$ and the confidence weight $\omega_C = 0.5$.

fill-order is either encouraging a better inpainting result and, more importantly, refrains from having a negative impact.

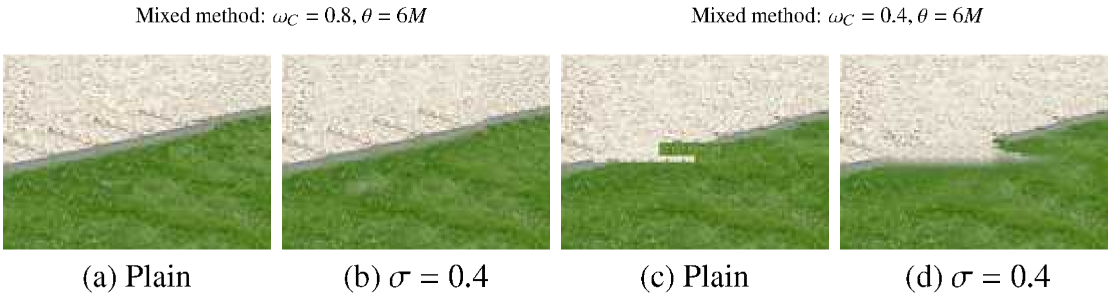

Additive Method: $\omega_S = 0.2, \omega_C = 0.5, \theta = 8M$
Additive Method: $\omega_S = 0.5, \omega_C = 0.8, \theta = 6M$

(a) Plain (b) $\sigma = 0.4$ (c) Plain (d) $\sigma = 0.4$

Figure 5.17: Excerpts of inpainting results obtained by applying the additive priority approach to the image `Gravel`. The used weights are specified in the column above the images. All results were computed using a static patch size $M = 9$.

Unfortunately, the influence it has on the additive priority fill-order is problematic. The images in Figure 5.17 show that the results produced by a plain filling is relatively sufficient with the strongest isophotes connected, merely containing repetitive patterns in the textures. With applied blending in Figure 5.17(b) the modification is minor and it can be observed that the isophotes are a little smoother compared to the plain result in Figure 5.17(a). The transition of the grass texture and the stone in the centre of the excerpt is less abrupt and a few of the patch-shape artefacts can be reduced. Additionally, the texture elements related to the stones that are introduced into the grass texture in Figure 5.17(a) have disappeared in the blended example. This suggest that blending is an improvement over the plain filling procedure. The second example, however, contradicts this. Figure 5.17(c) is another example of a sufficient inpainting result. Using the same settings with additional blending returns a less sufficient result with introduced patch-shaped artefacts and disrupted isophotes.

The reason for this negative impact on the fill-order is that the blending method smooths the coefficients in all four subbands W_a, W_h, W_v and W_d. Altering the coefficients in the source region close to the boundary implies that structure and texture characteristics are change resulting, in turn, in a modified fill-order. This alteration leads to a high influence on the additive priority method responding with a less sufficient fill-order, as shown in Figure 5.17(d). Although, this impact on the fill-order is not necessarily negative but can also provide improved fill-order results, it generally tends to be negative. This makes the application of the blending method in combination with the additive approach insufficient. Its main limitation is that it cannot be applied to an image with a sufficient fill-order to simply improving the transition between textures and reducing the artefacts. Its high impact on the fill-order risks a reduction in inpainting quality.



Mixed method: $\omega_C = 0.8, \theta = 6M$

Mixed method: $\omega_C = 0.4, \theta = 6M$

(a) Plain

(b) $\sigma = 0.4$

(c) Plain

(d) $\sigma = 0.4$

Figure 5.18: Excerpts of inpainting results obtained by applying the mixed priority approach to the image `Gravel`. The inpainting procedure was carried out using patch size $M = 9$.

Using the mixed priority equation (5.4) with applied blending, on the other hand, is more stable. The blending of source pixel coefficients imposes an amount of change that improves the visual impression with reduced artificial isophotes and repeated texture. Its impact on the fill-order, however, is limited due to better balanced confidence and structure components in equation (5.4). The images in figures 5.18(a) and 5.18(b) are results obtained by the same inpainting routine, the first one using the original filling, whereas the second one uses the blending method. It can be observed that the connected isophotes remain unchanged in the blended example with an improvement of the grass texture, where the repetitive patterns are slightly reduced.

As a contrasting example, the images in figures 5.18(c) and 5.18(d) both show insufficient inpainting results. They illustrate, however, that the applied blending method has only a limited impact on the fill-order. The transition between grass and gravel texture is smoothed in Figure 5.18(d) with the fill-order remaining mostly similar. On the one hand, this implies that the blending is incapable of improving the inpainting result generated by an insufficient fill-order. This ensures, on the other hand, that a more sufficient fill-order is not dragged into insufficiency by blending coefficients. This shows that the combination of the blending method with the mixed priority approach is more solid than its additive counterpart. The blending has little impact on the fill-order but introduces a reduction of repetitive patterns and smooths the transition between textures. It can, therefore, be stated that blending is best used with the mixed priority approach since it imposes an improvement of results without a noticeably increased influence on the fill-order.

The Blending Parameter

It remains to evaluate the influence of the blending parameter σ, described in section 5.1.3, on the inpainting process. A few different values for σ have therefore been selected and applied to the test set using additive and mixed priority calculation respectively. Their weighing parameters haven been set to 0.5 for all corresponding weights to ensure their comparability with differences mainly imposed by the varying blending parameter. The obtained PSNR results are displayed in Table 5.9. The data suggests that the used blended inpainting is an improvement over the Ignácio approach. The latter only provides a single inpainting result that is considered superior to the blended examples, whereas the blending imposes increased PSNR values.

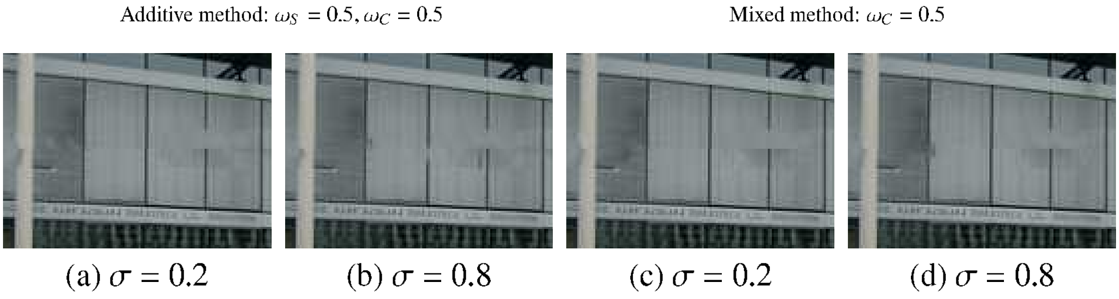
Additionally, the data shows that higher values for σ tends to produce higher PSNR values for both inpainting methods. This is an anticipated behaviour since σ resembles the standard deviation of the Gaussian distribution, as described in section 5.1.3. A higher σ reduces the degree of blending by lowering the curve of the Gaussian distribution and, therefore, the influence of sample coefficients. Hence, the data in Table 5.9 suggests that a reduction of the blending increases the inpainting quality. This is justified by the concept behind the blending method. New coefficients are created by the weighed combination of corresponding search and sample coefficients in each subband. These coefficients are less likely to match those in the original image since they are a combination of coefficients. Furthermore, the blending introduces a certain degree of blur into the inpainted region which adds to the difference between the inpainted and the original image. Although this limits the probability of achieving a perfectly accurate image, it is not stated that inpainted images are less sufficient for the PSNR-related reasons detailed in section 5.2.1.

	PSNR (dB)						
		σ (Additive)			σ (Mixed)		
Image	Ignácio	0.2	0.4	0.6	0.2	0.4	0.6
`G_Entrance_Pillar`	48.81	49.25	49.04	49.36	49.13	49.23	49.25
`G_Entrance_Top`	48.00	48.05	49.45	47.95	47.89	48.47	47.51
`G_Entrance_2sq`	43.65	42.43	42.52	42.92	44.52	44.62	43.19
`MikadoPlatz_TopLeft`	48.10	48.93	49.54	49.25	48.95	49.27	49.25
`LibFront_HorizLong`	39.33	40.33	33.52	40.40	40.39	40.78	40.34
`LibFront_RightBottom`	40.82	42.40	43.23	42.77	43.41	43.09	43.32
`Gravel`	33.88	34.09	34.30	34.40	33.75	33.80	33.94
`Menseria_LeftBottom`	42.63	43.77	40.79	43.90	43.47	43.78	43.92
`Menseria_Top`	33.65	35.92	34.47	35.98	36.13	35.37	36.21
`Obelisk_Centre`	33.82	33.84	34.03	34.10	33.74	33.86	33.86
`CampusWater`	35.14	32.65	31.71	32.88	32.82	33.31	32.12

Table 5.9: PSNR values computed for varying blending values for σ. The mixed and additive approaches both use a static patch size $M = 9$ with $\theta = 6M$. The additive weights are set to $\omega_S = 0.5$ and $\omega_C = 0.5$. Correspondingly the confidence weight for the mixed priority approach is set to $\omega_C = 0.5$.

Visually evaluating the examples in Figure 5.19 emphasises that the PSNR data discussed above is not representative for the quality of inpainting achieved by applying the blending method. The images in figures 5.19(c) and 5.19(d) depict a similar inpainted region where most isophotes are connected accurately. The improvement imposed by the blending mostly concerns the transition of differently shaded texture. The shutter contains less disrupting seams at the transition of these textures which is well-visible at the second vertical isophote from the right. The darker texture surrounding the isophote is blended into the brighter neighbouring texture. A similar effect is exposed by the artefacts at the left side of the pillar. Its edge is displaced producing step artefacts which are smoothed in figures 5.19(a) and 5.19(c). The isophote is still displaced but the blending reduces the rectangular shape of the displacement making it a less noticeable.

In conclusion, it can be stated that the use of blending in the filling of search patches can reduce the artefacts imposed by a suboptimal fill-order. The additive priority approach, however, is highly susceptible to its impact on boundary coefficients. In several cases, it results in less sufficient inpainting results, where those obtained without blending are more sufficient. It is, therefore, more important to synchronise the weighing parameters in the additive priority equation with σ to obtain the best result. The mixed priority method, on the other hand, is a more solid approach, less influenced by the blending method. Although artificial isophotes and repetitive texture are reduced in the



Additive method: $\omega_S = 0.5, \omega_C = 0.5$

Mixed method: $\omega_C = 0.5$

(a) $\sigma = 0.2$ (b) $\sigma = 0.8$ (c) $\sigma = 0.2$ (d) $\sigma = 0.8$

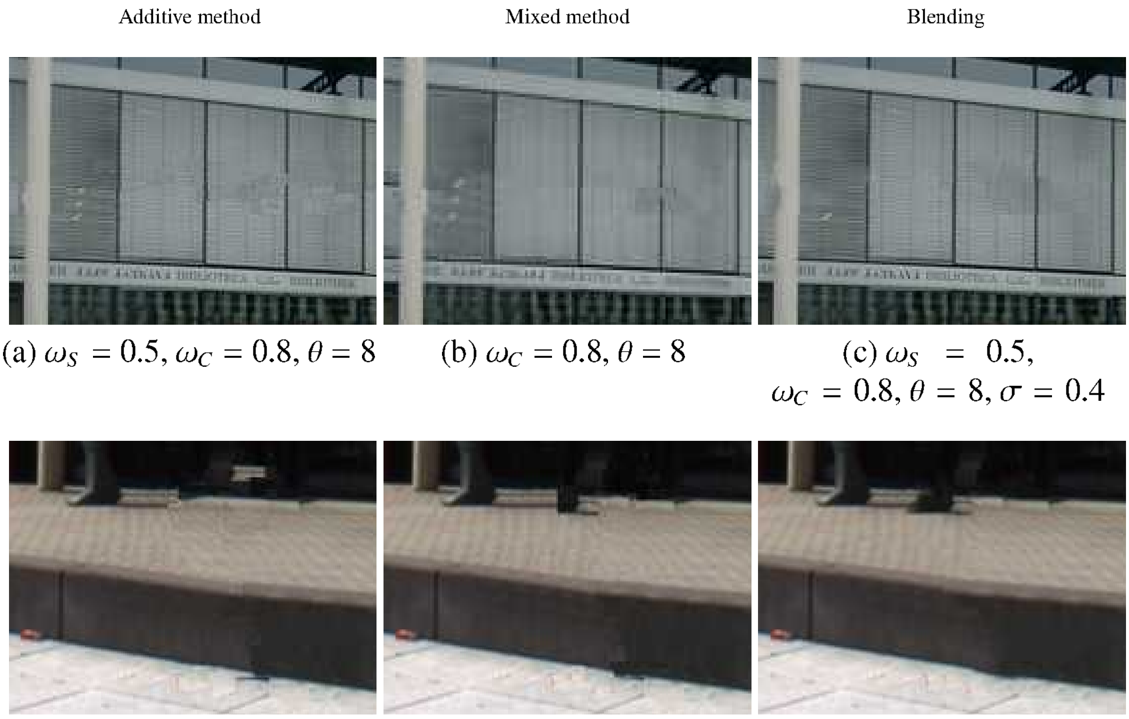
Figure 5.19: Inpainting results computed using additive and mixed priority inpainting respectively. The inpainting procedure was carried out using static patch size $M = 9$ with $\theta = 6M$. The images illustrate the results at varying values for the blending parameter σ.

inpainted region, it has a minor impact on the continuation of structural elements. Combining the mixed priority approach with the blending method is, therefore, preferable to the use of the additive priority approach.

5.2.6 Comparing the Modifications

The previous sections investigated the inpainting quality of the proposed modification individually. It remains to investigate how they perform in direct comparison to each other. This allows the identification of the modification providing the most suitable inpainting procedure for removing an object from the campus images. Evaluating the additive and mixed priority approaches in sections 5.2.3 and 5.2.4 respectively showed that both modifications have their advantages and disadvantages. The additive approach, for instance, is more adjustable to the specific properties of an input image than the mixed approach. Its parameters, however, are more complicated to adjust correctly. It is to determine whether a more adjustable approach with higher complexity is preferable or a method that is more rigid, though more intuitive, to adjust to the image characteristics.

The most important observation, in this respect, is that the optimised results for the same image using either modification method are almost the same. The inpainted images in Figure 5.20 and 5.21 clearly illustrate this. They depict the best results obtained by either modification without blending. They all used a static patch size and increased search window size for the reasons described in section 5.2.2. A closer look at these images shows that, although they have minor differences, their perceived quality is mostly similar. The images in figures 5.20(d) and 5.20(e), for instance, show only small differences in the inpainted pavement in the background and the isophotes enclosing the stone wall in the front are correctly continued through the target region. The similarity of the other example images depicted in Figure 5.20 and 5.21 are equivalent.



(a) $\omega_S = 0.5, \omega_C = 0.8, \theta = 8$

(b) $\omega_C = 0.8, \theta = 8$

(c) $\omega_S = 0.5, \omega_C = 0.8, \theta = 8, \sigma = 0.4$

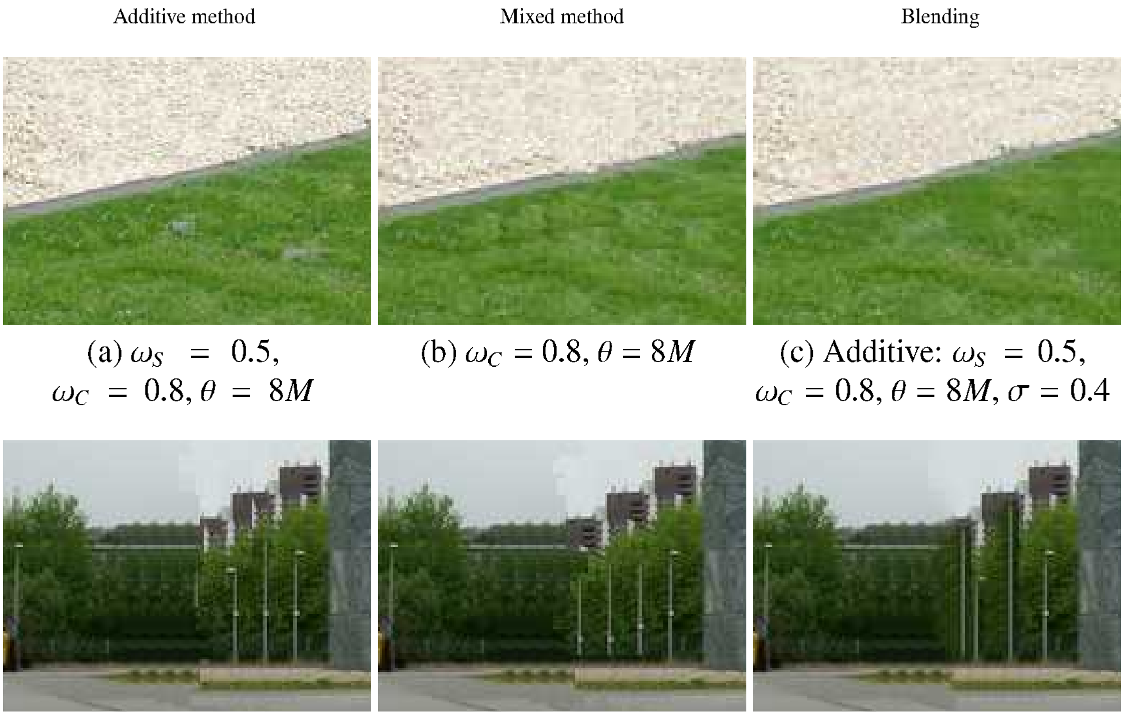
(d) $\omega_S = 0.5, \omega_C = 0.5, \theta = 8$

(e) $\omega_C = 0.4, \theta = 6$

(f) $\omega_C = 0.4, \theta = 6, \sigma = 0.4$

Figure 5.20: Images inpainted using the method specified above each column. All three procedure used a static path size $M = 9$. The additive and mixed priority method inpaint without blending applied. The blending column shows the best obtained result using the blending modification.

According to these similar inpainting results, they also suffer from a common problem that is inherited from the three other inpainting approaches. Their main objective is to connect isophotes abutting the target region correctly and inpaint the remaining texture around them. As this is a straightforward idea on simple or ideal images as described in section 4.2.1 it becomes more and more complex as the number of isophotes grows. Figure 5.20(a) and the other two corresponding results show such an example where the shutter consists of numerous isophotes in horizontal and vertical direction. It is difficult to determine a fill-order that propagates all isophotes in the most suitable order into the target region. This becomes even more complicated when many of these isophotes intersect in the target region. Finding the correct sample patch containing a similar intersection to that required in the target region is difficult and, depending on the textures in the source region, might not be possible at all. This leads to visible artefacts where isophotes are disconnected or displaced, as shown in the images in Figure 5.20. Retrieving a fill-order that allows to connect all isophotes correctly is hard to achieve in an image with such a complex structure. The evaluation and proposed modifications for the fill-order could reduce the problem but are still not able to fully reconstruct the target region without disconnecting isophotes and corresponding artefacts.


Additive method

Mixed method

Blending

(a) $\omega_S = 0.5$, $\omega_C = 0.8, \theta = 8M$

(b) $\omega_C = 0.8, \theta = 8M$

(c) Additive: $\omega_S = 0.5$, $\omega_C = 0.8, \theta = 8M, \sigma = 0.4$

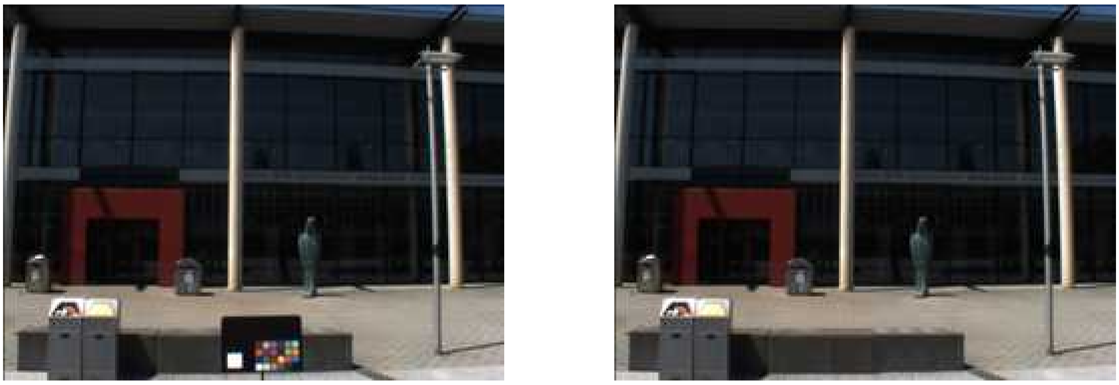
(d) $\omega_S = 0.5$, $\omega_C = 0.5, \theta = 8M$

(e) $\omega_S = 0.4, \theta = 6M$

(f) Mixed: $\omega_C = 0.4$, $\theta = 6M, \sigma = 0.4$

Figure 5.21: Inpainted images using additive and mixed priority inpainting with and without blending applied. The used method is specified above each column. The blending column shows the best results obtained using either additive or mixed priority inpainting. All methods used a static patch size $M = 9$.

A proposed solution to reduces the visibility of this problem was the introduction of a blending modification that reduces patch-shaped artefacts and smooths small differences in inpainted textures. It was described in section 5.2.5 that the combination of blending with the additive priority method can have a negative effect on the fill-order and even decrease the quality of the inpainted image compared to its non-blended variant. Combining it with the mixed priority modification, however, was shown to provide more stability by not decreasing a good fill-order. Hence, a suitable mixed priority fill-order for a specific image is generally improved by adding blended filling. In the images in figures 5.20(f) and 5.21(f) the patch-shaped artefacts in adjoining textures are reduced by blending the coefficients in the filling step and smoothing their transition compared to the corresponding inpainting results without blending. Blending also reduces the repetitiveness of texture described in section 4.3.2 that is introduced by sampling from the same neighbouring texture multiple times as it is achieved in Figure 5.21(c). Nonetheless, the improving influence of the blending modification is limited. Large repeated textures as well as adjoining textures that are too different are not removed or reduced by smoothing the coefficients. The repetitive background in Figure 5.21(f), for instance, is impossible to remove using the blending step. The still suboptimal inpainted texture produced by the modified fill-order contains too much repeating elements to blur them.



(a) Original

(b) Inpainted Image

Figure 5.22: Original and inpainted image with colour checker removed. Inpainting used the mixed priority approach with $\omega_C = 0.5$. The patch size is $M = 9$ with the search window size $\theta = 12M$ and blending uses $\sigma = 0.4$.

Summarising these observations shows that the most sufficient and reliable modifications the mixed priority approach extended by the blending modification. The similarity of optimised results produced by both an additive and a mixed fill-order lead to the objection of the additive approach in favour of its more solid counterpart. This requires the adjustment of a single parameter for the fill-order, the confidence weight ω_C, to adapt the priorities to the characteristic structure and texture relation in an image. The most suitable ω_C is to be retrieved empirically for each image but is more intuitive than adjusting two parameters in the additive priority approach. Together with the added blending that co-operates well with the mixed priority fill-order, the resulting approach provides sufficient results in the majority of examples. It is, therefore, more applicable to the colour checker problem than those investigated earlier in this work, as Figure 5.22 shows.

Chapter 6

Conclusion

This work investigated three different approaches from the fields of image inpainting and medical image processing respectively to evaluate their applicability to removing a colour checker from campus images. The results obtained by an initial evaluation of the selected approaches showed that applying the defect pixel interpolation used in image processing [AM01] provides no suitable solution for this problem due to its reliance on periodic low-frequency information in the Fourier domain. It, therefore, struggled with highly detailed and structured images as those provided in the test set which lead to its exclusion concerning further consideration. Image inpainting, however, turned out to be a much more solid field of research concerning the removal of objects from real images. The evaluation of two specific approaches developed to provide an inpainting solution showed that the Criminisi and Ignácio approaches provide good inpainting results for real images but still suffer from some several weaknesses. A close examination provided an insight into these approaches that allowed to identify the source of their shortcomings and, thereupon, the proposal of modifications to overcome these limitations.

The problem of a fill-order that is not suitable for the specific set of highly structured images was addressed by proposing two modifications concerning the calculation of priorities. The additive modifications are based on the concepts and observations described in [CHL$^+$05] which were investigated thoroughly in the first part of this work. A higher stability with more distinguishable priority values was achieved by adding the components in the priority equation instead of multiplying them. Additionally, two weights were introduced that allow to control the balance between its two structural components and, in turn, the resulting weighed structure value and the texture-related component. This offers the possibility of adjusting the fill-order to the specific proportion of structure and texture in an image. The correct adjustment results in an improved inpainting result. Finding the optimal set of parameters, however, is highly empirical and can be counterintuitive.

The second fill-order modification tries to reduce the complexity of the adjustment process by combining the structural components at equal terms, simply balancing the structure and texture components. A single adjustment parameter remains, reducing the complexity of finding a suitable weight. Furthermore, controlling the balance between texture and structure is less complicated due to their more intuitive resemblance in the image. In addition, reducing the number of parameters results in a less flexible fill-order that offers less control over the priorities than the additive approach. The inpainting results showed, however, that the results of optimised parameters for both modifications are in most cases equivalent which made the mixed priority approach the more appealing. Controlling the adjustment parameters allows a better control over the connection and continuation of isophotes as well as the inpainting of remaining textures. This provides good results for several images from the test set assuming the most suitable parameters are used. Several images, however, contain too complex structures and too many isophotes to be inpainted with the proposed modifications. Additionally, a large amount of isophotes intersecting in the target region make it especially difficult to find a suitable fill-order and construct a sufficient intersection from the available sample patches.

The last proposed modifications addressed the repetitiveness of texture and artefacts introduced by adjoining textures. A patch-based blending uses a Gaussian distribution to determine a blending weight for corresponding search and sample patch coefficients. The results of applying this modification in the filling process depends mostly on the used fill-order. It was discovered in the evaluation that the additive priority is easily influenced by the blended coefficients and the quality of the fill-order can be reduced. In contrast, the fill-order determined by the mixed priority approach is more stable and blending generally improves the inpainting results. It requires, however, a suitable fill-order for an image to add improving qualities and has no capability of creating sufficient inpainting results from a less suitable fill-order. Although it reduces the patch-shaped artefacts and repeating patterns, it is not possible to eliminate them completely.

In summary, it has to be stated that there is no generic solution to automatically remove an object such as the colour checker from an image. Although the contents of the images investigated in this work are very similar in terms of their texture and structure correlation, the remaining differences still require an adjustable solution to provide sufficient results. An approach proposing an automated solution, therefore, depends on the identification and classification of different image properties for an automated adjustment of the described parameters.

Future Work

Although the proposed modifications produce improved results on the test images compared to the considered existing approaches, several problems remain that offer the opportunity for further research. Achieving a fully automated inpainting algorithm is the most appealing of these remaining challenges. Assuming that an automated inpainting algorithm is desired, a technique that requires the adjustment of multiple parameters is difficult to use. The optimal values for a given input image would have to be retrieved automatically, determining its proportion of texture and structure. One idea would be to extract their proportion in an image from the high- and low-frequency coefficients in the approximation and detail coefficients in the respective subbands. The result could then be used to determine the most suitable parameter controlling the fill-order in either modification.

Another limitation of the proposed modifications is that the blending, applied when filling the search patch, is only producing a limited amount of smoothing and, therefore, cannot reduce all repeating textures and patch-shaped artefacts. Adjoining textures that are similar, yet change their shade relative to their position in the image, still persist with applied blending. The use of Poisson blending [PGB03] instead of the proposed method based on a Gaussian distribution could be a better alternative considering its described qualities concerning a similar application in [HE08]. It would also be interesting to investigate the latter approach proposing a database-driven search for sample textures synthesising images from a contextually similar dataset.

The search window that has been subject to a minor improvement by increasing its size in the proposed methods also provides a basis for an additional investigation and potential improvements. It was stated above, that repeated textures remains a problem in the proposed modifications. Reducing these repetitions depends on the selection of a suitable search window that increases the amount of available sample patches without increasing the probability of wrongly sampled patches as in [CPT04]. A possibility to circumvent this could be to adapt the search window. Instead of using a fixed search window with a certain size and shape, an initial clustering step that partitions the image into a few segments could be implemented. Numerous well-known clustering algorithms should be considered and compared in terms of efficiency determined by the increase in quality and the added computational expense. Possible candidates that come to mind are Gaussian mixture model (GMM) and a simple intensity quantisation.

Appendix A

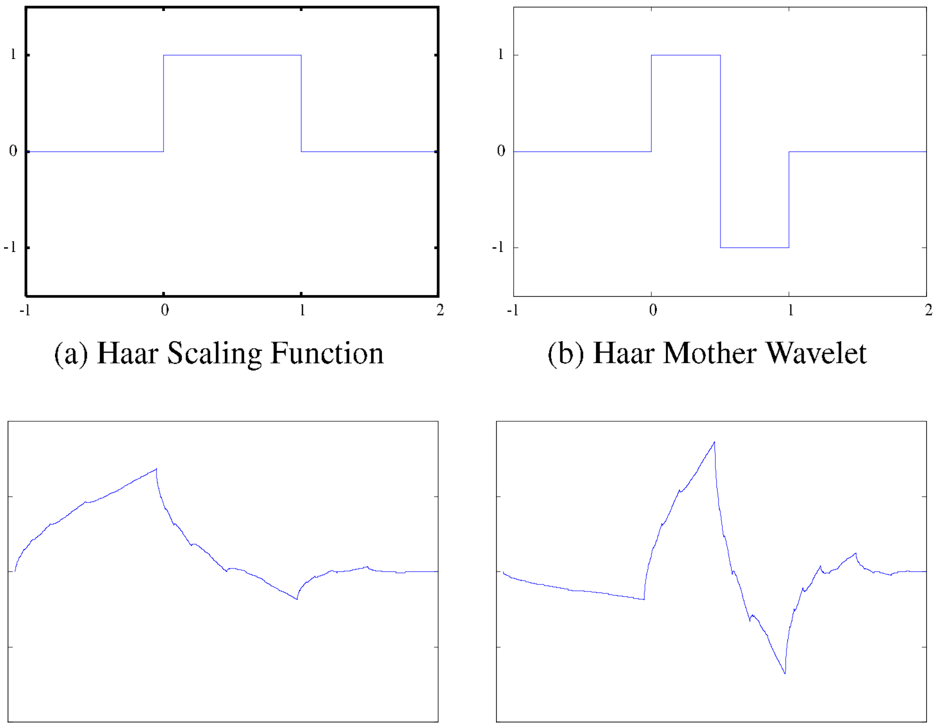
The Wavelet Transform

The use of wavelets and the wavelet transform is centre in this work and the approach described in [IJ07]. This chapter, therefore, provides a short introduction into the world of wavelets. It discusses the continuous wavelet transform (CWT) and its advantages in section A.1 and compares it to the short-time Fourier transform (STFT) in section A.2. This leads to the discrete wavelet transform (DWT) in section A.3 and its derived multiresolution representation which is eventually applied to two-dimensional signals.

A.1 The Continuous Wavelet Transform (CWT)

If spectral information is to be extracted from a signal, the Fourier transform is the first mathematical procedure that comes to mind. The continuous wavelet transform, however, describes another way to decompose a signal into its frequencies. Its advantage compared to other spectral transformations is that it allows you to examine spectral characteristics of a signal in frequency and time simultaneously. Similar to the Fourier transform, the CWT can be separated into wavelet *analysis* and *synthesis*. Wavelet analysis breaks down a signal into its spectral components, whereas synthesis combines these components back together, reconstructing the signal.

The basic idea is to use "'local' wavelike function[s]" [Add02] to analyse a signal. Such a small wave or function is called a *wavelet*. Figure A.1 shows two commonly used examples. A wavelet in its fundamental form is the *mother wavelet* that is used in the transform process to generate different *daughter wavelets*. They allow a much more detailed investigation of the signal. A new daughter wavelet is derived from its mother by shifting it along the time axis and stretching or squeezing it. These manipulations are referred to as *translation* and *dilation* respectively. Formally, a dilated and translated mother wavelet is written in normalized form as



(a) Haar Scaling Function

(b) Haar Mother Wavelet

(c) D4 Scaling Function

(d) D4 Mother Wavelet

Figure A.1: The plots above show two examples of commonly used wavelets. Plots (a) and (b) are the simplest wavelet called *Haar* named after Alfréd Haar who was the first to propose it. (c) and (d) are the scaling and mother wavelet from the well-known Daubechies wavelet family of discrete orthogonal wavelets named after Ingrid Daubechies.

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right) \tag{A.1}$$

a being the dilation parameter and b controlling the translation of the mother wavelet $\psi(t)$. With a specific wavelet and a range of parameters, a and b, equation A.1 generates multiple wavelets that are used to analyse the signal.

To illustrate the specific properties introduced by the wavelet transform, a one-dimensional signal (e.g. sound) is considered. One dilated daughter wavelet is then shifted along the time axis. At each location its shape is compared to the shape of the signal. The obtained coefficient qualifies the similarity of their respective shapes. Hence, a high coefficient indicates a well-matching wavelet, opposed to a low value that indicates the opposite. These observed coefficients denote the frequencies captured by this wavelet

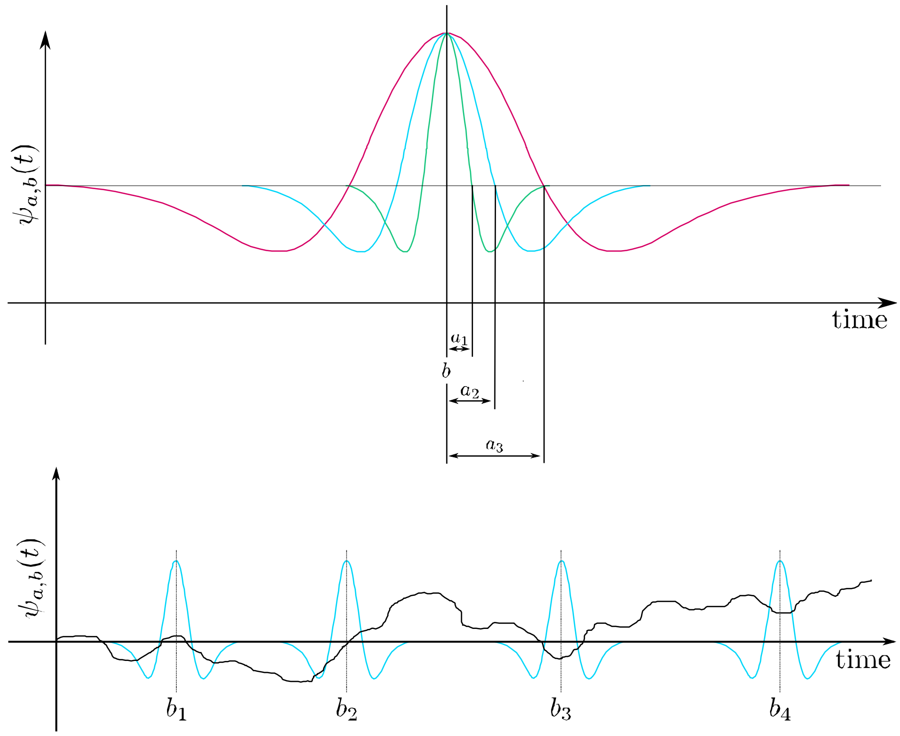


Figure A.2: The top graph illustrates different dilations of a wavelet. b denotes the position at which the wavelet is centred and a_1, a_2 and a_3 are different dilation parameters resulting in new daughter wavelets. Each of the daughter wavelets is shifted along the time axis, as shown in the graph at the bottom. The shifting parameters b_1 to b_4 define the locations the wavelet is centred at. Each shifted wavelet produces a response coefficient for these locations.

and how they vary over time. Figure A.2 illustrates the procedure of this comparison whilst sliding it along the axis. The same procedure is repeated with other wavelets retrieved by dilating the mother wavelet. They are shifted along the time axis as well, and their corresponding coefficients are collected. After all wavelets have been processed, the result in the time-frequency plane shows coefficients representing frequencies from high to low and how these frequencies change over time. This simultaneous representation of frequencies and time is of high interest and only has a restricted equivalent in the Fourier analysis.

A.2 Wavelets and the Windowed Fourier Transform

The windowed Fourier transform was developed to obtain an important set of information not provided by a basic Fourier transform. Time-related information is lost when transforming into frequency space due to the integration over the whole signal. In short, a spectral line contributes to the whole signal instead of just a single moment in time. Hence, it is not possible to identify the frequencies composing the signal at a specific time. This limits the use of the Fourier transform to applications that require frequency information only.

An attempt to master this limitation was to provide a window function that provides a snapshot of the signal and transforms this short interval into the frequency domain. Applying this to multiple different intervals of a signal returns time-located frequencies with respect to their interval. This is known as the windowed or short-time Fourier transform (STFT) and is defined as

$$F(f,b) = \int_{-\infty}^{\infty} x(t)h(t-b)e^{-i2\pi ft}dt \tag{A.2}$$

where $h(t-b)$ denotes the window function with b specifying the time shift. The complex sinusoid $e^{-i2\pi ft}$ is restricted by this window function, which [Add02] describes in more detail. Common window functions include the Hamming window [1], the Gauss windows and others. All these functions share a property; the time-resolution of all intervals in the transform is *fixed* once a window size is chosen.

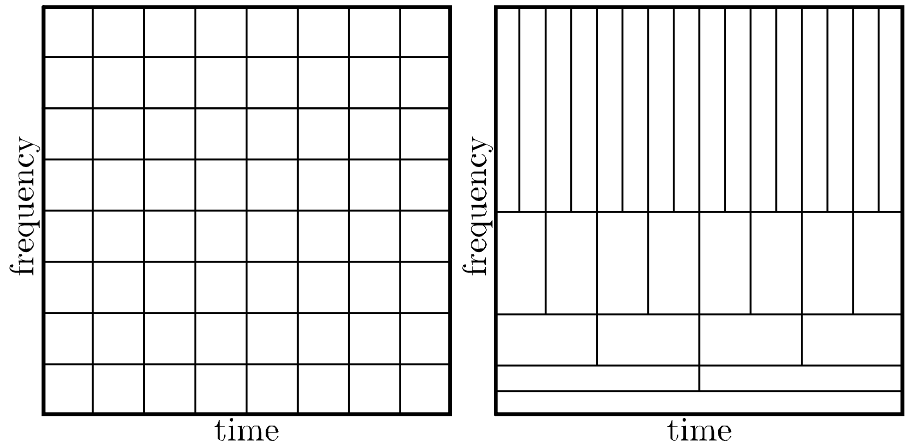
Comparing the STFT to the previously introduced wavelet transform unfolds a major difference looking at the time-frequency plans shown in Figure A.3. To understand the different representation, it is important to remember the *Uncertainty* or *Heisenberg Principle* that applies in both cases. It states that the product of frequency and time resolution is always greater than a minimum or in a more formal way:

$$\Delta_\omega \Delta_t \geq \frac{1}{2} \tag{A.3}$$

Δ_ω specifies the frequency bandwidth and Δ_t denotes time duration. In consequence, a high resolution in time causes a low frequency resolution, whereas increasing the frequency resolution eventually leads to a less accurate time resolution. In the STFT, the fixed window size constrained by the *Uncertainty Principle* induces a constant time and frequency resolution (Figure A.3). Therefore, depending on the selected window and its size, a good resolution to distinguish between high frequencies constitute a bad resolution for lower frequencies or vice versa.

On the contrary, the dilation of wavelets exploits the Uncertainty Principle to some extent by adapting its time and frequency resolution. It provides higher accuracy in time resolution for low frequencies, which usually influence a larger time interval. While the dilation parameter is modified, the precision changes at a similar pace. This produces

[1] Window function named after Richard Wesley Hamming who described it in his book "Digital Filters" [Ham98]



(a) STFT time-frequency resolution

(b) Wavelet time-frequency resolution

Figure A.3: The two time-frequency planes show the difference in time and frequency resolution using a short-time Fourier or wavelet transform. The shapes of the rectangular boxes correspond to their respective resolutions enforced by the Uncertainty Principle. The advantage of wavelets is their adaptation of resolution to frequencies over time whereas the STFT is restricted to the same time-frequency resolution.

high frequency and low time resolution for low frequencies, which is slowly turned around towards higher frequencies. Therefore, wavelet transforms are the favourable choice in applications where the correspondence between frequencies and their location in spatial or temporal domain are crucial.

A.3 The Discrete Wavelet Transform (DWT)

The continuous dilating and translating of wavelets results in highly redundant information. This redundancy has to be eliminated to make the CWT practical and efficient. Furthermore, to use the wavelet transform in signal processing requires discretely sampled wavelets that are dilated and shifted in discrete steps. Their respective step size is fixed for a transformation defining two parameters. a_0 denotes the dilation and b_0 the translation of the wavelet. They remain unchanged for a certain transformation. The only constraints applying are a_0 has to be greater then 1, whereas b_0 can take a value greater then 0 and is proportional to a_0. Under the assumption that m and n are integers, determining dilation and translation equation (A.1) can be transformed into

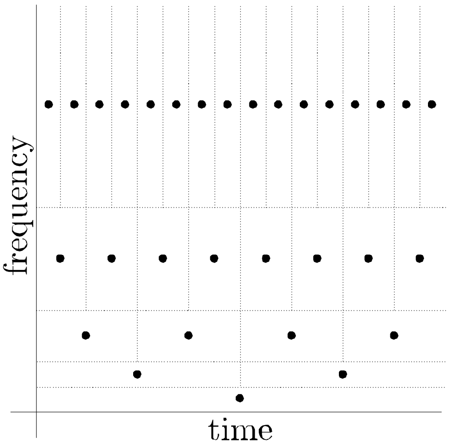


Figure A.4: The representation of *dyadic grid* above shows the sampling steps used for the discrete wavelet transform. It represents a logarithmic sampling controlled by $a_0 = 2$ and $b_0 = 1$. The dots represent the position where a discrete wavelet is centred.

$$\psi_{m,n}(t) = \frac{1}{\sqrt{a_0^m}} \psi \left(\frac{t - nb_0 a_0^m}{a_0^m} \right) \tag{A.4}$$

This equation postulates a so called *discrete wavelet.* In general, it is sampled on a *dyadic* grid which corresponds to $a_0 = 2$ and $b_0 = 1$. This is shown in Figure A.4.

Multiresolution Representation

The most important development for the use of wavelets in image processing came from the work of Stéphane Mallat [Mal89]. He found a way to link orthogonal wavelets to filters commonly used in image processing, namely high-pass and low-pass filters. This required the introduction of the *scaling function.* It is used in multiresolution decompositions to describe the connection between different resolution levels.

The *scaling function* that could be used for wavelet decomposition has to satisfy certain conditions. Together with its corresponding mother wavelet, it has to form an orthogonal basis and, therefore, cannot be dilated. This is to ensure the limitless dilation and translation of the mother wavelet. An example of two scaling functions and their respective mother wavelets is shown in Figure A.1.

The idea behind multiresolution decomposition using such a pair of wavelets is now very simple. The scaling function can be viewed as the low-pass filtering part of the decomposition and provides its so called *approximation coefficients.* The approximation is basically a smoothed version of the signal that is drained of its high detail information. These details however are encoded in the high-pass-filtered result extracted using the

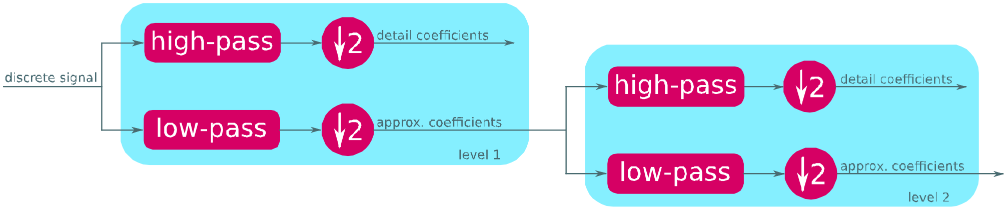


Figure A.5: The filter-based decomposition of a discrete signal into the wavelet domain applies a high-pass and low-pass filter to the signal respectively and downsamples the results. The cascade above shows the recursive application of these filters to retrieve multilevel decomposition. A decomposition is only applied to the previous approximation coefficients.

mother wavelets. As a result, coefficients are obtained that are usually referred to as *detail coefficients*. The separation of approximation and detail coefficients can be carried out recursively to achieve a multiresolution decomposition. The two decomposed parts of the next resolution level are computed by filtering the approximation coefficients on the current level as shown in Figure A.5. A more formal way of describing the decomposition for an univariate signal is

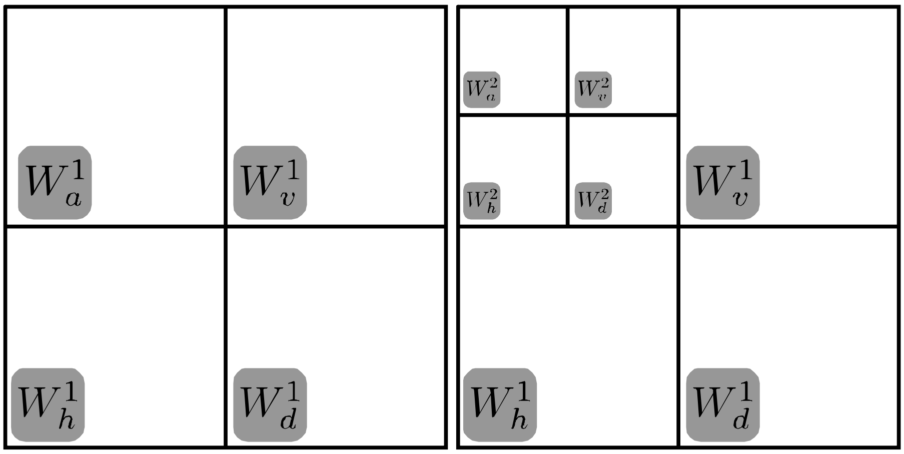
$$S_{m+1,n} = \frac{1}{\sqrt{2}} \sum_{k} c_k S_{m,2n+k} \tag{A.5}$$

to calculate the approximation coefficients c_k at location k in the previous approximation $S_{m,2n+k}$ where m and n denote the dilation and translation parameters. The detail coefficients are then calculated using

$$T_{m+1,n} = \frac{1}{\sqrt{2}} \sum_{k} b_k S_{m,2n+k} \tag{A.6}$$

where b_k denotes the coefficient at location k of the previous approximation of the signal. The decomposition at each level creates two transformed signals of equal length that are half the size of the input signal. This is justified by a downsampling step that follows each high-pass and low-pass filtering to reduce the amount of redundant information. Each higher level coefficient can be reconstructed from its lower level approximation and detail coefficient. This leaves the same amount of coefficients on every level of the decomposition. Related coefficients on a decomposition level are often summarised under the name *subband*.

A signal can now be decomposed using a discrete version of the wavelet transform. However, to be useful in real applications, the transform has to be reversible. The reconstruction of a higher level representation and eventually the original signal is formally defined as



(a) Tranform, 1 level

(b) Transform, 2 levels

Figure A.6: The two images show the common visual representation of a wavelet-transformed two-dimensional signal. Subband W_a^1 in (a) represents the approximation coefficients and W_v^1, W_h^1 and W_d^1 are the detail coefficients in vertical, horizontal and diagonal direction respectively. (b) represents the second level of decomposition where W_a^1 is decomposed into approximation and detail subbands.

$$S_{m-1,n} = \frac{1}{\sqrt{2}} \sum_k c_{n-2k} S_{m,k} + \frac{1}{\sqrt{2}} \sum_k b_{n-2k} T_{m,k} \tag{A.7}$$

In summary, the multiresolution representation is the tool for applying discrete wavelets to discrete sampled signals. Analysing a signal into approximation and detail subbands is easy to compute using orthogonal wavelets resembling image processing filters.

Two-Dimensional Wavelet Transforms

The last question that remains unanswered is: how to apply this to a two-dimensional signal, i.e. an image? It turns out that this is very simple. Similar to applying the Fourier transform, the wavelet decomposition is first carried out in horizontal direction followed by a second decomposition in the vertical direction. Figure A.6(a) shows the result of a one-level transform assembled of four wavelet subbands. The top-left subband W_a^1 denotes the approximation subband at the first level. W_h^1, W_v^1 and W_d^1 are the detail subbands encoding high-frequency responses in a horizontal, vertical and diagonal direction. The next level is computed by transforming only the approximation subband

W_a^1, as shown in Figure A.6(b). A formal definition to calculate the subbands is given by

$$S_{m+1,(n_1,n_2)} = \frac{1}{2}\sum_{k_1}\sum_{k_2} c_{k_1}c_{k_2}S_{m(2n_1+k_1,2n_2+k_2)} \tag{A.8}$$

$$T^h_{m+1,(n_1,n_2)} = \frac{1}{2}\sum_{k_1}\sum_{k_2} b_{k_1}c_{k_2}S_{m(2n_1+k_1,2n_2+k_2)} \tag{A.9}$$

$$T^v_{m+1,(n_1,n_2)} = \frac{1}{2}\sum_{k_1}\sum_{k_2} c_{k_1}b_{k_2}S_{m(2n_1+k_1,2n_2+k_2)} \tag{A.10}$$

$$T^d_{m+1,(n_1,n_2)} = \frac{1}{2}\sum_{k_1}\sum_{k_2} b_{k_1}b_{k_2}S_{m(2n_1+k_1,2n_2+k_2)} \tag{A.11}$$

where n_1 and n_2 denote the location indices. k_1 and k_2 are the scaling indices for the coefficients in each subband at a given scale $m + 1$.

This concludes the basic introduction of the wavelet transform. Although it provides sufficient insight in understanding the techniques used in this work, more comprehensive research on wavelets can be found in [Add02] and [Hub98].

Appendix B

Images: Test Set

B.1 Images With Ground Truth

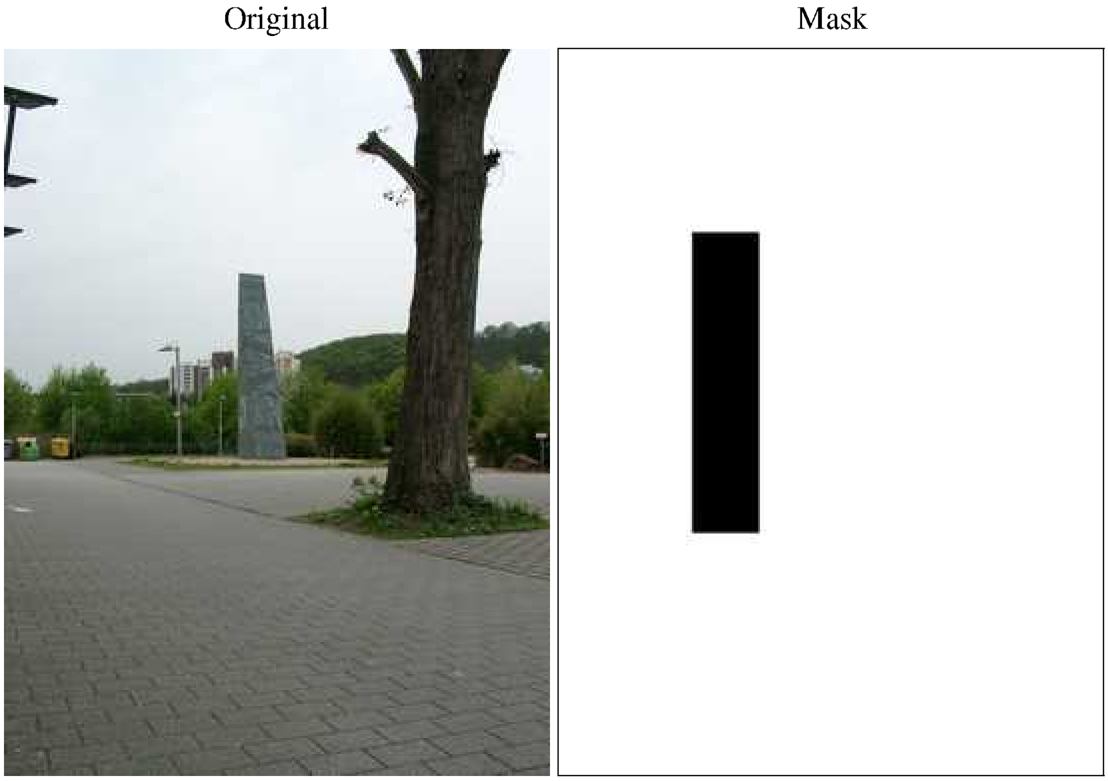
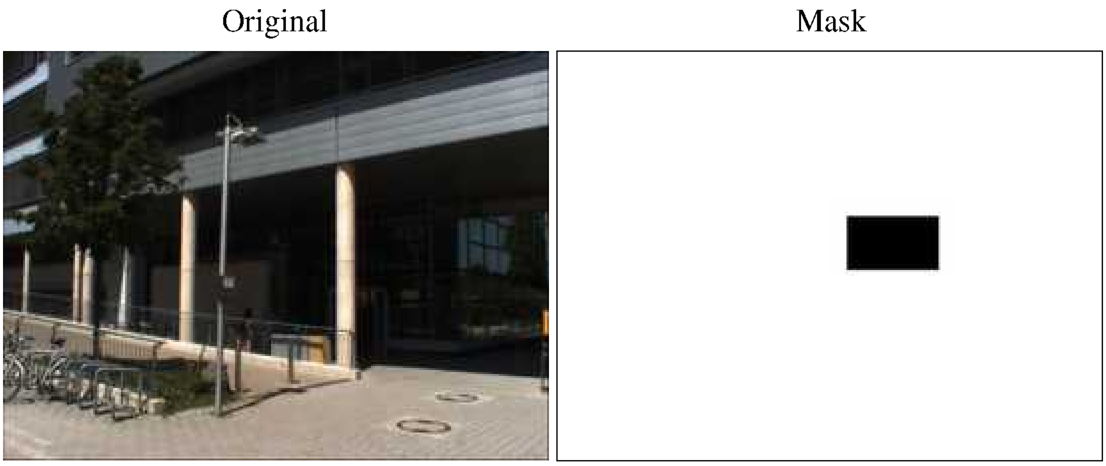


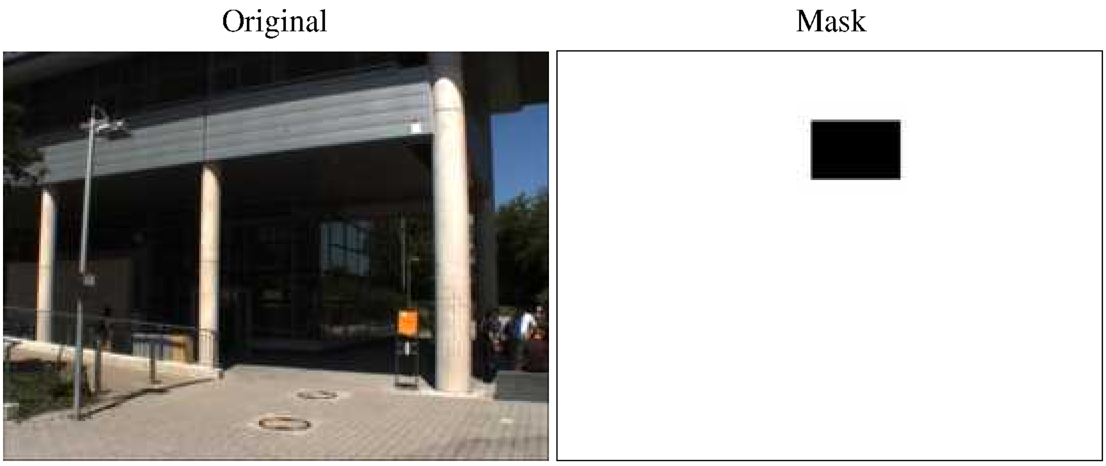
Figure B.1: `Obelisk_Centre`



Original

Mask

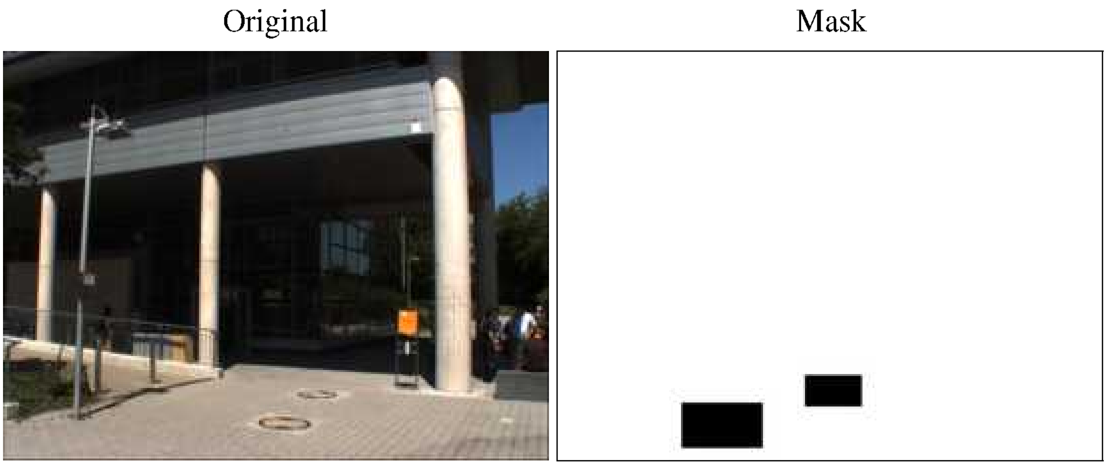
Figure B.2: `G_Entrance_Pillar`



Original

Mask

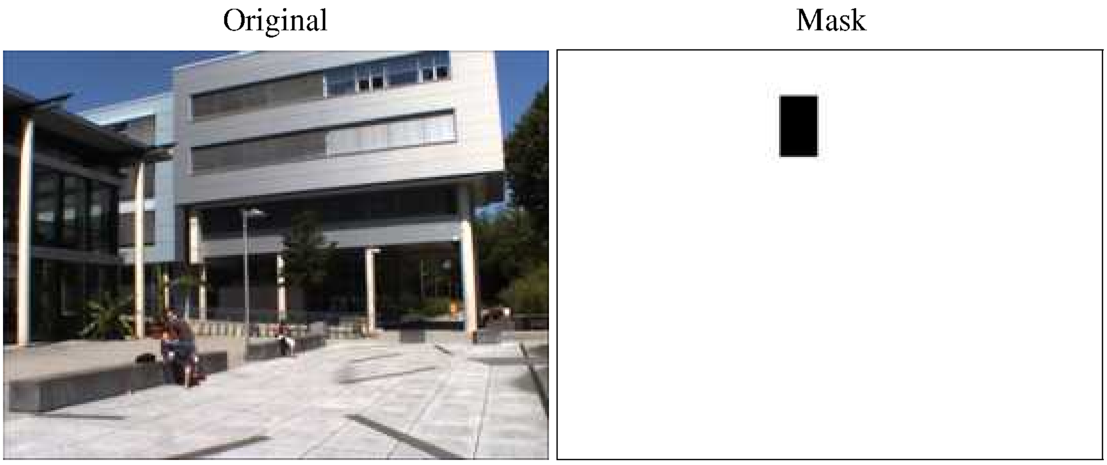
Figure B.3: `G_Entrance_Top`



Original

Mask

Figure B.4: `G_Entrance_2sq`



Original

Mask

Figure B.5: `MikadoPlatz_TopLeft`

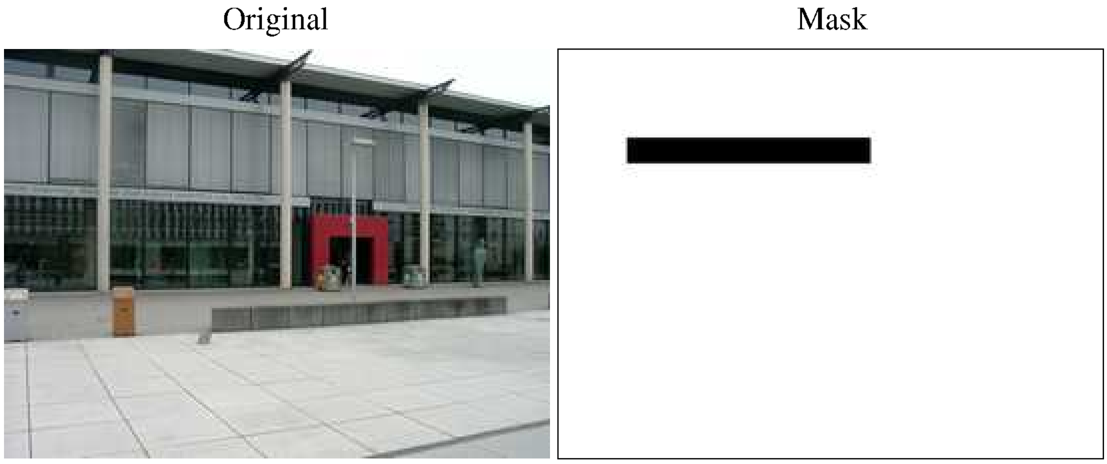


Figure B.6: `LibFront_HorizLong`

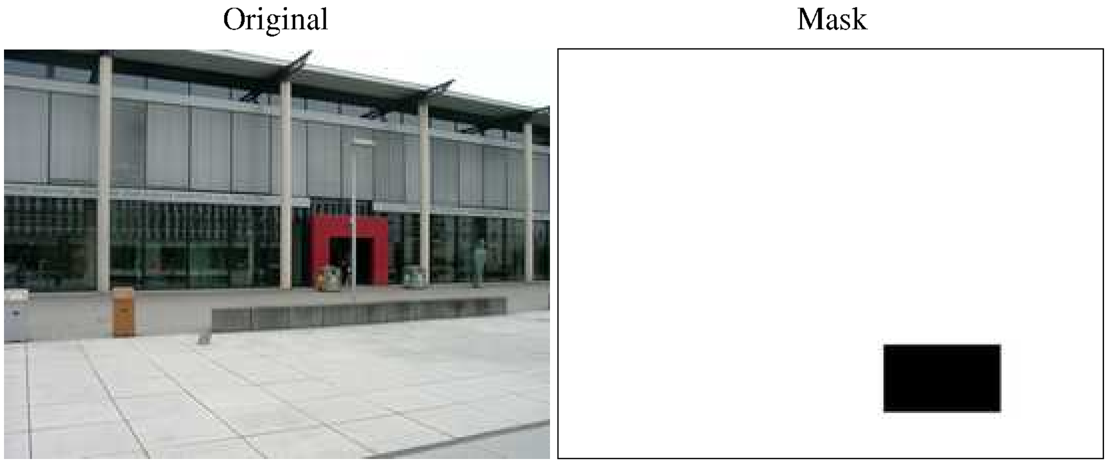
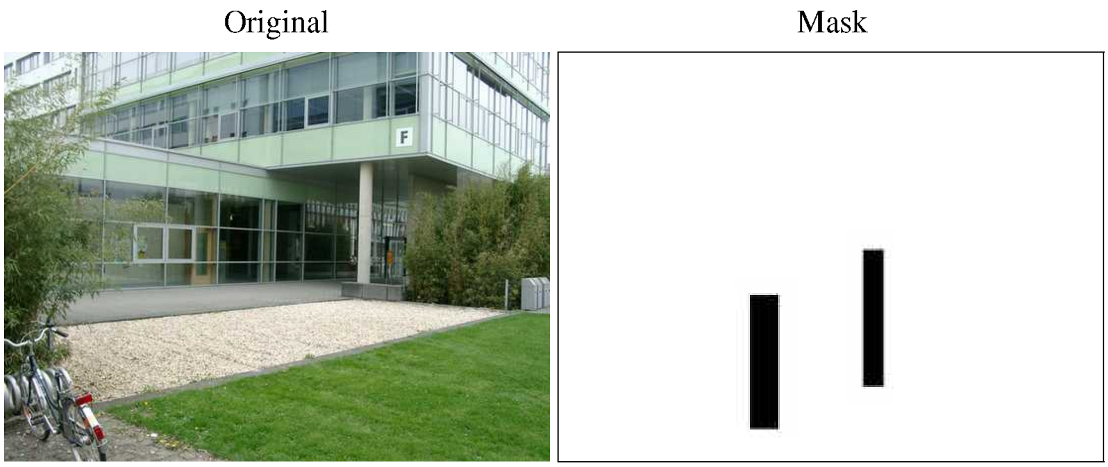


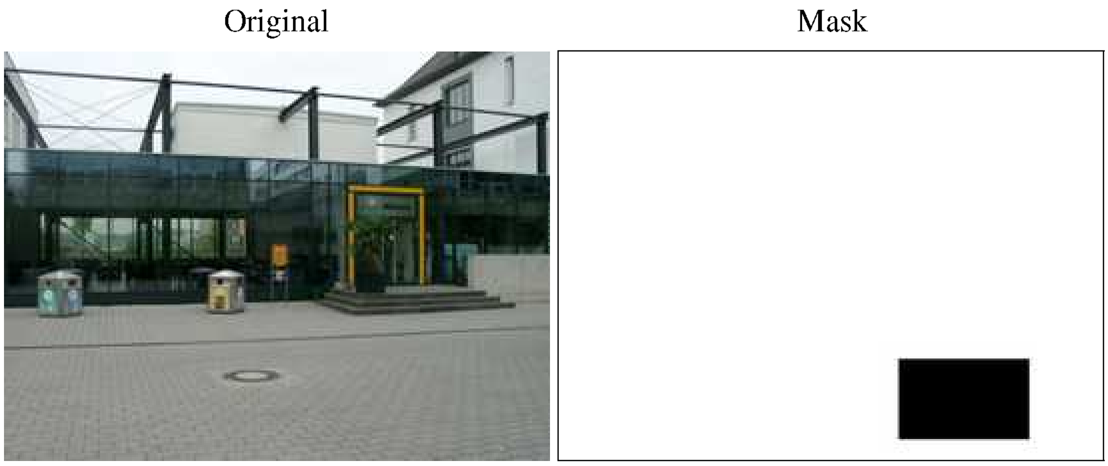
Figure B.7: `LibFront_RightBottom`



Original

Mask

Figure B.8: `Gravel`



Original

Mask

Figure B.9: `Menseria_LeftBottom`

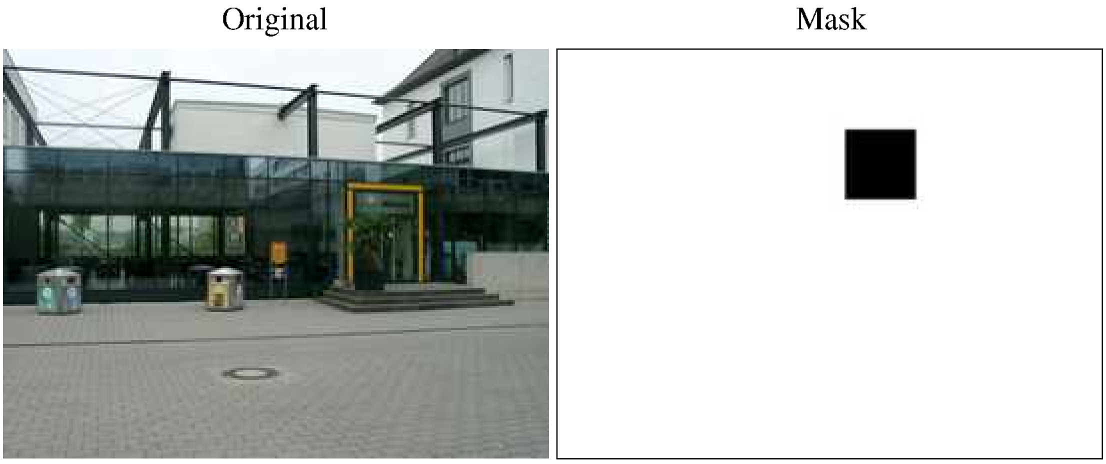


Figure B.10: `Menseria_Top`

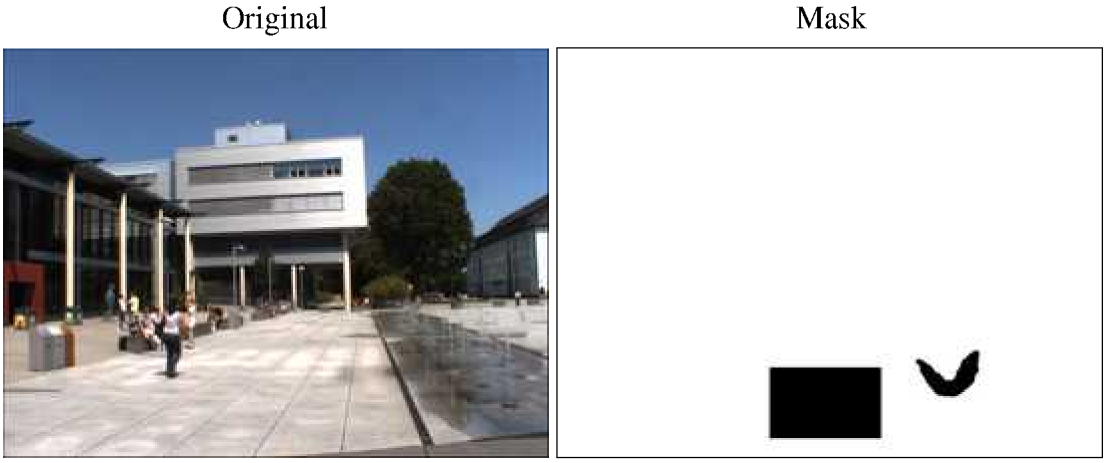
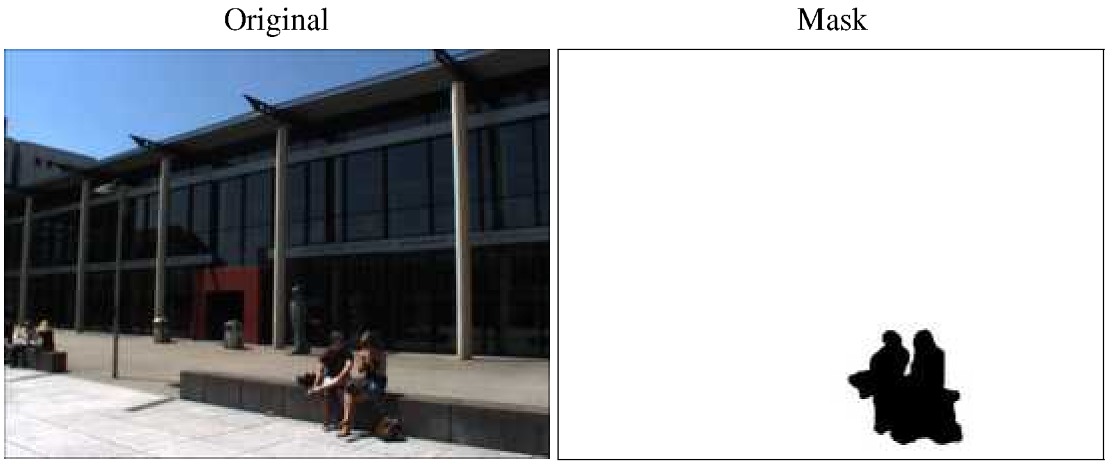


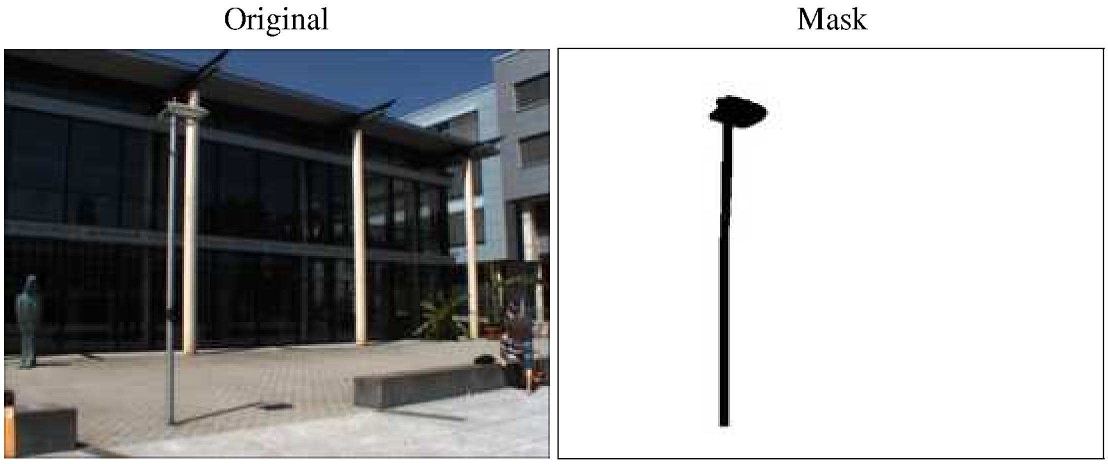
Figure B.11: `CampusWater`

B.2 Images Without Ground Truth


Original

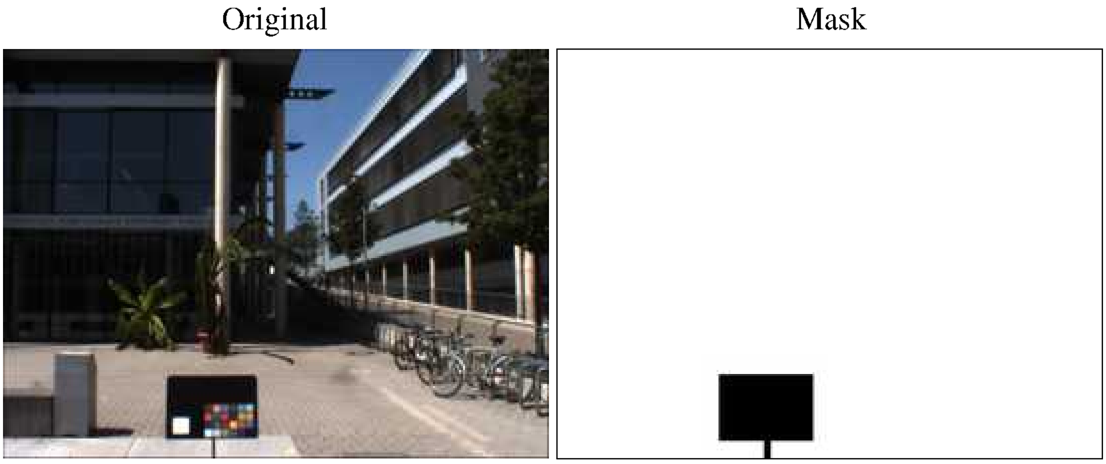
Mask

Figure B.12: `Bib_People`


Original

Mask

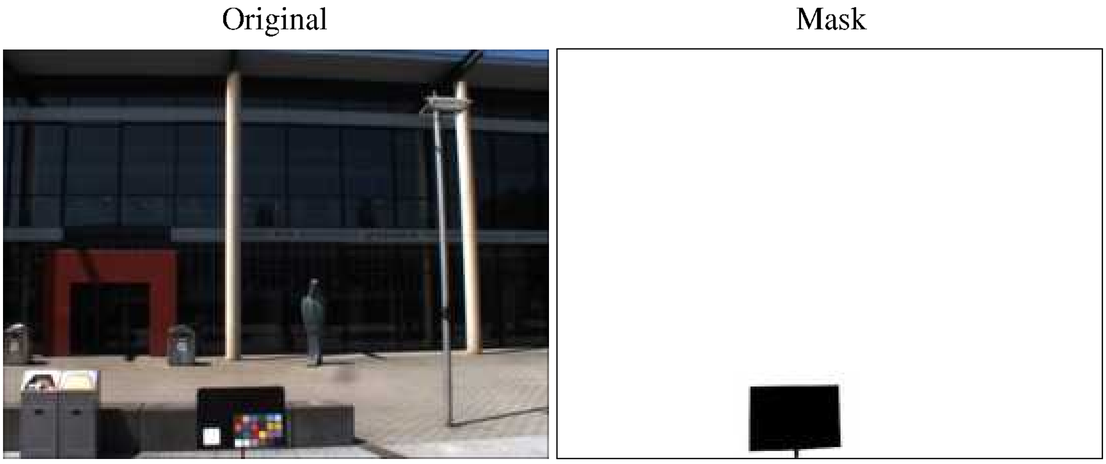
Figure B.13: `Bib_Lamppost`



Original

Mask

Figure B.14: `Bib_ColourChecker_BikeStand`



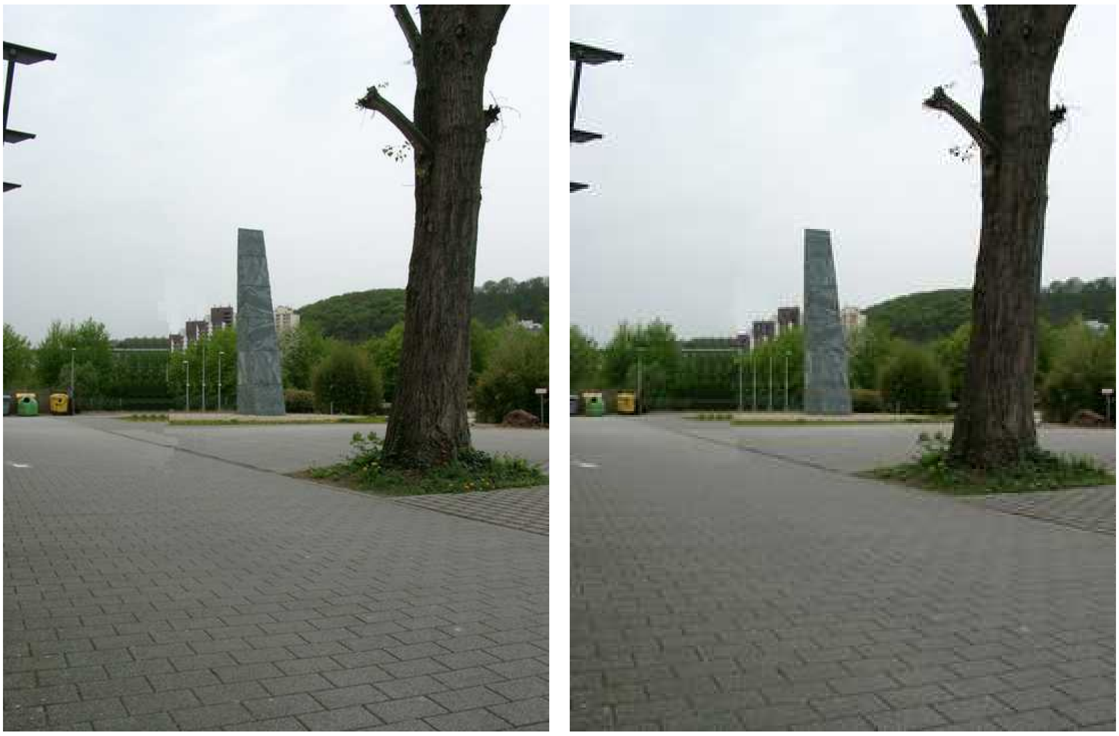
Original

Mask

Figure B.15: `Bib_ColourChecker_Entrance`

Appendix C

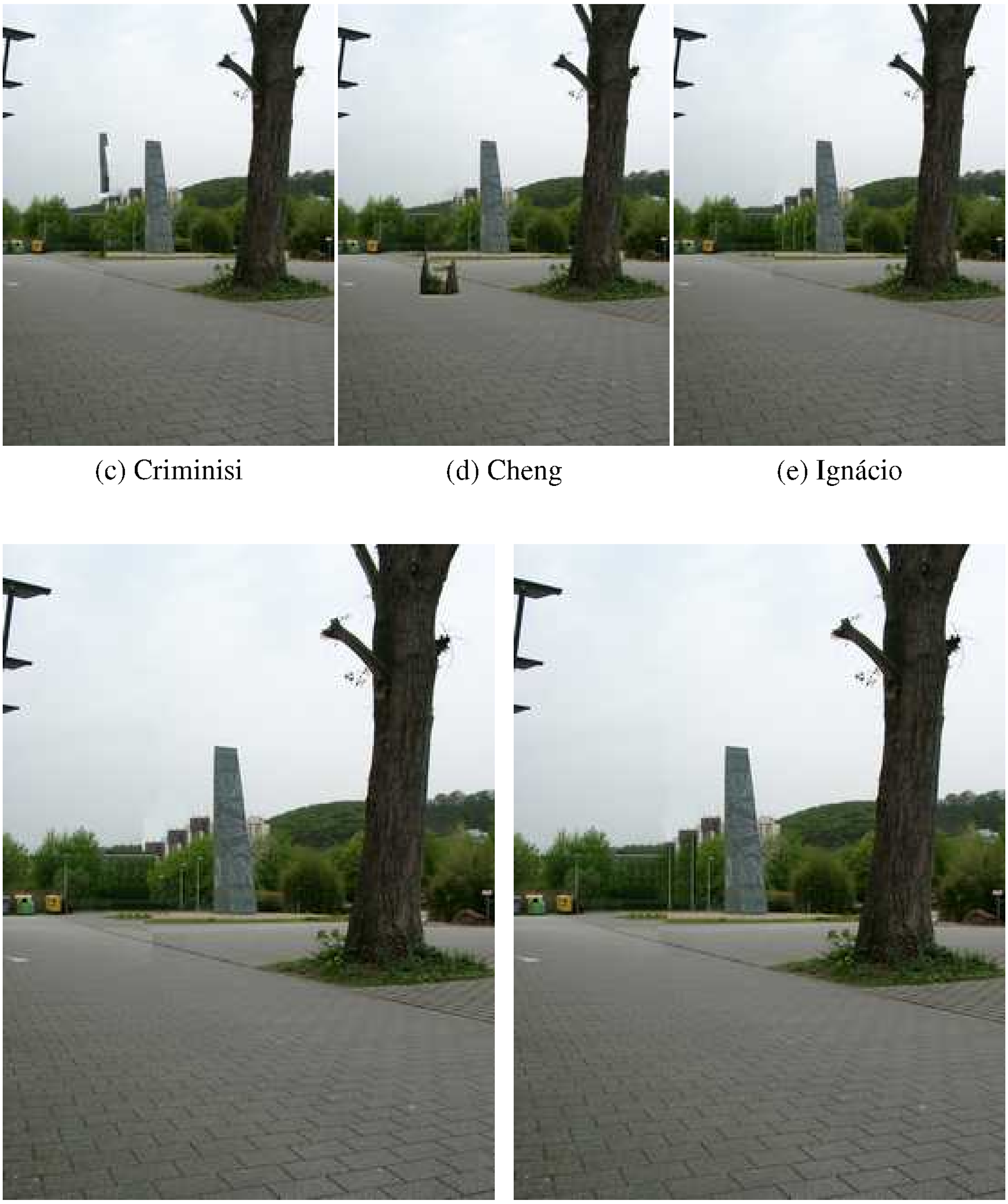
Images: Inpainting Results



(a) Additive: $\omega_S = 0.4, \omega_C = 0.5,$ $\theta = 6M, M = 9$, Plain

(b) Additive: $\omega_S = 0.5, \omega_C = 0.5,$ $\theta = 6M, M = 9, \sigma = 0.4$

Figure C.1: `Obelisk_Centre`



(c) Criminisi

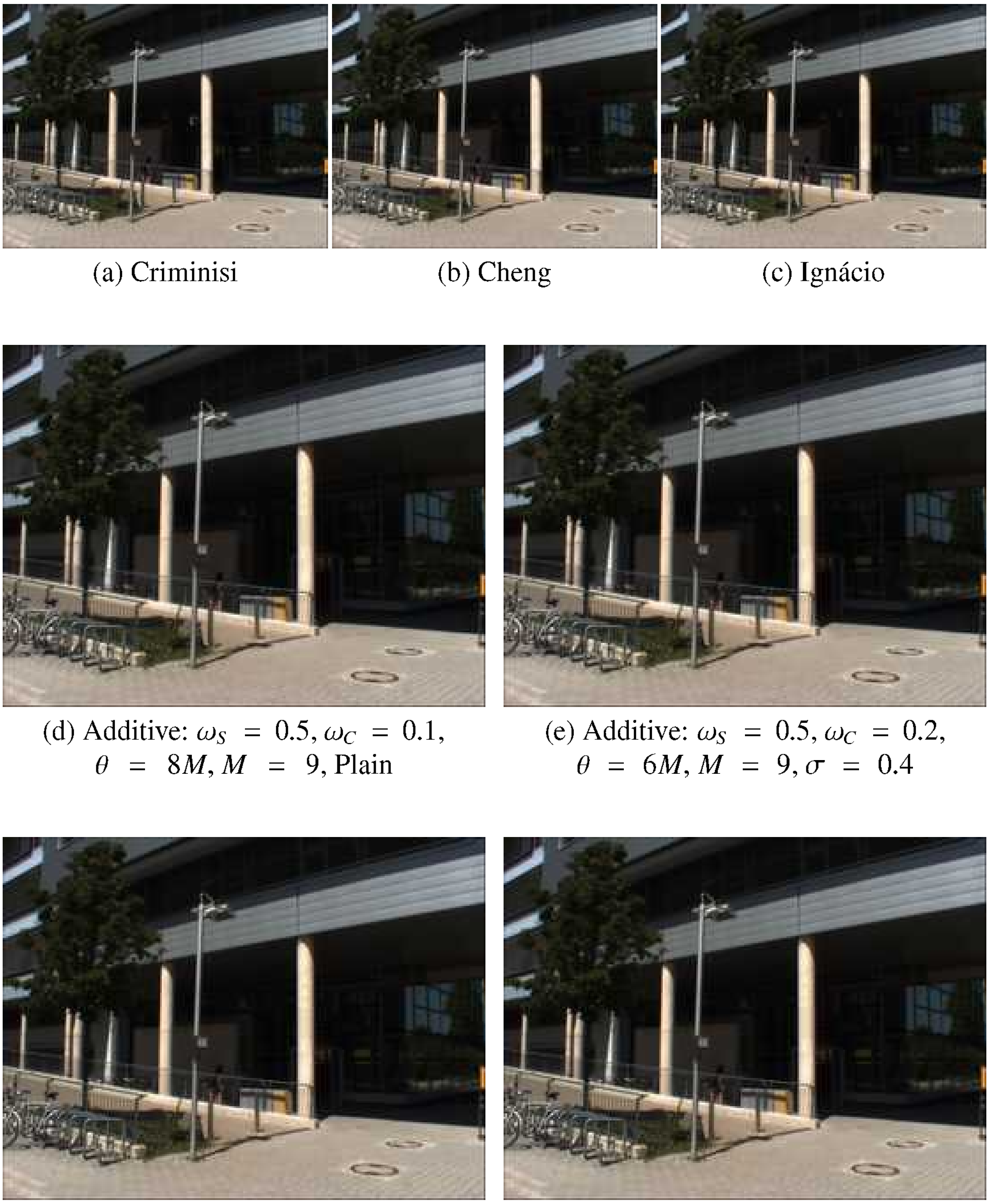
(d) Cheng

(e) Ignácio

(f) Mixed: $\omega_C = 0.8, \theta = 6M, M = 9$, Plain

(g) Mixed: $\omega_C = 0.5, \theta = 6M, M = 9, \sigma = 0.6$

Figure C.1: `Obelisk_Centre`



(a) Criminisi

(b) Cheng

(c) Ignácio

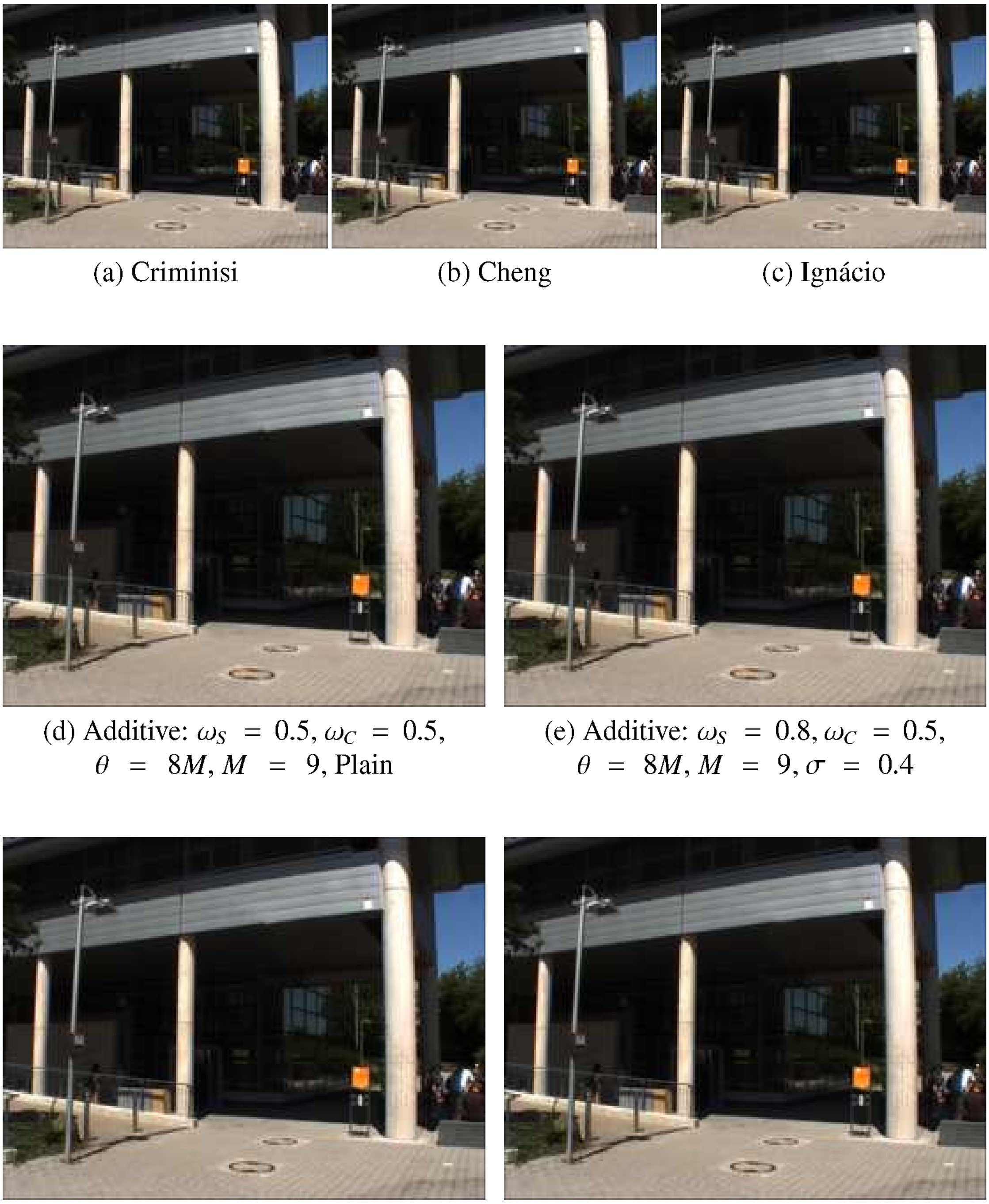
(d) Additive: $\omega_S = 0.5, \omega_C = 0.1$, $\theta = 8M, M = 9$, Plain

(e) Additive: $\omega_S = 0.5, \omega_C = 0.2$, $\theta = 6M, M = 9, \sigma = 0.4$

(f) Mixed: $\omega_C = 0.5$, $\theta = 6M, M = 9$, Plain

(g) Mixed: $\omega_C = 0.5$, $\theta = 6M, M = 9, \sigma = 0.6$

Figure C.2: `G_Entrance_Pillar`



(a) Criminisi

(b) Cheng

(c) Ignácio

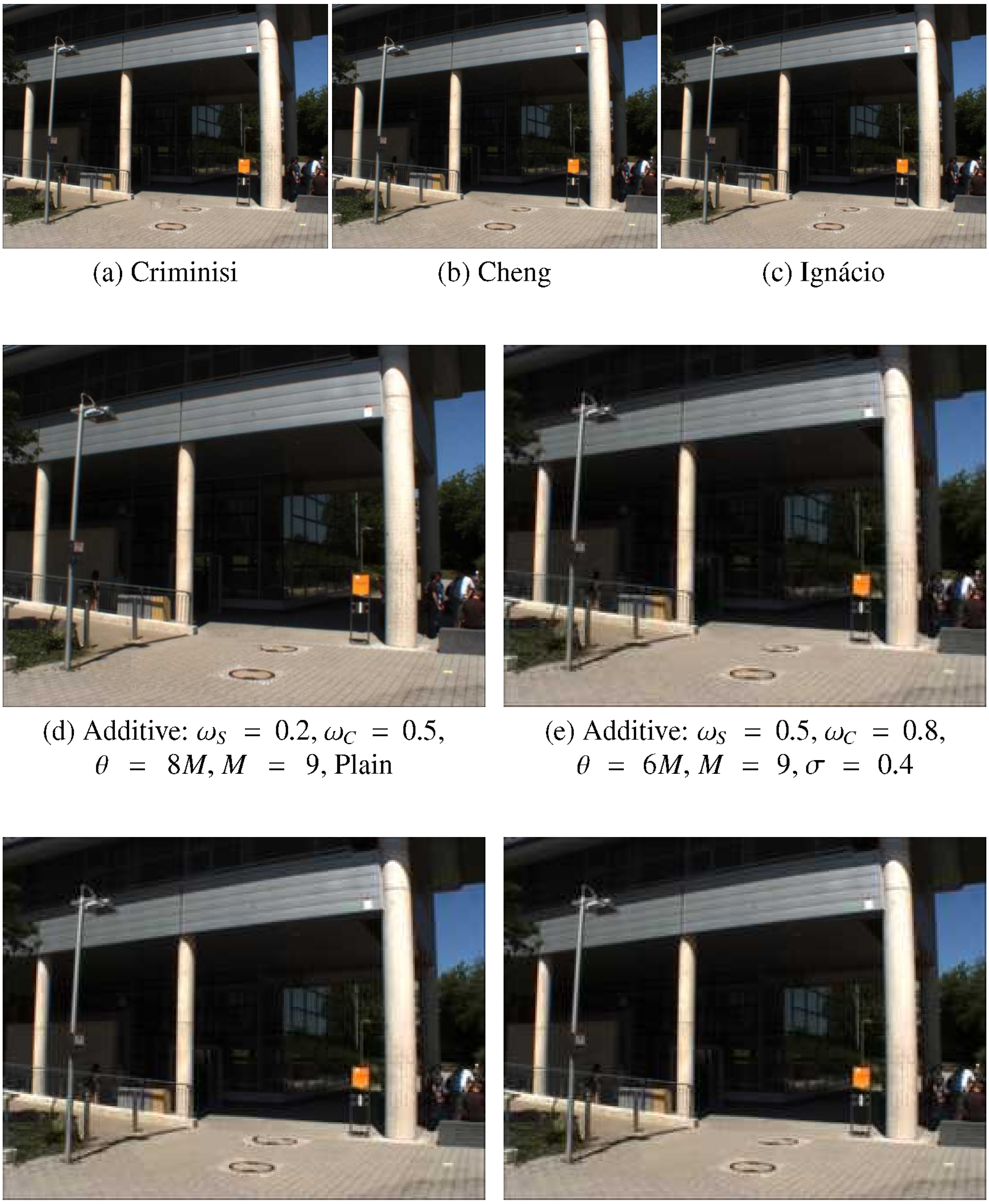
(d) Additive: $\omega_S = 0.5, \omega_C = 0.5$, $\theta = 8M, M = 9$, Plain

(e) Additive: $\omega_S = 0.8, \omega_C = 0.5$, $\theta = 8M, M = 9, \sigma = 0.4$

(f) Mixed: $\omega_C = 0.6$, $\theta = 6M, M = 9$, Plain

(g) Mixed: $\omega_C = 0.8$, $\theta = 6M, M = 9, \sigma = 0.4$

Figure C.3: `G_Entrance_Top`



(a) Criminisi

(b) Cheng

(c) Ignácio

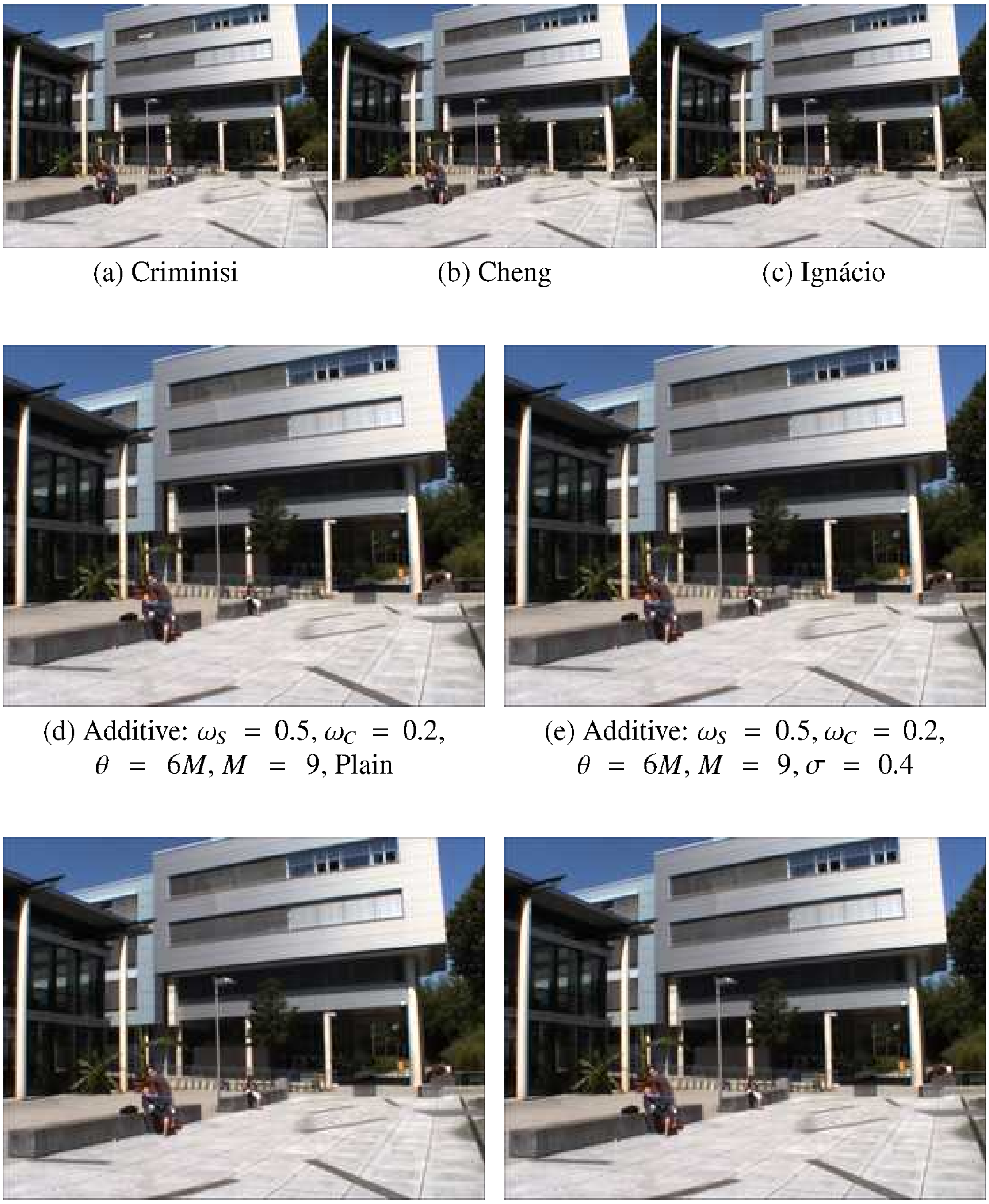
(d) Additive: $\omega_S = 0.2, \omega_C = 0.5,$ $\theta = 8M, M = 9,$ Plain

(e) Additive: $\omega_S = 0.5, \omega_C = 0.8,$ $\theta = 6M, M = 9, \sigma = 0.4$

(f) Mixed: $\omega_C = 0.8,$ $\theta = 6M, M = 9,$ Plain

(g) Mixed: $\omega_C = 0.8,$ $\theta = 6M, M = 9, \sigma = 0.4$

Figure C.4: `G_Entrance_2sq`



(a) Criminisi

(b) Cheng

(c) Ignácio

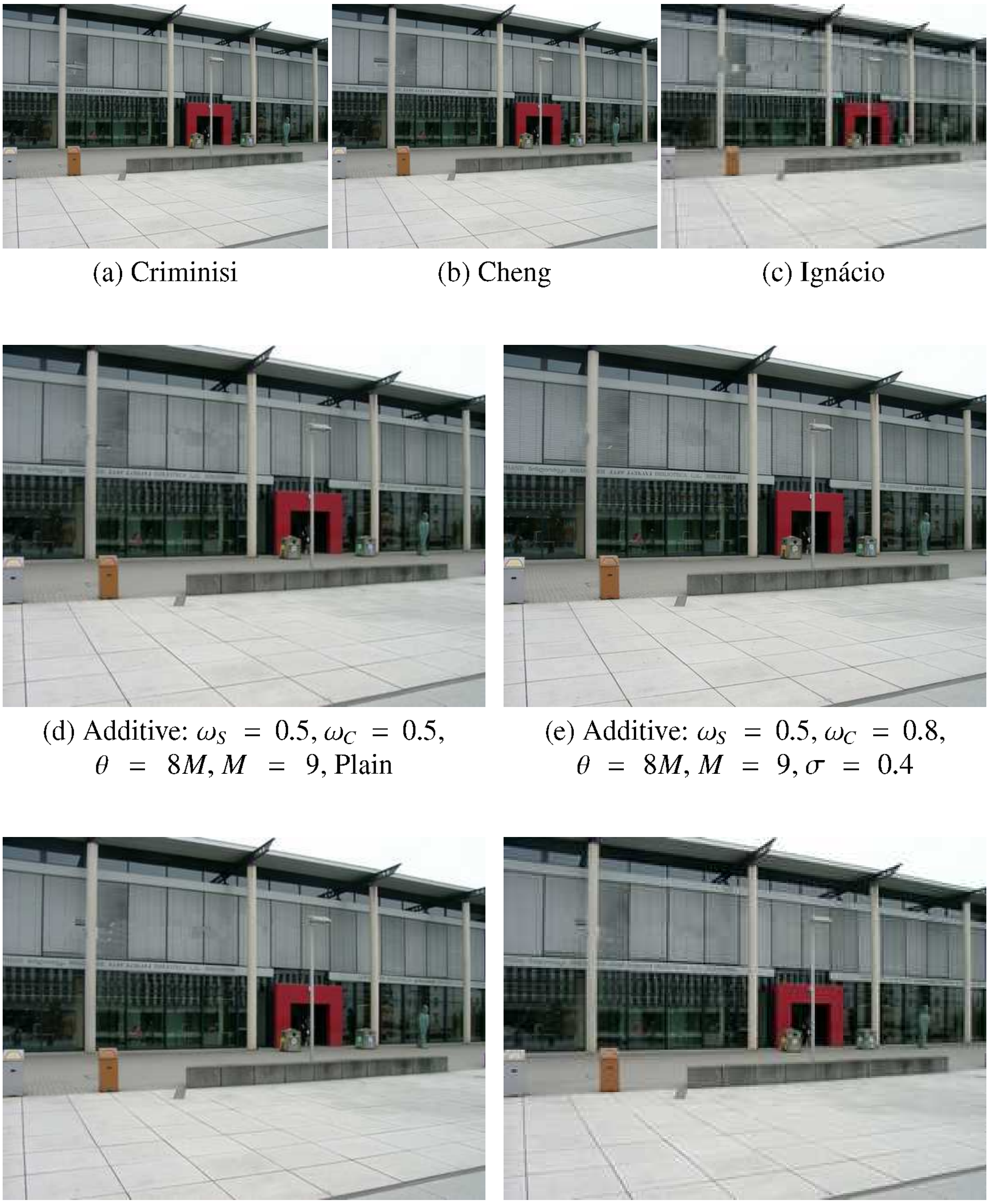
(d) Additive: $\omega_S = 0.5, \omega_C = 0.2,$ $\theta = 6M, M = 9,$ Plain

(e) Additive: $\omega_S = 0.5, \omega_C = 0.2,$ $\theta = 6M, M = 9, \sigma = 0.4$

(f) Mixed: $\omega_C = 0.5,$ $\theta = 6M, M = 9,$ Plain

(g) Mixed: $\omega_C = 0.8,$ $\theta = 6M, M = 9, \sigma = 0.4$

Figure C.5: `MikadoPlatz_TopLeft`



(a) Criminisi

(b) Cheng

(c) Ignácio

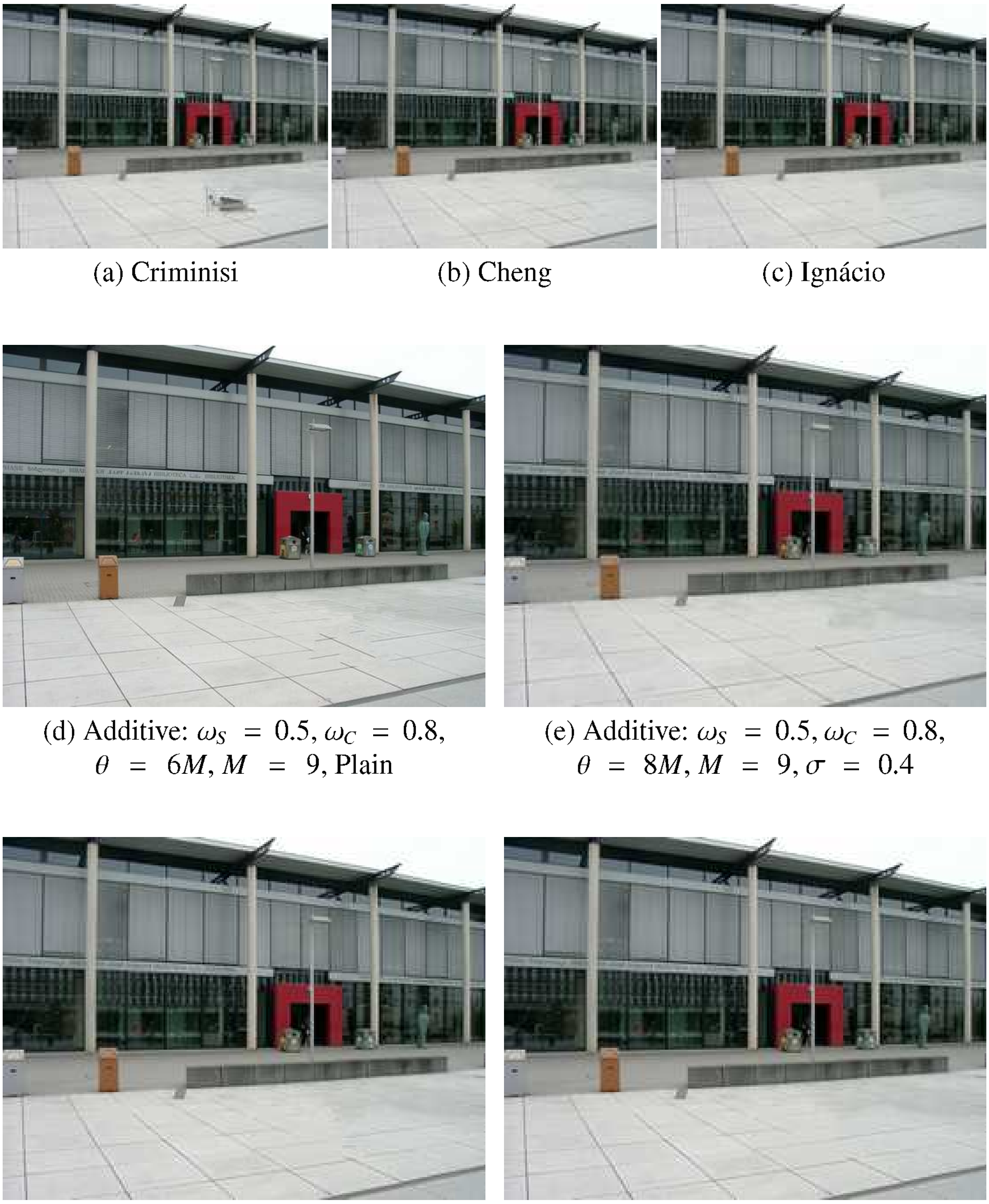
(d) Additive: $\omega_S = 0.5, \omega_C = 0.5$, $\theta = 8M, M = 9$, Plain

(e) Additive: $\omega_S = 0.5, \omega_C = 0.8$, $\theta = 8M, M = 9, \sigma = 0.4$

(f) Mixed: $\omega_C = 0.8$, $\theta = 6M, M = 9$, Plain

(g) Mixed: $\omega_C = 0.8$, $\theta = 6M, M = 9, \sigma = 0.4$

Figure C.6: `Lib_Front_HorizLong`



(a) Criminisi

(b) Cheng

(c) Ignácio

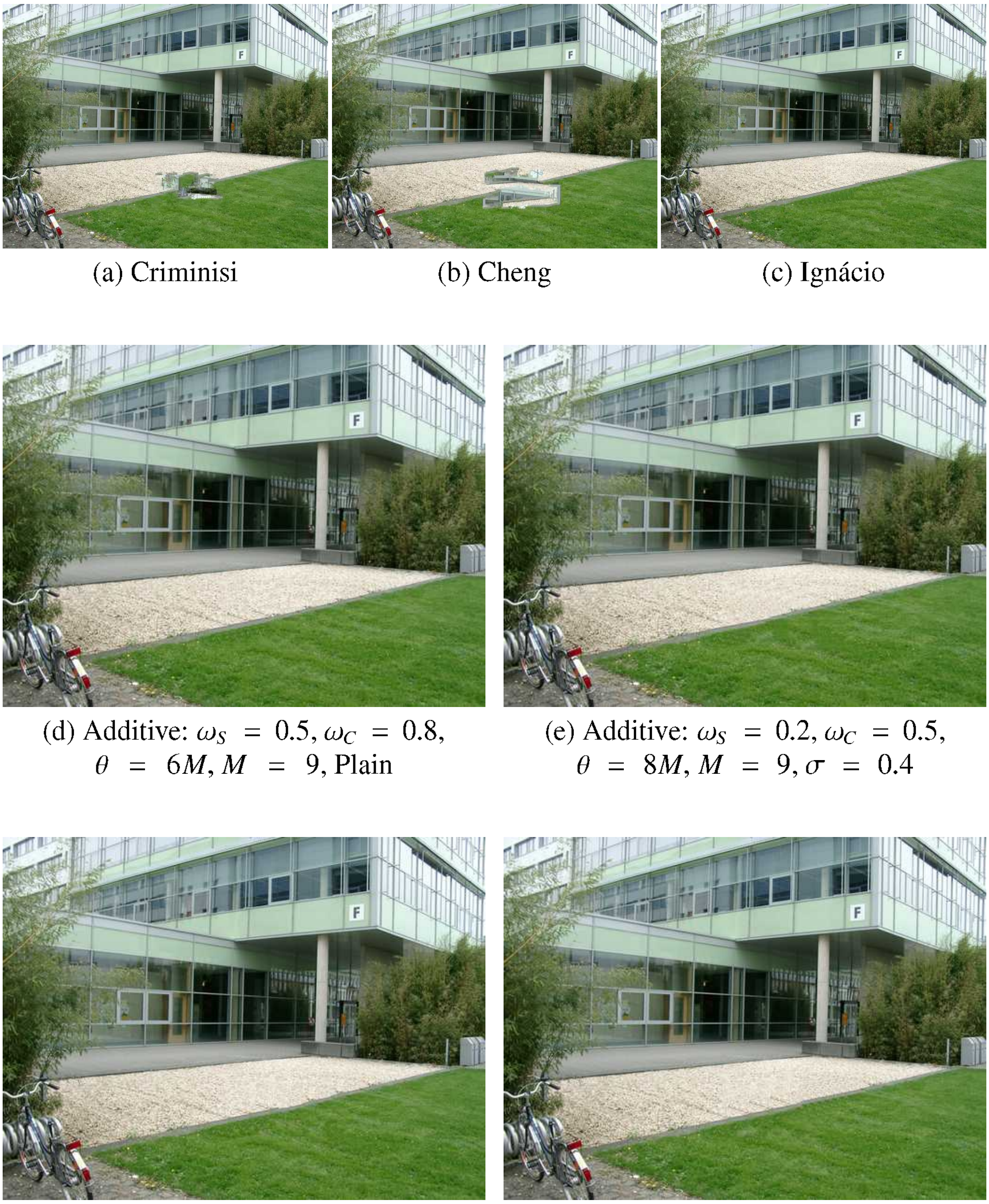
(d) Additive: $\omega_S = 0.5, \omega_C = 0.8,$ $\theta = 6M, M = 9,$ Plain

(e) Additive: $\omega_S = 0.5, \omega_C = 0.8,$ $\theta = 8M, M = 9, \sigma = 0.4$

(f) Mixed: $\omega_C = 0.4,$ $\theta = 6M, M = 9,$ Plain

(g) Mixed: $\omega_C = 0.4,$ $\theta = 6M, M = 9, \sigma = 0.4$

Figure C.7: `Lib_Front_RightBottom`



(a) Criminisi

(b) Cheng

(c) Ignácio

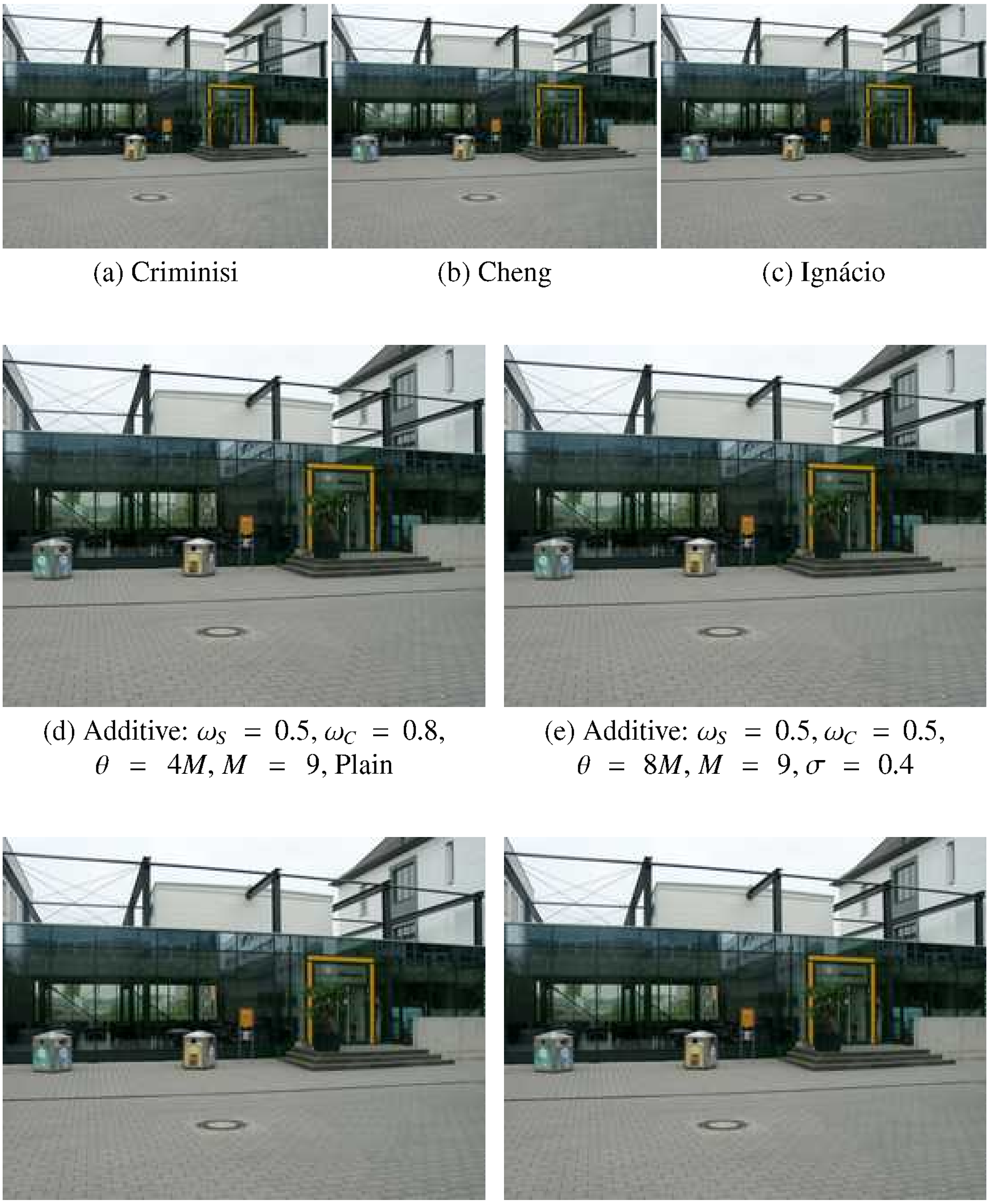
(d) Additive: $\omega_S = 0.5, \omega_C = 0.8, \theta = 6M, M = 9$, Plain

(e) Additive: $\omega_S = 0.2, \omega_C = 0.5, \theta = 8M, M = 9, \sigma = 0.4$

(f) Mixed: $\omega_C = 0.6, \theta = 6M, M = 9$, Plain

(g) Mixed: $\omega_C = 0.8, \theta = 6M, M = 9, \sigma = 0.4$

Figure C.8: `Gravel`



(a) Criminisi

(b) Cheng

(c) Ignácio

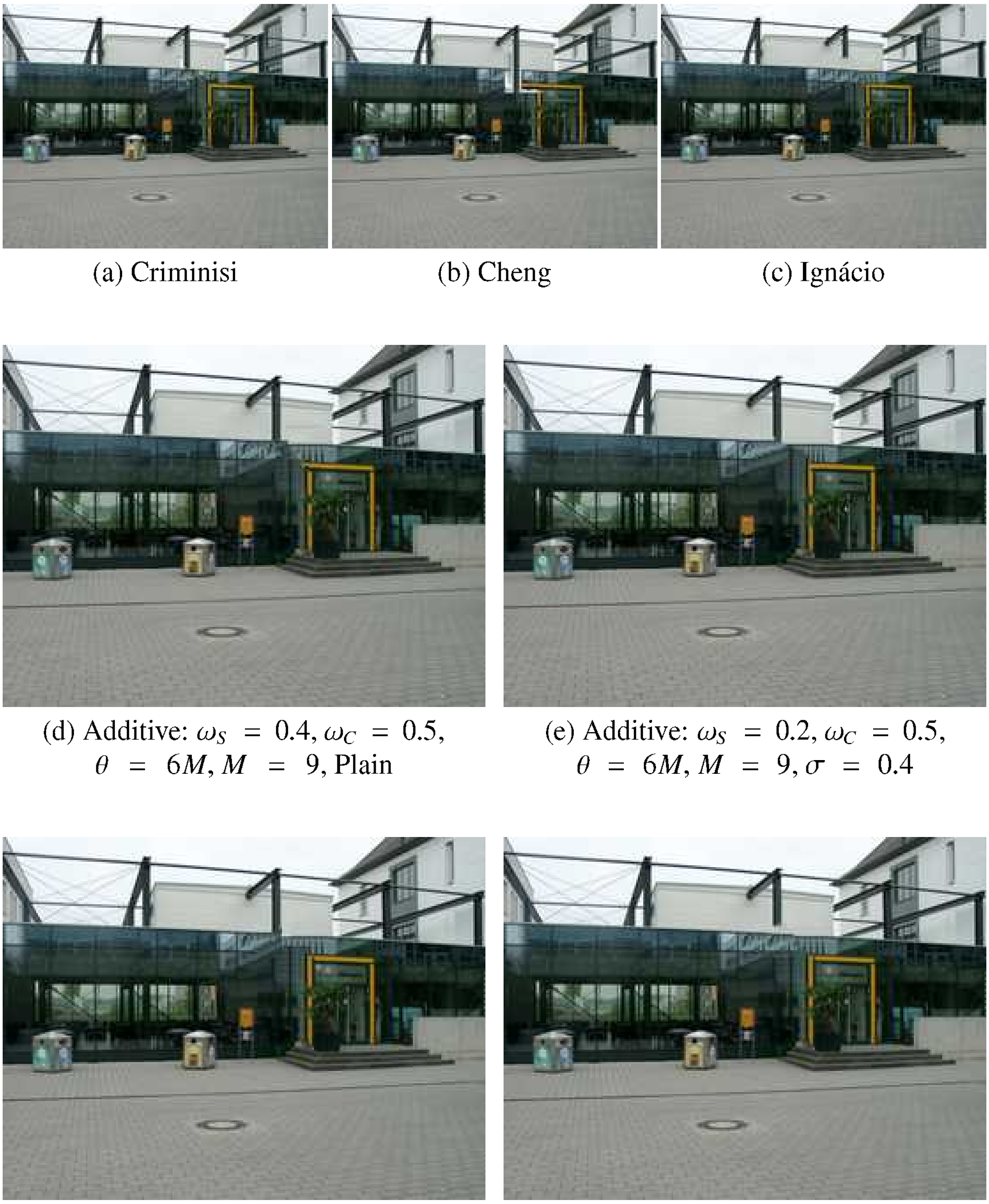
(d) Additive: $\omega_S = 0.5, \omega_C = 0.8$, $\theta = 4M, M = 9$, Plain

(e) Additive: $\omega_S = 0.5, \omega_C = 0.5$, $\theta = 8M, M = 9, \sigma = 0.4$

(f) Mixed: $\omega_C = 0.6$, $\theta = 6M, M = 9$, Plain

(g) Mixed: $\omega_C = 0.5$, $\theta = 6M, M = 9, \sigma = 0.4$

Figure C.9: `Menseria_LeftBottom`



(a) Criminisi

(b) Cheng

(c) Ignácio

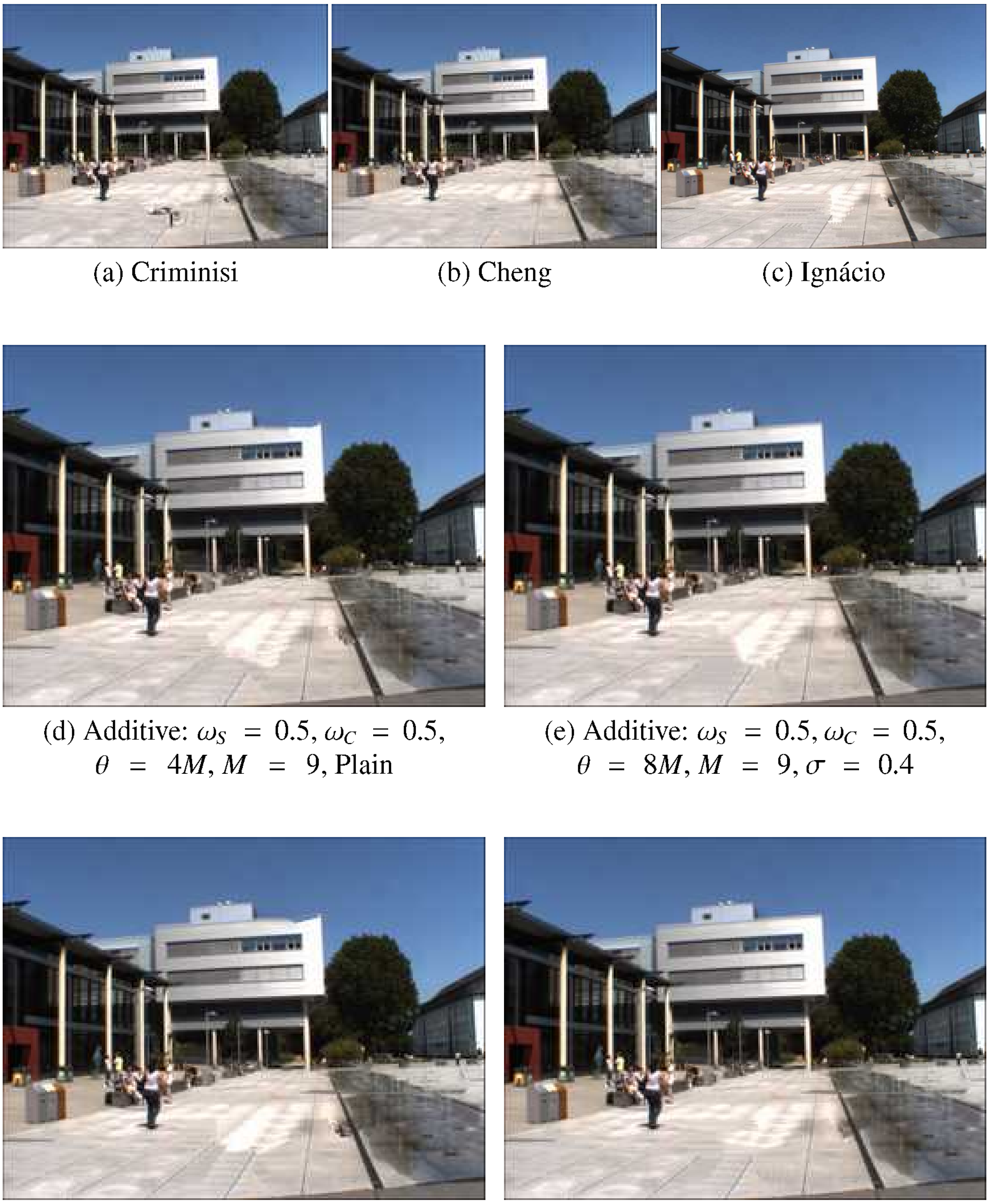
(d) Additive: $\omega_S = 0.4, \omega_C = 0.5,$
$\theta = 6M, M = 9,$ Plain

(e) Additive: $\omega_S = 0.2, \omega_C = 0.5,$
$\theta = 6M, M = 9, \sigma = 0.4$

(f) Mixed: $\omega_C = 0.5,$
$\theta = 3M, M = 9,$ Plain

(g) Mixed: $\omega_C = 0.2,$
$\theta = 6M, M = 9, \sigma = 0.4$

Figure C.10: `Menseria_Top`



(a) Criminisi

(b) Cheng

(c) Ignácio

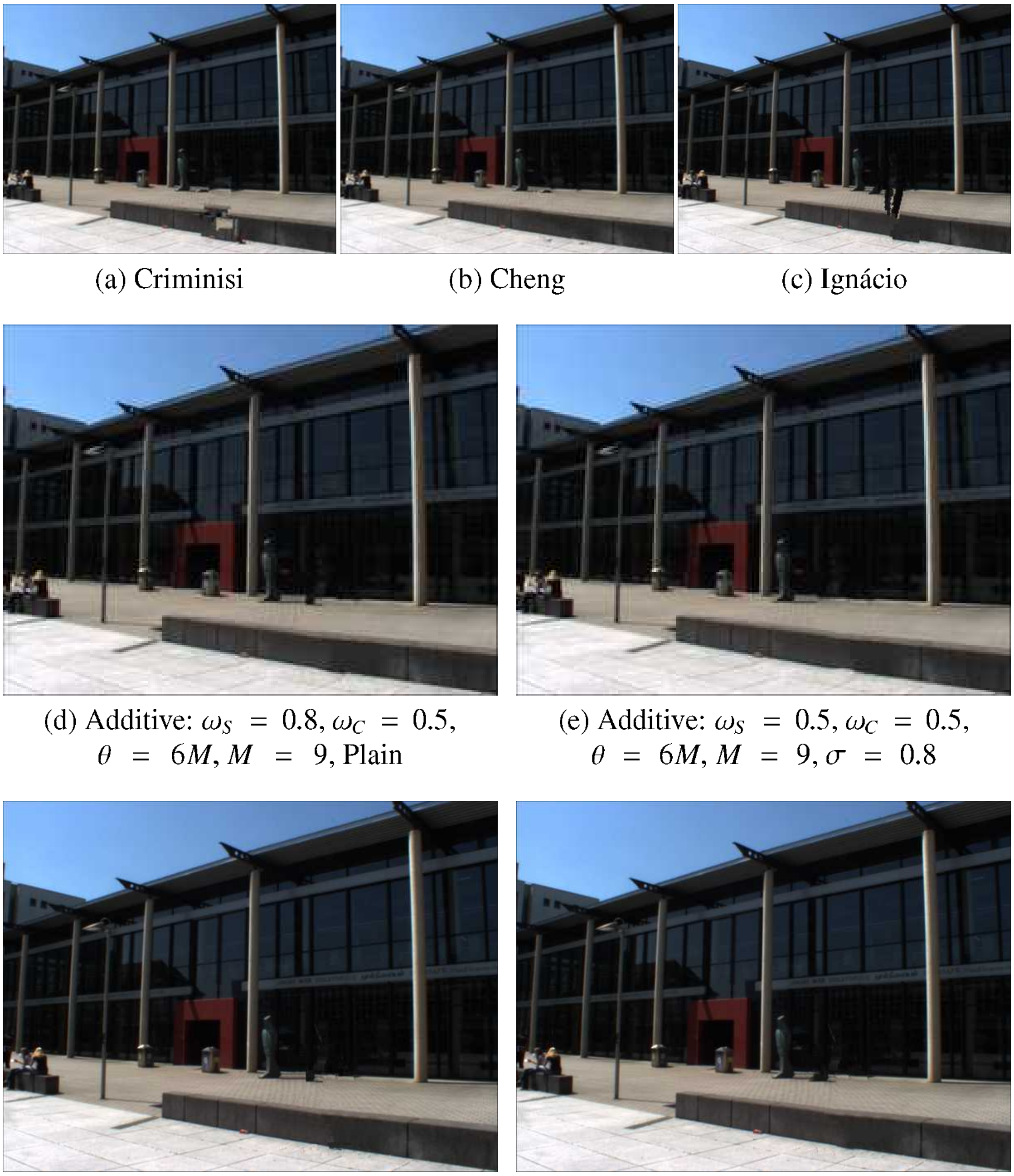
(d) Additive: $\omega_S = 0.5, \omega_C = 0.5$, $\theta = 4M, M = 9$, Plain

(e) Additive: $\omega_S = 0.5, \omega_C = 0.5$, $\theta = 8M, M = 9, \sigma = 0.4$

(f) Mixed: $\omega_C = 0.6$, $\theta = 3M, M = 9$, Plain

(g) Mixed: $\omega_C = 0.6$, $\theta = 6M, M = 9, \sigma = 0.4$

Figure C.11: `CampusWater`



(a) Criminisi

(b) Cheng

(c) Ignácio

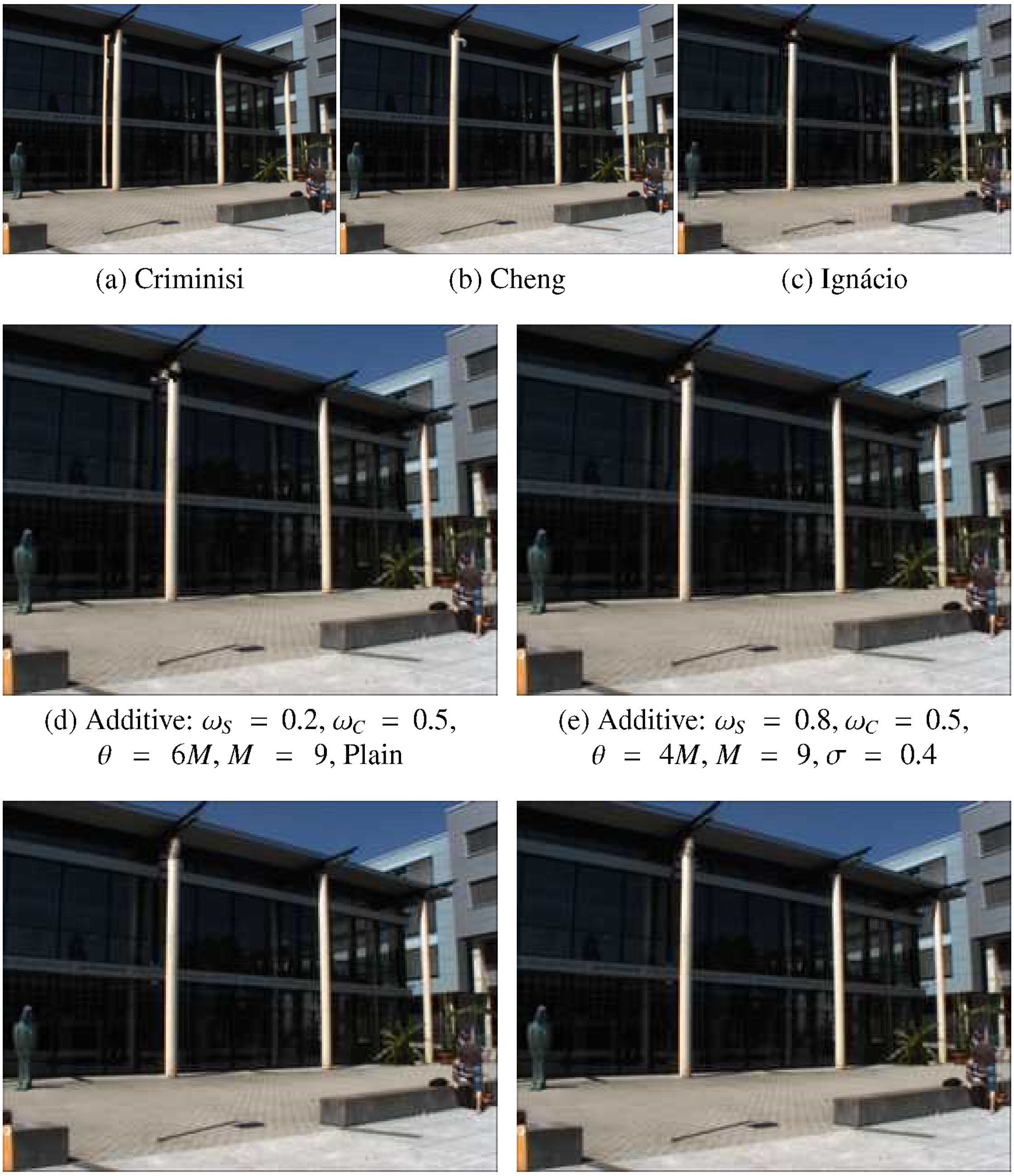
(d) Additive: $\omega_S = 0.8, \omega_C = 0.5$, $\theta = 6M, M = 9$, Plain

(e) Additive: $\omega_S = 0.5, \omega_C = 0.5$, $\theta = 6M, M = 9, \sigma = 0.8$

(f) Mixed: $\omega_C = 0.4$, $\theta = 6M, M = 9$, Plain

(g) Mixed: $\omega_C = 0.4$, $\theta = 6M, M = 9, \sigma = 0.4$

Figure C.12: `Bib_People`



(a) Criminisi

(b) Cheng

(c) Ignácio

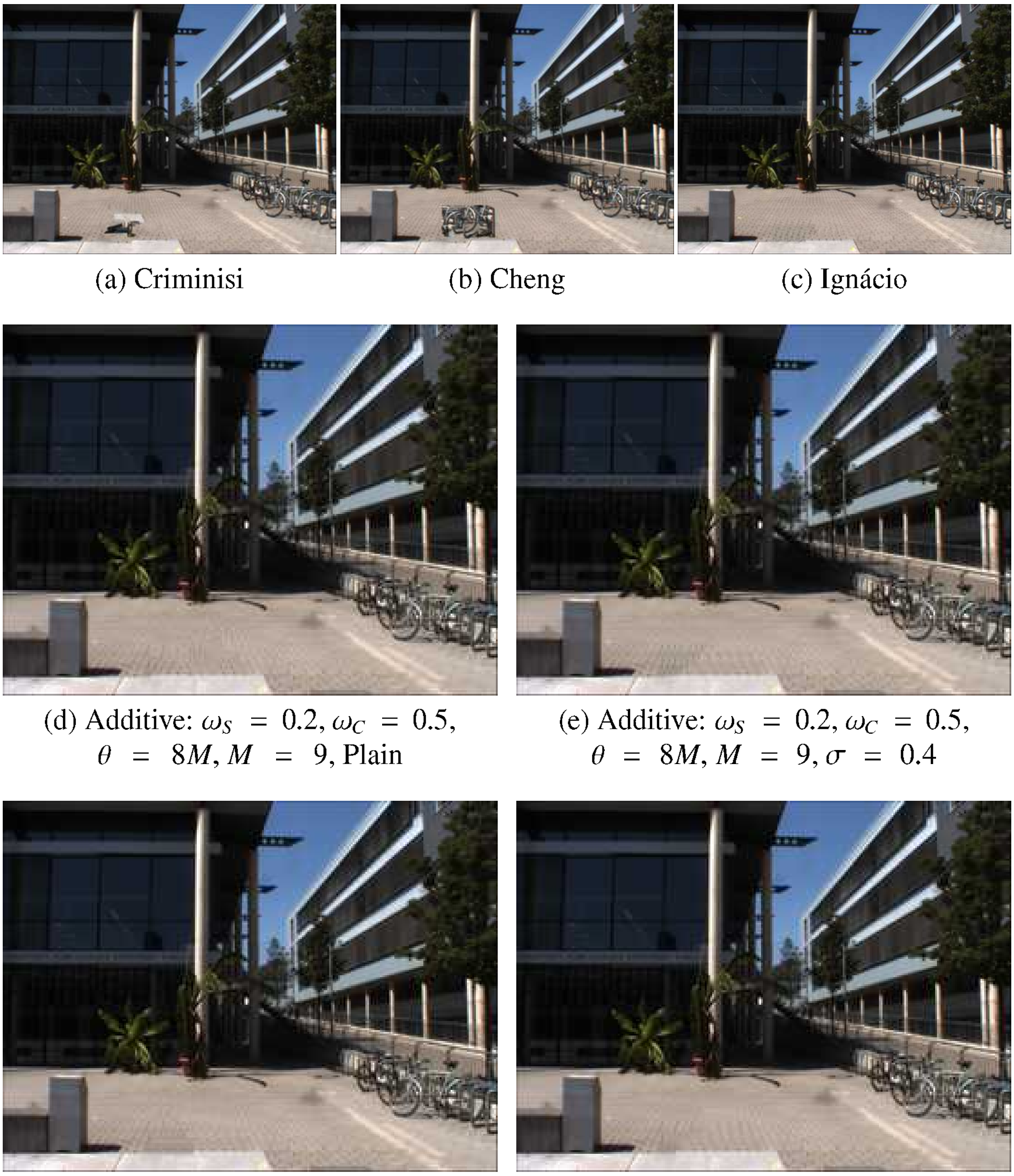
(d) Additive: $\omega_S = 0.2, \omega_C = 0.5$, $\theta = 6M, M = 9$, Plain

(e) Additive: $\omega_S = 0.8, \omega_C = 0.5$, $\theta = 4M, M = 9, \sigma = 0.4$

(f) Mixed: $\omega_C = 0.8$, $\theta = 3M, M = 9$, Plain

(g) Mixed: $\omega_C = 0.8$, $\theta = 6M, M = 9, \sigma = 0.4$

Figure C.13: `Bib_Lamppost`



(a) Criminisi

(b) Cheng

(c) Ignácio

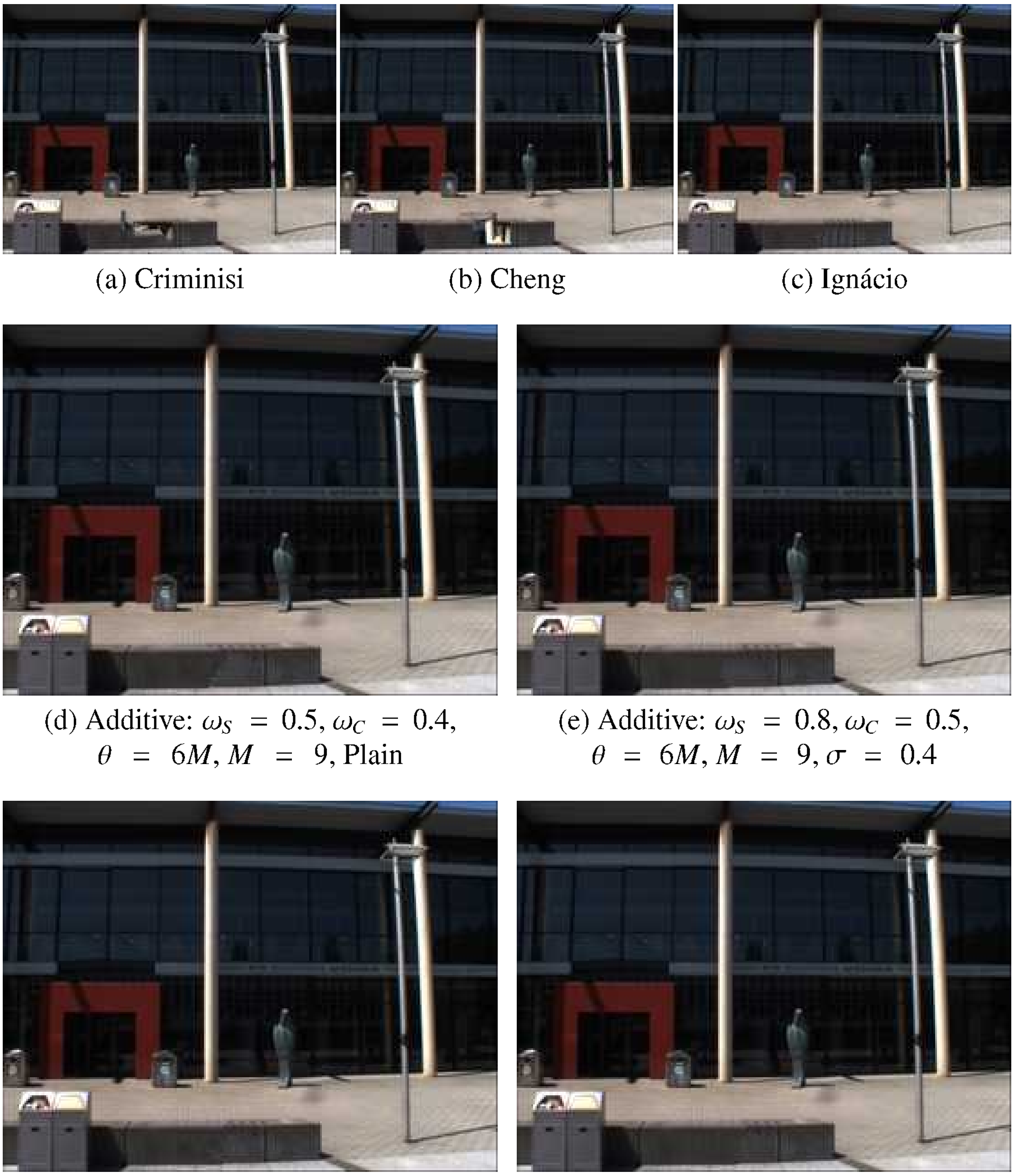
(d) Additive: $\omega_S = 0.2, \omega_C = 0.5,$
$\theta = 8M, M = 9$, Plain

(e) Additive: $\omega_S = 0.2, \omega_C = 0.5,$
$\theta = 8M, M = 9, \sigma = 0.4$

(f) Mixed: $\omega_C = 0.2,$
$\theta = 6M, M = 9$, Plain

(g) Mixed: $\omega_C = 0.8,$
$\theta = 6M, M = 9, \sigma = 0.4$

Figure C.14: `Bib_ColourChecker_BikeStand`



(a) Criminisi

(b) Cheng

(c) Ignácio

(d) Additive: $\omega_S = 0.5, \omega_C = 0.4,$ $\theta = 6M, M = 9,$ Plain

(e) Additive: $\omega_S = 0.8, \omega_C = 0.5,$ $\theta = 6M, M = 9, \sigma = 0.4$

(f) Mixed: $\omega_C = 0.8,$ $\theta = 6M, M = 9,$ Plain

(g) Mixed: $\omega_C = 0.8,$ $\theta = 6M, M = 9, \sigma = 0.4$

Figure C.15: `Bib_ColourChecker_Entrance`

References

[Add02] Paul S. Addison. *The illustrated wavelet transform handbook: introductory theory and applications in science, engineering, medicine and finance.* CRC Press, 2002.

[AM01] T. Aach and V. H. Metzler. Defect interpolation in digital radiography: how object- oriented transform coding helps. In *Proc. SPIE Vol. 4322, p. 824-835, Medical Imaging 2001: Image Processing, Milan Sonka; Kenneth M. Hanson; Eds.*, pages 824–835, 7 2001.

[BKK^{+}00] Jacob Beutel, Yongmin Kim, Harold L. Kundel, Steven C. Horii, and Richard L. Van Metter. *Handbook of Medical Imaging.* SPIE Press, 2000.

[Bra98] Carol Braverman. *Photoshop Retouching Handbook.* IDG Books Worldwide, 1998.

[BSCB00] Marcelo Bertalmio, Guillermo Sapiro, Vicent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of SIGGRAPH*, 2000.

[Bus03] David D. Busch. *Digital Retouching and Compositing.* Cengange Learning, 2003.

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

[CHL^{+}05] Wen-Huang Cheng, Chun-Wei Hsieh, Shen-Kai Lin, Chai-Wei Wang, and Ja-Ling Wu. Robust algorithm for exemplar-based image inpainting. In *Proceedings of International Conference on Computer Graphics, Imaging and Vision*, pages 64–69, 7 2005.

[CLLA06] Yan Chen, Qing Luan, Houqiang Li, and Oscar Au. Sketch-guided texture-based image inpainting. In *ICIP*, pages 1997–2000, 2006.

[CPT04] A. Criminisi, P. Perez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing*, 13(9), 9 2004.

[CSZ06] T. F. Chan, J. H. Shen, and H. M. Zhou. Total variation wavelet inpainting. *Journal cf Mathematical Imaging and Vision*, 25(1):107–125, 7 2006.

[DCOY03] Iddo Drori, Daniel Cohen-Or, and Hezy Yeshurum. Fragment-based image completion. In *Proceedings cf ACM SIGGRAPH 2003*, volume 22, pages 303 – 312, 7 2003.

[eGZW03] Cheng en Guo, Song-Chun Zhu, and Ying Nian Wu. Towards a mathematical theory of primal sketch and sketchability. *Computer Vision, IEEE International Conference on*, 2:1228, 2003.

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

[Fit08] Mark Fitzgerald. *Photoshop CS3 Restoration and Retouching Bible*. John Wiley and Sons, 2008.

[GS81] D. D. Garber and A. A. Sawchuk. Computational models for texture analysis and synthesis. In *Image Understanding Workshop*, pages 69–88, 1981.

[GW06] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 3 edition, 2006.

[Haa09] Judith Haas. *Analyse, Evaluation und Vergleich von Bildverarbeitungsbibliotheken aus Sicht der Scftwaretechnik*. 2009. Diplomarbeit, Universität Koblenz-Landau, Institut für Softwaretechnik.

[Ham98] Richard Wesley Hamming. *Digital Filters*. Courier Dover Publications, 1998.

[Har01] Paul Harrison. A non-hierarchical procedure for re-synthesis of complex textures. In *WSCG*, pages 190–197, 2001.

[HE08] James Hays and Alexei A. Efros. Scene completion using millions of photographs. *Commun. ACM*, 51(10), 2008.

[HT96] Anil N. Hirani and Takashi Totsuka. Combining frequency and spatial domain information for fast interactive image noise removal. In *SIGGRAPH*

'96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pages 269–276, New York, NY, USA, 1996. ACM.

[Hub98] Barbara Burke Hubbard. *The World According to Wavelets: The Story of a Mathematical Technique in the Making.* A. K. Peters, 1998.

[IJ07] U. A. Ignacio and C. R. Jung. Block-based image inpainting in the wavelet domain. *The Visual Computer*, 23(9-11), 9 2007.

[JT03] Jiaya Jia and Chi-Keung Tang. Image repairing: Robust image synthesis by adaptive nd tensor voting. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:643, 2003.

[KA08] Andreas Koschan and Mongi Abidi. *Digital Color Image Processing.* Wiley-Interscience, New York, NY, USA, 2008.

[Kin99] David King. *The Commissar Vanishes: The Falsification of Photographs and Art in Stalin's Russia.* Holt Paperbacks, 1999.

[KMFR95a] A. C. Kokaram, R. D. Morris, W. J. Fitzgerald, and P. J. W. Rayner. Detection of missing data in image sequences. *IEEE Trans. Image Processing*, 4(11):1496–1508, 11 1995.

[KMFR95b] A. C. Kokaram, R. D. Morris, W. J. Fitzgerald, and P. J. W. Rayner. Interpolation of missing data in image sequences. *IEEE Trans. Image Processing*, 4(11):1509–1519, 11 1995.

[LLH04] Yanxi Liu, Wen-Chieh Lin, and James Hays. Near-regular texture analysis and manipulation. *ACM Transactions on Graphics*, 23(3):368–376, 8 2004.

[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.

[Mal89] Stèphane Mallat. A theory of multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:674–693, 1989.

[Mey01] Yves Meyer. *Oscillating Patterns in Image Processing and Nonlinear Evolution.* American Mathematical Society, 2001.

[OH07] Michael T. Orchard and Gang Hua. Image inpainting based on geometrical modeling of complex wavelet coefficients. In *ICIP 2007*, 2007.

[OT06] A. Oliva and A. Torralba. Building the gist of a scene: the role of global image features in recognition. *Progress in brain research*, 155:23–36, 2006.

[Pap75] A. Papoulis. A new algorithm in spectral analysis and band- limited extrapolation. *IEEE Trans. on CAS*, CAS-22:735–742, 1975.

[PGB03] P. Perez, Michel Gangnet, and Andrew Blake. Poisson image editing. pages 313–318, 2003.

[PH03] Dietrich Paulus and Joachim Hornegger. *Applied pattern recognition: A practical introduction to image and speech processing in C++*. Advanced Studies in Computer Science. Vieweg, Braunschweig, 4 edition, 2003.

[PM90] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.

[PS03] Kedar A. Patwardhan and Guillermo Sapiro. Projection based image and video inpainting using wavelets. In *Image Processing, 2003*, 9 2003.

[PSB05] Kedar A. Patwardhan, Guillermo Sapiro, and Marcelo Bertalmio. Video inpainting of occluding and occluded objects. In *ICIP (2)*, pages 69–72, 2005.

[PSB07] K. A. Patwardhan, G. Sapiro, and Marcelo Bertalmio. Video inpainting under constrained camera motion. *IEEE Trans. Image Processing*, 16(2):545–553, 2 2007.

[PTN98] Luiz Pessoa, Evan Thompson, and Alva Noeuml;. Finding out about filling-in: A guide to perceptual completion for visual science and the philosophy of perception. *Behavioral and Brain Sciences*, 21(06):723–748, 1998.

[RFMT03] M. A. Rubin, W. T. Freeman, K. P. Murphy, and A. Torralba. Context-based vision system for place and object recognition. In *MIT AIM*, 2003.

[ROF92] L. I. Rudin, S. J. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60:259–268, 1992.

[RRS02] Shantanu D. Rane, Jeremiah Remus, and Guillermo Sapiro. Wavelet-domain reconstruction of lost blocks in wireless image transmission and packet-switched networks. In *IEEE ICIP*, pages 309–312, 2002.

[SH98] Stephen J. Sangwine and R. E. N. Horne. *The Colour Image Processing Handbook.* Chapman Hall, London, 1998.

[SIA90] Roland Sottek, Klaus Illgner, and Til Aach. An efficient approach to extrapolation and spectral analysis of discrete signals. In *ASST '90: ASST '90, 7. Aachener Symposium für Signaltheorie: Modellgestützte Signalverarbeitung*, pages 103–108, London, UK, 1990. Springer-Verlag.

[SYJS05] Jian Sun, Lu Yuan, Jiaya Jia, and Heung-Yeung Shum. Image completion with structure propagation. In *International Conference on Computer Graphics and Interactive Techniques*, pages 861 – 868, 2005.

[TWJ05] Leandro Tonietto, Marcelo Walter, and Cláudio Rosito Jung. Patch-based texture synthesis using wavelets. In *SIBGRAPI*, pages 383–389, 2005.

[TWJ06] Leandro Tonietto, Marcelo Walter, and Cláudio Rosito Jung. A randomized approach for patch-based texture synthesis using wavelets. *Comput. Graph. Forum*, 25(4):675–684, 2006.

[VO03] L. A. Vese and S. J. Osher. Modeling textures with total variation minimization and oscillating patterns in image processing. *J. Sci. Computing*, 19:553–572, 2003.

[Wal08] James S Walker. *A Primer on Wavelets and Their Scientific Applications.* Chapman & Hall/CRC, 2008.

[WBTC05] M. M. Wilczkowiak, G. J. Brostow, B. J. Tordoff, and R. Cipolla. Hole filling through photomontage. In *BMVC*, pages xx–yy, 2005.

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

[WO08] Alexander Wong and Jeff Orchard. A nonlocal-means approach to exemplar-based inpainting. In *IEEE Transactions on Image Processing*, 2008.

Internet Ressources

[1] Sooraj Bhat [online]. 2004 [cited 9 April 2009]. Available from World Wide Web: `http://www.cc.gatech.edu/~sooraj/inpainting`. Master thesis of Sooraj Bhat implementing the Criminisi approach in MATLAB®. It provides source code and results.

[2] JPEG 2000 Committee [online]. 2009 [cited 11 April 2009]. Available from World Wide Web: `http://www.jpeg.org/jpeg2000`. Official website of the JPEG Committee on their JPEG 2000 image compression standard.

[3] Kerstin Falkowski [online, cited 06 May 2009]. Available from World Wide Web: `http://www.uni-koblenz.de/FB4/Contrib/ER`. Former website of the project *Enhanced Reality* at the University of Koblenz-Landau now split up into several subprojects.

[4] Paul Harrison [online]. 2009 [cited 3 April 2009]. Available from World Wide Web: `http://www.logarithmic.net/pfh/resynthesizer`. Resynthesizer Plugin for *The GIMP!* developed by Paul Harrison.

[5] Adobe Systems Inc. [online]. 2009 [cited 3 April 2009]. Available from World Wide Web: `http://www.adobe.com/products/photoshop/family`. Digital image processing application.

[6] ACM SIGGRAPH [online, cited 14 May 2009]. Available from World Wide Web: `http://www.siggraph.org/s2000`. Website of the SIGGRAPH conference in 2000.

[7] The GIMP Team [online]. 2009 [cited 3 April 2009]. Available from World Wide Web: `http://www.gimp.org`. Open-source image processing application.

[8] Inc. The MathWorks [online]. 2009 [cited 23 April 2009]. Available from World Wide Web: `http://www.mathworks.com/products/matlab`. Official website for MATLAB®, a high-level language and interactive environement for technical computing.

[9] Yahoo! Inc. [online]. 2009 [cited 3 April 2009]. Available from World Wide Web: `http://www.flickr.com`. Photo sharing and community website.