

Towards object-based image editing

Citation for published version (APA):

Kanters, F. M. W. (2007). *Towards object-based image editing*. [Phd Thesis 1 (Research TU/e / Graduation TU/e), Biomedical Engineering]. Technische Universiteit Eindhoven. <https://doi.org/10.6100/IR617002>

DOI:

[10.6100/IR617002](https://doi.org/10.6100/IR617002)

Document status and date:

Published: 01/01/2007

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

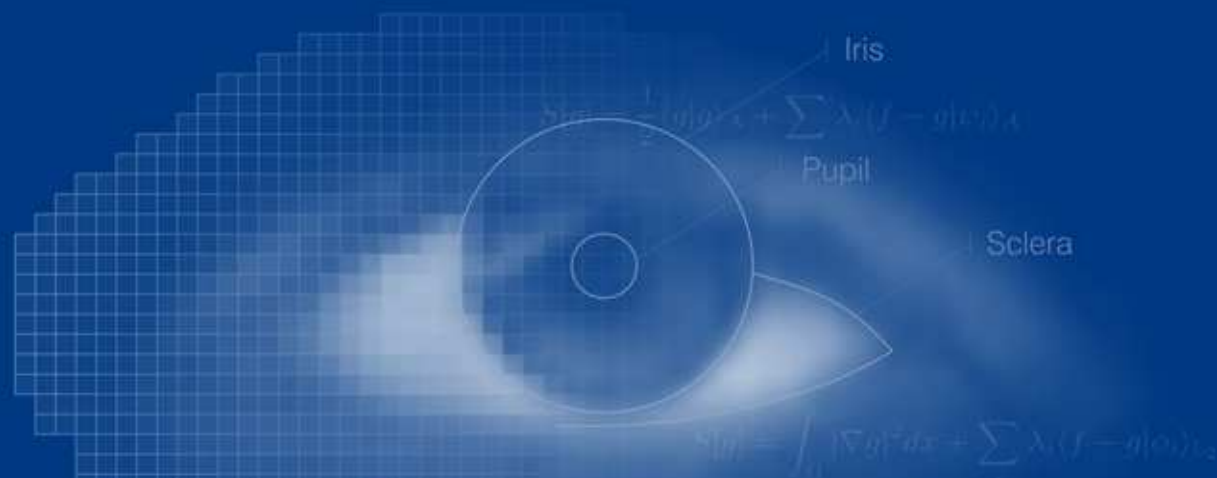
www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.



Towards Object-based Image Editing

$$S[g] = \frac{1}{2} \int_{\Omega} |g|^2 dx + \int_{\Omega} |\nabla g|^2 dx + \sum \lambda_i \lambda_i (f - g(o_i))_{L_2}$$

$$A = -\gamma \sqrt{-\Delta}$$

$$\begin{cases} \frac{\partial}{\partial s} u = \Delta u \\ \lim_{s \rightarrow 0} u(\cdot, s) = \varphi(\cdot) \end{cases}$$

$$\int_0^1 \sum_{n=0}^{\infty} (-1)^n \frac{\rho^{2n}}{n!} d\rho$$

$$\int_0^1 u(\alpha) \sum_{n=0}^{\infty} (-1)^n \frac{\rho^{2n}}{n!} d\alpha + (1 - u(\alpha)) \sum_{n=0}^{\infty} (-1)^n \frac{\rho^{2n}}{n!} d\alpha$$

$$\lim_{s \rightarrow 0} u(\cdot, s) = \varphi(\cdot) \text{ in } L_1(\mathbb{R}^N) \text{ sense } \quad \alpha \in (0, 1)$$

Towards Object-based Image Editing

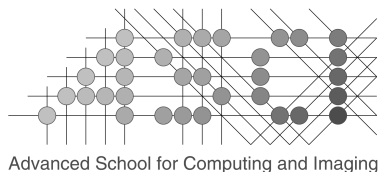
Colophon

This thesis was typeset by the author using $\text{\LaTeX}2_{\epsilon}$. Images were included formatted as Encapsulated Postscript, and represented in greyscale. The output was converted to PDF and transferred to film for printing.

About the cover

The front and back covers are designed by the author.

The top part of the front cover represents the transition from pixel based image editing towards object-based image editing. The bottom part shows the representation that is used in this thesis to edit images. In the background some of the formulas that are used in this thesis are sketched.



This work was carried out in the ASCI graduate school.
ASCI dissertation series number 135.

This work is part of the DSSCV project supported by the IST Programme of the European Union (IST-2001-35443).

A catalogue record is available from the Library Eindhoven University of Technology

ISBN-10: 90-386-2649-5

ISBN-13: 978-90-386-2649-9

Printed by PrintPartners Ipskamp, Enschede, The Netherlands

© 2006 F.M.W. Kanters, Beek en Donk, The Netherlands, unless stated otherwise on chapter front pages. All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without permission in writing from the copyright owner.

Towards Object-based Image Editing

Proefschrift

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de
Rector Magnificus, prof.dr.ir. C.J. van Duijn, voor een
commissie aangewezen door het College voor
Promoties in het openbaar te verdedigen
op dinsdag 27 februari 2007 om 16.00 uur

door

Franciscus Martinus Wilhelmus Kanters

geboren te Beek en Donk

Dit proefschrift is goedgekeurd door de promotoren:

prof.dr.ir. B.M. ter Haar Romeny

Copromotor:

dr. L.M.J. Florack

Contents

Colophon	ii
Contents	v
1 Introduction	1
2 Exploring Scale Space	5
2.1 Introduction	6
2.2 α -Scale Spaces	6
2.3 Approximation of the α -kernel in the spatial domain	11
2.3.1 Evaluation of the approximated kernel	13
2.3.2 Spatial derivatives of the approximated kernel	15
2.4 ScaleSpaceViz	15
2.4.1 Interest points	19
2.4.2 Approximated α -scale space kernel	20
2.4.3 Stability of scale space interest points	20
2.5 Conclusion and discussion	26
3 Image Reconstruction from a Sparse Pointset	27
3.1 Introduction	28
3.2 Image reconstruction algorithms	28
3.2.1 Standard linear reconstruction	28
3.2.2 Brownian reconstruction	31
3.2.3 Sobolev type inner product reconstruction	33
3.3 Evaluation of the reconstruction algorithms	40
3.3.1 Evaluation of the γ -parameter	41
3.3.2 Evaluation of the SVD settings	41
3.3.3 Evaluation of the number of reconstruction points	43
3.3.4 Evaluation of the reconstruction order	46
3.3.5 Stability with respect to noise on the position of the points	46
3.3.6 Stability with respect to noise on the features	46
3.3.7 Comparison of the proposed algorithms	49
3.4 Conclusion and discussion	49
4 Image Quality Measures	53
4.1 Introduction	54
4.2 Objective image quality measures	54
4.2.1 Generic mathematical measures	55
4.2.2 Multi-Scale Differential Error (MSDE)	65
4.2.3 Human Visual System based measures	68
4.3 Experiments	68

4.3.1	Setup	68
4.3.2	Evaluation	69
4.4	Conclusion and discussion	76
5	Scale Space Interest Points	79
5.1	Introduction	80
5.2	Interest points	80
5.2.1	Blobs	81
5.2.2	Corner points	83
5.2.3	Edge points	86
5.2.4	Ridge points	87
5.2.5	Top points or bifurcation points	87
5.2.6	Scale space saddle points	89
5.2.7	Scale adapted Hessian-Laplace points	93
5.2.8	Scale adapted Harris-Laplace points	93
5.3	Reconstruction experiments	95
5.3.1	Comparison of scale space interest points for reconstruction	95
5.3.2	Stability of the interest points	99
5.4	Combining interest points using canonical sets	99
5.4.1	Introduction	99
5.4.2	Canonical set framework	100
5.4.3	Incorporating distance lower-bound constraints	102
5.4.4	Experiments	104
5.5	Conclusion and discussion	110
6	Object-based Image Editing using Reconstruction from Scale Space Interest Points	113
6.1	Introduction	114
6.2	Proof of concept	115
6.3	A look into the future	120
6.3.1	Digital photo editing	120
6.3.2	Photo synthesis	122
6.3.3	Company advertisement	122
6.3.4	Animation and Morphing	123
6.3.5	Medical applications	123
6.4	Conclusion and discussion	124
7	Summary and Future Research	127
7.1	Summary	128
7.2	Future research	129
A	Error Measure Results	131
A.1	Introduction	132
A.2	Image Error Results	133

B Ordered Test Images	167
C Interest Points of Test Images	169
D Reconstructions from Interest Points of Test Images	183
E Stability Results of Interest Points from Test Images	209
Bibliography	215
Samenvatting (Summary in Dutch)	225
List of Publications	227
Acknowledgements	229
Curriculum Vitae	231

1

Introduction

Since the introduction of digital imaging sensors such as used in digital (video) cameras and medical imaging devices, digital image processing has been - and still is - a fascinating topic for many researchers around the world. One specific topic which is the focus of this thesis is image editing, where the content of a digital image is manipulated resulting in a different digital image. More specifically, we are interested in editing *objects* in the image. The word *editing* in this context is very broad, for example removing a person from a digital picture, replacing apples in a scene with pears or attenuating ribs in a CT image of the lungs to make lung nodules better visible. Nowadays, these operations can be performed using fast computers and much human interaction, but no automatic computer algorithm exists that can perform such tasks. The main reason is that most image processing algorithms are based on direct manipulation of the pixel data, while the pixel representation of an image is not suitable for tasks at an object level. In this thesis a framework is presented based on a different representation of images that is better suitable for editing images on object level, which is one more step towards object-based image editing.

Scale is an essential parameter in computer vision, since it is an immediate consequence of the integration process of the *measurements*. Observations are always done by integrating some physical property with a measurement device, for example integration (over a spatial area and a time frame) of reflected light intensity of an object with a CCD sensor in a digital camera, or a photoreceptor in our eye. The range of possibilities to observe certain sizes of objects is bounded on two sides: there is a minimal size, about the size of the smallest aperture, and there is a maximal size, about the size of the whole detector array. The *front-end* of the human visual system (the very first few layers of the visual system) is able to “detect” at different apertures. A good example for this necessity is an image mosaic as is shown in figure 1.1. To see the details of each patch we use a small aperture, which prevents us to see the global structure. At a large aperture we are able to see the global structure, but not the details of each patch. Objects are only detected at a certain *scale*.

Since one can not know a priori at which scales objects are present in a scene, it is necessary to detect at multiple scales simultaneously. Scale space theory is the theory of apertures, through which we and machines observe the world. For computer vision systems, the notion of aperture can be introduced as *blurring* the high resolution image with a kernel of a certain width. In 1962, Taizo Iijima derived the Gaussian kernel for this purpose from a set of basic axioms in a Japanese paper [61]. Later, papers about linear or Gaussian scale space followed from various other authors, including Witkin [141] and Koenderink [80] in the eighties. A more recent introduction to scale space can be found in the monograph by ter Haar Romeny [56].

Scale space theory makes it possible to look at (spatial) derivatives of the image in a mathematically well posed way. It is shown that the human front end visual system takes derivatives up to at least 4th order at various scales. It seems that the *differential structure* of an image is important to detect objects, for humans as

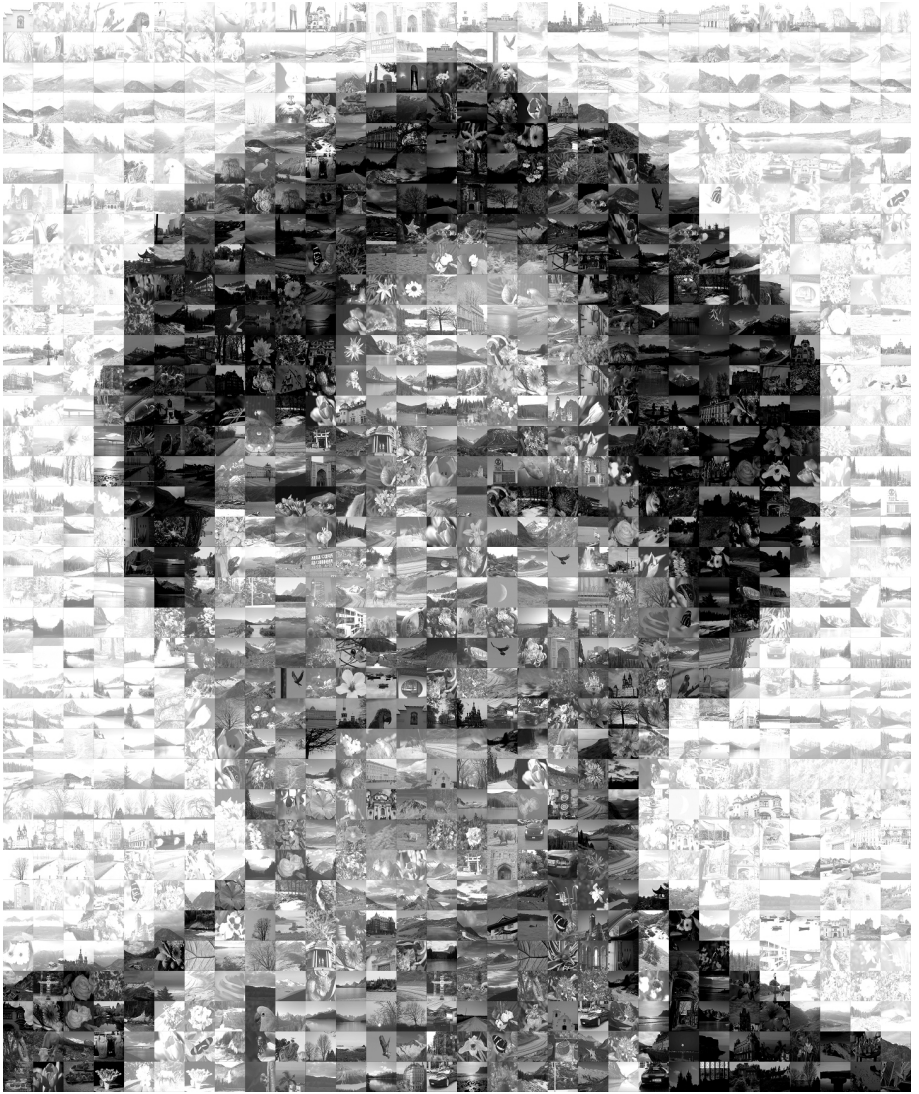


Figure 1.1: Image mosaic of Daniëlle. Seen from a close distance the image shows a collection of small pictures, but seen from a distance of 2-3 meters a picture of Daniëlle appears. The perceived image content depends on the scale of observation.

well as for machines. In the scale space of an image, several types of special scale space interest points can be identified, which are defined by the spatial derivatives in those points. Figure 1.2 shows an example of the scale space of an image and a 3D plot of top points, one of the types of scale space interest points used in this thesis. It is shown that such scale space interest points contain much information of the image, they can be used for image matching [111, 120, 119] and for image reconstruction [115, 93, 73]. The goal of this project was to further investigate the image information content in scale space interest points and to use these interest points for object-based image editing using image reconstruction from scale space interest points.

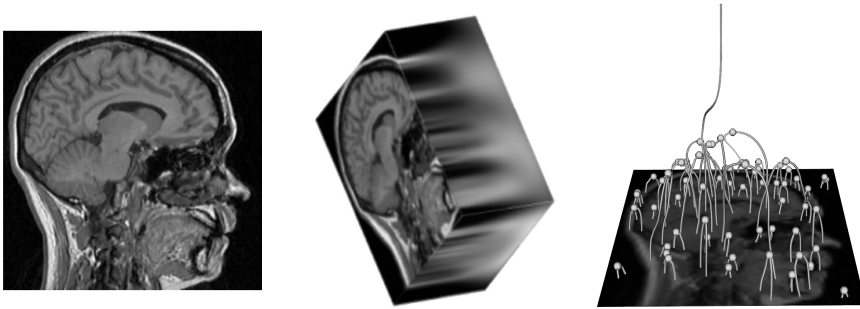


Figure 1.2: Scale space of an image. The left image shows an MR image of the brain (256×256 pixels), in the center the corresponding scale space is shown and on the right special points in the scale space, top points are shown (the third dimension represents scale).

This thesis is structured as follows. In chapter 2 α -scale spaces are introduced, a more general class of scale spaces of which the well known Gaussian scale space is a special case. A tool is introduced that is able to calculate and visualize interest points in such a scale space of an image. In chapter 3 an improved reconstruction scheme is introduced for image reconstruction from scale space interest points, which is compared to two other reconstruction algorithms for image reconstruction. For an objective comparison of the reconstruction results, objective image error measures are necessary. In chapter 4 a number of image quality measures found in literature are compared with each other with respect to image reconstruction. A novel scale space error measure is introduced and compared with the other quality measures. In chapter 5 we compare the information content of 10 types of scale space interest points by means of image reconstruction on two different databases of images. The results are evaluated using some of the image quality measures of chapter 4. We also introduce a novel method for combining different types of scale space interest points. In chapter 6 a feasibility study is presented using scale space interest points for object-based image editing. Finally, a summary and recommendations for further research are given in chapter 7.



Exploring Scale Space

This chapter is based on:

α -Scale Space Kernels in Practice.

F.M.W. Kanters, L.M.J. Florack, R. Duits, and B. Platel.

Pattern recognition and Image Analysis, Volume 15, Number 1, pp 208–211, March 2005.

2.1 Introduction

A scale space of a 2D image is a 3D volume with the scale s as the third dimension. In practice however, a scale space is often seen as a stack of images, sequentially blurred by convolution with a kernel. Figure 2.1 shows an example of a scale space of an image. For the well known Gaussian scale space the convolution kernel is a Gaussian bell-shaped kernel. Since a convolution in the spatial domain is equivalent to a multiplication in the Fourier domain, a scale space can also be built in the Fourier domain. For the Gaussian scale space this means multiplying the Fourier transformed image with the Fourier transformed Gaussian kernel followed by an inverse Fourier transform of the result. In theory these two methods are equivalent but in practice one might prefer one method over the other depending on several factors (described by Florack [41], see also the implementation notes “Spatial vs Fourier” on page 8).

It is shown that using reasonable axioms, a complete α -parameterized class of kernels exists resulting in so-called α -scale spaces [39, 118, 30]. The Gaussian scale space is one specific case ($\alpha = 1$). This chapter will focus on this broader class of scale spaces. In section 2.2 the concept of the α -scale spaces is described and the problem using their kernels in the spatial domain is introduced. Next an approximation of the α -kernel in the spatial domain is described in section 2.3. Because they have been hardly investigated and to explore possible applications, we decided to build an interactive visualization tool for α -scale spaces, named ScaleSpaceViz [72, 74], useful for visualizing α -scale spaces and special points within these scale spaces. Section 2.4 shows some results of different scale spaces visualized in ScaleSpaceViz. Finally conclusions and recommendations for future research are given in section 2.5.

2.2 α -Scale Spaces

In a Gaussian scale space the image is blurred according to the following evolution process:

$$\begin{cases} \frac{\partial}{\partial s} u = \Delta u \\ \lim_{s \downarrow 0} u(\cdot, s) = f(\cdot) \end{cases} \quad (2.1)$$

with s the scale, f the image, Δ the Laplacian and u the resulting scale space. This evolution process leads to convolution of the image with a Gaussian kernel. The α -scale spaces are created using a different evolution process:

$$\begin{cases} \frac{\partial}{\partial s} u = -(-\Delta)^\alpha u \\ \lim_{s \downarrow 0} u(\cdot, s) = f(\cdot) \text{ in } \mathbb{L}_2(\mathbb{R}^d)\text{-sense} \end{cases} \quad \alpha \in (0, 1] \quad (2.2)$$

Which leads to convolution with a so called α -kernel. Note that for $\alpha = 1$ we indeed obtain the evolution process of the Gaussian scale space. For $\alpha = \frac{1}{2}$ we obtain the evolution process of the Poisson scale space, which has some special properties and can be extended to a monogenic scale space as described by Felsberg [37]. Figure 2.2 shows an example of an image blurred with three different α -kernels.

Recall that the solution of the evolution process in (2.2) is given by:

$$u^{(\alpha)}(\underline{x}, s) = \left(k_s^{(\alpha)} * f \right) (\underline{x}) \quad (2.3)$$

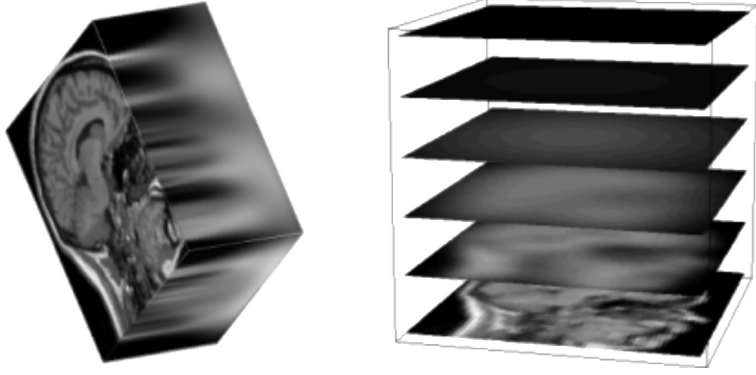


Figure 2.1: A scale space of a 2D image as a continuous 3D volume (left) and the same scale space as a stack of sequentially blurred images (right).

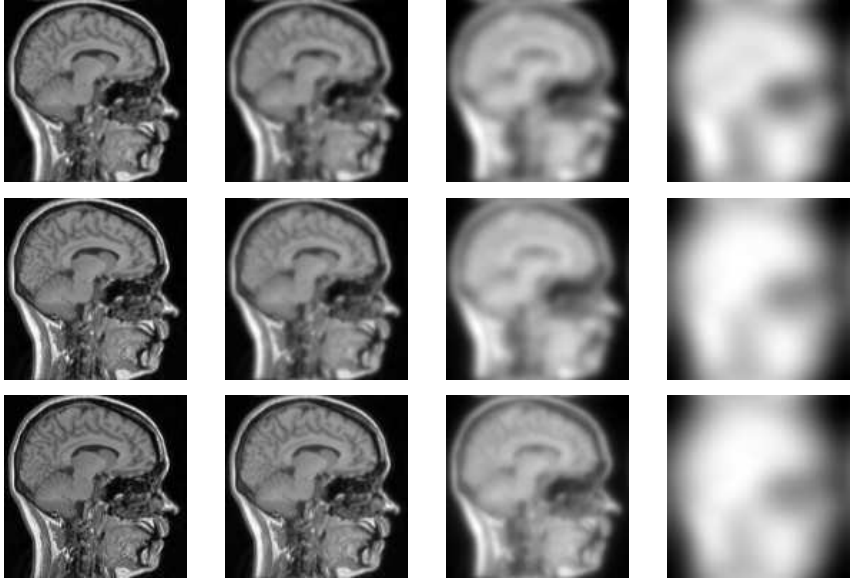


Figure 2.2: MR brain slice image (128x128) blurred by different kernels at various scales. Top row: $\alpha = 1$ (Gaussian scale space), center row: $\alpha = \frac{3}{4}$, bottom row: $\alpha = \frac{1}{2}$ (Poisson scale space). Grey-values have been mapped to full range for the sake of clarity. Note that details maintain longest in the Poisson scale space. Also note that the scales among the different α -scale spaces cannot be directly compared; scale parameter synchronization is necessary which is performed here by hand.

Implementation Notes

Spatial vs Fourier

Convolution in the spatial domain is equal to multiplication in the Fourier domain. In practice however there is a difference between these two in terms of speed and accuracy. Given an image of $N \times N$ pixels and a convolution kernel of $k \times k$ pixels, convolution needs k^2 multiplications and additions for every pixel in the image (boundaries not taken into account for simplicity) and thus has a complexity of $\mathcal{O}(k^2 N^2)$. To perform this operation in the Fourier domain one needs to calculate the Fourier transform of the image, which has a complexity of $\mathcal{O}(N^2 \log N)$ using FFT, followed by N^2 multiplications and finally perform an inverse Fourier transform. Thus for large kernels the operation in the Fourier domain will be faster while for small kernels convolution will be faster. However, with respect to the accuracy it is the opposite: small kernels (e.g. Gaussians with a small standard deviation) in the spatial domain will introduce large errors (since only a few samples are used to model the kernel) and wide kernels (e.g. Gaussians with a large standard deviation) will be small in the Fourier domain and thus introduce similarly large errors. For high accuracy, Fourier methods should be used for small kernels, while convolution should be used for large kernels. In practice, one should make a trade-off between speed and accuracy, dependent on the width of the kernel. A more rigorous approach can be found in the work of Florack [41]. Note that there are several more optimized algorithms available both for spatial and Fourier implementations, but the general idea still holds.

which is called an α -scale space¹, with the α -kernel $k_s^{(\alpha)} : \mathbb{R}^2 \rightarrow \mathbb{R}$ given in the Fourier domain by:

$$\mathcal{F}(k_s^{(\alpha)})(\underline{\omega}) = e^{-s||\underline{\omega}||^{2\alpha}} \quad (2.4)$$

with $0 < \alpha \leq 1$. Note that for $\alpha = 1$ one obtains the well known Gaussian kernel and for $\alpha = \frac{1}{2}$ the Poisson kernel. These so called α -kernels (with $0 < \alpha \leq 1$) are simple in the Fourier domain, but unfortunately for general α there is no closed-form representation in the spatial domain. In the case of a Gaussian or Poisson kernel, one easily can apply an inverse Fourier transform to obtain the kernel in the spatial domain, but for general α this is not possible. For any fixed α however, an expression can be found for the kernel in the spatial domain using the Hankel transform of the kernel in polar coordinates. Using:

$$\underline{x} = (r \cos \phi, r \sin \phi) \quad (2.5)$$

$$\underline{\omega} = (\rho \cos \varphi, \rho \sin \varphi) \quad (2.6)$$

¹also known as α -stable Lévy processes in probability theory.

Implementation Notes

Separability

An important feature of the Gaussian kernel is separability. This means that a convolution with a N -D Gaussian kernel can be performed by N 1D convolutions. This can be shown in the Fourier domain using the general N -D Gaussian kernel:

$$u_s(\underline{x}) = \mathcal{F}^{-1}\{\mathcal{F}(f(\underline{x}))(\omega) \cdot e^{-s\|\omega\|^2}\}$$

with $u_s(\underline{x})$ the blurred N -D image at scale s and $f(\underline{x})$ the original N -D image. Using N 1D convolutions one obtains:

$$u_s(\underline{x}) = \mathcal{F}^{-1}\{\mathcal{F}(f(\underline{x}))(\omega) \cdot \prod_{i=1}^N e^{-s\omega_i^2}\}$$

which can be rewritten to:

$$u_s(\underline{x}) = \mathcal{F}^{-1}\{\mathcal{F}(f(\underline{x}))(\omega) \cdot e^{-s(\sum_{i=1}^N \omega_i^2)}\}$$

Since the squared norm of a vector of length N is given by: $\|\underline{x}\|^2 = \sum_{i=1}^N x_i^2$ a convolution with an N -D Gaussian kernel is equal to N times a convolution with a 1D Gaussian kernel. In practice this means that the implementation of a convolution with a N -D kernel of size k^N on an N -D image of size M^N can be performed with complexity of $\mathcal{O}(NkM^N)$ instead of $\mathcal{O}(k^N M^N)$. However, this does not hold in general for the α -kernel. Consider blurring with a N -D α -kernel:

$$u_s(\underline{x}) = \mathcal{F}^{-1}\{\mathcal{F}(f(\underline{x}))(\omega) \cdot e^{-s\|\omega\|^{2\alpha}}\}$$

Using N convolutions with a 1D α -kernel one obtains:

$$u_s(\underline{x}) = \mathcal{F}^{-1}\{\mathcal{F}(f(\underline{x}))(\omega) \cdot e^{-s(\sum_{i=1}^N \omega_i^{2\alpha})}\}$$

But $\|\omega\|^{2\alpha} = (\sqrt{\sum_{i=1}^N \omega_i^2})^{2\alpha} \neq \sum_{i=1}^N \omega_i^{2\alpha}$ for $i \neq 1$.

the kernel in polar coordinates becomes:

$$\mathcal{F}(k_s^{(\alpha)})(\omega) = e^{-s\rho^{2\alpha}}, \rho = \|\omega\| \quad (2.7)$$

Any 2D function $f \in \mathbb{L}_2(\mathbb{R}^2)$ in polar coordinates can be decomposed as follows:

$$f(r, \phi) = \sum_{m=0}^{\infty} f_m(r) e^{-im\phi} \quad (2.8)$$

The Fourier transform of a product of a radial function with a harmonic function can

Implementation Notes

Sampling a Gauss kernel

The integral of a Gaussian kernel $\int_{-\infty}^{\infty} \frac{1}{\sqrt{\pi s^N}} e^{-\frac{s||\underline{x}||^2}{s}} = 1$. When sampling a Gaussian kernel, one should always make sure that the sum of elements of the kernel is *exactly* equal to 1. Straightforward sampling can introduce kernels with a sum smaller or larger than 1 due to quantization effects. This should be corrected, for example by adding a small amount to all or some of the kernel elements to obtain a sum of exactly 1. The importance can be shown with a simple example: Performing convolution with a kernel that has a sum of 0.98 on a constant image with greyscale 255 yields an image with a constant greyscale of less than 250. In theory the image should have been unchanged. Performing multiple convolutions (for example using separability) will amplify this effect.

be written in terms of the Hankel transform [36]:

$$\mathcal{F}(f)(\rho, \varphi) = \sum_{m=0}^{\infty} e^{im\varphi} \mathcal{H}_m(f_m)(\rho) \quad (2.9)$$

with $\mathcal{F}(f)(\rho, \varphi)$ the Fourier transform of f and $\mathcal{H}_m(f_m)(\rho)$ the Hankel transform of f_m defined by:

$$\mathcal{H}_m(f)(\rho) = i^m \int_0^{\infty} f(r) r J_m(r \rho) dr \quad (2.10)$$

with $J_\nu(z)$ the Bessel function defined by:

$$J_\nu(z) = \left(\frac{z}{2}\right)^\nu \sum_{k=0}^{\infty} \frac{(-1)^k}{k! \Gamma(k + \nu + 1)} \left(\frac{z}{2}\right)^{2k} \quad (2.11)$$

Since our kernel is independent of φ we have:

$$(\mathcal{H}_0(r \mapsto k_s^{(\alpha)}(\underline{x})))(\rho) = \mathcal{F}(k_s^{(\alpha)})(\omega) = e^{-s \rho^{2\alpha}} \quad (2.12)$$

The Hankel transformation equals its inverse transformation (for functions independent of ϕ), hence the inverse Fourier transform of our kernel becomes the Hankel transform of the kernel in the Fourier domain:

$$k_s^{(\alpha)}(r) = \mathcal{H}_0[\rho \mapsto (\mathcal{F}(k_s^{(\alpha)}))(\omega)](r) \quad , r = ||\underline{x}||, \rho = ||\underline{\omega}|| \quad (2.13)$$

Using the Hankel transformation it is possible to obtain an expression for the α -kernel in the spatial domain for a fixed α only. For $\alpha = \frac{1}{2}$ one obtains the Poisson kernel and for $\alpha = 1$ one obtains the well known Gaussian kernel, but for some values of

α the expression can become quite complicated, involving hypergeometric functions². In practice these can be hard to implement and therefore an approximation of the α -kernel in the spatial domain is presented.

2.3 Approximation of the α -kernel in the spatial domain

An approximation of the α -kernel for $\frac{1}{2} \leq \alpha \leq 1$ is proposed using a linear combination of one Gaussian kernel and one Poisson kernel. This is motivated by the fact that the α -scale space for $\frac{1}{2} \leq \alpha \leq 1$ gives a smooth transition between the Poisson scale space ($\alpha = \frac{1}{2}$) and the Gaussian scale space ($\alpha = 1$). In the Fourier domain the approximation becomes:

$$e^{-s \|\omega\|^{2\alpha}} \approx a(\alpha) e^{-s_g(\alpha) \|\omega\|^2} + b(\alpha) e^{-s_p(\alpha) \|\omega\|} \quad (2.14)$$

Note that a, b, s_g (Gauss scale) and s_p (Poisson scale) are positive constants, which may depend on α . In order to get a correct amplitude for $\omega = 0$ we must have $a + b = 1$, which results in:

$$e^{-s \|\omega\|^{2\alpha}} \approx a(\alpha) e^{-s_g(\alpha) \|\omega\|^2} + (1 - a(\alpha)) e^{-s_p(\alpha) \|\omega\|} \quad (2.15)$$

For dimensionality reasons, it is natural to assume:

$$s = t^\alpha, \text{ and } s_g = c_1(\alpha)t, \text{ and } s_p = c_2(\alpha)\sqrt{t} \quad (2.16)$$

An exact result is desired for $\alpha = 1$ and $\alpha = 1/2$, which results in the following constraints:

$$c_1(1) = 1 \text{ and } c_2(1/2) = 1 \quad (2.17)$$

which results in:

$$\begin{aligned} s_g(\alpha) &= c_1(\alpha)t = c_1(\alpha)s^{\frac{1}{\alpha}} \\ s_p(\alpha) &= c_2(\alpha)\sqrt{t} = c_2(\alpha)s^{\frac{1}{2\alpha}} \end{aligned} \quad (2.18)$$

Using the dimensionless frequency $\|\sqrt{t}\omega\| = \rho$, equation (2.15) can be rewritten to:

$$e^{-\rho^{2\alpha}} \approx a(\alpha) e^{-c_1(\alpha)\rho^2} + (1 - a(\alpha)) e^{-c_2(\alpha)\rho} \quad (2.19)$$

Note that for all relevant frequencies we can assume (roughly) that $\rho \leq 1$ (since frequencies above $\omega \approx \frac{1}{\sqrt{t}}$ should be suppressed by the blurring, these values in the kernel should be close to zero). Using a Taylor expansion up to order N around $\rho = 0$ we obtain:

$$e^{-\rho^{2\alpha}} \approx \sum_{n=0}^N (-1)^n \frac{\rho^{2\alpha n}}{n!} \approx a(\alpha) \sum_{n=0}^N (-1)^n \frac{c_1^n(\alpha) \rho^{2n}}{n!} (1 - a(\alpha)) \sum_{n=0}^N (-1)^n \frac{c_2^n(\alpha) \rho^n}{n!} \quad (2.20)$$

To obtain an expression for $a(\alpha)$, (2.20) may be integrated over all *relevant* frequencies ($0 \leq \rho \leq 1$):

²Expressions are typically half a page long. To save space, no example is given here.

$$\int_0^1 \sum_{n=0}^N (-1)^n \frac{\rho^{2\alpha n}}{n!} d\rho \approx \int_0^1 a(\alpha) \sum_{n=0}^N (-1)^n \frac{c_1^n(\alpha) \rho^{2n}}{n!} + (1 - a(\alpha)) \sum_{n=0}^N (-1)^n \frac{c_2^n(\alpha) \rho^n}{n!} d\rho \quad (2.21)$$

The left side of this equation is a function only depending on α , while the right hand side only depends on a , c_1 and c_2 . For a first order Taylor expansion ($N = 1$), the solution of $a(\alpha)$ in (2.21) can be expressed in α , c_1 and c_2 :

$$a(\alpha) = \frac{3(-2 + (1 + 2\alpha)c_2(\alpha))}{(1 + 2\alpha)(-2c_1(\alpha) + 3c_2(\alpha))} \quad (2.22)$$

To find an expression for $c_1(\alpha)$ and $c_2(\alpha)$, the amplitude of the α -kernel in the spatial domain at the origin is calculated using the Hankel transform (2.13):

$$k_s^{(\alpha)}(0) = \mathcal{H}_0[\rho \mapsto (\mathcal{F}(k_s^{(\alpha)}))(\omega)](0) = \frac{\Gamma(1 + \frac{1}{\alpha})}{4\pi s^{\frac{1}{\alpha}}} \quad (2.23)$$

with $\Gamma(z)$ the Euler gamma function given by:

$$\Gamma(z) \equiv \int_0^\infty t^{z-1} e^{-t} dt \quad (2.24)$$

The amplitudes at the origin for the Gaussian and Poisson kernel in two dimensions are given by:

$$G(0) = \frac{1}{4\pi s_g} \quad (2.25)$$

and:

$$P(0) = \frac{1}{2\pi s_p^2} \quad (2.26)$$

This results in the equation:

$$\frac{\Gamma(1 + \frac{1}{\alpha})}{4\pi s^{\frac{1}{\alpha}}} = \frac{a(\alpha)}{4\pi c_1(\alpha) s^{\frac{1}{\alpha}}} + \frac{1 - a(\alpha)}{2\pi c_2^2(\alpha) s^{\frac{1}{\alpha}}} \quad (2.27)$$

In order to solve this equation for c_1 and c_2 , an extra equation is needed. If $c_2^2 = 2c_1$ is chosen as an extra constraint, the amplitude for the α -kernel at the origin is not dependent on $a(\alpha)$ and we may readily solve for c_1 and thus c_2 :

$$\begin{aligned} c_1(\alpha) &= \frac{1}{\Gamma(1 + \frac{1}{\alpha})} \\ c_2(\alpha) &= \sqrt{\frac{2}{\Gamma(1 + \frac{1}{\alpha})}} \end{aligned} \quad (2.28)$$

Note that now the unknown parameters $a(\alpha)$, $b(\alpha)$, $s_g(\alpha)$ and $s_p(\alpha)$ of (2.14) are expressed in terms of α while obeying the constraints of (2.16) and (2.17). The final result can be found by substituting (2.28) in (2.18) and using this with (2.22) in equation (2.15).

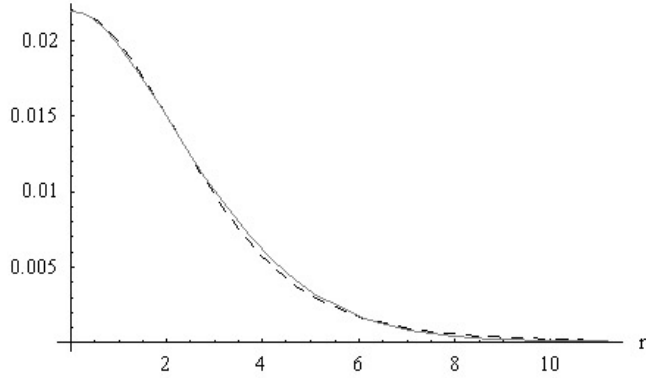


Figure 2.3: 1D Cross-section of the α -kernel (solid) and the approximated kernel (dashed) in the spatial domain for $\alpha = 0.75$ and $s = 3$.

2.3.1 Evaluation of the approximated kernel

The approximation presented in the previous section still contains the Gamma function, which may be computationally undesirable. Since we limit ourselves for the approximation to $\frac{1}{2} \leq \alpha \leq 1$, only a small, very smooth part of the Gamma function is actually used in our expressions. With Mathematica [142] a good numerical approximation can be found for this part of the Gamma function. Experiments were done using the following numerical approximations for $c_1(\alpha)$, $c_2(\alpha)$ and $a(\alpha)$:

$$\begin{aligned} c_1(\alpha) &= \frac{-1.5374 + 5.0722\alpha}{1 + 1.4548\alpha + 1.0800\alpha^2} \\ c_2(\alpha) &= \sqrt{\frac{-3.0748 + 10.1444\alpha}{1 + 1.4548\alpha + 1.0800\alpha^2}} \\ a(\alpha) &= \frac{3(-2 + (1 + 2\alpha)c_2)}{(1 + 2\alpha)(-2c_1 + 3c_2)} \end{aligned} \quad (2.29)$$

Using (2.15) and (2.29) simple approximations of the α -kernel in the spatial domain can be made. Figure 2.3 shows the 1D cross section of the α -kernel and the approximation for $\alpha = 0.75$ and $s = 3$.

The approximation is of course exact for $r = 0$ (due to the constraints) and for $r = \infty$. The maximum relative error between the α -kernel and the approximated kernel can be defined as:

$$\text{Maximum relative error} = \frac{\|k_s^{(\alpha)} - g_s^{(\alpha)}\|_\infty}{\|k_s^{(\alpha)}\|_\infty} \quad (2.30)$$

with $k_s^{(\alpha)}$ the α -kernel and $g_s^{(\alpha)}$ the approximated kernel. Figure 2.4 shows the maximum relative error between the α -kernel and the proposed approximation for various α and s . Note that the maximum relative error should be independent of scale, which is also found experimentally. The worst maximum relative error of 2.36% is found approximately at $\alpha = 0.65$.

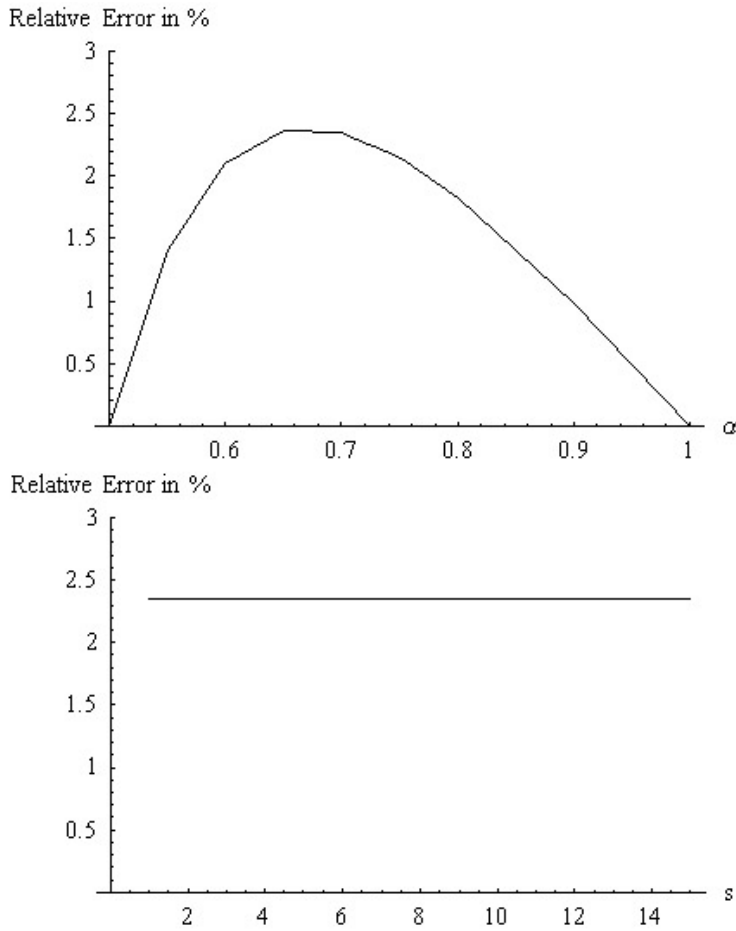


Figure 2.4: Maximum relative error between α -kernel and approximation for various α with scale fixed at $s=3$ (top) and for various s with α fixed at $\alpha = 0.7$ (bottom).

2.3.2 Spatial derivatives of the approximated kernel

In order to take derivatives of an image at a certain scale, one can do a convolution of the image with a derivative of the kernel. In this section we show how our approximation of the α -kernel performs if derivatives are taken. For simplicity we will only show the results for the Laplacian. The Laplacian of a function f in polar coordinates is defined by:

$$\mathcal{L}(r, \phi) = \frac{\partial^2 f}{\partial r^2} + \frac{1}{r} \frac{\partial f}{\partial r} + \frac{1}{r^2} \frac{\partial^2 f}{\partial \phi^2} \quad (2.31)$$

Which can be simplified if f is only dependent on r (which is the case for our kernels):

$$\mathcal{L}(r) = \frac{\partial^2 f}{\partial r^2} + \frac{1}{r} \frac{\partial f}{\partial r} \quad (2.32)$$

Using (2.32) on (2.15) one can find an approximation of the Laplacian kernel. However, the parameters are still optimized for the zeroth order kernel, hence the error may be non minimal. Figure 2.5 shows the resulting Laplacian kernel and the maximum relative error for the zeroth order optimization. To improve the approximation of the Laplacian kernel, the steps of equation (2.23) until (2.28) are repeated for the Laplacian kernel. First the amplitude of the kernel at the origin is again calculated:

$$k_s^{(\alpha)}(0) = \mathcal{H}_0[\rho \mapsto (\mathcal{F}(-\omega^2 k_s^{(\alpha)}))(\omega)](0) = -\frac{\Gamma(1 + \frac{2}{\alpha})}{8\pi s^{\frac{2}{\alpha}}} \quad (2.33)$$

The amplitude at the origin for the approximated kernel in two dimensions is given by:

$$\hat{k}_s^{(\alpha)}(0) = \frac{-\frac{a(\alpha)}{c_1^2(\alpha)} + \frac{12(a(\alpha)-1)}{c_2^2(\alpha)}}{4\pi s^{\frac{2}{\alpha}}} \quad (2.34)$$

Combined with (2.33) this results in the equation:

$$-\frac{\Gamma(1 + \frac{2}{\alpha})}{8\pi s^{\frac{2}{\alpha}}} = \frac{-\frac{a(\alpha)}{c_1^2(\alpha)} + \frac{12(a(\alpha)-1)}{c_2^2(\alpha)}}{4\pi s^{\frac{2}{\alpha}}} \quad (2.35)$$

If $c_2^4 = 12c_1^2$ is chosen as an extra constraint, the amplitude for the α -kernel at origin is not dependent on $a(\alpha)$ and we may readily solve for c_1 and thus c_2 :

$$\begin{aligned} c_1(\alpha) &= \sqrt{\frac{2}{\Gamma(1 + \frac{2}{\alpha})}} \\ c_2(\alpha) &= \sqrt[4]{\frac{24}{\Gamma(1 + \frac{2}{\alpha})}} \end{aligned} \quad (2.36)$$

Figure 2.6 shows the resulting kernel with the parameters optimized for the Laplacian kernel and the maximum relative error. Note that the results are indeed much improved. In a similar fashion, the parameters can be optimized for other derivatives of the kernel.

2.4 ScaleSpaceViz

The presented approximation is implemented in a software package named ScaleSpaceViz [47, 74]. It is designed for research purposes, to learn more about the intrinsic structure of

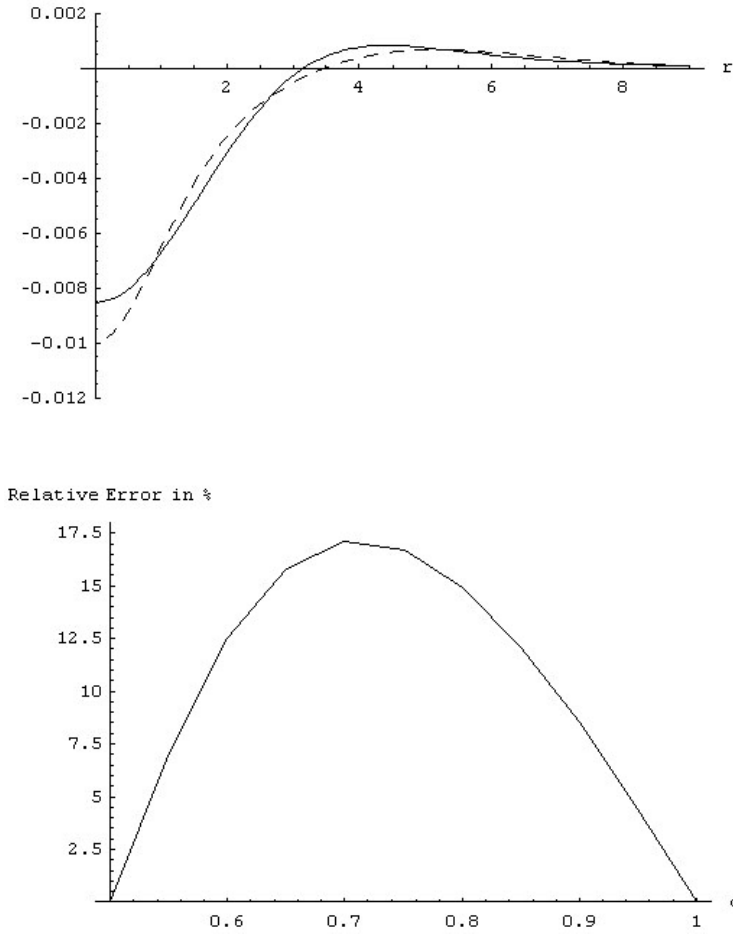


Figure 2.5: Top: Example of the Laplacian of the α -kernel (solid line) and the Laplacian of the approximated kernel (dashed line) for the zeroth order optimization ($\alpha = 0.7$, $s = 3$). Bottom: Maximum relative error between the Laplacian of the α -kernel and approximation for various α with scale fixed at $s=3$.

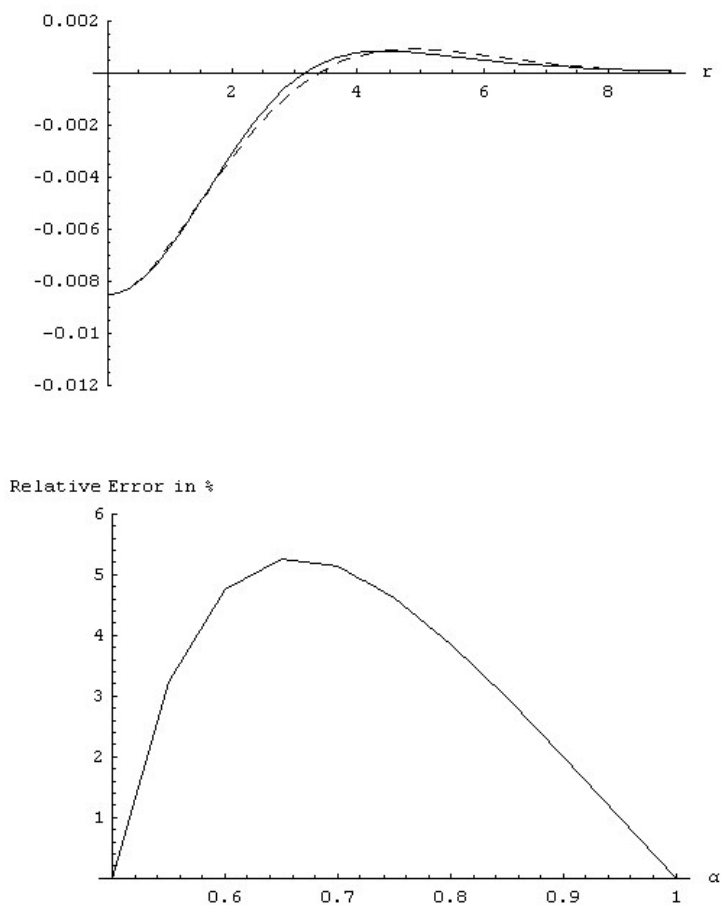


Figure 2.6: Top: Example of the Laplacian of the α -kernel (solid line) and the Laplacian of the approximated kernel (dashed line) optimized for the Laplacian ($\alpha = 0.7$, $s = 3$). Bottom: Maximum relative error between the Laplacian of the α -kernel and approximation for various α with scale fixed at $s=3$.

α -scale spaces and the *deep structure* of images [80]. ScaleSpaceViz provides a graphical user interface for calculating α -scale spaces of bitmap images with which we can visually inspect what happens in the deep structure. The calculation of a scale space can be done in the Fourier domain using the exact α -kernel and in the spatial domain using the approximated kernel. For Gaussian scale spaces both the spatial and the Fourier method are further optimized.

The α -scale spaces may be interesting for many applications. In this thesis we will focus on special points in scale space, so called *interest points* and their properties. In chapter 5 a number of commonly used interest points are presented and evaluated for the purpose of image reconstruction (see chapter 3). ScaleSpaceViz can be used to visualize (and calculate some of) these interest points. In this section a number of examples is presented using interest points called top points and scale space saddles. Some of the properties of these points are shown using the visualization tools of ScaleSpaceViz.

Implementation Notes

ScaleSpaceViz

ScaleSpaceViz is programmed in C++ using the Visualization Toolkit (VTK) [77]. As input it can take volume data from Mathematica [142] which already contains a scale space or it can calculate an α -scale space from a bitmap image, using a Fourier method, a convolution method or a combination of both.

From this volume data, level-crossing surfaces can be calculated using the marching cubes algorithm [102]. In practice, zero-crossings will be the most useful level-crossings. Intersections of these zero-crossing surfaces of two different volumes (e.g. x-derivative scale space and y-derivative scale space) lead to 3D lines (e.g. critical paths). Intersecting these lines with another zero-crossings surface leads to isolated points.

One limitation of this method to calculate interest points is that all three surfaces must be in memory, which makes it only useful for small volumes (smaller than $128 \times 128 \times 128$). This is solved by only taking two subsequent scale levels into account and sweep the volume through scale. Critical paths have to be back-tracked since loops can occur. Both implementations are available in ScaleSpaceViz.

Besides calculating interest points and scale spaces, ScaleSpaceViz is primarily designed to do interactive visualization. It is possible to “fly” through the scale space and look around. Multiple scale spaces of different images can be put in one scene for comparison. It is also possible to change color and opacity for all objects in the scene, put bounding boxes in the scene and change the background. ScaleSpaceViz has been demonstrated at the ECCV [72] and is publicly available on the internet at www.bmi2.bmt.tue.nl/image-analysis/people/FKanters/ in the software section.

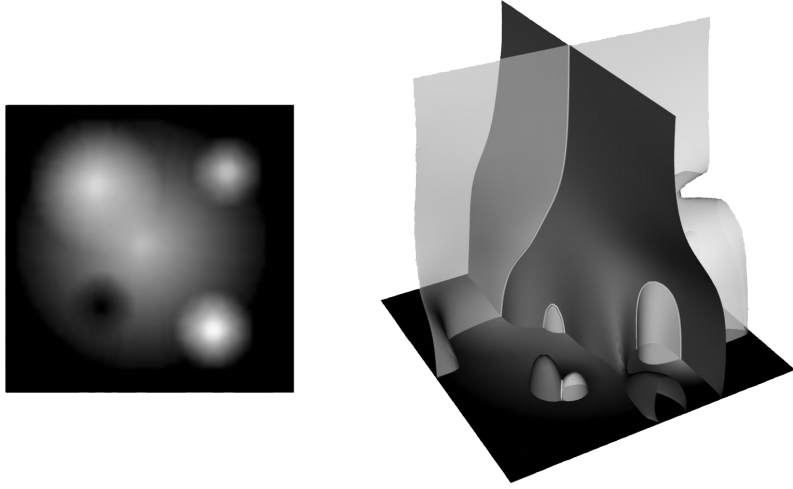


Figure 2.7: Left: Simple 2D image with some Gaussian blobs. Right: Zero crossings of x- and y-derivative surfaces (z axis is scale) and their intersections (critical paths).

2.4.1 Interest points

The program can calculate and visualize special interest points in scale space by calculating zerocrossings in 3D volumes. Consider for example spatial critical points defined by:

Definition 1. *Spatial critical points are points where the spatial gradient is zero. For 2D images these points are maxima, minima or saddles.*

If these points are tracked through scale, so called critical paths are obtained. These paths are the intersections of the zerocrossings of the x- and y-derivative of the scale space of a 2D image. Figure 2.7 shows the critical paths of a simple blob image plus the zerocrossing surfaces of the x- and y-derivative. In the diffusion process extrema can annihilate with saddles or extrema-saddle pairs can be created. This happens at the so called top points, which are defined by:

Definition 2. *Top points are spatial critical points where the Hessian degenerates ($\det H = 0$). For generic 2D images, these points are annihilations or creations of saddles with maxima or minima.*

The Hessian of a 2D image f is given by:

$$H(f) = \nabla \nabla^T f = \begin{pmatrix} \partial_x^2 f & \partial_x \partial_y f \\ \partial_y \partial_x f & \partial_y^2 f \end{pmatrix} \quad (2.37)$$

These points are the intersections of the zerocrossings of the determinant of the Hessian with the critical paths. Other points of interest are for example scale space saddles, which are spatial critical points, with a zero scale derivative (which by definition equals a zero Laplacian in a Gaussian scale space). These points are intersections of the zerocrossings

of the Laplacian with the critical paths. Figure 2.8 shows again the simple blob image, with the critical paths, top points and scale space saddles. In figure 2.9 some examples of critical paths and top points with creation-annihilation pairs are shown.

2.4.2 Approximated α -scale space kernel

Critical paths and top points can be calculated for any α -scale space. The paths of different α -scale spaces will not be the same for an image and even topological differences will occur. Figure 2.10 shows the critical paths and the top points of the same image for three different α 's. To validate the quality of the approximated kernels as described previously, the program can calculate critical paths and top points for α -scale spaces both using convolution in the spatial domain and multiplication in the Fourier domain. Figure 2.11 shows the critical paths and the top points of an image, using the Fourier and the spatial convolution method.

2.4.3 Stability of scale space interest points

For any practical application of scale space interest points it is crucial to have information about the stability of those points under small perturbations. For example critical points depend on the position of maxima, minima and saddles in the image. In almost homogeneous areas in the image, the extrema merely depend on noise. These points will thus be very unstable under small perturbations and not very useful for applications. Figure 2.12 shows the critical paths and top points of different noise realizations of an image. Note that indeed there is a difference between stable top points in regions with much structure and unstable top points in regions with little structure. Lifshitz and Pizer[92] looked at the stability regarding creations in scale space. Platel et al. presented a stability measure for top points based on the total variation norm [120] which could also be used for other critical points. Balmachnova et al. presented a more elaborate stability measure for top points which also takes directions into account [3].

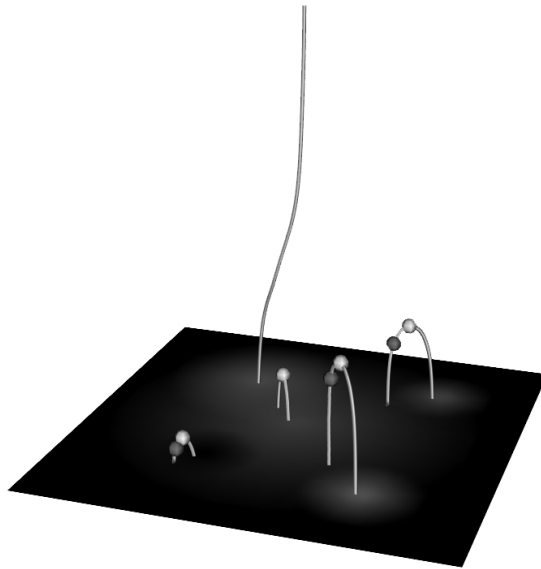


Figure 2.8: Critical paths of the blob image from Figure 2.7. The light dots show the top points and the dark dots show the scale space saddles. Note that not all paths have scale space saddles and there can be more saddles on one critical path. There is however *at least* one top point in each critical path (with the remaining path having a top point in infinity).

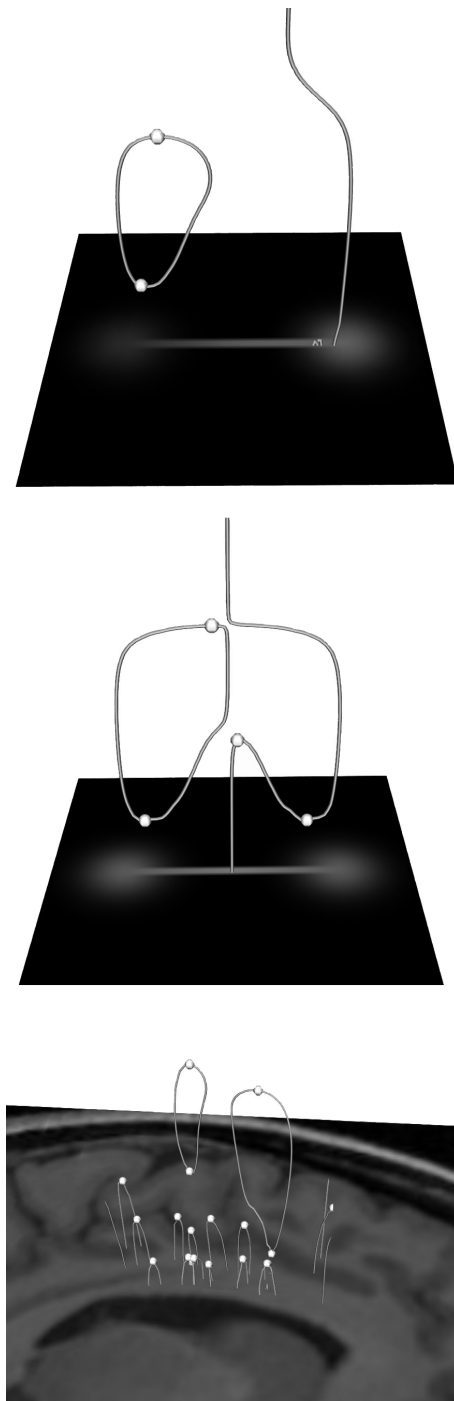


Figure 2.9: Examples of creation-annihilation pairs of extrema and saddles. Top and center: two artificial test images with dumbbell examples. Bottom: Part of an MR brain image with creation-annihilation pairs.

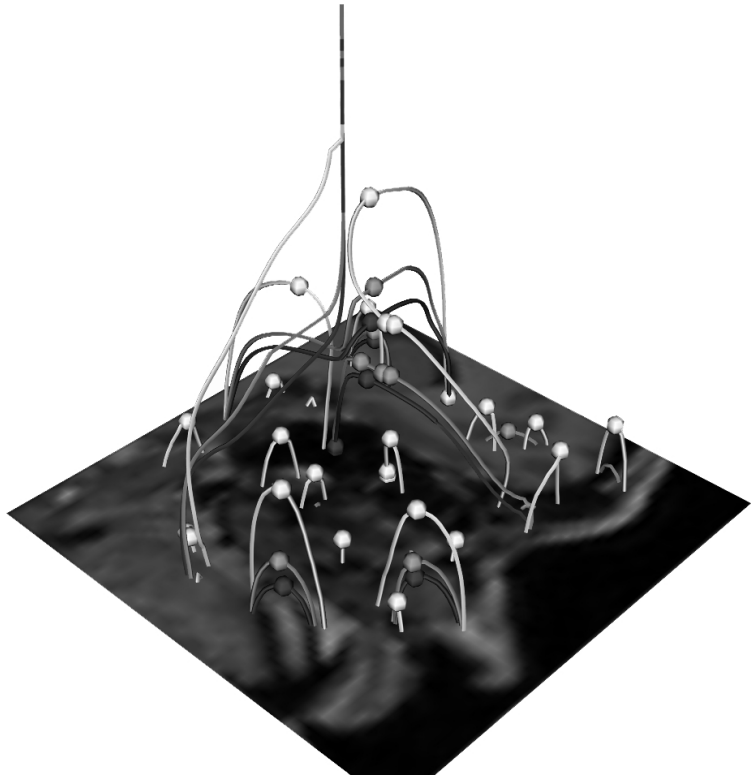


Figure 2.10: Critical paths from scale spaces of an MR image with $\alpha = 0.5$ (white), $\alpha = 0.75$ (light grey) and with $\alpha = 1.0$ (dark grey). Note that there are topological differences in the large paths in the center between the different α -scale spaces.

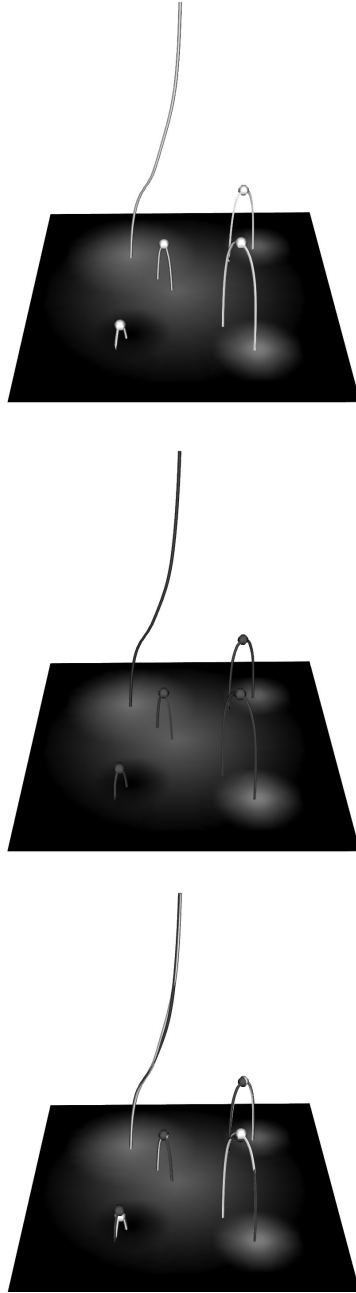


Figure 2.11: Experiments to validate the approximation of the α -kernel in the spatial domain. Top: Scale space calculated in the Fourier domain with original kernel ($\alpha = 0.65$). Center: The same scale space calculated with the approximated kernel in the spatial domain. Bottom: The two methods projected on top of each other; the Fourier method in white, the convolution method in grey.

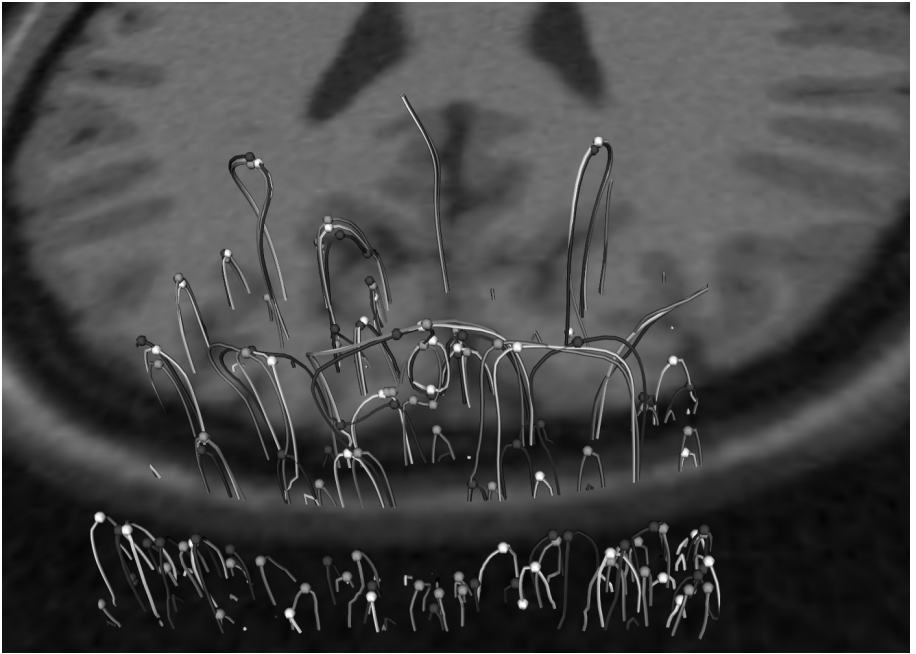


Figure 2.12: Critical paths and top points of three different noise realizations of a CT brain image (in white, light grey and dark grey). Note that there is no clear relation between the position of top points of the three noise realizations in the dark image area at the bottom; in this almost homogeneous part of the image, the critical paths and the top points are dominated by noise and are thus unstable. In areas with more structure (top part) one can see a clear relation between the critical paths and top points of the different noise realizations. The top points are very stable *perpendicular* to the elongated structure and less stable *along* the elongated structure. Note that the critical paths can be very different in stable areas, while the positions of the top points are almost the same.

2.5 Conclusion and discussion

Using α -scale spaces in practice can be a problem, since the corresponding kernels are only known in closed form in the Fourier domain for $0 < \alpha < 1$. Some applications however can make advantage of a spatial representation of an α -kernel. We solve this problem by approximating a 2D α -kernel in the spatial domain for $\frac{1}{2} < \alpha < 1$ with a linear combination of one Gaussian kernel and one Poisson kernel. The maximum relative error between the α -kernel and the approximation is low (less than 2.4%) and independent of scale. With the presented parameters, the error propagates in the derivatives of the kernel resulting in a maximum relative error of approximately 12% for a second order derivative of the kernel. This is solved by adding constraints for derivatives to calculate the parameters in a similar way as we have done for the zeroth order case, resulting in a maximum relative error of less than 5.4%.

The presented approximation is implemented in a software tool named ScaleSpace-Viz. This tool has proven to be useful in exploring the deep structure of images and constructing applications involving scale space interest points, such as reconstruction and matching. Future research should include calculation of more types of interest points (such as described in chapter 5). Also memory usage of the program should be optimized. Calculation of top points still consumes much memory, since all scale spaces necessary are kept in memory to calculate the intersections. More research should be performed on the differences between different α -scale spaces, only a few examples are calculated using different α on a limited number of images. The question which α should be used for which application still remains unanswered since for the remainder of this thesis, we limit ourselves to the Gaussian scale space ($\alpha = 1$) due to time limitations. In the next chapters, scale space interest points such as the top points introduced here, will be used for image reconstruction (chapter 3 and 5) and image editing (chapter 6).



Image Reconstruction from a Sparse Pointset

This chapter is based on:

A Linear Image Reconstruction Framework Based on Sobolev Type Inner Products.
B. Janssen, F.M.W. Kanters, R. Duits, L.M.J. Florack, and B.M. ter Haar Romeny.
International Journal of Computer Vision (IJCV), Volume 70, Number 3, pp 231–240,
December 2006.

On Image Reconstruction from Multiscale Top Points.
F.M.W. Kanters, M. Lillholm, R. Duits, B. Janssen, B. Platel, L.M.J. Florack,
and B.M. ter Haar Romeny. Proc. of the 5th Int. Conf. on Scale-Space and PDE Methods 2005,
Hofgeismar, Germany, LNCS Volume 3459, pp 431–442, April 2005.

3.1 Introduction

There are several types of special interest points that can be calculated from a scale space of an image. Chapter 5 describes a number of commonly used interest points and their properties. In chapter 2 we have presented software to visualize these special points and to calculate some of these interest points from a 2D image. To use such interest points as a good representation of the image, one needs to know how much image information is present in these points. One method to make the information content of these points explicit is image reconstruction. In this chapter we will present reconstruction algorithms that can be used to evaluate the information content of scale space interest points.

Nielsen, Lillholm and Griffin presented a linear minimal variance reconstruction scheme to reconstruct an image given a set of scale space interest points and the local N-jet in those points [115, 93]. Based on that reconstruction algorithm, Kanters et al. [73] presented a closed form solution for the Gram matrix of that linear framework. The prior used in these linear frameworks is however not sufficient to create visually attractive reconstructions if not enough constraints are used. To overcome this problem Nielsen and Lillholm proposed a prior based on natural image statistics, the Brownian reconstruction [115, 93]. Recently a generalization of the linear reconstruction framework is proposed by Janssen et al. which — in a different way — also tries to overcome this problem while maintaining linearity [63, 64]. This framework is also used in the context of motion extraction [65, 46].

In section 3.2 three reconstruction algorithms are described: The standard linear reconstruction as proposed by Nielsen and Lillholm [115, 93] (using the closed form solution of Kanters et al. [73]), the Brownian reconstruction algorithm proposed by Nielsen and Lillholm [115, 93] and the Sobolev type inner product reconstruction proposed by Janssen et al. [63]. In section 3.3 the algorithms are evaluated using some test images and finally some conclusions are given in section 3.4.

3.2 Image reconstruction algorithms

Given a set of points and some local “features” in those points, the goal of the proposed image reconstruction algorithms is to create an image that is visually close to the original image and has the same points and features as the original image. Points and local features in this context are localized filter responses, which will be defined more rigorously in the next sections. In general for sparse point sets different images could have exactly the same points and features, different images can be equivalent *seen through the set of filters*. All images having the same filter responses are elements of the so called *metameric class*. Figure 3.1 shows this schematically. The presented reconstruction algorithms differ in the sense that each minimizes a different prior to select a different instance from the same metameric class.

3.2.1 Standard linear reconstruction

Consider a filter response c_i localized at a certain point i that is defined as an \mathbb{L}_2 inner product between a given filter ϕ_i and the original image f :

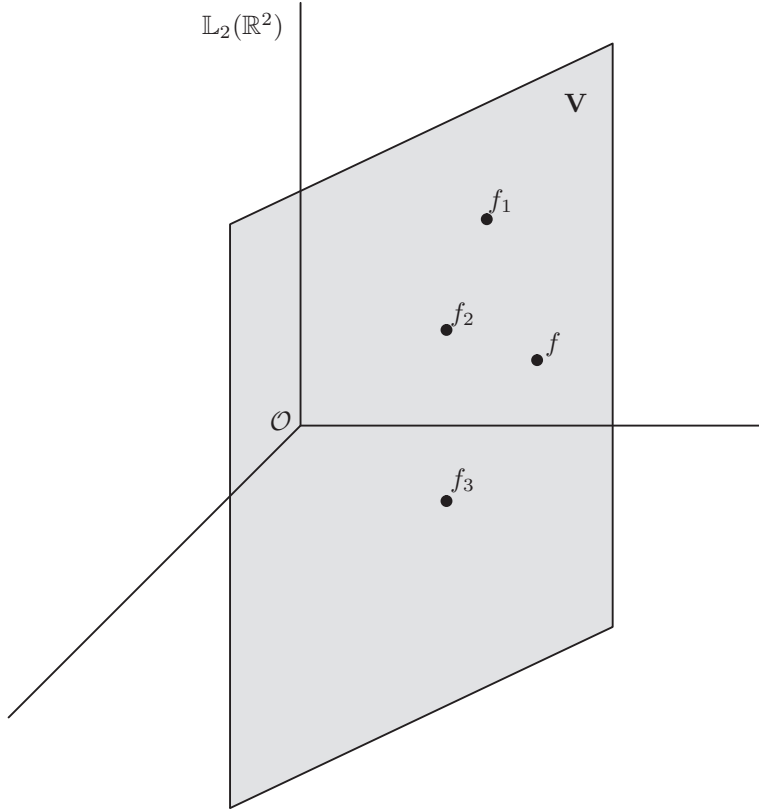


Figure 3.1: The metameric class V contains all images of $\mathbb{L}_2(\mathbb{R}^2)$ that have the same filter responses. These images are equivalent *seen through these filters*. The original image f is always a member of the metameric class. Images f_1 through f_3 are different instances of the metameric class V and thus sharing the same filter responses. Note that these images do not necessarily have to look similar to the original image.

$$c_i = \langle \phi_i | f \rangle_{\mathbb{L}_2} = \int_{\Omega} \phi_i(x) f(x) dx \quad (3.1)$$

All images in the metamer class V will have exactly the same filter responses c_i for all $i \in \{1, \dots, N\}$. The goal of the reconstruction algorithm is to find an image $g \in \mathbb{L}_2(\mathbb{R}^2)$ such that:

$$\langle \phi_i | g \rangle_{\mathbb{L}_2} = c_i \quad (3.2)$$

for all $i \in \{1, \dots, N\}$. When the number of filters N is relatively low, many images g with the constraints of (3.2) will exist. The standard linear reconstruction selects the candidate g that minimizes the \mathbb{L}_2 -norm, which results in the image of the metamer class V that is closest to the zero image in \mathbb{L}_2 -sense. Nielsen and Lillholm [115, 93] used a variational approach resulting in the following functional that should be minimized:

$$S[g] = \frac{1}{2} \langle g | g \rangle_{\mathbb{L}_2} + \sum_i \lambda_i \langle f - g | \phi_i \rangle_{\mathbb{L}_2} \quad (3.3)$$

Implementation Notes

Minimal variance reconstruction

The standard linear reconstruction selects the image from the metamer class V that is closest to the zero image in \mathbb{L}_2 -sense. One could also replace f with $f - \bar{f}$ (where \bar{f} represents the mean of the image f) before calculating the features c_i . The resulting reconstruction $g = \mathcal{P}_V f + \bar{f}$ will then select the image from the metamer class that is closest to the mean value of the original image instead of the zero image, again in \mathbb{L}_2 -sense. Nielsen and Lillholm refer to this as *minimal variance reconstruction* [115, 93].

The first part of equation (3.3) minimizes the \mathbb{L}_2 -norm while the second part, for properly chosen λ_i , makes sure that the constraints of (3.2) are met. Another approach to find the solution of (3.3) is by an orthogonal projection of the original image f on the linear space spanned by the filters c_i :

$$g = \mathcal{P}_V f = \langle \phi^i | f \rangle_{\mathbb{L}_2} \phi_i \quad (3.4)$$

where we have defined $\phi^i \stackrel{\text{def}}{=} G_{ij}^{-1} \phi_j$ with G the Gram matrix with elements:

$$G_{ij} = \langle \phi_i | \phi_j \rangle_{\mathbb{L}_2} \quad (3.5)$$

Note that Einstein's summation convention applies whenever two equal indices are used, e.g. $\phi^i \psi_i = \sum_i \phi_i \psi_i$. The orthogonal projection ensures a solution closest to the origin (zero image) while still obeying the constraints. Figure 3.2 shows this schematically. More details on the implementation can be found in the implementation notes "Standard

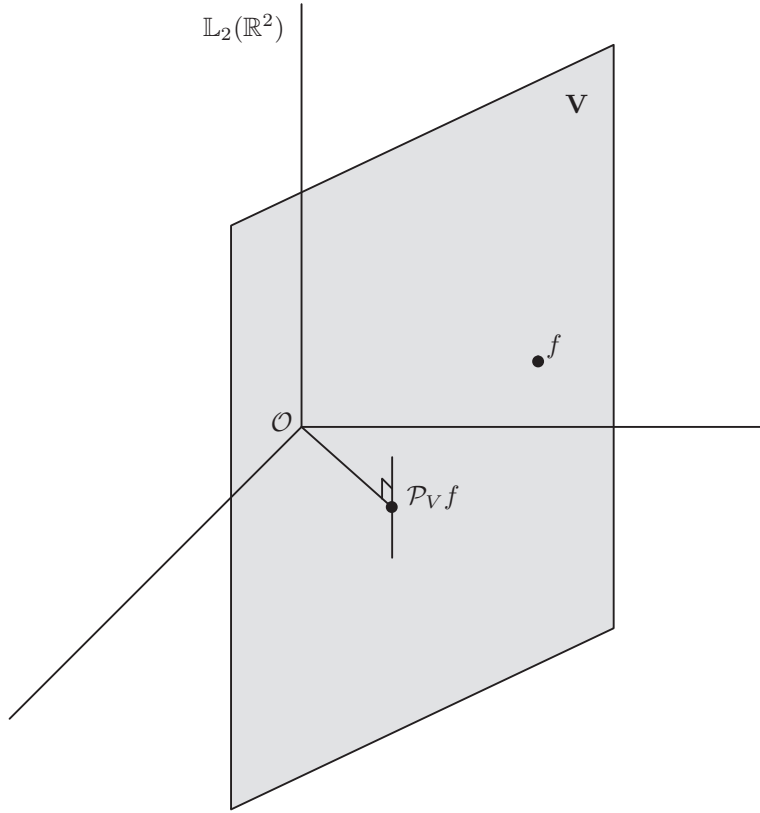


Figure 3.2: The metamerism class V contains all images of $\mathbb{L}_2(\mathbb{R}^2)$ that have the same filter responses. The orthogonal projection $\mathcal{P}_V f$ is the element of V that is closest to the origin, the zero image. This can be easily seen by applying the rule of Pythagoras.

linear reconstruction” on page 32, while a more rigorous mathematical analysis can be found in the work by Duits [29].

3.2.2 Brownian reconstruction

The first term of equation (3.3) represents the prior model of the reconstructed image. The general form of the functional to minimize can be written as:

$$S[g] = \Psi[g] + \sum_i \lambda_i \langle f - g | \phi_i \rangle_{\mathbb{L}_2} \quad (3.9)$$

where Ψ is some prior that should be minimized and the other terms are again the constraints. For the Brownian reconstruction a prior known as the Tikhonov regularizer [130] is used which results in:

Implementation Notes

Standard linear reconstruction

Consider the 2D Gaussian kernel φ_s of scale s :

$$\varphi_s(x, y) = \frac{1}{4\pi s} e^{-\frac{x^2+y^2}{4s}} \quad (3.6)$$

For the remainder of this thesis we will use derivatives of the shifted Gaussian kernel as filters for our reconstruction. We define ϕ_i as:

$$\phi_i(x, y) = \frac{\partial^{m+n} \varphi_s(x - \xi, y - \eta)}{\partial x^m \partial y^n} \quad (3.7)$$

with $i \stackrel{\text{def}}{=} (m, n, \xi, \eta, s) \in \mathbb{N}_0^2 \times \mathbb{R}^2 \times \mathbb{R}_+$. The algorithm for the standard linear reconstruction now becomes:

Algorithm 3.1: Standard linear reconstruction

1. Calculate the feature vector \underline{c} . (For all N points calculate c_i of equation 3.1).
 2. Calculate the Gram matrix \mathbf{G} . (For all i, j calculate \mathbf{G}_{ij} of equation 3.5).
 3. Calculate the normalization matrix \mathbf{S} with elements $\frac{1}{\sqrt{\mathbf{G}_{ij}}}$ on the diagonal and 0 otherwise.
 4. Solve the linear system $(\mathbf{SGS})\underline{x} = \mathbf{S}\underline{c}$ using Singular Value Decomposition or other method.
 5. Build reconstruction function $g(x, y) = \sum_i x_i S_{ii} \phi_i(x, y)$.
 6. Sample the reconstruction function $g(x, y)$.
-

Note that for step 2 one can use:

$$\langle \phi_{(m_i, n_i, \xi_i, \eta_i, s_i)} | \phi_{(m_j, n_j, \xi_j, \eta_j, s_j)} \rangle_{\mathbb{L}_2} = (-1)^{m_j+n_j} \phi_{(m_i+m_j, n_i+n_j, \xi_i-\xi_j, \eta_i-\eta_j, s_i+s_j)} \quad (3.8)$$

$$S[g] = \int_{\Omega} |\nabla g|^2 dx + \sum_i \lambda_i \langle f - g | \phi_i \rangle_{\mathbb{L}_2} \quad (3.10)$$

This prior is based on a Brownian motion image model and therefore the reconstruction

is referred to as Brownian reconstruction. The solution g of this minimization problem is implemented using an iterative algorithm, cf. Lillholm and Nielsen [93]. First the standard linear reconstruction is performed, as described in section 3.2.1. Using steepest descent a step is taken that decreases the functional of (3.10). Note that this solution is not necessarily an element of the metameric class V and thus a back projection to V is performed. These steps are repeated until convergence has been reached. Figure 3.3 shows this schematically. More details on the implementation can be found in the implementation notes “Brownian reconstruction” on page 33.

Implementation Notes

Brownian reconstruction

The Brownian reconstruction starts with the standard linear reconstruction and then iteratively tries to minimize the functional of (3.10). The filters used are the ones of equation (3.7). The algorithm for the Brownian reconstruction is as follows:

Algorithm 3.2: Brownian reconstruction

1. Calculate the initial $g_{i=0}$ using algorithm 3.1.
 - Do until convergence:
 2. $g'_{i+1} = g_i + \delta t \frac{\delta S}{\delta g}$, in which δt is the time step and $\frac{\delta S}{\delta g}$ the functional derivative of 3.10. Note that in general g'_{i+1} is not an element of V .
 3. Perform back projection into V : $g_{i+1} = \mathcal{P}_V g'_{i+1}$.
 - End Do
-

3.2.3 Sobolev type inner product reconstruction

For the Sobolev type inner product reconstruction the definition of the inner product is generalized. For a positive, symmetric operator A we can define the A -inner product:

$$\langle f|g \rangle_A = \langle f|g \rangle_{\mathbb{L}_2} + \langle Af|Ag \rangle_{\mathbb{L}_2} = \langle f|(I + A^\dagger A)g \rangle_{\mathbb{L}_2} = \langle (I + A^\dagger A)f|g \rangle_{\mathbb{L}_2} \quad (3.11)$$

The filter responses c_i of equation 3.1 can be rewritten in terms of the A -inner product:

$$c_i = \langle \phi_i|f \rangle_{\mathbb{L}_2} = \langle \psi_i|f \rangle_A \quad (3.12)$$

where ψ_i is given by:

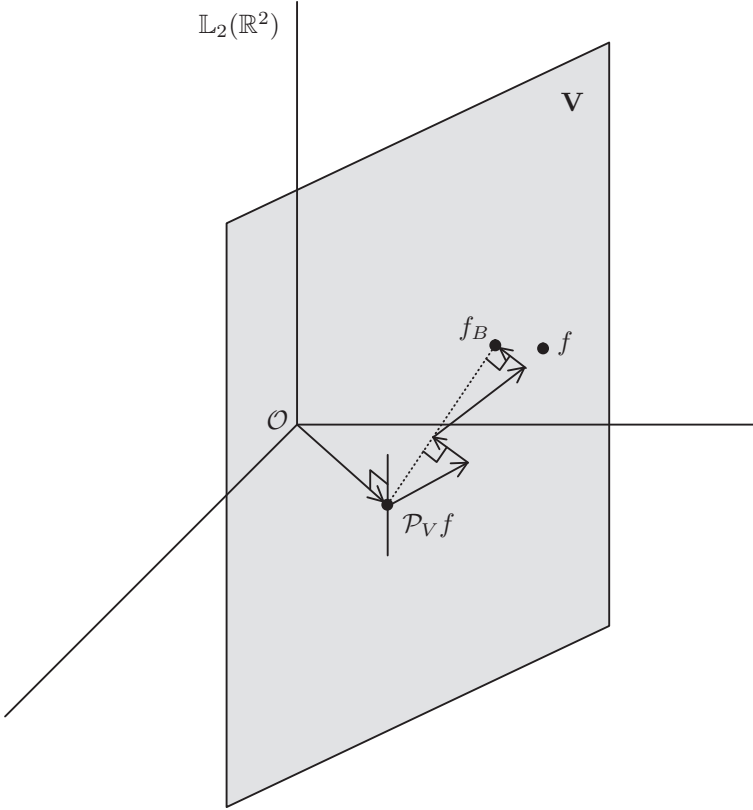


Figure 3.3: The metamer class V contains all images of $\mathbb{L}_2(\mathbb{R}^2)$ that have the same filter responses. The standard linear reconstruction performs an orthogonal projection $\mathcal{P}_V f$ that results in the element of V that is closest to the origin in \mathbb{L}_2 -sense, the zero image. The Brownian reconstruction starts with this reconstruction and then iteratively tries to minimize $\int_{\Omega} |\nabla f_B|^2 dx$, while back projecting to V in order to obey the constraints.

$$\psi_i = (I + A^\dagger A)^{-1} \phi_i \quad (3.13)$$

This can be shown by applying equation 3.11:

$$\langle \psi_i | f \rangle_A = \langle (I + A^\dagger A)^{-1} \phi_i | f \rangle_A = \langle (I + A^\dagger A)(I + A^\dagger A)^{-1} \phi_i | f \rangle_{\mathbb{L}_2} = \langle \phi_i | f \rangle_{\mathbb{L}_2} \quad (3.14)$$

Given the filters ψ_i one can minimize the A -norm $\| \cdot \|_A$ in a similar fashion as minimizing the \mathbb{L}_2 -norm for the given filters ϕ_i of equation 3.3. In this case the functional to minimize becomes:

$$S[g] = \frac{1}{2} \langle g | g \rangle_A + \sum_i \lambda_i \langle f - g | \psi_i \rangle_A \quad (3.15)$$

For the remainder of this thesis, the definition of A proposed by Janssen et al. [63, 64] and Duits [29] is used ¹:

$$A = -\gamma \sqrt{-\Delta} \quad (3.16)$$

Using this definition of A , a Sobolev type inner product is used in the reconstruction rather than the standard \mathbb{L}_2 inner product. Using (3.11) and (3.16) the functional of (3.15) becomes:

$$S[g] = \frac{1}{2} \int_{\Omega} g^2 dx + \frac{1}{2} \gamma^2 \int_{\Omega} |\nabla g|^2 dx + \sum_i \lambda_i \langle f - g | \phi_i \rangle_{\mathbb{L}_2} \quad (3.17)$$

In which again a Tikhonov regularizer can be identified, among with another term in the prior. Note that there is a free parameter γ which has an optimum value dependent on the image and the number of constraints [63, 64]. The solution g of this *linear* minimization problem boils down to an orthogonal projection on the intersection of the measurement filters, similar to the standard linear reconstruction cf. Janssen et al. [63, 64]. Note that for $\gamma = 0$ this boils down to the normal orthogonal projection in \mathbb{L}_2 . For $\gamma > 0$ this is a skew projection in \mathbb{L}_2 and an orthogonal projection in the Sobolev space. Figure 3.4 shows this schematically. More details of the implementation can be found in the implementation notes “Sobolev reconstruction”.

¹The operational significance of the fractional operator $-\sqrt{-\Delta}$, which is the generator of the Poisson scale space, is explained in detail by Duits et al. [29]. In Fourier space it corresponds to the multiplicative operator $-||\omega||$.

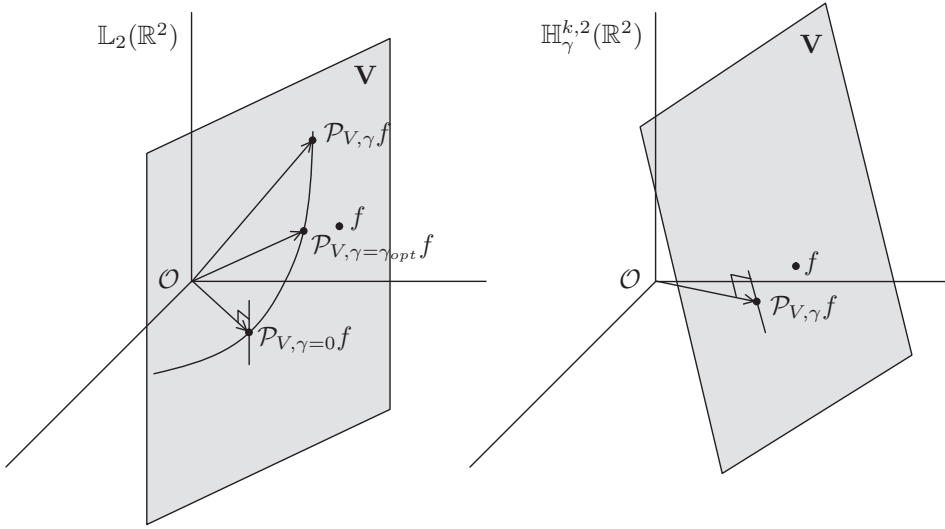


Figure 3.4: The metamer class V contains all images of $\mathbb{L}_2(\mathbb{R}^2)$ that have the same filter responses. The Sobolev reconstruction with $\gamma = 0$ is equal to the standard linear reconstruction and hence is an orthogonal projection in $\mathbb{L}_2(\mathbb{R}^2)$ (left figure). For $\gamma > 0$ this is an A -orthogonal projection or orthogonal projection in $\mathbb{H}_{\gamma}^{k,2}(\mathbb{R}^2)$ (right figure), which is a skew projection in $\mathbb{L}_2(\mathbb{R}^2)$ (left figure). The smoothness of the projection increases with $\gamma > 0$. In practice there is an optimal γ dependent on the original image.

Implementation Notes

Sobolev reconstruction

The Sobolev reconstruction is similar to the standard linear reconstruction, but uses a different type of inner product. Therefore the kernels and Gram matrix become more complicated. Using (3.16) we obtain the following kernel:

$$\psi_i = (I + A^\dagger A)^{-1} \phi_i = (I - \gamma^2 \Delta)^{-1} \phi_i = \mathcal{F}^{-1}(\omega \mapsto \frac{1}{1 + \gamma^2 \|\omega\|^2} \mathcal{F}(\phi_i)(\omega)) \quad (3.18)$$

In which \mathcal{F} is the standard Fourier transform. Since the filter in the spatial domain is hard to obtain, a different approach to calculate the Gram matrix in the Fourier domain is proposed. By the Parseval theorem and equation (3.11) we have that:

$$\begin{aligned} \langle \psi_i | \psi_j \rangle_A &= \langle \mathcal{F}(\psi_i) | \mathcal{F}(\psi_j) \rangle_A = \langle \frac{1}{1 + \gamma^2 \|\omega\|^2} \mathcal{F}(\phi_i) | \mathcal{F}(\phi_j) \rangle_{\mathbb{L}_2} = \\ &= \langle \frac{1}{1 + \gamma^2 \|\omega\|^2} | \mathcal{F}(\phi_i) \mathcal{F}(\phi_j) \rangle_{\mathbb{L}_2} = \langle \mathcal{F}^{-1}(\frac{1}{1 + \gamma^2 \|\omega\|^2}) | \phi_i * \phi_j \rangle_{\mathbb{L}_2} \end{aligned} \quad (3.19)$$

which boils down to taking a Gaussian derivative of $\mathcal{F}^{-1}(\frac{1}{1 + \gamma^2 \|\omega\|^2})$ at the origin using kernel $\phi_{(m_i + m_j, n_i + n_j, \xi_i - \xi_j, \eta_i - \eta_j, s_i + s_j)}$. The reconstruction function is also created in the Fourier domain so that the algorithm for the Sobolev reconstruction now is as follows:

Algorithm 3.3: Sobolev reconstruction

1. Calculate the feature vector \underline{c} . (For all N points calculate c_i of equation 3.1).
 2. Calculate the kernel $\mathbf{K} = \mathcal{F}^{-1}(\frac{1}{1 + \gamma^2 \|\omega\|^2})$ using FFT.
 3. Calculate the Gram matrix \mathbf{G} using Gaussian derivatives of the kernel \mathbf{K} .
 4. Calculate the normalization matrix \mathbf{S} with elements $\frac{1}{\sqrt{G_{ij}}}$ on the diagonal and 0 otherwise.
 5. Solve the linear system $(\mathbf{SGS})\underline{x} = \mathbf{S}\underline{c}$ using Singular Value Decomposition or other method.
 6. Build the reconstruction function using FFT $g(x, y) = \mathcal{F}^{-1}(\sum_i x_i S_{ii} \frac{1}{1 + \gamma^2 \|\omega\|^2} \mathcal{F}(\phi_i(x, y)))$.
 7. Sample the reconstruction function $g(x, y)$.
-

Implementation Notes

Speedup tricks

The previously described reconstruction algorithms can be improved in terms of speed using the following tricks. Consider the kernel ϕ of equation (3.7) and up to 4-th order derivatives for each point i . Since

$$\langle \phi_i | \phi_j \rangle = \langle \phi_j | \phi_i \rangle \quad (3.20)$$

the Gram matrix will be symmetric and hence a factor 2 can be saved calculating only the upper or lower half. Furthermore

$$\langle \phi_i | \frac{\partial}{\partial x^k} \phi_j \rangle = - \langle \frac{\partial}{\partial x^k} \phi_i | \phi_j \rangle \quad (3.21)$$

which means that for every point i there are pairwise symmetries in the derivatives. For up to m -th order derivatives, instead of $\frac{1}{4}(m+1)^2(m+2)^2$ inner products only $\frac{1}{2}(2m+1)(2m+2)$ (all combinations of derivatives) inner products have to be calculated. For up to 4-th order this means 45 inner products per point instead of 225 inner products per point. In practice one should calculate all possible combinations once and use a lookup table to place the parts in the Gram matrix.

Considering derivatives of the Gaussian, another implementation trick can be applied. Since $\frac{d}{dx}e^x = e^x$ we have that:

$$\frac{\partial}{\partial x^k} \phi_i(x, y) = \frac{\partial}{\partial x^k} \left(-\frac{x^2 + y^2}{4t} \right) \phi_i(x, y) \quad (3.22)$$

This means that for every derivative of ϕ_i there is a static part (ϕ_i itself) and a non-static polynomial part. Using Hermite polynomials this trick can be done with all derivatives of a Gaussian. In practice the fastest method is to hard-code these polynomials and calculate the ϕ_i only once per point.

Implementation Notes

Stability

In theory the Sobolev reconstruction for $\gamma = \infty$ should be close to the Brownian reconstruction, since the first term of (3.17) can be neglected. In practice however it is shown that there is an optimal γ and that for large γ the reconstruction can be even worse than the standard linear reconstruction. The reason lies in the stability of the linear system that has to be solved, which can be expressed in the condition number of the Gram matrix (quotient of largest and smallest eigenvalue of the Gram matrix).

When two interest points are close to each other and have similar derivatives the system will become more dependent. This will result in a bad condition number of the Gram matrix and a hard to solve linear system. One solution is to use Singular Value Decomposition (SVD) to remove near dependencies. However, this also means that information about the points is thrown away, which results in loss of information in the reconstruction. In practice one should carefully choose the threshold for the SVD to minimize errors in the inverse calculation of the Gram matrix, while maintaining as much information of the image as possible.

For the Sobolev reconstruction the condition number of the matrix is however dependent on γ . Algorithm 3.3 shows that in practice the Gaussian derivative of kernel K is calculated using a Gaussian kernel of scale $s = s_i + s_j$ located at $(x_i - x_j, y_i - y_j)$. Figure 3.5 shows the kernel K for some values of γ . Since the amplitude drops with increasing γ , the Gaussian derivatives at different positions will become more similar and due to limited machine precision, the condition number of the Gram matrix will become worse with increasing γ .

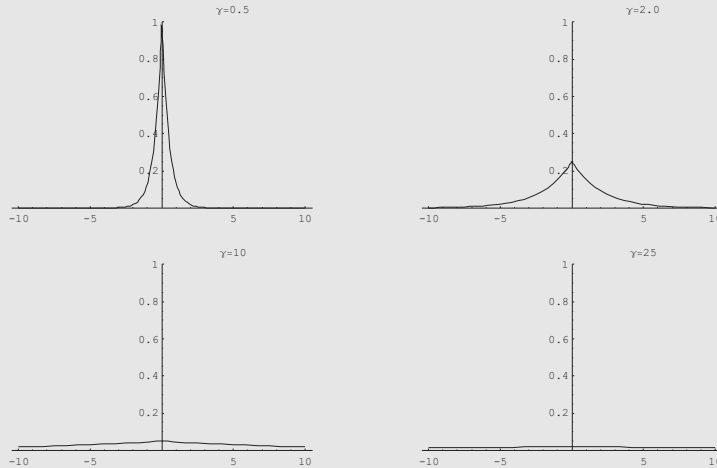


Figure 3.5: Kernel K in the spatial domain for $\gamma = 0.5$, $\gamma = 2.0$, $\gamma = 10$ and $\gamma = 25$.

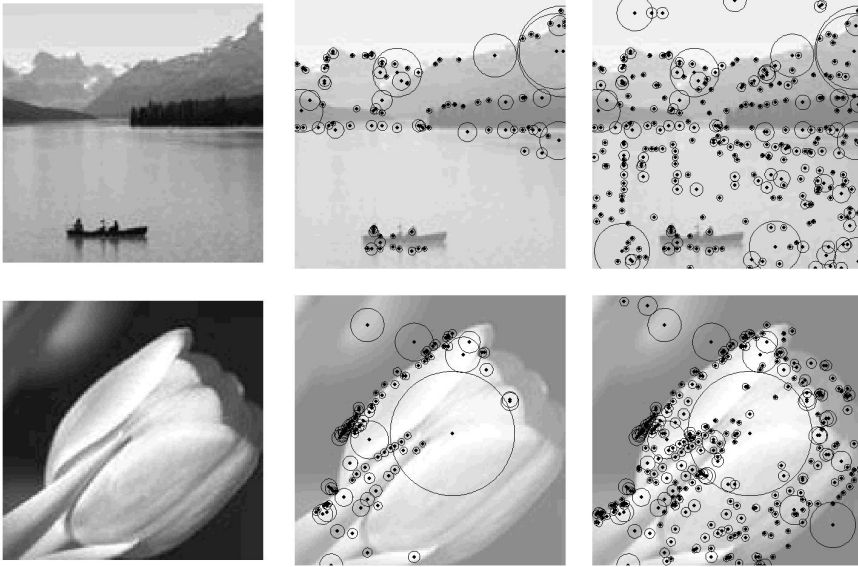


Figure 3.6: Two example images used for reconstruction. Top row, from left to right: *Canada1.jpg* test image (128×128 pixels) , 100 strongest corner points projected on the original image (the size of the circles represents the scale of the point) and 300 strongest corner points projected on the original image. Bottom row, from left to right: *Tulip.jpg* test image (128×128 pixels), 100 strongest Laplacian blobs projected on the original image and 300 strongest Laplacian blobs projected on the original image. For more details on the extraction of interest points from images, the reader is referred to chapter 5.

3.3 Evaluation of the reconstruction algorithms

In this section some experiments are presented that demonstrate the properties of the presented reconstruction algorithm. For these experiments two sample images are used that are shown in figure 3.6, *canada1.jpg* and *tulip.jpg*. More experiments using a set of 8000 image patches of the van Hateren database [134] and a set of 12 digital images of natural scenes are presented in chapter 5. For the *canada1* image 458 corner points were detected (for more information on the extraction of interest points from images the reader is referred to chapter 5) and for the *tulip* image 437 Laplacian blobs were detected (again we refer to chapter 5 for details). Subsets of these two point sets (shown in figure 3.6) are used to reconstruct the original images using several parameters and reconstruction algorithms. Gaussian filter responses with up to 4-th order derivatives located at the interest points were used as constraints for the reconstruction algorithms (see section 3.2 for details). The reconstruction results are evaluated using two different error measures, the Root Mean Square (RMS) error and the Multi-Scale Differential Error (MSDE). More details on these error measures can be found in chapter 4.

3.3.1 Evaluation of the γ -parameter

The proposed Sobolev type inner product reconstruction algorithm (further referred to as Sobolev reconstruction) has a free parameter γ that sets the *smoothness* of the image. For $\gamma = 0$ the Sobolev reconstruction equals the standard linear reconstruction while for $\gamma = \infty$ the Sobolev reconstruction equals the Brownian reconstruction (since the first term of equation (3.17) can be neglected for $\gamma = \infty$). In practice however, there are some limitations on γ because of limited machine precision as is shown in the implementation notes “Stability”. This section shows experiments using the two images with varying γ . The experiments are performed using our Mathematica implementation, where a higher precision is used at the cost of lower speed (currently the Mathematica implementation uses more than one hour for a 4-th order reconstruction of 100 points, while our less accurate C++ implementation uses less than one minute for the same reconstruction on the same machine). For the remainder of this thesis all experiments are performed using our C++ implementation, unless noted otherwise. To show the actual influence of the *prior* of the reconstructions, the subset of 100 points is used. Figure 3.7 shows some results of the Sobolev reconstruction using various γ . The reconstructions of varying γ are evaluated using the RMS error and the MSDE (for details, see chapter 4). Figure 3.8 shows the resulting graphs. Note that first the error decreases with increasing γ , while for higher γ the error increases for the *canada1* image due to limited machine precision (also visible in figure 3.7 on the two top right images). Another interesting effect is the difference between the RMS and the MSDE, especially for the *tulip* image. The MSDE is much smoother which actually better represents the visual differences between the reconstructed images as is also found in chapter 4. The small jump in the MSDE for the *canada1* image is due to numerical errors in the reconstruction algorithm that suddenly cause a large blob to change the mean gray-scale value of a large part of the image. For the remainder of this thesis $\gamma = 20$ is used for our experiments, since this gives a reasonable smoothing, while the C++ algorithm will still give stable results.

3.3.2 Evaluation of the SVD settings

The inverse of the Gram matrix used in the Sobolev reconstruction is calculated using the Singular Value Decomposition (SVD). Before calculating the actual inverse of the decomposition, equations corresponding to small singular values are removed. Equations are removed if the corresponding singular value is smaller than ϵ times the largest singular value, where $\epsilon < 1$ is referred to as the SVD tolerance. The higher the tolerance, the more equations are removed from the linear system and the more stable the inverse becomes (which means less numerical errors in the inverse). However, image information is lost while removing equations and thus the lower the tolerance, the more image information is maintained. In practice one should carefully choose the SVD tolerance to obtain a stable inverse, while maintaining as much image information as possible. In this section some results are presented of reconstructions using various settings for the SVD tolerance. Figure 3.9 shows some results of reconstructions using SVD tolerances of 10^{-1} , 10^{-3} , 10^{-5} and 10^{-8} respectively. Note that the images using a tolerance of 10^{-1} lack some detail, while the images using a tolerance of 10^{-8} contain large artifacts due to numerical errors. Figure 3.10 and figure 3.11 show the errors of the reconstructions for various settings of the tolerance for the *canada1* image and the *tulip* image respectively. Note that indeed first the error decreases with decreasing tolerance, but increases if the

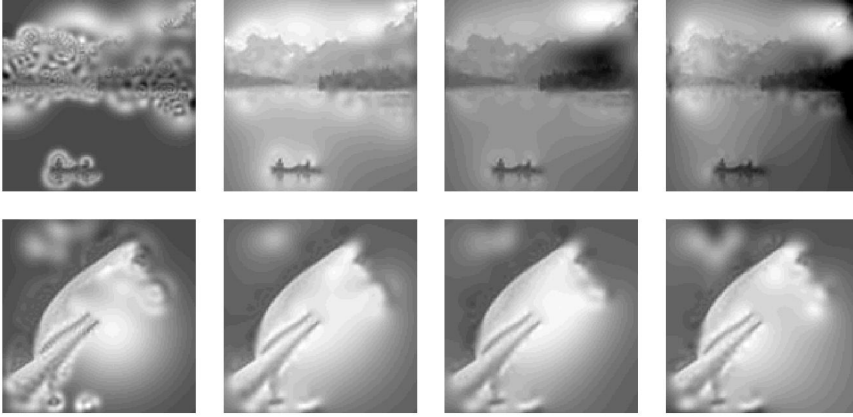


Figure 3.7: Reconstruction results using varying γ . The top row shows reconstructions of the *canada1* image using up to 4-th order derivatives in 100 corner points with $\gamma = 1$, $\gamma = 25$, $\gamma = 50$ and $\gamma = 100$ respectively. Note the large structures at the top-right part of the images for high γ due to numerical errors. The bottom row shows reconstructions of the *tulip* image using up to 4-th order derivatives in 100 Laplacian blobs with $\gamma = 1$, $\gamma = 25$, $\gamma = 50$ and $\gamma = 100$ respectively.

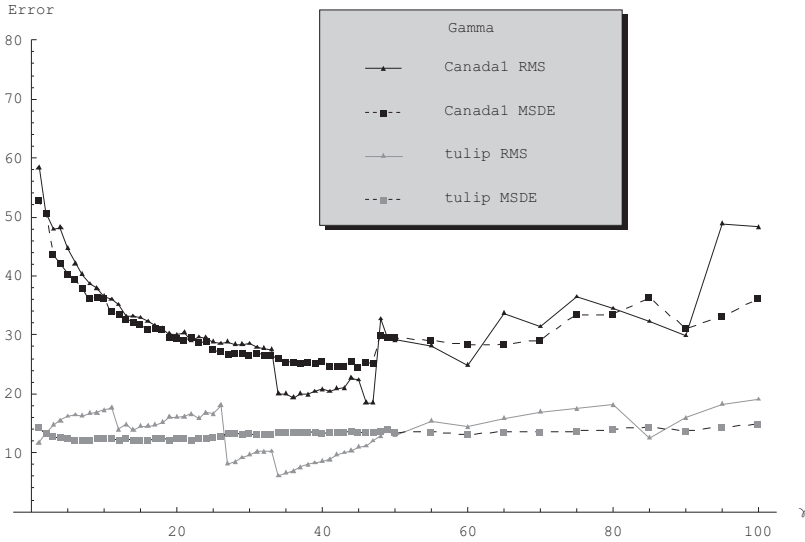


Figure 3.8: Reconstruction errors for varying γ . Reconstructions of the *canada1* image were made using up to 4-th order derivatives in 100 corner points while reconstructions of the *tulip* image were made using up to 4-th order derivatives in 100 Laplacian blobs. Note that first the error decreases with increasing γ , while for higher γ the error increases for the *canada1* image due to limited machine precision. The *tulip* image does not suffer from these numerical errors and shows a more constant MSDE for high γ . Also note that the MSDE is much smoother which actually better represents the visual differences between the reconstructed images as is also found in chapter 4.

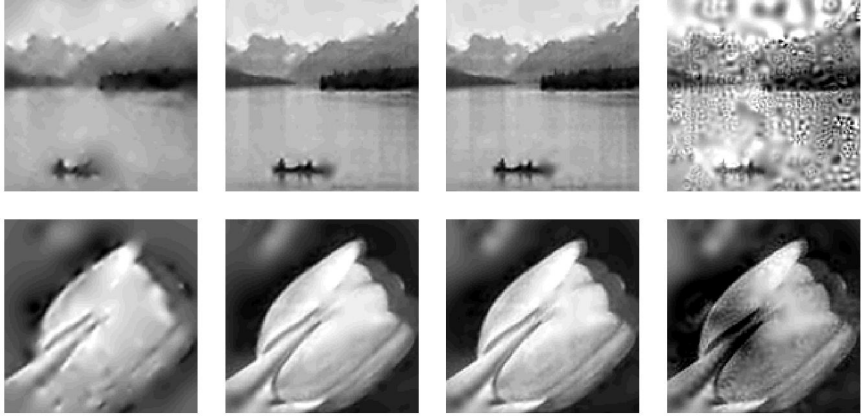


Figure 3.9: Reconstruction results using varying SVD tolerance. The top row shows reconstructions of the *canada1* image using up to 4-th order derivatives in 100 corner points with tolerance $\epsilon = 10^{-1}$, $\epsilon = 10^{-3}$, $\epsilon = 10^{-5}$ and $\epsilon = 10^{-8}$ respectively. The bottom row shows reconstructions of the *tulip* image using up to 4-th order derivatives in 100 Laplacian blobs with tolerance $\epsilon = 10^{-1}$, $\epsilon = 10^{-3}$, $\epsilon = 10^{-5}$ and $\epsilon = 10^{-8}$ respectively. Note the lack of detail in the images with tolerance $\epsilon = 10^{-1}$ and the large artifacts due to numerical errors for images with tolerance $\epsilon = 10^{-8}$.

tolerance is too small. The graphs also show the percentage of equations that have been removed by the SVD inverse. In cases where no equations are removed, the error is much larger due to numerical errors. For further experiments a value of 10^{-5} for the SVD tolerance is used since this seems to result in stable reconstructions while maintaining as much image information as possible.

3.3.3 Evaluation of the number of reconstruction points

The quality of the reconstruction will be dependent on the amount of constraints and the quality of the prior. The less constraints, the more important the prior becomes. The constraints used for the reconstruction are Gaussian (derivative) filter responses located in scale space interest points. In this section experiments using a varying number of points are presented. The points are ranked by strength or differential TV-Norm (see chapter 5 for details). For the reconstructions in this section, up to 4-th order derivatives are used as features in each interest point. Figure 3.12 shows some example reconstructions with 10, 50, 150 and 400 points respectively. As one would expect the reconstruction quality increases with the number of reconstruction points. Figure 3.13 shows the error of reconstructions with varying number of reconstruction points. Note that for the *canada1* image the error stabilizes after approximately 150 points. Most of the image information is thus contained in the first 150 corner points. With the *tulip* image this number is much higher (and not present in the graph). In practice for these 128×128 images using up to 4-th order derivatives, approximately 300 points will give reasonable reconstructions.

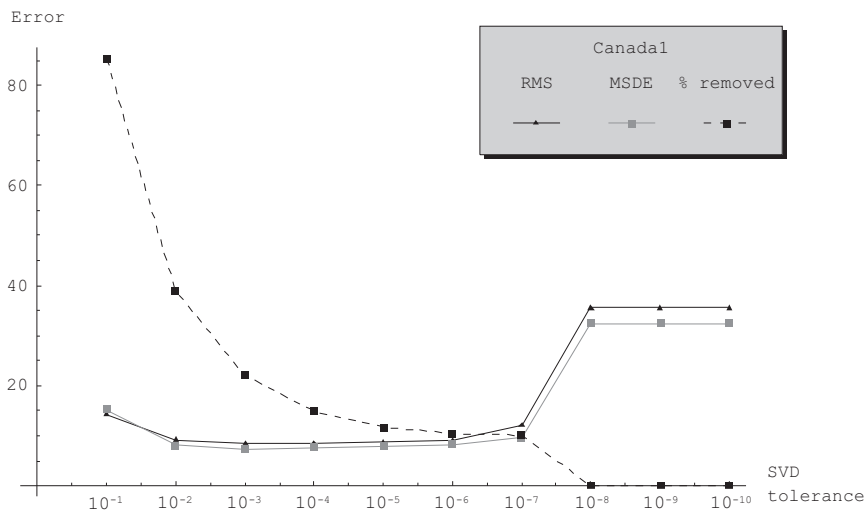


Figure 3.10: Reconstruction errors for varying SVD tolerance for the *canada1* image. The dashed line shows the percentage of removed equations for a given tolerance. Note that first the error decreases with decreasing tolerance, but increases if the tolerance is too small. In cases where no equations are removed, the error is much larger due to numerical errors.

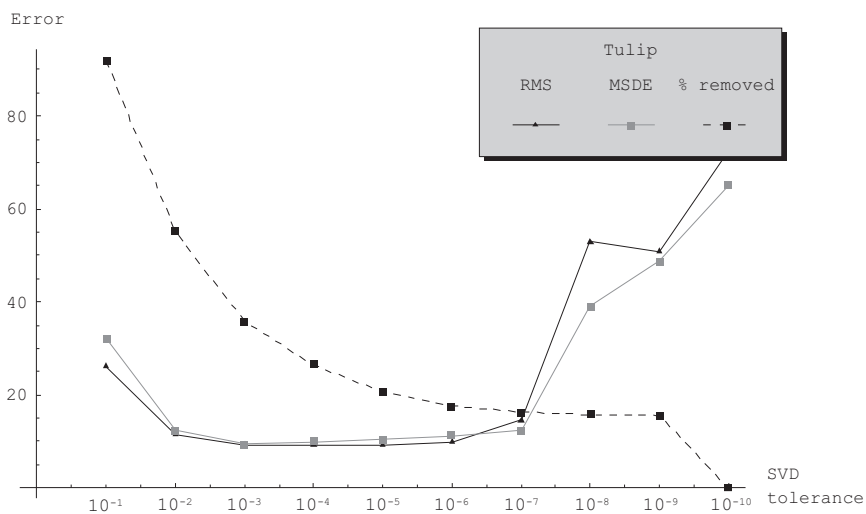


Figure 3.11: Reconstruction errors for varying SVD tolerance for the *tulip* image. The dashed line shows the percentage of removed equations for a given tolerance. Note that first the error decreases with decreasing tolerance, but increases if the tolerance is too small. In cases where no equations are removed, the error is much larger due to numerical errors.

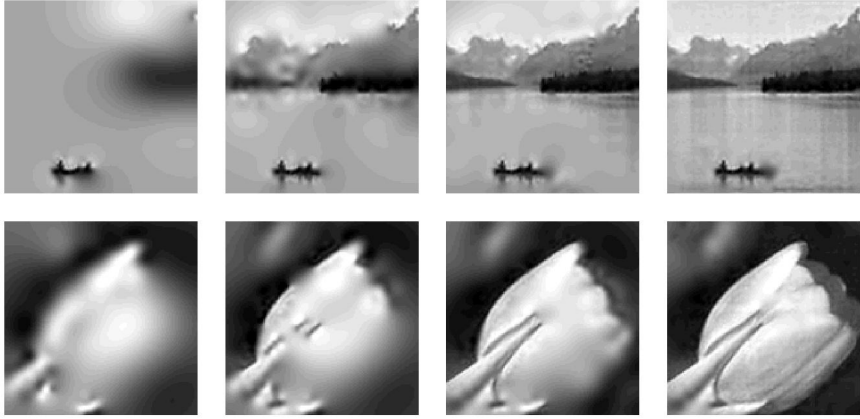


Figure 3.12: Reconstruction results using a varying number of scale space interest points. The top row shows reconstructions of the *canada1* image using up to 4-th order derivatives in 10, 50, 150 and 400 corner points respectively. The bottom row shows reconstructions of the *tulip* image using up to 4-th order derivatives in 10, 50, 150 and 400 Laplacian blobs respectively. As expected the image quality improves with an increasing number of interest points, since the metamer class is more constrained.

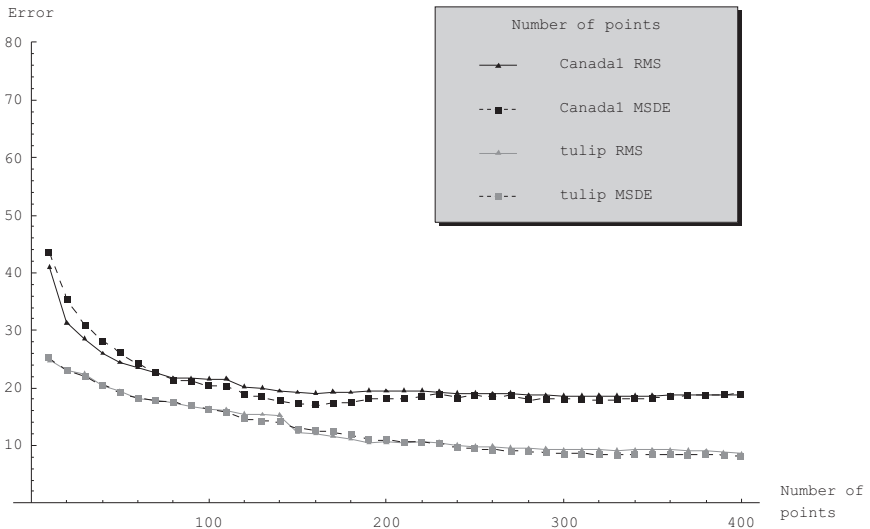


Figure 3.13: Reconstruction errors for a varying number of scale space interest points. Reconstructions of the *canada1* image were made using up to 4-th order derivatives in each corner point while reconstructions of the *tulip* image were made using up to 4-th order derivatives in each Laplacian blob point. Note that for the *canada1* image the error stabilizes after approximately 150 points. Most of the image information is contained in the first 150 corner points. With the *tulip* image this number is higher (and not present in the graph).

3.3.4 Evaluation of the reconstruction order

The number of constraints used for the reconstruction is not only dependent on the number of interest points. The amount of local information taken into account in each interest point is also important. In this section a number of experiments is performed showing the influence of the order of derivatives taken into account in each interest point. Figure 3.14 shows example reconstructions using derivatives up to order 0,1,2,3 and 4 respectively in each interest point (which means that in each point respectively 1, 3, 6, 10, 15, 21 features are used as constraints). The subsets of 100 interest points are used to show the influence of the features in each point since a high number of points can compensate a low number of features. Figure 3.15 shows the error of the reconstructions using varying orders of derivatives. Note that the error decreases, but the difference between up to 4-th order and up to 5-th order is relatively small while 40% of information is added. So the ratio between the error and the amount of information added decreases fast. In practice up to 4-th order will be sufficient if the number of points is not too low.

3.3.5 Stability with respect to noise on the position of the points

The reconstruction algorithm works with any type of filter response. In principle one could use random points in scale space and calculate Gaussian derivatives in those points as features. However, some points in scale space contain more image information than others. This can be demonstrated by comparing reconstructions from scale space interest points with reconstructions from perturbed interest points. In this section some experiments are shown using additive noise on the position of the scale space interest points with varying standard deviation σ . The features are recalculated in each point to simulate the effect of errors in the detection of the points. Figure 3.16 shows some example reconstructions using additive noise on the interest point position with a standard deviation of $\sigma = 0$, $\sigma = 5$, $\sigma = 12$ and $\sigma = 20$ pixels respectively. Note that indeed the image quality decreases with increasing noise. Figure 3.17 shows the errors for reconstructions using additive noise on the interest point positions. Note that the error increases with increasing noise. However, the visual quality seems to degrade faster than one would expect looking at the error. The peak in the graph for the *canada1* image is due to artifacts introduced by numerical errors since the linear system can get nearly dependent when noise is added. The conclusion is that there are indeed points in scale space that contain more image information than others. In chapter 5 an extensive evaluation is given using 10 different types of scale space interest points from a set of 8.000 image patches.

3.3.6 Stability with respect to noise on the features

The reconstruction from scale space interest points and some local features in those points relies on the accuracy of those features. In this section the influence of noise on the features in the reconstruction points is investigated. Multiplicative noise with mean 0 and standard deviation σ is added to the feature vector after calculation of the derivatives in each interest point. The use of multiplicative noise is due to the fact that the derivatives are not normalized and have different ranges in amplitude. Additive noise would in this case have much influence on high order derivatives and lower influence on low order derivatives. Figure 3.18 shows some reconstruction examples using noise on

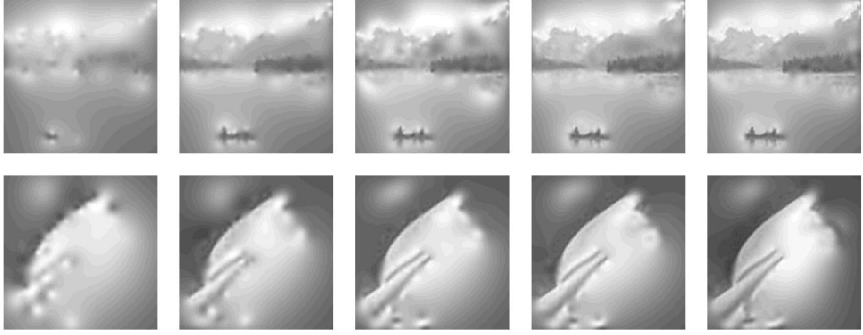


Figure 3.14: Reconstruction results using a varying number of features in each interest point. The top row shows reconstructions of the *canada1* image using 100 corner points and derivatives up to order 0,1,2,3 and 4 in each corner point respectively. The bottom row shows reconstructions of the *tulip* image using 100 Laplacian blobs and derivatives up to order 0,1,2,3 and 4 in each corner point respectively. As expected the image quality improves with an increasing number of features per point, since the metameric class is more constrained. Note that the difference between up to 3-rd order and up to 4-th order is hard to see.

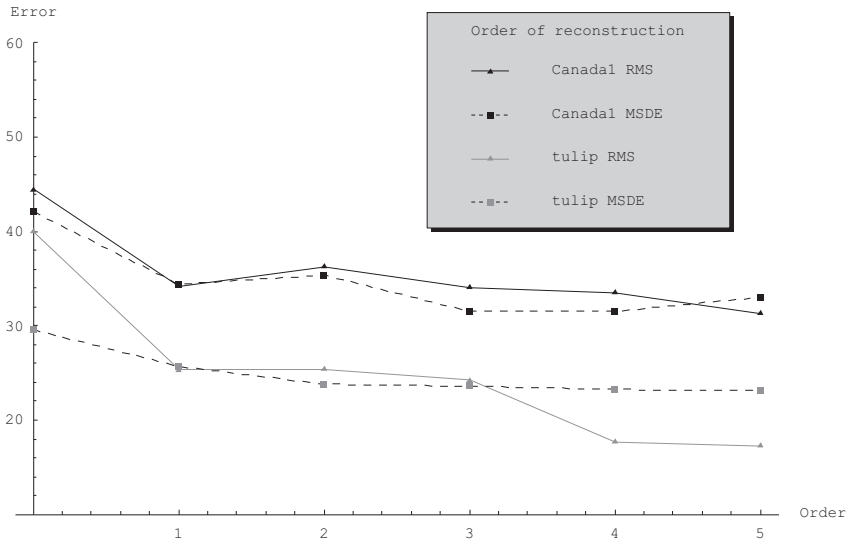


Figure 3.15: Reconstruction errors for a varying number of features in each interest point. Reconstructions of the *canada1* image were made using 100 corner points while reconstructions of the *tulip* image were made using 100 Laplacian blob points. Note that the error decreases with increasing order, but the difference becomes small for orders higher than 4.

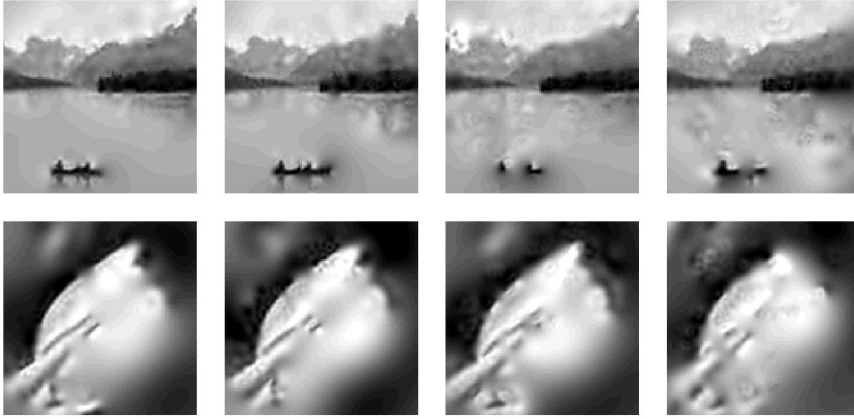


Figure 3.16: Reconstruction results using additive noise on the interest point positions using varying standard deviation σ . The top row shows reconstructions of the *canada1* image using 300 corner points and up to 4-th order derivatives in each corner point. The bottom row shows reconstructions of the *tulip* image using 300 Laplacian blobs and up to 4-th order derivatives in each Laplacian blob point. From left to right the additive noise on the position of the interest points has a standard deviation of $\sigma = 0$, $\sigma = 5$, $\sigma = 12$ and $\sigma = 20$ respectively.

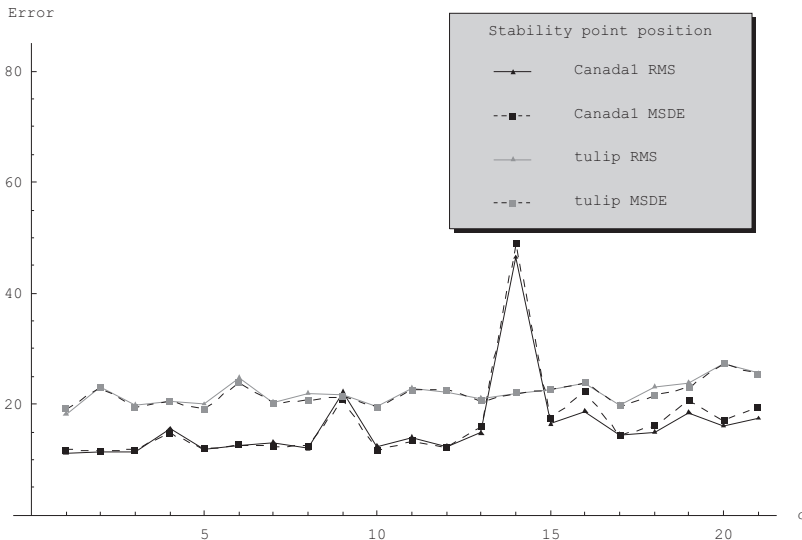


Figure 3.17: Reconstruction errors using additive noise on the interest point positions using varying standard deviation σ . Reconstructions of the *canada1* image were made using 100 corner points with up to 4-th order derivatives in each point while reconstructions of the *tulip* image were made using 100 Laplacian blob points with up to 4-th order derivatives in each point. Note that the error increases with increasing noise, but the visual image quality seems to degrade more than one would expect looking at the error.

the features with a standard deviation of $\sigma = 0$, $\sigma = 0.025$, $\sigma = 0.060$ and $\sigma = 0.1$ respectively. Note that the noise is added to all features at once and the influence of noise on the features is much higher than noise on the position of the interest points. Furthermore, image information is perturbed in the case of noise on the features, while this is not the case for noise on the position of the interest points since the features are recalculated there. Figure 3.19 shows the errors for noise on the features for various σ . Note that indeed the error increases fast when the standard deviation of the noise increases. In practice this means that accurate calculation of the features in each interest point is important.

3.3.7 Comparison of the proposed algorithms

The different point sets of the two example images shown in figure 3.6 are also used to compare the three reconstruction algorithms presented. The three algorithms differ in that they select a different image from the metamer class based on a different *prior*. One would expect the prior to have more influence when the number of constraints is low. Therefore reconstructions from both the 100 point subsets and the 300 point subsets are compared for the three reconstruction algorithms. Figure 3.20 shows the reconstruction results for the three reconstruction algorithms using the two interest point subsets. Note that the standard linear reconstruction performs worse than the Brownian and the Sobolev reconstruction in all cases. However, the difference between the Sobolev and the Brownian reconstructions is much smaller, but still in favor of the Brownian reconstruction (that is with the current implementation). Note that the visual quality of the reconstructions is not correctly reflected in the RMS error. The MSDE better reflects the difference in visual quality between the Brownian and Sobolev reconstructions. Also note that the influence of the prior is more clearly visible in the 100 point reconstructions. Table 3.1 shows the corresponding errors.

	standard linear reconstruction		Sobolev reconstruction		Brownian reconstruction	
Image	RMS	MSDE	RMS	MSDE	RMS	MSDE
<i>Canada1</i> , 100 pts	14.83	0.4988	8.56	0.3600	9.88	0.3384
<i>Canada1</i> , 300 pts	7.25	0.2657	6.01	0.2257	5.36	0.1756
<i>Tulip</i> , 100 pts	25.39	0.8886	16.39	0.6998	17.24	0.6142
<i>Tulip</i> , 300 pts	12.11	0.3896	6.04	0.2757	6.39	0.2018

Table 3.1: RMS and MSDE errors of the three different reconstruction algorithms.

3.4 Conclusion and discussion

To use scale space interest points for image editing, it is important to know how much image information is available in such a point set. One method to make the image information of a point set explicit is image reconstruction. If it is possible to create a good reconstruction from a set of scale space interest points, that set must contain almost all image information. Two existing reconstruction algorithms are presented in this chapter:

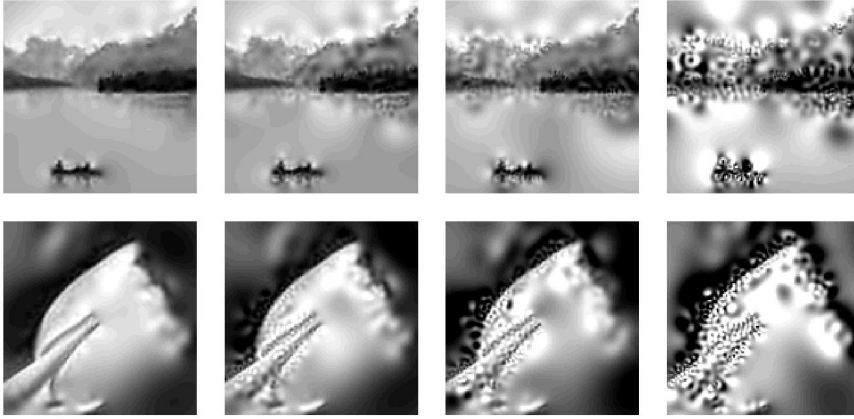


Figure 3.18: Reconstruction results using multiplicative noise on the features of each interest point with varying standard deviation σ . The top row shows reconstructions of the *canada1* image using 300 corner points and up to 4-th order derivatives in each corner point. The bottom row shows reconstructions of the *tulip* image using 300 Laplacian blobs and up to 4-th order derivatives in each Laplacian blob point. From left to right the multiplicative noise on the features in each interest point has a standard deviation of $\sigma = 0$, $\sigma = 0.025$, $\sigma = 0.060$ and $\sigma = 0.1$ respectively. Note that the visual quality decreases fast when noise is added to the features.

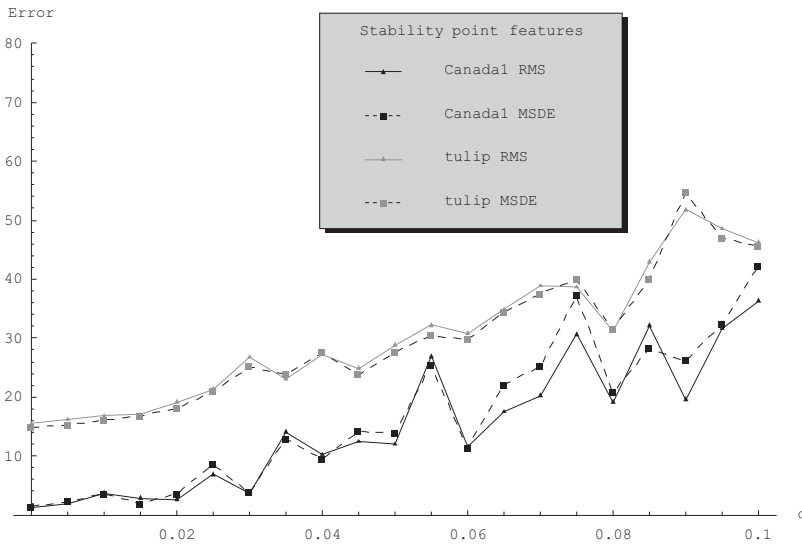


Figure 3.19: Reconstruction errors using multiplicative noise on the features of each interest point with varying standard deviation σ . Reconstructions of the *canada1* image were made using 300 corner points with up to 4-th order derivatives in each point while reconstructions of the *tulip* image were made using 300 Laplacian blob points with up to 4-th order derivatives in each point. Note that the error increases fast when the standard deviation of the noise increases.

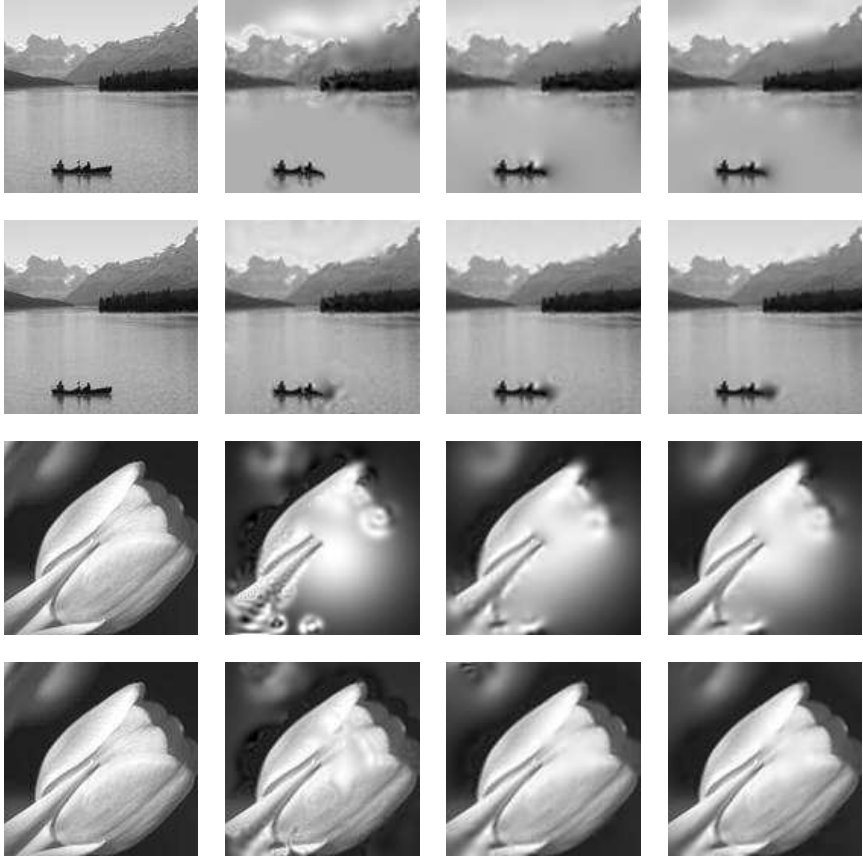


Figure 3.20: Reconstructions from scale space interest points and local features of two example images using three different reconstruction algorithms. From top to bottom: Reconstruction from 100 corner points of the *Canada1* image, reconstruction from 300 corner points of the *Canada1* image, reconstruction from 100 Laplacian blobs of the *Tulip* image and reconstruction from 300 Laplacian blobs of the *Tulip* image. From left to right: Original image, standard linear reconstruction, Sobolev reconstruction and Brownian reconstruction.

one simple linear reconstruction algorithm that is fast, but has visually unappealing results for a low number of points and a more sophisticated iterative reconstruction scheme that is slower, but gives a more appealing result for a low number of points. We presented a third method that is still linear, but the results are visually close to the iterative reconstruction scheme.

A number of experiments is performed to show the properties of the newly introduced reconstruction algorithm. This algorithm has a free parameter $\gamma > 0$ that represents the amount of blurring between the feature points. In theory setting this value to $\gamma = \infty$ would lead to equal results as the iterative reconstruction, but in practice due to numerical errors, one should not make the γ parameter too high. A reasonable value for our C++ implementation is $\gamma = 20$ to $\gamma = 25$. Another parameter that is of influence is the SVD tolerance. In theory this value should be small to maintain all image information in the points, but the numerical precision of the computer limits this. In our experiments a value of $\epsilon = 10^{-5}$ gives good results.

The quality of the reconstruction depends mostly on the amount of constraints that is used. In our experiments we used two types of scale space interest points and local derivatives up to 4th order in these points as constraints. Experiments using more images and more types of scale space interest points are presented in chapter 5. In the experiments of this chapter we showed that a larger number of interest points gives a better reconstruction but that the improvement is slowly decaying. A number of 300 scale space interest points seemed optimal for our 128×128 images. The same holds for the order of derivatives included in the reconstruction. The higher number of orders included, the better the reconstruction. However, again the improvement will decay and in practice using up to 4-th order derivatives for our test images was sufficient. It is also shown that adding noise to the position of the scale space interest points does affect the reconstruction quality. This shows that reconstruction from scale space interest points gives better results than just random points. With respect to the features it is shown that adding noise to the features greatly degrades the reconstruction quality and that one should thus detect the features as accurate as possible.

Finally it is shown that the proposed linear reconstruction algorithm performs very close to the existing iterative reconstruction algorithm and outperforms the old standard linear reconstruction algorithm. It is also shown that scale space interest points and some local properties of these points can contain a substantial amount of image information. More experiments showing this can be found in chapter 5. There are several improvements possible for the reconstruction algorithm. Currently there are boundary issues, since for the measurement filters as well as for the reconstruction filters boundary conditions have to be chosen for points close to the border of the image. This means assumptions have to be made about the outside of the image. A possible solution is to include the boundary of the image itself as a constraint. The flux features presented by Duits [29] could also result in a big improvement since for certain areas properties can be set instead of for single points. The question is however how this would affect the stability of the linear system. One should also reconsider the way constraints are used in the current implementation. Currently all points share the same type of feature, e.g. all points have up to 4-th order derivatives as features. It makes sense to make an implementation able of having different types of features for different types of interest points, for example first order derivatives for edge points and zeroth order derivatives for blobs or scale related as presented by Florack [42].

4

Image Quality Measures

This chapter is based on:

Comparison of Image Quality Measures for Evaluating Image Reconstruction.
F.M.W. Kanters, L.M.J. Florack, B. Platel and B.M. ter Haar Romeny.
In preparation.

Multi-Scale Differential Error: A Novel Image Quality Assessment Tool.
F.M.W. Kanters, L.M.J. Florack, B. Platel and B.M. ter Haar Romeny.
Proc. of the 8th Int. Conf. on Signal and Image Processing 2006, Honolulu, Hawaii, pp 188–194,
August 2006.

4.1 Introduction

In the previous chapter we have introduced three reconstruction algorithms that are able to reconstruct an image from multi scale interest points and their local features. To compare the three algorithms and to evaluate their properties, the *image quality* of the reconstructed images must be quantified in a reproducible manner. Image quality can be quantified in two ways: Using subjective measures (e.g. human observers, not reproducible and not quantitative) or using objective measures (e.g. computer algorithms, reproducible and quantitative). Some studies that address the problem of how to obtain subjective measurements on images can be found in the work of Hamberg and de Ridder [57] and the ISO20462 standard [76]. Objective image quality measures can roughly be divided into two groups: Generic mathematical measures and measures specifically based on (complex) models of the human visual system (HVS). There are several papers comparing generic mathematical measures with more sophisticated measures and with subjective measures, mostly in the field of image compression [52, 48, 34, 33, 107, 31, 35, 6, 28], halftone printing [106, 10, 11], CRT display quality [4, 131, 121] or specific for color imaging [150, 1, 62, 90, 70]. A more recent application involves measuring image quality of synthetic (rendered) images [22, 124, 18, 101]. A very comprehensive comparison of 26 different error measures can be found in the work by Avcibas [2]. In 2004 Wang and Simoncelli introduced a method to evaluate the difference between two image quality metrics [151]. One of the main problems in computer vision is to relate the objective measures with subjective measures. In 1996 Martens and Kayargadde presented a method to relate objective measures and subjective measures [105] by introducing a perceptual space and a psychometric space.

The problem with the current literature is however that it is biased to specific applications with aims very different from image reconstruction from multi-scale interest points. Compression for example has only limited types of artifacts (e.g. blocking or color-shift) which are usually known a priori. Image reconstruction from scale space interest points however has many types of artifacts which are not known a priori since they depend on the reconstruction algorithm, the interest points and their features. For this reason many commonly used objective quality measures will not be similar to how a human observer would rate the image quality. In section 4.2 we present 31 commonly used generic objective image quality measures, a new scale space image quality measure is introduced and 3 more complex human visual system based quality measures are presented. Section 4.3 presents a set of experiments comparing a human observer study with all presented objective image quality measures. Finally conclusions and discussion are given in section 4.4.

4.2 Objective image quality measures

In this section a number of objective image quality measures is presented. First some generic mathematical measures are defined and finally some more complex HVS based measures are presented. Note that some of the methods described are based on heuristics and contain choices that are not always obvious. Some methods are even unsuitable for a large class of images, but are included for completeness sake. We present these methods as described in the respective references, without judging the credibility of the methods. First we define our reference image $f[i, j]$ and some distorted image $g[i, j]$ with $1 \leq i \leq M$

and $1 \leq j \leq N$. All image values of f and g are scaled to the unit interval $(0,1)$.

4.2.1 Generic mathematical measures

4.2.1.1 MSE

Given the definition of reference image f and distorted image g we define the Mean Squared Error (MSE) as:

$$MSE(f, g) = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (f[i, j] - g[i, j])^2 \quad (4.1)$$

4.2.1.2 RMSE or RMS

The well known Root Mean Squared Error is defined as the square root of the MSE:

$$RMSE(f, g) = \|f - g\|_2 = \sqrt{\frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (f[i, j] - g[i, j])^2} \quad (4.2)$$

The RMS error is one of the most commonly used image quality measures mostly because its simplicity.

4.2.1.3 RRMSE

The Relative Root Mean Squared Error is defined as:

$$RRMSE(f, g) = \sqrt{\frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N \frac{(f[i, j] - g[i, j])^2}{f[i, j]^2}} \quad (4.3)$$

Note that if $f[i, j] = 0$ for any i, j the $RRMSE = \infty$, which makes this error measure useless for many types of images.

4.2.1.4 MI

The Mutual Information is defined as:

$$MI(f, g) = \sum_{u \in U} \sum_{v \in V} p_{fg}\{u, v\} \log \frac{p_{fg}\{u, v\}}{p_f\{u\}p_g\{v\}} \quad (4.4)$$

with U and V the set of gray scale values of f and g respectively. For 8-bits images with normalized values, $U = V = \{0, \frac{1}{255}, \frac{2}{255}, \dots, \frac{254}{255}, 1\}$, $p_f\{u\}$ is the number of pixels in f with value u divided by the total number of pixels in f , $p_g\{v\}$ is the number of pixels in g with value v divided by the total number of pixels in g and finally $p_{fg}\{u, v\}$ is the number of pixels in f with value u and in g with value v divided by the total number of pixels in f . The Mutual Information results in a value between 0 and the entropy of image f (when f and g are the same). To obtain a value between 0 and 1 we normalize the MI.

4.2.1.5 SNR

The Signal to Noise Ratio between 2 images is defined by:

$$SNR_{(dB)}(f, g) = 10^{10} \log \left(\frac{\sigma_f^2}{MSE(f, g)} \right) \quad (4.5)$$

with σ_f^2 the variance of the original image f .

4.2.1.6 PSNR

Conform [123] we define the Peak Signal to Noise Ratio (for normalized images) as:

$$PSNR_{(dB)}(f, g) = 10^{10} \log \left(\frac{1}{MSE(f, g)} \right) \quad (4.6)$$

4.2.1.7 MAD

The Maximum Absolute Difference is defined by:

$$MAD(f, g) = \|f - g\|_\infty = \max_{i,j} |f[i, j] - g[i, j]| \quad (4.7)$$

4.2.1.8 RMAD

The Relative Maximum Absolute Difference is defined by:

$$RMAD(f, g) = \max_{i,j} \frac{|f[i, j] - g[i, j]|}{f[i, j]} \quad (4.8)$$

Again note that if $f[i, j] = 0$ for any pair i, j , the $RMAD = \infty$, which makes it unusable for many types of images.

4.2.1.9 MAE

The Mean Absolute Error is defined by:

$$MAE(f, g) = \|f - g\|_1 = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N |f[i, j] - g[i, j]| \quad (4.9)$$

4.2.1.10 Minkowski metric \mathbb{L}_p

The \mathbb{L}_p norm is given by:

$$\mathbb{L}_p(f, g) = \|f - g\|_p = \left(\frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N |f[i, j] - g[i, j]|^p \right)^{\frac{1}{p}} \quad (4.10)$$

with $1 \leq p \leq \infty$. Note that for $p = 1$ one obtains the MAE and for $p = 2$ one obtains the RMSE.

4.2.1.11 Modified Minkowski infinity metric

For $p = \infty$ the Minkowski metric becomes the maximum pixel difference (MAD):

$$\mathbb{L}_\infty(f, g) = \max_{i,j} |f[i, j] - g[i, j]| \quad (4.11)$$

This is however very sensitive to noise. To make it more robust to noise one can define the modified Minkowski infinity metric (cf. D3 of Avcibas [2]):

$$MMIM_r(f, g) = \sqrt{\frac{1}{r} \sum_{m=1}^r \Delta_m^2(f[i, j] - g[i, j])} \quad (4.12)$$

where $\Delta_l(f[i, j] - g[i, j])$ is the l^{th} largest deviation among the pixels. Thus $\Delta_1(f[i, j] - g[i, j])$ is the normal Minkowski infinity metric which is equal to (4.7) and (4.11) and $\Delta_2(f[i, j] - g[i, j])$ is the second largest deviation, etc. Note that r is a free parameter, which influences the number of deviations that are taken into account.

4.2.1.12 Neighborhood Difference

Image distortion can also arise from displacements of pixels. A distortion measure that penalizes spatial displacements in combination with gray level differences is introduced by DiGesu and Staravoirov [27]:

$$ND_w(f, g) = \sqrt{\frac{1}{2(M-w)(N-w)}} \cdot \sqrt{\sum_{i=w/2}^{M-w/2} \sum_{j=w/2}^{N-w/2} \{M_w^{ij}(f, g) + M_w^{ij}(g, f)\}} \quad (4.13)$$

with:

$$M_w^{ij}(f, g) = \left(\min_{l, m \in w_{i,j}} \{d(f[i, j], g[l, m])\} \right)^2 \quad (4.14)$$

where $d(\cdot, \cdot)$ is some appropriate distance metric and w is some neighborhood size, for example 3 or 5 pixels. For the distance metric one could use for example the *city block* or *Manhattan* metric:

$$d_{\alpha, \beta}^{\text{city}}(f[i, j], g[l, m]) = \frac{(|i - l| + |j - m|)}{\alpha} + \frac{|f[i, j] - g[l, m]|}{\beta} \quad (4.15)$$

where α and β are two parameters to tune the penalties due to pixel shift and gray value difference. Note that this implementation ignores the boundaries and thus $ND_w(f, g) = 0 \nRightarrow f = g$. This can be solved by properly handling the boundaries of the image.

4.2.1.13 Multi-resolution Distance Measure

Most of the presented methods only look at the finest resolution of the images. The human visual system however, takes many different scales into account when observing images. Juffs [71] introduced a multiresolution distance measure which assigns larger weights to low resolutions and smaller weights to high resolutions (details in the image). Consider the various levels of resolution denoted by $r \geq 1$, cf. Avcibas [2]. For each value of r the image is split into blocks b_1 to b_n where n depends on the scale r . For example for $r = 1$, at the lowest resolution, only one block covers the whole image characterized by its average gray level $av^{(1)}$. For $r = 2$ one has four blocks each of size $(M/2 \times N/2)$ with average gray levels $av_{11}^{(2)}$, $av_{12}^{(2)}$, $av_{21}^{(2)}$ and $av_{22}^{(2)}$. For the r -th resolution level one would have 2^{2r-2} blocks of size $(\frac{M}{2^{r-1}} \times \frac{N}{2^{r-1}})$ characterized by the block average gray levels $av_{ij}^{(r)}$, $i, j = 1, \dots, 2^{r-2}$. The average difference in gray level at the resolution r has weight $1/2^r$. If one considers a total of R resolutions, the total distance becomes:

$$MrDM(f, g) = \sum_{r=1}^R \frac{1}{2^r} \frac{1}{2^{2r-2}} \sum_{i,j=1}^{2^{r-1}} |av_{ij}^{(r)}[f] - av_{ij}^{(r)}[g]| \quad (4.16)$$

where $av_{ij}^{(r)}[f]$ and $av_{ij}^{(r)}[g]$ are the average gray levels of the corresponding blocks at resolution r in image f and image g respectively.

4.2.1.14 Structural Content

Alternatively one could measure the closeness between two images in terms of correlation or similarity between two images. In this sense these correlation or similarity based methods are complementary to the difference based methods. One similarity based measure is the Structural Content (SC, cf. C1 in [2]):

$$SC(f, g) = \frac{\sum_{i=1}^M \sum_{j=1}^N (f[i, j])^2}{\sum_{i=1}^M \sum_{j=1}^N (g[i, j])^2} \quad (4.17)$$

Note that the Structural Content makes little sense if $g[i, j] = 0$ for any pair i, j .

4.2.1.15 Normalized Cross-Correlation Measure

Another correlation measure is the Normalized Cross-Correlation Measure, cf. C2 in [2]:

$$NCC(f, g) = \frac{\sum_{i=1}^M \sum_{j=1}^N f[i, j]g[i, j]}{\sum_{i=1}^M \sum_{j=1}^N (f[i, j])^2} \quad (4.18)$$

Note that also the Normalized Cross-Correlation Measure makes little sense if $f[i, j] = 0$ for any pair i, j .

4.2.1.16 Pratt Measure

In the perception of scene content by the human visual system, edges play a major role. Therefore it is important to include edge information in image quality measures. The following two measures include edge distortion measures. For this, binary edge maps of the images are obtained using the Canny edge detector [14]. The x-derivative $f_x[i, j]$

of image $f[i, j]$ at scale σ can be found by convolving image f with the derivative of a Gaussian:

$$f_x[i, j] = (f * G_{x, \sigma})[i, j] \quad (4.19)$$

where $G_{x, \sigma}$ is:

$$G_{x, \sigma} = \frac{\partial G_\sigma}{\partial x} = \frac{-x}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (4.20)$$

Similarly the y-derivative can be obtained by:

$$f_y[i, j] = (f * G_{y, \sigma})[i, j] \quad (4.21)$$

where $G_{y, \sigma}$ is:

$$G_{y, \sigma} = \frac{\partial G_\sigma}{\partial y} = \frac{-y}{2\pi\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (4.22)$$

The gradient magnitude $|\nabla f|$ of the image at scale σ is now defined as:

$$|\nabla_\sigma f[i, j]| = \sqrt{f_x[i, j]^2 + f_y[i, j]^2} \quad (4.23)$$

To obtain the edge map of image f , the gradient magnitude image is thresholded with $T = \alpha(\max_{i,j} |\nabla_\sigma f[i, j]| - \min_{i,j} |\nabla_\sigma f[i, j]|) + \min_{i,j} |\nabla_\sigma f[i, j]|$. Finally binary thinning is applied to obtain single pixel edges. This procedure can also be applied to the reconstructed image g in a straightforward way. Give the edge map of images f and g Pratt [122] introduced a measure that both considers edge location accuracy and missing or false edge elements. It is defined as:

$$Pratt_\beta(f, g) = \frac{1}{\max(n_g, n_f)} \sum_{i=1}^{n_g} \frac{1}{1 + \beta d_i^2} \quad (4.24)$$

where n_f and n_g are the number of edge pixels of f and g respectively and d_i is the distance to the closest edge candidate for the i -th detected edge pixel in image g . Note that β is a free parameter to weigh the importance of edge dislocation in the error measure.

4.2.1.17 Edge Stability Measure

Carevic [15] defined edge stability as the consistency of edge evidences across different scales in both the original and coded images. First the gradient of both images at different scales is obtained using the Canny edge detector as described in steps (4.19)-(4.23). Conform [2] we use scales $\sigma_m = 1.19, 1.44, 1.68, 2.0, 2.38$ for $m = 1, 2, 3, 4, 5$ respectively. The output of this operator at scale m is thresholded with T^m where $T^m = 0.1(C_{max}^m - C_{min}^m) + C_{min}^m$. In this expression C_{max}^m and C_{min}^m are defined by:

$$C_{max}^m = \max_{i,j} (|\nabla_{\sigma_m} f[i, j]|) \quad (4.25)$$

$$C_{min}^m = \min_{i,j} (|\nabla_{\sigma_m} f[i, j]|) \quad (4.26)$$

Now the edge map $E(i, j, \sigma_m)$ at scale σ_m of image f is defined by:

$$E(i, j, \sigma_m) = \begin{cases} 1, & \text{for } |\nabla_{\sigma_m} f[i, j]| > T^m \\ 0, & \text{otherwise} \end{cases} \quad (4.27)$$

An edge stability map $Q(i, j)$ is obtained by considering the longest subsequence $E(i, j, \sigma_m), \dots, E(i, j, \sigma_{m+l-1})$ of edge images such that $Q(i, j) = l$ where l is given by:

$$l = \operatorname{argmax}_l \bigcap_{\sigma_m \leq \sigma_k \leq \sigma_{m+l-1}} \{E(i, j, \sigma_k) = 1\} \quad (4.28)$$

The edge stability map $\hat{Q}(i, j)$ of the reconstructed image g can be calculated in a similar fashion. The Edge Stability Mean Square Error (ESMSE) now becomes:

$$ESMSE(f, g) = \frac{1}{n_f} \sum_{i,j=1}^{n_f} \left(Q(i, j) - \hat{Q}(i, j) \right)^2 \quad (4.29)$$

where n_f is again the number of edge pixels in image f . Note that this makes the measure non symmetrical.

4.2.1.18 Spectral Magnitude Distortion

First we define the discrete Fourier transform $F[\omega_1, \omega_2]$ of image $f[x, y]$ as:

$$F(\omega_1, \omega_2) = \frac{1}{\sqrt{MN}} \sum_{i=1}^M \sum_{j=1}^N (f[x, y] e^{i(\omega_1 x + \omega_2 y)}) \quad (4.30)$$

Consider the discrete Fourier transforms $F[\omega_1, \omega_2]$ and $G[\omega_1, \omega_2]$ of $f[i, j]$ and $g[i, j]$ respectively. The phase and magnitude of $F[\omega_1, \omega_2]$ can be written as:

$$\varphi_f(\omega_1, \omega_2) = \arctan(F(\omega_1, \omega_2)) \quad (4.31)$$

$$M_f(\omega_1, \omega_2) = |F(\omega_1, \omega_2)| \quad (4.32)$$

Similarly we can define the phase φ_g and magnitude M_g of image g . Following Avcibas [2] we define the Spectral Magnitude Distortion (SMD) as:

$$SMD(f, g) = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N |M_f(i, j) - M_g(i, j)|^2 \quad (4.33)$$

4.2.1.19 Spectral Phase Distortion

Using previous definitions the Spectral Phase Distortion (SPD) is defined as:

$$SPD(f, g) = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N |\varphi_f(i, j) - \varphi_g(i, j)|^2 \quad (4.34)$$

4.2.1.20 Weighted Spectral Distortion

The total spectral distortion can be calculated by weighing phase and amplitude. Following Avcibas [2] the Weighted Spectral Distortion (WSD) becomes:

$$WSD_\lambda(f, g) = \frac{1}{MN} \cdot \left(\lambda \sum_{i=1}^M \sum_{j=1}^N |\varphi_f(i, j) - \varphi_g(i, j)|^2 + (1 - \lambda) \sum_{i=1}^M \sum_{j=1}^N |M_f(i, j) - M_g(i, j)|^2 \right) \quad (4.35)$$

with λ a free parameter to balance phase and amplitude.

4.2.1.21 Median Spectral Magnitude Block Distortion

The previous spectral methods did however not take any localization into account. In order to include this, Minkowski averaging of block spectral distortions may be advantageous. Consider an image divided into L overlapping or non-overlapping blocks of size $b \times b$, say 32×32 , and blockwise spectral distortions can be computed as in (4.33)-(4.35). Consider the discrete Fourier transform $F^l(\omega_1, \omega_2)$ of the l -th block of image f and $G^l(\omega_1, \omega_2)$ of the l -th block of image g . The phase and magnitude of the l -th block of image f become:

$$\varphi_f^l(\omega_1, \omega_2) = \arctan \left(F^l(\omega_1, \omega_2) \right) \quad (4.36)$$

$$M_f^l(\omega_1, \omega_2) = |F^l(\omega_1, \omega_2)| \quad (4.37)$$

Similarly M_g^l and φ_g^l of the l -th block of image g can be calculated. Minkowski averaging of the magnitude differences over all blocks results in:

$$J_M^l = \left(\sum_{\omega_1=1}^M \sum_{\omega_2=1}^N \left(M_f^l - M_g^l \right)^\gamma \right)^{1/\gamma} \quad (4.38)$$

For $\gamma = 2$ we obtain the RMS of the magnitude error. The Median Spectral Magnitude Block Distortion (MSMBD) now becomes:

$$MSMBD(f, g) = \text{Median}_l J_M^l \quad (4.39)$$

4.2.1.22 Median Spectral Phase Block Distortion

Similar to the previous measure for the magnitude we can do Minkowski averaging of the phase differences over all blocks:

$$J_\varphi^l = \left(\sum_{\omega_1=1}^M \sum_{\omega_2=1}^N \left(\varphi_f^l - \varphi_g^l \right)^\gamma \right)^{1/\gamma} \quad (4.40)$$

For $\gamma = 2$ we now obtain the RMS of the phase error. The Median Spectral Phase Block Distortion (MSPBD) now becomes:

$$MSPBD(f, g) = \text{Median}_l J_\varphi^l \quad (4.41)$$

4.2.1.23 Median Weighted Spectral Block Distortion

Similar to (4.35) one can combine the phase block distortion and the magnitude block distortion:

$$J^l = \lambda J_M^l + (1 - \lambda) J_\varphi^l \quad (4.42)$$

and thus the Median Weighted Spectral Block Distortion (MWSBD) becomes:

$$MWSBD(f, g) = \text{Median}_l J^l \quad (4.43)$$

4.2.1.24 Fuzzy Similarity Measures

Van der Weken [133] presented a number of fuzzy similarity measures. The first one is based on the Minkowski metric:

$$S1_p(A, B) = 1 - \left(\frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N |A[i, j] - B[i, j]|^p \right)^{\frac{1}{p}} \quad (4.44)$$

Here $A[i, j]$ and $B[i, j]$ represent the fuzzy sets of $f[i, j]$ and $g[i, j]$ respectively. In practice for normalized images with gray-scale values between 0 and 1, $A[i, j] = f[i, j]$ and $B[i, j] = g[i, j]$, which makes this measure equal to one minus the \mathbb{L}_p -norm. The second similarity measure is based on the Kullback distance [88]:

$$S3(A, B) = 1 - \frac{1}{MN2\ln 2} \cdot \sum_{i=1}^M \sum_{j=1}^N [(A[i, j] - B[i, j]) \ln \left(\frac{1 + A[i, j]}{1 + B[i, j]} \right) (B[i, j] - A[i, j]) \ln \left(\frac{2 - A[i, j]}{2 - B[i, j]} \right)] \quad (4.45)$$

Again $A[i, j]$ and $B[i, j]$ represent the fuzzy sets of $f[i, j]$ and $g[i, j]$ respectively. Note that the name S_3 is adapted from Avcibas [2].

4.2.1.25 Ordered Histogram Similarity Measures

Instead of applying similarity measures on the pixel data directly, one could also apply similarity measures on histograms of the images. Van der Weken presented similarity measures based on ordered histograms [132]. Let us consider the histogram $h_A(x_i)$ of fuzzy set or image A . The value $h_A(x_i)$ is equal to the total number of pixels in image A with gray value x_i . Note that for 8-bits grayscale images ranging from 0 to 1, $x_i = \frac{0}{255}, \frac{1}{255}, \dots, \frac{255}{255}$ for $i = 1, \dots, 256$ respectively. The ordered histogram $o_A(i)$ is obtained by sorting the histogram $h_A(x_i)$ by its frequencies in descending order. The ordered histogram can be transformed into a fuzzy set by dividing by the largest component of the ordered histogram:

$$Oh_A(i) = \frac{o_A(i)}{o_A(1)} \quad (4.46)$$

with $o_A(1) = \max_{x_i} h_A(x_i)$. Now two similarity measures are used to compare image histograms, adapted from [132]:

$$H1_p(A, B) = 1 - \left(\frac{1}{L} \sum_{i=1}^L |Oh_A(i) - Oh_B(i)|^p \right)^{\frac{1}{p}} \quad (4.47)$$

$$H3(A, B) = 1 - \frac{\sum_{i=1}^L |Oh_A(i) - Oh_B(i)|}{\sum_{i=1}^L (Oh_A(i) + Oh_B(i))} \quad (4.48)$$

where $A[i, j]$ and $B[i, j]$ represent the fuzzy sets of $f[i, j]$ and $g[i, j]$ respectively. Note that $L = 256$ for 8-bit images and p is a free parameter for the Minkowski metric.

4.2.1.26 Combined Fuzzy Histogram Measures

In [132] the fuzzy similarity measures and the ordered histogram similarity measures are combined. Two combinations are evaluated in this paper:

$$Q1_p(A, B) = S1_p \cdot H1_p \quad (4.49)$$

$$Q3(A, B) = S3 \cdot H3 \quad (4.50)$$

4.2.1.27 Universal Image Quality Index

Wang and Bovik introduced the universal image quality index [146] defined as:

$$UIQI(f, g) = \frac{4 \sigma_{fg} \bar{f} \bar{g}}{(\sigma_f^2 + \sigma_g^2)[\bar{f}^2 + \bar{g}^2]} \quad (4.51)$$

where

$$\begin{aligned} \bar{f} &= \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N f[i, j] \\ \bar{g} &= \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N g[i, j] \\ \sigma_f^2 &= \frac{1}{MN-1} \sum_{i=1}^M \sum_{j=1}^N (f[i, j] - \bar{f})^2 \\ \sigma_g^2 &= \frac{1}{MN-1} \sum_{i=1}^M \sum_{j=1}^N (g[i, j] - \bar{g})^2 \\ \sigma_{fg} &= \frac{1}{MN-1} \sum_{i=1}^M \sum_{j=1}^N (f[i, j] - \bar{f})(g[i, j] - \bar{g}) \end{aligned}$$

4.2.1.28 SVD-based Error Measure

Shnayderman [126] introduced an error measure based on the singular value decomposition (SVD) of an image. Consider singular values $s_{f_k}^i = s_{f_k}^1, \dots, s_{f_k}^n$ of a block f_k with size $n \times n$ from image f . The distortion between the k -th non-overlapping image block of f and g is defined by:

$$D_k(f_k, g_k) = \sqrt{\sum_{i=1}^n (s_{f_k}^i - s_{g_k}^i)^2} \quad (4.52)$$

Note the block size n is a free parameter. The resulting error between image f and g becomes:

$$M_{SVD}(f, g) = \frac{\sum_{k=1}^{\frac{MN}{n^2}} |D_k(f_k, g_k) - D_{\text{mid}}(f_k, g_k)|}{\frac{MN}{n^2}} \quad (4.53)$$

where D_{mid} represents the mid point of the sorted $D_k(f_k, g_k)$ list and n is the block size.

4.2.1.29 GWSB-PSNR

Wang et al. [123] considered an image measure based on the PSNR but calculated in different blocks in the image. Consider the images divided into $m \times n$ non-overlapping blocks of size $h \times w$, which means $M = m \times w$ and $N = n \times h$. According to [123] the Block Mean Squared Error (BMSE) of the k -th block becomes:

$$\text{BMSE}_k(f_k, g_k) = \frac{1}{hw} \sum_{i=1}^w \sum_{j=1}^h (f_k[i, j] - g_k[i, j])^2 \quad (4.54)$$

Similarly, the Block Peak Signal to Noise ratio (BPSNR) can be calculated:

$$\text{BPSNR}_k(f_k, g_k) = 10^{10} \log \left(\frac{1}{\text{BMSE}_k(f_k, g_k)} \right) \quad (4.55)$$

Now the Geometry Weighted Separating Block Peak Signal to Noise Ratio (GWSB-PSNR) is defined by:

$$\text{GWSB PSNR}(f, g) = \frac{1}{mn} \sum_{k=1}^{mn} \lambda_k \text{BPSNR}_k(f_k, g_k) \quad (4.56)$$

where λ_k is equal to:

$$\lambda_k = \frac{\text{BMSE}_k(f_k, g_k)}{\text{MSE}(f, g)} \quad (4.57)$$

4.2.1.30 FIM

Another fuzzy image metric is proposed by Junli et al. [19, 20]. Consider two images X and Y as vectors, $X = (x_1, x_2, \dots, x_K)$ and $Y = (y_1, y_2, \dots, y_K)$ where K is the number of pixels in the image. The difference between X and Y is defined as:

$$|X - Y| = (|x_1 - y_1|, |x_2 - y_2|, \dots, |x_K - y_K|) \quad (4.58)$$

Now for 8-bits images the Fuzzy Image Metric (FIM) is defined as:

$$\text{FIM}(X, Y) = \max_{0 \leq i \leq 255} \min(i/255, \mu(N_{i/255}(|X - Y|))) \quad (4.59)$$

with:

$$\mu = \frac{|\{\cdot\}|}{K} \quad (4.60)$$

where $|\{\cdot\}|$ is the number of elements in $\{\cdot\}$ and

$$N_\alpha(f) = \{x | f(x) \geq \alpha\} \quad (4.61)$$

the elements in f larger than α .

4.2.1.31 IQE

The Image Quality Evaluation [20] is defined by:

$$IQE(X, Y) = \frac{5}{1 + (FIM(X, Y)/a)^b} \quad (4.62)$$

where $a = 0.0647$ and $b = 4.438$. These parameters from [20] are determined experimentally using human observer data.

4.2.2 Multi-Scale Differential Error (MSDE)

Additionally to the presented image quality measures, a new image quality measure is proposed named Multi-Scale Differential Error (MSDE). It is inspired by the human visual system and scale space theory. The goal of the MSDE is to provide a simple, effective error measure suitable for image reconstruction that is comparable to results of a human observer. The MSDE is based on a solid mathematical framework for multi-scale image analysis, viz. scale space theory. In 1962, Taizo Iijima derived the Gaussian kernel from a set of basic axioms to investigate signals at multiple scales [61]. Later, papers about linear or Gaussian scale space followed from various other authors independently, including Witkin [141] and Koenderink [80] in the eighties. Scale space theory also provides a nice mathematical framework for taking spatial derivatives of the image.

It is known that the human front end visual system also uses different scales of observation and takes derivatives up to at least 4th order at these scales [145, 83]. It seems that the *differential structure* of an image is important for humans to detect objects. This should thus be reflected in an error measure to compare two images. There exist several multi-scale methods [123, 5, 27, 71] and also measures using derivatives (such as the \mathbb{L}_2 gradient [127]) or edges [15, 122] are used commonly. However, no current method is known by the authors that combines scale and differential structure for image quality measures.

Consider again the gradient magnitude $|\nabla_\sigma f[i, j]|$ of image f at scale σ , cf. (4.23). For R different scales, the gradient magnitude error map $\Psi_{f,g}$ between image f and g is defined as:

$$\Psi_{f,g}[i, j] = \sqrt{\frac{1}{R} \sum_{\sigma=\sigma_1}^{\sigma_R} (|\nabla_\sigma f[i, j]| - |\nabla_\sigma g[i, j]|)^2} \quad (4.63)$$

Note that the largest scale σ_R should be chosen such that the complete image is covered. The Multi-Scale Differential Error is now defined as:

$$MSDE(f, g) = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N \Psi_{f,g}[i, j] \quad (4.64)$$

The MSDE is not sensitive to brightness changes (it is however sensitive to contrast changes, although less than the \mathbb{L}_p -error) and it takes the differential structure of the images (edges) into account. The MSDE also has no tunable parameters, except for the number of scales that are taken into account. However, for large R the exact value is unimportant, since almost nothing changes between scales (the sum converges to an integral with an appropriate scale measure). One could introduce σ_1 as a tuneable parameter to have some control on the highest frequency taken into account for the error. This might be beneficial for comparison of very noisy images. Note that normalized derivatives also could have been used, but some bias to low scale errors showed better results in practice.

Implementation Notes

Fast MSDE

Calculating the MSDE using Gaussian derivatives can be quite slow, depending on the number of scales taken into account. However, a much faster rough approximation can be used. Consider an image pyramid with various levels of resolution denoted by $r \geq 0$. Instead of the average gray-level, the average gradient using pixel differences can be used. First consider the gradient magnitude f' of image f using pixel differences:

$$f'[i, j] = \sqrt{(f[i, j] - f[i + 1, j])^2 + (f[i, j] - f[i, j + 1])^2} \quad (4.65)$$

Note that at the boundary one should use a proper condition, e.g. zero-padding, reflective boundary or constant boundary. Similarly the gradient magnitude g' of g can be calculated:

$$g'[i, j] = \sqrt{(g[i, j] - g[i + 1, j])^2 + (g[i, j] - g[i, j + 1])^2} \quad (4.66)$$

One should start at the highest resolution $r = 0$ and calculate the difference in each pixel between f' and g' . For $r = 1$ the resolution should be lowered by a factor 2 and the difference should be calculated in each corresponding pixel at that level of resolution. For $r = 2$ the resolution of the image of $r = 1$ should be lowered by a factor of 2, etc. This should be repeated until only one pixel is left. In case that an odd number of pixels is left at a certain scale, one line should either be neglected or added to the last pair. The Fast Multi-Scale Differential Error (FMSDE) now becomes:

$$FMSDE(f, g) = \sum_{r=0}^R \sum_{i=\frac{N}{2^r}, j=\frac{M}{2^r}}^{\frac{N}{2^r}-1, \frac{M}{2^r}-1} |f'_r[i, j] - g'_r[i, j]| \quad (4.67)$$

where $f'_r[i, j]$ is the image in the gradient magnitude image pyramid of f' at level r and $g'_r[i, j]$ is the image in the gradient magnitude image pyramid of g' at level r . Note that with the FMSDE there are no tuneable parameters, since the highest resolution is the pixel scale and the lowest resolution has only one value. However, one parameter α can be introduced related to the σ_1 parameter for the MSDE. It represents the number of low level scales that are not taken into account for the first sum:

$$FMSDE_\alpha(f, g) = \sum_{r=\alpha}^R \sum_{i=\frac{N}{2^r}, j=\frac{M}{2^r}}^{\frac{N}{2^r}-1, \frac{M}{2^r}-1} |f'_r[i, j] - g'_r[i, j]| \quad (4.68)$$

4.2.3 Human Visual System based measures

Several error measures based on models of the Human Visual System (HVS) exist. These measures can be quite complex and are usually not trivial to implement. Therefore, three measures are selected of which code is publicly available. Some other measures, not included in our experiments, can be found in the work by Bock et al. [10], Beghdadi et al. [5], Pons et al. [121], Eude et al. [35], Xu et al. [144], de Freitas et al. [24], Chin et al. [21], Rohaly et al. [125] and Westen et al. [140].

4.2.3.1 DCTune 2.0

DCTune [138] is a tool designed to optimize still image compression [139]. It is developed by NASA and based on a perceptual error measure. The latest version can calculate the perceptual error between two images as a single number, which is used in our experiments.

4.2.3.2 SSIM

The Structural SIMilarity (SSIM) index is an image quality measure that incorporates a Human Visual System model of degradation of structural information [148, 146]. The publicly available software package of Wang et al. [147] is used.

4.2.3.3 Quality Assessor v2.0

Carnec et al. [17] also presented a HVS based error measure based on the structural information between the original and distorted image. It is implemented in a tool called Quality Assessor (v2.0) [16] which is used for our experiments.

4.3 Experiments

4.3.1 Setup

For the experiments, four sets of 10 images are used, which are shown in figure 4.1. The *Einstein set* (top row) consists of a reference image (left) and 9 degraded images. The type of degradation differs from additive noise, greyscale transformation and compression artifacts to image reconstruction from a limited set of constraints (see chapter 3 and 5). The *Lena set* (second row) and the *Scotland set* (third row) are similar to the Einstein set: The left images are reference images and the other images have diverse degradations. The *Russia set* (bottom row) is used for validation. The left image is again the reference image and the following images are reconstructions using a decreasing set of constraints. This set is the only *ordered* set.

In order to compare the presented error measures with human observers, 33 volunteers were asked to order the images in the Einstein, Lena and Scotland set. The best degraded image (with the lowest perceptual error) obtained a score of 1, while the worst image (with the highest perceptual error) obtained a score of 9. The results of the human observer experiment are shown in figure 4.2. The images above the graphs are ordered from lowest error to largest error (Appendix B shows larger images of the Einstein, Lena and Scotland set, ordered by the human observers). The graphs show

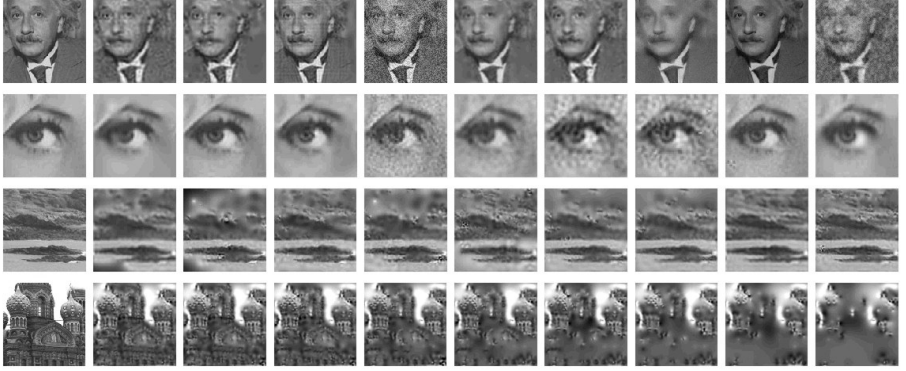


Figure 4.1: Image sets used for experiments: From top to bottom: Einstein set, Lena set, Scotland set and Russia set. Note that only the Russia set is ordered.

the mean and standard deviation of the scores from all 33 observers for all three image sets.

All of the presented error measures are used to calculate the error between the reference image and the degraded images of the three image sets. In all cases the results are ordered in the same way as the mean human observer ordering. This means that image 1 of the error measures corresponds to the best degraded image according to the human observers. Ideally the error measure should thus give monotonically increasing errors for each image set. Note that the Scotland set was considered hard to order by most human observers, which is reflected in the standard deviation of the ordering.

4.3.2 Evaluation

First some remarks can be made about the ordering of the sets by the human observers. Appendix B shows enlarged versions of the ordered test images for viewing purposes. In the Einstein set one can observe that the best rated image (image 1) is in fact the original image which is lowered in intensity slightly. Since all image information is still available and no noise is present, this image is logically rated as the best image (we will however see that many objective error measures rate this image differently). The second image seems to show some “grid” on the image, but is still quite detailed. The third image shows some compression artifacts. Image 4 has noise added to the image, but many human observers actually prefer some noise over missing information or strange artifacts. The worst image is a reconstruction with noise added to the points. The result is a blurred and yet “noisy” image, which is not clear. For the Lena image, the best three images are reconstructions from a decreasing number of constraints. Image 4 is a reconstruction from the same number of points as image 1, but with a small artifact in the lower-left corner. Humans are very sensitive to artifacts in areas where known objects (such as parts of a face) are present. In practice however it will be impossible to incorporate the advanced human interpretation of a scene in an image in any generic objective error measure. The last three images are reconstructions from the same points as the first three images, but with additive noise. For the Scotland set, the best image

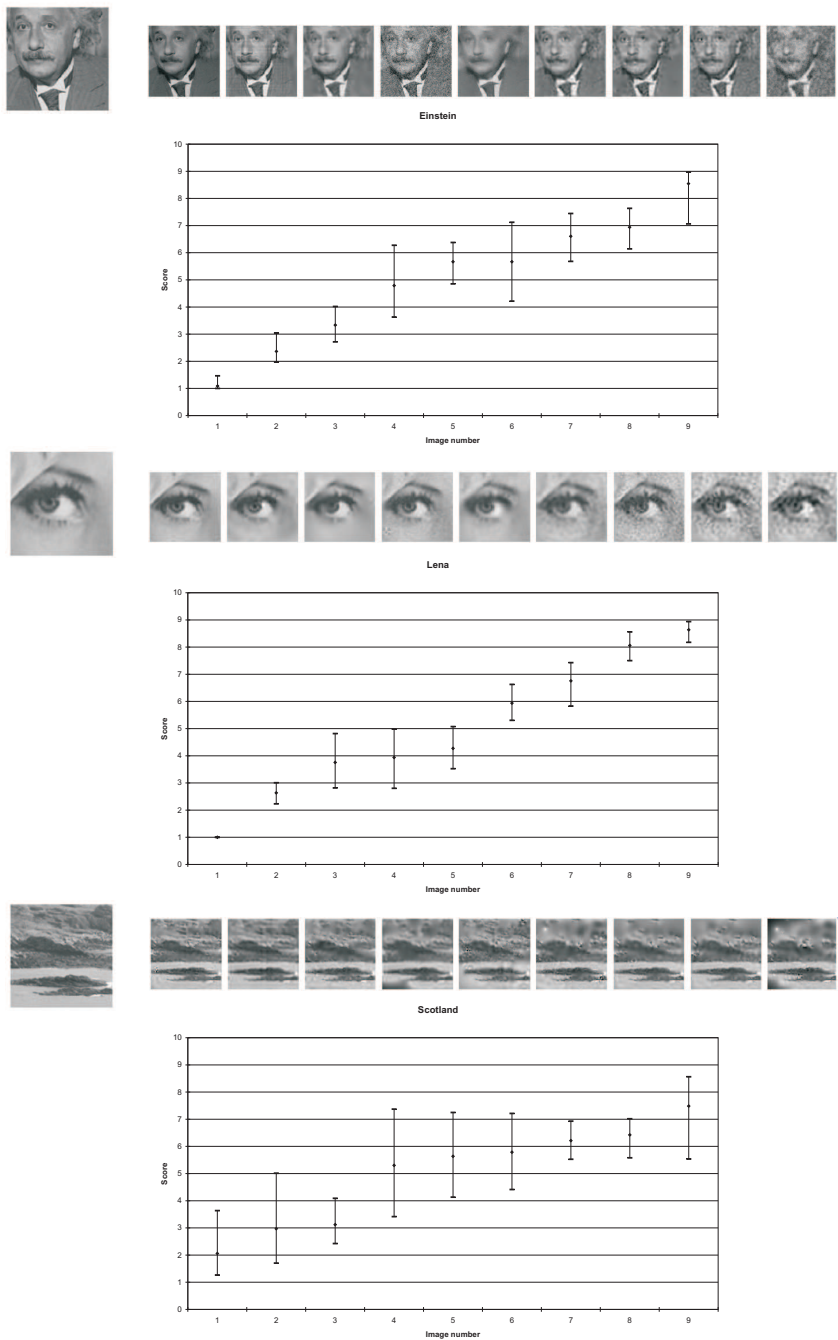


Figure 4.2: Human observer results. 33 Volunteers were asked to order the images of the three test sets. Top: Mean ordering for the Einstein set with the corresponding graph of mean and standard deviation of the scores. Center: Mean ordering and corresponding graph for the Lena set. Bottom: Mean ordering and corresponding graph for the Scotland set.

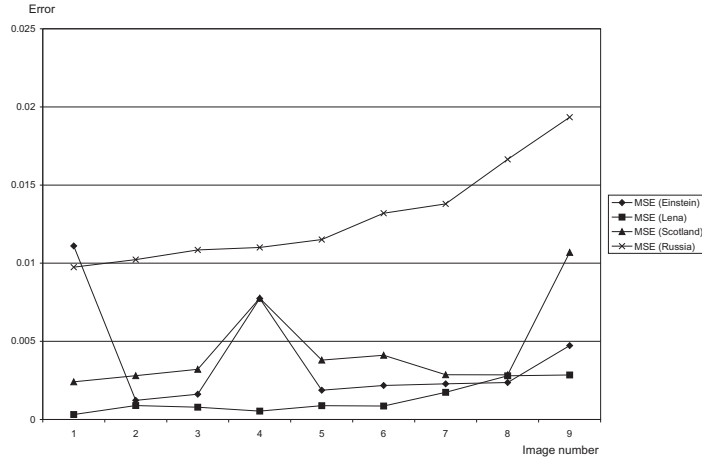


Figure 4.3: Mean Squared Error (MSE) results. Images are ordered by human observers from best (image 1) to worst (image 9). The corresponding error is shown.

has an artifact similar to the one in the Lena set. However, no clear known object is present which makes the artifact less disturbing. It is however clear from figure 4.2 that the human observers found it much harder to order this set. Ideally, this should also be reflected in the objective error measure.

In this section we will give some observations on the results of the objective error measures compared to the human observer study and show some of the resulting graphs. Only some of the actual results are shown here due to space limitations, the reader is referred to Appendix A for the results of all error measures, including different parameters that were used. The first presented error measure is the Mean Squared Error (MSE). Figure 4.3 shows the results of the MSE for the four test sets. The Russia set shows an increasing error from image 1 (left) to image 9 (right) as expected, since the number of constraints for the reconstruction is decreasing. For the first image of the Einstein set however, the MSE is large. This image shows a small decrease in brightness, but contains almost all image information. While the human observer rates this image as best quality, the MSE is the largest among the test images. The MSE is very sensitive to intensity changes of the image. Note that the fourth image of the Einstein and Scotland set also have a rather high MSE, due to the fact that the human observer prefers minor noise over absence of information, which is not reflected in the error measure. The standard deviation in the human observer results is however relatively large for these images.

The Root Mean Squared Error (RMSE) and the Relative root Mean Squared Error (RRMSE) show similar trends, where the RRMSE seems to suffer less from the intensity and noise problems of the MSE. The Mutual Information (MI) is a similarity measure rather than an error measure. This means that one would expect decreasing trends in the graphs rather than increasing lines as for error measures. Figure 4.4 shows the resulting graph of the MI. Note that the MI does not suffer from the intensity problem of the MSE based error measures. It does however rate noisy images very low as can

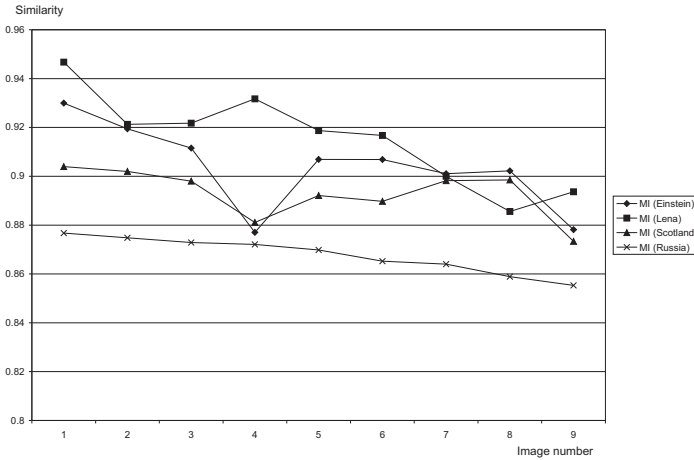


Figure 4.4: Mutual Information (MI) results. Images are ordered by human observers from best (image 1) to worst (image 9). The corresponding similarity is shown.

be seen in the Einstein set. Image 4 of the Lena set with the small artifact has high similarity, which shows that the MI is not really sensitive to small pixel distortion. The Signal to Noise ratio (SNR) and Peak Signal to Noise Ratio (PSNR) show similar trends, but do suffer from the intensity problem similar to the MSE.

The Maximum Absolute Difference (MAD) only uses the maximum difference in a single pixel. Therefore it is sensitive to outliers and reconstruction artifacts. Image 4 of the Einstein set has a low MAD since the noise that is added has a low variance. This is the reason it has the lowest MAD. This error measure does not resemble the human observer rating very well. This is similar for the Relative Maximum Absolute Difference (RMAD). The Mean Absolute Error (MAE) has similar properties as the MSE. The \mathbb{L}_p error is equal to the MAE for $p = 1$, equal to the RMSE for $p = 2$ and equal to the MAD for $p = \infty$. The \mathbb{L}_p ranges from global to local, with corresponding properties. For low p it resembles the MSE while for high p it resembles the MAD. Despite the fact that the Modified Minkowski Infinity Metric not only looks at the largest pixel difference, even for large r it resembles the behavior of the MAD; the method is still very sensitive to outliers. The Neighborhood Difference (ND) takes a small displacement of pixels into account. However, it still suffers from the intensity problem of the MSE. The trend of the ND is similar for all settings that have been tried.

The Multi-resolution Distance Measure (MDM) calculates the MAE at several levels of resolution, weighted differently for each resolution. The MDM is also very sensitive to intensity changes; the first image of the Einstein set has a very high MDM, almost three times as high as the second largest error. The Structural Content (SC) is a very simple method looking at the difference in total intensity. It gives the correlation between the two images. The corresponding graph is shown in figure 4.5. Since most images in the set are distorted versions of the original image, this error measure is very poorly describing the differences. The same holds for the Normalized Cross-Correlation (NCC).

The Pratt Measure (PM) incorporates edges in the image. Therefore it is not sensi-

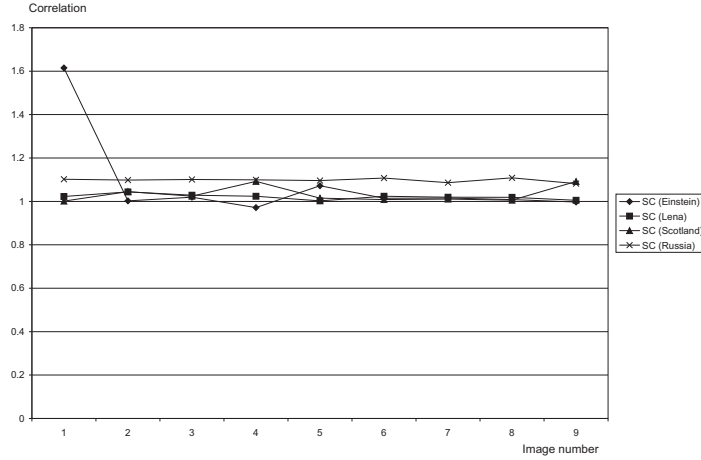


Figure 4.5: Structural Content (SC) results. Images are ordered by human observers from best (image 1) to worst (image 9). The corresponding correlation is shown.

tive to intensity changes. According to the PM for $\sigma = 1$, $\alpha = 0.2$ and $\beta = 1$ image 5 of the Einstein set is much worse than the other images, which is not the case in the human observer results. Image 5 does however lack detail. Figure 4.6 shows the corresponding graph of the PM. The PM resembles the human observer results reasonably well considering the standard deviation in the human observer results. The Edge Stability Measure also takes edges into account, but does not a good job describing the differences in the images. Even the Russia control set shows a non increasing trend.

The Spectral Magnitude Distortion (SMD), the Spectral Phase Distortion (SPD) and the Weighted Spectral Distortion (WSD) use the frequencies of the discrete Fourier transform to calculate the difference of the images. Since the zero frequency is also used, the magnitude (SMD) will suffer from the intensity problem. It is also sensitive to high frequency noise and artifacts. The phase (SPD) is not sensitive to the intensity problem, but is sensitive to high frequency noise. For the Russia image however, there is almost no difference according to the SPD, while a large peak is found for the noisy Einstein image. The combined magnitude and phase (WSD) combines the flaws of the two separate methods. The median can be taken for these measures localized in different blocks of the image (MSMBD, MSPBD and MWSBD, respectively) which improves the results, since outliers are removed. Especially the MSPBD, shown in figure 4.7 performs quite well, except that it is sensitive to high frequency noise such as image 4 of the Einstein set.

Some fuzzy measures are introduced, such as the Fuzzy Similarity Measures (S1 and S3). For high p S1 performs reasonably well, but is sensitive to the black background at the boundaries. The differences in both S1 and S3 between images of the Russia set are small, which makes these error measures not usable for the presented reconstruction evaluation. The Ordered Histogram Similarity Measures (H1 and H3) did not resemble the human observer results at all and thus the combined methods (Q1 and Q3) performed not well either. The Universal Image Quality Index did not show much

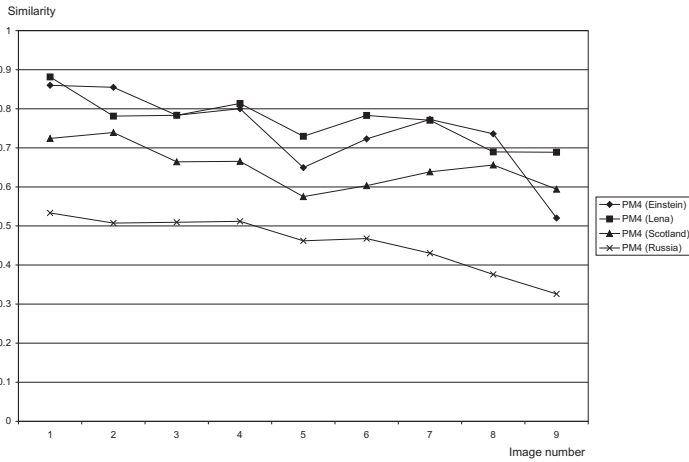


Figure 4.6: Pratt Measure (PM) results with $\sigma = 1$, $\alpha = 0.2$, $\beta = 1$. Images are ordered by human observers from best (image 1) to worst (image 9). The corresponding similarity is shown.

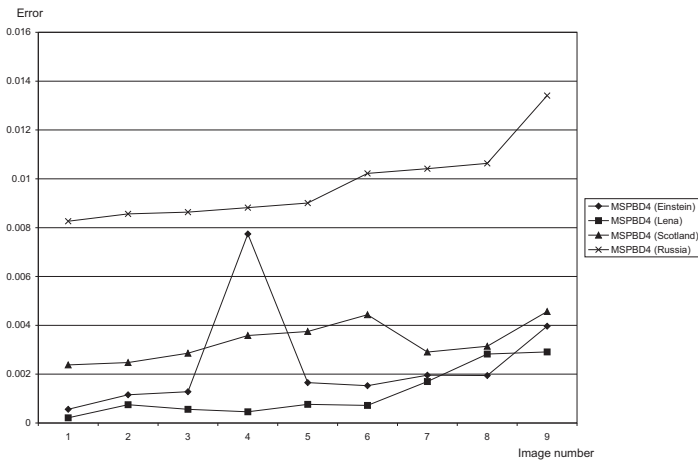


Figure 4.7: Median Spectral Phase Block Distortion (MSPBD) results with blocksize $b = 32$. Images are ordered by human observers from best (image 1) to worst (image 9). The corresponding error is shown.

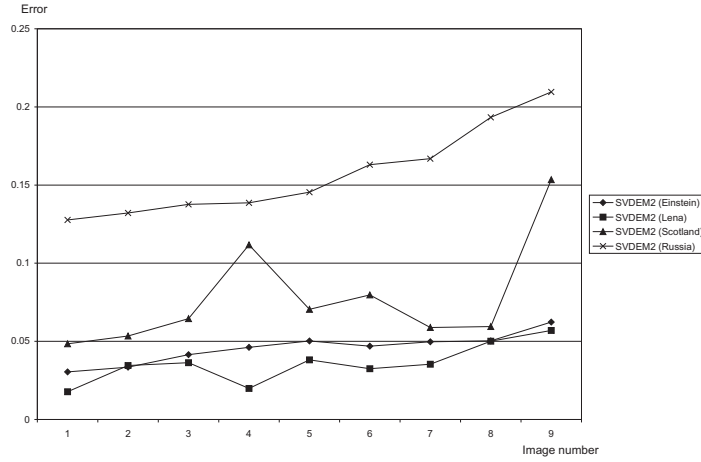


Figure 4.8: Singular Value Decomposition error measure M_{SVD} results with blocksize $b = 4$. Images are ordered by human observers from best (image 1) to worst (image 9). The corresponding error is shown.

discrepancy between the images except for the noisy image 4 of the Einstein set and the blurred image 4 of the Scotland set. The presented method using Singular Value Decomposition (M_{SVD}) also shows reasonable results except that for the Scotland set the black background near the boundaries is penalized too much. Furthermore the method shows some sensitivity to the artifacts of image 4 of the Lena set. The corresponding graph is shown in figure 4.8.

The Geometry Weighted Separating Block Peak Signal to Noise Ratio (GWSB-PSNR) did show reasonable results for the Russia and Lena image set, however for the Scotland and Einstein set the discrepancies between the images are very small. The Fuzzy Image Metric (FIM) shows good results for the Russia control set, but suffers from the intensity problem and is sensitive to noise which shows in the Einstein and Lena set. The related Image Quality Evaluation (IQE) does not solve the intensity problem and also still suffers from sensitivity to noise.

The proposed Multi-Scale Differential Error (MSDE) does not suffer from the intensity problem, but is still sensitive to noise. With proper settings for the lowest scale to take into account, this sensitivity can be controlled. Figure 4.9 shows the graphs for the MSDE with $\sigma_1 = 1.4$ pixels. The Russia control set shows good results and the Einstein set also follows the human observer results quite well. It does however rate the blurred image 5 of the Einstein set a bit too low. In the Lena set the artifact in image 4 is not really taken into account due to the higher scale and as such the image is rated a bit too low. The first few images and the last image of the Scotland set are rated correctly, but in between the ordering differs from the human observer results. The standard deviation of the human observer results is however large for these images.

The Fast Multi-Scale Differential Error - which is an approximation of the MSDE - shows similar results. The sensitivity to noise is however slightly worse. The FMSDE also suffers from the black background in the boundaries of the Scotland image. The

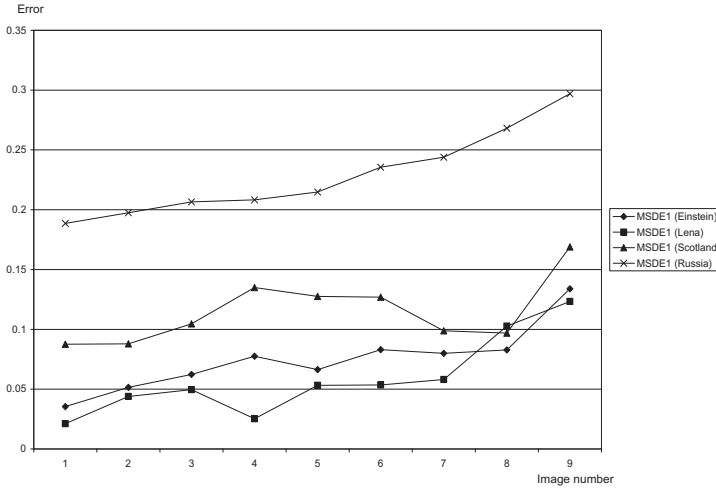


Figure 4.9: Multi-Scale Differential Error (MSDE) results with $\sigma_1 = 1.4$. Images are ordered by human observers from best (image 1) to worst (image 9). The corresponding error is shown.

more complex human visual system (HVS) based methods include the DCTune 2.0 error measure. Unfortunately this error measure is also not invariant to intensity changes and has problems with the black background on the borders of the Scotland image. The Structural Similarity Measure (SSIM) shown in figure 4.11 has good results on all sets but is rather sensitive to noise. The Quality Assessor 2.0 (QA2) did not work for the small Lena image. The results for the Russia image were good, but the discrepancy rather small. For the Einstein set the ordering is quite different from the human observer results.

4.4 Conclusion and discussion

To measure the quality of image reconstructions from scale space interest points, objective error measures are a necessity. Many types of error measures exist in literature, but no reference can be found describing image errors for such a broad spectrum of degradations as is the case with image reconstruction from a limited set of constraints. Therefore we presented 31 generic mathematical error measures and three more complex human visual system based error measures found in the literature and compared them to a simple human observer experiment for reconstructed images. Furthermore two new image error measures are presented and compared to the human observer results as well.

First it can be concluded that it is almost impossible to find an objective error measure that correctly resembles the human observer results. A human observer uses knowledge about the image and its contents that cannot be captured by a simple computer algorithm. This makes designing a perfect objective error measure impossible. However, some error measures are more suitable for certain tasks than others and for the image reconstruction this is reflected in the results of the experiments done in this

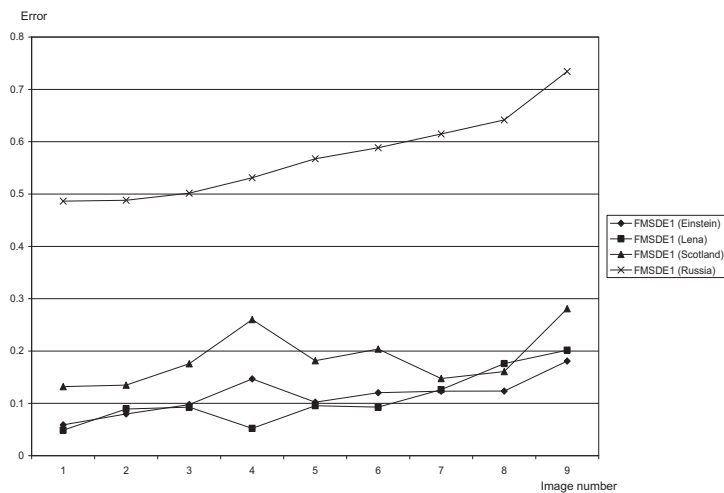


Figure 4.10: Fast Multi-Scale Differential Error (FMSDE) results with $\alpha = 1$. Images are ordered by human observers from best (image 1) to worst (image 9). The corresponding error is shown.

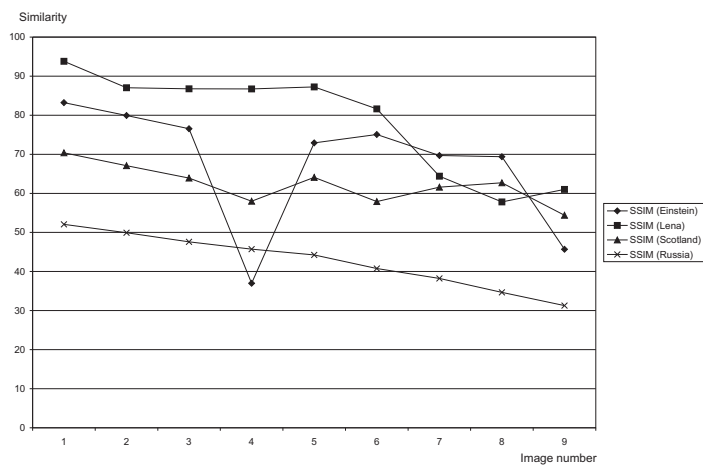


Figure 4.11: Structural Similarity (SSIM) results. Images are ordered by human observers from best (image 1) to worst (image 9). The corresponding similarity is shown.

chapter. Many of the methods described are not invariant to intensity changes. This is reflected in the results, since there is a large discrepancy between human observers and most error measures when the intensity is raised or lowered a bit. Noise is a problem for all error measures. Clearly some noise in the image does not affect the human observer too much, while this is not the case for the objective error measures. A human sometimes prefers some noise over missing information or strange artifacts. As for artifacts, much model information is used by humans. A strange set of spots in a face is more disturbing than a similar structure on a leaf or stone, while for a computer at a certain scale the images might be similar.

Of all the tested error measures, the Structural Similarity Measure (SSIM) probably resembles the human observer results best. However, this method is still sensitive to noise in the image. Our proposed Multi-Scale Differential Error (MSDE) and the fast version of it (FMSDE) are performing quite well. Both suffer from the noise problem, but in most cases less than the SSIM since a tunable parameter can control the highest resolution taken into account. However, the (F)MSDE is less sensitive to certain artifacts as the human observers and the SSIM. The edge based Pratt measure (PM) and the phase based MSPBD are also performing reasonably well, but worse than the (F)MSDE and SSIM. The well known and most commonly used Mean Squared Error (MSE) and related measures are performing not very well since they are not invariant to intensity changes. This also holds for the Signal to Noise Ratio (SNR) and related error measures.

The human observer experiment was rather small (only 33 subjects) and only a limited set of images was used. This makes it impossible to draw hard conclusions on the quality of the measures in general. A much larger experiment would be useful to conclude which error measure is the best. One of the problems with most error measures is the intensity problem. Some of the algorithms (including our proposed methods) handle this by making the measure invariant to intensity changes. In our opinion however, this is not exactly as human observers would see it. Up to a certain degree an intensity change would be not too disturbing for humans, but for large changes an image with minor blur might be better looking to human observers. Instead of intensity invariance, some sort of intensity *tolerance* should be introduced. The same holds for noise, but this is harder to implement. Introducing models of image contents could be useful for certain tasks, but is too complex for general applications.



Scale Space Interest Points

This chapter is based on:

A Comparison of Multi-scale Interest Points used for Image Reconstruction.

F.M.W. Kanters and M. Lillholm.

In preparation.

Canonical Sets in Image Reconstruction from Multiscale Interest Points.

F.M.W. Kanters, T. Denton, A. Shokoufandeh, L.M.J. Florack, B.M. ter Haar Romeny.

Submitted to ScaleSpace '07.

Image Reconstruction with Canonical Subsets of Multiple Types of Scale Space Interest Points.

T. Denton, F.M.W. Kanters, A. Shokoufandeh, L.M.J. Florack.

Submitted to IEEE Conference on Computer Vision and Pattern Recognition '07.

5.1 Introduction

In chapter 2 the α -scale spaces are introduced. We already noted that there are several special points in a scale space of an image, so called scale space interest points. Scale space interest points are often used for image matching and a comparison between different types of interest points with respect to image matching can be found in the work by Mikolajczyk and Schmid [110] and Platel [119]. However, points that are good for image matching might not be optimal for image reconstruction. Therefore we present in this chapter 10 different types of scale space interest points that are commonly used and we compare these types of interest points with respect to image reconstruction (rather than matching). For the reconstruction we used the Sobolev reconstruction presented in chapter 3 and for the evaluation we used the best image quality measures of chapter 4. In section 5.2 the 10 different types of interest points used in our comparison are defined. These 10 types of interest points are calculated in 8.000 image patches of the van Hateren database of natural images [134]. Reconstructions are made using these points and the results are evaluated using objective error measures, presented in section 5.3. It is shown that combining different types of scale space interest points can be beneficial for image reconstruction. It is however not trivial *how* to combine the different types of points. In section 5.4 a method is presented to combine scale space interest points using canonical sets. Finally the conclusions and discussion are presented in section 5.5.

5.2 Interest points

Consider a continuous signal $f : \mathbb{R}^d \rightarrow \mathbb{R}$. The linear scale space representation $u : \mathbb{R}^d \times \mathbb{R}_+ \rightarrow \mathbb{R}$ of f is defined as the solution of the heat equation:

$$\begin{cases} \frac{\partial}{\partial s} u = \Delta u \\ \lim_{s \downarrow 0} u(\cdot, s) = f(\cdot) \end{cases} \quad (5.1)$$

Where s is the scale. The unique solution to this equation leads to convolution with a Gaussian kernel, hence the name Gaussian scale space¹. For 2D images the scale space u can be obtained by the convolution:

$$u(x, y, s) = (G_s * f)(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x', y') G_s(x - x', y - y') dx' dy' \quad (5.2)$$

with f the original image and G_s a Gaussian of scale s defined by:

$$G_s(x, y) = \frac{1}{4\pi s} e^{-\frac{x^2 + y^2}{4s}} \quad (5.3)$$

Spatial derivatives of the image can be calculated by convolution with a derivative of a Gaussian:

$$\partial_{\nu^1, \dots, \nu^n} u(x, y, s) = (\partial_{\nu^1, \dots, \nu^n} G_s * f)(x, y) \quad (5.4)$$

¹It is shown that using reasonable axioms, a complete α -parameterized class of kernels exists resulting in so-called α -scale spaces [39, 118, 30]. The Gaussian scale space is one specific case ($\alpha = 1$).

with ν^1, \dots, ν^n the spatial indices (for a 2D image this can be any combination of x and y). For further reference we will use the short notation $u_{s,\nu^1\dots\nu^n}$ for $\partial_{\nu^1,\dots,\nu^n}u(x,y,s)$. Lindeberg [96, 99] introduced scaled derivatives where the γ -normalized derivative operator is defined by:

$$\partial_{\xi,\gamma\text{-norm}} = s^{\frac{\gamma}{2}} \partial_x \quad (5.5)$$

which corresponds to the change of variables

$$\xi = \frac{x}{s^{\frac{\gamma}{2}}} \quad (5.6)$$

In the special case when $\gamma = 1$, these ξ -coordinates and their associated normalized derivative operator are *dimensionless* and the corresponding derivatives are scale invariant [44].

Spatial derivatives in x - and y -direction are however not rotation invariant, which is desirable for detecting interest points. For this reason *gauge coordinates* [8] are introduced. In a first order gauge coordinate system one coordinate axis is locally fixed to the gradient of the image and one axis perpendicular to it. The local first order gauge coordinate frame (\vec{v}, \vec{w}) is defined by:

$$\vec{w} = \left(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y} \right) \quad (5.7)$$

$$\vec{v} = \left(\frac{\partial u}{\partial y}, -\frac{\partial u}{\partial x} \right) \quad (5.8)$$

Since these coordinate frames are locally fixed to the image, derivatives with respect to these coordinates will always be rotation invariant. For example, the first order derivatives are given by (short notation used):

$$u_w = \sqrt{u_x^2 + u_y^2} \quad (5.9)$$

$$u_v \doteq 0 \quad (5.10)$$

The first order derivative with respect to \vec{v} is zero since it is always pointing in the direction of the isophote, while the first order derivative with respect to \vec{w} is equal to the gradient since it is always pointing in the direction of the gradient.

5.2.1 Blobs

Scale-space blobs [94, 95, 100] are defined as the positive local maxima (or negative local minima) in space and scale of the normalized Laplacian of the image:

$$(x, y, s) = \underset{x,y,s}{\operatorname{arglmax}} \{ |s^\gamma (u_{s,xx} + u_{s,yy})| \} \quad (5.11)$$

with $\operatorname{arglmax}$ the arguments for which a local maximum is found and using γ -normalization with $\gamma = 1$. Lowe [103, 104] approximated the Laplacian by Difference of Gaussians (DoG). He showed that

$$u_{ks}(x, y) - u_s(x, y) \approx (k-1)s(u_{s,xx} + u_{s,yy}) \quad (5.12)$$

which is very efficient to calculate and has a low error if $k \approx 2$.

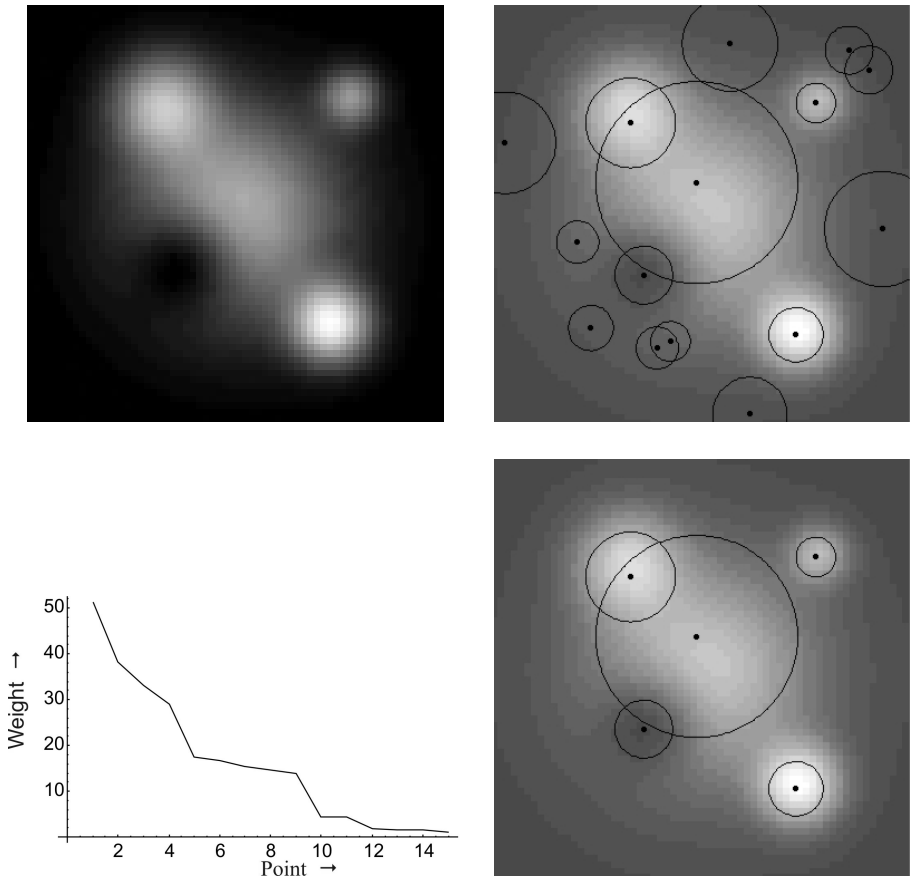


Figure 5.1: Laplacian Blobs in a synthetic image. Top left: Original synthetic image. Top right: all Laplacian blobs of the image, the radius relates to the scale of the blob. Bottom left: Weight (equal to the strength) of the Laplacian blobs. Bottom right: 5 Strongest Laplacian blobs.

Alternatively scale space blobs can be defined as the local maxima of the squared normalized determinant of the Hessian.

$$(x, y, s) = \operatorname{arglmax}_{x, y, s} \{s^{4\gamma} (u_{s,xx} u_{s,yy} - u_{s,xy}^2)^2\} \quad (5.13)$$

Again using γ -normalization with $\gamma = 1$. Blobs can be ordered in strength by the magnitude of the response of their respective filters [115, 93]. Figure 5.1 shows an example of Laplacian blobs on a synthetic image. The strength of the blobs is shown in the graph and the 5 strongest blobs are shown on the right. Note that the expected blobs indeed have a higher strength than the other detected blob structures. Figure 5.2 shows the result for Hessian blobs. Again the strength is shown in the graph and the strongest 7 blobs are shown on the bottom-right. Note that 2 extra blobs are detected with a high strength using this method. Figure 5.3 shows the two different types of blobs from a natural image.

5.2.2 Corner points

The isophote curvature κ is a measure of how *curved* lines of equal intensity are. The curvature κ is defined by:

$$\kappa = -\frac{u_{vv}}{u_w} \quad (5.14)$$

In order to give a stronger response near edges, the level curvature can be multiplied by the gradient magnitude to the power 3 [8, 56] which is referred to as an affine invariant corner detector:

$$\Theta = \kappa u_w^3 = -u_w^2 u_{vv} = 2u_x u_{xy} u_y - u_{xx} u_y^2 - u_x^2 u_{yy} \quad (5.15)$$

Note that Lindeberg [99] and Brunnström et al. [13] used a slightly different definition for the curvature:

$$\kappa = \frac{u_{vv}}{u_w} \quad (5.16)$$

and corresponding *re-scaled level curve curvature*:

$$|\kappa| = |u_w^2 u_{vv}| = |u_{xx} u_y^2 + u_x^2 u_{yy} - 2u_x u_{xy} u_y| \quad (5.17)$$

which results in the same local extrema. Corner points in the image can now be defined as points with high curvature and high intensity gradient:

$$(x, y, s) = \operatorname{arglmax}_{x, y, s} \{s^{2\gamma} (2u_{s,x} u_{s,xy} u_{s,y} - u_{s,xx} u_{s,y}^2 - u_{s,x}^2 u_{s,yy})\} \quad (5.18)$$

using γ -normalization with $\gamma = 7/8$ [99]. Note that for ordering the corner points in strength, the magnitude of the corresponding filter response has to be normalized with $\gamma = 1$ to make magnitude values at different scales comparable. Figure 5.4 shows the result of the corner detector. Note that the displacement of the corner points increases with the scale due to the blurring process and that there are some spurious corners due to responses to edges, especially near the boundaries. Figure 5.6 shows corner points from a natural image.

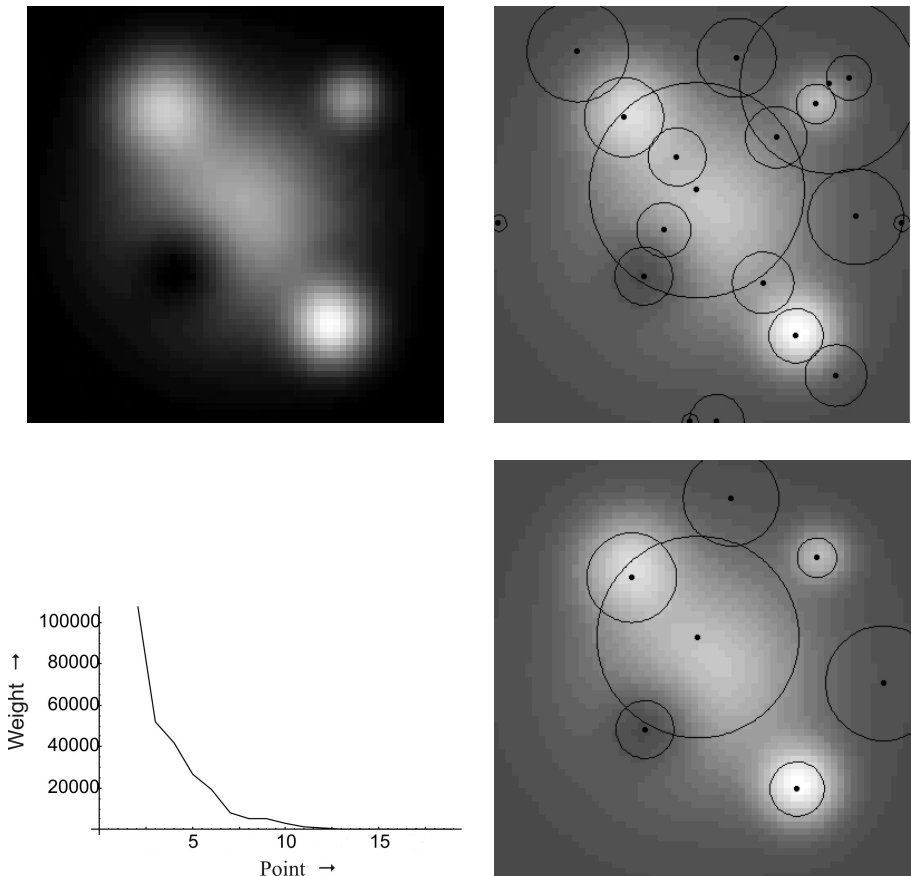


Figure 5.2: Hessian Blobs in a synthetic image. Top left: Original synthetic image. Top right: all Hessian blobs of the image, the radius relates to the scale of the blob. Bottom left: Weight (equal to the strength) of the Hessian blobs. Bottom right: 7 Strongest Hessian blobs.

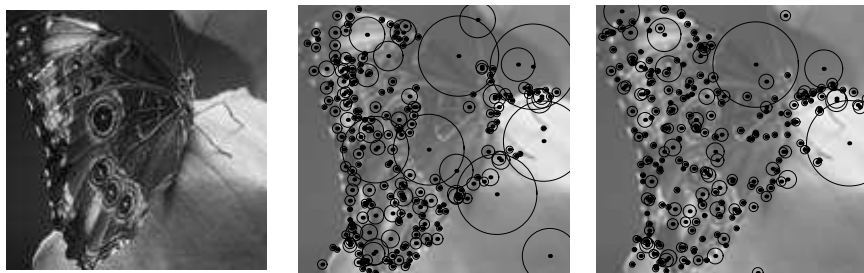


Figure 5.3: Blob points of the *butterfly.jpg* image projected on the original image. Left the original image, in the center the 250 strongest Laplacian blobs and right the 250 strongest Hessian blobs.

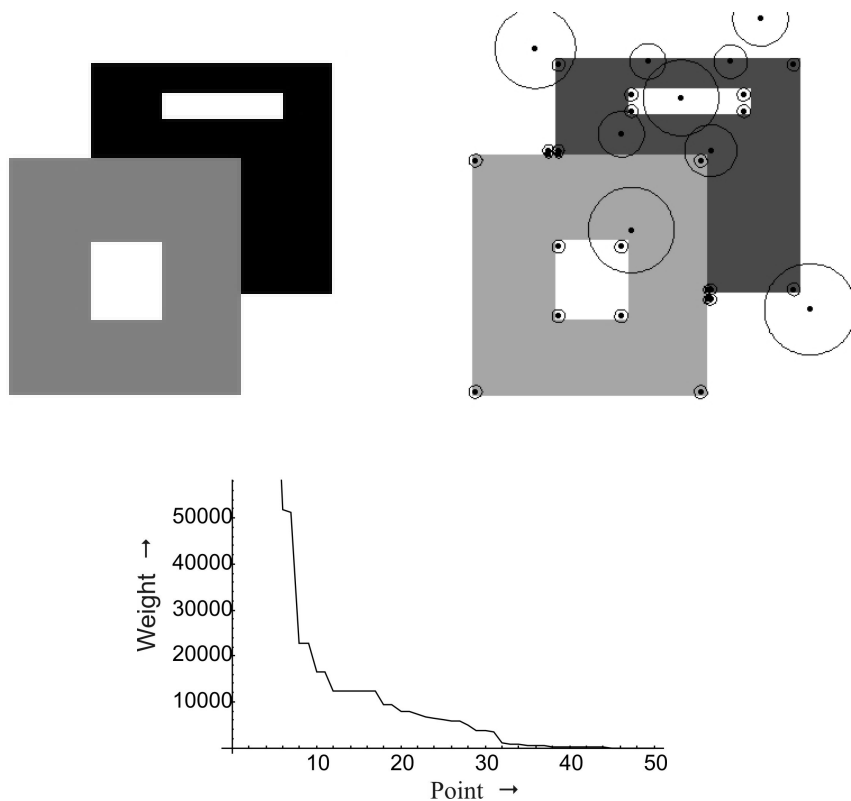


Figure 5.4: Corners in a synthetic image. Top left: Original synthetic image. Top right: 40 Strongest corner points. Bottom: Weight (equal to the strength) of the corner points.

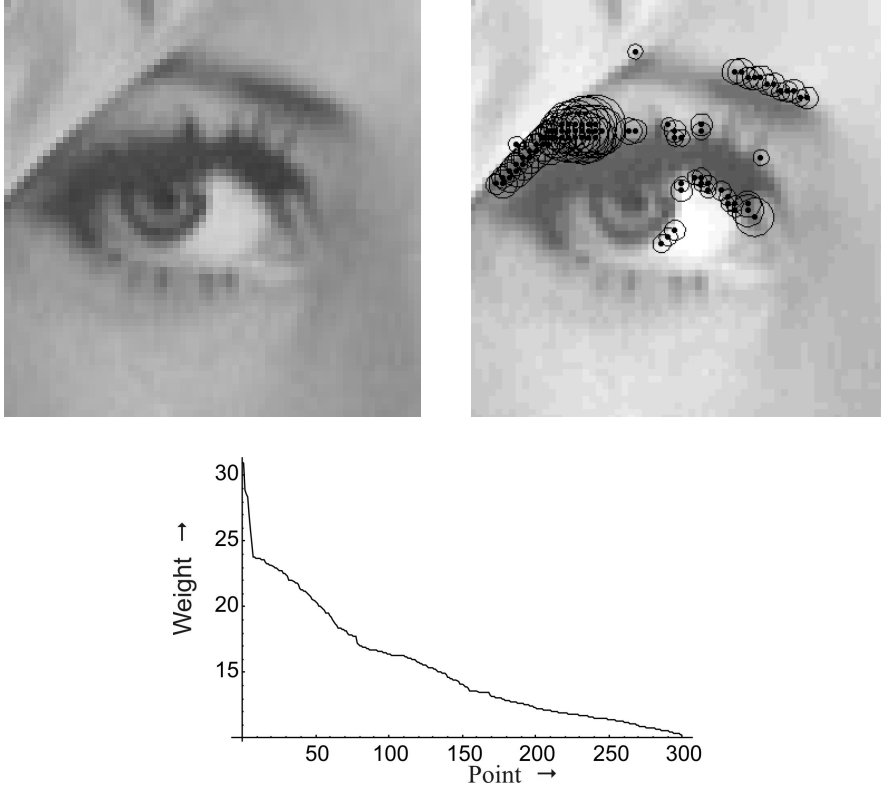


Figure 5.5: Edges in an image. Top left: Original image of Lena's eye. Top right: 80 Strongest edge points. Bottom: Weight (equal to the strength) of the edge points.

5.2.3 Edge points

Edges play an important role in the human visual system and thus it plays an important role in image analysis. If one considers a gauge coordinate system as described before, an edge can be seen as a curve where the gradient magnitude u_w is large. To find multi scale edge points this can be translated into two constraints [93, 115]:

$$\begin{cases} s^\gamma u_{ww} = 0 \\ \text{lmax}_s \{s^{\gamma/2} u_w\} \end{cases} \quad (5.19)$$

using γ -normalization with $\gamma = 1/2$ [98]. For edge strength, the gradient magnitude is used, re-normalized with $\gamma = 1$. Figure 5.5 shows the result for the edge points. The edge point strength is plotted in the graph and the 80 strongest edge points are shown on the right. Figure 5.6 shows edge points from a different natural image.

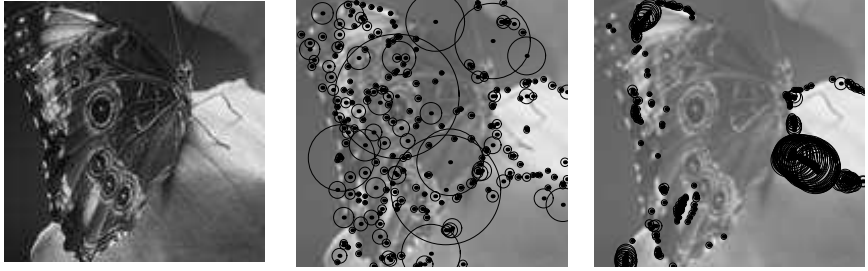


Figure 5.6: Corner points and Edge points of the *butterfly.jpg* image projected on the original image. Left the original image, in the center the 250 strongest corner points and right the 250 strongest edge points.

5.2.4 Ridge points

In topography a ridge is defined as a separator between regions from which water flows into different sinks. However, there have been several approaches to a mathematical formulation of a ridge. An overview of the developments can be found in the work by Koenderink and van Doorn [84]. Most commonly used is the definition of a ridge as being a connected set of points for which the intensity assumes a local extremum (maximum for bright ridges, minimum for dark ridges) in the main principal curvature direction [58, 98]. If one is interested in ridge strength, without a response to blobs, the *square of the γ -normalized Hessian eigenvalues* can be used [98]:

$$\mathcal{A}_\gamma u(x, y, s) = \{s^{2\gamma}((u_{s,xx} - u_{s,yy})^2 + 4u_{s,xy}^2)\} \quad (5.20)$$

using $\gamma = 3/4$ [98]. Ridge points are defined as the local extrema of this ridge strength:

$$(x, y, s) = \underset{x, y, s}{\operatorname{arglmax}} \{\mathcal{A}_\gamma u(x, y, s)\} \quad (5.21)$$

Note that for ordering the ridge points in strength, again the magnitude of the corresponding filter response has to be normalized with $\gamma = 1$ to make magnitude values at different scales comparable. Figure 5.7 shows the ridges of an image of a map. Again the ridge strength is shown in the graph and the 90 strongest ridge points are shown on the right.

5.2.5 Top points or bifurcation points

Local extrema and saddle points are important features in a 2D image. These spatial critical points (at a fixed scale) are defined by:

$$\nabla u_s = (u_{s,x}, u_{s,y})^T = \mathbf{0} \quad (5.22)$$

Many people have investigated the behavior of spatial critical points over scale, where these points will form paths (which are called critical paths). Special point in scale space are points where these paths come together, which are referred to as catastrophe points [129, 51, 23]. Lindeberg et al. referred to these points as bifurcation points [100, 97] and Johansen referred to these points as top-points [69, 67, 68]. Koenderink et

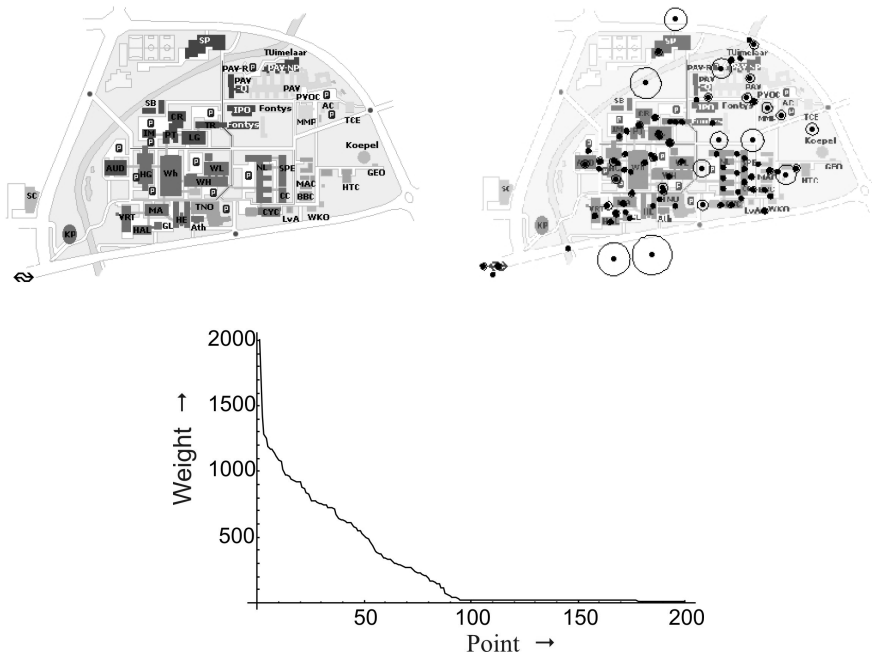


Figure 5.7: Ridges in an image. Top left: Original image. Top right: 90 Strongest ridges points. Bottom: Weight (equal to the strength) of the ridge points.

al. [80, 82, 81] and Florack et al. [43, 40, 38] also studied the behavior of critical points in scale space. For *generic* 2D images only two catastrophe points exist: Annihilation points of an extremum (maximum or minimum) path with a saddle path or creation points of an extremum path with a saddle path. These points (x, y, s) are defined by:

$$\begin{cases} \nabla u_s = (u_{s,x}, u_{s,y})^T = \mathbf{0} \\ \det \mathcal{H}(u_s) = u_{s,xx}u_{s,yy} - u_{s,xy}^2 = 0 \end{cases} \quad (5.23)$$

where $\mathcal{H}(u_s)$ is the 2-nd order Hessian matrix defined by:

$$\mathcal{H}(u_s) = \begin{pmatrix} u_{s,xx} & u_{s,xy} \\ u_{s,xy} & u_{s,yy} \end{pmatrix} \quad (5.24)$$

In practice these points can be detected by intersecting the 3D zero-crossing surfaces $u_{s,x} = 0$, $u_{s,y} = 0$ and $u_{s,xx}u_{s,yy} - u_{s,xy}^2 = 0$. Platel et al. and Kanters et al. showed that top points can be ordered by a stability norm called differential TV-norm [120, 75]. The amount of structure contained in a *spatial* area around a critical point can be quantified by the *total (quadratic) variation* (TV) norm over that area [12]. By using a spatial Taylor series around a considered critical point the TV-norm simplifies to Eqn. (5.25) which is referred to as the *differential TV-norm* [120].

$$\text{diff tv} = 4s^2(u_{s,xx}^2 + u_{s,yy}^2 + 2u_{s,xy}^2) \quad (5.25)$$

An example of top points of a synthetic image is shown in figure 5.8. On the left side, the original image with the critical paths and the top points are shown in a 3D view. The third dimension represents scale. In the center the strength of the top points is shown in a graph. On the right the top points are shown projected on the original image. Figure 5.11 shows an example of top points of a natural image.

It has been shown that top points of the Laplacian of an image (rather than top points of the gray level image itself) can be used for image matching [120] and reconstruction [72]. They are defined by:

$$\begin{cases} u_{s,xxx} + u_{s,xyy} = 0 \\ u_{s,xyx} + u_{s,yyy} = 0 \\ (u_{s,xxx} + u_{s,xyy})(u_{s,xyx} + u_{s,yyy}) - (u_{s,xxxy} + u_{s,xyyy})^2 = 0 \end{cases} \quad (5.26)$$

Laplacian top points can be seen as points in scale space where extrema and saddles of the Laplacian of the image annihilate. Instead of describing the behavior of local extrema in scale space, Laplacian top points describe the behavior of blobs through scale. Figure 5.9 shows the Laplacian top points of a synthetic image.

5.2.6 Scale space saddle points

Kuijper et al. [85, 86, 87] considered scale-space saddle points, which are defined by:

$$\nabla u(x, y, s) = (u_x, u_y, u_s)^T = \mathbf{0} \quad (5.27)$$

Since $u_x = 0$ and $u_y = 0$ these points are also on the critical paths like the top points. By definition the derivative to scale is equal to the Laplacian, thus we can rewrite (5.27) to:

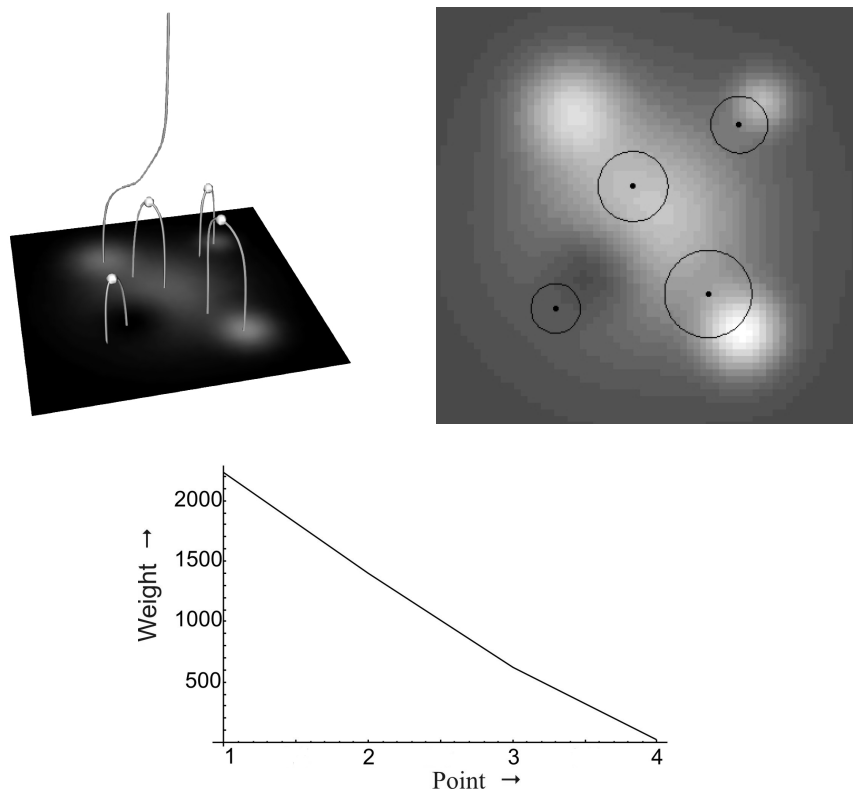


Figure 5.8: Top points of a synthetic image. Top left: Original image with the critical paths and top points (third dimension represents scale). Top right: All 4 top points projected on the image plane. Bottom: Weight (equal to the strength) of the top points.

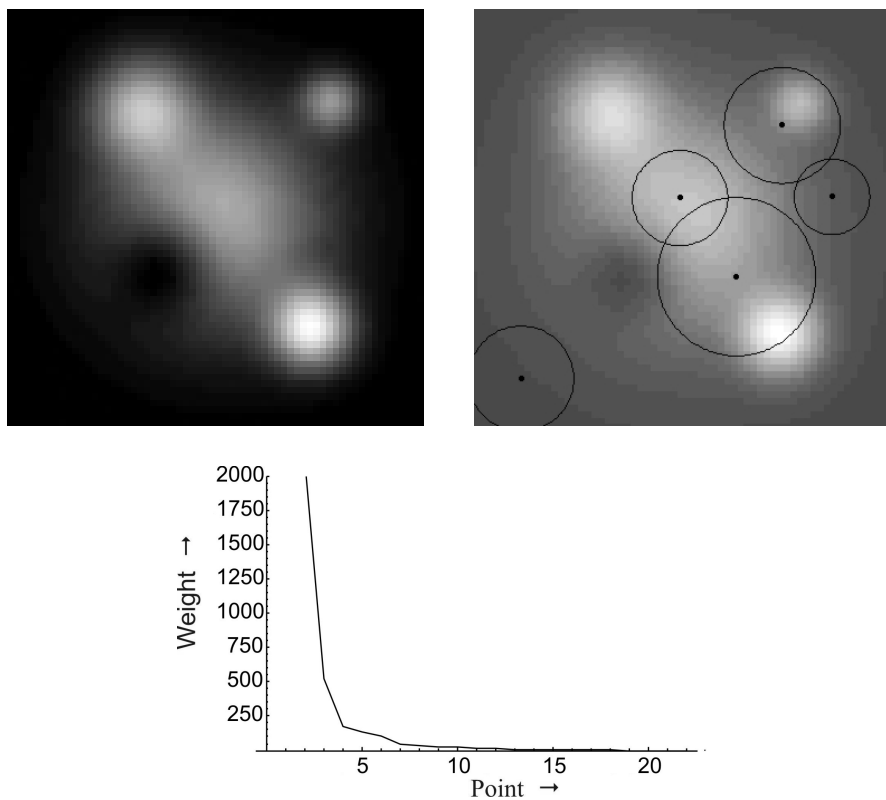


Figure 5.9: Laplacian top points of a synthetic image. Top left: Original image. Top right: Strongest 5 Laplacian top points projected on the image plane. Bottom: Weight (equal to the strength) of the Laplacian top points.

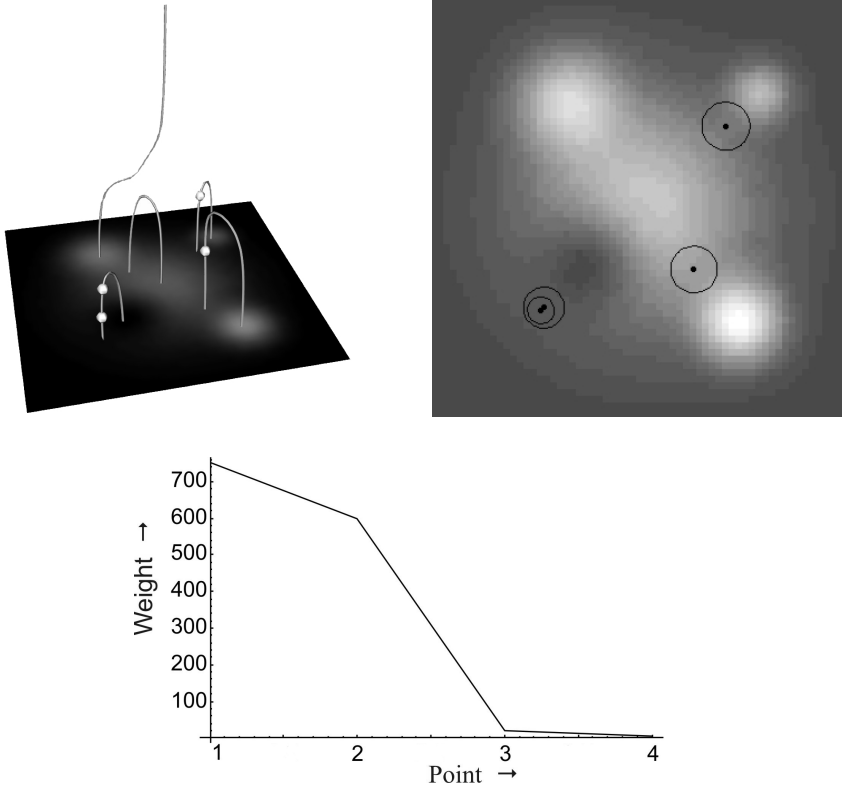


Figure 5.10: Scale space saddle points of a synthetic image. Top left: Original image with the critical paths and scale space saddle points (third dimension represents scale). Top right: All 4 scale space saddle points projected on the image plane. Bottom: Weight (equal to the strength) of the scale space saddle points.

$$\begin{cases} u_{s,x} = 0 \\ u_{s,y} = 0 \\ u_{s,xx} + u_{s,yy} = 0 \end{cases} \quad (5.28)$$

Scale-space saddle points can thus be easily detected by intersecting the 3D zero-crossing surfaces $u_{s,x} = 0$, $u_{s,y} = 0$ and $u_{s,xx} + u_{s,yy} = 0$. Also these points can be ordered by *differential TV-norm* (5.25). Note that scale space saddles of the Laplacian of the image are in fact equal to Laplacian blobs. Figure 5.10 shows an example of scale space saddles of a synthetic image. On the left side, the original image with the critical paths and the scale space saddle points are shown in a 3D view. The third dimension represents scale. In the center the strength of the scale space saddle points is shown in a graph. On the right the scale space saddle points are shown projected on the original image. Note that multiple scale space saddles can occur on a single saddle path. Figure 5.11 shows an example of scale space saddle points of a natural image.

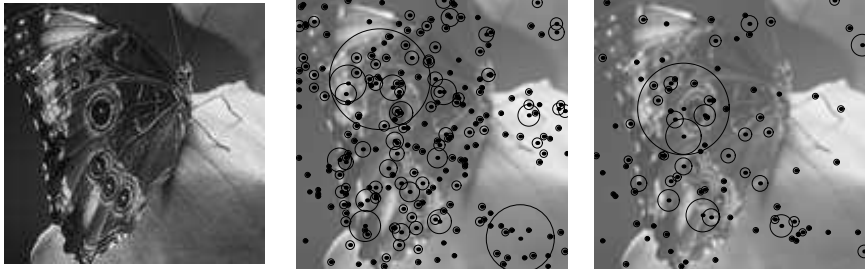


Figure 5.11: Top points and scale space saddle points of the *butterfly.jpg* image projected on the original image. Left the original image, in the center all 235 top points and right all 93 scale space saddle points.

5.2.7 Scale adapted Hessian-Laplace points

Mikolajczyk and Schmid [111, 110] introduced a hybrid blob detection method where the local spatial maxima of the square of the determinant of the Hessian matrix (introduced by Lowe [104] to eliminate edge response) are combined with the local scale maxima of the Laplacian.

$$\begin{cases} (x, y) = \underset{x, y}{\operatorname{arglmax}} \{s^4(u_{s,xx}u_{s,yy} - u_{s,xy}^2)^2\} \\ s = \underset{s}{\operatorname{arglmax}} \{s^2(u_{s,xx} + u_{s,yy})\} \end{cases} \quad (5.29)$$

For our experiments the implementation of Mikolajczyk et al. [108] is used. An explanation of the parameters can be found in their paper [109] (no thresholding was used on the strength of the points). Figure 5.12 shows an example of Hessian-Laplace points of a synthetic image. For ordering the points by strength, the Laplacian is used. Figure 5.13 shows an example of Hessian Laplace points for a natural image.

5.2.8 Scale adapted Harris-Laplace points

Mikolajczyk and Schmid [109] also introduced a scale adapted version of the Harris corner detector [59]. Consider the scale-adapted second moment matrix:

$$\mu(s_D, s_I) = s_D G_{s_I} * \begin{pmatrix} u_{s_D,xx} & u_{s_D,xy} \\ u_{s_D,xy} & u_{s_D,yy} \end{pmatrix} \quad (5.30)$$

with s_D the *differentiation* scale, s_I the *integration* scale and G_{s_I} a Gaussian at scale s_I as defined in (5.3). The Harris measure [59] combines the trace and determinant of this matrix as a measure for cornerness. Scale selection is based on the local maxima over scale of the Laplacian. The scale adapted Harris-Laplace points are defined as:

$$\begin{cases} (x, y) = \underset{x, y}{\operatorname{arglmax}} \{\det(\mu(s_D, s_I)) - \alpha \operatorname{trace}^2(\mu(s_D, s_I))\} \\ s = \underset{s}{\operatorname{arglmax}} \{s^2(u_{s,xx} + u_{s,yy})\} \end{cases} \quad (5.31)$$

Note that in practice, $s_D = \beta s_I$ with β a suitable constant. For our experiments the implementation of Mikolajczyk et al. [108] is used. An explanation of the parameters

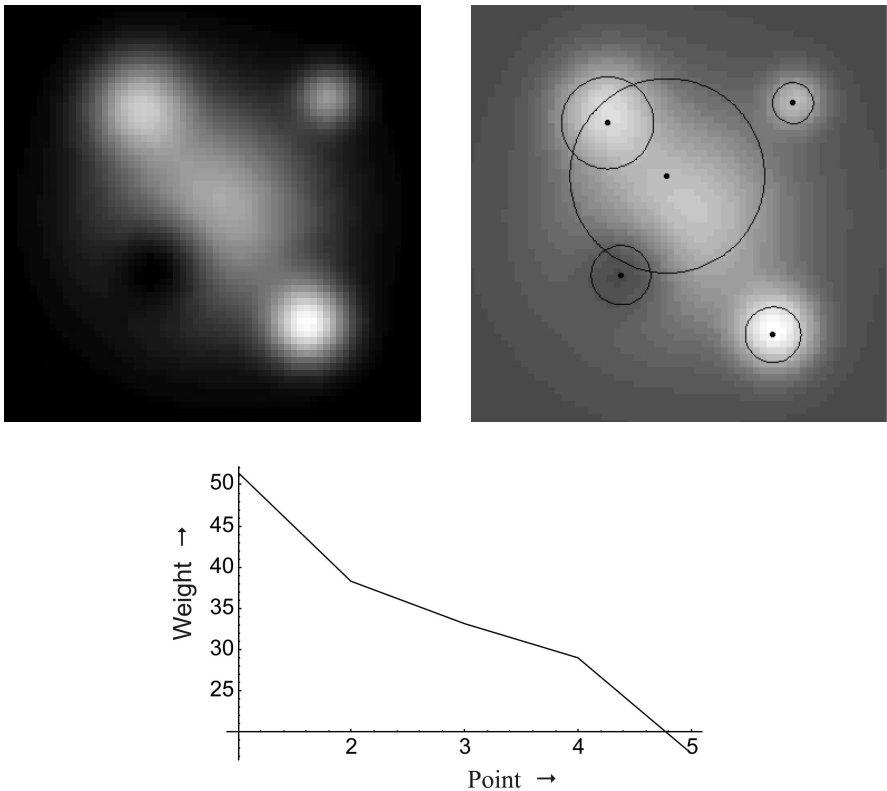


Figure 5.12: Hessian-Laplace points of a synthetic image. Top left: Original synthetic image. Top right: All 5 Hessian-Laplace points. Bottom: Weight (equal to the strength) of the Hessian-Laplace points.

(e.g. α) can be found in their paper [109]. Again the Laplacian is used as strength measure. Figure 5.13 shows an example of Harris Laplace points for a natural image. For details on the implementation of the other presented scale space interest point detectors the reader is referred to the implementation notes “Interest point detectors” on page 96.

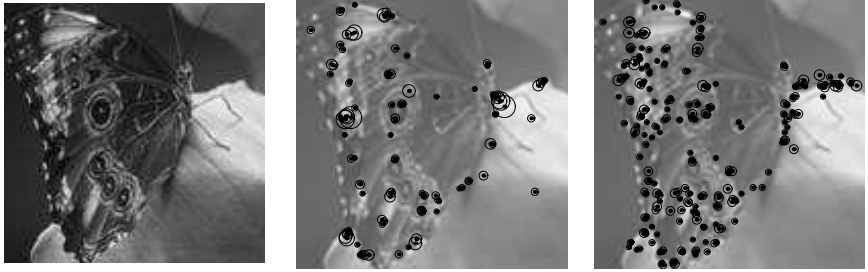


Figure 5.13: Harris Laplace points and Hessian Laplace points of the *butterfly.jpg* image projected on the original image. Left the original image, in the center all 110 Harris Laplace points and right the strongest 250 Hessian Laplace points.

5.3 Reconstruction experiments

5.3.1 Comparison of scale space interest points for reconstruction

To compare the different types of scale space interest points described in section 5.2 for image reconstruction, experiments on two sets of images are performed. The first set of images consists of 8.000 random patches (64×64 pixels) of the van Hateren database of natural images [134]. The second set consists of 12 images (128×128 pixels) sub-sampled from images of a digital camera. For all images in both sets, all 10 described types of scale space interest points are calculated. For the second set, the result of the scale space interest point detectors is shown in appendix C. Using the Sobolev reconstruction described in chapter 3, reconstructions are made for all images and all 10 different point sets (80.000 reconstructions in total for the first set, 120 for the second set). For the first set, a maximum of 50 points is used for each reconstruction, while for the second set a maximum of 400 points is used for each reconstruction. The reconstruction results of the second set are shown in appendix D. All reconstructions use up to 4-th order derivatives in each interest point and an SVD tolerance of 10^{-5} . For a quantitative comparison of the scale space interest points with respect to image reconstruction, all reconstructions are compared to the corresponding original images using 6 objective image quality measures presented in chapter 4: The Mean Squared Error (MSE, section 4.2.1.1), the Pratt Measure (PM, section 4.2.1.16), the Median Spectral Phase Block Distortion (MSPBD, section 4.2.1.22), the Multi-Scale Differential Error (MSDE, section 4.2.2), the Fast Multi-Scale Differential Error (FMSDE, implementation notes “Fast MSDE” on page 67) and the Structural SIMilarity measure (SSIM, section 4.2.3.2). The mean and standard deviation of the errors is calculated for both test sets. Table 5.1 shows the results for the first set and in table 5.2 the results for the second set are shown. Note that in general blobs, ridge points and corner points are better suitable for image reconstruction

Implementation Notes

Interest point detectors

The 10 different types of scale space interest points presented can be detected in an image using derivatives of scale spaces of that image. For our experiments, derivative scale spaces up to 4-th order (and up to second order in gauge coordinates) of the image are calculated. For detecting the interest points from the derivative scale spaces, either local maxima (blobs, corner points and ridges), intersections of zero-crossings (top points, top points of the Laplacian and scale space saddles) or a combination of both (edge points) have to be found.

To detect local maxima in the volume of a scale space (e.g. local maxima of the absolute value of the Laplacian scale space for Laplacian blobs), each voxel and its 26 neighboring voxels are evaluated. If the center voxel is larger than all neighboring voxels, a local maximum is found.

To detect intersections of zero-crossings in the volumes of the corresponding scale spaces (e.g. intersections of zero-crossings of the x-derivative, y-derivative and Laplacian for scale space saddles), each voxel is evaluated in the three corresponding volumes. A single voxel contains a zero-crossing of a volume if and only if at least one corner point value of the voxel differs in sign from the other corner point values of the voxel. A voxel can only have an intersection of three zero-crossing surfaces if the corresponding voxel contains zero-crossings in all three volumes. To speed up detection, this condition is checked first for each voxel in the volumes. For every possible candidate voxel, intersection points between the three surfaces are calculated using a marching cubes like algorithm [102]. In case of ambiguities all possible combinations are calculated. This can result in a slight overestimation of the number of interest points.

For the edge points, every voxel in the u_{ww} scale space is evaluated. If a candidate voxel contains a zero-crossing, the value of the corresponding voxel in the u_w scale space is compared to the 9 neighboring voxels one scale higher and 9 neighboring voxels one scale lower than the candidate voxel in the u_w scale space. If the value in the candidate voxel is larger than the other 18 voxels, an edge point is found.

than the other scale space interest points and that Harris Laplace points, often used for image matching, are not very suitable for image reconstruction. One reason is that the number of Harris Laplace points is in general very low. However, adding features in this low number of points might improve the reconstruction quality, while maintaining a comparable number of constraints as the reconstructions from blobs for example. Note that top points of the Laplacian perform relatively bad in the first set, while in the second set they perform very well, even among the best. A possible cause can be the fact that the first set contains many patches with very little structure, while in the second set

every image contains much structure. It is possible that top points of the Laplacian perform bad in images with little structure. Also the ordering of the top points of the Laplacian might be a cause, since for the first set only a quarter of the points is used in general, while for the second set almost half the number of points is used. As can be seen in appendix D, the visual quality of reconstructions from top points of the Laplacian in the second set supports the low errors in table 5.2. For reconstructions with lower quality there is a deviation in ordering for the different image quality measures (e.g. according to the MSE the Harris Laplace points are the worst points for reconstruction, while according to the SSIM the top points of the Laplacian are the worst points for reconstruction).

Reconstruction from:	MSE Mean (σ)	PM Mean (σ)	MSPBD Mean (σ)	MSDE Mean (σ)	FMSDE Mean (σ)	SSIM Mean (σ)
Laplacian blobs (128)(50)	0.0017 (0.0063)	0.5676 (0.1286)	0.0015 (0.0032)	0.0225 (0.0177)	0.0103 (0.0088)	80.8663 (11.8492)
Hessian blobs (232)(50)	0.0020 (0.0066)	0.5434 (0.1414)	0.0017 (0.0037)	0.0249 (0.0209)	0.0113 (0.0101)	77.9351 (12.6089)
Ridge points (80)(48)	0.0028 (0.0078)	0.4775 (0.1505)	0.0023 (0.0051)	0.0310 (0.0270)	0.0139 (0.0127)	77.8055 (13.5613)
Corner points (110)(50)	0.0030 (0.0135)	0.4778 (0.1424)	0.0024 (0.0049)	0.0340 (0.0274)	0.0148 (0.0127)	74.1752 (13.8028)
Edge points (1215)(50)	0.0046 (0.0101)	0.3126 (0.1516)	0.0035 (0.0062)	0.0473 (0.0332)	0.0228 (0.0173)	67.5569 (14.7262)
Hessian Laplace points (39)(34)	0.0057 (0.0136)	0.3060 (0.1377)	0.0039 (0.0071)	0.0478 (0.0313)	0.0241 (0.0165)	67.5219 (16.0539)
Top points (58)(46)	0.0058 (0.0210)	0.3892 (0.1471)	0.0045 (0.0094)	0.0480 (0.0444)	0.0204 (0.0199)	67.4022 (16.2812)
Top points of the Laplacian (188)(50)	0.0070 (0.0152)	0.4098 (0.1357)	0.0060 (0.0115)	0.0599 (0.0555)	0.0240 (0.0229)	58.5503 (18.0585)
Scale space saddles (24)(24)	0.0084 (0.0306)	0.2704 (0.1215)	0.0055 (0.0107)	0.0593 (0.0529)	0.0263 (0.0251)	63.3623 (17.1148)
Harris Laplace points (11)(11)	0.0094 (0.0261)	0.1451 (0.0817)	0.0051 (0.0088)	0.0633 (0.0458)	0.0321 (0.0254)	60.3909 (18.8402)

Table 5.1: Error measures of reconstructions from scale space interest points for 8.000 random patches (64×64 pixels) of the van Hateren database of natural images [134], using a maximum of 50 interest points per reconstruction. The first number behind the point type is the mean number of interest points detected, the second number shows the mean number of points used for the reconstruction. The mean and standard deviation for all 8.000 patches are given. Note that the PM and the SSIM are similarity measures, while the others are error measures.

Reconstruction from:	MSE Mean (σ)	PM Mean (σ)	MSPBD Mean (σ)	MSDE Mean (σ)	FMSDE Mean (σ)	SSIM Mean (σ)
Laplacian blobs (558)(400)	0.0174 (0.0239)	0.7381 (0.0763)	0.0028 (0.0044)	0.0292 (0.0098)	0.0125 (0.0035)	80.6530 (7.8326)
Hessian blobs (958)(400)	0.0166 (0.0234)	0.7579 (0.0880)	0.0025 (0.0038)	0.0285 (0.0104)	0.0124 (0.0040)	80.6920 (8.8361)
Ridge points (258)(258)	0.0289 (0.0356)	0.5559 (0.1331)	0.0059 (0.0086)	0.0482 (0.0214)	0.0193 (0.0081)	71.6460 (11.9290)
Corner points (427)(389)	0.0197 (0.0253)	0.6940 (0.1030)	0.0033 (0.0045)	0.0349 (0.0115)	0.0142 (0.0043)	77.6370 (8.3314)
Edge points (4490)(400)	0.0410 (0.0315)	0.4184 (0.0947)	0.0064 (0.0058)	0.0702 (0.0248)	0.0357 (0.0113)	62.7980 (10.3860)
Hessian Laplace points (294)(260)	0.0370 (0.0326)	0.5700 (0.1131)	0.0052 (0.0068)	0.0544 (0.0189)	0.0288 (0.0076)	68.7970 (8.7003)
Top points (235)(235)	0.0301 (0.0246)	0.5351 (0.0893)	0.0055 (0.0051)	0.0526 (0.0179)	0.0216 (0.0065)	71.2280 (5.2343)
Top points of the Laplacian (888)(400)	0.0167 (0.0228)	0.7574 (0.0788)	0.0029 (0.0046)	0.0288 (0.0103)	0.0123 (0.0039)	80.7120 (7.8331)
Scale space saddles (95)(95)	0.0918 (0.0857)	0.2656 (0.0924)	0.0141 (0.0120)	0.1084 (0.0488)	0.0470 (0.0207)	51.2780 (11.6350)
Harris Laplace points (104)(104)	0.0669 (0.0472)	0.3544 (0.0792)	0.0111 (0.0099)	0.0852 (0.0273)	0.0423 (0.0133)	57.4540 (10.9340)

Table 5.2: Error measures of reconstructions from scale space interest points for 12 images (128×128 pixels) from a digital camera, using a maximum of 400 interest points per reconstruction. The first number behind the point type is the mean number of interest points detected, the second number shows the mean number of points used for the reconstruction. The mean and standard deviation for the 12 images are given. Note that the PM and the SSIM are similarity measures, while the others are error measures.

5.3.2 Stability of the interest points

Another method to compare the descriptive power of scale space interest points is to compare the stability of the reconstruction with respect to the localization error of the scale space interest points. If a point set contains much local image information, randomizing the position of the points should degrade the reconstruction quality. To simulate the localization error, normally distributed noise with a varying standard deviation is added to the interest point positions and the corresponding features are recalculated. This is done for all images and all 10 types of interest points of the second test set. Reconstructions of these noisy point sets with varying σ are made using the Sobolev reconstruction described in chapter 3. A maximum of 100 points per reconstruction is used to amplify the effects of the noise. Again, all reconstructions use up to 4-th order derivatives in each interest point and an SVD tolerance of 10^{-5} . For quantitative analysis two image quality measures of chapter 4 are calculated: The Root Mean Squared error (RMS, section 4.2.1.2) and the Multi-Scale Differential Error (MSDE, section 4.2.2). The mean RMS error and the mean MSDE over all images in the second test set for all 10 types of interest points and standard deviation σ ranging from $\sigma = 1$ pixel to $\sigma = 20$ pixels are presented in appendix E. Reconstructions from corner points, blobs, ridge points and top points show an increasing error for increasing σ . However, for edge points and Hessian Laplace points the error *decreases* with increasing σ . This means that randomizing these point positions actually improves reconstruction quality. For edge points this can be explained since many edge points close together contain much redundant information. Randomizing these points removes some of that redundancy and improves the reconstruction quality. Reconstructions from scale space saddles and Harris Laplace points do not show much difference for various σ . Note that these stability results support the earlier result that some of the scale space interest points (such as corner points, blobs and ridge points) perform better for image reconstruction than others.

5.4 Combining interest points using canonical sets

5.4.1 Introduction

As can be seen from the reconstruction results in appendix D, the different types of interest points capture different aspects of image structure. Nielsen and Lillholm showed that a combination of different types of interest points can improve the reconstruction quality substantially [115, 93]. It is however not clear *how* exactly these points should be combined. Specifically, given a large set of different types of interest points, how should one pick points for an optimal subset? Consider for example the test image *butterfly.jpg* from the second test set. All 10 types of scale space interest points previously defined are calculated and combined in one large point set. One possible option to do this, is to combine all types of points in a large set and re-order all points using their differential TV-norm or a similarly defined strength measure. The result of the 200 strongest combined interest points of the test image ordered by TV-norm and the corresponding reconstruction is shown in figure 5.14.

Note that the quality of the combined scale space interest point reconstruction is lower than some of the separate interest point reconstructions shown in appendix D. The reason for this is that in the combined point set many points that are close to each other

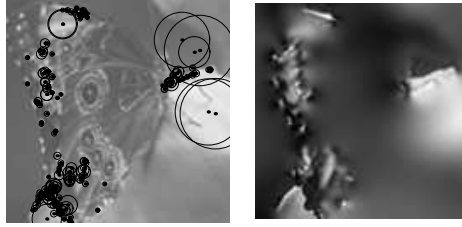


Figure 5.14: Left: 200 strongest combined scale space interest points of the *butterfly.jpg* image ordered by differential TV-norm. Right: Reconstruction of the *butterfly.jpg* image using the 200 strongest combined scale space interest points ordered by differential TV-norm.

will share a similar differential TV-norm (or other strength measure). Using strength measures as the sole criteria for ordering interest points may result in selecting points close to each other and consequently poor reconstruction results since these points will contain much redundant information. Table 5.3 shows the MSDE for reconstructions from the 200 strongest scale space interest points of different types and 200 of the strongest combined scale space interest points using the TV-norm to rank the points. Note that the MSDE of the reconstruction from combined scale space interest points, ordered by TV-norm, is higher than reconstructions from the best single type interest points.

Reconstruction from	MSDE	Reconstruction from	MSDE
Top points of the Laplacian	0.0723	Hessian Laplace points	0.2527
Hessian blobs	0.0858	Harris Laplace points	0.2784
Corner points	0.0928	Edge points	0.2246
Laplacian blobs	0.0989	Scale space saddles	0.2460
Top points	0.0963	Combined interest	
Ridge points	0.1493	points using TV-norm	0.1367

Table 5.3: MSDE of reconstructions from separate scale space interest points.

5.4.2 Canonical set framework

In this section, we propose an optimization framework for selecting scale space interest points for image reconstruction using canonical sets [25, 116] of scale space features. We will formulate the feature selection problem as a quadratic optimization problem and propose an approximation algorithm for its solution. Using only the strength measure of points can result in the selection from the combined point set of too many points that are close to each other. In order to take into account the spatial distance between points as well as the strength of the points, we select canonical subsets of the combined point sets. Canonical sets are subsets of points with special properties, namely:

- I. Points in the canonical set are minimally similar.
- II. Points outside the canonical set are maximally similar to points in the canonical set.
- III. Points in the canonical set have high stability compared to elements outside the set.

The canonical set problem is formulated in terms of a quadratic integer programming optimization. Many problems of this type are known to be intractable [49], but they admit good approximation algorithms [54]. In what follows, we give a brief overview of the problem formulation and its solution (for an in-depth treatment of the problem the reader is referred to [116, 26, 25]).

The input set to the canonical set problem consists of a set of points $\mathcal{P} = \{p_1, \dots, p_n\}$, an associated set of strength (stability) measures $\{t_1, \dots, t_n\}$, $t_i \in \mathbb{R}^+$ $1 \leq i \leq n$, and a similarity function $\mathcal{W} : \mathcal{P} \times \mathcal{P} \rightarrow \mathbb{R}^+$. In this work we let t_i equal the differential TV-norm of point p_i (See Eqn. 5.25), and define the similarity of two points, p_i and p_j , as

$$\mathcal{W}_{ij} = \frac{1}{1 + d_{ij}},$$

where d_{ij} denotes the Euclidean distance between the points p_i and p_j .

The problem is formulated as a multi-objective quadratic integer program where the outcome of the optimization will determine whether each point is in the canonical set, \mathcal{P}^* , or not. Specifically, for each point, p_i , an indicator variable, y_i , is used that will be equal to 1 if $p_i \in \mathcal{P}^*$ and -1 otherwise. Using these indicator variables, it can be shown that the aforementioned properties I, II, and III can be stated as optimization objectives:

$$\text{Minimize} \quad \frac{3}{4} \sum_{i,j} \mathcal{W}_{ij} y_i y_j + \frac{1}{2} \sum_{i=1}^n y_i \sum_{j=1}^n \mathcal{W}_{ij} + \frac{3}{4} \sum_{ij} \mathcal{W}_{ij} \quad (5.32)$$

$$\text{Minimize} \quad \frac{1}{2} \sum_{i=1}^n t_i (1 - y_i) \quad (5.33)$$

$$\text{Subject to} \quad y_i \in \{-1, +1\}, \quad \forall 1 \leq i \leq n. \quad (5.34)$$

The optimal solution to this integer program is a vector $\mathbf{y} = [y_1, \dots, y_n]^T$, indicating which points belong to the canonical set.

Vector labeling and lifting [53] are used to relax and reformulate the problem as a semidefinite program. To get rid of the linear terms we increase the dimension of the indicator vector by 1 and introduce a set indicator variable, y_{n+1} , which acts as a reference for membership in the canonical set. In an optimal solution, p_i is a member of the canonical set only if $y_i = y_{n+1}$. The integrality constraints are removed by substituting a vector for each indicator variable, that is, we replace each indicator variable y_i , $1 \leq i \leq n+1$, with a vector $x_i \in S_{n+1}$, where S_{n+1} is the unit sphere in \mathbb{R}^{n+1} . Let $w_\Sigma = \sum_{i,j} \mathcal{W}_{ij}$, define \mathbf{d} as a column vector in \mathbb{R}^n whose i^{th} entry has value $d_i = \sum_{j=1}^n \mathcal{W}_{ij}$, and let $t_\Sigma = \sum_{i=1}^n t_i$, and $\hat{\mathbf{0}}$ be an $n \times n$ matrix of zeros. The objectives are then encoded into matrices \mathcal{C} and \mathcal{T} ,

$$\mathcal{C} = \begin{bmatrix} \frac{3}{4}\mathcal{W} & \frac{1}{4}\mathbf{d} \\ \frac{1}{4}\mathbf{d}^T & \frac{3}{4}w_\Sigma \end{bmatrix}, \mathcal{T} = \begin{bmatrix} \hat{\mathbf{0}} & -\frac{1}{4}\mathbf{t} \\ -\frac{1}{4}\mathbf{t}^T & \frac{1}{2}t_\Sigma \end{bmatrix}, \quad (5.35)$$

where \mathbf{t} is a vector in \mathbb{R}^n whose i^{th} entry is t_i (the strength of point i).

These objectives are combined using Pareto optimality by defining a parameter $\alpha \in [0, 1]$, and defining the matrix \mathcal{Q} as a weighted convex combination of \mathcal{C} and \mathcal{T} , where $\mathcal{Q} = \alpha\mathcal{C} + (1 - \alpha)\mathcal{T}$. This implies that for any given α the combined objective is convex and a solution will be optimal for that α . The semidefinite program formulation of the canonical set can be stated as

CANSET:

$$\text{Minimize} \quad \mathcal{Q} \bullet \mathcal{X} \quad (5.36)$$

$$\text{Subject to} \quad \mathcal{D}_i \bullet \mathcal{X} \geq 0, \quad \forall i = 1 \dots, m, \quad (5.37)$$

$$\mathcal{X} \succeq 0, \quad (5.38)$$

where $m = n + 3$ and denotes the number of constraint matrices, and $\mathcal{X} \succeq 0$ denotes a positive semidefinite matrix. The notation $\mathcal{A} \bullet \mathcal{B}$ denotes the Frobenius inner product of matrices \mathcal{A} and \mathcal{B} , i.e., $\mathcal{A} \bullet \mathcal{B} = \text{Trace}(\mathcal{A}^T \mathcal{B})$. The first $n + 1$ constraint matrices, $\mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_{n+1}$, are all zeros with a single 1 that moves along the main diagonal, enforcing the $x_i^T x_i = 1$ constraints from Eqn. (5.34). The matrices \mathcal{D}_{n+2} and \mathcal{D}_{n+3} , encode constraints that are used to bound the size of the canonical set.

Once the solution to this semidefinite program is computed, a rounding step is performed to obtain an approximate integer solution. This step identifies the set of values for indicator variables y_1, \dots, y_n . We use a standard rounding scheme based on Cholesky decomposition and a multivariate normal hyperplane method [135].

The parameter α controls the relative significance of stability versus spatial distribution of the selected features. Specifically, in one extreme, setting $\alpha = 0$ will select the most stable features, and in the other extreme setting $\alpha = 1$ will select dispersed features from all across the image plane without regarding their stability.

5.4.3 Incorporating distance lower-bound constraints

In this section we show that the canonical set framework is flexible enough for incorporating distance lower bound constraints among the members of canonical set. Specifically, in addition to requirements I, II, and III, we would like to make sure that for a given radius parameter r , if $p_i \in \mathcal{P}^*$ then $p_j \notin \mathcal{P}^*$ if the distance between p_i and p_j is less than or equal to r . This set of constraints will prevent the selection of canonical features that are too close to each other. The structure of *forbidden* pairs with respect to \mathcal{P} will be encoded in the form of a 0,1 binary matrix $\mathcal{F} = \mathcal{F}(r)$ where entry $\mathcal{F}_{ij} \in \{0, 1\}$. Let $\mathcal{F}_{ij} = 1$ if the distance between features p_i and p_j is less than or equal to r and zero otherwise. Intuitively, the matrix \mathcal{F} encodes the *forbidden* pairs, that is, if two features are closer than r to each other at least one of them should be outside the canonical set. This constraint can be written as: as:

$$\frac{1}{4} \sum_{i,j} \mathcal{F}_{ij} (1 + y_i y_{n+1}) (1 + y_j y_{n+1}) = 0. \quad (5.39)$$

Using the fact that \mathcal{F} is symmetric and $y_i^2 = 1$, for each $1 \leq i \leq n+1$, the constraint (5.39) can be restated as:

$$\begin{aligned} \frac{1}{4} \sum_{i,j} \mathcal{F}_{ij} + \frac{1}{4} \sum_{i,j} \mathcal{F}_{ij} y_i y_j + \\ \frac{1}{2} \sum_{i=1}^n y_i y_{n+1} \sum_{j=1}^n \mathcal{F}_{ij} &= 0. \end{aligned} \quad (5.40)$$

Letting $f_\Sigma = \sum_{i,j} \mathcal{F}_{ij}$, and defining \mathbf{f} as a column vector in \mathbb{R}^n whose i^{th} entry is equal to $f_i = \sum_{j=1}^n \mathcal{F}_{ij}$, we can then express this latter constraint as

$$\hat{\mathcal{F}} \bullet \mathcal{X} = 0, \text{ where } \hat{\mathcal{F}} = \begin{bmatrix} \frac{1}{4}\mathcal{F} & \frac{1}{4}\mathbf{f} \\ \frac{1}{4}\mathbf{f}^T & \frac{1}{4}f_\Sigma \end{bmatrix}. \quad (5.41)$$

Our new SDP formulation for the canonical set with forbidden relations can be stated as

CANSET2:

$$\text{Minimize} \quad \mathcal{Q} \bullet \mathcal{X} \quad (5.42)$$

$$\text{Subject to} \quad \mathcal{D}_i \bullet \mathcal{X} \geq 0, \quad \forall i = 1 \dots, m, \quad (5.43)$$

$$\mathcal{X} \succeq 0, \quad (5.44)$$

where \mathcal{Q} is as described in (5.36), $m = n+4$, and the first $n+3$ constraint matrices are as described in (5.37), and constraint matrix $n+4$ is (5.41). The approximate solution for this modified formulation can be estimated in a similar manner to the original canonical set problem of Denton *et al.* [26].

5.4.3.1 Computing the optimal radius

We would like the value of r to be as large as possible, thereby spreading the canonical set members as far apart as possible. Without loss of generality, let us assume that the distances between all p_i and p_j are normalized to fall between zero and one. To find the optimal value of r , we note that when $r = 0$ the formulation reduces to the standard canonical set formulation, i.e. the matrix \mathcal{F} is all zeros. On the other hand if $r = 1$, the matrix \mathcal{F} is all ones and the problem is infeasible for our purposes since only one vertex can be in the canonical set. We next show how we can find the optimal r in polynomial time.

Lemma 5.4.1. *Let r be some value such that the CANSET2 algorithm is feasible. Then any value $r - \delta$ where $\delta > 0$ is feasible.*

Proof. Any edge that is forbidden by r is also forbidden by $r - \delta$. This is clearly the case since, if the distance between two set members, p_i and p_j is less than $r - \delta$ it is also less than r . Thus a problem using $r - \delta$ is less constrained than a problem using r . \square

Lemma 5.4.2. *There are at most $n^2/2$ values for r that give distinct matrices \mathcal{F} , where $n = |\mathcal{P}|$, the cardinality of \mathcal{P} .*

Proof. Since matrix \mathcal{F} is symmetric and $\mathcal{F}_{ij} = 1$ if the distance between features p_i and p_j is less than or equal to r and zero otherwise, the number of distinct \mathcal{F} matrices is bounded by the number of distinct distances, which can be no more than $n^2/2$. \square

Lemma 5.4.3. *The number of ones in the matrix \mathcal{F} increases monotonically as the value of r increases.*

Proof. Let d_{ij} be the distance between p_i and p_j . Clearly, if $d_{ij} \leq r$ then $d_{ij} \leq r + \delta$, where $\delta > 0$. \square

Theorem 5.4.4. *The optimal value for r can be found by running the CANSET2 algorithm at most $O(\log n)$ times.*

Proof. Let r^* be the optimal value for r . By lemma 5.4.2 there are at most $n^2/2$ possible values for r that must be searched, by lemma 5.4.1 and lemma 5.4.3 we know for any particular r whether $r^* < r$ or $r^* > r$. We can thus perform a binary search to find r^* , running the CANSET2 algorithm at most $O(\log n)$ times. \square

5.4.3.2 Algorithm complexity

Our algorithm for computing the approximate canonical subset with forbidden edges, which we denote CANSET3 thus consists of running the CANSET2 algorithm at most $O(\log n)$ times, performing a binary search to find the optimal radius r . Since SDP algorithms such as CANSET2 run in polynomial time, it follows that CANSET3 also runs in polynomial time.

5.4.4 Experiments

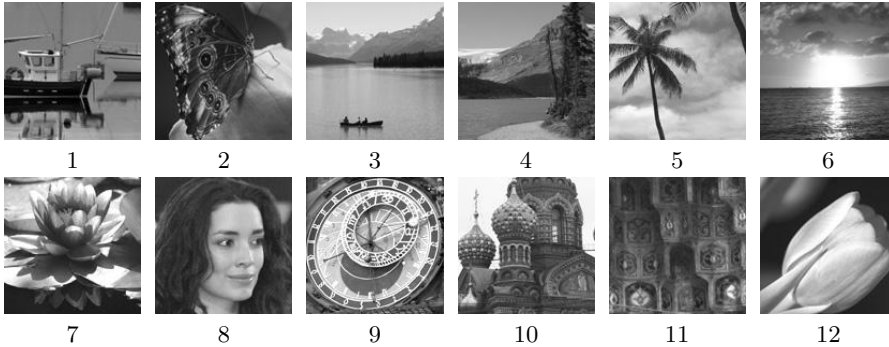


Figure 5.15: Images used for the canonical set experiments, all images are 128 by 128 pixels.

For our experiments, we used the 12 test images shown in Figure 5.15. For each image we extracted the 10 types of scale space interest points described in Section 5.2 and combined them into one file ordered by TV-norm. The total number of interest points extracted ranged from 7160 for the tulip image (image number 12 in Figure 5.15) to over 10438 for the clock image (image number 9 in Figure 5.15).

From each of the combined point files we extracted the top 200 points ordered by TV-norm, created reconstructions, and measured the MSDE rate. The large number of interest points precluded running the CANSET3 algorithm on the full sets of combined

features, so we reduced the size of the combined feature sets by discarding points within a radius of 3 pixels of one with a higher TV-norm. The size of the resulting filtered combined sets was in the order of 1000 interest points. We ran the CANSET3 algorithm on each of the reduced combined interest point sets, varying the convexity parameter, α , between 0 and 1 in 0.1 increments and selected subsets of 200 points. We created reconstructions from each of the subsets and measured the resultant error. Figure 5.16 shows the MSDE rates obtained from reconstructions based on the TV-norm subsets and the canonical subsets.

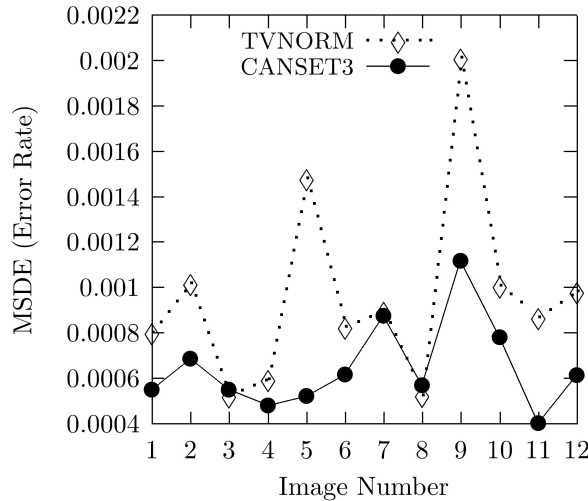


Figure 5.16: Error rates for top 200 TV-norm versus best canonical set.

Examination of the graph shows that subsets selected based on the canonical set algorithm have lower error rates for all but two of the example images. Figure 5.17 shows a detail for image 3, where the reconstruction from the canonical set had higher error. The top row shows an area of the clouds where the canonical subset provided greater detail than the TV-norm subset. The bottom row shows the boat area, for which the canonical set gives a poorer reconstruction.

Figure 5.18 shows the reconstruction results for the images in Figure 5.15. The left column in each cell shows the original image, the second column shows the reconstructions from the top 200 points selected by TV-norm, the right column shows reconstructions from 200 points selected using the canonical set algorithm. The reconstructions from the canonical sets of image points have lower error for all but two of the images. Further examination of the reconstructions shows reconstructions using the canonical sets avoid the large empty regions (areas with no selected features) present in the reconstructions using the ordered TV-norm.

We next inspected the error rates for reconstructions using the top 200 points from the filtered combined interest point sets. Recall that the reduced sets are generated by discarding points within a radius of 3 pixels of one with a higher TV-norm. Figure 5.19 shows the result of this comparison. The canonical set is better in 8 out of the 12

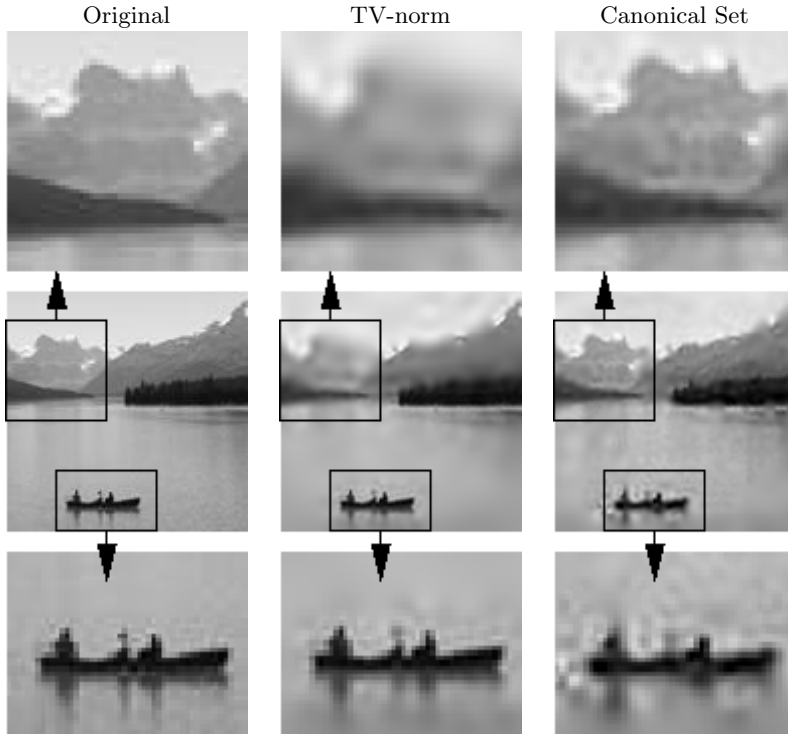


Figure 5.17: The left column shows the original image, the middle column shows the reconstruction using the TV-norm, and the right column shows a reconstruction using the canonical set. Even though the canonical set gives a better reconstruction of the clouds, the boat has reconstruction artifacts, resulting in a higher MSDE.

reconstructions and only slightly worse in 3 instances in terms of error rates. One reason for this behavior (being only slightly worse) is that for $\alpha = 0$, the canonical set algorithm is only considering the stability (TV-norm) of the points in the objective and the spreading of the points is enforced through the constraint (5.41).

Figure 5.20 shows reconstructions of test image 9 from canonical sets computed using various values for α . The MSDE error rate for $\alpha = 0.1$ is higher than the MSDE error rate for the TV-norm reconstruction and all of the others are lower, with $\alpha = 0.2$ giving the best result.

Finally, we examined the effect of the number of selected features on the quality of reconstruction. Our preliminary investigations indicate a progressive improvement on the quality of reconstruction as a function of subset size. Figure 5.21 shows the results of this investigation for image 9 from our test set. Figure 5.22 shows reconstruction results for various subset sizes.



Figure 5.18: Image reconstructions, the left columns show the original images, the middle columns show reconstructions with the top 200 scale space points ordered by TV-norm, the right columns show reconstructions with the canonical set of scale space points computed by the CANSET3 algorithm.

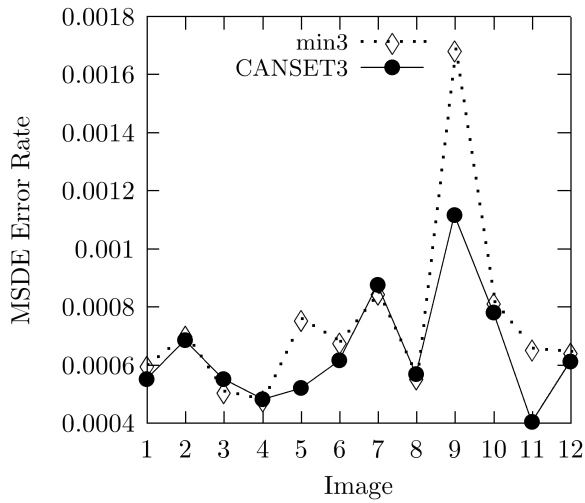


Figure 5.19: Error for top 200 min3 filtered versus best canonical set.

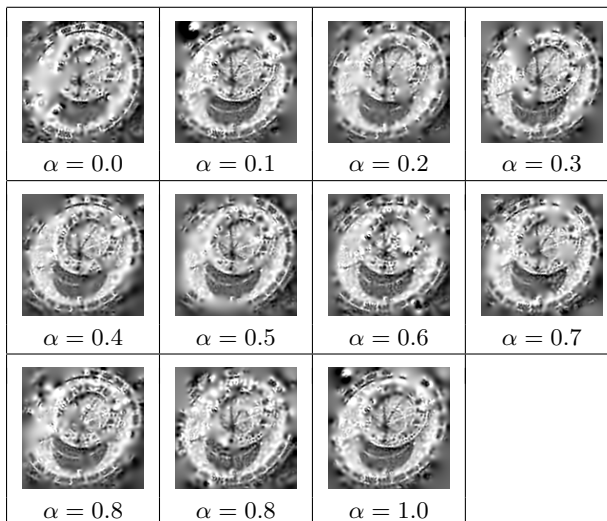


Figure 5.20: Reconstructions from subsets computed with different values for α parameter.

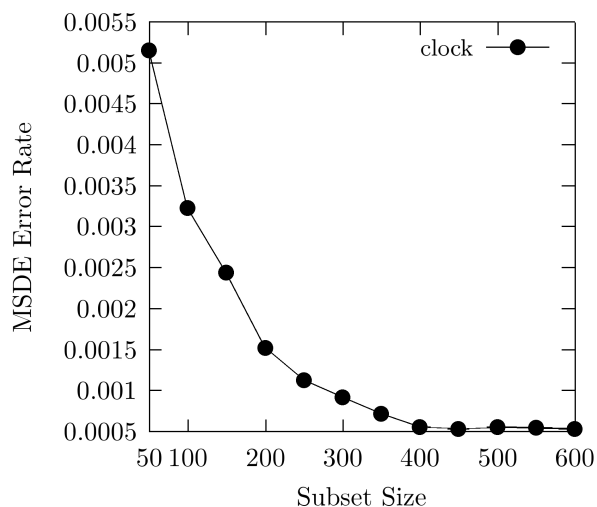


Figure 5.21: The effect of canonical subset size on the quality of reconstruction for image 9 from Figure 5.15. The reconstruction error decreases as the size of the subset increases.

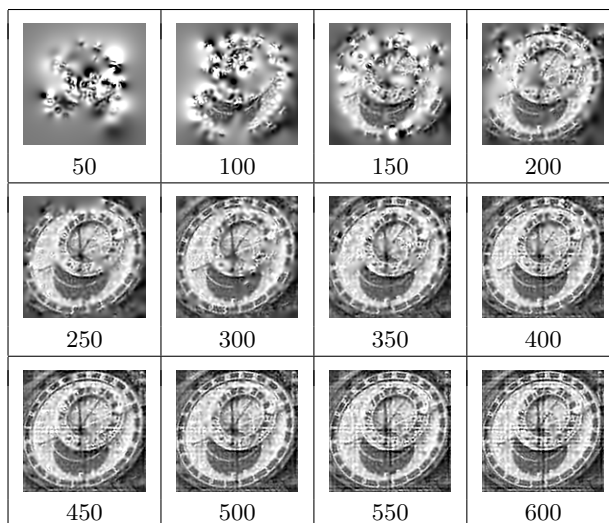


Figure 5.22: Reconstruction based canonical sets of increasing size. The size of canonical sets for each reconstruction is indicated in the cells. The line artifacts are due to numerical errors in the reconstruction algorithm.

5.5 Conclusion and discussion

In this chapter 10 commonly used scale space interest points are compared for the purpose of image reconstruction: Corner points, edge points, Harris Laplace points, Hessian blobs, Hessian Laplace points, Laplacian blobs, ridge points, scale space saddles, top points and top points of the Laplacian. These scale space interest points are calculated for all images in two test sets. Test set one consists of 8.000 patches of the van Hateren database of natural images and the second set consists of 12 images taken with a digital camera and later sub-sampled. For all these point sets, reconstructions are made using the Sobolev reconstruction scheme presented in chapter 3. These reconstructions are evaluated using several image quality measures presented in chapter 4. The results show that certain types of interest points perform better for image reconstruction than others. More specifically, in general blobs, ridge points and corner points perform better than the other scale space interest points for reconstruction. Harris Laplace points, often used for image matching, are not very suitable for image reconstruction. One reason is that the number of Harris Laplace points is in general very low. However, adding features in this low number of points might improve the reconstruction quality, while maintaining a comparable number of constraints as the reconstructions from blobs for example. Top points of the Laplacian perform relatively bad in the first set, while in the second set they perform very well, even among the best. A possible cause can be the fact that the first set contains many patches with very little structure, while in the second set every image contains much structure. Also the ordering of the top points of the Laplacian might be a cause, since for the first set only a quarter of the points is used in general, while for the second set almost half the number of points is used.

Another method presented in this chapter to compare the descriptive power of scale space interest points is to compare the stability of the reconstruction with respect to the localization error of the scale space interest points. If a point set contains much local image information, randomizing the position of the points should degrade the reconstruction quality. Results from experiments with varying noise on the position of the scale space interest points show again that some types of points are more suitable for image reconstruction than others. More specifically, reconstructions from corner points, blobs, ridge points and top points show an increasing error for increasing standard deviation of the noise. In contrast, for edge points and Hessian Laplace points the error *decreases* with increasing standard deviation of the noise. For edge points this can be explained since many edge points close together contain much redundant information. Randomizing these points removes some of that redundancy and improves the reconstruction quality.

It is shown in literature that a combination of different types of interest points can improve the reconstruction quality substantially. It is however not clear *how* exactly these points should be combined. The conventional feature selection method for image reconstruction is driven by stability or strength measures associated with scale space interest points. Experimental studies show that ordering interest points by TV-norm and selecting the strongest points will result in good reconstructions for some single type scale space interest points, but will not necessarily result in good reconstructions for *combined* types of scale space interest points. In this chapter we proposed a framework for feature selection that accounts for both stability of scale space interest points as well as their spatial distribution. Our method is a generalization of canonical feature selection algorithm. We presented a new formulation for the canonical set problem that

incorporates constraints on the minimum distance between selected features. We also presented the feasibility analysis for optimal solution of the parametric form of spatial canonical sets. Finally, the complexity analysis for approximating the canonical set for image reconstruction in polynomial time was discussed. Through a set of experiments, we compared the reconstructions based on selected features using our new method with those ordered by TV-norm. For our test images we considered 10 types of scale space interest points. The preliminary experiments show the quality of the reconstructions based on the subsets of image features selected by the canonical set algorithm produced lower error as measured by the Multi Scale Differential Error (MSDE) in 10 of 12 reconstructions. In 8 of the 12 case studies the error was significantly lower, and in the other cases, they were comparable. In all cases the subsets selected using our algorithm avoided the large empty areas in images (areas with no selected features) that were present using the subsets selected by the ordered TV-norm.

The best type of scale space interest points will depend on the image itself. It is shown in this chapter that the top points of the Laplacian for certain images are very suitable, but for a different set of images are not. Some other types of scale space interest points are less sensitive to the actual image content and are suitable for a broad class of images. Future research should include more experiments evaluating the stability of different types of scale space interest points for varying types of images. With respect to combining different types of scale space interest points it is shown that besides the strength, also the spatial distribution is important. The similarity measure used is currently solely based on the Euclidean distance in 2D. The scale of the interest points should also be included in future research. For future work we also intend to consider the performance of feature selection on a plane by plane basis, i.e. selecting a subset of each feature type and then combining them into a unified set.



Object-based Image Editing using Reconstruction from Scale Space Interest Points

This chapter is based on:

Image Editing using Scale Space Interest Points.

F.M.W. Kanters, L.M.J. Florack, B. Platel and B.M. ter Haar Romeny.

In preparation.

6.1 Introduction

Since the introduction of digital images, image editing with a computer is used to manipulate the contents of an image. In the eighties the first commercial programs to manipulate digital images became available and articles about this subject appeared in several journals [79, 9]. Nowadays editing digital images such as pictures from a digital camera has become a daily practice for many professional photographers and even consumers often edit the pictures from their digital cameras. Also in other fields such as special effects in movies or cosmetic surgery, digital image editing has become popular. The early programs to edit digital images manipulated the pixels of the digital images directly. Many of today's commonly used image editing packages such as Adobe PhotoShopTM are still based on the direct manipulation of pixels or groups of pixels in the image. However, for some operations an object-based approach to edit images is preferred. For example to automatically replace apples in a scene with pears it is necessary for the computer program to have a more abstract representation than the raw pixel data alone. Object-based image editing operations however have traditionally been limited to well defined graphical objects such as lines, rectangles and circles. Examples of object-based image editing software with simple geometrical objects are Adobe IllustratorTM and Corel DrawTM.

In the last decade, several approaches to object-based image editing for complex scenes have been presented. Some properties of good object-based image editing operations are proposed by Mortensen [32]. However, no indication is given *how* to implement these properties. Some approaches use a different representation of the image to perform the edits in, such as contours [66, 113], verge points [137] or a skeleton plus residual representation [89] and perform a reconstruction of the edited representation to obtain the edited image. To improve the visual quality of the edits performed with an image editing operation, adaptation of an area to its background is important. Methods like Poisson image editing [117] and covariant derivatives [50] can improve the image quality drastically by adapting the edited region to its surrounding background. Zeng et al. [149] propose a variational image model based on image derivatives that has similar properties. A content-aware tool is presented by Diakopoulos et al. [112] which uses a simple database of textures to seamlessly compose image edits. These methods however include no information on the object level, only on some local properties of neighboring pixels. Kar-Han Tan [128] presents an interactive tool to select objects in an image using simple freehand sketches. This however needs some basic human interaction to obtain the object in an image and no editing is performed. Wu et al. [143] present a fully automatic object based editing algorithm to remove eyeglasses from face images, but this is very application specific. Huang et al. [60] present a method to edit images using a recovery of the 3D shape from objects in a scene. However, some knowledge on the scene content is necessary for their algorithm. Barret and Cheney [136] introduce object-based image editing by selecting objects with some simple clicks and performing an automatic segmentation of the object using watersheds. After this step, using anchor points, interactive editing operations can be performed while corrections for geometry changes and background filling are automatically performed in real-time. While this method shows very good results, it is still impossible to automatically replace apples with pears in a scene since no object recognition is included in their framework.

In this chapter we propose a framework to perform object-based image editing based on scale space interest points. It is shown that some scale space interest points are

suitable for object detection and recognition [111, 120, 119]. In section 6.2 a proof of concept is given to show that scale space interest points can also be used for image editing using the image reconstruction algorithms proposed in chapter 3. Since this representation is suitable to automatically detect objects in a scene (given some reference object) and to replace one object with another in a visually coherent way, the objective to automatically replace apples with pears in a scene is within reach. In section 6.3 a look into the future is given about the possibilities of the proposed image editing framework. Finally, conclusions are given in section 6.4.

6.2 Proof of concept

In this section a proof of concept is given for image editing using scale space interest points. Consider the test image shown in figure 6.1. The image shows a sailing boat with a complex background. From this image, all Hessian blobs, Laplacian blobs, corner points and ridge points are calculated. These points are combined in a single set using a minimum distance constraint, ordered by differential TV-norm (see section 5.4 for more details). The 395 remaining points in the combined interest point set are shown in figure 6.1. Using this combined point set and the Brownian reconstruction from chapter 3 we try to manipulate the objects in the image.

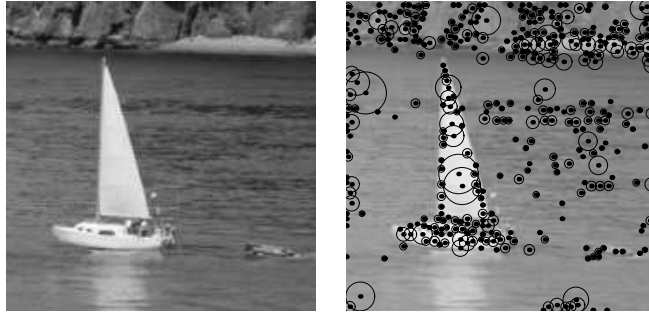


Figure 6.1: Original test image *sailboat.jpg* for our proof of concept (left). The original image is 180×180 pixels. A combined set of 395 scale space interest points of the test image is shown on the right side, projected on the original image (the diameter of the circles represent the scale of the interest points). Using reconstruction from manipulated scale space interest points the objects in the image can be manipulated.

As a first test we will try to remove the sailboat from the image. In order to do this, all interest points belonging to the sailboat are removed. This is done by applying a mask, which is shown in figure 6.2. All points within the white area of the mask are removed from the combined point set. Note that no scale information is taken into account for the masking; only spatial information is used. The mask is created manually for this experiment, but in a practical application one should create a mask automatically. Using an image matching algorithm based on scale space interest points [120, 119], one can identify which interest points of a complex scene belong to a given reference object image provided one has a database object similar to the reference

object. Specifically, given our image of a sailboat in a scene it will be possible to extract the interest points corresponding to the sailboat from the image. A mask can be created using the outline of the filters used for reconstruction, but in this case having the interest points is sufficient. The reconstruction from the *sailboat.jpg* image without the interest points belonging to the sailboat itself is shown in figure 6.2. Note that indeed the sailboat is removed and that the background is filled in by the prior of the reconstruction algorithm. However, there is no texture of the water in the area where once our sailboat was present.



Figure 6.2: Removing the sailboat from *sailboat.jpg*. The top row, from left to right shows the original image, the 395 combined scale space interest points projected on the original image and the mask used to filter the interest points respectively. The bottom left image shows the remaining 307 interest points after applying the mask and the bottom right image shows a reconstruction from the remaining interest points.

In order to make a more realistic image, the water in the area where the sailboat was present should be textured. In the presented framework this can be done by copying information from the background water. First a number of points (58 in total for a good distribution of the points) is placed in the area where the sailboat was present using the mask created previously. A randomly varying scale is used to reduce the chance of artifacts being visible. Next, derivatives of the points in the right side of the image (the textured water part) are used for the new points as constraints for the reconstruction algorithm. Figure 6.3 shows the results. Note that the water is now textured where once the sailboat was present and therefore gives a more realistic image

with the sailboat removed from the scene.

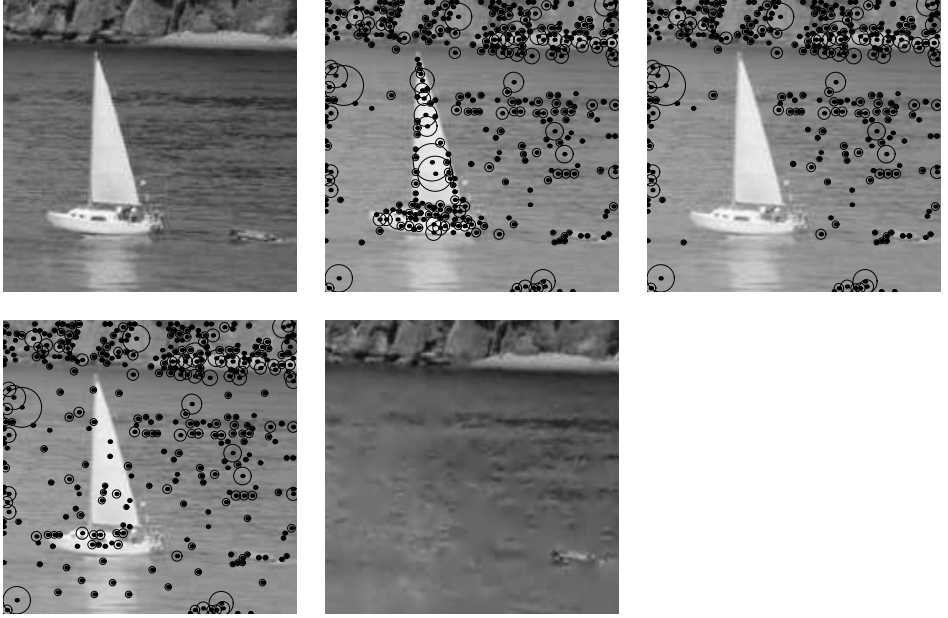


Figure 6.3: Texturing the water. The top left image shows the original test image. The top center image shows all 395 interest points taken from the image. The top right image shows the 307 interest points of the scene with the interest points of the boat removed. The bottom left image shows the 365 interest points with the sailboat interest points removed and the additional texture points added, projected on the original image. The bottom right image is a reconstruction from these manipulated interest points. The sailboat is removed from the scene and the hole is filled with water texture.

It is also possible to insert a new object into the scene. In the following experiment, a new object will replace the sailboat in the *sailboat.jpg* image. The new object is a rubber duck, obtained from an image of the coil database [114] shown in figure 6.4. The size of the duck image is 328×328 pixels. From this image the canonical set of combined scale space interest points is calculated using the minimum distance constraint. The image should be flipped horizontally, so from all points in the canonical set, the x-coordinates are inverted. The interest points of the duck image are combined with the interest points of the background image with the sailboat removed. Points of the background image that are within a radius of 4 pixels of points from the duck are removed. A reconstruction is shown in figure 6.4. To obtain sharp edges the derivatives that are used as constraints for the reconstruction are recalculated. At the boundary information of the background image and the object is combined using a reconstruction from the object points and a mask created from interest points of the object. Within the mask, the reconstruction of the object is used to re-calculate the features, outside the mask the background image is used (which is in this case also a reconstruction).

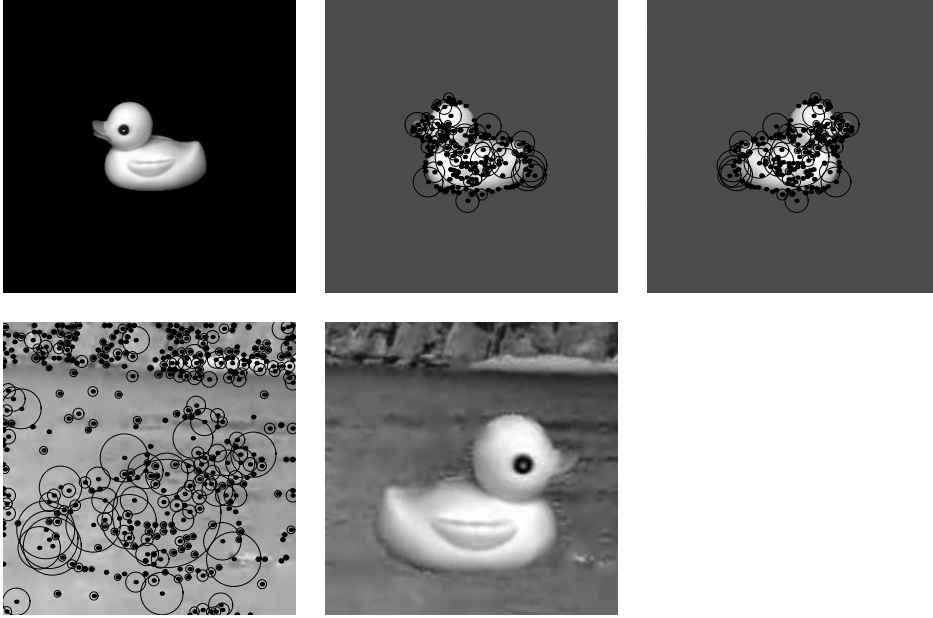


Figure 6.4: Replacing the sailboat with a rubber duck. The top left image shows the rubber duck image (328×328 pixels) and the top center image shows the 159 scale space interest points of the duck. The top right image shows the flipped duck points. The bottom left image shows the points of the background combined with the points of the rubber duck (406 interest points in total). The bottom right shows a reconstruction from these points. Note that derivatives in points at the object boundary used as constraints for the reconstruction are recalculated.

Now we try to find a new background for the sailboat that has been retrieved from the *sailboat.jpg* image. An image of the sea, shown in figure 6.5, is used for this purpose. Again scale space interest points are calculated from this image and combined with the scale space interest points of the sailboat. Points of the background within a radius of 4 pixels of points from the object are again removed and the features at the boundary of the object are recalculated as described in the previous experiment. The result is shown in figure 6.5. Since there is no sampling done in our representation, scaling of objects is simple. In figure 6.6 a second sailboat is added to the image, while the first one is translated. The second sailboat is scaled with a factor 2 by multiplying the x- and y-coordinates of all interest points of the object with a factor 2 and the scale s of the interest points with a factor 4 ($\sigma = \sqrt{2s}$). The result is shown in figure 6.6. Note that the second sailboat is scaled without the need of interpolation, since the reconstruction algorithm takes care of sampling. Due to the increased scale of the interest points of the second sailboat, this part of the image is more blurred. This might be resolved by changing the derivatives of the points to sharpen edges, for example using the work by Florack [45].

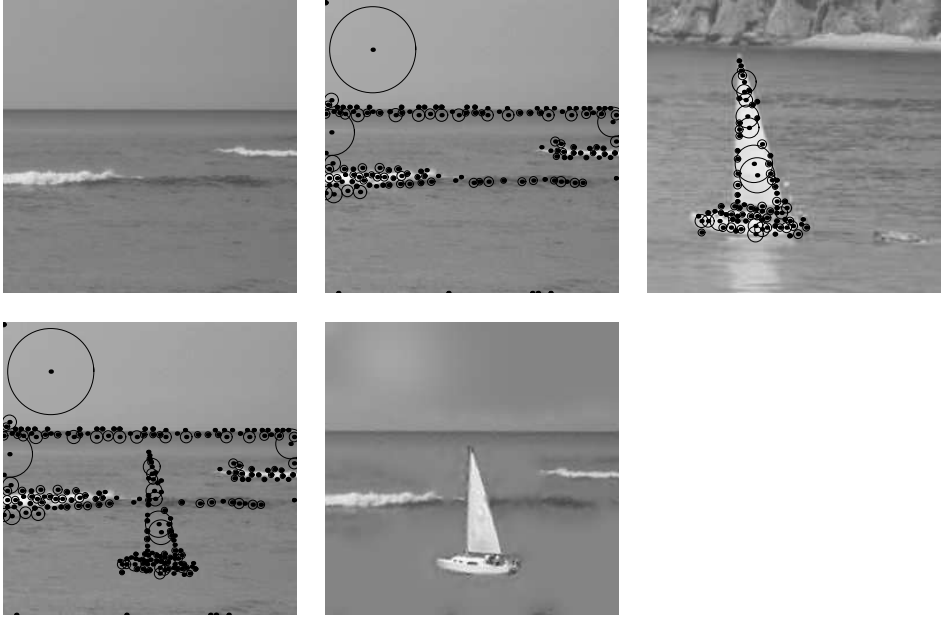


Figure 6.5: Giving the sailboat a new background. The top left image shows the new background image. The top center image shows the 150 scale space interest points of the new background image. The top right image shows the original *sailboat.jpg* image with the 88 interest points of the sailboat projected. The bottom left image shows the combined interest points (235 in total) and the bottom right image shows a reconstruction from these points.

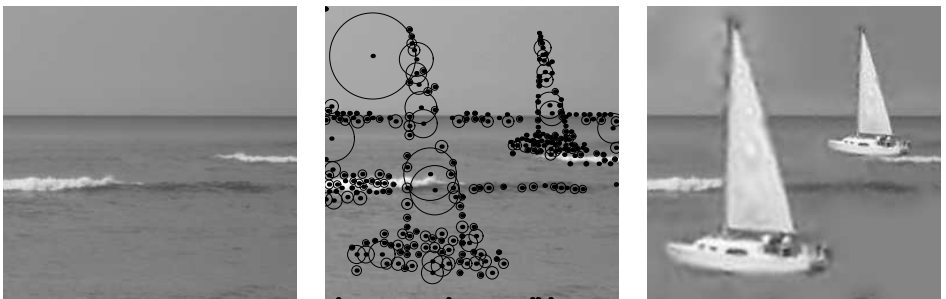


Figure 6.6: Adding a larger sailboat. The left image shows the background image, the center image shows the 281 combined scale space interest points of the background, the translated small boat and the scaled large boat. The right image shows a reconstruction from these points. Note that no interpolation was needed, since the reconstruction algorithm performs the sampling afterwards. Due to the increased scale of the points of the large sailboat, this part is more blurred.

Since there is no need to worry about interpolation, it is also possible to use smaller objects in the scene. For example, if we would like to add the arbitrarily chosen logo shown in figure 6.7 in the sail of the large sailboat this can be done by manipulating the scale space interest points. All x - and y -coordinates of the logo are divided by 4 and the scale s is divided by 16. Again a reconstruction is made from the combined interest points as described in previous experiments. Depending on the sampling of the reconstruction function, all detail of the logo will still be present. For our purpose a relative low resolution is used to sample the final reconstruction, hence some detail is lost. Note that still much of the logo is visible, which would not be the case if simple sub-sampling of the object had been used.

The trick of replacing the sailboat with a rubber duck can again be performed in this new background. In figure 6.8 we replaced the interest points of the large sailboat with the interest points of the duck image and performed a reconstruction. The company logo can also be placed into the image as is shown in figure 6.9. The gray-level values of the logo are changed by changing the intensity value and derivatives in each interest point of the logo image. In this way, objects can be blended in a scene in a simple way. A full alpha blending of background and object is also possible by making separate reconstructions first and then recalculate combined features from the separate reconstructions.

6.3 A look into the future

In the previous section a proof of concept is presented which shows that image editing using scale space interest points is feasible. The reconstruction quality and methods to combine scale space interest points are however not yet optimal and much research has to be performed before real-life applications using this technique can be expected. In this section a short overview will be given of what might be possible once the reconstruction quality, speed and combination methods have improved to a higher performance/quality level.

6.3.1 Digital photo editing

Imagine a photo editor, connected to a large database (for example over the internet) that contains objects in a scale space interest point representation. The user loads an image, taken with his or her digital camera. Using a fast scale space interest point matching algorithm, the objects in the scene are identified. If there are new objects, the user is asked if the object should be included in the database locally or on the internet. Since the interest points of the objects are matched with the interest points of the scene, masks of the objects can be made with a simple click. If one would like to change the position, size or orientation of an object, this can be simply done by dragging the object to its new place or use the size and orientation handles. A fast interactive reconstruction algorithm could immediately update the image. If the objects in the background are recognized in the detection step, the information on the interest points from the database could be used to fill in the holes in the image left by the moved object. If the object is not known, for example if it is a picture of the sea or a desert, statistics on the database can give the best point set to fill in the hole, without knowing exactly what was behind the object. Replacing the object with another object is also easy, one can just click the desired object in the database or a second image from his or



Figure 6.7: Putting advertisement on the sailboat. The top left image shows the logo used to place on the sail. The top center image shows the 122 scale space interest points of the logo and the top right image shows the 281 points of the two sailboats and the background image. The bottom left shows the 403 combined interest points and the bottom right image shows the reconstruction from these points. Note that no interpolation is necessary and that the sampling of the reconstruction function determines the amount of detail in the final result.

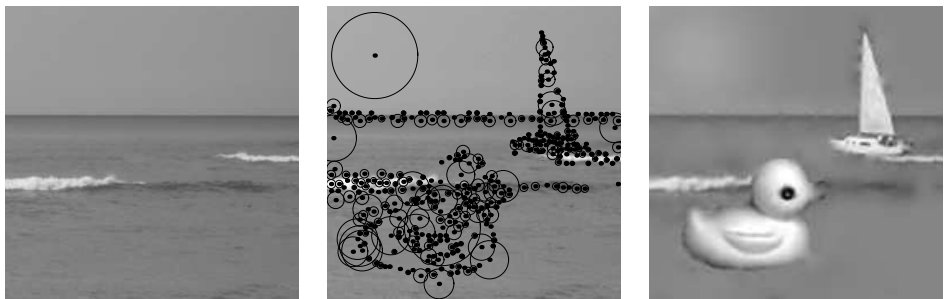


Figure 6.8: Replacing a sailboat with a duck again. The left image shows the original background image, the center image shows the 366 combined scale space interest points with the large sailboat points replaced with the duck points and the right image shows the reconstruction from these points.

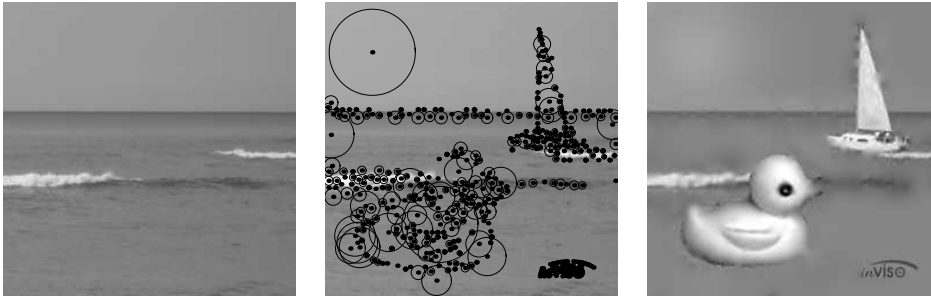


Figure 6.9: Adding a watermark to the scene. The left image shows the original background image, the center image shows the 488 combined scale space interest points with the watermark logo where the features have been adapted to get a gray-scale blending and the right image shows the reconstruction from these points.

her digital camera and the corresponding interest points can automatically be replaced with the new ones. Resizing the image is actually just re-sampling the reconstruction function. Also automatic categorization of your digital pictures is easier, since it can be automatically connected with the objects that are present in the scene.

6.3.2 Photo synthesis

Given a large database of objects in the compact scale space interest point representation, synthesizing images of complex scenes is an easy task. One can just drag and drop objects from the database, scale, rotate and translate as desired and decide which resolution the end result will be rendered in. Since the representation can be very compact (especially compared to high resolution images) and resizing without losing information is possible, this method is more powerful than just using the sampled version of the images. It is also possible to combine objects with classic sampled images by calculating the interest point representations for the classic images. Synthesizing simple objects is also possible by defining some important scale space interest points such as corners, edges, ridges and blobs and the local derivatives in those points to tune for example edge strength. Photo-realistic scenes can be composed in a simple fashion, without worrying about resolution. This method can replace the stock photos that are often used for creating advertisements.

6.3.3 Company advertisement

There are more possibilities useful for company advertisements. If a company changes its logo or style, automatic replacement would be very useful. Imagine a soccer game with advertisement boards and shirt sponsoring. If a company wants to show another logo, adapted for example to some specific topic (such as google does sometimes) this can be automatically done by searching and tracking the scale space interest points of the old logo and replace them with the new ones. Of course, only parts of the image with the logos have to be reconstructed, which can be done in real-time. Also adapting the logo for viewers in specific countries is possible. The same principle could also

be implemented in copiers and fax machines. Imagine a copier or fax machine that automatically detects low resolution or old company logos and automatically replaces them with the latest, high resolution version. Or a fax machine that detects all objects to reduce bandwidth and still can create high resolution output.

6.3.4 Animation and Morphing

Another application can be found in the animation and film industry. An object in the scale space interest point representation is disconnected from a grid, and has several key points that can be manipulated as a group which is described previously, but also separately. It is possible to build a tree or graph from the interest point structure and use this to manipulate parts of an object. For example a person has arms, legs, a body and a head. An arm has a hand with fingers attached, etc. All parts have their own scale space interest points and therefore can be manipulated separately or as a group. This is ideal for animation of objects with a photo-realistic nature. Such an application is also useful for special effects in a movie, making it possible to edit photo-realistic objects in a way, that normally can only be done with synthetic objects. In both applications it is possible to put constraints on the movement of (sub) parts of the objects. This can for example prevent a hand to move into the body or to constrain the angle of rotation for a group of points. Also morphing one object into another object can be done with a scale space interest point representation. Specific paths can be given for each interest point from one object into the other or the shortest path using some distance measure can be automatically determined (using complex constraints of movements of sub-parts). A smooth morphing with constraints specific for each object can be enforced with this method.

6.3.5 Medical applications

A very challenging field for image analysis is medical imaging. Objects in medical images usually do not have a nice, clear boundary in contrast with previously described applications. Therefore it will be much harder to describe objects in these images with scale space interest points. However, if one succeeds to find good object descriptors (which might possibly not be points, but other descriptors such as lines, regions with certain texture or a combination of different types of descriptors) these might help with many medical applications. For example on lung CT images, lung nodules can be hard to see due to the presence of the ribs. If the ribs in the image can be described with scale space descriptors, they can be attenuated using some reconstruction algorithm. This is inspired by a noted instruction of a radiology professor to his trainees: "Think away the ribs". In such a case, the lung nodules can become better visible in the image, which helps (automatic) detection. Another example where scale space descriptors could be useful is the detection of the heart state or deviations in the motion of the heart, by comparing the scale space descriptors with statistical data of many other images. Statistics on a small number of meaningful descriptors is much easier than statistics on the image data itself. For plastic surgery, the scale space interest point representation might help building realistic models predicting the outcome of an operation. Also 3D images can be used creating 4D scale spaces. Reconstructions will be more computationally intensive, but the algorithm will not become more complex.

6.4 Conclusion and discussion

Many of today's image editing software is still based on direct manipulation of the pixel data of the digital images. However, for some more complex operations such as replacing all apples from a scene with pears, this pixel representation is not suitable. There have been recent advances in object based image editing, but they either need manual interaction to select an object (boundary) or are very task specific. The presented scale space interest point representation has already been successfully applied to image matching problems, identifying known objects in a complex scene. In this chapter we showed that it is feasible to use a scale space interest point representation for editing the contents of an image. Despite the fact that the presented results are far from optimal, a combination of the matching and editing possibilities of the scale space interest point representation looks promising. A number of possible applications is given in the fields of photo editing, photo synthesis, company advertisement, animation, morphing and medical applications.

However, the current quality of the reconstruction algorithm and the edit operations of the interest points is not yet sufficient for these applications. A question that may rise is why not use standard methods to segment the image into objects and perform standard pixel based methods? In principle any image editing operation performed on a sampled image resulting in another sampled image can be done directly in the pixel domain. But despite the fact that these pixel based methods still result in higher quality images, there are some advantages of the presented method to perform object based image editing. Due to the compact representation it is easier to do statistics on objects, which may help filling in regions that are unknown. Also occlusion of objects is easier to handle with the scale space interest point representation than with a pixel based scheme. This can help both in the detection of objects as well as for the edit operations. An extension to 3D images or time series is straightforward. Building a database of objects, independent of resolution is not possible for the pixel based methods. Also scaling and rotating objects is easier, since in the pixel based methods a good interpolation scheme (for example spline interpolation) is necessary. The representation also provides points that can be used for controlled motion or morphing effects that are hard to perform in the pixel domain.

Before these benefits really can be exploited in the proposed applications, the reconstruction quality and the quality of the edits should be further improved. The presented proof of concept used blobs, corner points and ridge points for editing the image. The object recognition is however performed with different points, such as Harris Laplace points and top points of the Laplacian. It should be further investigated if the interest points used for editing are also suitable for matching. The reconstruction algorithm is currently being improved by including boundary conditions and a way to have a prescribed boundary as a constraint for the reconstruction [7]. This makes it possible to extract a region from an image, perform some edits, make a reconstruction and finally a seamless integration back into the image can be performed. The reconstruction will also be improved by including different types of constraints and filters. For example the gray-scale flow of a certain area or line features opposed to single points. The masking presented in the proof of concept is relatively poor. Better masking methods (for example the one described by Kar-Han Tan [128]) is likely to improve the quality substantially. Also the way of combining derivatives of points near boundaries (used as constraints for the reconstruction) should be investigated more carefully. The method

used in this chapter is rather ad-hoc and in some cases causes stability problems in the reconstruction. Another important aspect that prevents the current algorithms to be used in real-life applications is the speed of the calculations and the memory consumption. This has not been a major concern during this research, but some comments can be made. The scale space interest point detection algorithm and the reconstruction algorithms can be implemented on a parallel machine in a very efficient way (mostly convolutions are used). Therefore a hardware implementation using fpga's could improve speed with a few orders of magnitude compared to our current Mathematica and C++ implementations. Also having the fast increasing capabilities of modern computers in mind, using the algorithms for real-life applications should be possible in the near future. Finally an extension to color images is necessary for many real-life applications. This can be done in a simple way, dividing the image into the basic colors red, green and blue and handling everything separately, or using more sophisticated methods to create color interest points and color features for editing and reconstruction.

7

Summary and Future Research

7.1 Summary

Since the introduction of digital images, image editing with a computer is used to manipulate the contents of an image. Nowadays editing digital images such as pictures from a digital camera has become a daily practice for many professional photographers and even consumers often edit the pictures from their digital cameras. Many of today's commonly used image editing packages such as Adobe PhotoShopTM are still based on the direct manipulation of pixels or groups of pixels in the image. However, for some operations an object-based approach to edit images is preferred. For example to automatically replace apples in a scene with pears it is necessary for the computer program to have a more abstract representation than the raw pixel data alone. In this thesis a framework is presented based on a different representation of images that is one more step towards object-based image editing. The representation used in this thesis is based on scale space theory, which is the theory of apertures, through which we and machines observe the world. For computer vision systems, the notion of aperture can be introduced as *blurring* the high resolution image with a kernel of a certain width. Scale space theory also makes it possible to look at (spatial) derivatives of the image in a mathematically well posed way. It is shown that the human front end visual system takes derivatives up to at least 4th order at various scales. It seems that the *differential structure* of an image is important to detect objects, for humans as well as for machines.

In chapter 2 the α -scale space of an image is introduced, a more general class of scale spaces of which the well known Gaussian scale space is a special case. One problem with the α -scale space is that there is no closed form expression for the α -kernel (for general α) in the spatial domain, only in the Fourier domain. However, in some cases a spatial implementation is preferred instead of a Fourier implementation. In chapter 2 we present an approximation of the α -kernel in the spatial domain that is implemented in a novel software package called ScaleSpaceViz, that is created for calculation and visualization of general scale spaces, among which α -scale spaces. The main purpose of this software package is exploration of the scale space of an image. Special points exist in a scale space of an image, so called scale space interest points which can also be visualized in ScaleSpaceViz. These points represent the image quite well and can be used for image matching and image reconstruction. In this thesis the scale space interest point representation of an image plays a key role in the journey towards object-based image editing.

In chapter 3 we compare three image reconstruction algorithms that can be used to reconstruct an image from scale space interest points. The first reconstruction algorithm is simple to implement, but does not yield visually attractive results if the number of constraints is low. The second algorithm uses a different prior that smoothes the image, which results in visually more attractive images for reconstructions with a low number of constraints. This method however uses an iterative implementation that is computationally more expensive. For the third reconstruction algorithm, we introduce a novel algorithm that is computationally less expensive as the second algorithm (comparable to the first algorithm), but results in images that are visually more attractive than the first algorithm. Since less smoothing is used, the results are still a bit behind the results from the second algorithm. Some practical notes on the implementation of all three algorithms are also given in this chapter.

In order to compare reconstruction results in a reproducible manner, 34 objective image quality measures are compared in chapter 4. This comparison shows that many

image quality measures only resemble the human notion of image quality for single type distortions (e.g. noise or blocking artifacts) and are not suitable to compare images that have as many different types of distortion as image reconstructions from different types of scale space interest points. Only some of the more complex human visual system based image quality measures are able to simulate the human notion of image quality for this type of images. In chapter 4 we propose a novel image quality measure based on scale space theory that is both simple and has a reasonable resemblance with the human notion of image quality. The presented method outperforms simple mathematical image quality measures and has similar results as more complex human visual system based methods.

In chapter 5, 10 different types of scale space interest points are compared with respect to image reconstruction. This comparison is performed on two image databases; the first set contains 8.000 random image patches from the van Hateren database of natural images and the second test set contains 12 downsized images of various subjects taken with a digital camera. For all images in both test sets, 10 types of scale space interest points are calculated and reconstructions are made from all these point sets. The results are evaluated using the best image quality measures from chapter 4. It is shown that some interest points are more suitable for image reconstruction than others. More specifically, corner points, blobs and ridge points are suitable for image reconstruction, while for example top points and scale space saddles are less suitable for image reconstruction. Most points are ordered by their strength, which is different for each type of interest point. This makes combining different types of interest points difficult. In chapter 5 we propose an optimization framework for selecting scale space interest points for image reconstruction using canonical sets of scale space features. The feature selection problems is formulated as a quadratic optimization and an approximation algorithm for its solution is presented. Experimental results show improved performance over ranking the points in the combined point set with the TV-norm, which works well for single type scale space interest points.

In chapter 6 we present a feasibility study on the useability of the scale space interest point representation for image editing. In previous work by others it is shown that scale space interest points are suitable for image matching, e.g. finding an object in a complex scene given a reference object. In this thesis we showed that scale space interest points can be used for image reconstruction and moreover we showed in chapter 6 that by manipulation of the scale space interest points the image can be manipulated. Since the scale space interest point representation is suitable for grouping structure into objects, image editing can be performed at object level rather than pixel level. A number of possible applications is given in the fields of photo editing, photo synthesis, company advertisement, animation, morphing and medical applications. Despite the fact that much more further research is necessary to realize these applications, this thesis provides one more step towards object-based image editing.

7.2 Future research

To reach the goal of true object-based image editing using the presented framework, more research needs to be performed. The scale space interest points as described in this thesis are all obtained in a Gaussian scale space. Scale space interest points obtained from α -scale spaces (for general α) should be investigated. Considering α as an

additional dimension could introduce additional interest points. More research about the reconstruction algorithms could lead to improved visual quality of image reconstructions from scale space interest points, for example improving the boundary conditions and improving the stability of the implementation. Also adding different types of constraints (e.g. lines or areas instead of points) could lead to improved image quality. Adapting the features to the points used for reconstruction (especially for mixed types of scale space interest points) could also improve image quality while reducing the total amount of information used for the reconstruction. For objective comparison of reconstructions the presented error measure has to be further improved. More experiments using human observers should be performed to evaluate the new objective error measure. When the visual quality of reconstructions from scale space interest points has reached a level that humans can hardly see the difference between original and reconstruction, the way is open to perform image editing using reconstruction from scale space interest points. A major challenge will be the definition of the edits on the point set. Scale space interest points have been proven to be useful for image matching, but a clear hierarchical grouping of the scale space interest points into objects in a scene is not yet presented. Further research about grouping of scale space interest points in a complex scene will be crucial for true object-based image editing. Once this grouping is possible, edits on the groups and separate points can be defined. Problems like seamless integration of the objects and filling in holes in the image can be solved afterwards using some of the hints presented in chapter 6. The problem of true object-based image editing will be an interesting topic of research for many years to come.



Error Measure Results

A.1 Introduction

This appendix shows the results of the image error measures presented in chapter 4. First a set of original and degraded images were presented to 33 volunteers and all were asked to order the degraded images from best (1) to worse (9). The results of this experiment are shown in figure 4.2. Appendix B shows an enlarged version of the ordered image sets for viewing purposes. All described error measures were used to calculate the errors between the degraded and original images. The graphs in the next section show the results. An example is given in figure A.1 for the Mean Squared Error (MSE). On the horizontal axis the (degraded) image number is shown. The ordering is the same as the human observer ordering. On the vertical axis the corresponding error is shown. Note that some measures present the similarity or correlation rather than the error. If an error measure resembles the human observer results well, one should expect a monotonically increasing or decreasing trend. In figure A.1 one can see that for the Russia set (which is a control set that has only one type of degradation) the MSDE shows indeed an increasing trend. However, for the other sets this is not the case. The graphs in the next section are similar and show the results for all error measures. In the caption the name of the error measure, the corresponding section in chapter 4 as well as the choice of parameters, if present, can be found.

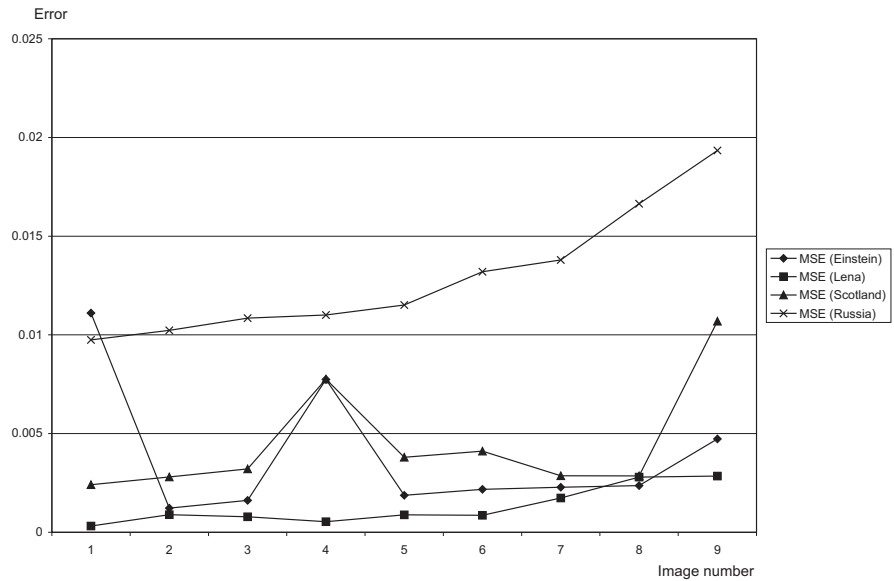


Figure A.1: Example: MSE (Section 4.2.1.1).

A.2 Image Error Results

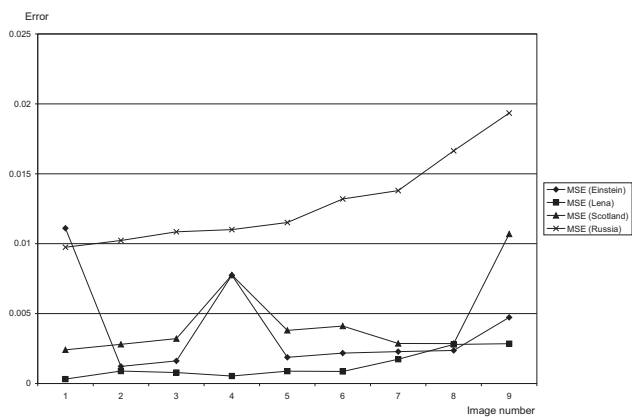


Figure A.2: MSE (Section 4.2.1.1).

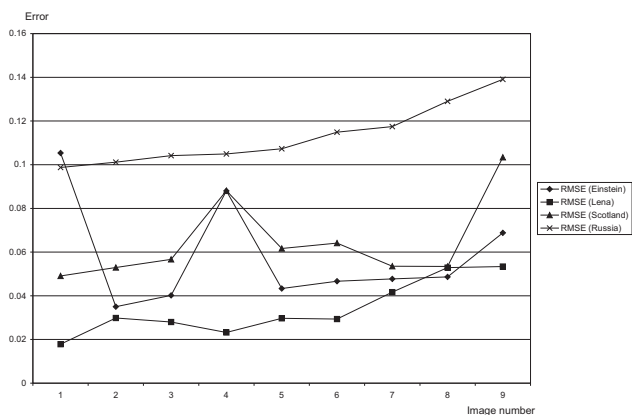


Figure A.3: RMSE (Section 4.2.1.2).

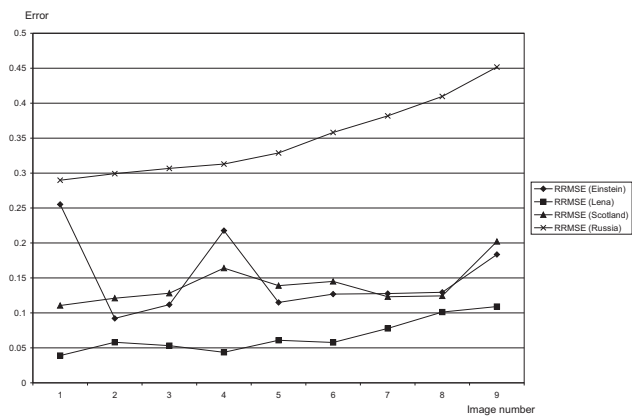


Figure A.4: RRMSE (Section 4.2.1.3).

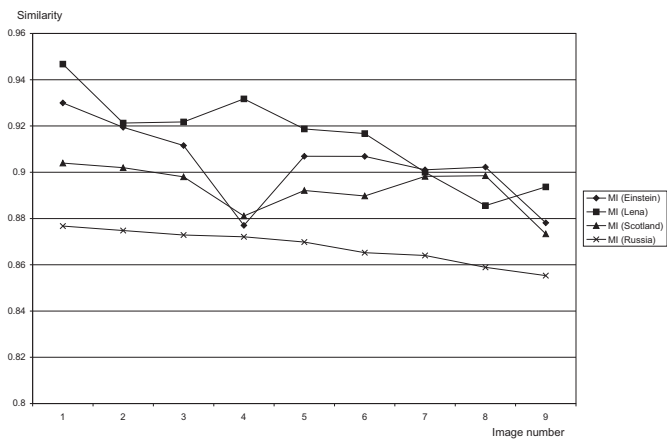


Figure A.5: MI (Section 4.2.1.4).

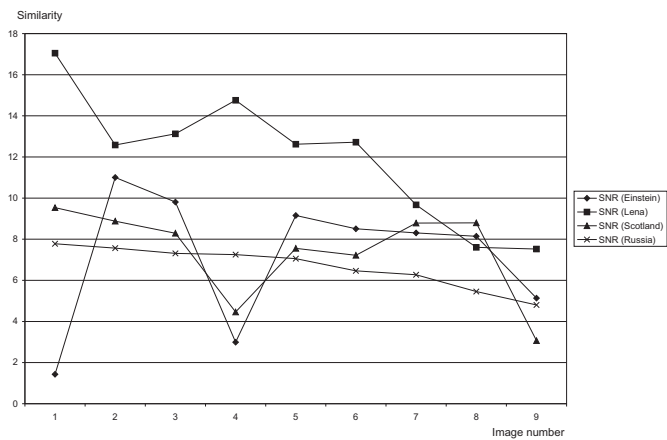


Figure A.6: SNR (Section 4.2.1.5).

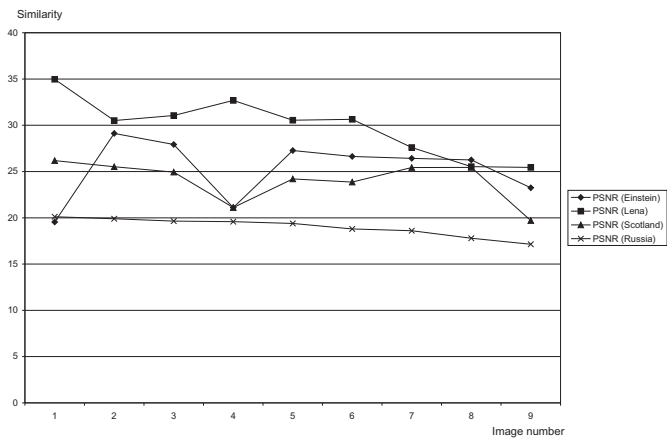


Figure A.7: PSNR (Section 4.2.1.6).

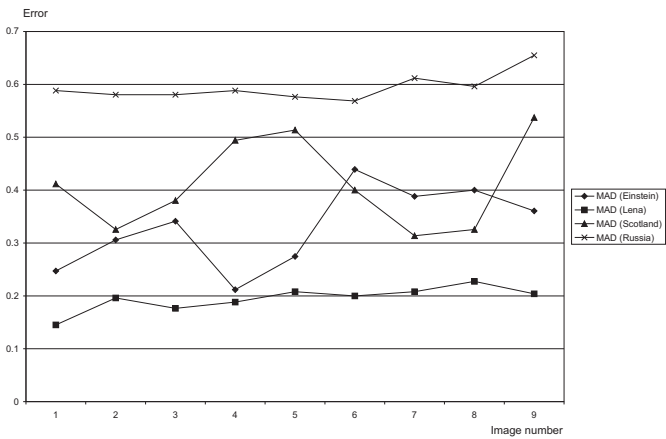


Figure A.8: MAD (Section 4.2.1.7).

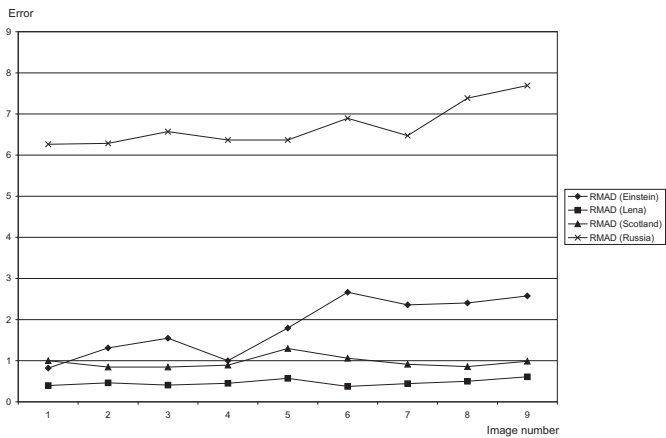


Figure A.9: RMAD (Section 4.2.1.8).

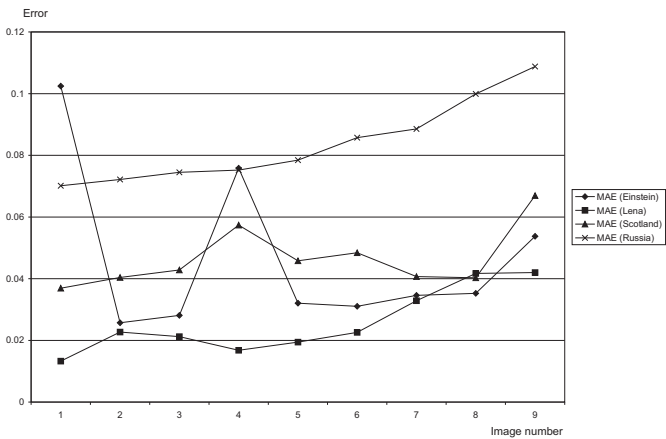


Figure A.10: MAE (Section 4.2.1.9).

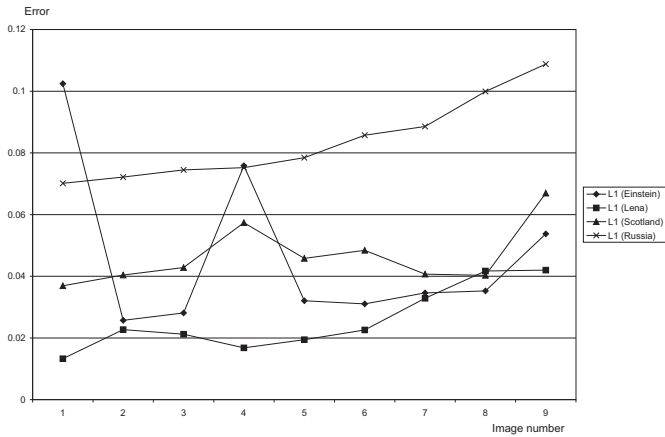


Figure A.11: \mathbb{L}_p with $p = 1$ (Section 4.2.1.10).

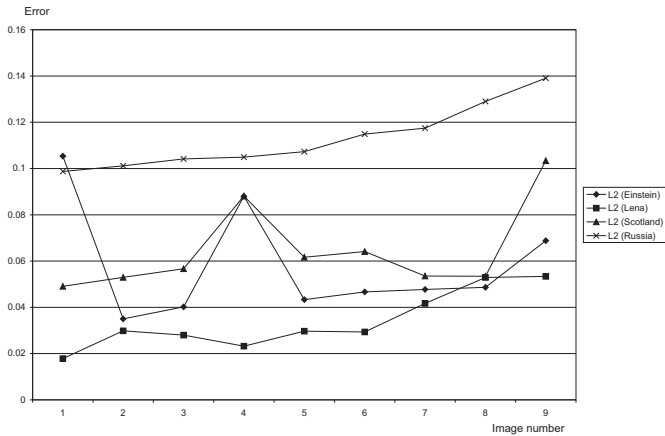


Figure A.12: \mathbb{L}_p with $p = 2$ (Section 4.2.1.10).

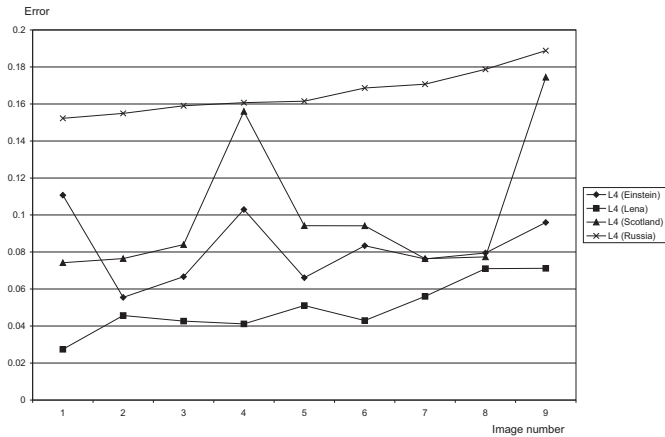
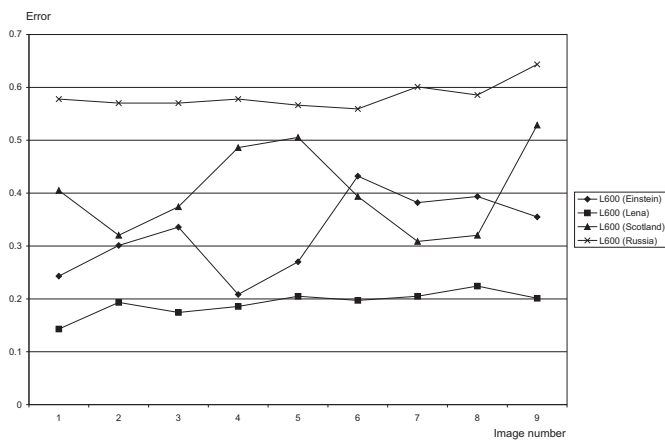
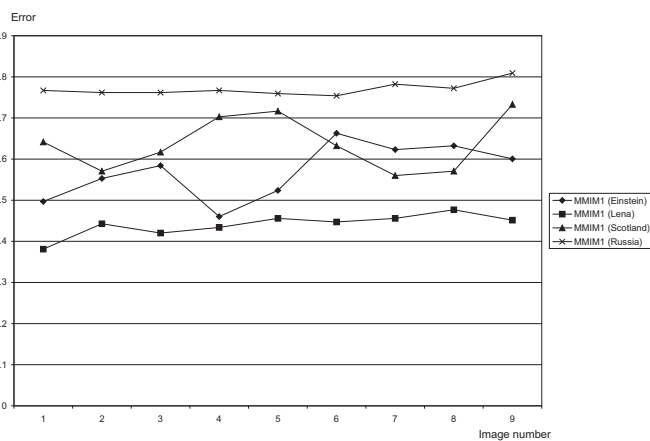
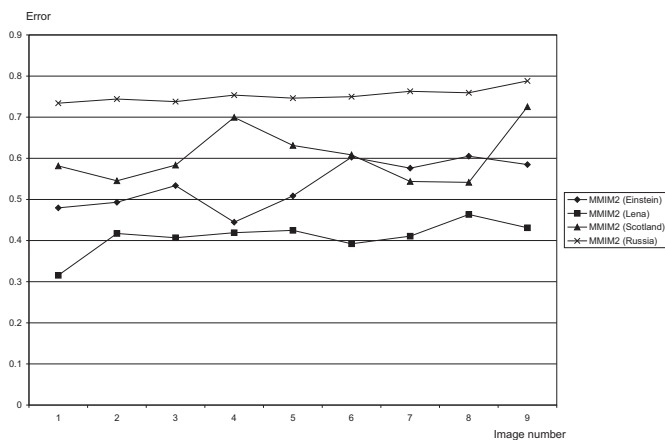


Figure A.13: \mathbb{L}_p with $p = 4$ (Section 4.2.1.10).

Figure A.14: \mathbb{L}_p with $p = 600$ (Section 4.2.1.10).Figure A.15: MMIM with $r = 1$ (Section 4.2.1.11).Figure A.16: MMIM with $r = 5$ (Section 4.2.1.11).

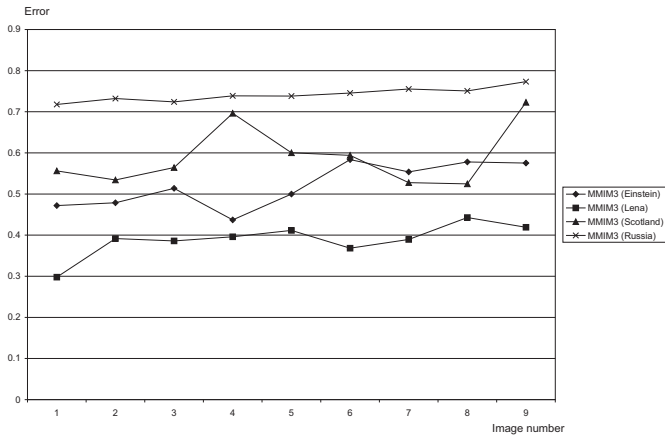


Figure A.17: MMIM with $r = 10$ (Section 4.2.1.11).

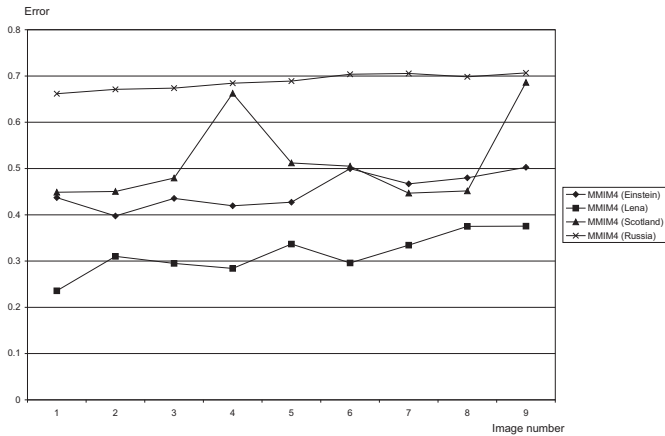


Figure A.18: MMIM with $r = 100$ (Section 4.2.1.11).

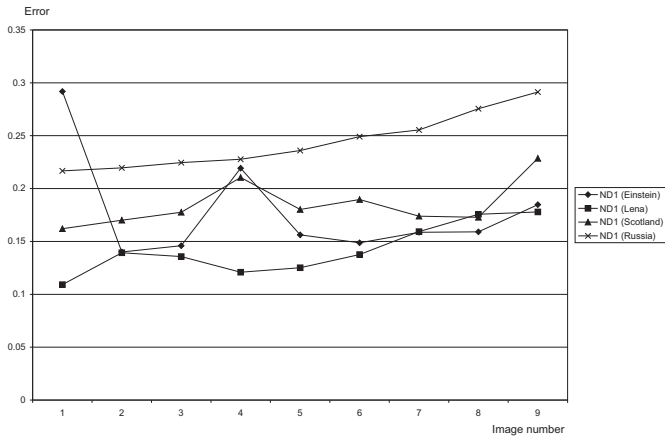
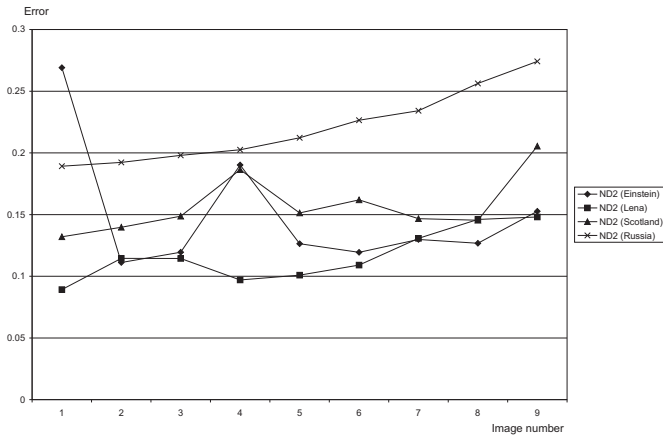
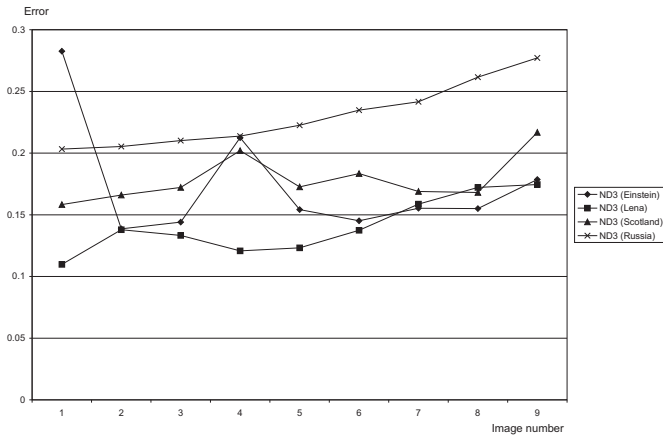
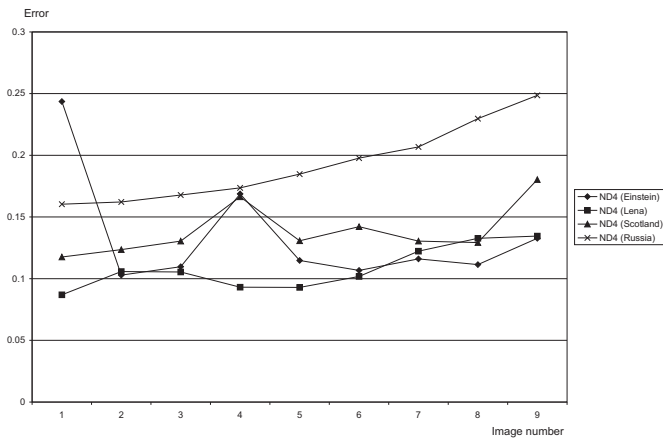


Figure A.19: ND with $w = 3$, $\alpha = 256$, $\beta = 1$ (Section 4.2.1.12).

Figure A.20: ND with $w = 5$, $\alpha = 256$, $\beta = 1$ (Section 4.2.1.12).Figure A.21: ND with $w = 3$, $\alpha = 64$, $\beta = 1$ (Section 4.2.1.12).Figure A.22: ND with $w = 5$, $\alpha = 64$, $\beta = 1$ (Section 4.2.1.12).

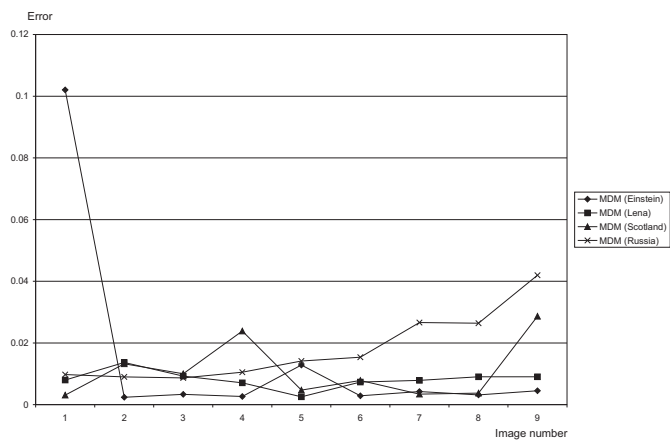


Figure A.23: MDM (Section 4.2.1.13).

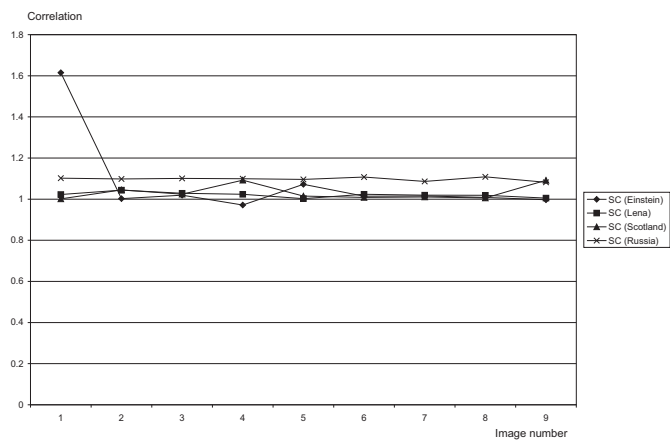


Figure A.24: SC (Section 4.2.1.14).

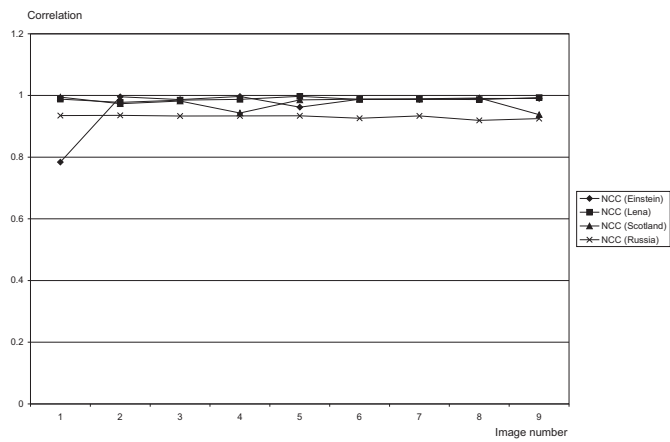
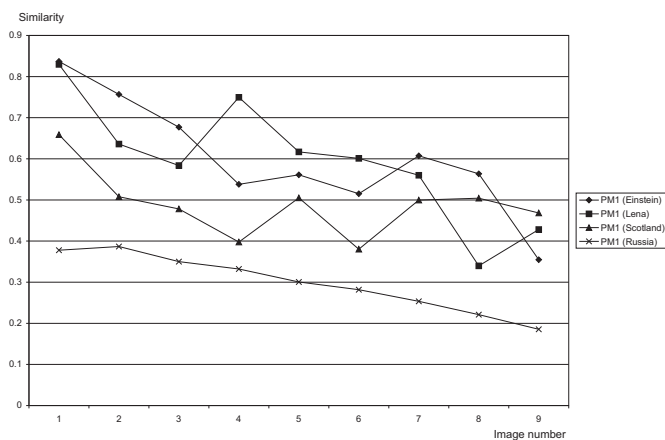
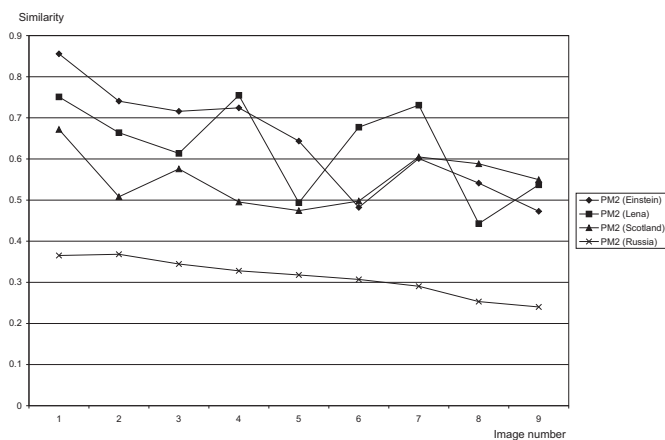
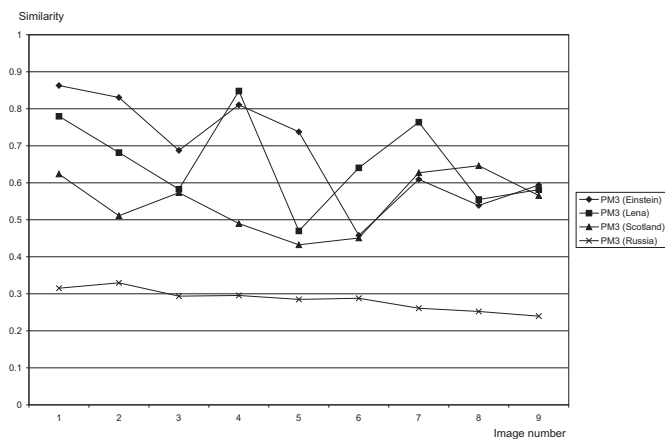


Figure A.25: NCC (Section 4.2.1.15).

Figure A.26: PM with $\sigma = 1, \alpha = 0.3, \beta = 1$ (Section 4.2.1.16).Figure A.27: PM with $\sigma = 2, \alpha = 0.3, \beta = 1$ (Section 4.2.1.16).Figure A.28: PM with $\sigma = 4, \alpha = 0.3, \beta = 1$ (Section 4.2.1.16).

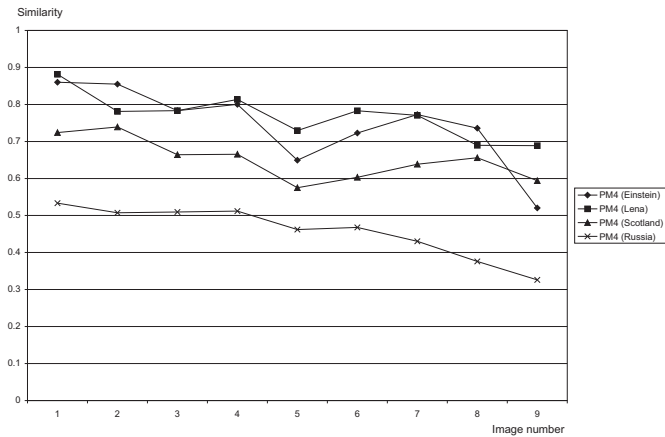


Figure A.29: PM with $\sigma = 1$, $\alpha = 0.2$, $\beta = 1$ (Section 4.2.1.16).

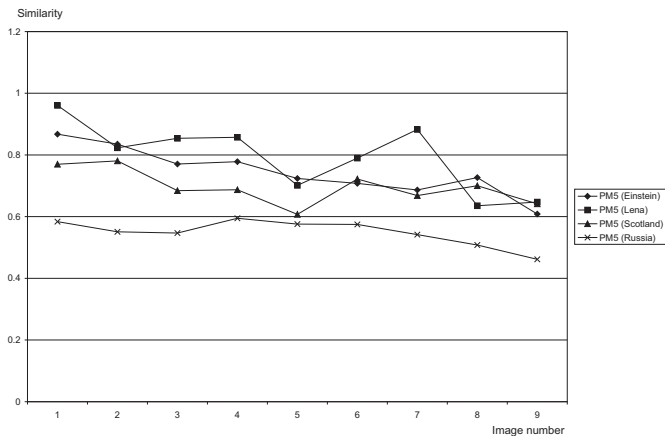


Figure A.30: PM with $\sigma = 2$, $\alpha = 0.2$, $\beta = 1$ (Section 4.2.1.16).

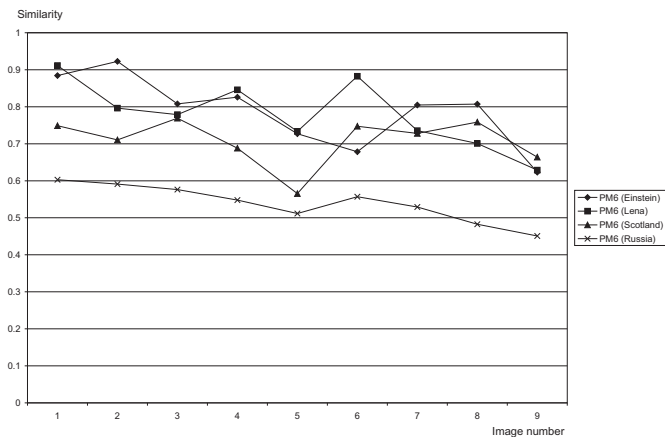
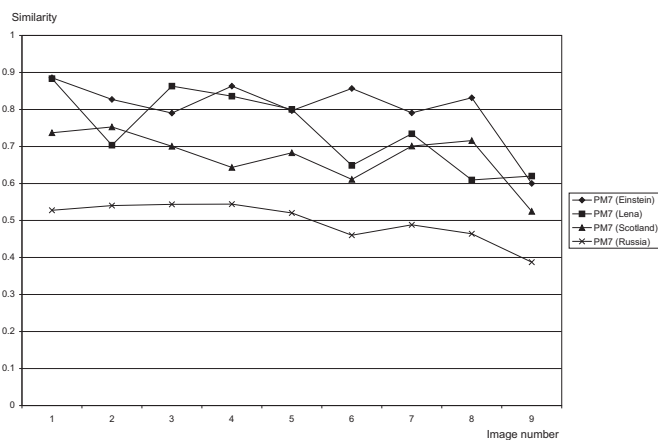
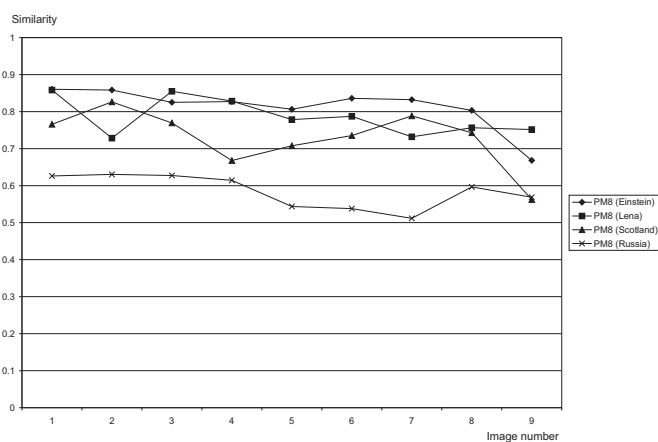
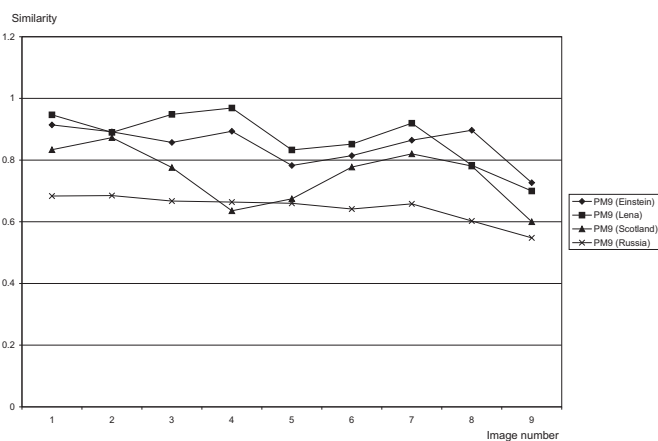


Figure A.31: PM with $\sigma = 4$, $\alpha = 0.2$, $\beta = 1$ (Section 4.2.1.16).

Figure A.32: PM with $\sigma = 1, \alpha = 0.4, \beta = 1$ (Section 4.2.1.16).Figure A.33: PM with $\sigma = 2, \alpha = 0.4, \beta = 1$ (Section 4.2.1.16).Figure A.34: PM with $\sigma = 4, \alpha = 0.4, \beta = 1$ (Section 4.2.1.16).

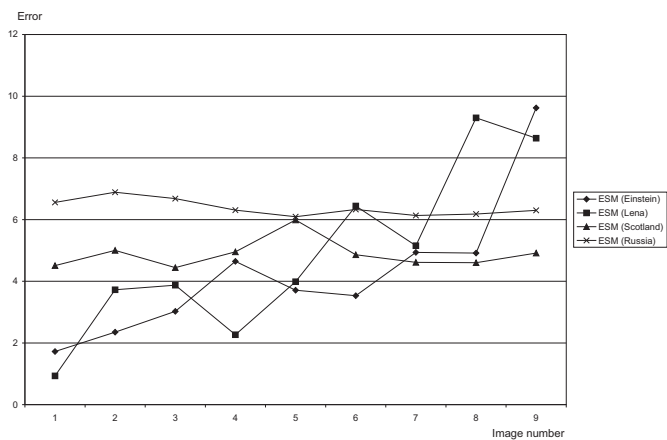


Figure A.35: ESM (Section 4.2.1.17).

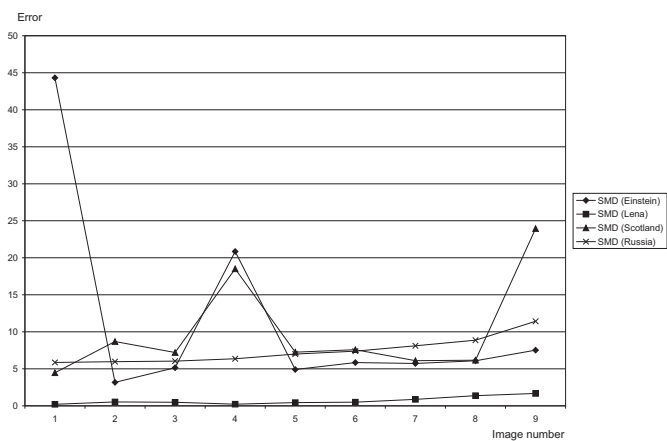


Figure A.36: SMD (Section 4.2.1.18).

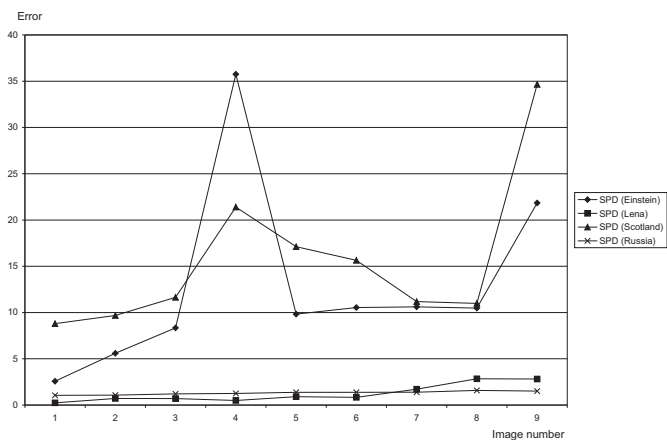
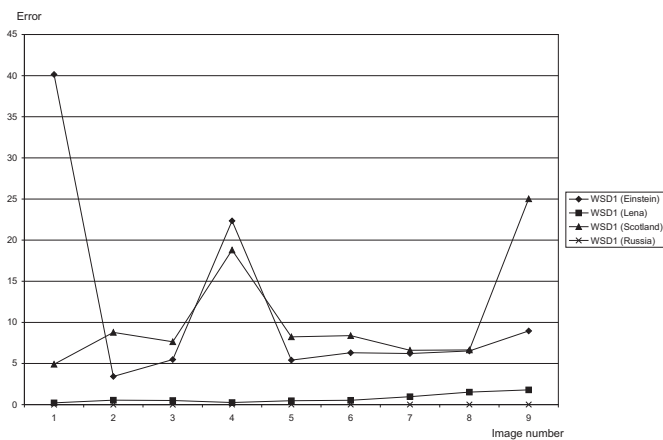
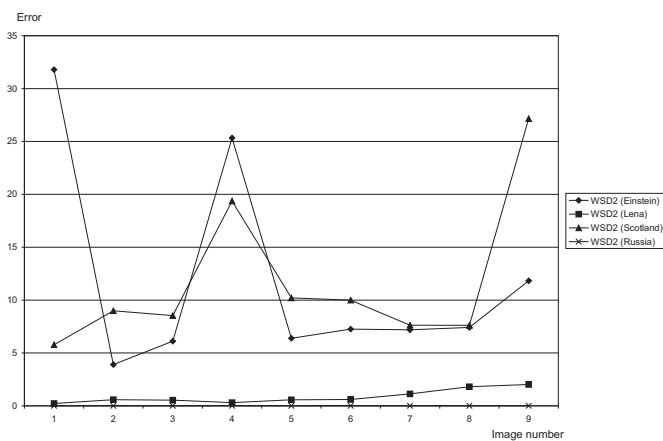
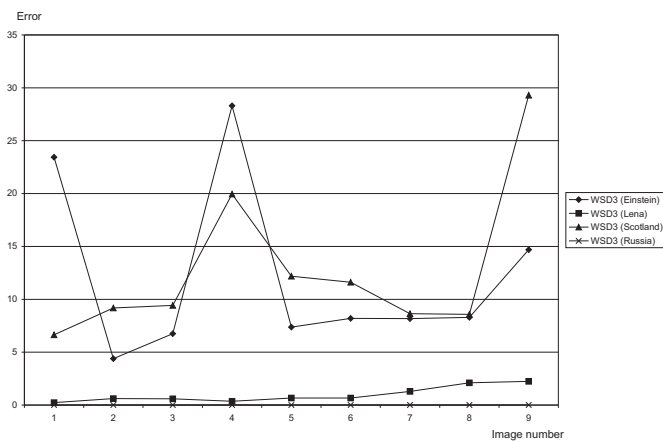


Figure A.37: SPD (Section 4.2.1.19).

Figure A.38: WSD with $\gamma = 0.1$ (Section 4.2.1.20).Figure A.39: WSD with $\gamma = 0.3$ (Section 4.2.1.20).Figure A.40: WSD with $\gamma = 0.5$ (Section 4.2.1.20).

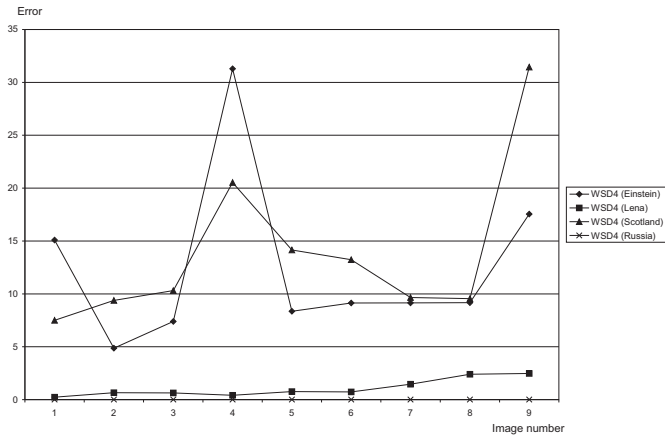


Figure A.41: WSD with $\gamma = 0.7$ (Section 4.2.1.20).

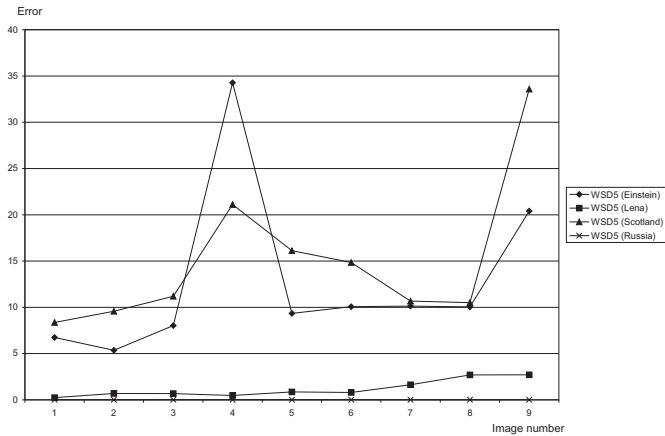


Figure A.42: WSD with $\gamma = 0.9$ (Section 4.2.1.20).

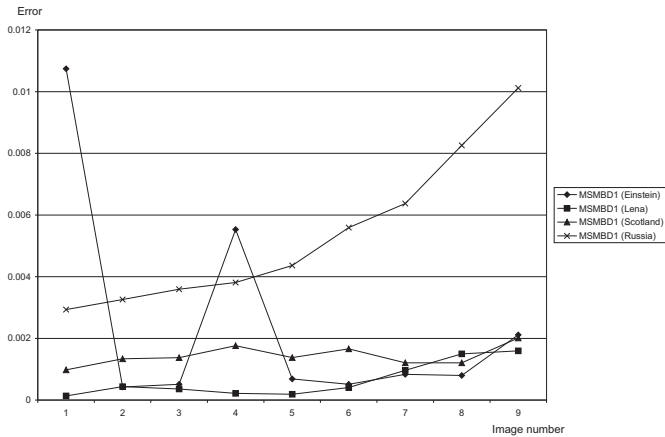
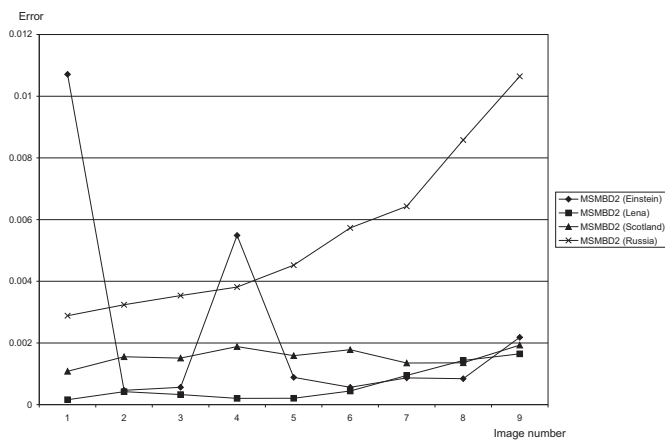
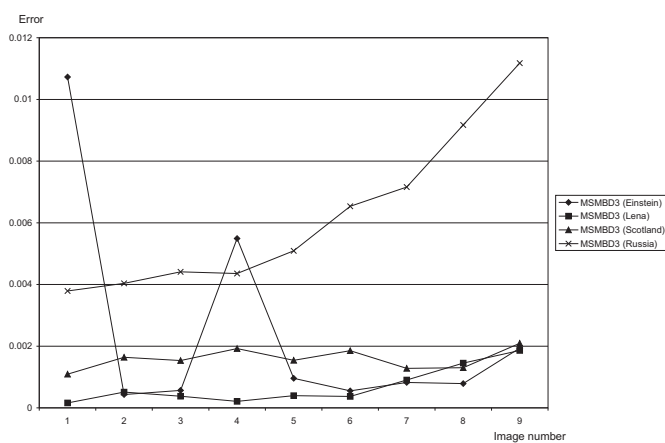
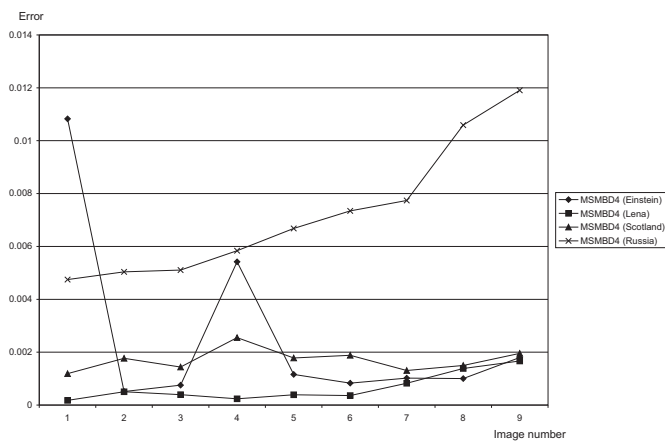


Figure A.43: MSMBD with blocksize $b = 4$ (Section 4.2.1.21).

Figure A.44: MSMBD with blocksize $b = 8$ (Section 4.2.1.21).Figure A.45: MSMBD with blocksize $b = 16$ (Section 4.2.1.21).Figure A.46: MSMBD with blocksize $b = 32$ (Section 4.2.1.21).

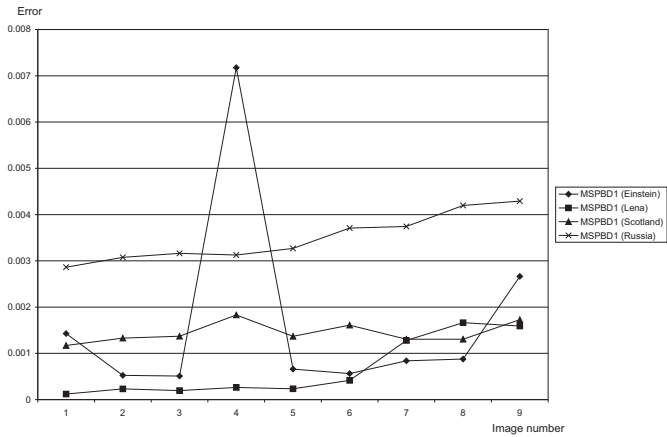


Figure A.47: MSPBD with blocksize $b = 4$ (Section 4.2.1.22).

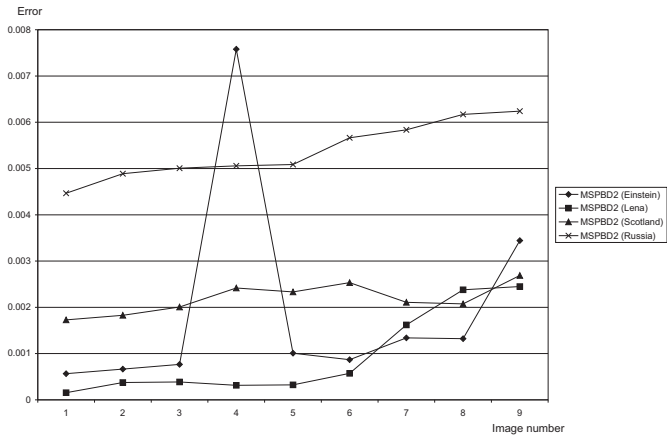


Figure A.48: MSPBD with blocksize $b = 8$ (Section 4.2.1.22).

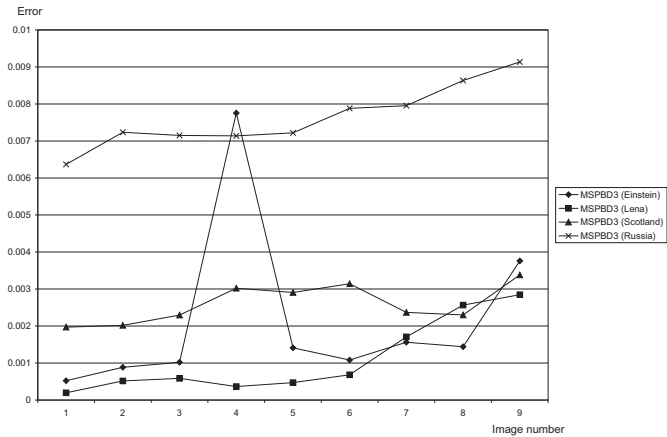
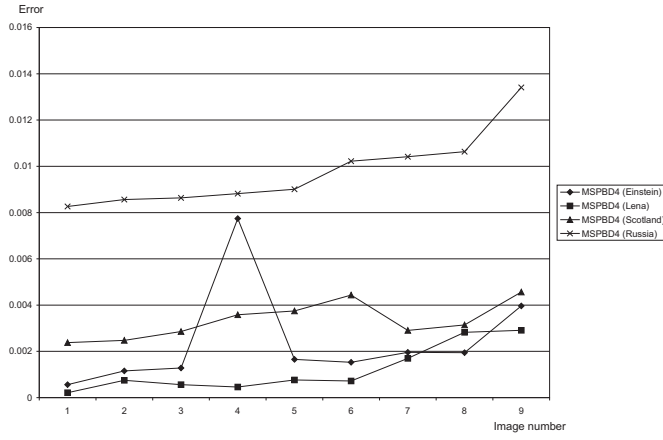
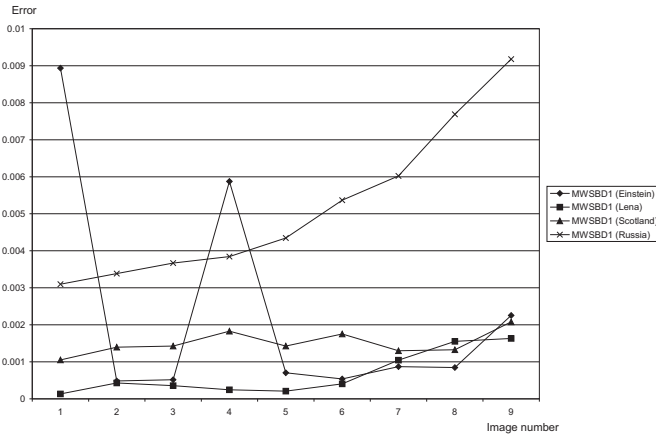
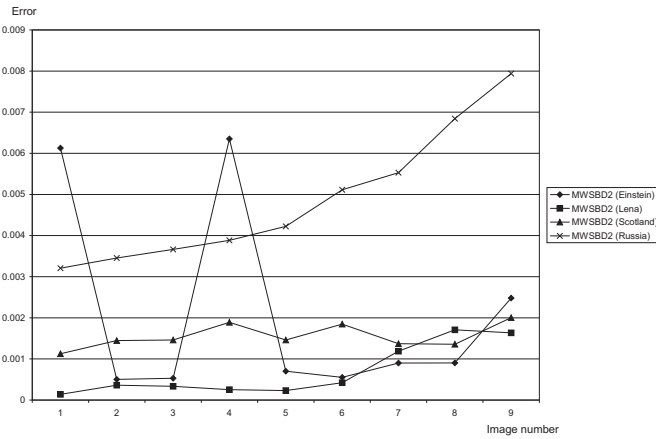


Figure A.49: MSPBD with blocksize $b = 16$ (Section 4.2.1.22).

Figure A.50: MSPBD with blocksize $b = 32$ (Section 4.2.1.22).Figure A.51: MWSBD with blocksize $b = 4$, $\gamma = 0.2$ (Section 4.2.1.23).Figure A.52: MWSBD with blocksize $b = 8$, $\gamma = 0.2$ (Section 4.2.1.23).

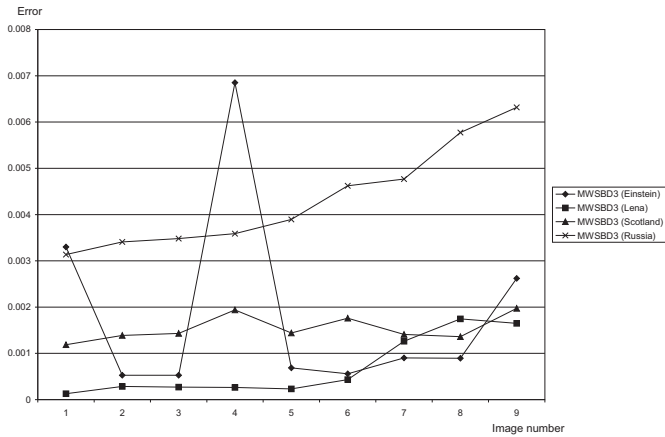


Figure A.53: MWSBD with blocksize $b = 16$, $\gamma = 0.2$ (Section 4.2.1.23).

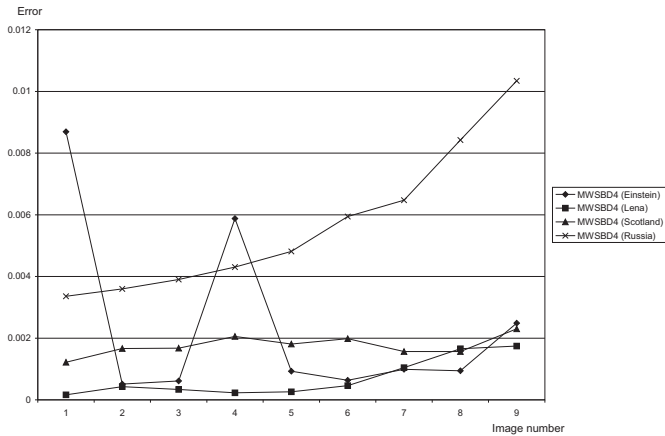


Figure A.54: MWSBD with blocksize $b = 32$, $\gamma = 0.2$ (Section 4.2.1.23).

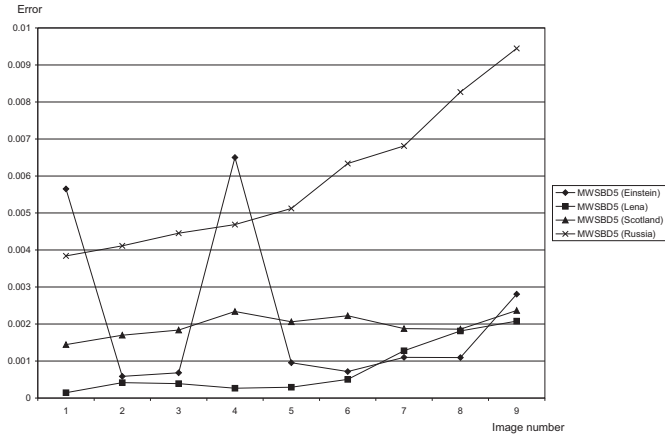
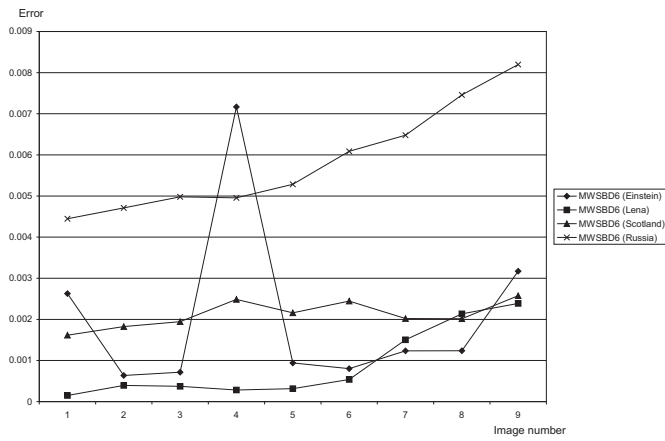
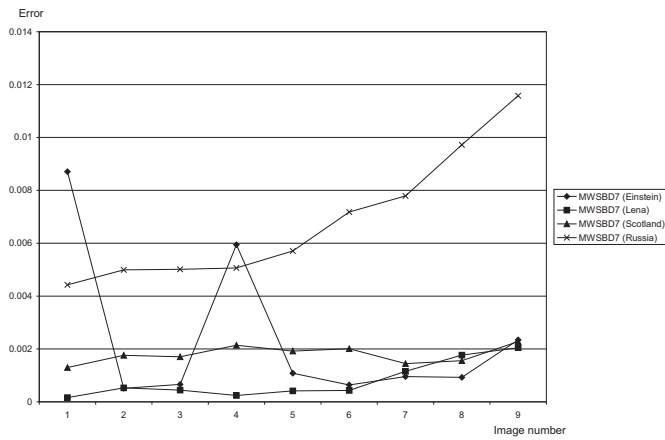
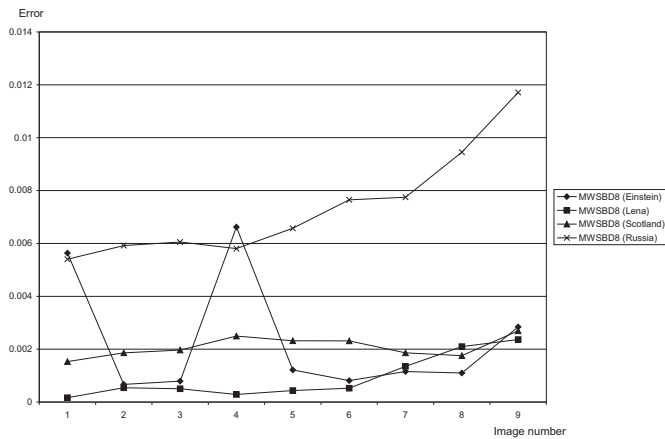


Figure A.55: MWSBD with blocksize $b = 4$, $\gamma = 0.5$ (Section 4.2.1.23).

Figure A.56: MWSBD with blocksize $b = 8$, $\gamma = 0.5$ (Section 4.2.1.23).Figure A.57: MWSBD with blocksize $b = 16$, $\gamma = 0.5$ (Section 4.2.1.23).Figure A.58: MWSBD with blocksize $b = 32$, $\gamma = 0.5$ (Section 4.2.1.23).

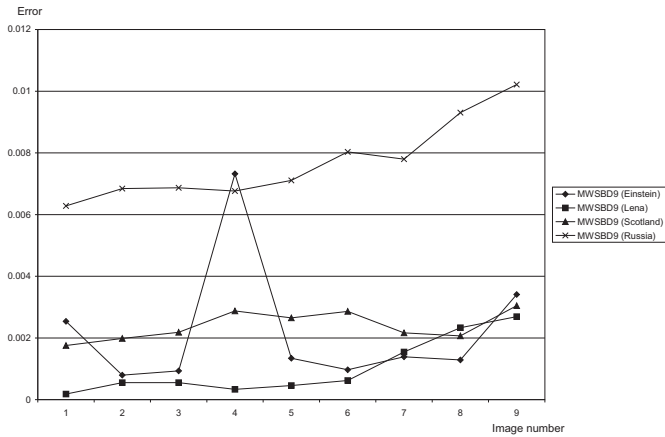


Figure A.59: MWSBD with blocksize $b = 4$, $\gamma = 0.8$ (Section 4.2.1.23).

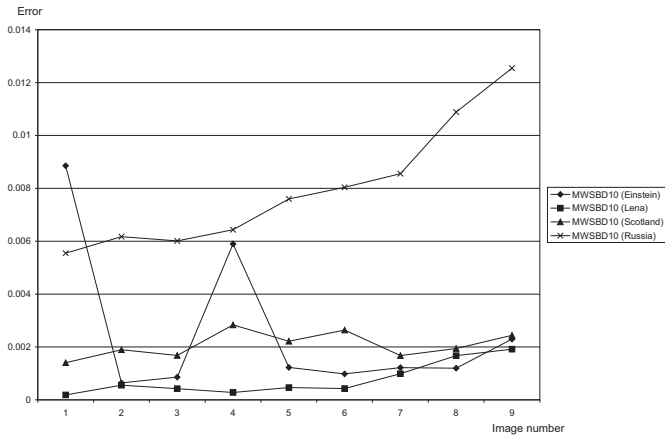


Figure A.60: MWSBD with blocksize $b = 8$, $\gamma = 0.8$ (Section 4.2.1.23).

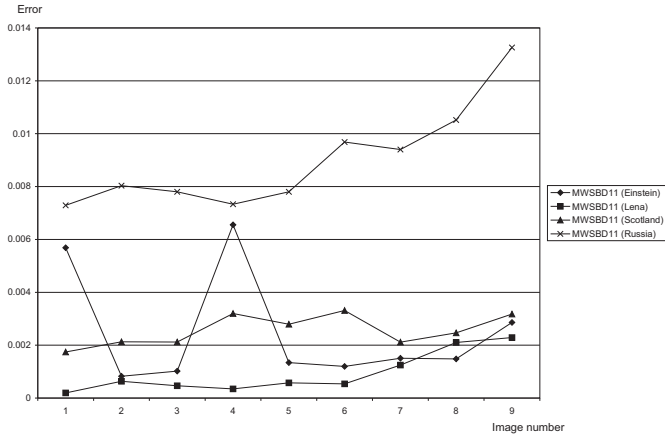
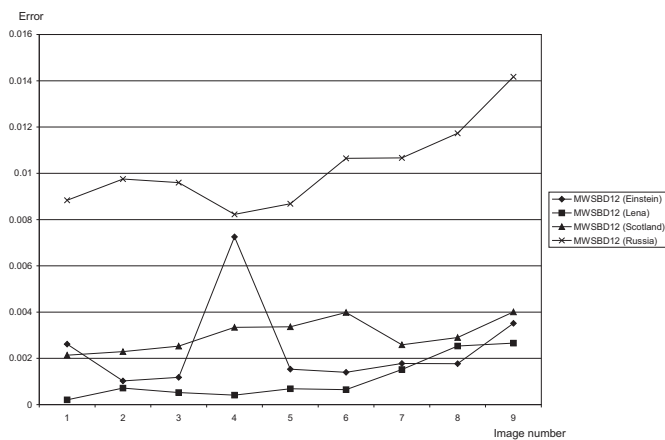
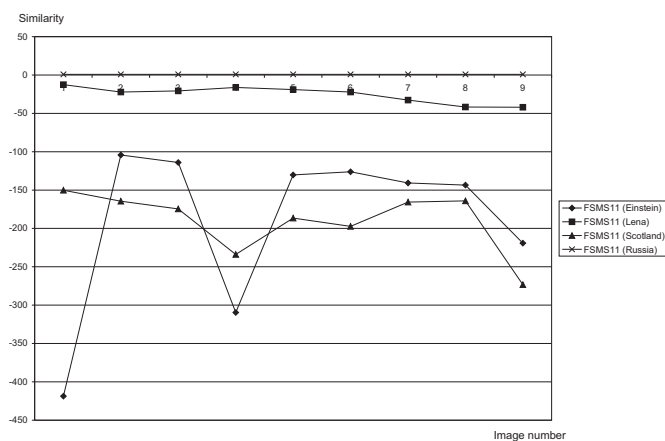
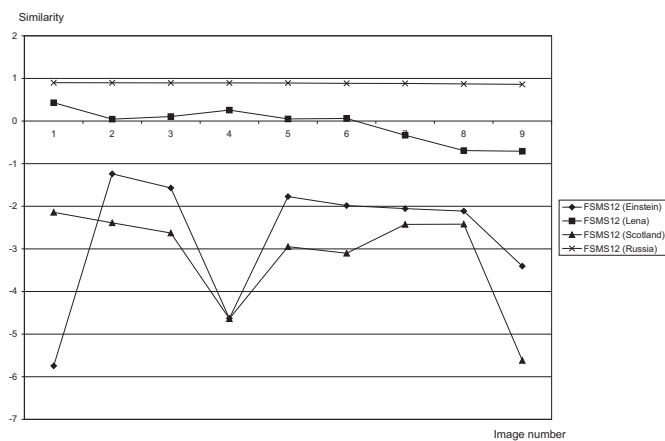


Figure A.61: MWSBD with blocksize $b = 16$, $\gamma = 0.8$ (Section 4.2.1.23).

Figure A.62: MWSBD with blocksize $b = 32$, $\gamma = 0.8$ (Section 4.2.1.23).Figure A.63: S1 with $p = 1$ (Section 4.2.1.24).Figure A.64: S1 with $p = 2$ (Section 4.2.1.24).

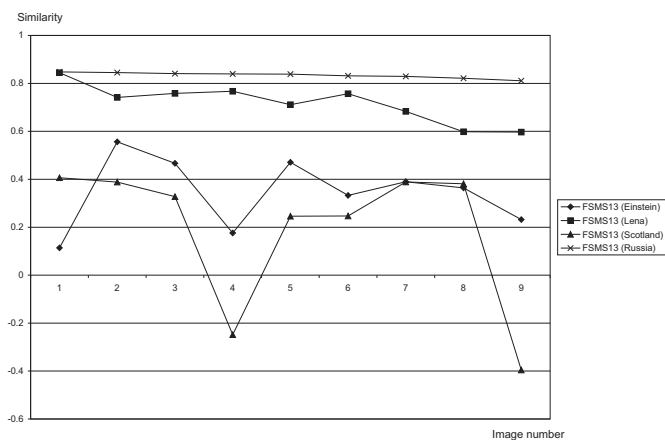


Figure A.65: S1 with $p = 4$ (Section 4.2.1.24).

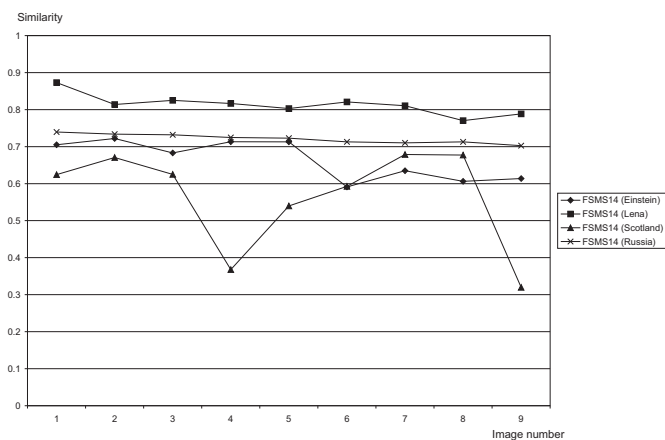


Figure A.66: S1 with $p = 10$ (Section 4.2.1.24).

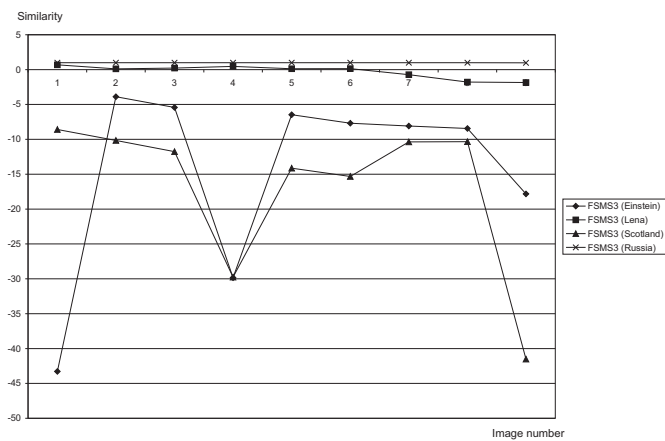


Figure A.67: S3 (Section 4.2.1.24).

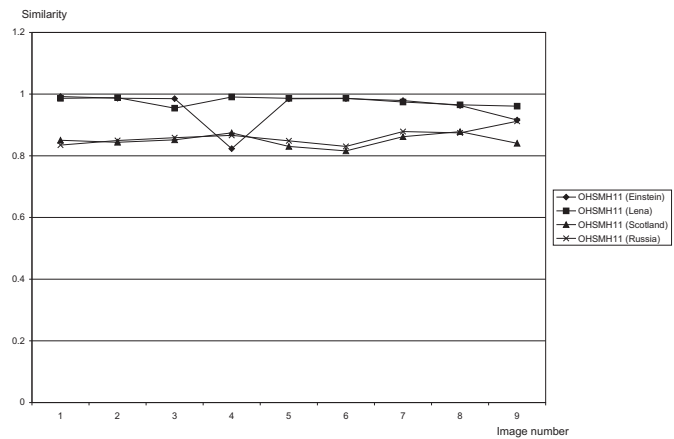


Figure A.68: H1 with $p = 1$ (Section 4.2.1.25).

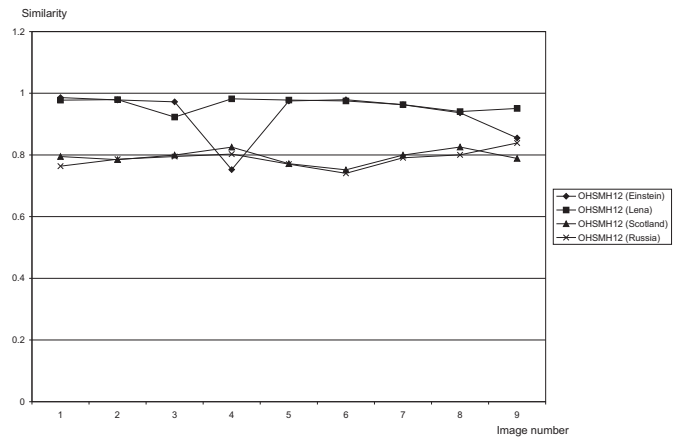


Figure A.69: H1 with $p = 2$ (Section 4.2.1.25).

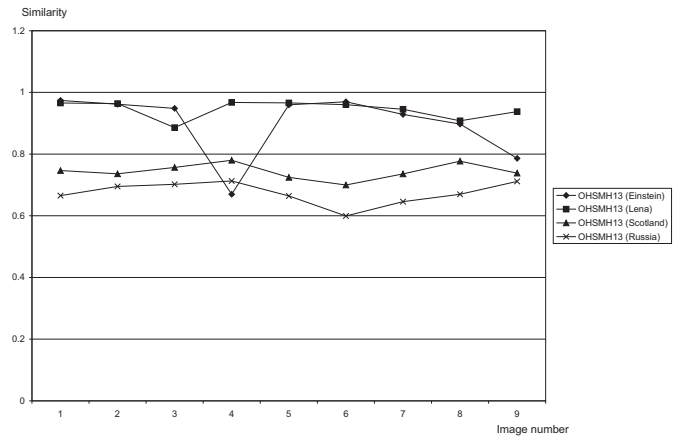


Figure A.70: H1 with $p = 4$ (Section 4.2.1.25).

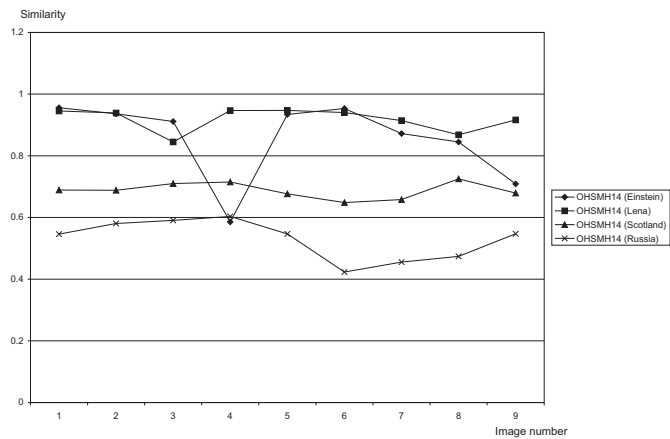


Figure A.71: H1 with $p = 10$ (Section 4.2.1.25).

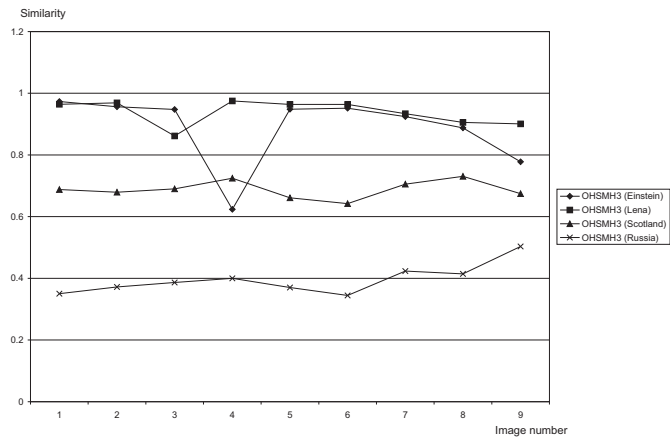


Figure A.72: H3 (Section 4.2.1.25).

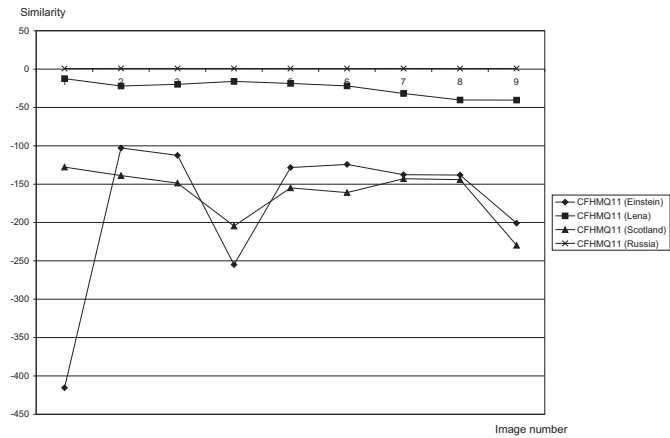
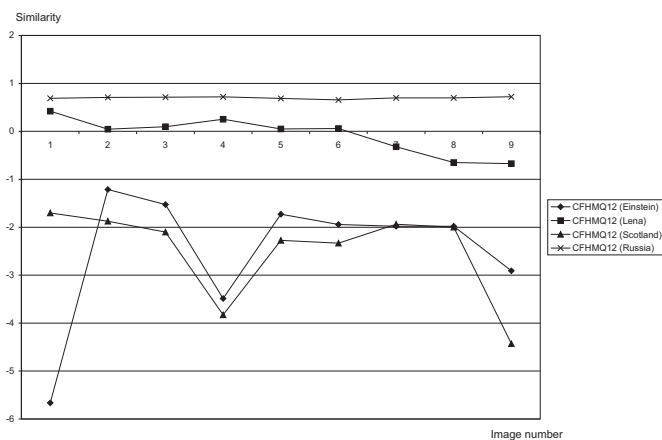
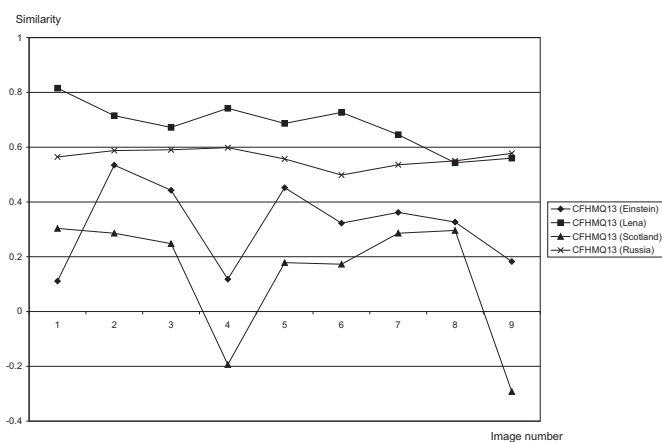
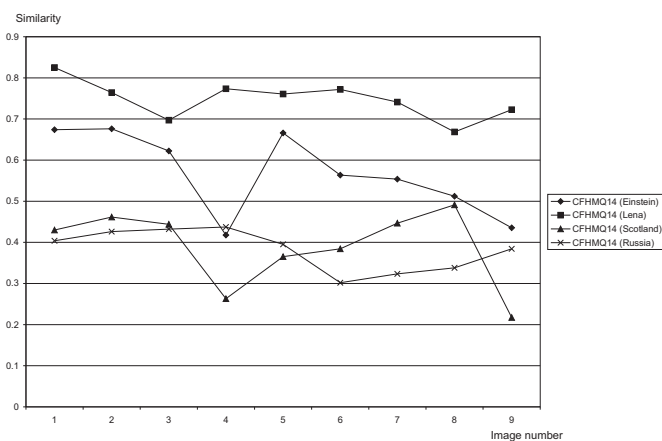


Figure A.73: Q1 with $p = 1$ (Section 4.2.1.26).

Figure A.74: Q1 with $p = 2$ (Section 4.2.1.26).Figure A.75: Q1 with $p = 4$ (Section 4.2.1.26).Figure A.76: Q1 with $p = 10$ (Section 4.2.1.26).

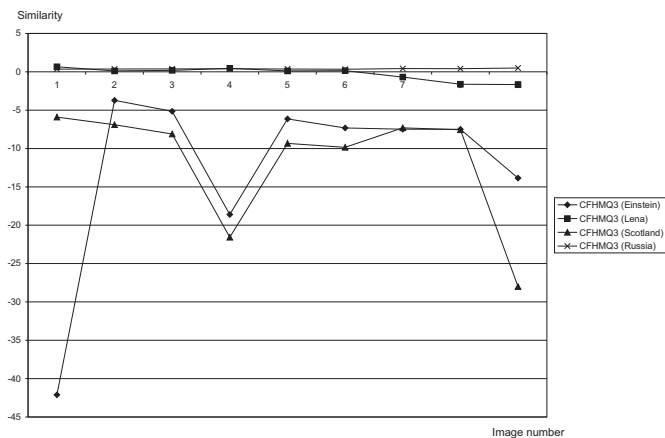


Figure A.77: Q3 (Section 4.2.1.26).

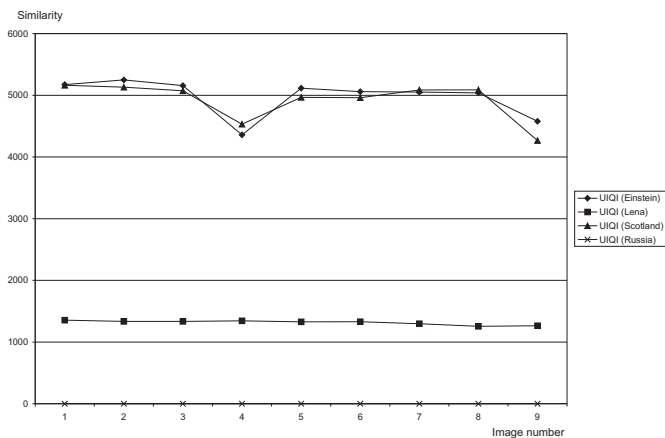


Figure A.78: UIQI (Section 4.2.1.27).

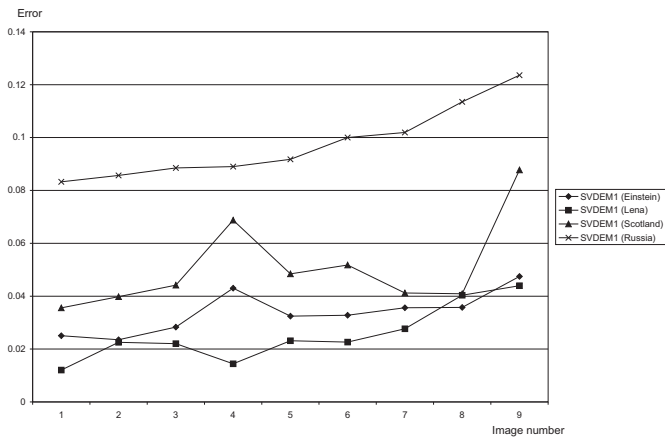
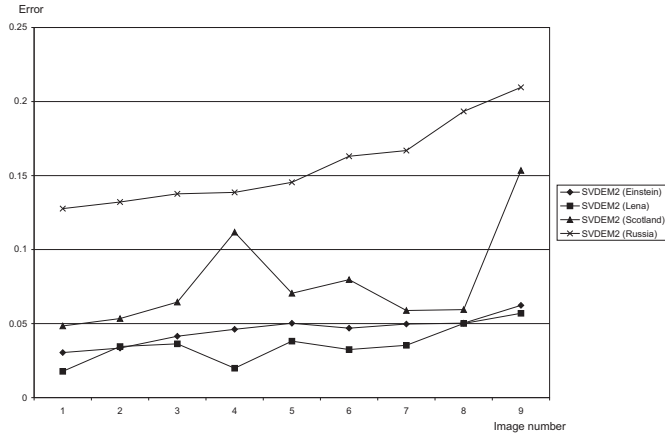
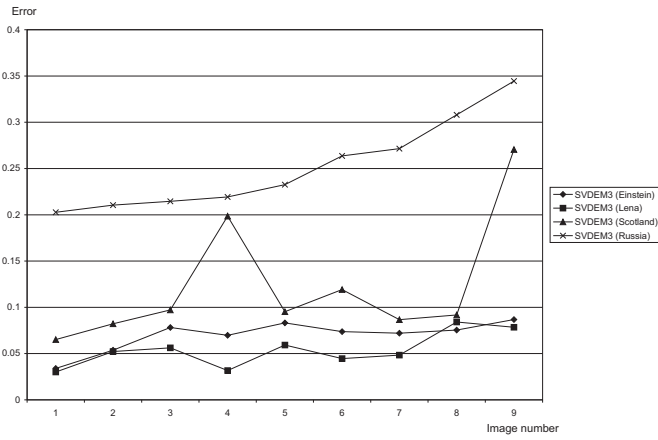
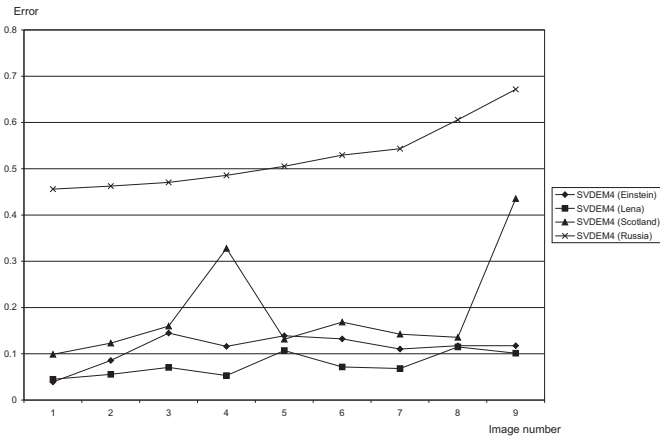


Figure A.79: M_{SVD} with blocksize $b = 2$ (Section 4.2.1.28).

Figure A.80: M_{SVD} with blocksize $b = 4$ (Section 4.2.1.28).Figure A.81: M_{SVD} with blocksize $b = 8$ (Section 4.2.1.28).Figure A.82: M_{SVD} with blocksize $b = 16$ (Section 4.2.1.28).

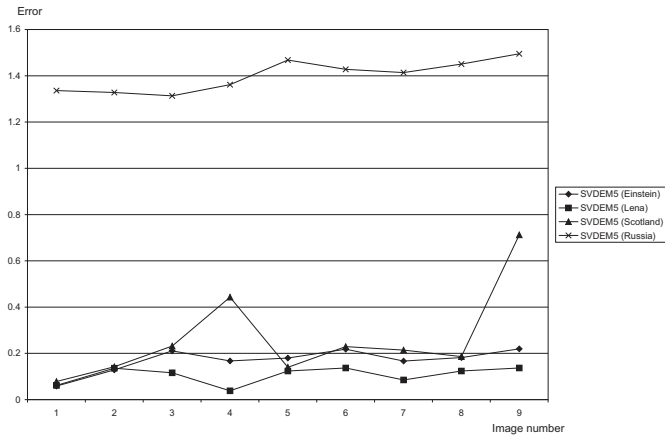


Figure A.83: M_{SVD} with blocksize $b = 32$ (Section 4.2.1.28).

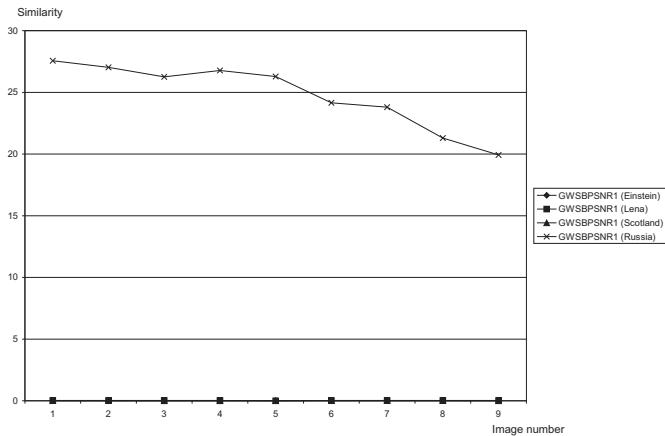


Figure A.84: GWSB-PSNR with blocksize $b = 2$ (Section 4.2.1.29).

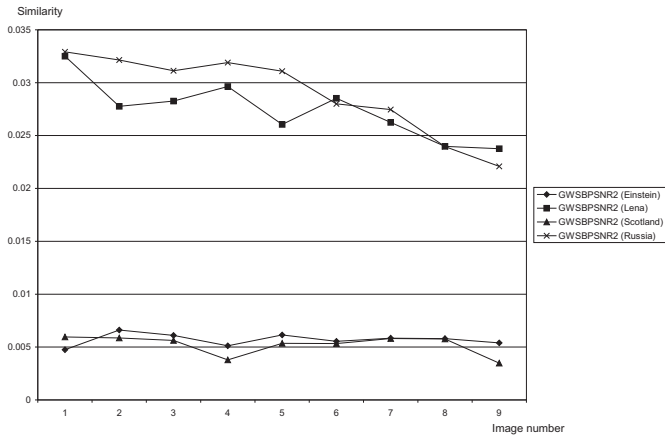
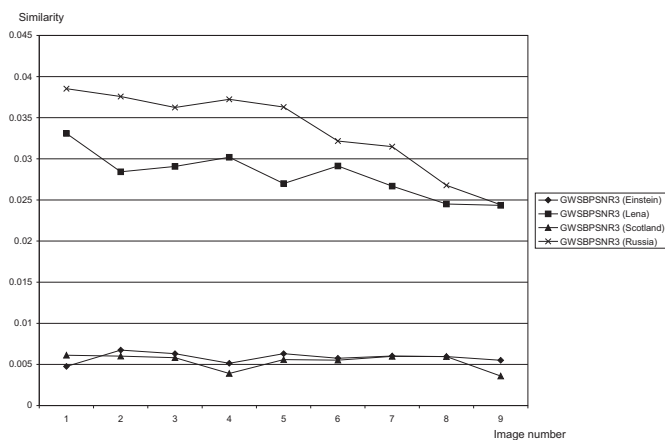
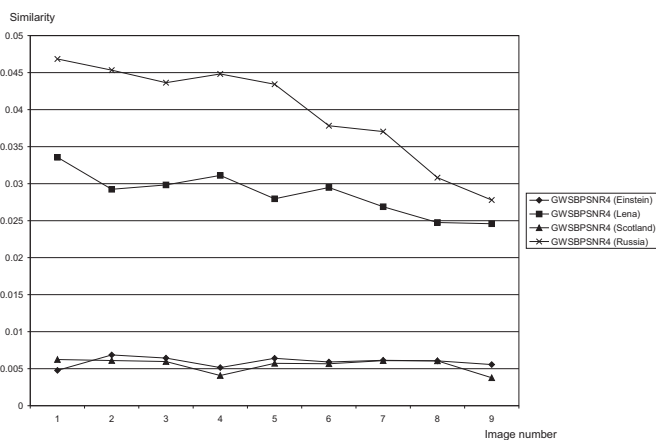
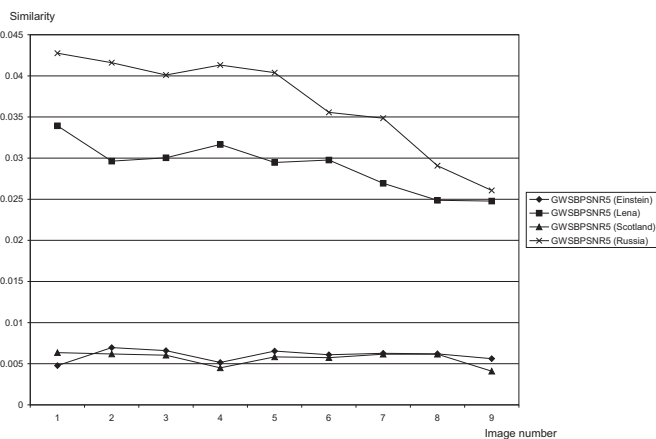


Figure A.85: GWSB-PSNR with blocksize $b = 4$ (Section 4.2.1.29).

Figure A.86: GWSB-PSNR with blocksize $b = 8$ (Section 4.2.1.29).Figure A.87: GWSB-PSNR with blocksize $b = 16$ (Section 4.2.1.29).Figure A.88: GWSB-PSNR with blocksize $b = 32$ (Section 4.2.1.28).

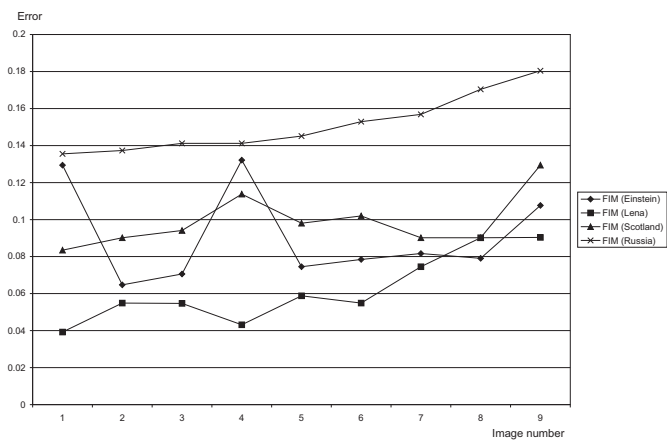


Figure A.89: FIM (Section 4.2.1.30).

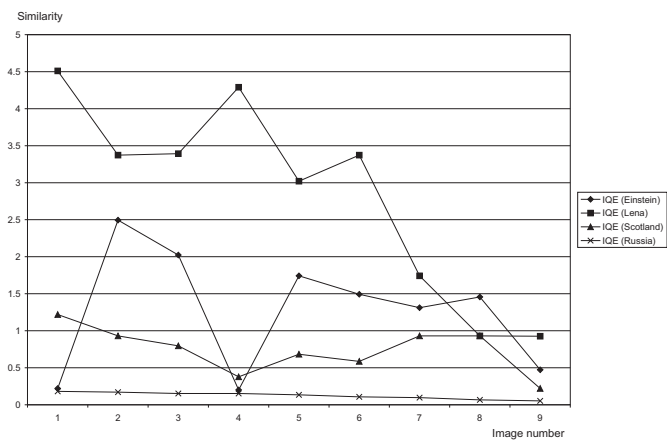


Figure A.90: IQE (Section 4.2.1.31).

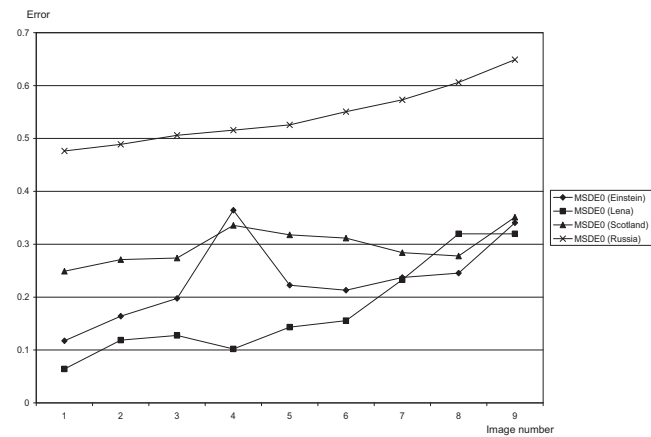
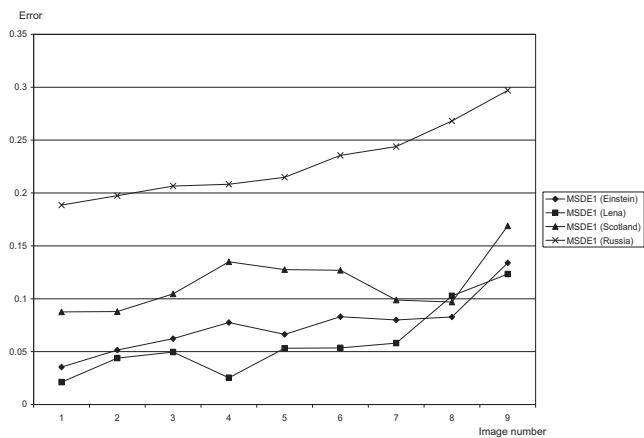
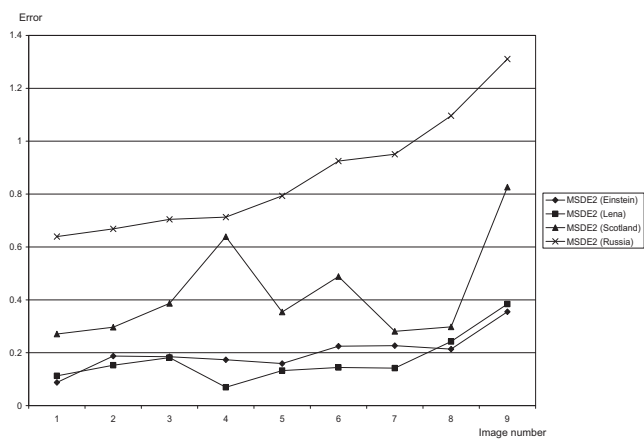
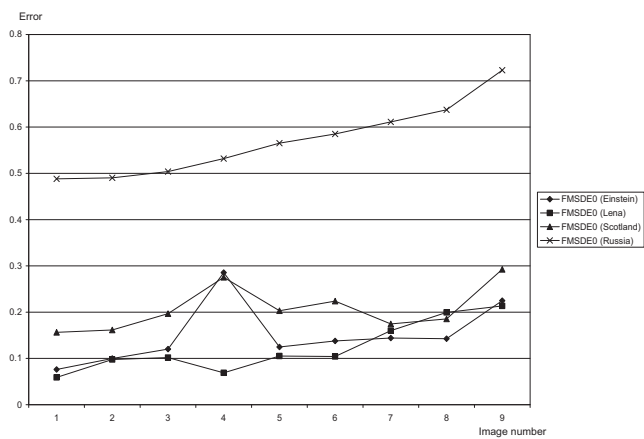


Figure A.91: MSDE with $\sigma_1 = 0.6$ (Section 4.2.2).

Figure A.92: MSDE with $\sigma_1 = 1.4$ (Section 4.2.2).Figure A.93: MSDE with $\sigma_1 = 3.1$ (Section 4.2.2).Figure A.94: FMSDE with $\alpha = 0$ (Section 4.2.2).

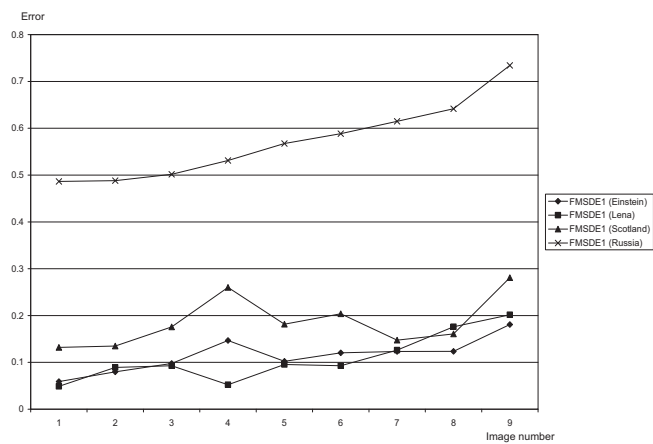


Figure A.95: FMSDE with $\alpha = 1$ (Section 4.2.2).

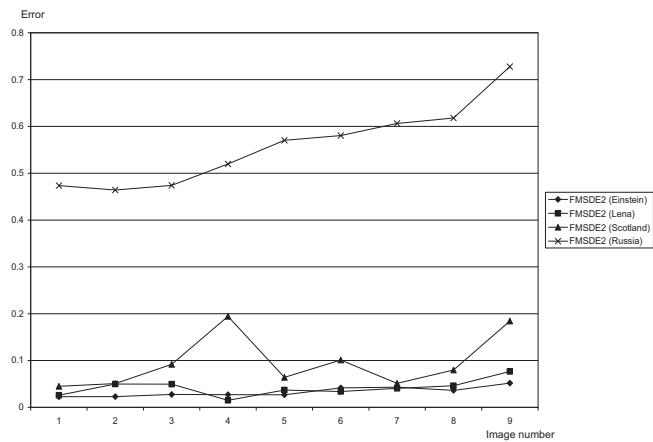


Figure A.96: FMSDE with $\alpha = 4$ (Section 4.2.2).

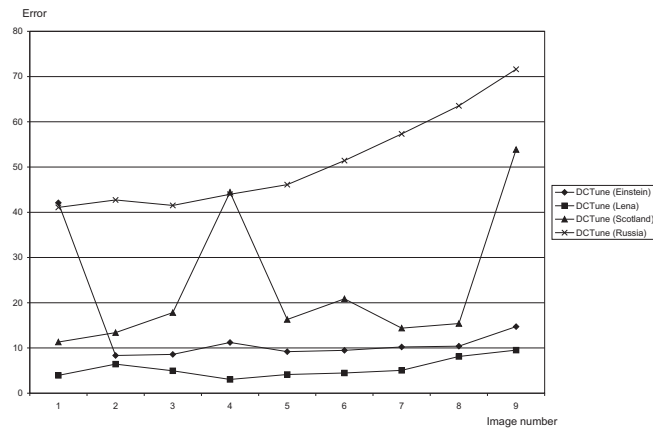


Figure A.97: DCTune 2.0 (Section 4.2.3.1).

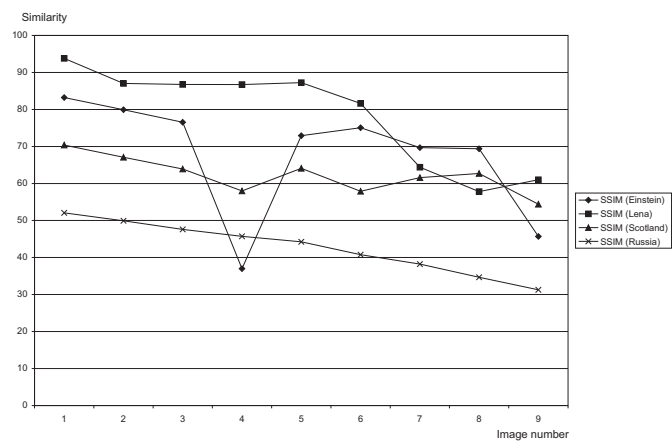


Figure A.98: SSIM (Section 4.2.3.2).

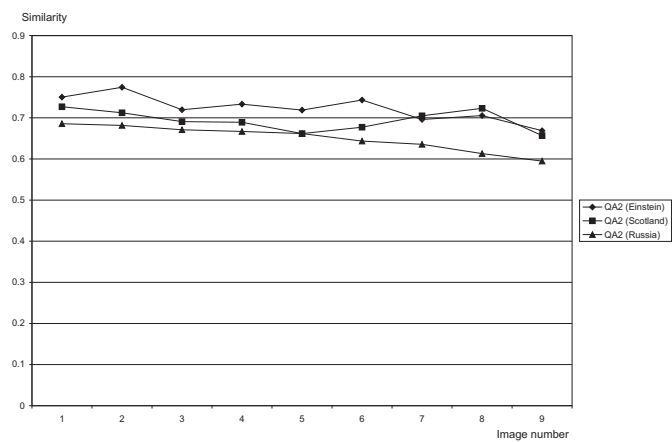
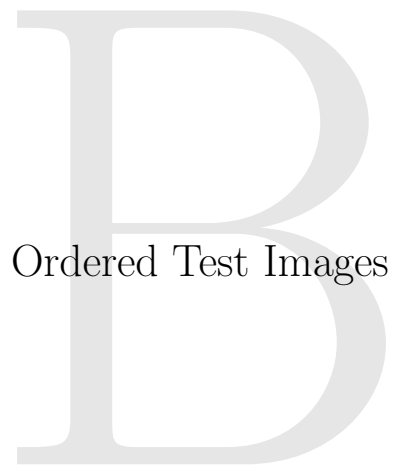


Figure A.99: Quality Assessor v2.0 (Section 4.2.3.3).



Ordered Test Images

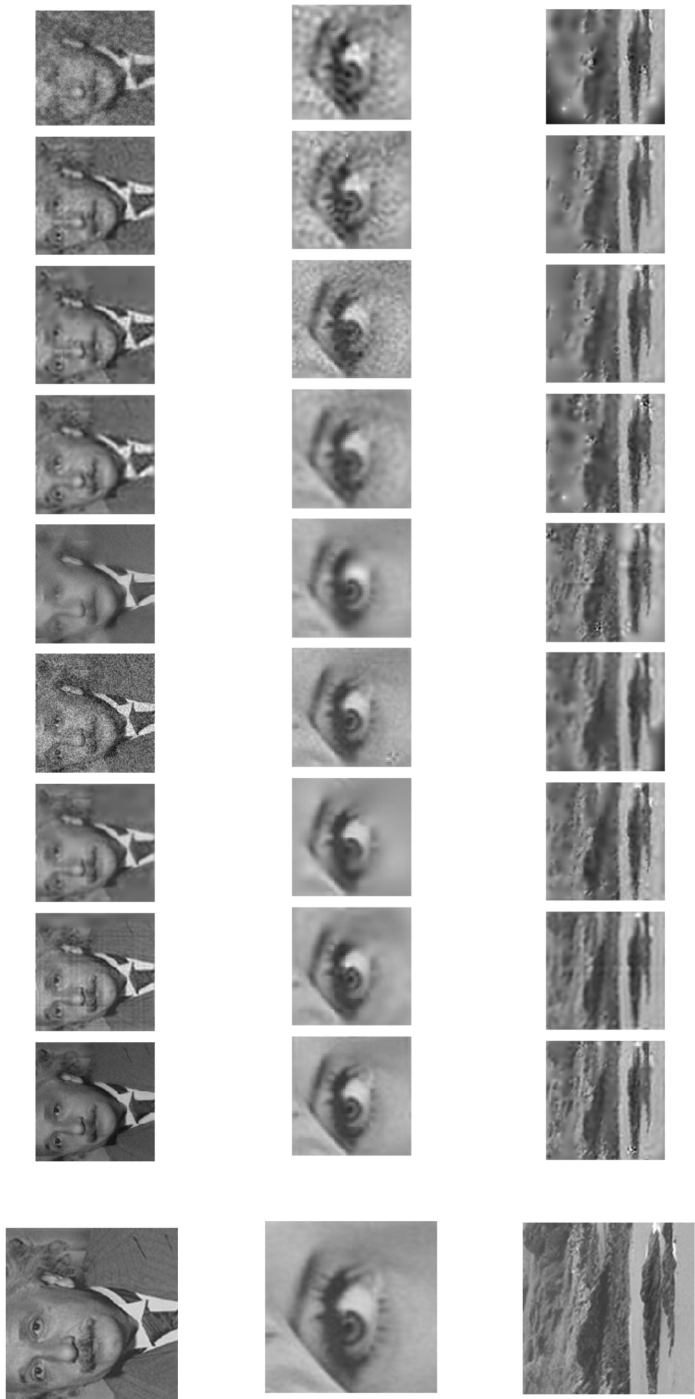


Figure B.1: Test images for the experiments of chapter 4, ordered by human observers. The large images represent the originals, the small images are degraded images rated from best to worse respectively averaged over 33 human observers.



Interest Points of Test Images

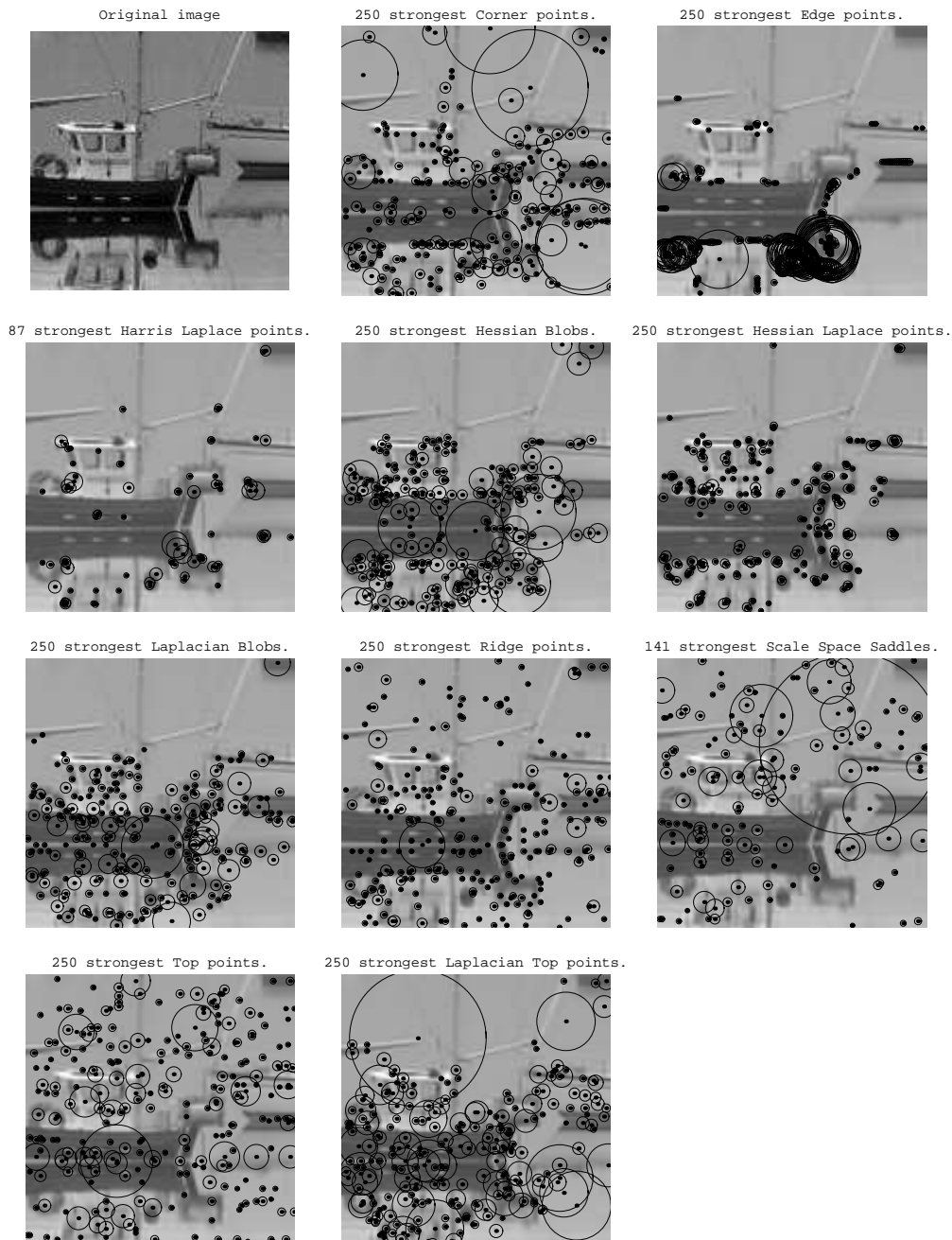


Figure C.1: Interest points of the *boat.jpg* image projected on the original image. The size of the circles represent the scale of the interest points. Maximum 250 points of each set are shown for the sake of clarity.

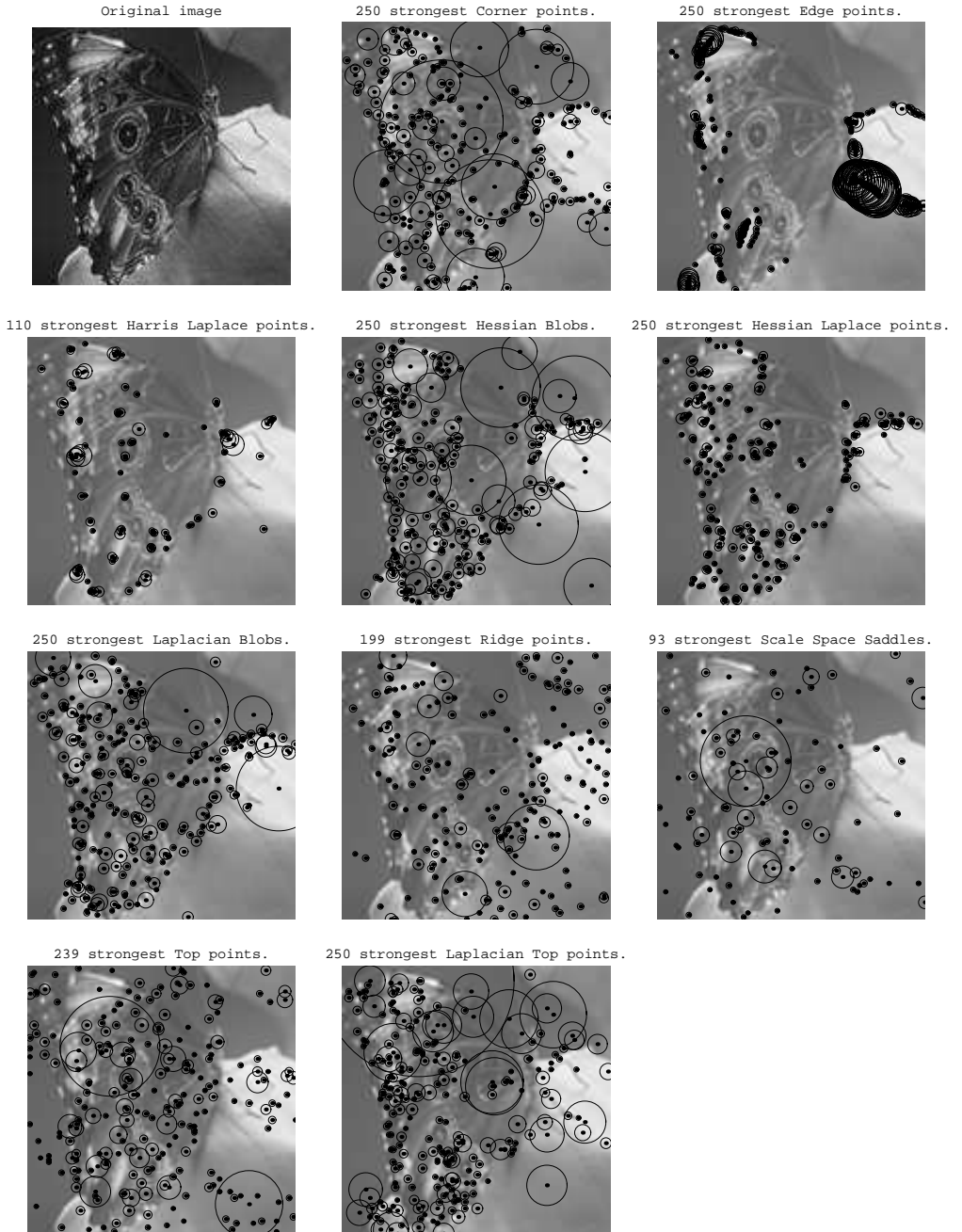


Figure C.2: Interest points of the *butterfly.jpg* image projected on the original image. The size of the circles represent the scale of the interest points. Maximum 250 points of each set are shown for the sake of clarity.

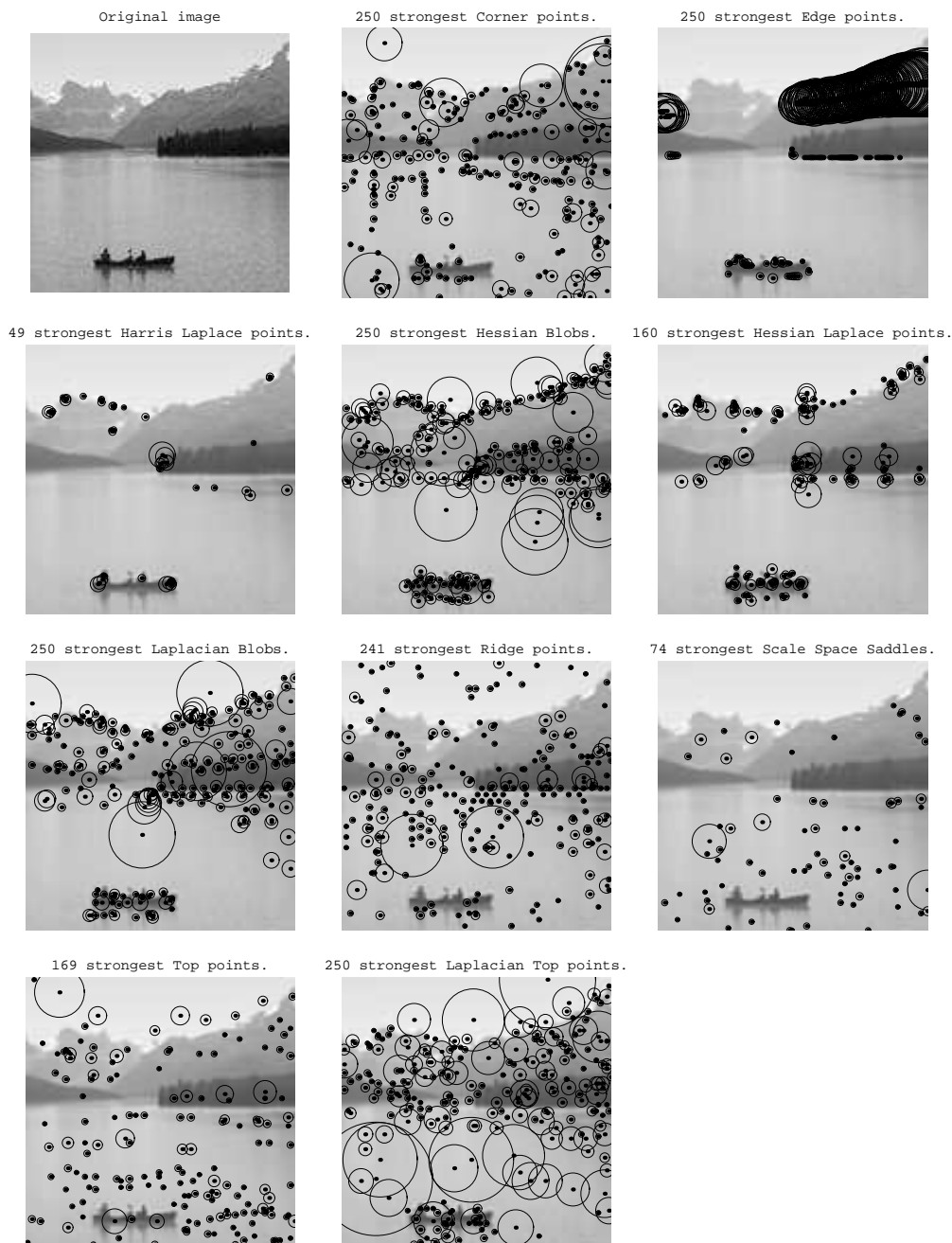


Figure C.3: Interest points of the *canada1.jpg* image projected on the original image. The size of the circles represent the scale of the interest points. Maximum 250 points of each set are shown for the sake of clarity.

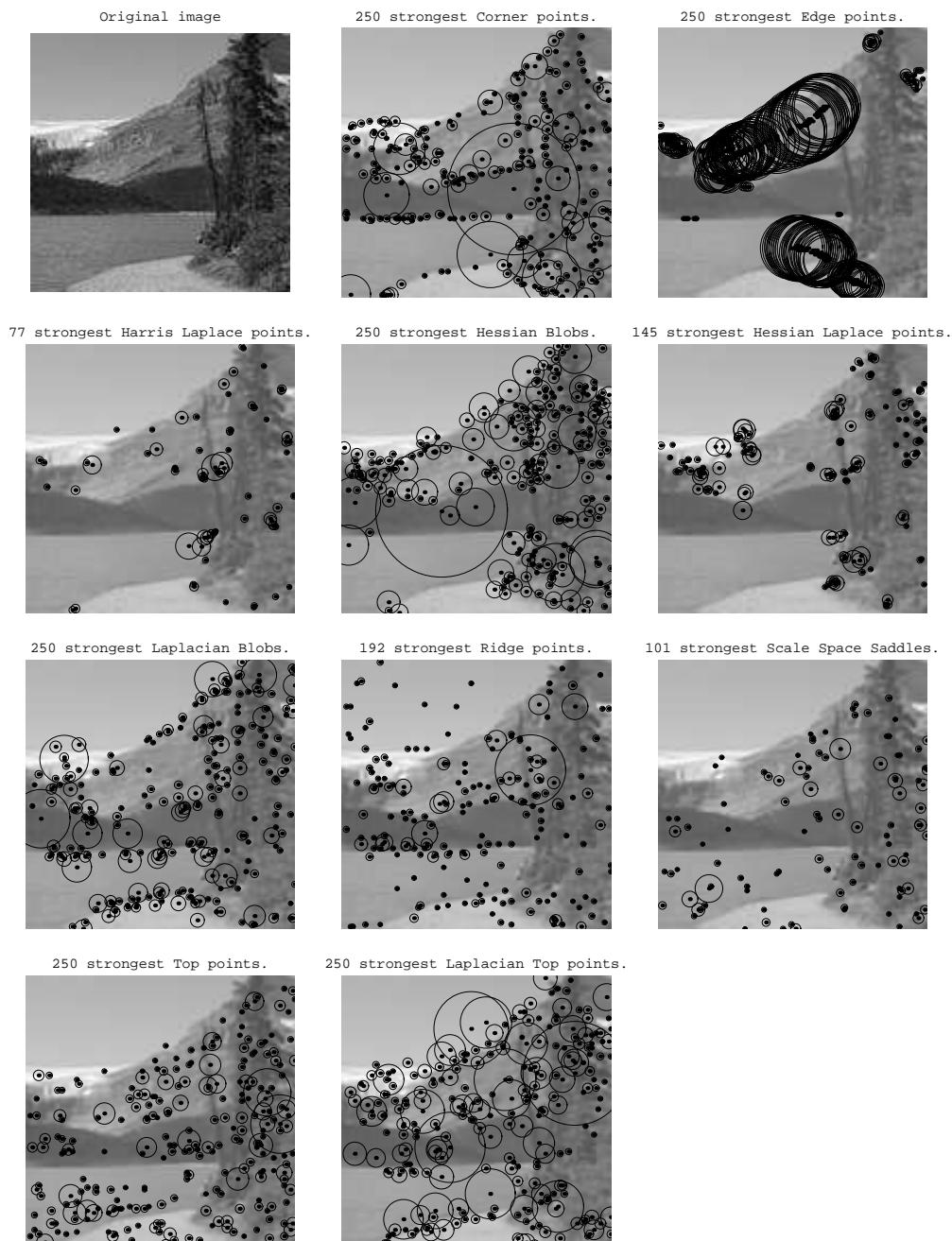


Figure C.4: Interest points of the *canada2.jpg* image projected on the original image. The size of the circles represent the scale of the interest points. Maximum 250 points of each set are shown for the sake of clarity.

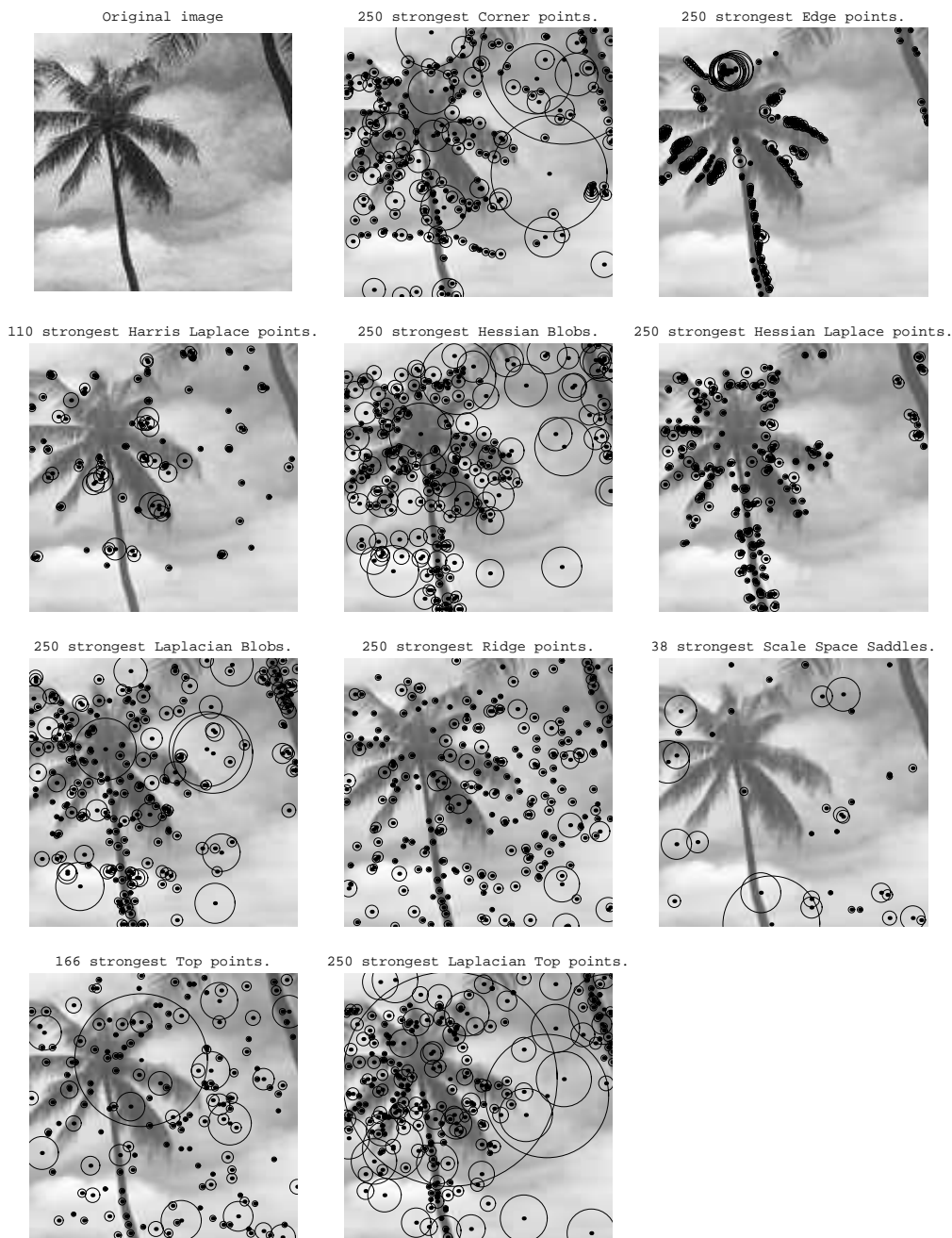


Figure C.5: Interest points of the *hawaii1.jpg* image projected on the original image. The size of the circles represent the scale of the interest points. Maximum 250 points of each set are shown for the sake of clarity.

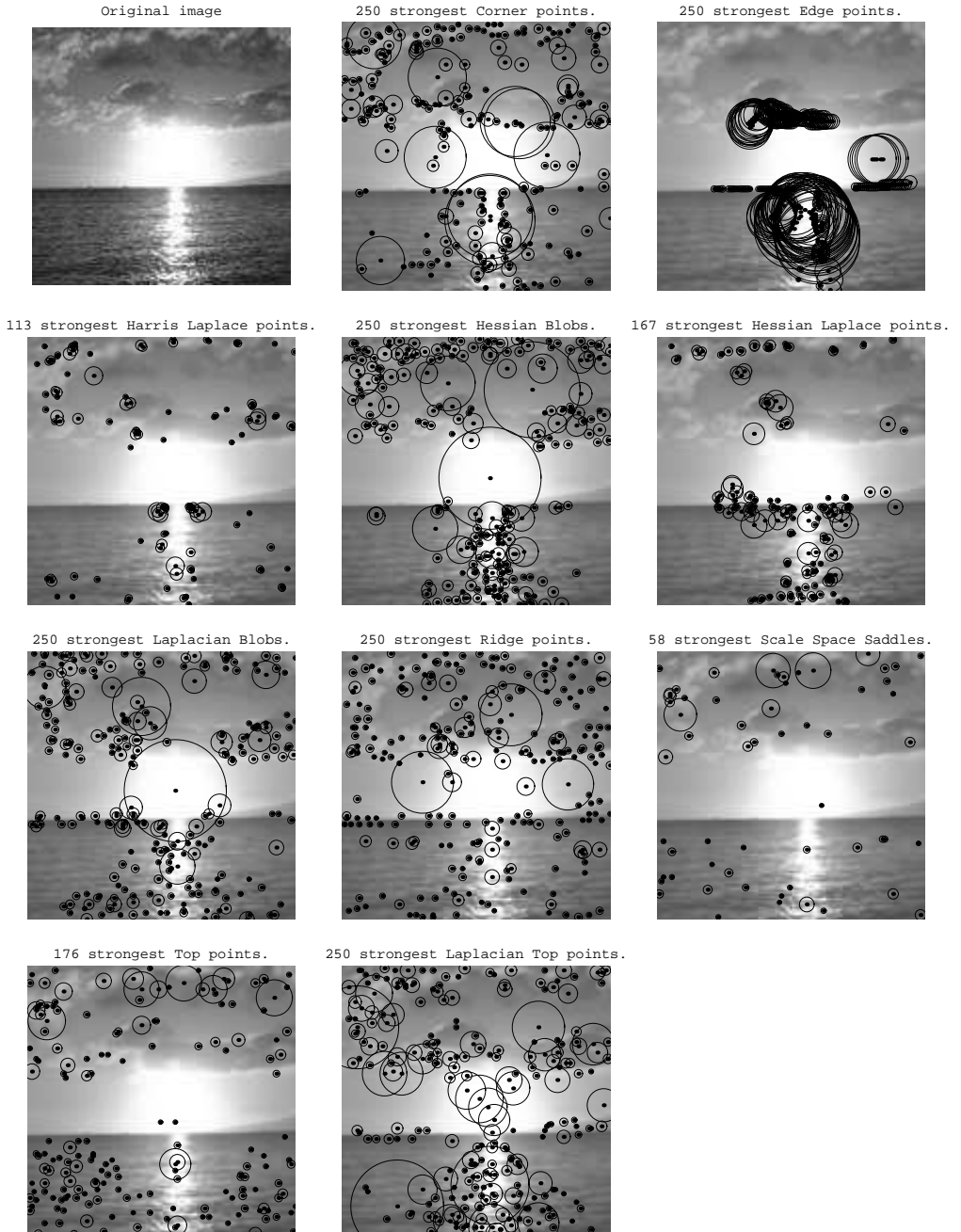


Figure C.6: Interest points of the *hawaii2.jpg* image projected on the original image. The size of the circles represent the scale of the interest points. Maximum 250 points of each set are shown for the sake of clarity.

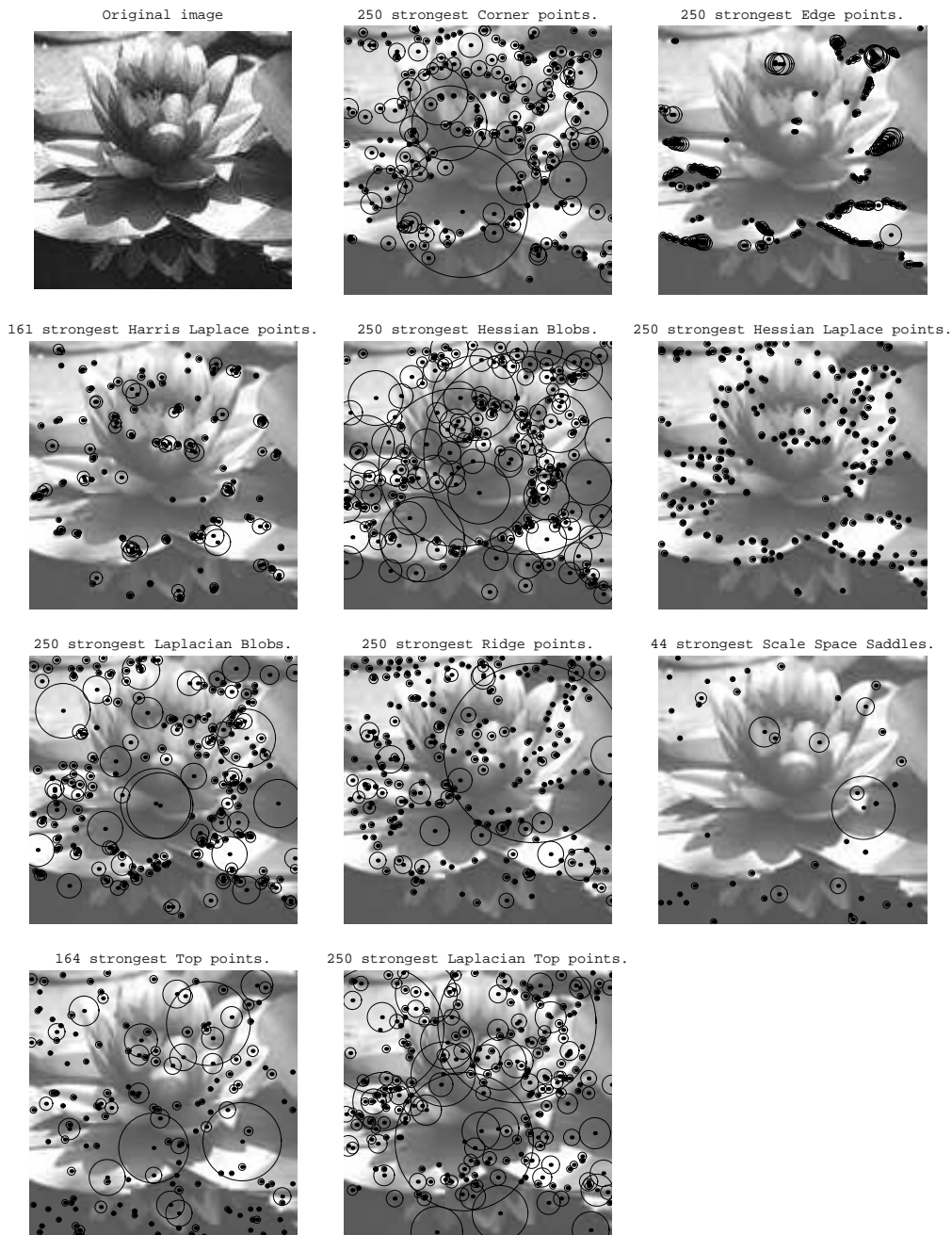


Figure C.7: Interest points of the *lilly.jpg* image projected on the original image. The size of the circles represent the scale of the interest points. Maximum 250 points of each set are shown for the sake of clarity.

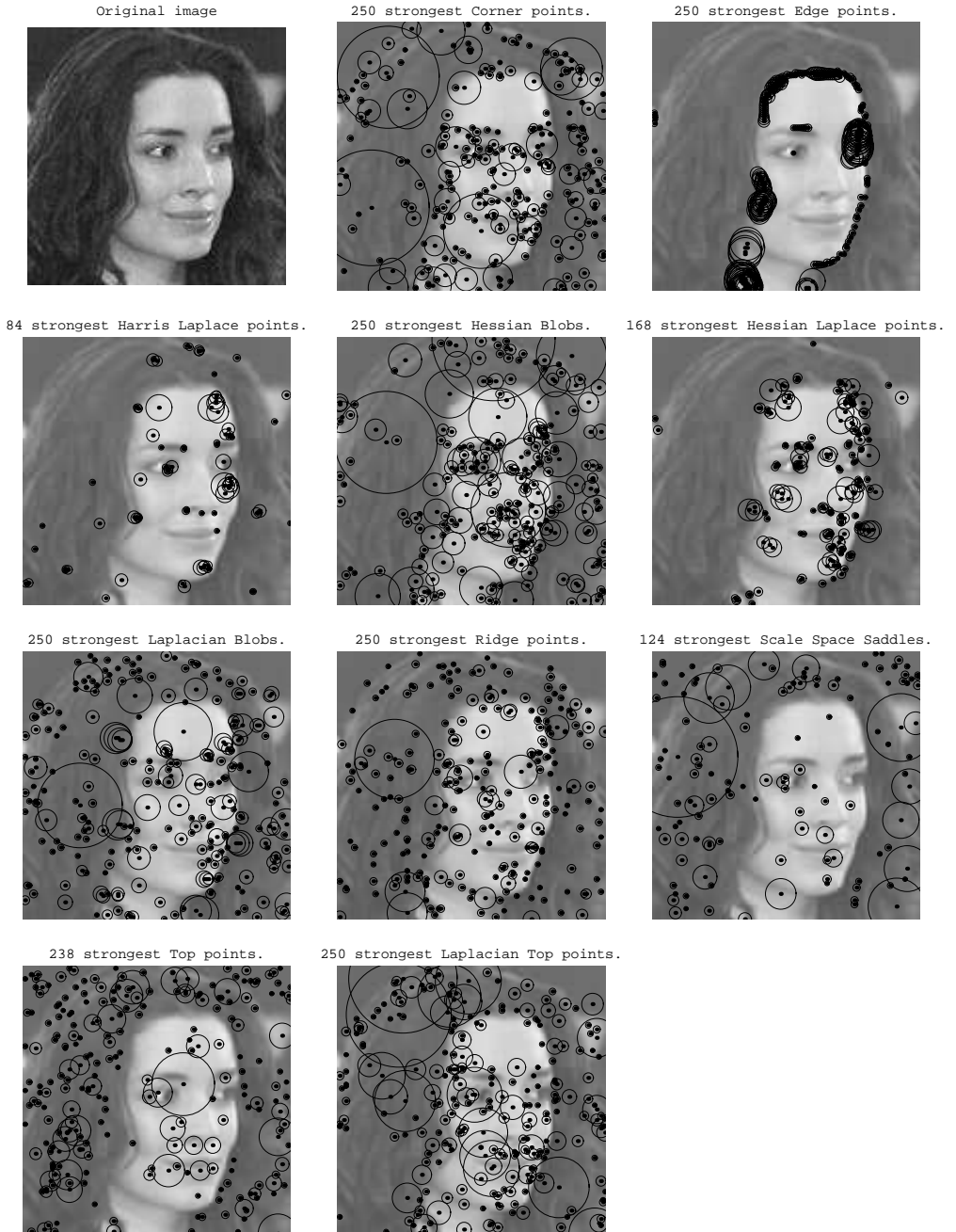


Figure C.8: Interest points of the *model.jpg* image projected on the original image. The size of the circles represent the scale of the interest points. Maximum 250 points of each set are shown for the sake of clarity.

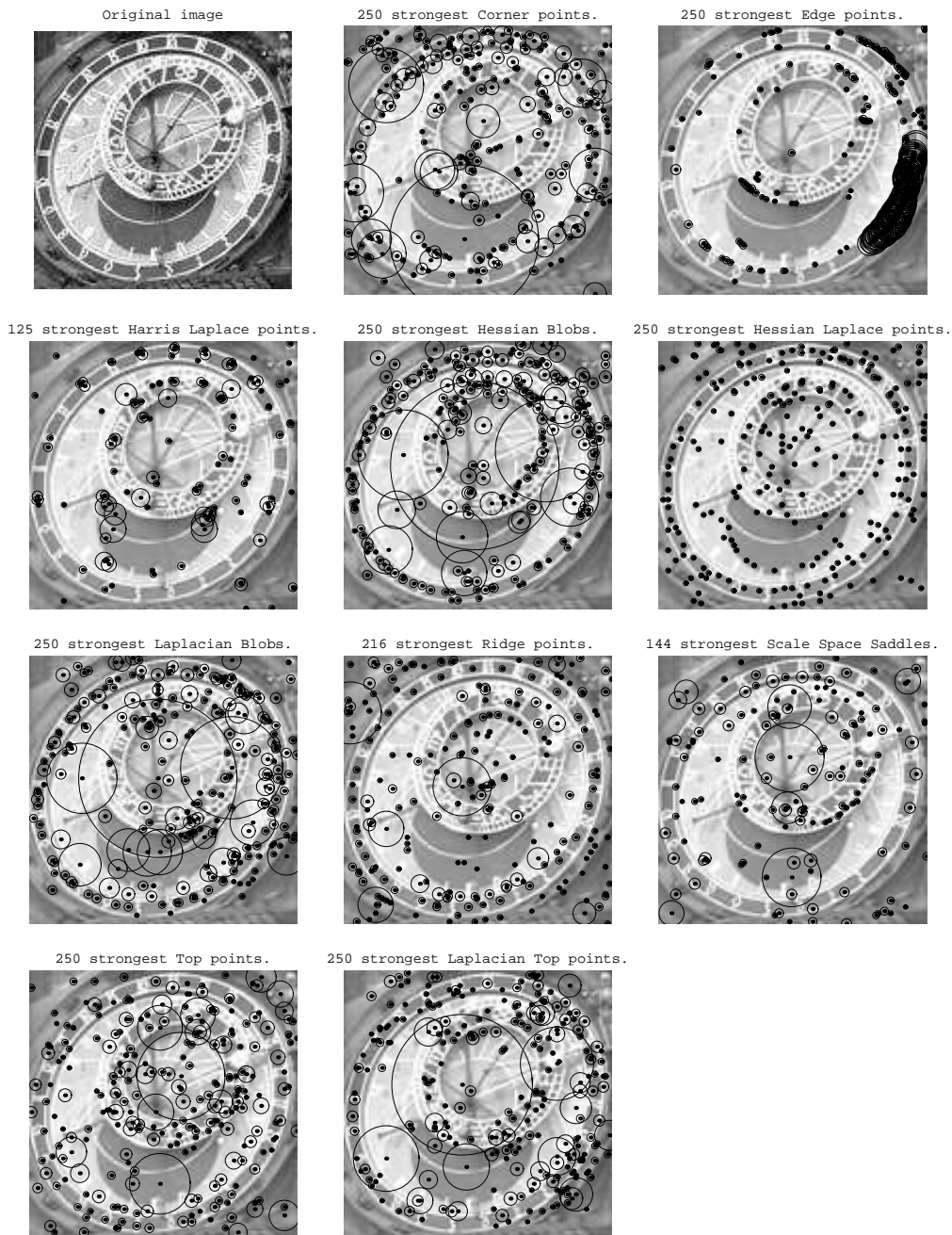


Figure C.9: Interest points of the *prague.jpg* image projected on the original image. The size of the circles represent the scale of the interest points. Maximum 250 points of each set are shown for the sake of clarity.

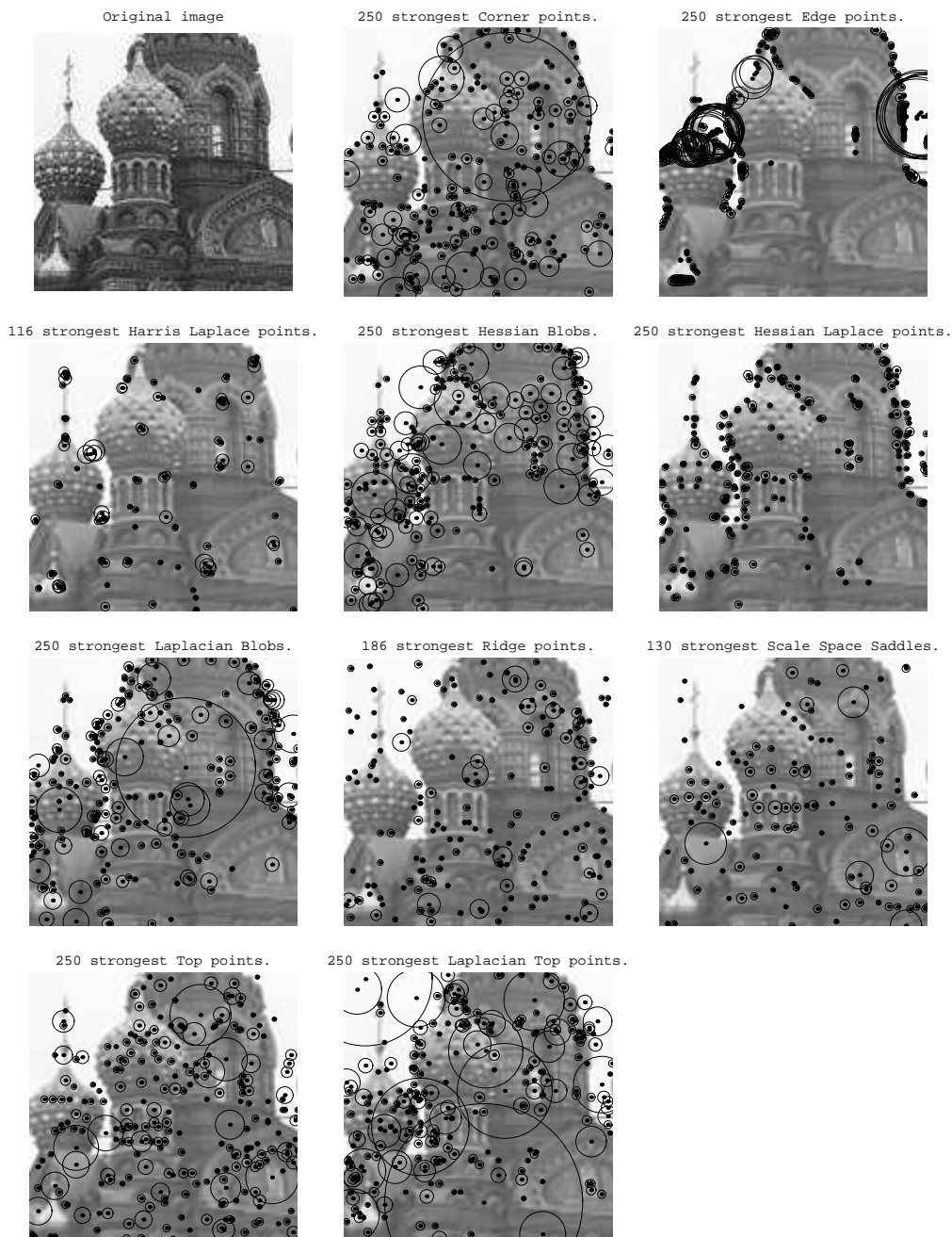


Figure C.10: Interest points of the *russia1.jpg* image projected on the original image. The size of the circles represent the scale of the interest points. Maximum 250 points of each set are shown for the sake of clarity.

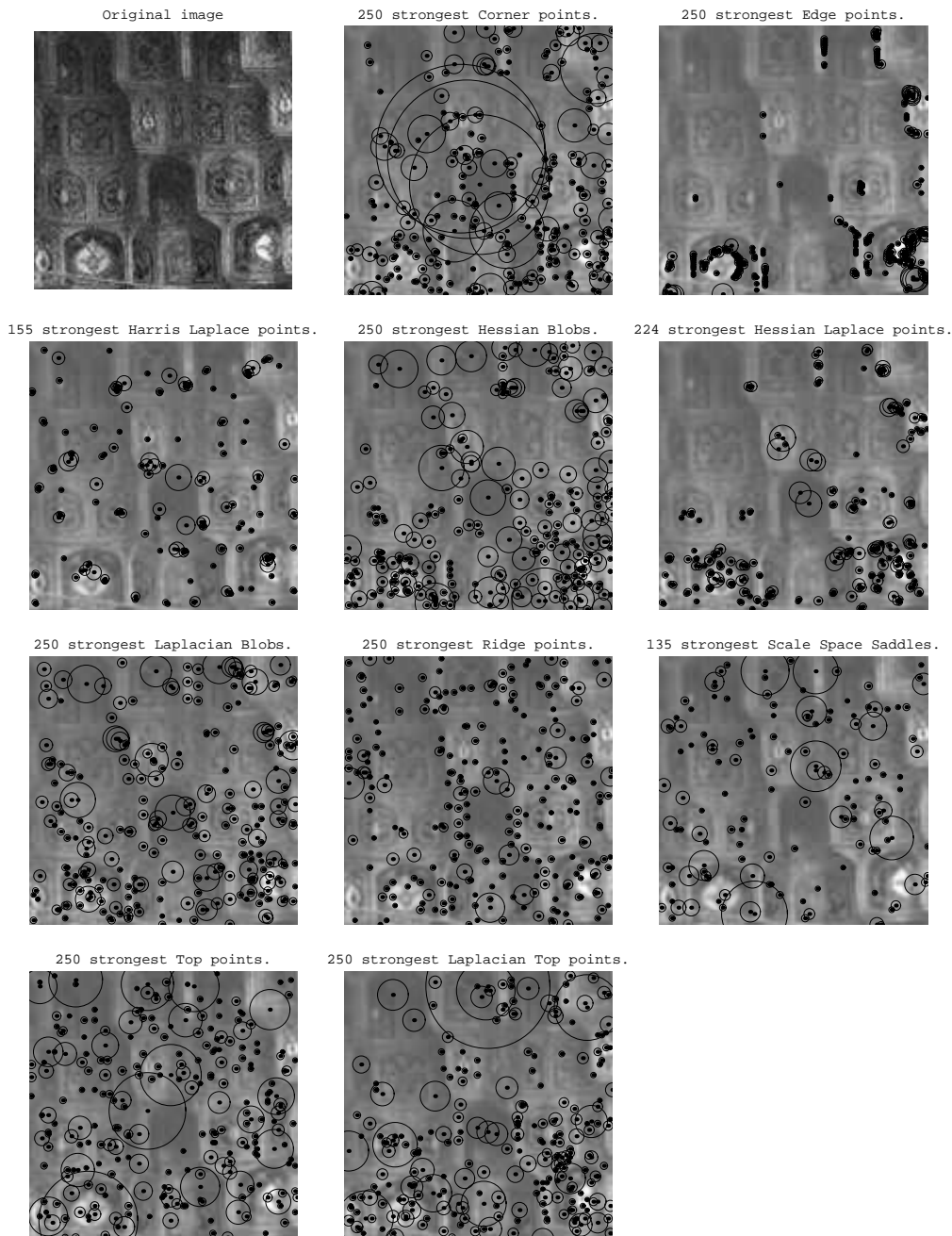


Figure C.11: Interest points of the *russia2.jpg* image projected on the original image. The size of the circles represent the scale of the interest points. Maximum 250 points of each set are shown for the sake of clarity.

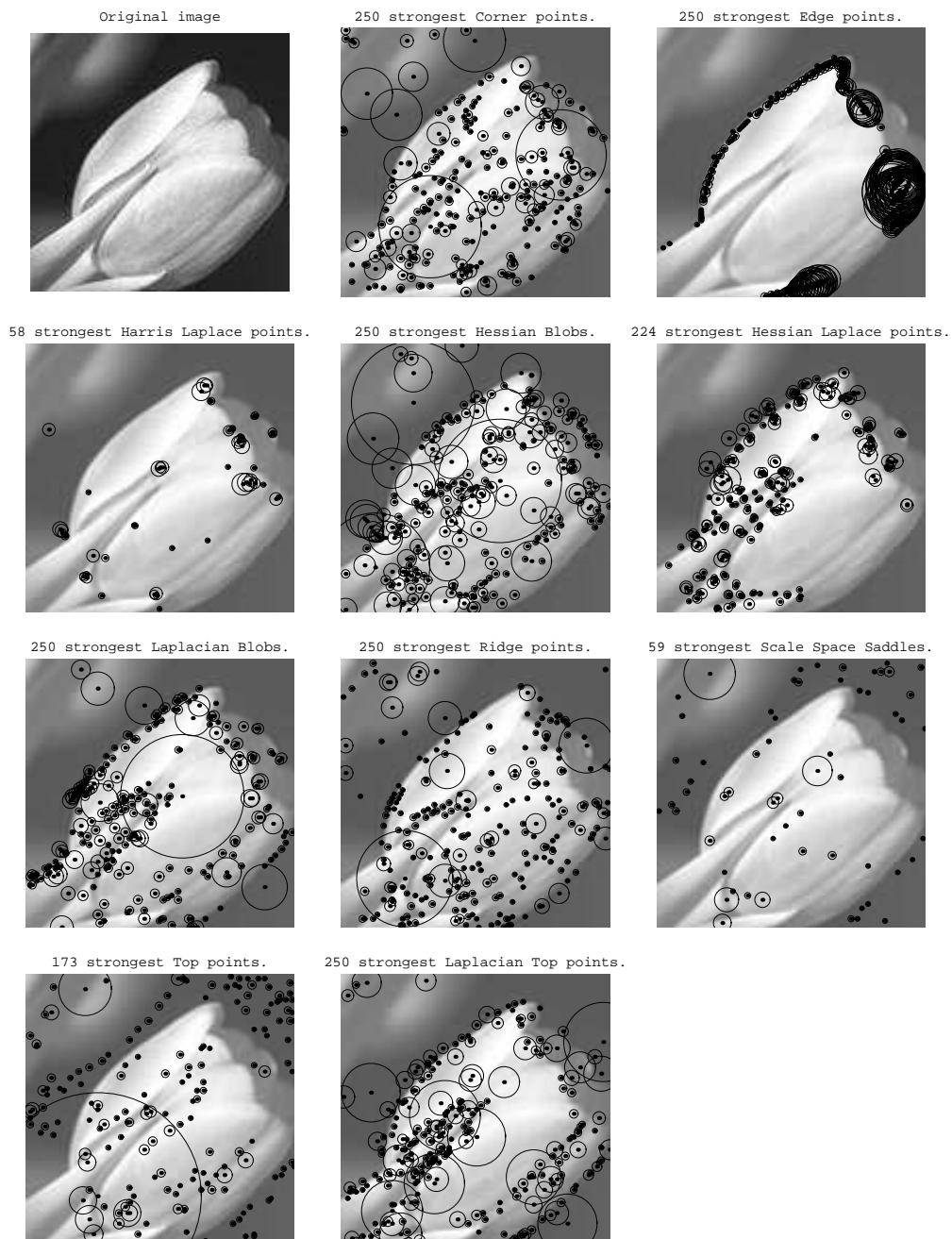


Figure C.12: Interest points of the *tulip.jpg* image projected on the original image. The size of the circles represent the scale of the interest points. Maximum 250 points of each set are shown for the sake of clarity.



Reconstructions from Interest Points of Test Images

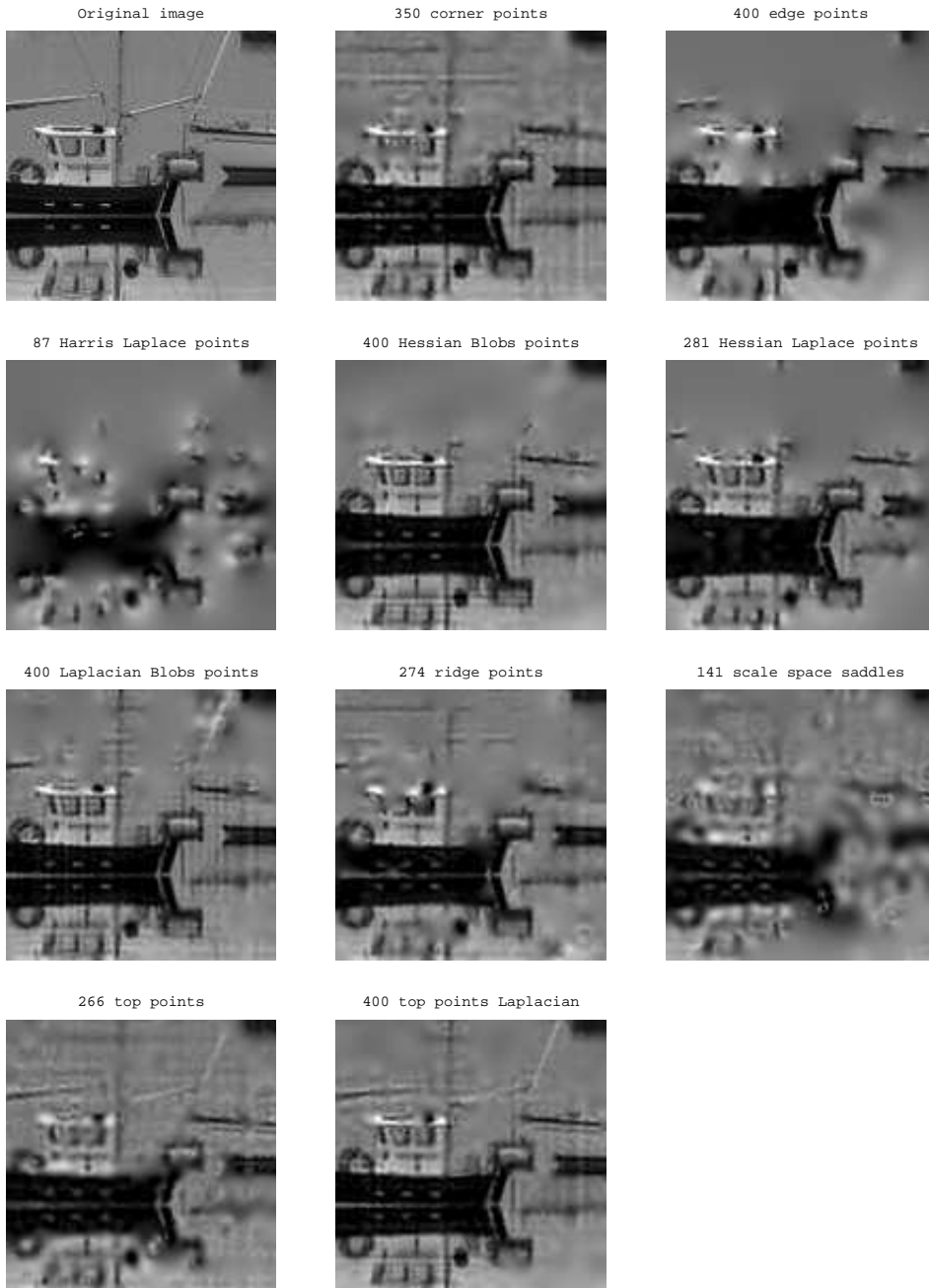


Figure D.1: Reconstructions of the *boat.jpg* image for 10 different types of scale space interest points. Maximum 400 points of each set are used for reconstruction.

Reconstruction from:	MSE	PM	MSPBD	MSDE	FMSDE	SSIM
Laplacian blobs (597)(400)	0.0111	0.7431	0.0018	0.0338	0.0135	76.3530
Hessian blobs (994)(400)	0.0103	0.7693	0.0018	0.0286	0.0124	77.9910
Ridge points (274)(274)	0.0164	0.5942	0.0029	0.0475	0.0168	69.4360
Corner points (350)(350)	0.0134	0.7005	0.0024	0.0410	0.0146	68.4570
Edge points (4563)(400)	0.0263	0.4819	0.0049	0.0543	0.0233	67.3380
Hessian Laplace points (281)(281)	0.0189	0.6464	0.0029	0.0385	0.0168	73.4610
Top points (266)(266)	0.0150	0.5792	0.0032	0.0410	0.0149	69.1210
Top points of the Laplacian (853)(400)	0.0094	0.7837	0.0017	0.0283	0.0115	78.9700
Scale space saddles (141)(141)	0.0276	0.3866	0.0066	0.0662	0.0239	57.2330
Harris Laplace points (87)(87)	0.0408	0.3432	0.0085	0.0727	0.0362	58.8890

Table D.1: Error measures of reconstructions from scale space interest points for the *boat.jpg* image using a maximum of 400 interest points per reconstruction. The first number behind the point type is the number of interest points detected, the second number shows the number of points used for the reconstruction. Note that the PM and the SSIM are similarity measures, while the others are error measures.



Figure D.2: Reconstructions of the *butterfly.jpg* image for 10 different types of scale space interest points. Maximum 400 points of each set are used for reconstruction.

Reconstruction from:	MSE	PM	MSPBD	MSDE	FMSDE	SSIM
Laplacian blobs (523)(400)	0.0088	0.7372	0.0021	0.0272	0.0130	82.3160
Hessian blobs (878)(400)	0.0083	0.7660	0.0020	0.0283	0.0132	80.7660
Ridge points (199)(199)	0.0296	0.3731	0.0072	0.0583	0.0283	64.0660
Corner points (404)(400)	0.0122	0.5931	0.0035	0.0334	0.0151	75.1050
Edge points (4225)(400)	0.0278	0.4445	0.0048	0.0591	0.0332	65.2900
Hessian Laplace points (300)(300)	0.0300	0.5318	0.0053	0.0546	0.0308	66.8880
Top points (239)(239)	0.0180	0.6246	0.0050	0.0441	0.0196	71.4070
Top points of the Laplacian (929)(400)	0.0080	0.7682	0.0021	0.0266	0.0121	82.2490
Scale space saddles (93)(93)	0.0563	0.2203	0.0117	0.0989	0.0415	50.1760
Harris Laplace points (110)(110)	0.0614	0.3924	0.0110	0.0858	0.0487	55.8290

Table D.2: Error measures of reconstructions from scale space interest points for the *butterfly.jpg* image using a maximum of 400 interest points per reconstruction. The first number behind the point type is the number of interest points detected, the second number shows the number of points used for the reconstruction. Note that the PM and the SSIM are similarity measures, while the others are error measures.

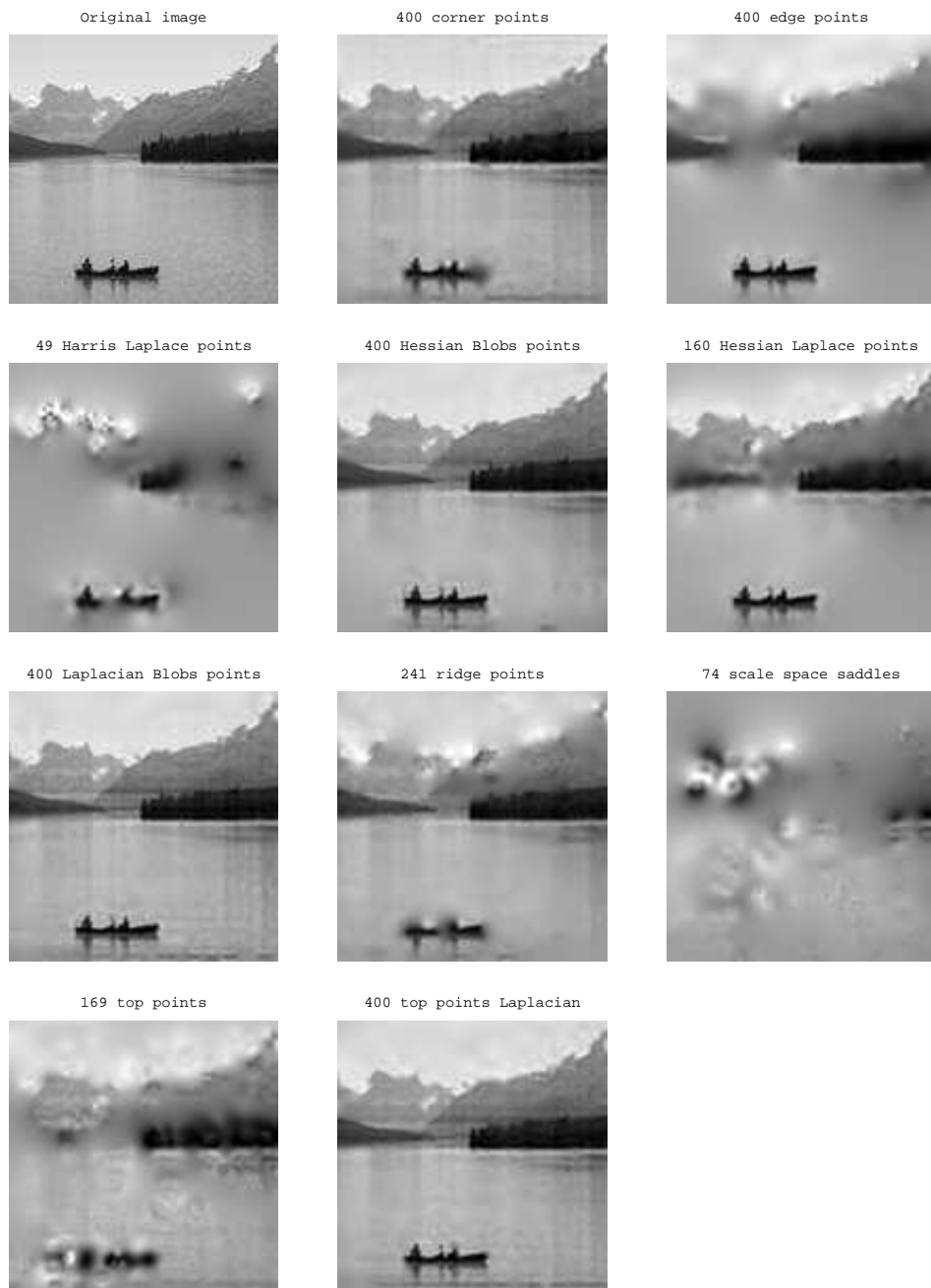


Figure D.3: Reconstructions of the *canada1.jpg* image for 10 different types of scale space interest points. Maximum 400 points of each set are used for reconstruction.

Reconstruction from:	MSE	PM	MSPBD	MSDE	FMSDE	SSIM
Laplacian blobs (536)(400)	0.0040	0.8118	0.0007	0.0228	0.0109	85.6280
Hessian blobs (933)(400)	0.0027	0.8761	0.0004	0.0189	0.0098	90.1250
Ridge points (241)(241)	0.0076	0.6513	0.0015	0.0327	0.0133	83.6950
Corner points (558)(400)	0.0051	0.7561	0.0009	0.0266	0.0107	84.3610
Edge points (3937)(400)	0.0111	0.4250	0.0012	0.0376	0.0177	81.2000
Hessian Laplace points (160)(160)	0.0079	0.6500	0.0012	0.0354	0.0168	84.1650
Top points (169)(169)	0.0176	0.4412	0.0026	0.0578	0.0216	71.6170
Top points of the Laplacian (867)(400)	0.0042	0.7668	0.0006	0.0234	0.0103	85.3610
Scale space saddles (74)(74)	0.1007	0.1527	0.0074	0.1129	0.0625	61.3770
Harris Laplace points (49)(49)	0.0567	0.3493	0.0060	0.0802	0.0459	70.3000

Table D.3: Error measures of reconstructions from scale space interest points for the *canada1.jpg* image using a maximum of 400 interest points per reconstruction. The first number behind the point type is the number of interest points detected, the second number shows the number of points used for the reconstruction. Note that the PM and the SSIM are similarity measures, while the others are error measures.

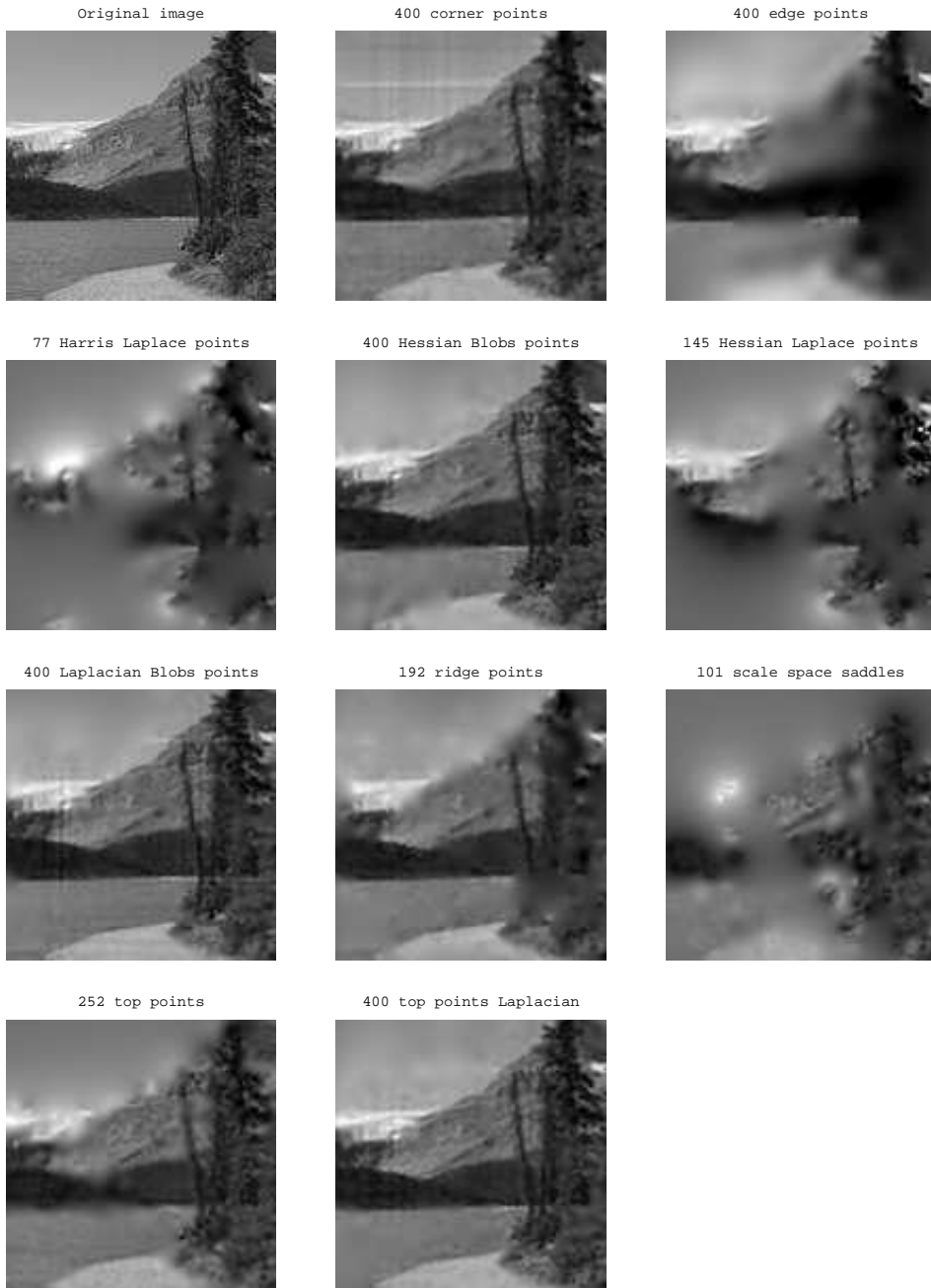


Figure D.4: Reconstructions of the *canada2.jpg* image for 10 different types of scale space interest points. Maximum 400 points of each set are used for reconstruction.

Reconstruction from:	MSE	PM	MSPBD	MSDE	FMSDE	SSIM
Laplacian blobs (554)(400)	0.0054	0.7957	0.0009	0.0187	0.0092	84.5200
Hessian blobs (1005)(400)	0.0049	0.8122	0.0010	0.0201	0.0091	84.6950
Ridge points (192)(192)	0.0108	0.4860	0.0025	0.0328	0.0149	74.5190
Corner points (440)(400)	0.0067	0.7651	0.0012	0.0251	0.0107	79.3320
Edge points (4260)(400)	0.0283	0.2407	0.0026	0.0547	0.0413	60.8650
Hessian Laplace points (145)(145)	0.0339	0.3573	0.0043	0.0583	0.0319	65.5800
Top points (252)(252)	0.0180	0.5511	0.0018	0.0329	0.0229	79.2960
Top points of the Laplacian (955)(400)	0.0048	0.7965	0.0010	0.0171	0.0073	85.2090
Scale space saddles (101)(101)	0.0485	0.3431	0.0045	0.0708	0.0444	61.8570
Harris Laplace points (77)(77)	0.0375	0.2673	0.0052	0.0728	0.0383	61.2010

Table D.4: Error measures of reconstructions from scale space interest points for the *canada2.jpg* image using a maximum of 400 interest points per reconstruction. The first number behind the point type is the number of interest points detected, the second number shows the number of points used for the reconstruction. Note that the PM and the SSIM are similarity measures, while the others are error measures.

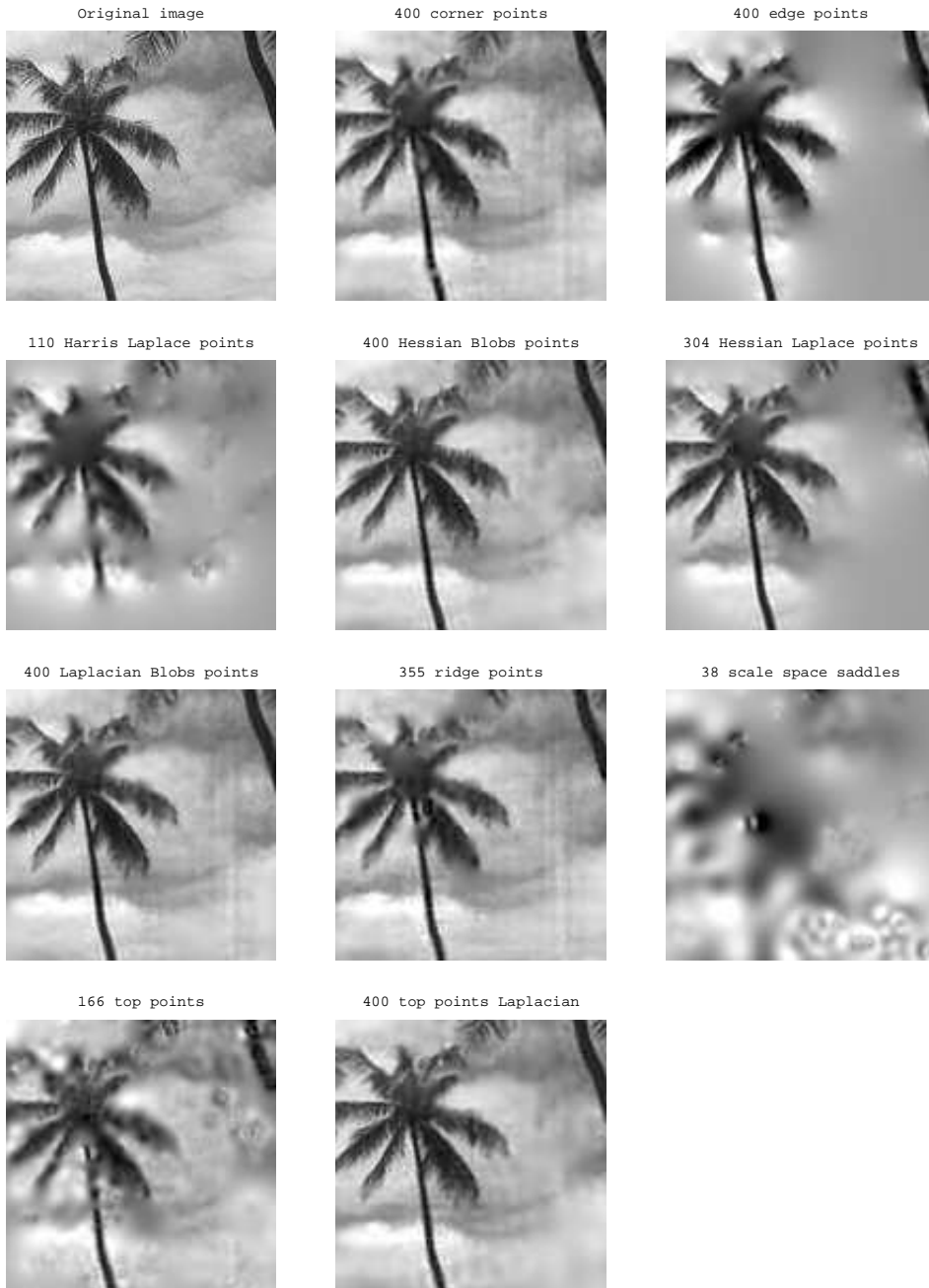


Figure D.5: Reconstructions of the *hawaii1.jpg* image for 10 different types of scale space interest points. Maximum 400 points of each set are used for reconstruction.

Reconstruction from:	MSE	PM	MSPBD	MSDE	FMSDE	SSIM
Laplacian blobs (521)(400)	0.0056	0.8255	0.0014	0.0227	0.0105	87.2130
Hessian blobs (907)(400)	0.0051	0.8370	0.0012	0.0210	0.0089	87.8450
Ridge points (355)(355)	0.0101	0.7183	0.0022	0.0326	0.0126	82.2030
Corner points (480)(400)	0.0079	0.7825	0.0015	0.0294	0.0133	85.7730
Edge points (5068)(400)	0.0364	0.5970	0.0051	0.0799	0.0439	72.0990
Hessian Laplace points (304)(304)	0.0237	0.6991	0.0034	0.0443	0.0288	77.3090
Top points (166)(166)	0.0173	0.6158	0.0039	0.0545	0.0190	72.6950
Top points of the Laplacian (793)(400)	0.0051	0.8697	0.0011	0.0211	0.0095	87.0820
Scale space saddles (38)(38)	0.0796	0.1482	0.0174	0.1312	0.0496	45.3280
Harris Laplace points (110)(110)	0.0393	0.4530	0.0090	0.0753	0.0398	67.3930

Table D.5: Error measures of reconstructions from scale space interest points for the *hawaii1.jpg* image using a maximum of 400 interest points per reconstruction. The first number behind the point type is the number of interest points detected, the second number shows the number of points used for the reconstruction. Note that the PM and the SSIM are similarity measures, while the others are error measures.

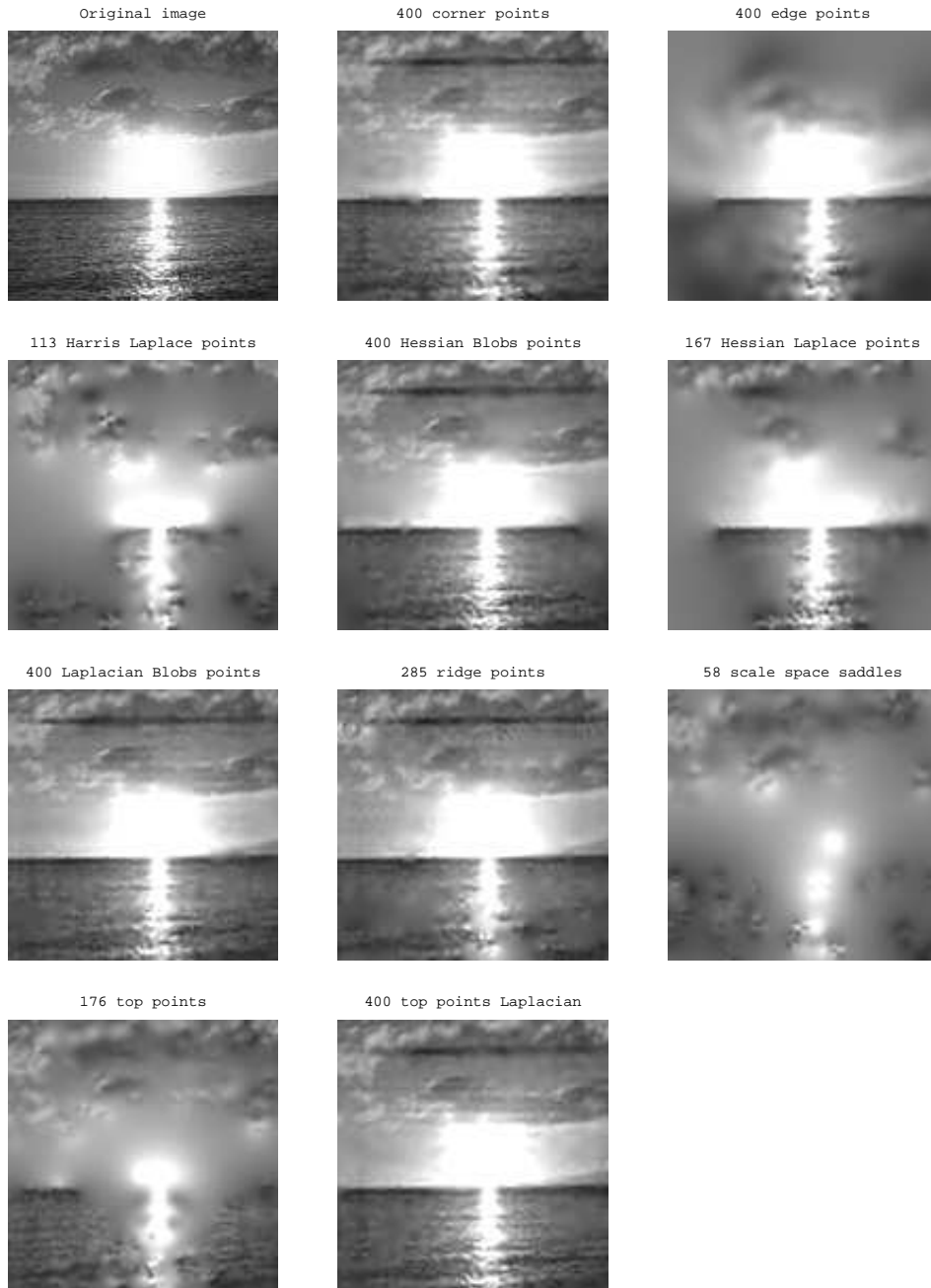


Figure D.6: Reconstructions of the *hawaii2.jpg* image for 10 different types of scale space interest points. Maximum 400 points of each set are used for reconstruction.

Reconstruction from:	MSE	PM	MSPBD	MSDE	FMSDE	SSIM
Laplacian blobs (494)(400)	0.0187	0.6312	0.0012	0.0282	0.0109	81.1880
Hessian blobs (915)(400)	0.0175	0.6805	0.0013	0.0287	0.0116	80.2410
Ridge points (285)(285)	0.0218	0.5134	0.0019	0.0364	0.0138	74.3190
Corner points (446)(400)	0.0190	0.6107	0.0014	0.0318	0.0122	78.8650
Edge points (4041)(400)	0.0255	0.3222	0.0035	0.0475	0.0206	63.8130
Hessian Laplace points (167)(167)	0.0408	0.4551	0.0022	0.0474	0.0294	67.9910
Top points (176)(176)	0.0466	0.4064	0.0050	0.0686	0.0344	68.7340
Top points of the Laplacian (819)(400)	0.0182	0.6767	0.0013	0.0291	0.0113	81.1940
Scale space saddles (58)(58)	0.0722	0.2332	0.0085	0.0877	0.0435	56.2540
Harris Laplace points (113)(113)	0.0401	0.4070	0.0038	0.0588	0.0262	65.8030

Table D.6: Error measures of reconstructions from scale space interest points for the *hawaii2.jpg* image using a maximum of 400 interest points per reconstruction. The first number behind the point type is the number of interest points detected, the second number shows the number of points used for the reconstruction. Note that the PM and the SSIM are similarity measures, while the others are error measures.

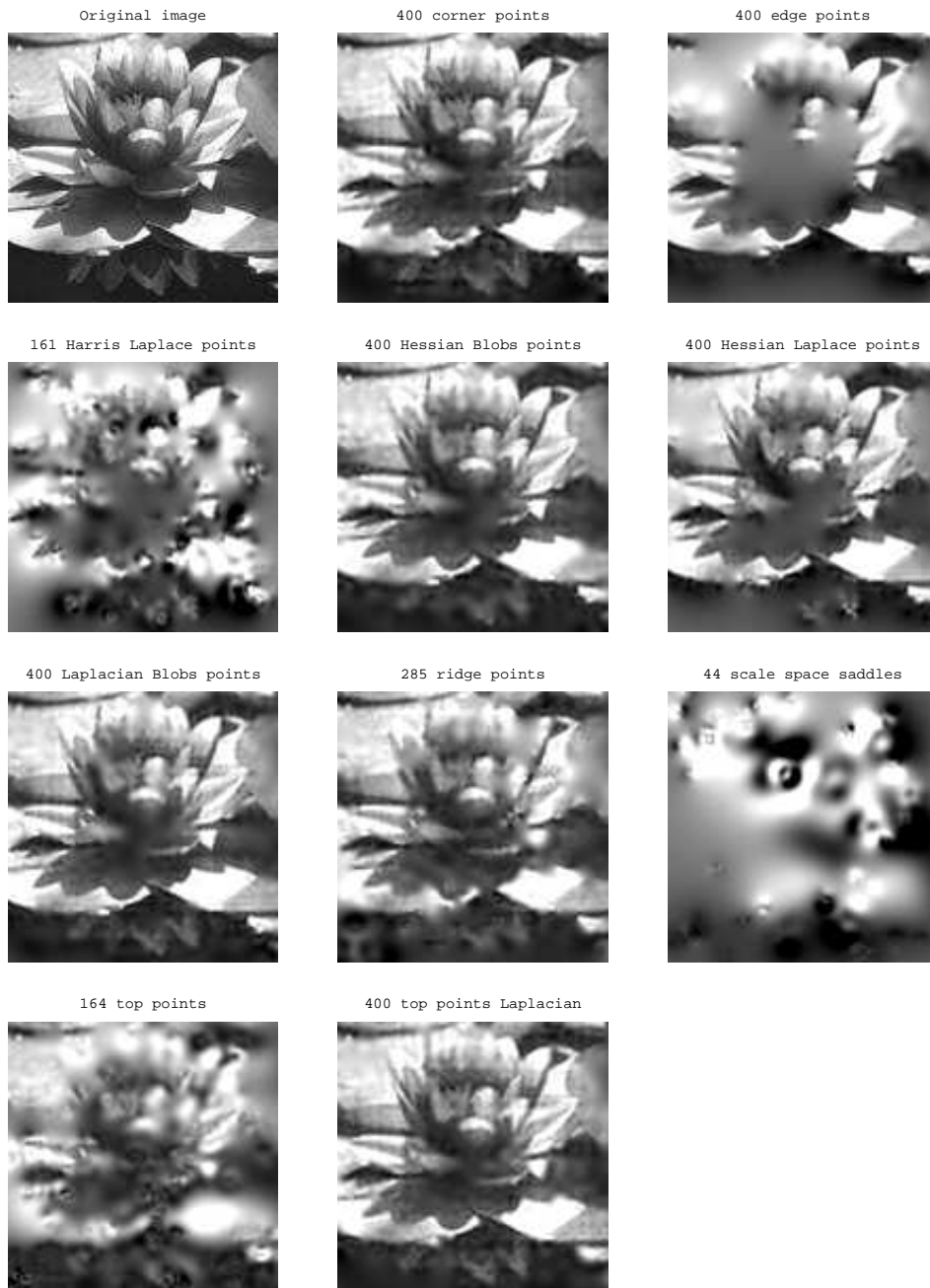


Figure D.7: Reconstructions of the *lilly.jpg* image for 10 different types of scale space interest points. Maximum 400 points of each set are used for reconstruction.

Reconstruction from:	MSE	PM	MSPBD	MSDE	FMSDE	SSIM
Laplacian blobs (606)(400)	0.0120	0.8082	0.0020	0.0389	0.0150	83.5630
Hessian blobs (886)(400)	0.0129	0.7865	0.0025	0.0373	0.0150	82.8010
Ridge points (285)(285)	0.0259	0.6905	0.0059	0.0607	0.0229	74.1730
Corner points (458)(400)	0.0150	0.8035	0.0029	0.0463	0.0186	81.3430
Edge points (4369)(400)	0.0508	0.4721	0.0094	0.0908	0.0397	62.5090
Hessian Laplace points (532)(400)	0.0258	0.7251	0.0041	0.0596	0.0317	75.2750
Top points (164)(164)	0.0423	0.4858	0.0134	0.0858	0.0286	63.2620
Top points of the Laplacian (750)(400)	0.0142	0.8050	0.0027	0.0390	0.0167	82.7400
Scale space saddles (44)(44)	0.3375	0.1469	0.0405	0.2391	0.0979	22.8830
Harris Laplace points (161)(161)	0.1024	0.4559	0.0226	0.1362	0.0577	48.4710

Table D.7: Error measures of reconstructions from scale space interest points for the *lilly.jpg* image using a maximum of 400 interest points per reconstruction. The first number behind the point type is the number of interest points detected, the second number shows the number of points used for the reconstruction. Note that the PM and the SSIM are similarity measures, while the others are error measures.

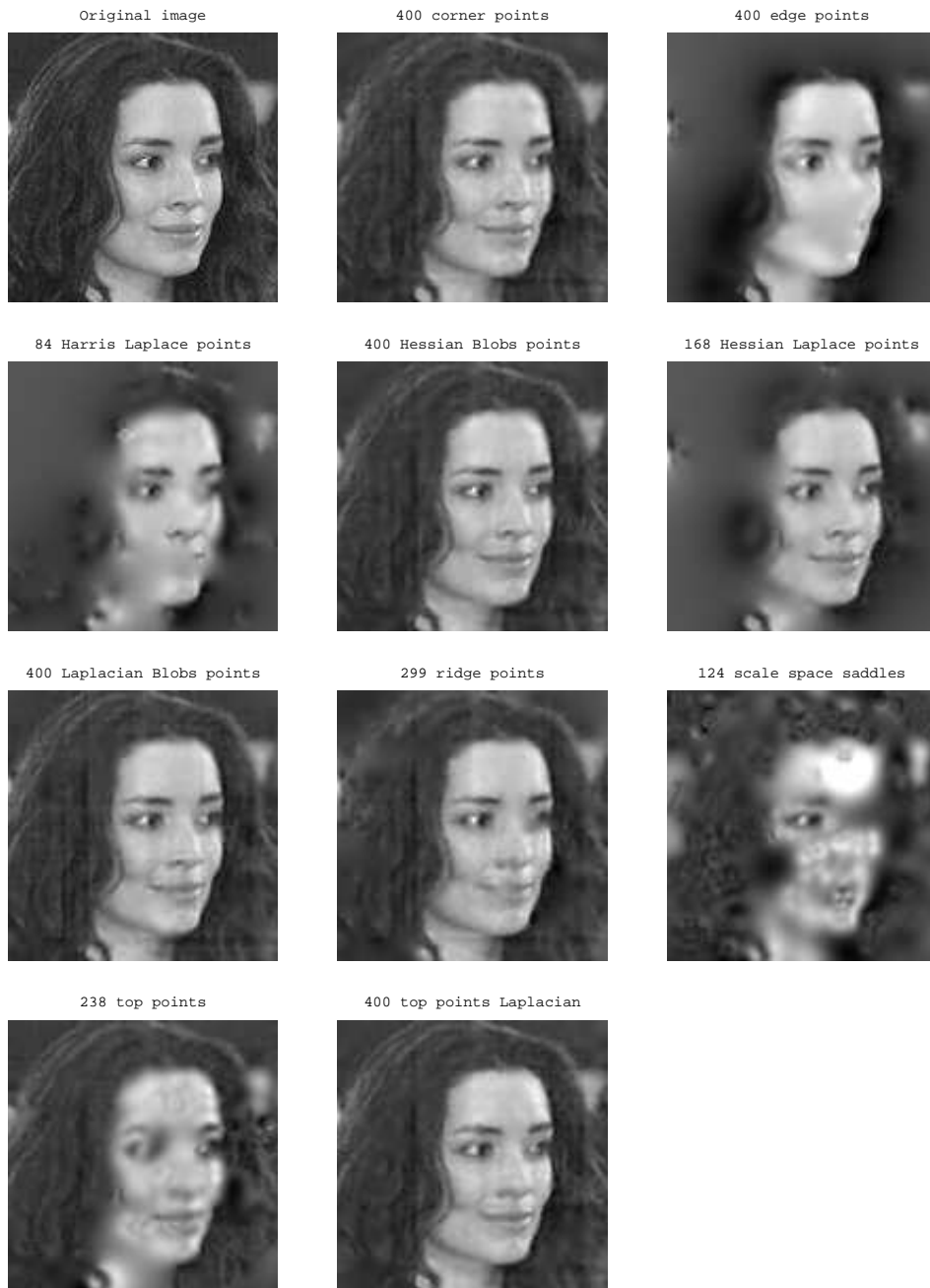


Figure D.8: Reconstructions of the *model.jpg* image for 10 different types of scale space interest points. Maximum 400 points of each set are used for reconstruction.

Reconstruction from:	MSE	PM	MSPBD	MSDE	FMSDE	SSIM
Laplacian blobs (479)(400)	0.0049	0.7524	0.0007	0.0216	0.0100	85.9570
Hessian blobs (859)(400)	0.0045	0.8140	0.0007	0.0201	0.0096	86.5980
Ridge points (299)(299)	0.0069	0.6975	0.0015	0.0291	0.0126	81.0600
Corner points (421)(400)	0.0049	0.8382	0.0008	0.0212	0.0099	86.4490
Edge points (4745)(400)	0.0218	0.4947	0.0032	0.0586	0.0323	67.4680
Hessian Laplace points (168)(168)	0.0221	0.6426	0.0021	0.0412	0.0249	70.6120
Top points (238)(238)	0.0107	0.5173	0.0017	0.0395	0.0169	78.0540
Top points of the Laplacian (915)(400)	0.0052	0.7878	0.0009	0.0223	0.0105	85.2000
Scale space saddles (124)(124)	0.0344	0.2769	0.0064	0.0788	0.0294	57.2580
Harris Laplace points (84)(84)	0.0212	0.4438	0.0031	0.0542	0.0256	67.7230

Table D.8: Error measures of reconstructions from scale space interest points for the *model.jpg* image using a maximum of 400 interest points per reconstruction. The first number behind the point type is the number of interest points detected, the second number shows the number of points used for the reconstruction. Note that the PM and the SSIM are similarity measures, while the others are error measures.

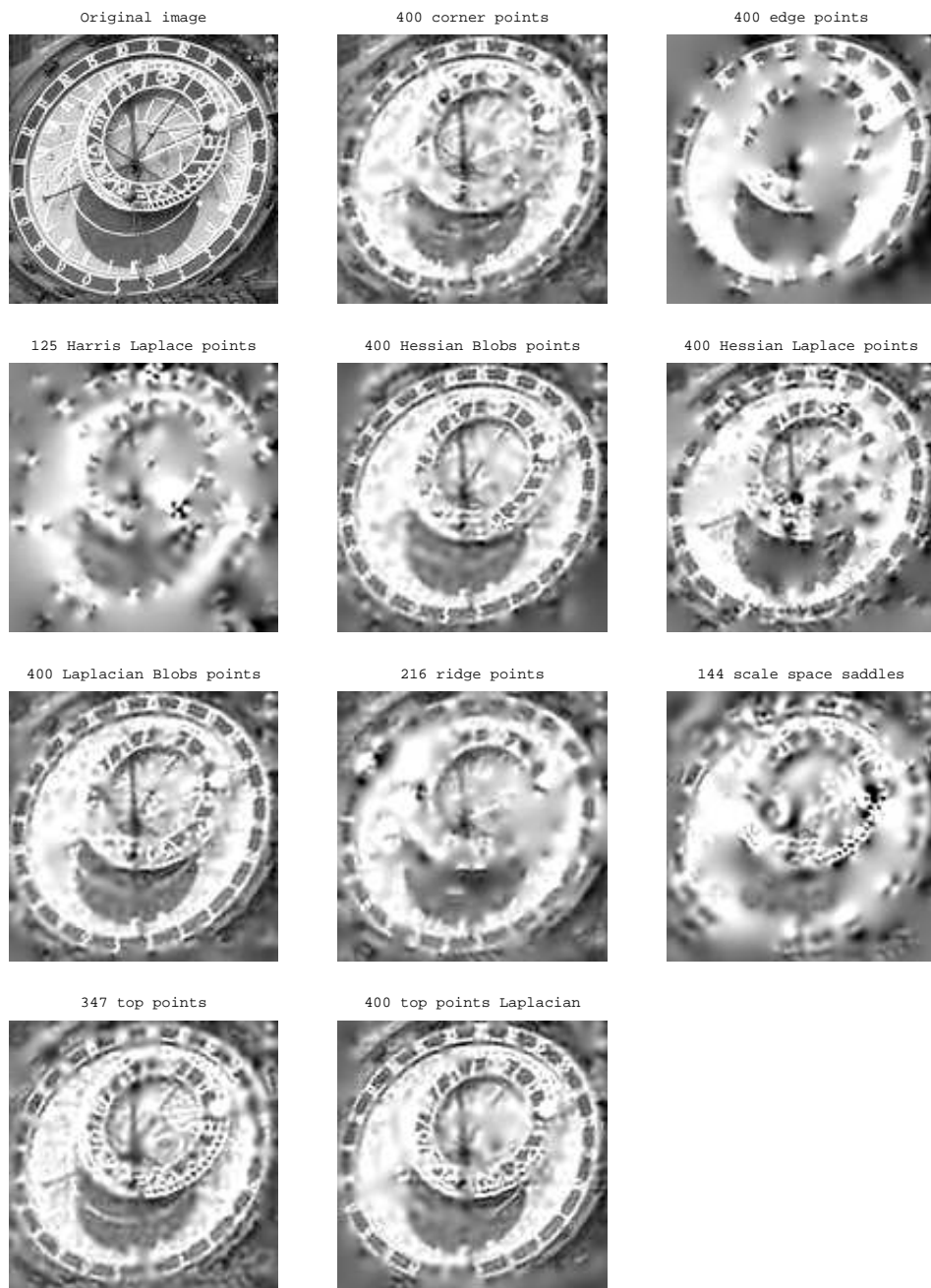


Figure D.9: Reconstructions of the *prague.jpg* image for 10 different types of scale space interest points. Maximum 400 points of each set are used for reconstruction.

Reconstruction from:	MSE	PM	MSPBD	MSDE	FMSDE	SSIM
Laplacian blobs (704)(400)	0.0905	0.6028	0.0166	0.0534	0.0220	60.0740
Hessian blobs (1204)(400)	0.0882	0.5520	0.0144	0.0549	0.0232	58.3590
Ridge points (216)(216)	0.1354	0.4006	0.0318	0.0983	0.0372	47.0680
Corner points (423)(400)	0.0964	0.5854	0.0171	0.0611	0.0244	60.1090
Edge points (5269)(400)	0.1288	0.3108	0.0233	0.1214	0.0534	40.2640
Hessian Laplace points (677)(400)	0.1286	0.5154	0.0265	0.1065	0.0411	52.0940
Top points (347)(347)	0.0943	0.6478	0.0182	0.0558	0.0234	63.4130
Top points of the Laplacian (1074)(400)	0.0862	0.5738	0.0173	0.0550	0.0222	60.1900
Scale space saddles (144)(144)	0.1458	0.3411	0.0356	0.1187	0.0458	40.9220
Harris Laplace points (125)(125)	0.1736	0.2271	0.0378	0.1273	0.0505	32.1460

Table D.9: Error measures of reconstructions from scale space interest points for the *prague.jpg* image using a maximum of 400 interest points per reconstruction. The first number behind the point type is the number of interest points detected, the second number shows the number of points used for the reconstruction. Note that the PM and the SSIM are similarity measures, while the others are error measures.



Figure D.10: Reconstructions of the *russia1.jpg* image for 10 different types of scale space interest points. Maximum 400 points of each set are used for reconstruction.

Reconstruction from:	MSE	PM	MSPBD	MSDE	FMSDE	SSIM
Laplacian blobs (616)(400)	0.0204	0.6602	0.0037	0.0341	0.0139	72.7150
Hessian blobs (1028)(400)	0.0203	0.6840	0.0031	0.0347	0.0147	70.5440
Ridge points (186)(186)	0.0477	0.3518	0.0096	0.0756	0.0285	51.8450
Corner points (350)(350)	0.0280	0.5020	0.0048	0.0441	0.0182	67.4210
Edge points (3973)(400)	0.0513	0.4058	0.0081	0.0726	0.0375	49.4990
Hessian Laplace points (348)(348)	0.0320	0.5099	0.0052	0.0522	0.0298	58.6630
Top points (317)(317)	0.0266	0.6565	0.0049	0.0441	0.0176	69.3920
Top points of the Laplacian (924)(400)	0.0215	0.6789	0.0039	0.0350	0.0143	71.4350
Scale space saddles (130)(130)	0.1069	0.3584	0.0169	0.1260	0.0594	43.1810
Harris Laplace points (116)(116)	0.0753	0.3077	0.0101	0.0887	0.0455	46.6790

Table D.10: Error measures of reconstructions from scale space interest points for the *russia1.jpg* image using a maximum of 400 interest points per reconstruction. The first number behind the point type is the number of interest points detected, the second number shows the number of points used for the reconstruction. Note that the PM and the SSIM are similarity measures, while the others are error measures.



Figure D.11: Reconstructions of the *russia2.jpg* image for 10 different types of scale space interest points. Maximum 400 points of each set are used for reconstruction.

Reconstruction from:	MSE	PM	MSPBD	MSDE	FMSDE	SSIM
Laplacian blobs (636)(400)	0.0052	0.6871	0.0012	0.0206	0.0095	80.7700
Hessian blobs (1155)(400)	0.0044	0.7140	0.0011	0.0197	0.0092	80.2990
Ridge points (307)(307)	0.0091	0.5251	0.0017	0.0317	0.0139	74.1730
Corner points (532)(400)	0.0057	0.6685	0.0014	0.0236	0.0101	79.7480
Edge points (5431)(400)	0.0205	0.4313	0.0043	0.0609	0.0341	58.4070
Hessian Laplace points (224)(224)	0.0130	0.4655	0.0029	0.0469	0.0231	61.6420
Top points (311)(311)	0.0072	0.4524	0.0016	0.0280	0.0117	78.0440
Top points of the Laplacian (885)(400)	0.0047	0.7629	0.0012	0.0210	0.0095	80.7620
Scale space saddles (135)(135)	0.0148	0.3735	0.0030	0.0502	0.0182	63.3430
Harris Laplace points (155)(155)	0.0212	0.3545	0.0041	0.0558	0.0244	60.6710

Table D.11: Error measures of reconstructions from scale space interest points for the *russia2.jpg* image using a maximum of 400 interest points per reconstruction. The first number behind the point type is the number of interest points detected, the second number shows the number of points used for the reconstruction. Note that the PM and the SSIM are similarity measures, while the others are error measures.

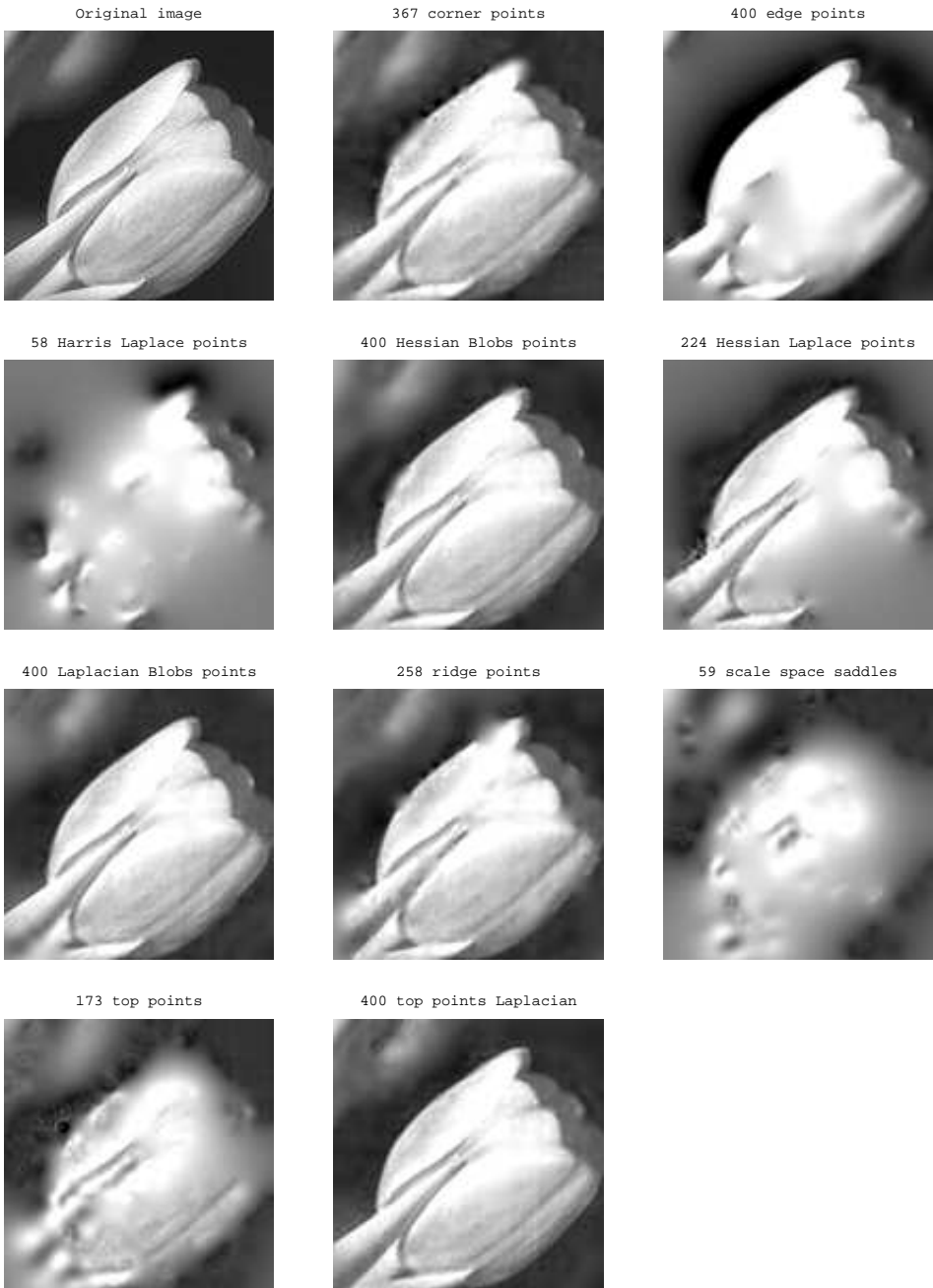


Figure D.12: Reconstructions of the *tulip.jpg* image for 10 different types of scale space interest points. Maximum 400 points of each set are used for reconstruction.

Reconstruction from:	MSE	PM	MSPBD	MSDE	FMSDE	SSIM
Laplacian blobs (437)(400)	0.0217	0.8017	0.0007	0.0289	0.0117	87.5410
Hessian blobs (727)(400)	0.0201	0.8033	0.0008	0.0295	0.0121	88.0380
Ridge points (258)(258)	0.0258	0.6694	0.0016	0.0433	0.0169	83.1960
Corner points (367)(367)	0.0223	0.7221	0.0011	0.0348	0.0127	84.6800
Edge points (3706)(400)	0.0628	0.3946	0.0059	0.1050	0.0512	64.8240
Hessian Laplace points (224)(224)	0.0678	0.6417	0.0028	0.0683	0.0400	71.8790
Top points (173)(173)	0.0484	0.4426	0.0047	0.0786	0.0288	69.6960
Top points of the Laplacian (896)(400)	0.0192	0.8186	0.0008	0.0276	0.0120	88.1560
Scale space saddles (59)(59)	0.0771	0.2064	0.0114	0.1203	0.0481	55.5280
Harris Laplace points (58)(58)	0.1340	0.2515	0.0116	0.1148	0.0682	54.3420

Table D.12: Error measures of reconstructions from scale space interest points for the *tulip.jpg* image using a maximum of 400 interest points per reconstruction. The first number behind the point type is the number of interest points detected, the second number shows the number of points used for the reconstruction. Note that the PM and the SSIM are similarity measures, while the others are error measures.



Stability Results of Interest Points from Test
Images

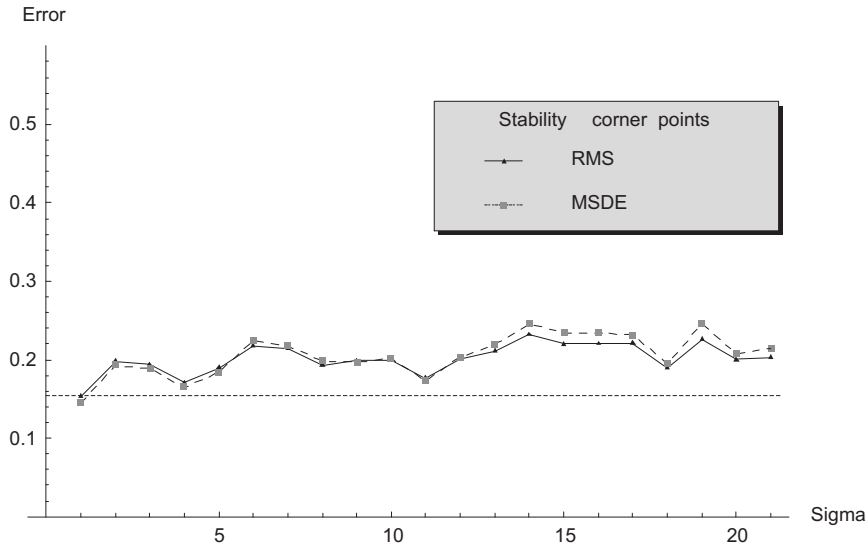


Figure E.1: RMS and MSDE error of reconstructions from corner points with noise on the position of the interest points with various standard deviation σ . The mean error is taken over the 12 images of the second test set. The dashed line is the mean RMS error of the unperturbed point set.

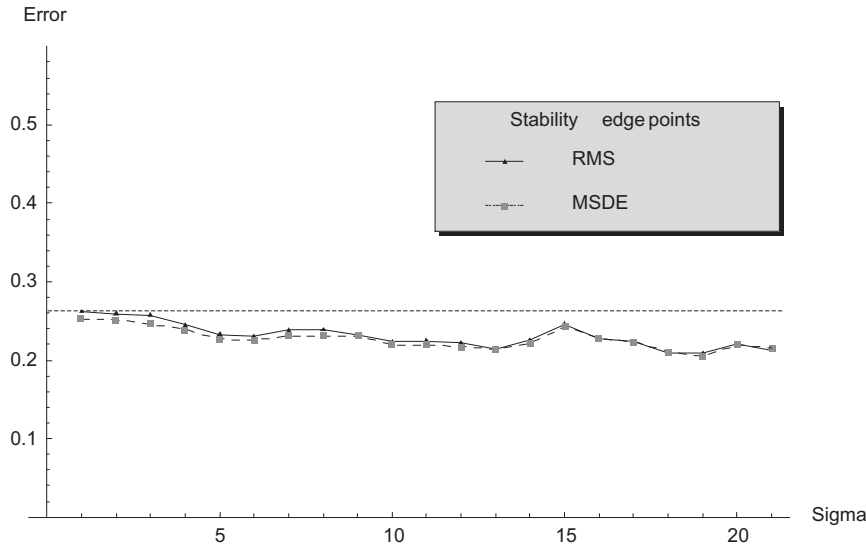


Figure E.2: RMS and MSDE error of reconstructions from edge points with noise on the position of the interest points with various standard deviation σ . The mean error is taken over the 12 images of the second test set. The dashed line is the mean RMS error of the unperturbed point set.

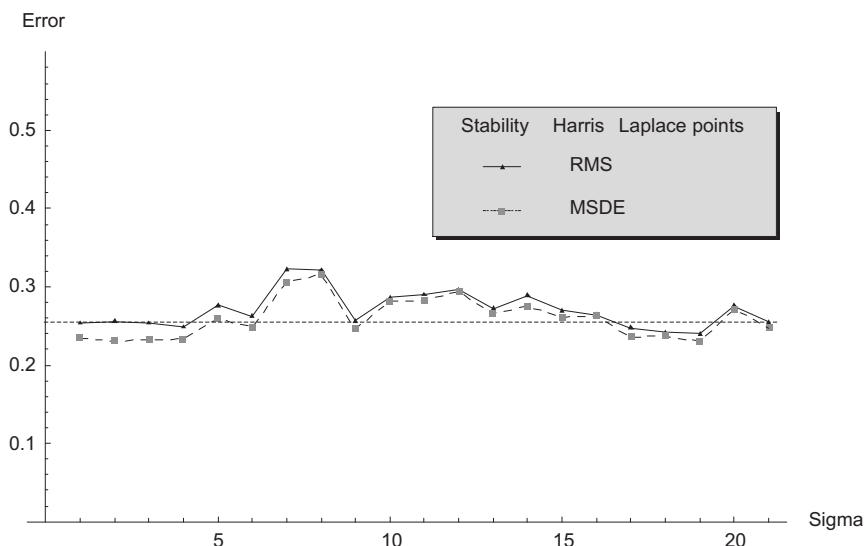


Figure E.3: RMS and MSDE error of reconstructions from Harris Laplace points with noise on the position of the interest points with various standard deviation σ . The mean error is taken over the 12 images of the second test set. The dashed line is the mean RMS error of the unperturbed point set.

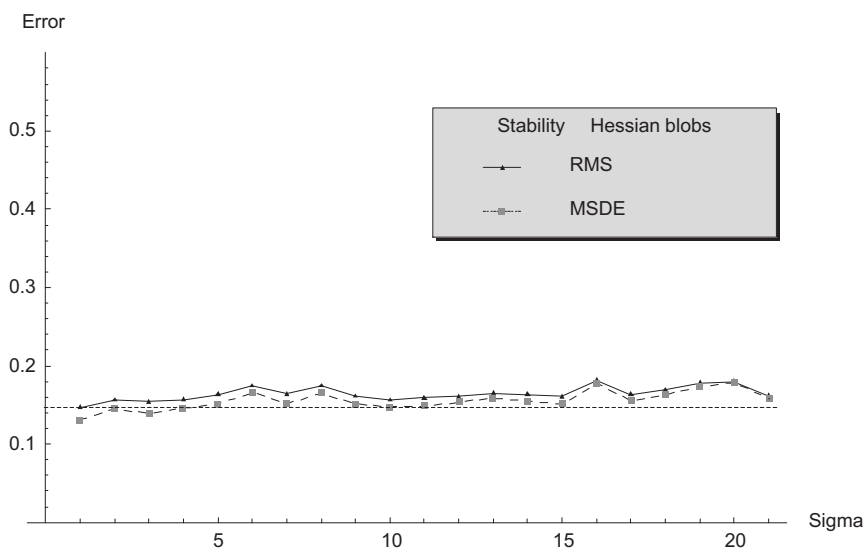


Figure E.4: RMS and MSDE error of reconstructions from Hessian blob points with noise on the position of the interest points with various standard deviation σ . The mean error is taken over the 12 images of the second test set. The dashed line is the mean RMS error of the unperturbed point set.

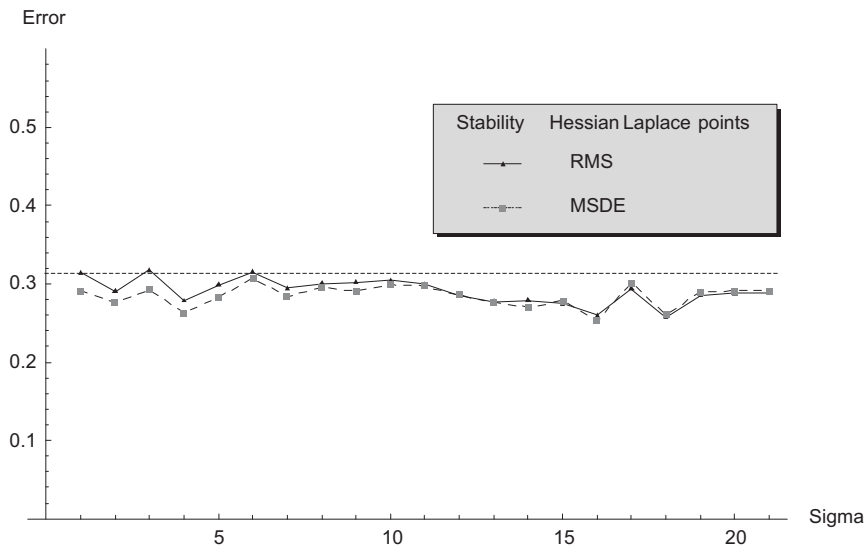


Figure E.5: RMS and MSDE error of reconstructions from Hessian Laplace points with noise on the position of the interest points with various standard deviation σ . The mean error is taken over the 12 images of the second test set. The dashed line is the mean RMS error of the unperturbed point set.

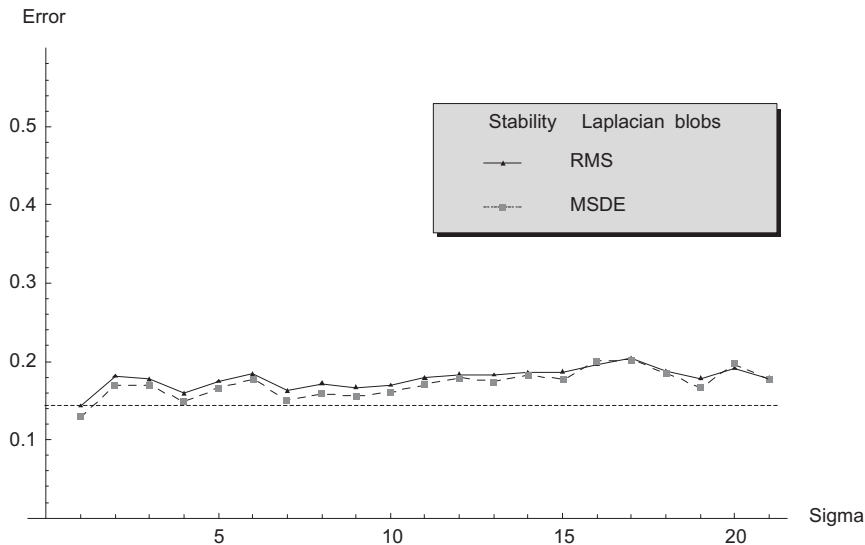


Figure E.6: RMS and MSDE error of reconstructions from Laplacian blob points with noise on the position of the interest points with various standard deviation σ . The mean error is taken over the 12 images of the second test set. The dashed line is the mean RMS error of the unperturbed point set.

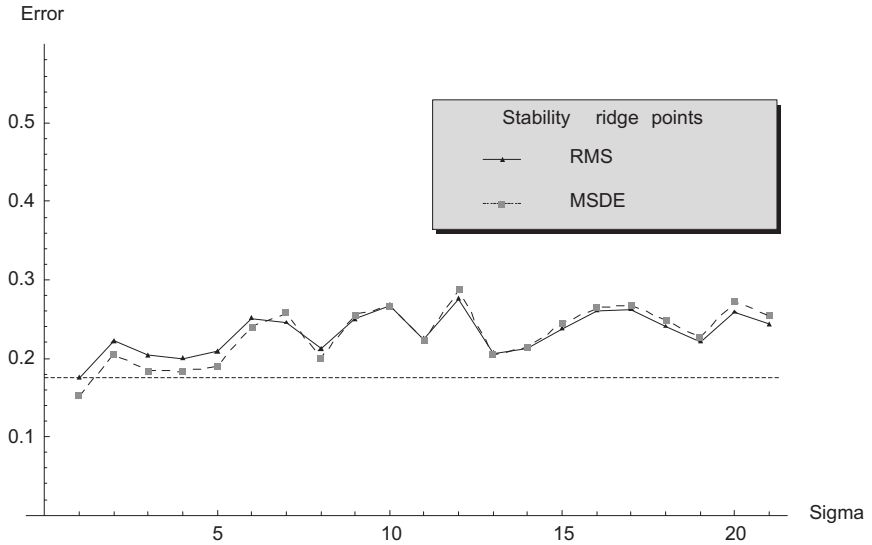


Figure E.7: RMS and MSDE error of reconstructions from ridge points with noise on the position of the interest points with various standard deviation σ . The mean error is taken over the 12 images of the second test set. The dashed line is the mean RMS error of the unperturbed point set.

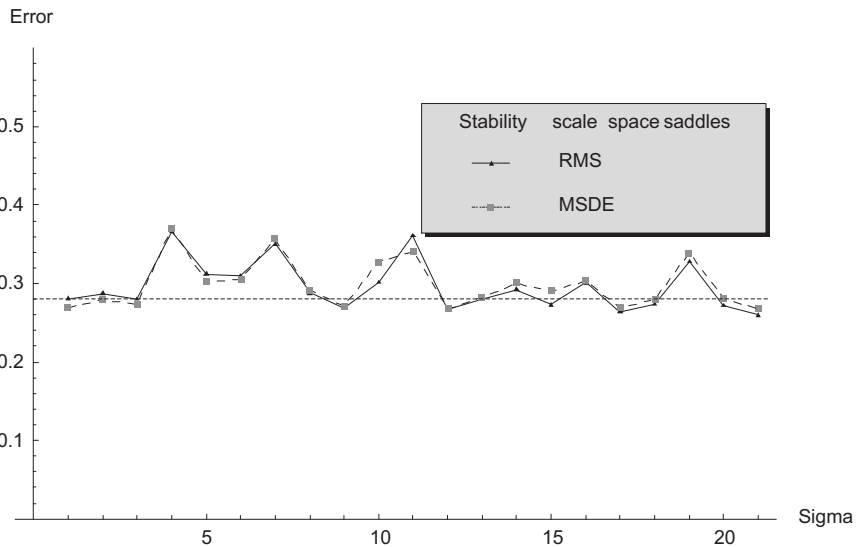


Figure E.8: RMS and MSDE error of reconstructions from scale space saddle points with noise on the position of the interest points with various standard deviation σ . The mean error is taken over the 12 images of the second test set. The dashed line is the mean RMS error of the unperturbed point set.

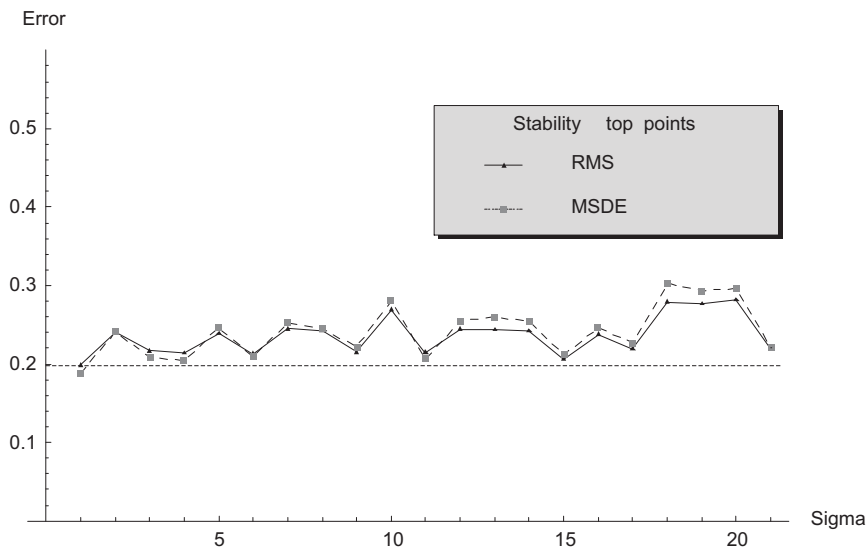


Figure E.9: RMS and MSDE error of reconstructions from top points with noise on the position of the interest points with various standard deviation σ . The mean error is taken over the 12 images of the second test set. The dashed line is the mean RMS error of the unperturbed point set.

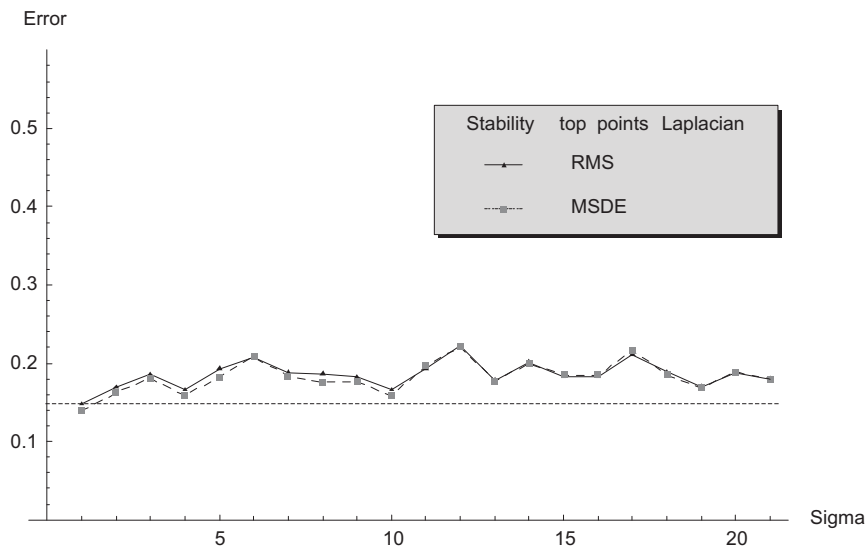


Figure E.10: RMS and MSDE error of reconstructions from top points of the Laplacian with noise on the position of the interest points with various standard deviation σ . The mean error is taken over the 12 images of the second test set. The dashed line is the mean RMS error of the unperturbed point set.

Bibliography

- [1] V. R. Avadhanam, N.; Algazi. Evaluation of a human vision system based image fidelity metric for image. In *Proceedings of the SPIE The International Society for Optical Engineering*, volume 3808, pages 569–579, 1999.
- [2] I. Avcibas, B. Sankur, and K. Sayood. Statistical evaluation of image quality measures. *Journal of Electronic Imaging.*, 11(2):206–223, April 2002.
- [3] E. Balmachnova, L. M. J. Florack, B. Platel, F. M. W. Kanters, and B. M. ter Haar Romeny. Stability of top-points in scale space. In Kimmel et al. [78], pages 62–72.
- [4] P. G. J. Barten. Evaluation of subjective image quality with the square root integral method. *Journal of the Optical Society of America A Optics and Image Science.*, 7(10):2024–2031, Oct. 1990.
- [5] A. Beghdadi. Design of an image distortion measure using spatial/spatial frequency analysis. In *First International Symposium on Control, Communications and Signal Processing.*, volume 1, pages 29–32, 2004.
- [6] M. Bernas. Image quality evaluation. In *Proceedings VIPromCom 2002. 4th EURASIP IEEE Region 8 International Symposium on Video/Image Processing and Multimedia Communications*, volume 02EX553, pages 133–136, 2002.
- [7] B.J.Janssen and R. Duits. Linear image reconstruction by sobolev norms on the bounded domain. Submitted to Scale Space 2007.
- [8] J. Blom. *Topological and Geometrical Aspects of Image Structure*. PhD thesis, University of Utrecht, Department of Medical and Physiological Physics, Utrecht, The Netherlands, 1992.
- [9] Dawson B.M. Changing perceptions of reality. *Byte*, 14(13):293–304, December 1989.
- [10] F. Bock, A. Steudel, H. Clausert, and M. Glesner. A fuzzy rule-based system for the subjective quality evaluation of distorted images. In *Sixth International Conference on Image Processing and its Applications*, volume 1, pages 448–452, 1997.
- [11] M.H Brill, J. Lubin, and D. Wolin. Perceptual scaling of quality metrics for hard-copy image evaluation. In *IS&T's NIP 15: International Conference on Digital Printing Technologies*, volume 1, pages 435–438, 1999.
- [12] T. Brox and J. Weickert. A tv flow based local scale measure for texture discrimination. In T. Pajdla and J. Matas, editors, *Proc. 8th European Conference on Computer Vision, Prague, Czech Republic.*, volume 2 of *Computer Vision - ECCV*, pages 578–590. Springer LNCS 3022, May 2004.
- [13] K. Brunnstrom, J. Eklundh, and T. Lindeberg. Active detection and classification of junctions by foveation with a head-eye system guided by the scale-space primal sketch, tech. rep., isrn kth/na/p-91/31-se, royal inst. tech., s-100 44 stockholm., 1991.
- [14] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, June. 1986.

- [15] D. Carevic and T. Caelli. Region based coding of color images using klt. *Graphical Models and Image Processing*, 59(1):27–38, Jan. 1997.
- [16] M. Carnec. Quality assessor v2.0. Website, <http://membres.lycos.fr/dcapplications/main.php>.
- [17] M. Carnec, P. Le Callet, and D. Barba. An image quality assessment method based on perception of structural information. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 2, pages III–185–188, 2003.
- [18] A. Chalmers, S. Daly, A. McNamare, K. Myszkowski, and T. Troscianko. Image quality metrics. Cours notes, Course 44 at Siggraph 2000, July 2000.
- [19] Junli Li; Gang Chen; Zheru Chi. A fuzzy metric for image quality assessment. In *10th IEEE International Conference on Fuzzy Systems*, volume 3, pages 562–565, 2001.
- [20] Junli Li; Gang Chen; Zheru Chi. A fuzzy image metric with application to fractal coding. *IEEE Transactions on Image Processing*, 11(6):636–643, June 2002.
- [21] C. S. Chin, F. Z. C.; Xydeas. Dual-mode image quality assessment metric. In *Proceedings VIPromCom 2002. 4th EURASIP IEEE Region 8 International Symposium on Video/Image Processing and Multimedia Communications*, volume 1, pages 137–140, 2002.
- [22] Scott Daly. The visible differences predictor: an algorithm for the assessment of image fidelity. *Digital images and human vision*, pages 179–206, 1993.
- [23] J. Damon. Local Morse theory for solutions to the heat equation and Gaussian blurring. *Journal of Differential Equations*, 115(2):368–401, January 1995.
- [24] R. de Freitas Zampolo, R.; Seara. Perceptual image quality assessment based on bayesian networks. In *International Conference on Image Processing ICIP*, volume 1, pages 329–332, 2004.
- [25] Trip Denton, Jeff Abrahamson, and Ali Shokoufandeh. Approximation of canonical sets and their application to 2d view simplification. In *CVPR*, volume 2, pages 550–557, June 2004.
- [26] Trip Denton, M. Fatih Demirci, Jeff Abrahamson, and Ali Shokoufandeh. Selecting canonical views for view-based 3-d object recognition. In *ICPR*, pages 273–276, August 2004.
- [27] V. DiGesù and V. V. Staravoirov. Distance-based functions for image comparison. *Pattern Recognition Letters*, 20(2):207–213, feb. 1999.
- [28] J. Duek. Human vision models of image quality evaluation for jpeg2000 compression designed in matlab. In *Proceedings of the MATLAB 2004 Conference*, volume 1, page ???, 2004.
- [29] R. Duits. *Perceptual Organization in Image Analysis: A mathematical approach based on scale, orientation and curvature*. Phd thesis, Eindhoven University of technology, September 2005.
- [30] R. Duits, M. Felsberg, L. Florack, and B. Platel. α scale spaces on a bounded domain. In Griffin and Lillholm [55], pages 494–510.
- [31] A. P. Eckert, M. P.; Bradley. Perceptual quality metrics applied to still image compression. *Signal Processing*, 70(3):177–200, Nov. 1998.

- [32] Mortensen E.N. Vision-assisted image editing. *Computer Graphics*, 33(4):55–57, November 1999.
- [33] A. Eude, T.; Mayache. An evaluation of quality metrics for compressed images based on human visual. In *ICSP '98. 1998 Fourth International Conference on Signal Processing*, volume 1, pages 779–782, 1998.
- [34] H. Eude, T.; Cherifi. Quality metrics for low bitrate coding. In *Proceedings of the SPIE The International Society for Optical Engineering.*, volume 3016, pages 70–81, 1997.
- [35] T. Eude, A. Mayache, and C. Milan. A psychovisual quality metric based on multiscale image texture analysis. In *Proceedings of the SPIE The International Society for Optical Engineering.*, volume 3644, pages 235–244, 1999.
- [36] L. Faraut and K. Harzallah. *Deux cours d'analyse harmonique - ecole d'Ete d'Analyse Harmonique de Tunis*. Birkhaeuser, Basel, 1987.
- [37] M. Felsberg, R. Duits, and L. Florack. The monogenic scale space on a bounded domain and its applications. In Griffin and Lillholm [55], pages 209–224.
- [38] L. Florack and A. Kuijper. The topological structure of scale-space images. *Journal of Mathematical Imaging and Vision*, 12(1):65–79, February 2000.
- [39] L. M. J. Florack. *The Syntactical Structure of Scalar Images*. PhD thesis, University of Utrecht, Department of Medicine, Utrecht, The Netherlands, November 10 1993. (cum laude).
- [40] L. M. J. Florack. *Image Structure*, volume 10 of *Computational Imaging and Vision Series*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997.
- [41] L. M. J. Florack. A spatio-frequency trade-off scale for scale-space filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(9):1050–1055, September 2000.
- [42] L. M. J. Florack and R. Duits. Regularity classes for locally orderless images. In Griffin and Lillholm [55], pages 255–265.
- [43] L. M. J. Florack, B. M. ter Haar Romeny, J. J. Koenderink, and M. A. Viergever. Scale and the differential structure of images. *Image and Vision Computing*, 10(6):376–388, July/August 1992.
- [44] L. M. J. Florack, B. M. ter Haar Romeny, J. J. Koenderink, and M. A. Viergever. General intensity transformations and differential invariants. *Journal of Mathematical Imaging and Vision*, 4(2):171–187, 1994.
- [45] L. M. J. Florack, B. M. ter Haar Romeny, J. J. Koenderink, and M. A. Viergever. The Gaussian scale-space paradigm and the multiscale local jet. *International Journal of Computer Vision*, 18(1):61–75, April 1996.
- [46] L.M.J. Florack, B.J. Janssen, F.M.W. Kanters, and R. Duits. Towards a new paradigm for motion extraction. In *Proceedings of the Third International Conference on Image Analysis and Recognition, Póvoa de Varzim, Portugal, September 2006*, volume LNCS Volume 4141, pages 743–754, 2006.
- [47] ScaleSpaceViz: Software for visualizing α -scale spaces. www.bmi2.bmt.tue.nl/image-analysis/people/fkanters, 2004.

- [48] D. R. Fuhrmann, J. A. Baro, and J. R. Cox Jr. Experimental evaluation of psychophysical distortion metrics for jpeg encoded. *Journal of Electronic Imaging*, 4(4):397–406, October 1995.
- [49] M. R. Gary and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. Freeman, San Francisco, 1979. (ND2,SR1).
- [50] T. Georgiev. Covariant derivatives and vision. In Leonardis et al. [91], pages 56–69.
- [51] R. Gilmore. *Catastrophe Theory for Scientists and Engineers*. Dover Publications, Inc., New York, 1993. Originally published by John Wiley & Sons, New York, 1981.
- [52] B. Girod. Objective quality measures for the design of digital image transmission systems. In *ICASSP 81. Proceedings of the 1981 IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 3, pages 1132–1135, 1981.
- [53] M. X. Goemans. Semidefinite programming in combinatorial optimization. *Mathematical Programming*, 79:143–161, 1997.
- [54] Michel X. Goemans and David P. Williamson. 878-approximation algorithms for max cut and max 2sat. In *Twenty-sixth Annual ACM Symposium on Theory of Computing*, pages 422–431, New York, 1994.
- [55] L. D. Griffin and M. Lillholm, editors. *Scale-Space Methods in Computer Vision: Proceedings of the Fourth International Conference, Scale-Space 2003, Isle of Skye, UK*, volume 2695 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, June 2003.
- [56] B. M. ter Haar Romeny. *Front-End Vision and Multi-Scale Image Analysis: Multi-Scale Computer Vision Theory and Applications, written in Mathematica*, volume 27 of *Computational Imaging and Vision Series*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2003.
- [57] H. Hamberg, R.; de Ridder. Continuous assessment of perceptual image quality. *Journal of the Optical Society of America A Optics, Image Science and Vision*, 12(12):2573–2577, Dec 1995.
- [58] R.M. Haralick. Ridges and valleys on digital images. *Computer Vision, Graphics and Image Processing*, 22:28–38, 1983.
- [59] C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. Alvey Vision Conf., Univ. Manchester*, pages 147–151, 1988.
- [60] Kun Huang, Wei Hong, and Yi Ma. Symmetry-based photo-editing. *Pattern Recognition*, 38(6):825–834, June 2005.
- [61] T. Iijima. Basic theory on normalization of a pattern (in case of typical one-dimensional pattern). *Bulletin of Electrical Laboratory*, 26:368–388, 1962. (in Japanese).
- [62] R. E. Jacobson. Problems and solutions in quantifying image quality. In *Proceedings of ICIS'02, Tokyo, International Congress of Imaging Science*, volume 1, pages 3–4, 2002.
- [63] B. Janssen, F. M. W. Kanters, R. Duits, L. M. J. Florack, and B. M. ter Haar Romeny. A linear image reconstruction framework based on Sobolev type inner products. In Kimmel et al. [78], pages 85–96.

- [64] B. Janssen, F. M. W. Kanters, R. Duits, L. M. J. Florack, and B. M. ter Haar Romeny. A linear image reconstruction framework based on Sobolev type inner products. *International Journal of Computer Vision*, 70(3):231–240, December 2006.
- [65] B.J. Janssen, L.M.J. Florack, R. Duits, and B.M. ter Haar Romeny. Optic flow from multi-scale dynamic anchor point attributes. In *Proceedings of the Third International Conference on Image Analysis and Recognition, Póvoa de Varzim, Portugal, September 2006*, volume LNCS Volume 4141, pages 767–779, 2006.
- [66] Elder J.H. and Goldberg R.M. Image editing in the contour domain. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):291–296, March 2001.
- [67] P. Johansen. On the classification of toppoints in scale space. *Journal of Mathematical Imaging and Vision*, 4(1):57–67, 1994.
- [68] P. Johansen. Local analysis of image scale space. In J. Sporring, M. Nielsen, L. M. J. Florack, and P. Johansen, editors, *Gaussian Scale-Space Theory*, volume 8 of *Computational Imaging and Vision Series*, chapter 10, pages 139–146. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997.
- [69] P. Johansen, S. Skelboe, K. Grue, and J. D. Andersen. Representing signals by their top points in scale-space. In *Proceedings of the 8th International Conference on Pattern Recognition (Paris, France, October 1986)*, pages 215–217. IEEE Computer Society Press, 1986.
- [70] M. D. Johnson, G. M.; Fairchild. From color image difference models to image quality metrics. In *Proceedings of ICIS'02: Tokyo : International Congress of Imaging Science*, volume 1, pages 326–327, 2002.
- [71] P. Juffs, E. Beggs, and F. Deravi. A multiresolution distance measure for images. *IEEE Signal Processing Letters*, 5(6):138–140, June. 1998.
- [72] F. Kanters, L. Florack, R. Duits, and B. Platel. Scalespaceviz: Visualizing α -scale spaces. Demonstration software presentend at the Eighth European Conference on Computer Vision, Prague, Czech Republic, May 2004.
- [73] F. Kanters, L. Florack, B. Platel, and B. ter Haar Romeny. Image reconstruction from multiscale critical points. In Griffin and Lillholm [55], pages 464–478.
- [74] F. M. W. Kanters, L. M. J. Florack, R. Duits, and B. Platel. α -scale spaces in practice. *Pattern Recognition and Image Analysis*, 15(1):208–211, 2005.
- [75] F. M. W. Kanters, M. Lillholm, R. Duits, B. Janssen, B. Platel, L. M. J. Florack, and B. M. ter Haar Romeny. On image reconstruction from multiscale top points. In Kimmel et al. [78], pages 431–442.
- [76] H. Keelan, B. W.; Urabe. Iso 20462: a psychophysical image quality measurement standard. In *Proceedings of the SPIE The International Society for Optical Engineering.*, volume 5294, pages 181–189, 2003.
- [77] Bill Lorensen Ken Martin, Will Schroeder. The visualization toolkit (vtk). www.vtk.org, 1993-2006.
- [78] R. Kimmel, N. Sochen, and J. Weickert, editors. *Scale Space and PDE Methods in Computer Vision: Proceedings of the Fifth International Conference, Scale-Space 2005, Hofgeismar, Germany*, volume 3459 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, April 2005.

- [79] W. Koditz. Image processing and image analysis. *Elektronik-Praxis*, 22(3):138–140, March 1987.
- [80] J. J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.
- [81] J. J. Koenderink. *Solid Shape*. MIT Press, Cambridge, 1990.
- [82] J. J. Koenderink and A. J. van Doorn. Dynamic shape. *Biological Cybernetics*, 53:383–396, 1986.
- [83] J. J. Koenderink and A. J. van Doorn. Receptive field families. *Biological Cybernetics*, 63:291–298, 1990.
- [84] J.J. Koenderink and A.J. van Doorn. Two-plus-one-dimensional differential geometry. *Pattern Recognition Letters*, 15(5):439–443, May 1994.
- [85] A. Kuijper and L. M. J. Florack. Hierarchical pre-segmentation without prior knowledge. In *Proceedings of the 8th International Conference on Computer Vision (Vancouver, Canada, July 9–12, 2001)*, pages 487–493. IEEE Computer Society Press, 2001.
- [86] A. Kuijper and L.M.J. Florack. Understanding and modeling the evolution of critical points under gaussian blurring. In *Proceedings of the 7th European Conference on Computer Vision (ECCV)*, volume LNCS 2350, pages 143–157, 2002.
- [87] A. Kuijper and L.M.J. Florack. The hierarchical structure of images. *IEEE-Transactions-on-Image-Processing*, 12(9):1067–1079, Sept. 2003.
- [88] S. Kullback. *Information Theory and Statistics*. Wiley, New York, 1959.
- [89] Lun-Chia Kuo and Sheng-Jyh Wang. A flexible architecture for feature-based image editing. In *2005 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 1177–1180, 2005.
- [90] D. Le Callet, P.; Barba. A perceptual color image quality metric using adequate error pooling for coding scheme evaluation. In *Proceedings of the SPIE The International Society for Optical Engineering.*, volume 4662, pages 173–180, 2002.
- [91] A. Leonardis, H. Bischof, and A. Prinz, editors. *Proceedings of the Ninth European Conference on Computer Vision (Graz, Austria, May 2006)*, volume 3951–3954 of *Lecture Notes in Computer Science*, Berlin, Heidelberg, 2006. Springer-Verlag.
- [92] L. M. Lifshitz and S. M. Pizer. A multiresolution hierarchical approach to image segmentation based on intensity extrema. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6):529–541, 1990.
- [93] M. Lillholm, M. Nielsen, and L. D. Griffin. Feature-based image analysis. *International Journal of Computer Vision*, 52(2/3):73–95, 2003.
- [94] T. Lindeberg. Scale-space for discrete signals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(3):234–245, 1990.
- [95] T. Lindeberg. Scale-space behaviour of local extrema and blobs. *Journal of Mathematical Imaging and Vision*, 1(1):65–99, March 1992.
- [96] T. Lindeberg. On scale selection for differential operators. In *Proc. 8th Scandinavian Conference on Image Analysis*, volume 1, pages 857–866, 1993.

- [97] T. Lindeberg. *Scale-Space Theory in Computer Vision*. The Kluwer International Series in Engineering and Computer Science. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.
- [98] T. Lindeberg. Edge detection and ridge detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):117–156, November 1998.
- [99] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116, Feb. 1998.
- [100] Tony Lindeberg. Detecting salient blob-like image structures and their scales with a scale space primal sketch:a method for focus-of-attention. *International Journal of Computer Vision*, 11(3):283–318, March 1993.
- [101] A. Longhurst, P.; Chalmers. User validation of image quality assessment algorithms. In *Proceedings. Theory and Practice of Computer Graphics.*, volume 1, pages 196–202, 2004.
- [102] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *Computer Graphics (Proceedings of SIGGRAPH '87)*, 21(4):163–169, 1987.
- [103] D. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157, 1999.
- [104] D. G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [105] V. Martens, J. B.; Kayargadde. Image quality prediction in a multidimensional perceptual space. In *Proceedings. International Conference on Image Processing*, volume 1, pages 877–880, 1996.
- [106] O. Matsushiro, N.; Asada. Objective quality measure of halftone images based on optimal pixel assignment. In *Proceedings of the SPIE The International Society for Optical Engineering.*, volume 2657, pages 437–447, 1996.
- [107] A. Mayache, T. Eude, and H. Cherifi. A comparison of image quality models and metrics based on human visual sensitivity. In *Proceedings 1998 International Conference on Image Processing.*, volume 3, pages 409–413, 1998.
- [108] K. Mikolajczyk and C. Schmid. Software for calculating affine hessian and affine harris points. <http://www.robots.ox.ac.uk/vgg/research/affine/detectors.html>.
- [109] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, October 2004.
- [110] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, October 2005.
- [111] Krystian Mikolajczyk, Bastian Leibe, and Bernt Schiele. Local features for object class recognition. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 2*, pages 1792–1799, Washington, DC, USA, 2005. IEEE Computer Society.
- [112] Diakopoulos N., Essa M., and Ramesh Jain. Content based image synthesis. In *Third International Conference on Image and Video Retrieval CIVR-2004.*, volume LNCS 3115, pages 299–307. Springer-Verlag, 2004.

- [113] Kondo N. and Nakashizuka M. Contour domain image editing based on a wavelet transform. *Journal of the Institute of Image Information and Television Engineers*, 59(2):302–308, February 2005. In Japanese.
- [114] S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (coil-20). Technical report, Technical Report, Columbia University, <http://www.cs.columbia.edu/CAVE/>, 1996.
- [115] M. Nielsen and M. Lillholm. What do features tell about images? In M. Kerckhove, editor, *Scale-Space and Morphology in Computer Vision: Proceedings of the Third International Conference, Scale-Space 2001, Vancouver, Canada*, volume 2106 of *Lecture Notes in Computer Science*, pages 39–50. Springer-Verlag, Berlin, July 2001.
- [116] John Novatnack, Trip Denton, Ali Shokoufandeh, and Lars Bretzner. Stable bounded canonical sets and image matching. In *Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, pages 316–331, November 2005.
- [117] Perez P., Gangnet M., and Blake A. Poisson image editing. *ACM Transactions on Graphics*, 22(3):313–318, July 2003.
- [118] E. J. Pauwels, L. J. Van Gool, P. Fiddelaers, and T. Moons. An extended class of scale-invariant and recursive scale space filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(7):691–701, July 1995.
- [119] B. Platel, E. G. Balmachnova, L. M. J. Florack, and B. M. ter Haar Romeny. Top-points as interest points for image matching. In Leonardis et al. [91], pages 418–429.
- [120] B. Platel, L. M. J. Florack, F. M. W. Kanters, and E. G. Balmachnova. Using multiscale top points in image matching. In *Proceedings of the 11th International Conference on Image Processing (Singapore, October 24–27, 2004)*, pages 389–392. IEEE, 2004.
- [121] A. M. Pons, J. Malo, J. M. Artigas, and P. Capilla. Image quality metric based on multidimensional contrast perception models. *Displays*, 20(2):93–110, Aug. 1999.
- [122] W.K. Pratt. *Digital Image Processing*. Wiley, New York, 1978.
- [123] Wang Yuanji; Li Jianhua; Lu Yi; Fu Yao; Jiang Qinzong. Image quality evaluation based on image weighted separating block peak signal to noise ratio. In *Proceedings of 2003 International Conference on Neural Networks and Signal Processing*, volume 2, pages 994–997, 2003.
- [124] Mahesh Ramasubramanian, Sumanta N. Pattanaik, and Donald P. Greenberg. A perceptually based physical error metric for realistic image synthesis. In Alyn Rockwood, editor, *Siggraph 1999, Computer Graphics Proceedings*, pages 73–82, Los Angeles, 1999. Addison Wesley Longman.
- [125] A.M. Rohaly, A. J. Ahumada Jr., and A. B. Watson. A comparison of image quality models and metrics predicting object detection. In *SID International Symposium Digest of Technical Papers*, volume 1, pages 45–48, 1995.
- [126] A. Shnayderman, A. Gusev, and A. M. Eskicioglu. Multidimensional image quality measure using singular value decomposition. In *Proceedings of the SPIE The International Society for Optical Engineering.*, volume 5294(1), pages 82–92, 2003.

- [127] Guangsheng Chen Shutao Li. Clarity ranking for digital images. In *Fuzzy Systems and Knowledge Discovery. Second International Conference, FSKD 2005. Proceedings, Part II Lecture Notes in Artificial Intelligence*, volume 3614, pages 610–613, 2005.
- [128] Kar-Han Tan and Ahuja N. Selecting objects with freehand sketches. In *Proceedings of the Eighth IEEE International Conference on Computer Vision ICCV-2001*, volume 1, pages 337–344, 2001.
- [129] R. Thom. *Structural Stability and Morphogenesis (translated by D. H. Fowler)*. Benjamin-Addison Wesley, New York, 1975.
- [130] A. Tikhonov and V. Y. Arseninn. *Solution of Ill-Posed Problems*. John Wiley & Sons, New York, 1977.
- [131] D. Travis, T. F. M. Stewart, and C. Mackay. Evaluating image quality. *Displays, Technology and Applications.*, 13(3):139–147, March 1992.
- [132] D. Van der Weken, M. Nachtegael, and E. Kerre. Improved image quality measures using ordered histograms. In *IEEE 6th Workshop on Multimedia Signal Processing.*, volume 1, pages 67–70, 2004.
- [133] Dietrich Van der Weken, Mike Nachtegael, and Etienne E. Kerre. Using similarity measures and homogeneity for the comparison of images. *Image and Vision Computing*, 22(9):695–702, 2004.
- [134] van der Schaaff A. van Hateren J.H. Independent component filters of natural images compared with simple cells in primary visual cortex. In *Proc. Royal Soc. Lond. B*, volume 265, pages 359–366, 1997.
- [135] Vijay V. Vazirani. *Approximation Algorithms*. Springer-Verlag, Berlin, Germany, second edition, 2003. ISBN 3-540-65367-8.
- [136] Barrett W.A. and Cheney A.S. Object-based image editing. In *ACM Transactions on Graphics*, volume 21(3), pages 777–784, 2002.
- [137] Sheng-Jyh Wang, Lun-Chia Kuo, Hsin-Haw Jong, and Zong-Han Wu. Representing images using points on image surfaces. *IEEE Transactions on Image Processing*, 14(8):1043–1056, August 2005.
- [138] A. B. Watson. Dctune 2.0 software. Website, <http://vision.arc.nasa.gov/dctune/>.
- [139] A. B. Watson. Dctune: A technique for visual optimization of dct quantization matrices for individual images. In *J. Morreale, ed., SID Digest 24 (Society for Information Display: Santa Ana, CA)*, volume 24, pages 946–949, 1993.
- [140] S. J. P. Westen, R. L. Lagendijk, and J. Biemond. Perceptual image quality based on a multiple channel hvs model. In *International Conference on Acoustics, Speech, and Signal-Processing. Conference Proceedings*, volume 4, pages 2351–2354, 1995.
- [141] A. P. Witkin. Scale-space filtering. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1019–1022, Karlsruhe, Germany, 1983.
- [142] S. Wolfram. *Mathematica: A System for doing Mathematics by Computer*. Addison-Wesley, second edition, 1991.
- [143] Chenyu Wu, Ce Liu, Heung Yueng Shum, Ying Qing Xy, and Zhengyou Zhang. Automatic eyeglasses removal from face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3):322–336, March 2004.

- [144] G. Xu, W.; Hauske. Picture quality evaluation based on error segmentation. In *Proceedings of the SPIE The International Society for Optical Engineering.*, volume 2308(3), pages 1454–1465, 1994.
- [145] R. A. Young. The Gaussian derivative model for machine vision: Visual cortex simulation. *Journal of the Optical Society of America*, July 1986.
- [146] A. C. Bovik Z. Wang. A universal image quality index. *IEEE Signal Processing Letters*, 9(3):81–84, March 2002.
- [147] H. R. Sheikh Z. Wang, A. C. Bovik and E. P. Simoncelli. Ssim software. <http://www.cns.nyu.edu/~lcv/ssim/>.
- [148] H. R. Sheikh Z. Wang, A. C. Bovik and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2002.
- [149] Yun Zeng, Wei Chen, and Qun Sheiig Peng. A novel variational image model: towards a unified approach to image editing. *Journal of Computer Science and Technology English Language Edition*, 21(2):224–231, 2006.
- [150] Xuemei Zhang, D. A. Silverstein, J. E. Farrell, and B. A. Matsushiro1996 Wandell. Color image quality metric s-cielab and its application on halftone texture visibility. In *Forty Second IEEE Computer Society International Conference Proceedings*, volume 97CB36028, pages 44–48, 1997.
- [151] E. P. Zhou Wang; Simoncelli. Stimulus synthesis for efficient evaluation and refinement of perceptual image quality metrics. In *Proceedings of the SPIE The International Society for Optical Engineering*, volume 5292(1), pages 99–108, 2004.

Samenvatting

Digitale beelden nemen een steeds belangrijkere rol in de maatschappij in. Denk bijvoorbeeld aan artsen die met behulp van computers bewerkte röntgen foto's bekijken of het bewerken van beelden van een digitale camera. Veel software pakketten gebruikt voor beeldbewerking, zoals bijvoorbeeld Adobe PhotoShopTM, zijn gebaseerd op het direct manipuleren van pixels (beeld elementen) of groepen daarvan. Echter voor sommige toepassingen is deze pixel representatie van een beeld niet geschikt. Bijvoorbeeld voor het vervangen van appels in een scène door peren is een representatie op object niveau noodzakelijk. Momenteel is er geen enkel pakket beschikbaar dat beeldbewerking op object niveau mogelijk maakt. Dit proefschrift beschrijft een alternatieve beeld representatie gebaseerd op scale space theorie, die gebruikt kan worden om objecten in een beeld te lokaliseren en die gebruikt kan worden om een beeld aan te passen. Hiermee zetten we een nieuwe stap richting object gebaseerde beeldbewerking. De scale space representatie van een beeld is geïnspireerd door het menselijk visueel systeem. Het menselijk visueel systeem bekijkt een beeld op verschillende schalen. Neem bijvoorbeeld een beeldmozaïek, waarbij op een kleine afstand (lage schaal) elk detail goed te zien is, maar het grote geheel niet, terwijl op een grote afstand (hoge schaal) het grote geheel goed te zien is, maar de details niet. Omdat van te voren niet bekend is in welke schaal we geïnteresseerd zijn, dienen alle schalen tegelijk te worden beschouwd. In de beeldbewerking wordt het effect van schaal gemaakt door een beeld te vervagen. Het is ook aangetoond dat het menselijk visueel systeem tot 4e orde spatiële afgeleiden neemt van een beeld op verschillende schalen. Het blijkt dat dit belangrijk is voor het detecteren van objecten, zowel voor mensen als voor computer systemen.

In hoofdstuk 2 wordt de zogenaamde α -scale space geïntroduceerd, een klasse van scale spaces waarvan de bekende Gaussische scale space een specifiek geval is (namelijk $\alpha = 1$). Een probleem bij deze α -scale space is dat er geen uitdrukking bestaat voor de α -kern in het spatiële domein voor algemene α , alleen in het Fourier domein bestaat een dergelijke uitdrukking. Omdat in de praktijk de spatiële kern vaak wordt gebruikt, presenteren we in hoofdstuk 2 een benadering van de α -kern in het spatiële domein voor algemene α . Deze benadering is geïmplementeerd in een programma, ScaleSpaceViz genaamd. ScaleSpaceViz is bedoeld voor het uitrekenen en visualiseren van α -scale spaces, met name om een beter inzicht te krijgen in de verschillende speciale punten die zich in een α -scale space bevinden. Deze punten, zogenaamde "scale space interest points", bevatten informatie over een beeld en kunnen gebruikt worden om een beeld te representeren. Het is aangetoond dat deze punten geschikt zijn voor het detecteren van objecten in een beeld en gegeven deze representatie kan een redelijke benadering van het oorspronkelijke beeld worden gemaakt. In dit proefschrift zal deze scale space interest point representatie van een beeld een centrale rol innemen.

In hoofdstuk 3 worden drie reconstructie algoritmen met elkaar vergeleken. Deze reconstructie algoritmen genereren, gegeven een aantal scale space interest points, een benadering van het oorspronkelijke beeld waar de scale space punten van zijn uitgerekend. Het eerste algoritme is eenvoudig te implementeren, maar geeft een visueel onaantrekkelijk resultaat dat niet veel lijkt op het oorspronkelijke beeld indien weinig punten worden gebruikt voor de reconstructie. Het tweede algoritme probeert een glad beeld te maken wat een veel aantrekkelijker resultaat oplevert. Deze methode gebruikt echter een iteratieve implementatie die veel langzamer is. In hoofdstuk 3 presenteren we

een algoritme dat het beeld glad maakt, maar toch eenvoudig en snel te implementeren is. Het algoritme is net iets langzamer dan het eerste algoritme, maar de beeldkwaliteit komt dicht bij het tweede algoritme.

Om de reconstructies van scale space punten met elkaar te kunnen vergelijken op een reproduceerbare manier, zijn objectieve foutmaten nodig. In hoofdstuk 4 zijn 34 kwaliteitsmaten voor beelden met elkaar vergeleken aan de hand van een testset die beoordeeld is door 33 personen. Uit de resultaten blijkt dat veel foutmaten redelijk overeenkomen met de menselijke beoordelingen indien het om een enkele soort afwijking gaat (b.v. ruis of jpeg artefacten). Slechts enkele, complexere foutmaten gebaseerd op het menselijk visueel systeem, blijken ook geschikt voor het evalueren van beelden met zeer diverse afwijkingen zoals bij beeld reconstructie. In hoofdstuk 4 presenteren we een nieuwe foutmaat gebaseerd op scale space theorie, waarbij op verschillende schalen naar spatiële afgeleiden van het beeld wordt gekeken. Deze foutmaat is wiskundig gezien eenvoudig en volgt de menselijke beoordeling van de beelden redelijk, vergelijkbaar met de meer complexe, visueel systeem gebaseerde methoden.

In hoofdstuk 5 worden 10 verschillende soorten scale space punten vergeleken met betrekking tot de geschiktheid voor beeld reconstructie. De vergelijking is uitgevoerd op twee datasets; de eerste set bestaat uit 8.000 willekeurige stukjes (van 64×64 pixels) van beelden uit de van Hateren database voor natuurlijke beelden en de tweede dataset bestaat uit 12 beelden gemaakt met een digitale camera. Voor alle beelden uit de twee datasets zijn alle 10 verschillende soorten scale space punten berekend en er zijn reconstructies gemaakt van al deze puntverzamelingen. De resultaten zijn geëvalueerd met behulp van de in hoofdstuk 4 geïntroduceerde foutmaten. Uit de resultaten blijkt dat bepaalde punten, b.v. hoekpunten en blobs beter geschikt zijn voor beeld reconstructie dan andere punten zoals b.v. top punten of scale space zadelpunten. De meeste punten kunnen gerangschikt worden aan de hand van de *sterkte* van de punten. Echter, deze sterkte is verschillend voor de verschillende soorten punten en kunnen vaak niet direct met elkaar worden vergeleken. Het is daarom lastig om een zinvolle deelverzameling te nemen van een set gecombineerde scale space punten. In hoofdstuk 5 wordt een nieuwe methode gepresenteerd die gebruikt kan worden om een goede subset van gecombineerde scale space punten te vinden, rekening houdend met de sterkte van de punten en de spatiële distributie van de punten. Resultaten laten zien dat deze methode betere reconstructies oplevert dan een rangschikking van de gecombineerde punten aan de hand van de sterkte alleen.

In hoofdstuk 6 presenteren we een haalbaarheids-studie voor het gebruik van de scale space interest point representatie van een beeld voor beeldbewerking. In eerder werk van verschillende auteurs is aangetoond dat scale space punten geschikt zijn voor het detecteren van objecten in een complexe scène. In dit proefschrift is aangetoond dat scale space punten ook gebruikt kunnen worden om de inhoud van beelden aan te passen, door b.v. objecten uit een beeld te verwijderen, objecten in een beeld te plaatsen of objecten te vervangen. Door het combineren van detectie van objecten en het bewerken van beelden aan de hand van scale space punten, kunnen beelden op een object niveau worden aangepast in plaats van op pixel niveau. Een aantal mogelijke toepassingen op het gebied van beeldbewerking, beeldsynthese, advertenties, animatie en medische toepassingen zijn gegeven in hoofdstuk 6. Hoewel nog veel extra onderzoek nodig is om deze toepassingen te realiseren met het gepresenteerde raamwerk, heeft dit proefschrift een extra stap richting object gebaseerde beeldbewerking gezet.

List of Publications

2003

- [1] F.M.W. Kanters, L.M.J. Florack, B. Platel, and B.M. ter Haar Romeny. Image Reconstruction from Multiscale Critical Points. In *"Scale Space Methods in Computer Vision". Proc. of the 4th int. conference Scale Space 2003*, Isle of Skye, UK, June 2003. pp 464–478.
- [2] F.M.W. Kanters, B. Platel, L.M.J. Florack, and B.M. ter Haar Romeny. Content Based Image Retrieval using Multiscale Top Points. In *"Scale Space Methods in Computer Vision". Proc. of the 4th int. conference Scale Space 2003*, Isle of Skye, UK, June 2003. pp 33–43.

2004

- [3] F.M.W. Kanters, L.M.J. Florack, R. Duits, and B. Platel. Alpha Scale Space Kernels in Practice. In *"New Information Technologies". Proc. of the 7th int. conf. on Pattern Recognition and Image Analysis*, St. Petersburg, Russian Federation, Oct 2004. pp 260–263.
- [4] B. Platel, F.M.W. Kanters, L.M.J. Florack, E. Balmachnova. Using Multiscale Top Points in Image Matching. In *11th International Conference on Image Processing; Singapore*, Singapore, October 2004, pp 389–392.

2005

- [5] R. Duits, B.J. Janssen, F.M.W. Kanters, and L.M.J. Florack. Linear Image Reconstruction from a Sparse Set of alpha-Scale Space Features by Means of Inner Products of Sobolev Type. *Proc. on the 1st Int. Workshop DSSCV 2005*, Maastricht, the Netherlands, June 2005, Lecture notes in computer science, 3753, pp 96–111.
- [6] B. Platel, E. Balmachnova, L.M.J. Florack, F.M.W. Kanters, and B.M. ter Haar Romeny. Using Top Points as Interest Points for Image Matching. *Proc. on the 1st Int. Workshop DSSCV 2005*, Maastricht, the Netherlands, June 2005, Lecture notes in computer science, 3753, pp 211–222.
- [7] R. Duits, F.M.W. Kanters, L.M.J. Florack, and B.M. ter Haar Romeny. A Comparison of the Deep Structure of alpha-Scale Spaces. *Proc. on the 1st Int. Workshop DSSCV 2005*, Maastricht, the Netherlands, June 2005, Lecture notes in computer science, 3753, pp 234–248.
- [8] B.J. Janssen, F.M.W. Kanters, R. Duits, L.M.J. Florack, and B.M. ter Haar Romeny. A Linear Image Reconstruction Framework Based on Sobolev Type Inner Products. *Proc. on the 5th Int. Conference on Scale Space and PDE methods*, Hofgeismar, Germany, June 2005, Lecture notes in computer science, 3459, pp 85–95.
- [9] B. Platel, M. Fatih Demirci, A. Shokoufandeh, L.M.J. Florack, F.M.W. Kanters, B.M. ter Haar Romeny, and S. Dickinson. Discrete Representation of Top Points via Scale Space Tessellation. *Proc. on the 5th Int. Conference on Scale Space and PDE methods*, Hofgeismar, Germany, June 2005, Lecture notes in computer science, 3459, pp 73–80.

- [10] F.M.W. Kanters, M. Lillholm, R. Duits, B.J.P. Jansen, B. Platel, L.M.J. Florack, and B.M. ter Haar Romeny. On Image Reconstruction from Multiscale Top Point. *Proc. on the 5th Int. Conference on Scale Space and PDE methods*, Hofgeismar, Germany, June 2005, Lecture notes in computer science, 3459, pp 431-439.
- [11] E. Balmachnova, L.M.J. Florack, B. Platel, F.M.W. Kanters, and B.M. ter Haar Romeny. Stability of Top-Points in Scale Space. *Proc. on the 5th Int. Conference on Scale Space and PDE methods*, Hofgeismar, Germany, June 2005, Lecture notes in computer science, 3459, pp 62-70.

2006

- [12] B.J. Janssen, F.M.W. Kanters, R. Duits, L.M.J. Florack, and B.M. ter Haar Romeny. A Linear Image Reconstruction Framework Based on Sobolev Type Inner Products. *International Journal of Computer Vision (IJCV)*, Volume 70, Number 3, pp 231-240, December 2006.
- [13] L.M.J. Florack, B.J. Janssen, F.M.W. Kanters, and R. Duits. Towards a New Paradigm for Motion Extraction. In *ICIAR 2006: Proceedings of the Third International Conference on Image Analysis and Recognition*, Póvoa de Varzim, Portugal, September 2006, LNCS Volume 4141, pp 743-754.
- [14] F.M.W. Kanters, L.M.J. Florack, B. Platel and B.M. ter Haar Romeny. Multi-Scale Differential Error: A Novel Image Quality Assessment Tool. *Proc. of the 8th Int. conf. on Signal and Image Processing 2006*, Honolulu, Hawaii, pp 188-194, August 2006.

2007

- [15] F.M.W. Kanters, L.M.J. Florack, R. Duits, B. Platel, and B.M. ter Haar Romeny. ScaleSpaceViz: α -Scale Spaces in Practice. *Pattern Recognition and Image Analysis*, MAIK "Nauka/Interperiodica", Pleiades Publishing, vol. 17, no. 1, 2007

Submitted / in preparation

- [16] F.M.W. Kanters, T. Denton, A. Shokoufandeh, L.M.J. Florack, and B.M. ter Haar Romeny. Canonical Sets in Image Reconstruction from Multiscale Interest Points. Submitted to *First International Conference on Scale Space Methods and Variational Methods in Computer Vision*.
- [17] T. Denton, F.M.W. Kanters, A. Shokoufandeh, L.M.J. Florack. Image Reconstruction with Canonical Subsets of Multiple Types of Scale Space Interest Points. Submitted to *IEEE Conf. on Comp. Vision and Pattern Recogn.*
- [18] F.M.W. Kanters, L.M.J. Florack, B. Platel and B.M. ter Haar Romeny. Comparison of Image Quality Measures for Evaluating Image Reconstruction. In preparation.
- [19] F.M.W. Kanters and M. Lillholm. A Comparison of Multi-scale Interest Points used for Image Reconstruction. In preparation.
- [20] F.M.W. Kanters, L.M.J. Florack, B. Platel and B.M. ter Haar Romeny. Image Editing using Scale Space Interest Points. In preparation.

Acknowledgements

This thesis would not have existed without the help of many people that I would like to thank here. First I would like to thank prof.dr.ir. Paul v.d. Bosch, for bringing me in contact with the Biomedical Image Analysis group and for accepting the invitation to join the committee. Next I would like to thank my supervisors, prof.dr.ir. Bart ter Haar Romeny and dr. Luc Florack, without their help this thesis would be quite empty. Thanks to Bart for embracing me in his great research group, for his endless enthusiasm and support, and for all the great conversations we had about work and non-work related topics. I could always walk into his room to discuss any topic I liked, which I appreciated very much! Thanks to Luc for his mathematical insights and ideas, for our many fruitful discussions, for correcting very thoroughly all the papers I wrote (which must have cost a great amount of red pencils), and last but not least for introducing me to the great pleasure of good food and wine. Bart and Luc, many of my professional skills I learned from you during the past 5 years!

I also would like to thank all my great colleagues of the Biomedical Image Analysis group for their support and discussion. A special thanks to Bram Platel, who has been working together with me on the “DSSCV” project. Besides our many technical discussions about the *deep structure* of images, I remember all the good times we had during our visits to foreign countries. Also a special thanks to Remco Duits, who taught me a lot of mathematics, necessary to survive in this world ;). Many parts of this thesis are inspired or based on his ideas. Also thanks for our great discussions, both at the university as well as during our walks through St. Petersburg. Special thanks to my room mates, Evguenia, Bart and Erik for their support, discussion and all the fun we had. I also want to thank Anna, Markus, Arnold, Henri, Sander, Petr, Johannes, Thorsten and Hans for their support.

One of the privileges of participating in a European project is the possibility to meet many interesting people from all over the world. During literature research for my M.Sc. thesis I found the names of important people that contributed substantially to the field. Many people of that list I have met and some even have become personal friends in the past four years. To start I would like to thank prof. Mads Nielsen from the IT-University in Copenhagen. He was the project leader of the “DSSCV” project. Mads, thanks for all the discussions we had and the great hospitality we received in Copenhagen every time we visited you! I also want to thank the other members of the DSSCV consortium, Ole Fogh Olsen, Arjan Kuijper, Martin Lillholm, Anna Östlin, Philip Bille, and Inge Li Gørtz from the IT-University of Copenhagen, Peter Giblin and Andre Diatta from the University of Liverpool, and Jon Sparring and Kerawit Somchaipeng from the University of Copenhagen, for their support and many discussions during our meetings. I learned a lot from all of you and it was a great pleasure to work with you.

Furthermore I would like to thank professor Sven Dickinson and dr. Ali Shokoufandeh, which we first met in Copenhagen during a course some years ago. During our project we have had several collaborations and it was always a great pleasure to work with you. In these years we have become friends, instead of just colleagues. I also want to thank Martin Lilholm from the IT-University of Copenhagen and Trip Denton from Drexel University in Philadelphia who helped a great deal with chapter 5 of this thesis. Thanks also to Peter Johansen for his scientific input in the project and for his hospitality during my visits in Copenhagen. I want to thank Marco Loog for the discussions

about the use of all my research and for keeping me sharp ;). Finally I want to thank all colleague researchers that contributed in some way to this thesis and are not mentioned here by name.

I never would have been able to accomplish all this without the endless support and understanding of my parents Wim and Hanny, who always believed in me and supported every crazy idea I have ever had. I also want to thank my sister Erika and my brother Jos for their support and patience. Thanks to all family and friends for their involvement and for keeping me from talking too much about work. Last but not least I want to thank Daniëlle for her endless support, for believing in me, for listening to all my stories about work, and for her patience, especially during the last few months of writing this thesis. Daniëlle, thanks for everything!

This work is part of the DSSCV project supported by the IST Programme of the European Union (IST-2001-35443).

Curriculum Vitae

Frans Kanters was born in Beek en Donk, the Netherlands, in 1978. He graduated from the VWO at the Commandrij College in Gemert in 1996. In the same year he started his scientific education at the Electrical Engineering Department of the Eindhoven University of Technology (TU/e), from which he received his M.Sc. degree in 2002. His graduation work was a feasibility study on the usability of scale space interest points for image matching and reconstruction that formed the basis for this PhD thesis, which he started in October 2002. In December 2005, he started a company called Inviso B.V. with the goal to bridge the gap between small companies and universities in the field of industrial image processing. He received a 25.000 euro STW Valorisation Grant in 2006 to implement the fundamental orientation score theory in a scanning system for stone slabs. His research interests include scale space theory, orientation score theory, texture analysis, hardware implementations of image processing algorithms (specifically on fpga's) and industrial image processing applications.